Република Србија

КРИМИНАЛИСТИЧКО-ПОЛИЦИЈСКИ УНИВЕРЗИТЕТ

Број: 43/1-4

Датум: 4.7.2022

ВЕЋУ ДЕПАРТМАНА ИНФОРМАТИКЕ И РАЧУНАРСТВА
КРИМИНАЛИСТИЧКО-ПОЛИЦИЈСКОГ УНИВЕРЗИТЕТА

Предмет: Извештај о урађеној докторској дисертацији докторанда мр. Игора Вуковића

Поштовани чланови Већа департмана информатике и рачунарства,

одлуком Већа научних области природно-математичких и техничко-технолошких студија Криминалистичко - полицијског универзитета 28 Бр. 85/4-2-2022 од 28.6.2022. године, именовани смо за чланове Комисије за оцену докторске дисертације докторанда мр. Игора Вуковића под насловом

Разрешавање идентитета и груписање дигиталних доказа о осумњиченима применом технологија препознавања лица и система софтверских интелигентних агената заснованог на неаксиоматском резоновању

(Identity resolving and grouping of digital evidence of suspects using face recognition technologies and software intelligent agents system based on non-axiomatic reasoning)

После прегледа достављене дисертације, Комисија подноси следећи

ИЗВЕШТАЈ

1. УВОД

1.1. Хронологија одобравања и израде дисертације

Мр. Игор Вуковић је докторске академске студије Информатике уписао на Департману информатике и рачунарства КПУ школске 2018/19 године.

Након што је положио све испите предвиђене студијским програмом са просечном оценом 10,00 поднео је пријаву теме докторске дисертације број 124/7 дана 16.9.2021. године, за одобравање теме докторске дисертације из научне области Рачунарске науке, Електротехничко и рачунарско инжењерство.

Веће научних области природно-математичких и техничко-технолошких студија (Одлуком 28 бр. 85/5-2-2021 од 17.11.2021. године) прихватило је тему докторске дисертације под наведеним називом и студенту именовало ментора проф. др Бранкицу Поповић.

У складу са чланом 25. Правилника о докторским студијама студент мр Игор Вуковић дана 15.6.2022. године, завршену докторску дисертацију поднео је за оцену надлежном Већу департмана, уз прилагање потребних доказа.

Сходно члану 25. став 2. ментор је дана 22.6.2022. године дала писану сагласност (број 43/1-2), којом је констатовала да је докторска дисертација студента мр Игора Вуковића подобна за оцену и да студент има најмање један рад који је повезан са садржајем докторске дисертације у коме је он први аутор, објављен у часопису са импакт-фактором са SCI листе.

У складу са чланом 27. Правилника о докторским студијама извршена је провера оригиналности докторске дисертације одговарајућим софтвером. Сходно члану 27. став 4. ментор је дана 22.6.2022. године писано мишљење (број 43/1-3) о оцени процене оригиналности и резултате софтверсе анализе доставила руководиоцу Департмана информатике и рачунарства, којим је констатовала да је **докторска дисертација оригиналан резултат научног истраживања које је докторанд спровео и да су у истој у потпуности испоштована академска правила цитирања и навођења извора**.

По добијању позитивног извештаја, Веће департмана информатике и рачунарства је на седници одржаној 27.6.2022. године донело Предлог одлуке (24 бр. 81/7-10-2022) о образовању Комисије за оцену докторске дисертације у саставу:

- др. Кристијан Кук, председник, ванредни професор Криминалистичко-полицијског универзитета,
- др. Петар Чисар, члан, редовни професор Криминалистичко-полицијског универзитета,
- др Милош Банђур, члан, ванредни професор на Факултету техничких наука Универзитета Приштина, са седиштем у Косовској Митровици

Веће научних области природно-математичких и техничко-технолошких студија Криминалистичко - полицијског универзитета је Одлуком 28 Бр. 85/4-2-2022 од 28.6.2022. године, образовало Комисију за оцену докторске дисертације докторанда мр. Игора Вуковића у наведеном саставу.

1.2. Научна област дисертације

Докторске студије Информатике на КПУ су акредитоване у оквиру образовно-научног поља Природно-математичких наука, научна област **Рачунарске науке**, па констатујемо да је докторска дисертација пријављена у научној области за коју је Криминалистичко-полицијски униврситет матичан:

Научна област: Рачунарске науке, Електротехничко и рачунарско инжењерство
Ужа научна област: Информационе технологије, Софтверско инжењерство

Предложени ментор, проф. др Бранкица Поповић, је редовни професор на Департману информатике и рачунарства КПУ. Задовољава услове неопходне за ментора и налази се на Листи ментора Департмана информатике и рачунарства КПУ. Има избор у ужој научној области Информационе технологије, а њен научи опус је мултидисциплинаран.

Увидом у базу Конзорцијума библиотека Србије за обједињену набавку (KoBSON) види се да има 19 репрезентативних референци од којих је 13 радова објављених у часописима са СЦИ листе, доминантно из области информационих технологија и софтверског инжењерства. Увидом у базу SCOPUS види се листа од 21. рада који су укупно цитирани 62 пута (хетероцитати), уз остварени h-index 4.

1.3. Биографски подаци о кандидату

Докторанд мр Игор Вуковић је рођен 07.07.1978. године у Зрењанину. Војну академију Универзитета у Београду завршио је 2001. године са просечном оценом 8,13 чиме је стекао звање Дипломирани официр везе.

Магистарске студије је завршио 2010. године на Техничком факултету 'Михајло Пупин' у Зрењанину, Универзитета у Новом Саду, са оствареном просечном оценом у току студија 9,66. Одбраном магистарске тезе под насловом **'Развој информационог система за прикупљање и криминалистичку обраду дигиталних доказа'**, на студијском програму '**Информатика**', стекао је звање **Магистра техничких наука**. Докторске студије информатике је уписао на Департману информатике и рачунарства КПУ школске 2018/19 године.

Запослен је у Министарству унутрашњих послова Републике Србије, на радном месту начелника одељења.

Докторанд мр Игор Вуковић је на КПУ био изабран за сарадника ван радног односа (2019.-2021.године) где је у том својству школске 2019/2020 и 2020/2021 изводио вежбе на предмету Безбедност података и информација (на 3. години Основних академских студија информатике и рачунарства). У својству истакнутог стручњака из праксе, континуирано од 2015. године изводи гостујуће предавање на предмету Дигитална форензика који се слуша на Мастер академским студијама Информатике и рачунарства КПУ.

Учествовао је у реализацији међународног научноистраживачког пројекта под називом '*Scalable Privacy preserving Intelligence analysis for Resolving Identities - SPIRIT*' који је у периоду 2018.-2021. године реализован у оквиру програма Европске уније за истраживање и иновације - Хоризонт 2020 (*Horizon* 2020).

Докторанд мр Игор Вуковић је до сада објавио 7 радова (6 као првопотписани аутор) следећих категорија: М14 (1), М22 (1), М23 (1), М33(3), М53 (1). Додатно је коаутор практикума под називом 'Безбедност података и информација' за истоимени предмет који се слуша на Основним академским студијама информатике и рачунарства на Криминалистичко-полицијском универзитету.

Имао је више студијских боравака у иностранству ради усавршавања где се посебно издвајају завршени курсеви из области дигиталне форензике:

- општи (Берлин, Немачка 2007., Сеул, Јужна Кореја 2010., Мајнц, Немачка 2010.)
- дигитална форензика Линукс оперативног система (Мајнц, Немачка 2011.)
- употреба форензичког софтвера X-Ways (Мајнц, Немачка 2011.)
- дигитална форензика мобилних телефона – основни ниво (Мајнц, Немачка 2011.)
- дигитална форензика Windows оперативног система (Мајнц, Немачка 2012.)
- дигитална форензика мобилних телефона – напредни ниво (Мајнц, Немачка 2013.)
- рачунарски криминал (Валбандон, Хрватска 2013.)
- посебни криминалистички послови и криминалистичко обавештајни послови (Валбандон, Хрватска 2013.)
- употреба форензичког софтвера AccessData Forensic Toolkit (Опатија, Хрватска 2014.)
- дигитална форензика интернет артефаката (Загреб, Хрватска 2014.)
- извршавање кривичних дела на интернету (Хангџоу, 2017.)
- анализа саобраћаја у бежичним рачунарским мрежама (Штокхолм 2017., Скотсдејл 2018.)

2. ОПИС ДИСЕРТАЦИЈЕ

2.1. Садржај дисертације

Докторска дисертација писана на енглеском језику има 140 страница и састоји се од следећих целина:

Списак акронима, табела, слика и листинга

1. **Увод**
2. **Преглед сродних истраживања**
3. **Материјали и методе**
4. **Прикупљање дигиталних доказа за презентацију и разрешавање идентитета помоћу алата отвореног кода**
5. **Утицај техника побољшања слике на ефикасност детекције и подударања слика лица добијених у неконтролисаним условима**

6. **Груписање по истинитосној вредности слика лица добијених у неконтролисаним условима**
7. **Мулти-агентни систем за напредне полицијске активности (mASAP)**
8. **Закључак**

Списак литературе

Дисертација садржи укупно 65 тематски адекватних слика, 8 табела и 5 листинга кода, а Списак литературе садржи 135 репрезентативних библиографских јединица које су доминантно новијег датума (после 2015. године).

2.2. Кратак приказ појединачних поглавља

У првом поглављу се елаборира позадина и мотивација истраживања, указујући на потребу полицијских служби (пре свих криминалистичке полиције) за поузданим разрешењем идентитета осумњичених, а у вези прикупљања доказног материјала односно идентификације и груписања свих, са њима повезаних, релевантних података. Експлицитно су наведени предмет и циљ истраживања. Истакнуто је да је коришћењем биометријске карактеристике лица у сврху препознавања особе могуће повезати велику количину разнородних података о особи које се могу пронаћи у дистрибуираним базама података информационих система службе (нпр. фотографије на којима се налазе, видео записи, документа у којима су слике лица одређене особе пронађене, интернет странице и сл.). Такође је наглашено да би аутоматизација овог процеса, у условима када истражитељи манипулишу огромном количином података (између осталог добијених у процесу дигиталне форензике, видео надзора али и прикупљањем из отворених извора - OSINT), значајно утицала на повећање њихове ефикасности. Посебно је указано да ће се, у условима коришћења дистрибуираних база података, као и ограничења у смислу расположивих рачунарских (хардверских и софтверских) ресурса, испитати могућност примене технологије агената и неаксиоматског резоновања. Такође су у овом поглављу наведене радна и помоћне хипотезе истраживања, нацрт саме дисертације као и њен научни допринос.

У другом поглављу је дат преглед стања у области која се односи на предмет истраживања. Ту је апострофирано груписање слика лица добијених у реалном амбијенту (неконтролисани услови - енгл. *Unconstrained face recognition*) (мала резолуција, лош квалитет, различите позе, могућа оклузија...), затим утицај препроцесирања слика лица на њихову детекцију и класификацију/подударање, као и примену графова у поступку разрешавања идентитета.

Треће поглавље описује примењене методе и материјале везане за спроведено истраживање. У овом поглављу је дат преглед база података лица (и њихових карактеристика) које су коришћене у спроведеним експериментима, а у вези са класификацијом и груписањем (кластеризацијом) слика лица добијеним у неконтролисаним условима. Посебно је описан HOG (енгл. *Histogram of Oriented Gradients*) метод детекције лица, као метод који се користи за екстракцију обележја али и као специфична метода класификације. У овом поглављу је предложен нови приступ за кластеровање слика лица (добијених у неконтролисаном окружењу) у дистрибуираном систему, који се заснива на адаптацији NAL-а (енгл. *Non-Axiomatic Logic*). Последњи део поглавља објашњава приступе евалуацији класификације и груписања.

Четврто поглавље описује могућности коришћења софтвера отвореног кода за прикупљање и припрему података за презентацију груписаних доказа и разрешавање идентитета. Од свих извора дигиталних доказа које прикупља криминалистичка полиција, дигитални форензички извештаји се издвајају по количини и разноврсности типова датотека и података. Ово поглавље садржи преглед и тестирање команди које могу да изврше конверзије различитих формата датотека (првенствено оних који се најчешће проналазе у дигиталним форензичким извештајима) у неколико које интернет претраживачи могу да прикажу, као и њихов унос и складиштење. Такође, тестирана је могућност управљања алатима отвореног кода из програмског језика Python коришћењем подпроцеса.

Пето поглавље садржи истраживање о утицају препроцесирања слика лица коришћењем аутоматизације метода (визуелног) побољшања (нормализација, изоштравање и др.) на детекцију и идентификацију лица коришћењем HOG дескриптора и косинусне удаљености. Прво су описане методе визуелног побољшања слика лица, а затим су објашњене експерименталне поставке (*setup*). На крају је дата дискусија добијених резултата.

Шесто поглавље представља проблем груписања слика лица добијених у неконтролисаним условима. Указано је на проблеме са груписањем лица у паровима заснованом на графовима. Након тога, описан је експериментални оквир за тестирање предложене методе груписања прилагођавањем концепта 'истинитосне вредности (енгл. *the truth-value concept*)' неаксиоматског закључивања. Термин 'неаксиоматски' подразумева да не постоји 'аксиом' међу знањем из домена, већ се истинитост исказа увек може ревидирати новим доказима. Тако се сва знања из домена могу научити, а не уградити у систем. Примена неаксиоматског закључивања је анализирана у смислу повећања ефективности како у процесу препознавања тако и у сврху ефикасног груписања података када није унапред познат број класификационих група, а на располагању су ограничени хардверски ресурси. Указано је да се успешном детекцијом, а затим идентификацијом особа или њиховим груписањем, као и датотека и других информација које се могу повезати са њима, омогућава презентација свих расположивих података криминалистичким инспекторима у циљу разрешавања идентитета, односно издвајања потенцијалног доказног материјала који се односи на конкретног осумњиченог и елиминисање инцидентних проналазака. На крају, у поглављу су представљени резултати примене предложеног решења као и резултати поређења са другим, тренутно најпопуларнијим, методама груписања.

У седмом поглављу је описано предложено решење мулти-агентског система који би био подршка (напредним) полицијским активностима (енгл. *Multi-Agent System for Advance Policing - mASAP*). Представљени модел би се користио за подршку одлучивању при разрешавању идентитета, и заснован је на груписању прикупљених дигиталних доказа коришћењем метода груписања (слика) лица. Поглавље садржи преглед интелигентних агената као и приказ примењеног мулти-агентског окружења *mASAP*-а. Суштинска улога *mASAP*-а је представљена у форми практичне употребе подпроцеса и команди отвореног кода, затим комуникације агента у циљу проширења предложене методе груписања у дистрибуираном окружењу и коначно у примени агената за груписање, презентацију и сортирање података.

Кончано у последњем поглављу је дат закључак где су појединачно наведени остварени научни доприноси и где су наведени публиковани радови који су резултат спроведеног истраживања.

3. ОЦЕНА ДИСЕРТАЦИЈЕ

3.1. Савременост и оригиналност

Рад криминалистичке полиције у савременом друштву одликује пролиферација података и информација које треба обрађивати, већи захтеви за ограничењима у раду са личним подацима, појачани надзор пре свега јавности, већа очекивања у ефикасности откривања извршилаца кривичних дела, али и даље присутан проблем у вези недостатка ресурса, како људских тако и материјалних. Један од сложенијих задатака јесте разрешавање идентитета, чијом променом се настоје прикрити криминалне активности, односно сам извршилац који је у бекству. Да би се разрешио идентитет, потребно је груписати и презентовати све расположиве доказе везане за одређене особе.

У дисертацији је предложен нови приступ груписању података поређењем парова вектора одлика лица екстрахованих из слика насталих у неконтролисаним условима, а заснован на резоновању применом неаксиоматске логике и графова. Групе (кластери) слика лица представљају централне тачке око којих се групишу подаци из различитих полицијских извештаја и база података. Такође је предложен модел система у коме ће значајну улогу имати софтверски агенти, пре свега у повезивању тачака дистрибуираног окружења које у пракси формирају полицијски информациони системи.

Нови приступ груписању (кластеровању) је експериментално испитан на шест различитих база података лица карактеристичних по томе што су креиране на начин којим се симулирају неконтролисани услови. Добијени резултати предложеног решења су упоређени са тренутно најпопуларнијим и најбољим доступним методама. Резултати тестирања су показали да предложено решење даје боље резултате у поређењу са другим приступима.

Оно што даје посебну предност у односу на остале методе јесте могућност коришћења механизама из неаксиоматске логике попут ревизије и дедукције, помоћу којих се могу стицати нова знања на основу информација из различитих нодова (чворова) система или у локалној бази знања.

По нашем најбољем сазнању у дисертацији је приказан јединствен и иновативан приступ анализираној проблематици у условима дистрибуираног окружења (извора података). Додатну вредност овом истраживању даје и чињеница да готово и да нема публикованих резултата истраживања у овој области који су везани за могућност примене наведених метода у раду полиције у амбијенту ограничених рачунарских (хардверских и софтверских) ресурса.

3.2. Осврт на референтну и коришћену литературу

Литература коришћена у дисертацији, осим оних референци које су кључне за развој појмова и теоријских поставки из тематске области дисертације, претежно садржи најновије научне радове релевантне за анализирану проблематику (објављене након 2015. године). Поред научних радова, литература садржи и референтне књиге. Велики број радова новијег датума указује на актуелност обрађене тематике, док су у дисертацији јасно истакнуте предности и иновативност предложених метода груписања у односу на моделе у постојећој литератури. На основу свеобухватности референци наведених на крају дисертације, може се закључити да је кандидат детаљно и темељно истражио научну област којој дисертација припада, као и да прати савремена дешавања у датој области.

3.3. Опис и адекватност примењених научних метода

Истраживање је било засновано на примени основних и најважнијих савремених теоријских резултата, односно на прикупљању и проучавању доступне научне литературе, њеној анализи и систематизацији ради прецизног одређивања проблема и расположивих решења. На основу теоријског разматрања као и предлога унапређења/оптимизације постојећих решења изграђен је функционални модел система над којим су извршена експериментална испитивања и где су добијени резултати статистички анализирани.

Научне методе примењене у истраживањима у оквиру ове докторске дисертације обухватају: историјски метод; метод компаративне анализе; метод анализе и синтезе; метод индукције и дедукције; метод генерализације и специјализације; метод моделовања; статистички метод; методе класификације; метод посматрања и мерења; експериментални метод. Посебно су коришћене: методе дигиталне обраде слика; методе примењиве у области рачунарског вида и методе примене алгоритама вештачке интелигенције. То је конкретизовано кроз:

- Анализу доступне, актуелне и релеватне литературе из области претпроцесирања слике лица, груписања (кластеризације) лица, неаксиоматске логике, софтверских агената и граф база, ради утврђивања тренутног стања у области од интереса, идентификовање критичних проблема у области и могућности за унапређење поступака груписања лица.
- Анализа постојећих решења предложених у литератури и селекција најбољих решења која су онда коришћена за компаративну анализу са решењима предложеним у дисертацији.
- Предлог новог метода заснованог на коришћењу механизама из неаксиоматског резоновања попут ревизије и дедукције, помоћу којих се могу стицати нова знања на основу информација из различитих нодова система или у локалној бази знања. Систем резоновања примењен у овој тези заснива се на концептима доказа и истинитосне вредности преузетим из неаксиоматског резоновања који је додатно унапређен за примену на различитим формама података (у конкретном случају је у питању слика лица), а која је погоднија за примену у реалним условима од аксиоматске логике (погодна у идеализованим ситуацијама).
- Развој симулационог окружења за тестирање перформанси постојећих решења као и решења предложеног у дисертацији за:
  - поређења лица - коришћен је Python програмски језик, при чему су перформансе одређиване у виду броја детектованих лица у анализираним базама података пре и након примене предложених метода побољшања слика;
  - груписање лица - предложена је примена неаксиоматског закључивања у условима дистрибуираног окружења;
- Компаративну анализу предложених и постојећих решења - на основу мерења добијених у реализованом симулационом окружењу и анализе добијених резултата, извршена је компаративна анализа којом је утврђен квалитет предложених решења;
- Хардверско - софтверску имплементацију мулти-агентског система у напредним полицијским пословима у Python програмском језику за обраду велике количине података - агентски системи за прикупљање и припрему података, као и за генерисање истинитосних вредности. Посебан акценат је стављен на анализу комуникације између агената два различита *mASAP*-а.

На основу наведених примењених научних метода, утврђено је да су употребљене методе адекватне за истраживања спроведена у оквиру ове дисертације.

3.4. Примењивост остварених резултата

У оквиру компаративне анализе резултата истраживања са резултатима других метода груписања (кластеризације) слика лица, извршена је практична провера резултата у реалном хардверско - софтверском окружењу, користећи шест различитих база података лица. Ово је уједно и потврда успешне примењивости остварених резултата у полицијској пракси. Резултати истраживања ће очекивано омогућити и постизање општег циља, а то је њихова примена у ширем опсегу делатности где је потребно груписање података везаних за одређене особе, а на располагању су фотографије које садрже њихова лица, попут алата намењених дигиталној форензици, аналитичких алата везаних за видео надзор и др.

Један од важнијих циљева овог истраживања је софтверска реализација и имплементација предложених метода, односно њихова практична примена унутар информационог система Министарства унутрашњих послова Републике Србије, чиме ће резултати овог истраживања имати потпуну практичну имплементацију

3.5. Оцена достигнутих способности кандидата за самостални научни рад

Кандидат је у оквиру рада на дисертацији реализовао обимно истраживање литературе из области претпроцесирања слике лица, њиховог груписања, примене графова и неаксиоматске логике. Извршио је свеобухватан преглед и анализу постојећег стања у области истраживања, детектовао критичне проблеме и урадио преглед и анализу постојећих решења.

Такође, кандидат је урадио компаративну анализу предложених и постојећих решења, уз квалитативан приказ и анализу резултата, као и адекватан избор сценарија који су коришћени у компаративној анализи.

Током рада на тези, кандидат је исказао завидну самосталност као и критичко размишљање. Објављивањем резултата истраживања у оквиру докторске дисертације кроз научне радове, међу којима су и радови у часописима са SCI листе, кандидат је потврдио способност за представљање својих научних доприноса у научној јавности.

На основу наведеног, може се констатовати да је кандидат достигао адекватан степен способности за самостални научни рад.

4. ОСТВАРЕНИ НАУЧНИ ДОПРИНОС

4.1. Приказ остварених научних доприноса

Научни доприноси докторске дисертације се могу укратко сумирати на следећи начин:

- У циљу побољшања прецизности предложен је нови приступ груписања (кластеровања) у паровима модификацијом NAL приступа и његовим проширењем/адаптацијом на нови тип података (слика лица).
- Предложена је адаптација концепта истинитости којом је размена података између тачака дистрибуираног система уређена на начин да помаже процесу груписања и ствара ново знање користећи механизме неаксиоматског закључивања као што су ревизија и дедукција.
- Предложени метод појашњава процес груписања неопходан за транспарентност система који укључује вештачку интелигенцију.
- Потврда позитивног утицаја техника побољшања слике (претпроцесирања) на процес препознавања лица, посебно на детекцију и подударање помоћу HOG-a.
- Предложени метод примене Bourne-Again SHell (Bash) команди и апликација отвореног кода контролисаних из програмског језика Python обезбеђује побољшане могућности софтверског агента у прикупљању и припреми података за чување у бази података, претрагу и презентацију коришћењем веб технологија.
- Примена агената на начин предложен у тези омогућава коришћење свих информација дистрибуираног система који помажу у груписању лица, док је крајњим корисницима омогућено да виде само податке које су прикупили. На тај начин је додатно оснажен концепт заштите приватности података о личности.
- Систематски преглед литературе који истиче најзначајније радове који се односе на истраживану проблематику.

4.2. Критичка анализа резултата истраживања

Кандидат је успешно идентификовао могућности за унапређење области препознавања лица и потврдио исправност полазних хипотеза формулисаних на почетку свог истраживања везаног за израду докторске дисертације. На основу извршених истраживања литературе, тренутног стања области и постојећих решења груписања слика лица, кандидат је успео да предложи сопствено решење које унапређује област препознавања лица.

По нашем најбољем сазнању у дисертацији је приказан јединствен и иновативан приступ анализираној проблематици у условима дистрибуираног окружења и непознатог броја група (класа). Предложено решење остварује веома добре перформансе уз прихватљиве хардверске и софтверске захтеве, при чему су остварене перформансе упоредиве или боље у односу на перформансе постојећих решења. Реализована унапређења заокружују комплетност предложеног решења. При томе, компаративном анализом је утврђено да перформансе у погледу груписања лица превазилазе перформансе постојећих решења у свим тестираним сценаријима, што потврђује целокупан квалитет предложеног решења. Квалитет постигнутих резултата истраживања је верификован и публиковањем научних радова, међу којима се издвајају два рада у часописима са SCI листе.

4.3. Верификација научних доприноса

Радове наведене у овој секцији је кандидат објавио за време докторских студија и истраживања у оквиру рада на дисертацији и у непосредној су вези са докторском дисертацијом. У наставку је дат списак тих радова:

Категорија М22:

1. **Vuković, I.**, Kuk, K., Čisar, P., Banđur, M., Banđur, Đ., Milić, N., & Popović, B.(**2021**). Multi-agent system observer: Intelligent support for engaged e-learning. *Electronics*, 10(12), 1370. DOI: 10.3390/electronics10121370 (**IF=2.69**) (ISSN 2079-9292)

Категорија М23:

1. **Vukovic, I.**, Cisar, P., Kuk, K., Bandjur, M., & Popovic, B. (**2021**). Influence of image enhancement techniques on effectiveness of unconstrained face detection and identification. *Elektronika ir Elektrotechnika*, 27(5), 49–58. DOI: https://doi.org/10.5755/j02.eie.29081 (**IF=1.059**) (ISSN 1392-1215)

Категорија М33:

1. **Vuković, I.** (**2019**). Centralized digital forensic analytical system: Rationalization using open- source software of general purposes. *In Thematic conference proceedings of international significance, International scientific conference 'Archibald Reiss Days'*, 6-7 November 2019, University of Criminal Investigation and Police Studies, Belgrade, **2019**, 9(2): 361-375.

2. **Vuković, I.**, Čisar, P., Kuk, K., & Popović, B. (**2021**). Challenges of contemporary predictive policing. *In Thematic conference proceedings of international significance, International scientific conference 'Archibald Reiss Days'*, 9-10 November 2021, University of Criminal Investigation and Police Studies, Belgrade, **2021**, 11(2): 513-526.

5. ЗАКЉУЧАК И ПРЕДЛОГ

Комисија за оцену докторске дисертације под називом ‘Разрешавање идентитета и груписање дигиталних доказа о осумњиченима применом технологија препознавања лица и система софтверских интелигентних агената заснованог на неаксиоматском резоновању’ кандидата Игора Вуковића сматра да је наведена докторска дисертација адекватно методолошки постављена, правилно спроведена и заснована на актуелним сазнањима. Дисертација је урађена у складу са образложењем наведеним у пријави теме и садржи све елементе који се захтевају Правилником о докторским студијама КПУ.

Комисија такође сматра да докторска теза мр Игора Вуковића представља оригинални научни допринос у оквиру научних области Рачунарских наука и Електротехничког и рачунарског инжењерства.

Докторанд мр Игор Вуковић је темељно анализирао научна достигнућа у области препознавања особа на основу слике лица, а затим предложио унапређења у тој области. Предложио је методе које се могу користити у условима ограничених рачунарских (хардверских и софтверских) ресурса. Експериментално је показао предности предложених решења у:

- области детекције и препознавања лица применом одговарајућих техника претпроцесирања слике лица;
- области груписања лица унапређењем концепта неаксиоматског закључивања на податке добијене из слике лица;
- области примене софтверских агената за прикупљање, обраду и презентацију података у дистрибуираном окружењу, заснованом на веб технологијама.

Сва решења су иплементирана алатима отвореног кода чиме је омогућена њихова практична примена на конкретним пословима криминалистичке полиције. У том смислу је предложено и оригинално решење мулти-агентског система намењеног за подршку (напредним) полицијским активностима (енгл. *Multi-Agent System for Advance Policing - mASAP*).

Током рада на тези, докторанд мр Игор Вуковић је исказао завидну самосталност као и критичко размишљање. Резултате проистекле из истраживања спроведеног у оквиру докторске дисертације, кандидат је објавио у врхунским међународним часописима и презентовао на конференцијама међународног значаја.

На основу свега изложеног, имајући у виду научне резултате истраживања, практичну примену истраживања, методолошки оквир, научни и друштвени допринос, Комисија даје позитивну оцену докторске дисертације и Већу департмана информатике и рачунарства, Криминалистичко-полицијског универзитета у Београду

ПРЕДЛАЖЕ

да се докторска дисертација под називом '**Разрешавање идентитета и груписање дигиталних доказа о осумњиченима применом технологија препознавања лица и система софтверских интелигентних агената заснованог на неаксиоматском резоновању**' кандидата Игора Вуковића **прихвати, изложи на увид јавности и упути на коначно усвајање Већу научних области природно-математичких и техничко-технолошких студија Криминалистичко-полицијског универзитета.**

У Београду, 04.7.2022. године

ЧЛАНОВИ КОМИСИЈЕ

Кук Кристијан

др Кристијан Кук, председник, ванредни професор,
Криминалистичко-полицијски универзитет

Čisar Petar

др Петар Чисар, члан, редовни професор,
Криминалистичко-полицијски универзитет

Милош Банђур

др Милош Банђур, члан, ванредни професор,
Универзитет Приштина са седиштем у Косовској
Митровици - Факултет техничких наука

Република Србија
КРИМИНАЛИСТИЧКО - ПОЛИЦИЈСКИ
УНИВЕРЗИТЕТ
24 Број: 81/8-3-2022
Датум: 08.07.2022

На основу члана 57. став 1. тачка 3. Статута Криминалистичко-полицијског универзитета (21 број 72/1-2-2022 од 10.02.2022. године) а у вези члана 29. став 1. Правилника о докторским студијама (22 број 79/4-2-2019 од 15.03.2019. године), Веће Департмана информатике и рачунарства на седници одржаној дана 8. 7. 2022. године, а на основу Извештаја комисије за оцену докторске дисертације број 43/1-4 од 4. 7. 2022. године, донело је следећи

ПРЕДЛОГ ОДЛУКЕ

I Прихвата се позитивна оцена докторске дисертације, под називом: **„Разрешавање индентитета и груписање дигиталних доказа о осумњиченима применом технологија препознавања лица и система софтверских интелигентних агената заснованог на неаксиоматском резоновању“** студента Игора Вуковића.

II Докторске дисертација под називом: **„Разрешавање индентитета и груписање дигиталних доказа о осумњиченима применом технологија препознавања лица и система софтверских интелигентних агената заснованог на неаксиоматском резоновању“** студента Игора Вуковића и Извештај комисије за оцену докторске дисертације (број 43/1-4 од 4. 7. 2022. године), ставља се на увид јавности у року од 30 дана на интернет страници Установе и у библиотеци Криминалистичко-полицијског универзитета.

Образложење

Студент докторских академских студија Информатика, Игор Вуковић, поднео је пријаву теме докторске дисертације, број 124/7 дана 16. 9. 2021. године, за одобравање теме докторске дисертације из научне области Рачунарске науке, Електротхничко и рачунарско инжењерство.

Веће научних области природно-математичких и техничко-технолошких студија (Одлуком 28 број 85/5-2-2021 од 17. 11. 2021. године) прихватило је тему докторске дисертације под наведеним називом и студенту именовало ментора проф. др Бранкицу Поповић.

У складу са чланом 25. Правилника о докторским студијама студент Игор Вуковић дана 15. 6. 2022. године, завршену докторску дисертацију поднео је за оцену надлежном Већу департмана уз прилагање потребних доказа.

Сходно члану 25. став 2. ментор је дала писану сагласност дана 22. 6. 2022. године (број 43/1-2), којом је констатовала да је докторска дисертација студента Игора Вуковића подобна за оцену и да студент има објављен најмање један рад који је повезан са садржајем докторске дисертације у коме је он први аутор објављен или прихваћен за објављивање у часопису са импакт фактором са SCI листе.

У слкаду са чланом 27. Правилника о докторским студијама, Веће научних области (Одлуком 28 број 85/4-2-2022 од 28. 6. 2022. године) именовало је Комисију за оцену докторске дисертације у саставу: проф др Кристијан Кук, председник, проф. др Петар Чисар, члан, проф. др Милош Банђур, члан.

У складу са чланом 28. Правилника о докторским студијама, именована комисија је дана 4. 7. 2022. године поднела Извештај о оцени докторске дисертације (број 43/1-4) и предложила да надлежно Веће департмана прихвати позитивну оцену докторске дисертације наведеног студента.

Чланом 29. Правилника о докторским студијама, прописано је да у случају да Веће Департмана усвоји извештај о оцени докторске дисертације Универзитет је дужан да предлог одлуке о прихватању оцене, уз извештај о оцени и докторску дисертацију, учини доступном јавности, и то на званичној интернет страници установе, као и у штампаном облику у библиотеци Криминалистичко-полицијског универзитета, најмање 30 дана пре разматрања на надлежном Већу научних области.

Сходно наведеном одлучено је као у изреци.

ДОСТАВИТИ:
- *Већу научних области*
- *Одсеку за студенска питања*

Председавајући Већа

проф. др Бранкица Поповић

UNIVERSITY OF CRIMINAL INVESTIGATION AND POLICE STUDIES

Department of Informatics and Computing

Igor M. Vuković

Identity resolving and grouping of digital evidence of suspects using face recognition technologies and software intelligent agents system based on non-axiomatic reasoning

Doctoral Dissertation

Belgrade, 2022.

КРИМИНАЛИСТИЧКО-ПОЛИЦИЈСКИ УНИВЕРЗИТЕТ

Департман информатике и рачунарства

Игор М. Вуковић

Разрешавање идентитета и груписање дигиталних доказа о осумњиченима применом технологија препознавања лица и система софтверских интелигентих агената заснованог на неаксиоматском резоновању

Докторска дисертација

Београд, 2022.

Dissertation title: "Identity resolving and grouping of digital evidence of suspects using face recognition technologies and software intelligent agents system based on non-axiomatic reasoning"

Abstract:

The work of criminal police in modern society is characterized by the proliferation of data and information to be processed, greater demands for restrictions on personal data, increased public monitoring, and higher expectations in the efficiency of detecting perpetrators, but still lack resources, both human and material. One of the more complex tasks is to resolve the identity, the change of which seeks to cover up criminal activities, i.e., the perpetrator himself, who is on the run.

In order to resolve the identity, it is necessary to group and present all available evidence related to specific persons. The thesis proposes a clustering approach by comparing pairs of face feature vectors extracted from images created in unconstrained conditions and based on reasoning using non-axiomatic logic and graphs. Face clusters will be the central points around which data from various police reports will be grouped. A system model has also been proposed in which software agents will play a significant role, primarily in connecting the distribution environment points formed in practice by police information systems.

The clustering approach was experimentally tested with six different face image databases characterized by the fact that they were created in a way that simulates unconstrained conditions. The obtained results of the proposed solution are compared with other state-of-the-art methods. The results showed that the approach gives similar but mostly better results than the others. What gives a notable advantage over other methods is the possibility of using mechanisms from non-axiomatic logic such as revision and deduction, which can be used to acquire new knowledge based on information from different system nodes, or in the local knowledge base, respectively.

Keywords: face clustering, non-axiomatic logic, digital forensics, open-source tools, software agent, graph database.

Scientific field: Computer Science, Electrical and Computer Engineering

Scientific subfield: Information Technology, Software Engineering

Наслов докторске дисертације: ”Разрешавање идентитета и груписање дигиталних доказа о осумњиченима применом технологија препознавања лица и система софтверских интелигентих агената заснованог на неаксиоматском резоновању”

Сажетак:

Рад криминалистичке полиције у савременом друштву одликује пролиферација података и информација које треба обрађивати, већи захтеви за ограничењима у раду са личним подацима, појачани надзор пре свега јавности, већа очекивања у ефикасности откривања извршилаца кривичних дела, али и даље недостатак ресурса, како људских тако и материјалних. Један од сложенијих задатака јесте разрешавање идентитета чијом променом се настоје прикрити криминалне активности, односно сам извршилац који је у бекству.

Да би се разрешио идентитет, потребно је груписати и презентовати све расположиве доказе везане за одређене особе. У дисертацији је предложен нови приступ кластеровању поређењем парова вектора одлика лица екстрахованих из слика насталих у неконтролисаним условима, а заснован на резоновању применом неаксиоматске логике и графова. Кластери слика лица представљају централне тачке око којих се групишу подаци из различитих полицијских извештаја. Такође је предложен модел система у коме ће значајну улогу имати софтверски агенти, пре свега у повезивању тачака дистрибуираног окружења које у пракси формирају полицијски информациони системи.

Нови приступ кластеровању је експериментално испитан са шест различитих база података лица карактеристичних по томе што су креиране на начин којим се симулирају неконтролисани услови. Добијени резултати предложеног решења су упоређени са осталим врхунским методама. Резултати су показали да приступ даје приближне, али углавном боље резултате од осталих. Оно што даје посебну предност у односу на остале методе јесте могућност коришћења механизама из неаксиоматске логике попут ревизије и дедукције, помоћу којих се могу стицати нова знања на основу информација из различитих нодова система, или у локалној бази знања, респективно.

Кључне речи: кластеровање лица, неаксиоматска логика, дигитална форензика, алати отвореног кода, софтверски агенти, граф базе

Научна област: Рачунарске науке, Електротехничко и рачунарско инжењерство

Научна подобласт: Информационе технологије, софтверско инжењерство

Dedication

To my late colleagues Vladimir Gujaničić, Dragić Bojić, and Slobodan Stojiljković who left us too soon.

You'll Always Be Remembered.

Acknowledgments

This extensive endeavor would not have been possible without my parents and my wife.

I would like to express my deepest gratitude to my supervisor, professor Brankica Popović, for all the support and advice.

I would also like to thank professor Kristijan Kuk and professor Petar Čisar. Their experience provided me with valuable feedback, in that way improving my dissertation.

Special thanks to my colleagues from the European Union's Horizon 2020 SPIRIT project consortium (Grant No. 786993), who provided insight and expertise that greatly assisted the research.

Contents

Acronyms **VIII**

List of Tables **XI**

List of Figures **XII**

Listings **XV**

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Subject and goals of research 5

1.3 Research hypotheses 6

1.4 Contribution 7

1.5 Thesis Outline 8

2 Related work **9**

3 Materials and methods **14**

3.1 Face image databases 14

3.1.1 Imdb-Wiki database 15

3.1.2 Extended Yale Face Database B 15

3.1.3 Labeled Faces in the Wild 16

3.1.4 YouTube Faces 17

3.1.5 IARPA Janus Benchmark-B Face Dataset 17

3.1.6 SSIM database 18

3.2 Face recognition 19

3.2.1 Histogram of Oriented Gradients 19

3.2.2 Feature extraction and matching 21

3.3 The proposed approach to face clustering 22

3.3.1 The concept of evidence and truth-value 22

3.3.2 Adaptation of truth-value for face clustering purpose 24

3.3.3 Inference based on truth-values 27

3.4 Evaluation 30
3.4.1 Evaluation methods for classification 30
3.4.2 Evaluation methods for clustering 31

4 Gathering digital evidence for presentation and identity resolution using open-source tools 33
4.1 Collecting digital evidence from forensic reports 34
4.1.1 Using open-source tools to collect evidence 35
4.1.2 Collecting information about files 37
4.1.3 Collecting textual content 38
4.1.4 Conversion of various formats containing face images 41
4.1.5 Data import, analysis and reporting 43
4.2 Subprocesses 44
4.3 Results and discussion 46
4.4 Summary 50

5 Influence of image enhancement techniques on effectiveness of unconstrained face detection and matching 52
5.1 Image enhancement 53
5.2 Experimental settings 54
5.3 Results and discussion 58
5.4 Summary 63

6 Truth-value unconstrained face clustering 64
6.1 Unconstrained face clustering 65
6.1.1 Graph-based pairwise face clustering issues 67
6.2 Experimental settings 69
6.2.1 System setup 70
6.2.2 Experiment run 70
6.3 Results and discussion 73
6.4 Summary 80

7 Multi-Agent System for Advance Policing - mASAP 82
7.1 mASAP system architecture 83
7.2 Software agents 85
7.3 Multi-Agent System 87
7.3.1 Smart Python multi-Agent Development Environment - SPADE 87
7.4 The roles of mASAP agents 88
7.4.1 Data preparation 89

7.4.2 Communication and revision 93
7.4.3 Grouping and presentation of data 98
7.5 Conclusion remarks 102

8 Conclusion 104
8.1 Publications from the doctoral dissertation 107

References 108

Acronyms

ACID	Atomicity, Consistency, Isolation, Durability 48
ACL	Agent Communication Language 96, 98
AGI	Artificial General Intelligence 22, 23, 86
AJAX	Asynchronous JavaScript And XML 83
AQL	ArangoDB Query Language 70

Bash	Bourne-Again SHell 7, 36, 43, 45, 49–51, 84, 89, 92, 105, 106, XIV
BCL	Ball Cluster Learning 11
BDI	Belief-Desire-Intention 86

CCTV	Closed-Circuit Television 34
CDA	Clustering-based Domain Adaptation 12
CDP	Consensus-Driven Propagation 10
CLI	Command-Line Interface 38, 44, 48, 84
CNN	Convolutional Neural Network 10, 11, 66
ConPaC	Conditional Pairwise Clustering 10

DA-Net	Density Aware Feature Embedding Network 11
DBSCAN	Density-Based Spatial Clustering of Applications with Noise 11, 66
DDC	Deep Density Clustering 11
DOM	Document Object Model 83

FIPA	Foundation for Intelligent Physical Agentss 87, 96, 98

GAMA	GAma Modeling Language 87
GCN	Graph Convolutional Networks 10, 11, 66
GUI	Graphical User Interface 38, 41, 44, 84

HOG	Histogram of Oriented Gradients 3, 4, 7, 8, 19–21, 28, 52, 53, 63, 67, 68, 70, 72, 105, XIII
HTML	HyperText Markup Language 35, 43, 45, 50, 83
IJB-B	IARPA Janus Benchmark-B Face Dataset 14, 15, 17, 18, 79, 80, XI, XII
IL	Inheritance Logic 23
JADE	Java Agent Development Framework 87
JID	Jabber Identity 97
JPEG	Joint Photographic Expert Group 35
JSON	JavaScript Object Notation 35, 43, 83
k-NN	k-Nearest Neighbors 10, 11
LBC	Linkage-Based Face Clustering 11
LEA	Law Enforcement Agencies 2–4, 34
LFW	Labeled Faces in the Wild 14, 16, 17, 70, 75, 76, 78, 79, XII–XIV
MAS	Multi-Agent System 86, 87
mASAP	Multi-Agent System for Advance Policing 8, 83–85, 88, 89, 91, 93, 95–102, 105, 106, XIV
MIME	Multipurpose Internet Mail Extensions 90–92, 103, 105
NAL	Non-Axiomatic Logic 3, 5–8, 10, 12, 22–24, 26–29, 69, 73, 80, 83, 86, 104, 107, XII
NARS	Non-AxiomaticReasoning System 23, 28
NMI	Normalized Mutual Information 31, 32, 70, 73, 79
OSINT	Open-source intelligence 1, 16, 34, 64, 80, 99, 104, 106
PAHC	Proximity-Aware Hierarchical Clustering 10
PDF	Portable Document Format 35, 38, 39, 42, 45, 46, 92, 93, XII
PNG	Portable Network Graphics 35, 41, 42, 45, 90
PSD	Photoshop Document 42
ResNet	Residual Neural Network 3, 4, 10, 12, 19, 21, 68, 72, 105

RGCN	Residual Graph Convolutional Network 11
SIFT	Scale-Invariant feature transform 19
SPADE	Smart Python multi-Agent Development Environment 84, 87, 88, 96, 98, 103, XV
SVM	Support-Vector Machine 10
TVC	Truth-value clustering 69, 70, 73, 77–80, 93, 95, 98, 99, 103, 105, 106, XIV
URL	Uniform Resource Locator 88
XHR	XMLHttpRequest 83
XML	Extensible Markup Language 35, 83, 88
XMPP	eXtensible Messaging and Presence Protocol 87, 88, 96, 103

List of Tables

3.1 Characteristics of face images in databases. 14

3.2 Examples of revision based on a combination of different values of frequency and confidence. 29

5.1 The number of detected faces by databases and applied image enhancement methods. 59

5.2 Scores of the classification performed using the weight value *w*. 62

6.1 Performance comparison with different face image data sets. 75

6.2 Results of different face clustering algorithms testing on multiple databases. 79

6.3 Results of different face clustering algorithms testing on IJB-B database. 80

7.1 Results of using the **file** command with the –mime option on a set of files, in original and processed form. 90

List of Figures

3.1 Representative examples from the Imdb-Wiki database. 15

3.2 Representative examples from the Extended Yale Face Database B database. 16

3.3 Representative examples from the LFW database. 16

3.4 Representative examples from the YouTube Faces database. 17

3.5 Example from the IJB-B database. 18

3.6 Preview of SSIM database. 18

3.7 Representative examples from the SSIM database. 19

3.8 Illustration of the feature vector extraction process from a face image. 21

3.9 Graph representing a sample of system experience K in a) before and b) after adaptation. 25

3.10 Graph representing a) intensions and extensions of terms T_1 and T_2 and b) "similar feature vector" $\vec{V}_i^{\vartheta}$ to vectors $\vec{V}_1$ and $\vec{V}_2$ replacing intesions and extension after adaptation. 26

3.11 Influence of positive and negative evidence on resulting truth-values a) in NAL, and b) after adaptation. 27

3.12 Rules of inference in NAL: a) deduction, b) abduction and c) induction. 28

3.13 Applied inference method after NAL adaptation. 28

4.1 Collecting file information from the file system. 37

4.2 Collecting metadata with exiftool command. 38

4.3 Conversion of Office and PDF file contents into a text. 39

4.4 Extraction of SQLite tables content into a text file. 40

4.5 Reading text files and printable characters from binary files. 40

4.6 Comparison of the results of the strings command and pdftotext command working on the same file. 41

4.7 Example of conversion vector graphics files into a raster-graphic file format. 42

4.8 Importing collected data into a NoSQL database. 43

4.9 Database search and report creation. 44

4.10 Detected faces report that is the result of the Python script execution. 45

4.11 Output data obtained during Python script execution. 45

4.12 A snapshot of the process showing concurrent command execution. 46

4.13 The five instances of the Terminal program use the same script and at the same time import data into the database. 47
4.14 Change of the number of documents in the database during import. 47
4.15 Change of the log file size during the data import. 47
4.16 An example of two documents from the same database collection with a difference in the structure 48
4.17 MongoDB queries enable different searches for forensic artifacts. 49
4.18 An example of a query that helps during triage. 50

5.1 Resolution distribution in the databases used in the experiment. 53
5.2 Comparison of a) unprocessed and b) enhancement image and representation of HOG descriptors c) before and d) after applied preprocessing. 54
5.3 The procedure of conducting the experiment 55
5.4 The confusion matrix for a binary classification. 56
5.5 Examples of histograms of the distribution of a) true negatives (TN) and b) true positives (TP). 57
5.6 Impact of enhancement on the total number of detected persons. 59
5.7 Similarity values were obtained by matching one face with all others in the SSIM database with a) unprocessed images and b) normalization applied. The similarity value indicating the same person is represented by dark dots, while light dots represent different people's faces. 60
5.8 The relationship between the size of the uncertainty interval and the number of similarities within the interval after applying different image enhancement methods. . . . 61
5.9 The total ratio of the number of correctly classified negatives through all databases and applied the image enhancement method with a high $TNrate$. 61
5.10 The total ratio of the number of correctly classified positives through all databases and applied the image enhancement method with a high $TPrate$. 62
5.11 The ratio of scores for classification by applying weight values w. 63

6.1 Representation of HOG face descriptors and 68-keypoints face landmarks. 67
6.2 The calculation of the distance feature vectors obtained based on the HOG descriptors of the same person shown under a) and b) gives a lower value of cosine distance than the images under c) and d) belonging to different persons. 68
6.3 Result of determining the value of the similarity threshold ϑ_t within the uncertainty interval. 69
6.4 Proposed face clustering framework. 71
6.5 The steps of conducting the experiment. 71
6.6 The experiment results of applying the proposed clustering approach with samples of a) IMDb-Wiki, b) SSIM, c) Yale B and d) LFW database. 76

6.7 Influence of different f_t and ϑ_t values on $B^3 Precision$ and $B^3 Recall$ when $B^3 F\ score >$ 0.95 and applying TVC clustering on IMDb-Wiki database face images. 77

6.8 Influence of different f_t and ϑ_t values on $B^3 Precision$ and $B^3 Recall$ when $B^3 F\ score >$ 0.95 and applying TVC clustering on SSIM database face images. 77

6.9 Influence of different f_t and ϑ_t values on $B^3 Precision$ and $B^3 Recall$ when $B^3 F\ score >$ 0.95 and applying TVC clustering on Yale B database face images. 77

6.10 Influence of different f_t and ϑ_t values on $B^3 Precision$ and $B^3 Recall$ when $B^3 F\ score >$ 0.95 and applying TVC clustering on LFW database face images. 78

6.11 Results of an experiment where TVC is applied on images set YTFaces database containing video frames a) shuffled and b) sorted. 78

7.1 The architecture of the mASAP system. 83

7.2 The relationship between Bash and applications within the operating system. 84

7.3 The relationship between file extensions and different file type information. 89

7.4 Collecting truth-values from other nodes of the distributed system. 94

7.5 Initiating communication between agents of two different mASAP. 96

7.6 Establishing communication between agents of two different mASAP and sending requests. 97

7.7 An agent receives responses from other mASAP node with truth-value and ends direct communication. 97

7.8 Representation of the set of data collected and processed by the mASAP and their relation. 99

7.9 Grouping of data based on keyword search with limitation of analysis space only on data from a) reports in which query hit was found and b) with extension to all members of the face cluster. 100

7.10 Grouping of data based on keyword search with an extension of analysis space to all data for which there are mutual relations in the mASAP. 100

7.11 Grouping of data based on a query containing an image with a) limited space for analysis to the cluster to which one of the detected faces belongs and b) extension to all data related to other cluster members. 101

7.12 Model of a) data and information presentation and b) sorting strategy. 102

Listings

4.1 Example of subprocess application in Python programming language. 44
4.2 Concurrent creation of subprocess from Python programing language. 46
7.1 An example of code for implementation of the agent learning process 92
7.2 Calculating the resulting truth-value 95
7.3 An example of code implementing sending a message by an agent via the SPADE platform 98

Chapter 1

Introduction

In the process of identifying suspects and gathering evidence, criminal police investigators follow the principles of criminology, which are mandatory guidelines for more efficient achieving a positive outcome in everyday practice (Aleksić & Škulić, 2010). However, the challenges in the fight against modern forms of crime lead to opposing certain principles. The amount of data and information collected from various sources during the criminal investigation through search procedures, forensic expertise, electronic and physical surveillance, and access to publicly available sources by Open-source intelligence (OSINT) tools is enormous. Therefore, they need to make significant compromises in respecting the principles of efficiency, then the principles to be systematic (which seek to ensure the most rational use of staff, time, and resources), and the principles of thoroughness and perseverance, which can affect the results of the investigation. The internet has caused significant changes in enabling organized crime groups to reach a large number of victims and commit a wide range of crimes on a large scale (Casanovas, Morris, González-Conejero, Teodoro, & Adderley, 2018). The challenge for police organizations is how to timely and efficiently apply the astounding amount of data to solve problems they face in everyday work and trying not to create new ones.

1.1 Background and Motivation

It is imperative that all collected information investigators use to determine the true identity of the perpetrators of criminal acts and then collect evidence related to them. In such circumstances and with such a volume of data and information, traying to process them as quickly as possible, but as thoroughly as possible, it is most likely that investigators are going to include as evidence a large number of those related to persons not associated with the investigation. The pronounced trend is that in addition to personal data collected by conventional police methods, more and more attention is paid to the privacy and protection of data collected from the internet.[1] Moreover, large economies that have

[1] The principle of legality in Serbia, for example, used to be primarily related to compliance with the Criminal Procedure Code, now the Law on Personal Data Protection has an increasing influence on the work of the criminal police

long resisted regulating personal data over the internet, which they used for business, are now writing appropriate laws. First, the European Union presented the General Data Protection Regulation a few years ago, and then California Consumer Privacy Act is a similar example in the United States.

A particularly interesting set of data for criminal investigations are photographs and videos that enable the identification of perpetrators of crimes and represent an essential set of digital evidence. It is a set of data that is the most demanding for analysis since it is conditioned by direct insight into each image or video and recollection of previously observed for comparison, aiming to determine who the person is and whether it is necessary to include a specific image or video into evidence. Assistance in this work can be offered to investigators by face identification tools where matching faces is performed with all feature templates stored in the database to find the subject identity among several possibilities that match the most (Jayaraman, Gupta, Gupta, Arora, & Tiwari, 2020). However, a different approach is needed when the identity has yet to be determined (there is no appropriate template in the database), especially when there is a need for identity resolution. Identity resolution requires grouping, presenting, and comparing all data, images, and video frames related to the same person.

Furthermore, effective identification and identity resolution should address and eliminate incidental findings that may be found in evidence that violates the privacy of people unrelated to the crime or perpetrator. The term incidental findings come from research in medicine and genetics and include both false positives and marginal findings (Schmücker, 2016). The meaning of false positives is applicable in the same way in different research fields. At the same time, marginal findings represent knowledge that has no clinical significance. Still, the meaning can be transferred to finding and handling private data of persons not relevant to the investigation, whose rights may be violated or negatively affect the everyday life of citizens.

The pressure on Law Enforcement Agencies (LEA) to reduce crime rates is constantly growing. Social networks give particular impetus by providing unlimited resources for the exchange and distribution of information generated by crowdsourcing, which is arguments most often for criticizing the work of the police and demonstrating omissions. Also, the supervision of the police due to their authority, one could interpret as "incredible discretion" (Selbst, 2017), is especially interesting for the non-governmental sector, various organizations, and the media. Under such pressure, the LEA is turning to modern technologies to improve police resource management, increase efficiency in crime prevention and reduce unnecessary contact with personal data. To present all data related to the subject of investigation is necessary to group evidence on a person.

One way to achieve that is by facial biometrics and the face clustering process. Face clustering is a process in which a set of images of different faces creates a set of clusters, C by grouping faces belonging to the same person where one of the main requirements is $C = \{\forall\, C_i, C_j \mid C_i \cap C_j = \emptyset, i \neq j\}$. The process of face clustering finds and groups faces of a previously unknown number of persons within the data collected during the investigation, which contains many images and videos. The clustering presented in this thesis is based on pairwise similarities (distance calculation) between the representation of a face, i.e., face descriptors. Clusters consist of descriptors with minimal mutual distances. The problem with accurate clustering, which is further transferred to identity resolution and exclusion of incidental findings, is the material (images and videos) created in unconstrained conditions, which determines evidence collected by open-source intelligence, surveillance, digital forensics, and other police methods. Aggravating factors that affect face clustering can be summarized as those that cause variation in the appearance of persons on images (age, occlusion, illumination); those that affect the similarity between different faces (kinship); and those that affect the technical quality of facial images (poor imaging equipment) (Taskiran, Kahraman, & Erdem, 2020). These factors produce the uncertainty interval within which the same similarity values (for example, the cosine distance between the descriptor vectors) can indicate both the same and the different person (Vukovic, Cisar, Kuk, Bandjur, & Popovic, 2021). Even deep networks that give perfect results with available face sets can have a problem with unconstrained face images because it is difficult to find a representative data set needed for training that matches the complexity of the real world (M. Wang & Deng, 2020).

Solving identity resolution and incidental findings requires maximum precision from the perspective of the confusion matrix, $(TP/(TP+FP)) \rightarrow 1, FP \rightarrow 0$, but also finding all relevant data (recall) $TP/(TP+FN)) \rightarrow 1, FN \rightarrow 0$. The confusion matrix is a table used to evaluate the performance of the binary classification method, and TP, FP, and FN denote the number of true positive, false positive, and false negative values, respectively.[2] This thesis proposes a new approach to the data-driven method of unsupervised unconstrained face clustering that uses pairwise distances between face descriptors in the clustering process and graph theory, but primarily important is reasoning based on Non-Axiomatic Logic (NAL) principles to increase the accuracy of the process itself. In addition, adaptation NAL has another important function that allows the exchange of data points of the distributed system.

Given that LEA is funded by communities, cities, or countries in which they work, another critical requirement for a system attended to assist in criminal investigations is the choice of affordable technology that does not require excessive hardware resources and that uses affordable software (ideally free of charge). Therefore, the Dlib version of Histogram of Oriented Gradients (HOG) face detection and Residual Neural Network (ResNet) for extractions feature vectors, open-source face recognition technology, were used to create the system. It is a C++ library, and its HOG face detector implementation is an off-the-shelf solution that achieves reasonable precision and processing speed (Johnston & de Chazal, 2018). The process of creating a HOG descriptor consists of (1) dividing the image into small connected regions, called cells; (2) edge orientation histograms are calculated for each cell; (3) normalizing the histogram values concerning all block cells; (4) integration of normalized blocks

[2] True positives represent the number of correct predictions of a positive outcome, false positives are the incorrect prediction of positive outcomes, and false negatives are the incorrect prediction of negative outcomes.

builds a HOG descriptor (Jayaraman et al., 2020).

The choice of use ResNet solution is primarily due to the relatively small, 128-dimensional feature vector that represents the face descriptor. In this thesis, the feature vector is recorded in the ArangoDB graph database collection, primarily for the clustering process. Graph databases enable cluster indexing, which allows efficient search of available data. The ArangoDB is another open-source solution. Solutions from the world of open-source software, characterized by the fact that they are free, i.e., free to use, will also be used for preprocessing photos of faces and collecting evidence that will be grouped with the help of face clustering.

Furthermore, to provide assistance in the work of criminal investigators, on whom the final decision remains, the methodology of application and particular implementation of software agent technology will be researched, which would enable permanent work on data preparation, processing, and analysis, but also in distributed systems as imposed in the functioning of large organizations such as the police (state and interstate level). The main feature of LEA is segmentation by different units and levels, services, directorates, and departments, where they often deal with the same type of crime, and thus the same perpetrators as individuals or organized criminal groups. There are also everyday situations where the same criminal groups or individuals are found again within different lines of police work (financial investigation, drug, homicide, and weapon investigations) because money and funds obtained illegally try to be introduced into regular flows, complex criminal organizations deal with a different type of illegal activities, and other reasons. Structurally divided, LEA often use separate systems. Due to different requirements, such as data privacy or confidentiality, specific information on individuals and cases that would assist in the investigation remains unavailable. Agent technology should overcome data access limitations by allowing information from different databases to be used without enabling insight to end-users who do not have permission to do so.

The applied methods will be available to end-users through web technologies for the expected easier availability of information and distribution of changes and upgrades. In addition, the thesis will explore the development of the system itself. This approach also affects how evidence will be collected and processed using open-source tools. The collected data and information must be in the form that internet browsers could present.

It should be pointed out that the technologies of artificial intelligence in police applications, in which a large amount of data on persons is used, face many problems. One of the increasingly used technologies is predictive policing, which collects and analyzes historical police data on crimes committed to identify and statistically predict perpetrators and geographical locations associated with increased likelihood of criminal activity to assist police in developing policy strategies and tactics, prevention and response (Meijer & Wessels, 2019). During the research of challenges of contemporary predictive policing, whole groups of such challenges are identified such as legal and ethical, problems of user perception, transparency, biases and classification errors, and above all, the problem of input data quality based on prediction and decision is making (Vuković, Čisar, Kuk, & Popović, 2021).

The application of artificial intelligence, which in this thesis is essentially different from that in predictive policing systems, must not ignore the same challenges. When it comes to ethical oversight of artificial intelligence, it is necessary to understand the computational techniques it deploys and essentially understand the data sets over which artificial intelligence operates, how data is collected and the biases that those datasets may represent (Asaro, 2019).

Besides criminal investigation, the clustering method can also be used for image retrieval, social media, and surveillance applications. The same application of this method for a criminal investigation in this thesis introduces another significant possibility of application: personal data protection, i.e., reduction of unnecessary use of personal data, in this case, photographs and videos and other data related to them.

1.2 Subject and goals of research

Face detection and then their classification and clustering, as well as grouping of files and other related information, would enable the presentation of all available data to criminal investigators to resolve the identity, i.e., isolate potential evidence related to a particular suspect and exception of incidental findings. Therefore, the subject of the proposed scientific research is theoretical and experimental research on the application of facial biometrics in order to identify and group information about persons based on particular face descriptors and thus other data related to suspects (for example, photographs, videos, documents in which are pictures of the face of a specific person found and websites). In addition, the application of artificial intelligence, i.e., non-axiomatic reasoning, and other methods to increase the effectiveness of the face recognition process is being investigated. Emphasis is placed on reducing classification errors when applying freely available off-the-shelf software solutions (libraries) to identify faces that are considered sufficiently efficient and easy to implement in a wide range of applications. In order to achieve comprehensiveness related to the principle of thoroughness, the subject of research is also methods for improving the success of face detection in the material of a poorer quality that appears in real situations in criminal investigations (unconstrained face recognition or face recognition in the wild). It is also of interest to apply reasoning for efficient clustering when the number of classification groups is not known in advance, and limited hardware resources are available.

The individual goal of scientific research is to improve the application of appropriate off-the-shelf solutions for face recognition and descriptor extraction in identifying individuals based on those descriptors, i.e., finding ways to face clustering based on concepts of NAL and grouping data related to them using intelligent software agents enabling the identity resolution of the suspects and evidence collection. Furthermore, building a functional model on which experimental tests would be performed creates a basis for implementing research results in real systems, which would reduce the gap between the principle of thoroughness and the principle of operability and speed caused by a large amount of data and information collected by criminal police.

It also is necessary to perform detailed tests and checks to meet ethical acceptability, from the aspect of human rights, such as a three-part test (Degeling & Berendt, 2018): a suitability test, whether the applied measure will achieve the goals, and a necessity test, whether there are less intrusive measures that can give the required results, and a proportionality test, where it is weighed whether the violation is more severe than the value of the goal to be achieved, where the goal is crime prevention, the police must not go beyond the scope of their tasks. The aim is also to show that the system developed on methods and technologies applied within the thesis meets the test requirements.

The results of the research would enable the achievement of a general goal, which is their application in a broader range of activities where it is necessary to group data related to particular persons when images containing their faces are available, such as tools for digital forensics, analytical tools related to video surveillance and other.

In that sense, the primary scientific goal of the proposed research is to examine the possibility of adapting and applying the principles of NAL in solving the problem of classification and clustering faces without prior knowledge of the number of groups. In addition, the scientific goal of this research is to determine the impact of preprocessing of images of faces created in actual conditions on the effectiveness of face detection and matching.

1.3 Research hypotheses

The working hypotheses: NAL adaptation enables the improvement of the accuracy of classification, i.e., clustering, as well as the exchange of information related to clustering between points of the distribution system.

Sub-hypotheses:

- The application of agent technology enables the improvement of clustering in a distributed environment by using data from all system points without allowing end-users to see data that does not belong to them.
- The truth-value NAL concept enables the clustering method, which could be explained to end-users and thus the necessary transparency of the primary system process.

- The application of graph database and graph theory, in general, enables cluster indexing and a more efficient presentation of related data.
- Preprocessing improves detection and classification using Dlib HOG descriptors, which will positively influence unconstrained face clustering with the same type of descriptors.
- Open-source commands and applications enable the necessary file content conversions for presentation data over internet browsers.
- Using open-source commands and applications (no need to develop complex functionalities and there are mostly free of charge) by controlling them from the Python scripts allows the creation of the most rational possible system.

1.4 Contribution

Thesis contributions can be briefly summarized as follows:

- A new clustering approach has been proposed to improve the precision of pairwise clustering by NAL adaptation.
- Adaptation of the truth-value concept allows the exchange of data between points of the distributed system that aids the clustering process and creates new knowledge using NAL inference mechanism as revision and deduction.
- The proposed method clarifies the clustering process necessary for the transparency of systems involving artificial intelligence.
- Confirmation of positive influence of image enhancement techniques (preprocessing) on face recognition process, especially on detection and matching using HOG.
- The proposed method of applying Bourne-Again SHell (Bash) commands and open-source applications controlled from the Python programming language provides enhanced capabilities of the agent in collecting and preparing data for recording to a database, search, and presentation using web technologies.
- The application of agents in the way proposed in the thesis allows the use of all information of the distributed system that helps face clustering while allowing end-users to see only the data they have collected.
- Systematic literature review highlighting notable articles related to HOG face recognition, agent technology, and possibilities of open-source solutions.

1.5 Thesis Outline

The second chapter presents an overview of the situation in the field related to research subjects, primarily unconstrained face clustering, the impact of preprocessing of face images on their detection and classification/matching, and the application of graphs in identity resolution.

The third chapter contains a review of face databases and their characteristics, due to which they represent suitable material when performing experiments with classification and clustering in unconstrained conditions. The chapter presents the HOG face detection method, the method used for descriptor extraction, as well as the specific classification method. The adaptation of NAL for unconstrained face clustering in a distributed system is proposed. The last part of the chapter presents the approaches to evaluating classification and clustering.

The fourth chapter discusses the possibilities of using open-source software to collect and prepare data for presentations for grouping evidence and identity resolution. Of all the sources of digital evidence collected by the criminal police, digital forensics reports stand out in terms of the amount and variety of file types. The chapter contains an overview and testing of commands that can perform conversions of many file formats, primarily those most common in digital forensic reports, into several that internet browsers can display and their import and storage. Also, the ability to manage open-source tools from the Python programming language using subprocesses was tested.

The fifth chapter contains research on the influence of preprocessing of face images using automation of visual enhancement (normalization, sharpening, and others) on detecting and identifying faces using HOG descriptors and cosine distance. First, the methods of visual enhancement of facial images are described, and then the experimental setup is explained. Finally, the obtained results are discussed.

The sixth chapter presents a classification problem witch affects pairwise clustering. After that, the experimental setup was described to test the proposed clustering method by adapting the truth-value concept of non-axiomatic reasoning. Finally, the chapter presents the results and comparison with other clustering methods.

The last chapter describes Multi-Agent System for Advance Policing (mASAP), a decision-support system model for the identity resolution based on grouping the collected digital evidence using face clustering. The chapter contains an overview of intelligent agents and a presentation of the applied multi-agent environment of mASAP. The essential role of mASAP is presented primarily (1) practical usage of subprocesses and open-source commands, (2) agent communication to extend the clustering method, and (3) agent strategies for grouping, presentation, and sorting data.

Chapter 2

Related work

Many scientific papers deal with the preprocessing of images in face recognition. However, they do not consider the visual enhancements of images. The following are examples that represent groups of such research.

One example is when the preprocessing is intended for the appropriate technologies applied, so the conversion of color photography to black and white (shades of gray) is performed due to the use of Gabor filters to extract features that represent the face (Saabia, El-Hafeez, & Zaki, 2018). Different comparative studies also measure the impact of illumination preprocessing techniques on face recognition performance, concluding that such techniques give almost perfect results in controlled lighting variations and that better visual preprocessing results do not guarantee better recognition accuracy (Han, Shan, Chen, & Gao, 2013). Another way to approach preprocessing is to replace missing image information with existing ones. Creating an approximately symmetric virtual face image using the least degraded half of the original face image is one method (Xu, Zhang, Lu, & Yang, 2016). For example, construct an entire face with half which is not in the shade. In this way, preprocessing reduces the negative effects of heterogeneous lighting. On the other hand, it creates a set of samples for face recognition instead of using original photographs, which solves a problem with an insufficient amount of images of appropriate quality for matching.

Similar to the previous example of image preprocessing, using existing information and trained intelligent and expert systems, methods are proposed to normalize variations in poses (Haghighat, Abdel-Mottaleb, & Alhalabi, 2016). Furthermore, in some research, image preprocessing is suggested in the paper. However, improvement in recognition rate is achieved later in the process, for example, by combining two descriptors that merge at the classifier level (Nanni, Lumini, & Brahnam, 2017).

In this thesis, various techniques of image preprocessing will be applied to improve their visual appearance, such as sharpening, normalization, and resizing. For this purpose, an open-source tool will be used, enabling process automation and uniform processing of numerous files.

The central subject of the thesis research is clustering by determining pairwise similarity and NAL adaptation. An analysis of recent related work has shown that the studied processes of faces clustering could be summaries of the following activities: first extracting a deep feature from face images for more efficient data processing, then using features to calculate different distances/similarities, and finally, a method to determine which value distances/similarities classify images into the same cluster, i.e., how this is achieved without prior knowledge of the number of clusters. Methods often use a similarity graph $G(V,E)$ defined for the given data points. Nodes represent face images, while edges are calculated distance (weights). The differences in the clustering process are most significant in the methods, i.e., in the ways based on which their implementations are optimized - algorithms. Finding the optimal values of parameters intended for determining which cluster the data points belong to is a fundamental problem because choosing too large or too small values of distances and similarities produces errors that affect the method's accuracy. Therefore, classical clustering methods are upgraded with different machine learning algorithms.

Supervised machine learning algorithms mostly used for classification k-Nearest Neighbors (k-NN) and Support-Vector Machine (SVM) are also used in face clustering. Zhan et al. (Zhan, Liu, Yan, Lin, & Loy, 2018) proposed a face clustering method called Consensus-Driven Propagation (CDP) approach for k-NN graph construction using deep features for unlabeled data clustering them into "pseudo-classes". The approach uses a "committee" module that provides pairwise multi-view information and a "mediator" module that aggregates all the information for a final decision. Shi et al. (Y. Shi, Otto, & Jain, 2018) proposed Conditional Pairwise Clustering (ConPaC) which directly estimates the adjacency matrix based on the similarities between face representations obtained by coaching ResNet deep network. The method is further extended to the semi-supervised clustering using a set of pairwise constraints ("must-link" pairs and "cannot-link" pairs). There is also a version of ConPaC based on k-NN that is suitable for large datasets. Yang et al. (Yang et al., 2019) propose a supervised approach "learning to cluster" using the Graph Convolutional Networks (GCN) on the set of extract deep features to capture cluster patterns on an affinity graph. The approach uses cosine similarity to find k-NN for each sample. After that, a connection between neighbors creates an affinity graph. This approach of clustering is in a divisive hierarchical manner, and using detection and segmentation GCN modules seeks to handle clusters of complicated structures. Lin et al. (Lin, Chen, Ranjan, et al., 2018) propose a face clustering algorithm, namely the Proximity-Aware Hierarchical Clustering (PAHC), which calculates PAHC similarity scores between deep features using SVM. SVM is trained with the corresponding neighborhoods of the samples forming clusters by thresholding the similarity scores.

Otto et al. (Otto, Wang, & Jain, 2017) used 320-dimensional output as a feature vector from a Convolutional Neural Network (CNN) based face model and developed a version of the rank-order clustering algorithm, the form of agglomerative hierarchical clustering, that measures pairwise similarity based on the number of shared nearest neighbors to achieve the desired scaling and precision clustering. The method uses an internal per-cluster quality measure to rank individual clusters for manual exploration.

Instead of computing the linkage likelihood by heuristic metrics, Wang et al. (Z. Wang, Zheng, Li, & Wang, 2019) propose a Linkage-Based Face Clustering (LBC) clustering algorithm which uses the classification ability of the GCN in determining whether two nodes belong to the same cluster concluded based on context. The algorithm predicts linkages between an instance and its nearest neighbors (pivot and Instance Pivot Subgraph around that pivot), determining the local context. It uses GCN for reasoning to conclude which pivot-neighbor pair from the subgraph should be connected. Qi at al. (Qi et al., 2021) propose a solution that improves the performance of face clustering by overcoming limitations in learning ability due to the use of the shallow GCN structure by deepening the layer. First, the k-NN algorithm is used to create subgraphs, followed by the Residual Graph Convolutional Network (RGCN) algorithm to determine the probability of connecting two nodes to avoid the vanishing gradient problem, which is the bottleneck in GCN. Tapaswi et al. (Tapaswi, Law, & Fidler, 2019) propose a supervised approach Ball Cluster Learning (BCL) which carves the embedding space into balls of equal size (representing a cluster) where the ball radius (which is the result of training, or learned value) corresponds to the threshold used as a stopping criterion of the clustering algorithm.

One of the methods to perform clustering without prior knowledge of the number of clusters and discover clusters of arbitrary shape and size is the application of density-based algorithms. Guo et al. (Guo et al., 2020) propose a Density Aware Feature Embedding Network (DA-Net) which utilizes both local and non-local information, to learn a robust feature embedding. For this purpose, DA-Net uses two networks, first one GCN, aiming to learn feature embeddings from the contextual information contained in the local neighborhood, and then incorporates non-local information using a long-range chain network based on CNN attention network. The long-range chain uses the density peak to connect the samples more likely from the same class. Ahmed et al. (Ahmed, Hemayed, & Fayek, 2020) establish a face clustering system on an updated version of the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) with a post-clustering optimization procedure that applies the minimum distance rule between the 200-dimensional feature vector. Wang et al. (L. Wang, Ding, & Jia, 2019) propose a spectral clustering based on the passing message, using a density adaptive similarity measure to improve performance. Algorithm use message passing to describe the relations between data points and obtain high-quality cluster centers in affinity propagation. Lin et al. (Lin, Chen, Castillo, & Chellappa, 2018) proposed an unsupervised clustering algorithm for unconstrained face images without knowing the number of subjects called Deep Density Clustering (DDC) based on measuring density affinities between local neighborhoods in the feature space.

Chang et al. (Chang, Pérez-Suárez, & González-Mendoza, 2019) present a graph-based method for clustering unconstrained faces that uses a 128-dimensional face descriptor from ResNet to calculate the Euclidean distance between two descriptors. Next, their implemented the Chinese whispers clustering method uses a single distance threshold value that gave the best possible results according to their research. Wang and Deng (M. Wang & Deng, 2020) propose an unsupervised Clustering-based Domain Adaptation (CDA) method designed for a face recognition task in which the source and target domain do not share any classes. CDA applies a simplified spectral clustering algorithm that requires neither overlapping classes nor the number of target classes. The proposed method uses extracted deep features to construct a similarity matrix of dimensions corresponding to the number of images in the target domain, which contains cosine similarity between target representations. A clustering graph is built based on the matrix where the nodes represent face images. Edges signify two images have larger cosine-similarity (over a certain threshold for edges). Each connected component with at least three nodes in the graph is saved as a cluster (identity). The method handles scattered points clustered around the corresponding prototype, computed by the average representation of all target samples in one cluster obtained by the connected component, calculating cosine similarity with a scattered point, which should be greater than a certain threshold.

Shi et al. (X. Shi, Guo, Xing, Cai, & Yang, 2018) proposed a self-learning framework for face clustering simulating how people learn. The authors use a 2D face image matrix to apply self-paced learning based on the following observations - the learning process ranges from easy to complex samples, and prior knowledge can change with increasing learned experience. Greater prior knowledge leads to better prediction accuracy.

The new approach to unconstrained face clustering proposed in this thesis applies methodological solutions found in related work, such as pairwise clustering based on cosine distance between two 128 dimensional descriptors and graphs. Still, it upgrades it not by building a similarity graph based on the particular threshold but by the threshold of the truth value. The adapted truth-value concept from NAL maximizes the distance value between the two feature vectors.

The adaptation of truth-value and then application of some of the NAL mechanism enables clustering improvement by exchanging information in a distributed environment. Working in a distributed environment allows data points that would remain outliers to be linked to information obtained from other nodes in the system and correct existing clusters as needed. Although a specific out-of-shelf method for face recognition and cosine similarity has been applied in the thesis, clustering with the proposed algorithm can be performed on top of other descriptors and types of distances. Need to be noted that the algorithm's complexity is characteristic of pairwise clustering and is at the level of most proposed methods.

Regarding graph-based identity resolution methods, there are examples of research that include not just face images but various data of individuals with a focus on social media and identifying their users (Shu, Wang, Tang, Zafarani, & Liu, 2017; Phillips, Amirhosseini, & Kazemian, 2020). These studies use the topological advantages of graph structure in analyzing data organized on the principles of social networks. Based on appropriate attributes, data from several different profiles and networks are linked to determine whether it is a single user. In addition, some researchers extract features from the content attributes of users on social networks, such as part-of-speeches, symbols, emoticons, and common high-frequency words for identity resolution (Srivastava & Roychoudhury, 2020).

But one of the problems of identity resolution based on different textual data is the multilingual information (data from Europol databases, for example, and data from local country databases) and the use of different fonts. In this sense, grouping data based on facial biometrics is a rare universal solution that transcends the language barrier. In addition, in this paper, data indexing is one of the reasons for which graphs are used, primarily database for graphs.

Chapter 3

Materials and methods

3.1 Face image databases

For the purposes of the experiments in this thesis, sets of face images from 5 publicly available databases were used: IMDb-Wiki database (Rothe, Timofte, & Van Gool, 2018), Extended Yale Face Database B (Georghiades, Belhumeur, & Kriegman, 2001), Labeled Faces in the Wild (LFW) database (Huang & Learned-Miller, 2014), YouTube Faces (Wolf, Hassner, & Maoz, 2011) and IARPA Janus Benchmark-B Face Dataset (IJB-B) (Whitelam et al., 2017). The last sixth database, internally called SSIM, was created for unconstrained face detection, matching, and clustering research to simulate evidence collected by digital forensics or otherwise during a criminal investigation.

In order to provide a ground-truth dataset and a dataset with equal impact on the experimental results, the size of the databases was reduced. For example, the analysis of IMDb-Wiki databases revealed problems with the wrong labeled faces, which required correction manually. Also, on many images, especially in the YouTube Faces database, there are several faces in the same image, with only one labeled, preventing accurate evolution in preprocessing and clustering.

Table 3.1: Characteristics of face images in databases.

			Invariant to						
Database	**№ images**	**№ people**	**Expression**	**Age**	**Pose**	**Illumin.**	**Occlusion**	**Low res.**	**Out of focus**
SSIM	6,000	328	No	No	No	No	No	No	No
IMDb-Wiki	10,000	330	No	No	No	No	No	No	Yes
LFW	3,500	143	No	No	No	No	No	No	Yes
Yale B	4,200	28	Yes	Yes	No	No	Yes	Yes	Yes
YouTube Faces	4,383	166	No	No	No	No	No	No	No
IJB-B 64	2,080	86	No	No	No	No	No	No	No
IJB-B 128	5,219	164	No	No	No	No	No	No	No
IJB-B 512	18,172	628	No	No	No	No	No	No	No
IJB-B 1024	36,071	1133	No	No	No	No	No	No	No

Only images that could be confirmed to be related to the correct person and images with one person are taken to create the most accurate datasets from the face database. The only difference is made with the IJB-B database, where significant changes are not made to avoid interference with cluster evaluation protocols. Tab. 3.1 shows summarized characteristics of the each database.

3.1.1 Imdb-Wiki database

The IMDb-Wiki database contains 523,051 photos of celebrities collected crawling from IMDb and Wikipedia websites. The database is interesting for this thesis because it contains photographs with significant differences in the age of the same persons (from childhood to adulthood). Because faces belong primarily to actors, the transformations in appearance due to trends or movie roles often make recognition a challenge. Images contain scenes from films that can be said to simulate unconstrained conditions. Also, they vary in image resolution too. The authors of this database downloaded information about the name, date of birth, gender, and all images related to a person from IMDb celebrities' profiles. It was assumed that when there is one person on an image within profile data, it is probably a photo of the person whose profile it is. In contrast, when it comes to pictures with more than one person, they only use images where the second strongest face detection is below a threshold.

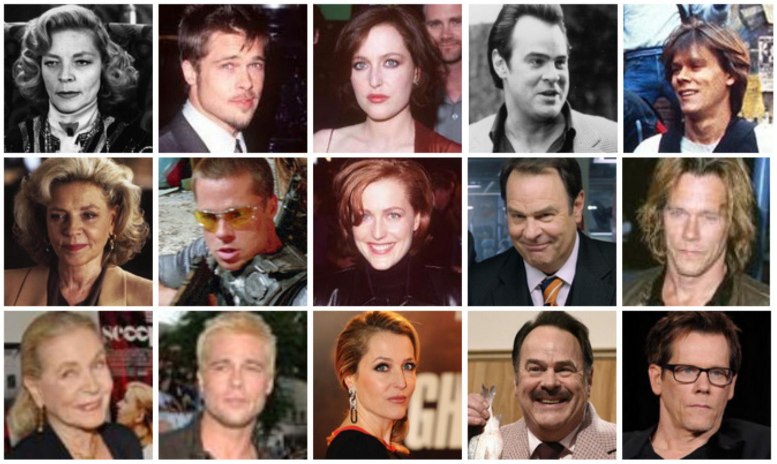


Figure 3.1: Representative examples from the Imdb-Wiki database.

Fig. 3.1 shows an example of face images from the IMDb-Wiki with some of the essential features of the database that stand out: noticeable differences in the age of the same person in different pictures and their distinctive visual transformation because of the roles and some events they attend.

3.1.2 Extended Yale Face Database B

The Extended Yale Face Database B is a set that initially contained 16,128 photos of the faces of 28 different people under nine poses and 64 illumination conditions. The authors created the database for the systematic testing of face recognition methods under extensive variations in illumination and pose, without the influence of different facial expressions, aging, and occlusion (Georghiades et al., 2001). Examples in Figs. 3.2 show the features of the base on which it stands out concerning other databases: although different lighting conditions and pose changes are simulated, all images are of the same quality, in focus and high resolution.

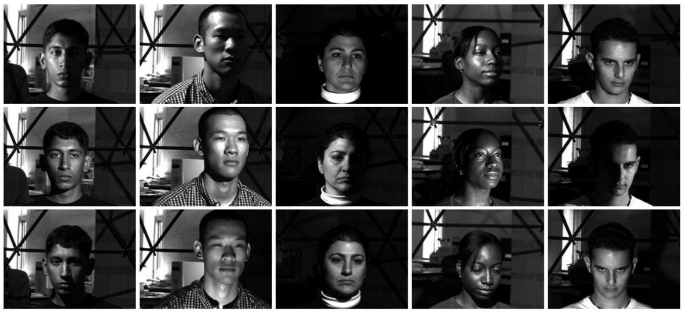


Figure 3.2: Representative examples from the Extended Yale Face Database B database.

3.1.3 Labeled Faces in the Wild

LFW database's main contribution is to provide a large number of relatively unconstrained images with the diversity of conditions under which they were taken (Huang, Mattar, Berg, & Learned-Miller, 2008). However, illumination conditions are mainly optimal.

The development of LFW began with newspaper photographs depicting people in different poses and facial expressions under different lighting, collected as part of the Berkeley project "Faces in the Wild." One of the database characteristics is that the resolutions of face photos are mainly less than 150 pixels. The database's selection is related to the part of criminal police work on identity resolutions by OSINT, which is similar to the collection process in which the authors obtained this set of photos. The database contains 13,233 images belonging to 5,749 people, of which 1,680 have two or more pictures. Fig. 3.3 shows examples from the LFW database.

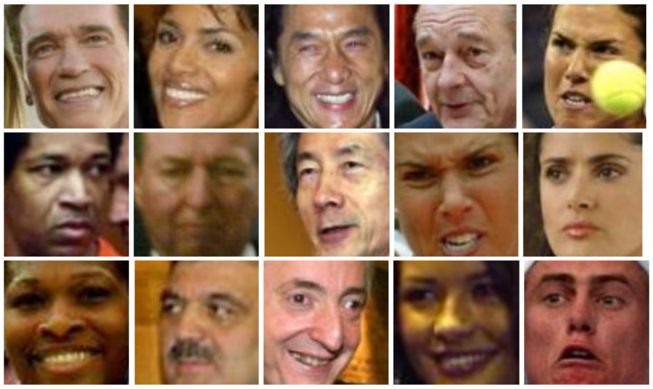


Figure 3.3: Representative examples from the LFW database.

3.1.4 YouTube Faces

The YouTube Faces database was created using 5,749 names from the LFW data set that was used for YouTube searches to find matching videos with the same individuals. First, the top six results of each search are downloaded, and then the videos are split into frames at 24 fps. Finally, the database's authors manually check the correctness of the video's face labels and the prevention of their duplication, ending with 18,899 original videos and 3,345 individuals on 3,425 videos of 1,595 subjects. What characterizes most face images in sets is low resolution (characteristic of videos compared to photos); faces are often blurred as a result of poor quality of the original record, further degraded by subsequent conversions (often from analog to digital format); multiple persons in addition to the subject on the video recognizable enough to be detected (creates a problem with labeling such persons after detection).

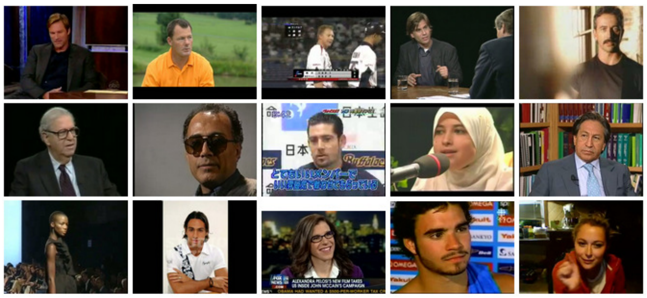


Figure 3.4: Representative examples from the YouTube Faces database.

Fig. 3.4 shows examples from the YouTube Faces database that point to distinctive features compared to other databases: originating from video frames makes face images significantly lower quality, especially in resolution, than ones in other databases.

3.1.5 IARPA Janus Benchmark-B Face Dataset

IJB-B database contains 1,845 subjects on 21,798 images and 7,011 videos. About half of the total number of images are still images with faces. In contrast, the other half are faceless images for face detection testing. The images feature faces from all continents, in different poses, with different resolutions, image quality, facial occlusion, and greater illumination variation. Although there is the latest version of the set (version C), the thesis will use the IJB-B data set to compare with other solutions. When it was current, IJB-B was the largest annotated unconstrained joint detection, recognition, and clustering benchmark dataset and the most geographically diverse.

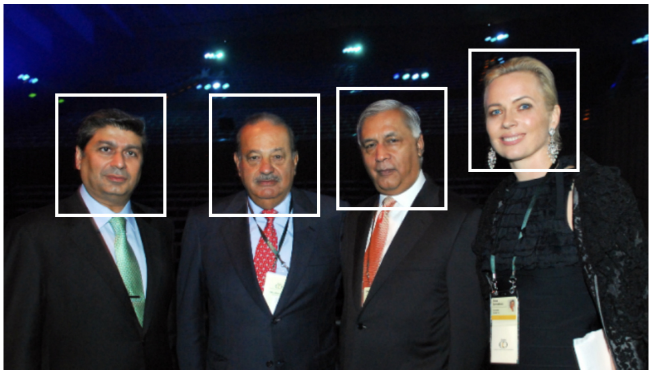


Figure 3.5: Example from the IJB-B database.

Fig. 3.5 shows an example from the IJB-B database, like YouTube Faces, but different from others presented in the thesis, contains full photos, not just images of individual faces. In addition, each image and video frame is annotated with the bounding box locations of all faces, in Fig. 3.5 represented by white squares, which is, for example, an issue for YouTube Faces labeling where there are multiple faces on a still image of a video frame.

Four sets of face images created by the authors of the IJB-B database were used for the experiments. Those sets are intended for testing clustering algorithms (the last four rows in Tab. 3.1).

3.1.6 SSIM database

The database of 6,000 face images internally called SSIM was created to simulate the material obtained by digital forensics units or by covert surveillance police operations. A preview of the SSIM database can be seen in Fig. 3.6.

Figure 3.6: Preview of SSIM database.

Within the SSIM database, 328 different persons were visually identified. For most identified individuals, the photos were taken over a period longer than 15 years, so significant physical appearance changes due to aging can be noticed. Representative examples of photographs from the data set used in the experiment are shown in Fig. 3.7.

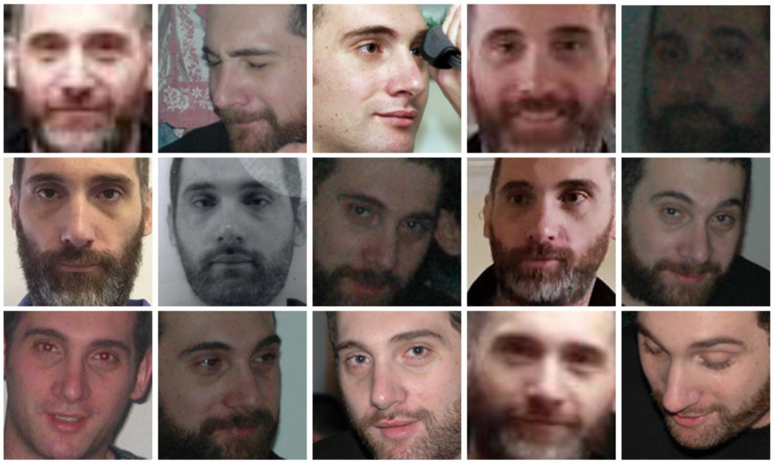


Figure 3.7: Representative examples from the SSIM database.

Face images vary in resolution from 31 x 31 pixels to 2385 x 2385 pixels. The quality of images is different, and considering various resolutions, is another essential feature of pictures collected for criminal investigation purposes. It is necessary to point out that apart from a few professional ones, all other images are created with amateur or consumer equipment. The illumination is mainly artificial and placed frontally but is for low-intensity sources. Due to the lower intensity of the light source, the faces are often in shadow. Image sharpness is generally satisfactory for visual recognition, but images that are more or less out of focus predominate. Although front-facing images represent the majority in the set, there are different poses, with many photos with semi-profiles and profiles. After conducting the detection experiment, it needs to point out that the SSIM database exists only as a feature vector set.

3.2 Face recognition

The face recognition system generally consists of face detection, feature extraction, and feature matching (Jayaraman et al., 2020; Dong et al., 2020). In the thesis, the Dlib version of a HOG face detector will be used, along with ResNet feature vector extractor and cosine distance for matching.

3.2.1 Histogram of Oriented Gradients

The HOG is an image descriptor proposed by Dalal and Triggs (Dalal & Triggs, 2005), who were inspired by a Scale-Invariant feature transform (SIFT) descriptor. It is a local feature-based method of face recognition, which is invariant to 2D rotations and scaling (Jayaraman et al., 2020). The idea behind the HOG descriptor is that the distribution of intensity gradients can characterize the shape of objects in images. (Kumar, Happy, & Routray, 2016).

The image needs to be divided into blocks of pixels, i.e., cells, to obtain a HOG descriptor. A histogram of the orientation of edges, i.e., gradient directions, for each cell is calculated. The first step is to calculate the amplitude of the gradients of each pixel of the image $I(x,y)$ in the horizontal and vertical directions, based on which the magnitude of the gradients $|G|$ and the orientation angle γ are obtained (Nhat & Hoang, 2019; Manju & Radha, 2020; B. Li & Huo, 2016).

The following formulas are used to calculate the magnitude and orientation:

$$dx = I(x+1,y) - I(x-1,y) \tag{3.2.1}$$

$$dy = I(x,y+1) - I(x,y-1) \tag{3.2.2}$$

$$|G| = \sqrt{dx^2 + dy^2} \tag{3.2.3}$$

$$\gamma(x,y) = tan^{-1}(\frac{dx}{dy}), \gamma \in [-\pi, \pi]\,. \tag{3.2.4}$$

Split the orientation range into k bins, computation the histogram within the cell (HC), and then integration it into the block by combining $b1 \times b2$ cells to obtain the histogram of the block HB (Manju & Radha, 2020):

$$HC(k)_i = HC(k)_i + |G|,\ if\ I(x,y) \in cell_i\ and\ \gamma \in bin(k) \tag{3.2.5}$$

$$HB_j = \{HC_1, HC_2, ...HC_{b1\times b2}\}. \tag{3.2.6}$$

The orientation histogram of each cell is constructed by accumulating the gradient magnitudes related to the corresponding class interval concerning the corresponding orientation angle (Nhat & Hoang, 2019). In addition, the blocks overlap so that each cell contributes more than one block (Laucka, Adaskeviciute, & Andriukaitis, 2019).

Due to variations in the image, such as brightness or contrast between foreground and background, the gradient values also vary significantly, which is why L2-norm block normalization is applied (Manju & Radha, 2020; B. Li & Huo, 2016; Kapoor, Gupta, Jha, & Kumar, 2018):

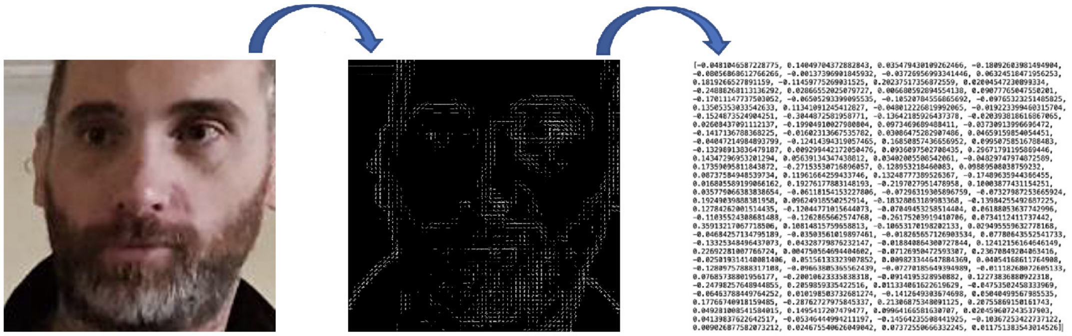
$$[h]NHB_j = \frac{HB_j}{\sqrt{\|HB_j\|_2^2 + e^2}}. \tag{3.2.7}$$

The NHB_j is a normalized block, while e is a constant with a purpose to avoid division by zero. Integration of normalized blocks builds a HOG descriptor. Although not currently state-of-the-art, the Dlib HOG face detector's implementation is an off-the-shelf solution that achieves reasonable precision and processing speed (Johnston & de Chazal, 2018).

3.2.2 Feature extraction and matching

The descriptor extraction transforms multidimensional arrays of discrete spatial units (pixels) into lower dimensions while retaining enough data to represent the information. The face is represented by the vector, which contains the unique features of the face (Jayaraman et al., 2020).

After creating the HOG descriptor Dlib libraries use ResNet, a deep residual network for 128-dimensional face features vector extraction. A simplified example of the whole process is illustrated in Fig. 3.8 showing the face image followed by the representation of the HOG descriptor from which the feature vector of face landmarks was extracted. The visual representation of the HOG shown in Fig. 3.8 was created using the *scikit-image* Python library (Van der Walt et al., 2014). Extraction yields a 128-dimensional feature vector that is stored in a database and used further in the matching and clustering process.



```
[-0.0481046587228775, 0.14049704372882843, 0.035479430109262466, -0.18092603981494904,
-0.0805686861276626б, -0.001373969018459З2, -0.03726956993341446, 0.06324518471956253,
0.1819266527891159, -0.11459775269031525, 0.20237517356872559, 0.020045472308993З4,
-0.24888268113136292, 0.02866552025079727, 0.006680592894554138, 0.09077765047550201,
-0.17011147737503052, -0.06505293399095535, -0.10520784556865692, -0.09765323251485825,
0.13505353033542633, 0.11341091245412827, -0.048012226819992065, -0.019223399460315704,
-0.1524873524904251, -0.30448725819587771, -0.13642185926437378, -0.020393818616867065,
0.02608437091112137, -0.19904910027980804, 0.09734696894884ll, -0.037309139966964 72,
-0.1417136788368225, -0.016023136675357 82, 0.030864752829074 86, 0.046591598540544 51,
-0.040472149848937 99, -0.124143943190574 65, 0.168508574366569 52, 0.099507585167884 83,
-0.132989138364791 87, 0.009299442172050476, 0.093609750270 8435, 0.29671791195869446,
0.14347296953201294, 0.056391343474388 12, 0.034020055085420 61, -0.048297479748725 89,
0.17359095811843872, -0.27153530716896057, 0.128953218460083, 0.098895080387592 32,
0.08737584948539734, 0.11961664259433746, 0.13248777389526367, -0.17489635944366455,
0.016805589199066162, 0.19276177883148193, -0.2197027951478958, 0.10003877431154251,
0.035779066383838654, -0.06118154153227806, -0.07296319305896759, -0.07327987253665924,
0.19249039888381958, 0.09624918550252914, -0.18328063189983368, -0.13984255492687225,
0.12784262001514435, -0.12044771015644073, -0.07049453258514404, 0.06188053637742996,
-0.11035524308681488, -0.12628656662574768, -0.26175203919410706, 0.07341124117374 42,
0.35913217067718506, 0.10814815759658813, -0.10653170198202133, 0.029495559632778168,
-0.046842571347951 89, -0.03503561019897461, -0.018265657126903534, 0.07780643552541733,
-0.13325348496437073, 0.04328779876232147, -0.01884086430072 7844, 0.12412156164646149,
0.22692281007766724, 0.004750564694404602, -0.07126950472593307, 0.23670849204063416,
-0.025019314140081406, 0.05156133323907852, 0.009823344647884369, 0.04054168611764908,
-0.12809757888317108, -0.09663805365562439, -0.07270185649394989, -0.011182680726051 33,
0.07685738801956177, -0.20010623335838318, -0.09141953289508 82, 0.12273836880922318,
-0.24798257648944855, 0.20598593354225 16, 0.011334061622619629, -0.04753502458333969,
-0.06463788449764252, 0.010198503732681274, -0.14126493036746 98, 0.05040499567985535,
0.17766740918159485, -0.28762727975845337, 0.21306875348091125, 0.20755869150161743,
0.049281008541584015, 0.1495417207479477, 0.09964166581630707, 0.020459607243537903,
0.04139837622642517, -0.05346444994211197, -0.14564235508441925, -0.10367253422737122,
0.009026877582073212, 0.024675540626049042, 0.07372550666332245, 0.01175138354301452 6]
```

Figure 3.8: Illustration of the feature vector extraction process from a face image.

For experiments regarding image clustering and the examing influence of image enhancement on face matching, after obtaining feature vectors, face matching was performed by calculating the similarity of ϑ, i.e., cosine distance ordered pairs of vectors that make up the currently obtained and all previously obtained face feature vectors.

$$\{(\vec{V}_i, \vec{V}_j) | \vec{V}_i, \vec{V}_j \in \mathbb{R}^{128},\ i \neq j\} \tag{3.2.8}$$

$$\vartheta \ = cos\Theta = \frac{\vec{V}_i \cdot \vec{V}_j}{||\vec{V}_i|| \, ||\vec{V}_j||} \tag{3.2.9}$$

Calculation of the cosine placing the similarity value between 0 and 1. With the greater similarity between features (smaller angle between the two feature vectors), the cosine distance value approaches 1. Conversely, the smaller the similarity of features is (the greater the angle value), the calculated distance value weighs toward 0.

3.3 The proposed approach to face clustering

The reasoning system applied in this thesis is based on the concepts of evidence and truth-value taken from NAL. NAL represents a formalism for the reasoning system in the domain Artificial General Intelligence (AGI) and includes a symbolic grammar, a set of inference rules, and a semantic theory (Mitrović, 2015). The difference between axiomatic and non-axiomatic logic is reflected in the following (P. Wang, 2013):

- In axiomatic logic, initial knowledge is presented as axioms, and theorems provide all solutions to problems through deduction where the truth of axioms guarantees the theorem's truth. In contrast, in NAL all knowledge can be challenged by new evidence, and concerning the limited resources, it can be based on a piece of all relevant knowledge.
- Axiomatic logic is suitable for an idealized situation, while NAL is more suitable for realistic conditions.

Knowledge is uncertain and not necessarily consistent. New evidence can be accepted at any time and can affect the veracity of any existing statement because the system usually does not have enough resources in terms of space and time to consult the entire knowledge base or apply all rules of reasoning when solving the problem (Mitrović, 2015). What differentiates NAL from other formalisms used in computing is that it is a logic of terms where the statement is given in the form subject-copula-predicate, where subject and predicate are terms (T) (Sredojević, Vidaković, Ivanović, & Mitrović, 2017).

$$S \to P \tag{3.3.1}$$

It should be noted that the original notation used in theory formalization (P. Wang, 2013, 2009b) in this thesis will be modified for practical reasons and does not represent the essence of the adaptation of the concept.

3.3.1 The concept of evidence and truth-value

The inheritance copula, "$\to$," is a binary relation of one term to another and defined by being reflexive and transitive (P. Wang, 2010). The subject *S* represents a specialization of the predicate *P*, while *P* is said to be the generalization of *S*, which roughly corresponds to "S is a kind of P" (Mitrović, 2015). The system experience *K* is an empty set, and a finite set of statements where the subject term and the predicate term are different (P. Wang, 2006).

$$K = \{T_i \to T_j \; \forall \; T_i, T_j \in T, \; i \neq j\} \tag{3.3.2}$$

Inheritance statements in the form $T_i \to T_i$ is called tautology and is excluded from the systemic experience K for redundancy reasons (P. Wang, 2013; Mitrović, 2015) hence the condition $i \neq j$ when defining K in Eq. 3.3.2. The experience of K is extended by sentences derived from K based on the transitivity of the inheritance relation, which provides systemic knowledge $K^* = \{T_i \to T_k, T_k \to T_j, T_i \to T_j\}$, so a statement can be said to be true if it is in K^* (P. Wang, 2013, 2006). The statement is false if the relation does not exist.

Given experience K, the meaning of a term T consists of its extension T^X and intensions T^I (P. Wang, 2009a).

$$T^X = \{x \mid x \in V_K,\ x \to T\} \tag{3.3.3}$$

$$T^I = \{x \mid x \in V_K,\ T \to x\} \tag{3.3.4}$$

V_K is the system's vocabulary and consists of all terms that appear in the K experience, except the term T if $T^X = T^I = \{\}$ which is meaningless to the system.

All that has been said so far represents the so-called NAL-0 or Inheritance Logic (IL)-1 and is an idealized version of NAL, in the sense that it is similar to NAL in language, semantics, and inference rule. However, it assumes sufficient knowledge and resources, and its purpose is to define the NAL (P. Wang, 2006, 2010).

Non-AxiomaticReasoning System (NARS), a unified AGI system which works under the assumption of insufficient knowledge and resources is utilizes the NAL for inference (Hammer, Lofthouse, & Wang, 2016). The central issue in NARS is to treat truthfulness as a matter of degree, so the concept of evidence is first defined within NAL and then truth-value as a function of available evidence, which together lays the foundations for formal reasoning in NAL (P. Wang, 2006; Mitrović, 2015). For a statement $T_1 \to T_2$ positive evidence (e^+) and negative evidence (e^-) are defined as follows:

$$e^+ = \{T_1^X \cap T_2^X\} \cup \{T_1^I \cap T_2^I\} \tag{3.3.5}$$

$$e^- = \{T_1^X \setminus T_2^X\} \cup \{T_1^I \setminus T_2^I\} \tag{3.3.6}$$

$$\omega^+ = |e^+| = |T_1^X \cap T_2^X| + |T_1^I \cap T_2^I| \tag{3.3.7}$$

$$\omega^- = |e^-| = |T_1^X \setminus T_2^X| + |T_1^I \setminus T_2^I| \tag{3.3.8}$$

$$\omega = \omega^+ + \omega^- = |T_i^X| + |T_j^I| \tag{3.3.9}$$

The cardinality of the set of positive evidence is denoted with ω^+, negative evidence with ω^-, while the total number of evidence is represented by the value of the variable ω.

Truth-value of a statement is represented as a pair of real numbers in $[0,1]$ called frequency f and confidence c (P. Wang, 2010).

$$f = \frac{\omega^{+}}{\omega} \tag{3.3.10}$$

$$c = \frac{\omega}{\omega + k} \tag{3.3.11}$$

Frequency represents the proportion of positive evidence among all evidence, while confidence is the ratio of the amount of evidence available now and in the near future. Binary statement plus truth-value becomes a multi-valued "judgment" (P. Wang, 2013).

$$T_1 \rightarrow T_2 \langle f, c \rangle \tag{3.3.12}$$

Evidential horizon k in Eq. 3.3.11 is a constant representing the amount of evidence that will come in the near future. The greater the influx of such evidence, the more current the value of the statement's truth will be. On the other hand, the more evidence there is, the higher the confidence, that is, new evidence will have little effect on the system's belief. The k constant for different systems can have different values, and it is difficult if at all possible, to determine the best (P. Wang, 2010). However, it should be noted that its role is to prevent the system from comparing the potentially infinite future to the relatively short past (Mitrović, 2015). In this thesis, k will play an essential role in acquiring new knowledge from a distributed environment, allowing the comparison of amounts of evidence available to nodes.

The knowledge base based on NAL statements can be represented by a so-called property graph G, which can contain different types of edges for representing inheritance and similarity, and which each vertex or an edge can have any number of key → value pairs attached to it (Sredojević et al., 2017).

$$G = (V, E) \tag{3.3.13}$$

$$V = V_K \tag{3.3.14}$$

$$E \subseteq \{(T_i, T_j) | (T_i, T_j) \in K,\ T_i, T_j \in V_K,\ i \neq j\} \tag{3.3.15}$$

3.3.2 Adaptation of truth-value for face clustering purpose

Truth-value requires certain adaptations for reasoning use in face clustering, primarily because algorithms work with feature vectors instead of terms. The relationship between two feature vectors is always symmetrical. NAL recognizes such a relation, i.e., similarity statement where two terms are connected with symmetric inheritance relation "$T_1 \leftrightarrow T_2$" defined with two inheritance statements as $T_1 \leftrightarrow T_2 \equiv (T_1 \rightarrow T_2),\ (T_2 \rightarrow T_1)$ (P. Wang, 2013, 2009a). Applied to feature vectors Eq. 3.3.16 is obtained.

$$\vec{V}_1 \leftrightarrow \vec{V}_2 \tag{3.3.16}$$

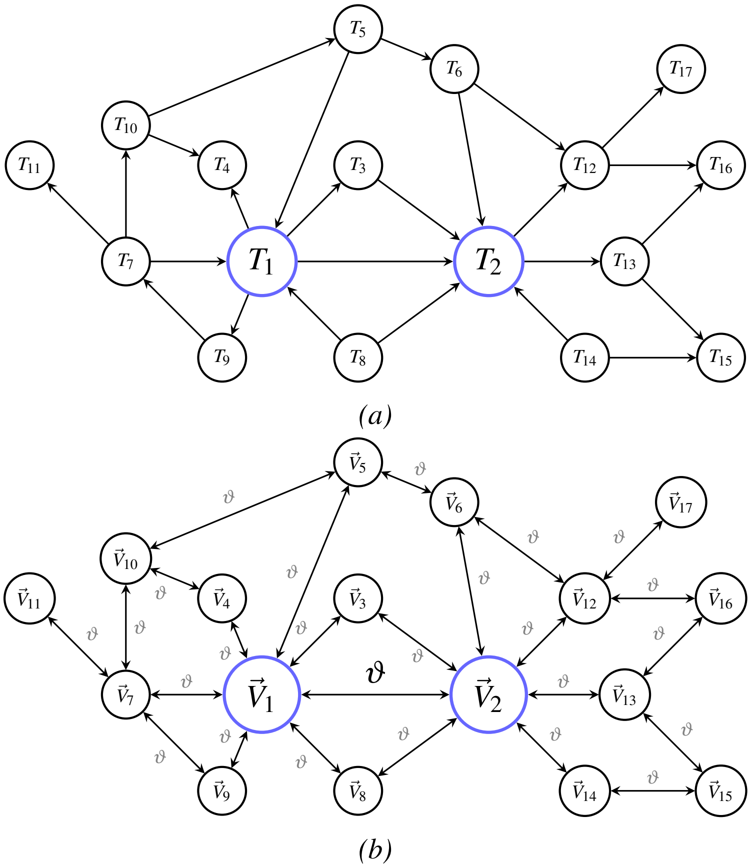
The system experience K will be represented by a graph with changes in notation, especially when it comes to vertex. Since clustering uses images of faces, i.e., feature vectors, instead of terms, the vocabulary is replaced by the album of the system V_A ($V_A \approx V_K$). Edges represent similarities between ordered pairs of feature vectors $(\vec{V}_i, \vec{V}_j)$.

$$G = (V, E) \tag{3.3.17}$$

$$V = V_A \tag{3.3.18}$$

$$E \subseteq \{(\vec{V}_i, \vec{V}_j) | (\vec{V}_i, \vec{V}_j) \in K,\ \vec{V}_i, \vec{V}_j \in V_A,\ i \neq j,\ \vartheta_{\vec{V}_i, \vec{V}_j} > \vartheta_t\} \tag{3.3.19}$$

Fig. 3.9b shows a graph that is part of the experience of the clustering system where nodes are denoted by the most prominent parameter, namely the feature vector (collection in the graph database contains beside feature vector other information about face image and source file) and bidirectional edges, thus indicating similarity (cosine similarity is the weight assigned to each edge). Fig. 3.9a and Fig. 3.9b show differences in the nature of the feature vector and terms and how they affect changes in the representation of system experience.



(a)

(b)

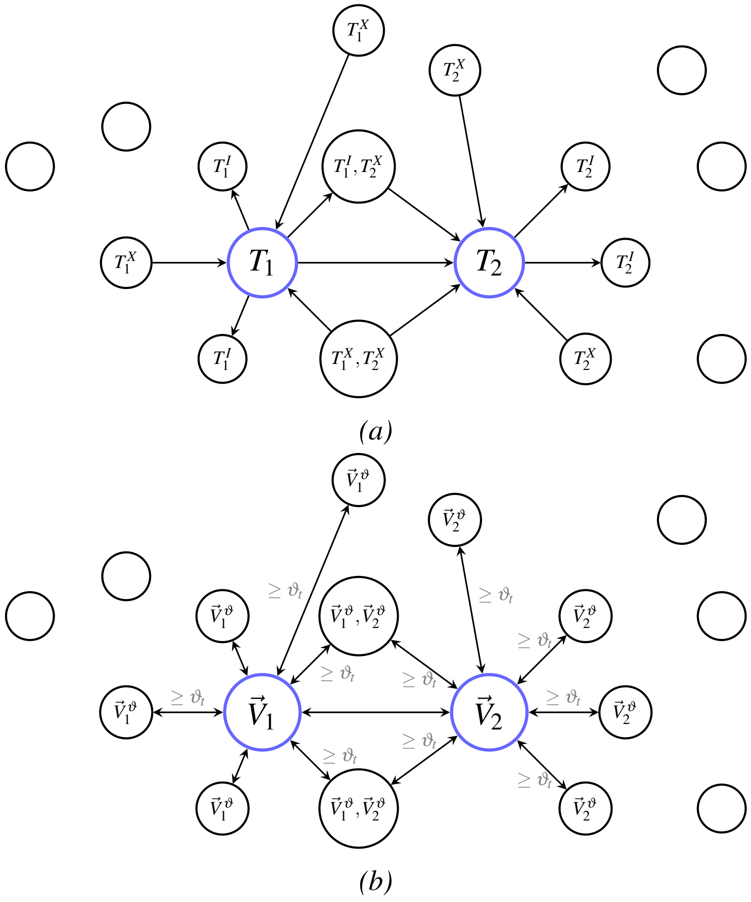
Figure 3.9: Graph representing a sample of system experience K in a) before and b) after adaptation.

Due to the symmetrical nature of the similarity relation between feature vectors, there are no intensions and extensions, but cosine distance value ϑ exceeds the corresponding ϑ_t threshold. These differences are essential in adapting non-axiomatic reasoning for the needs of face clustering.

The set consisting of face feature vectors $\vec{V}_i^{\vartheta}$ that are similar to a particular vector $\vec{V}_i$ is determined as follows:

$$\vec{V}_i^{\vartheta} = \{x \mid x \in V_A,\ x \leftrightarrow \vec{V}_i,\ \vartheta_{x,\vec{V}_i} \geq \vartheta_t\} \tag{3.3.20}$$

Fig. 3.10 shows a graph representing the difference between intension and extension in NAL and the solution applied after adaptation.



(a)

(b)

Figure 3.10: Graph representing a) intensions and extensions of terms T_1 and T_2 and b) "similar feature vector" $\vec{V}_i^{\vartheta}$ to vectors $\vec{V}_1$ and $\vec{V}_2$ replacing intesions and extension after adaptation.

Positive evidence e^+ of similarity between two feature vectors is the intersection of a set of vectors that satisfy the similarity conditions with both vectors being compared. In contrast, negative evidence e^- is obtained from the elements that distinguish the two sets. The amount of evidence used to calculate cardinality *omega* of the intersection or the difference between the sets.

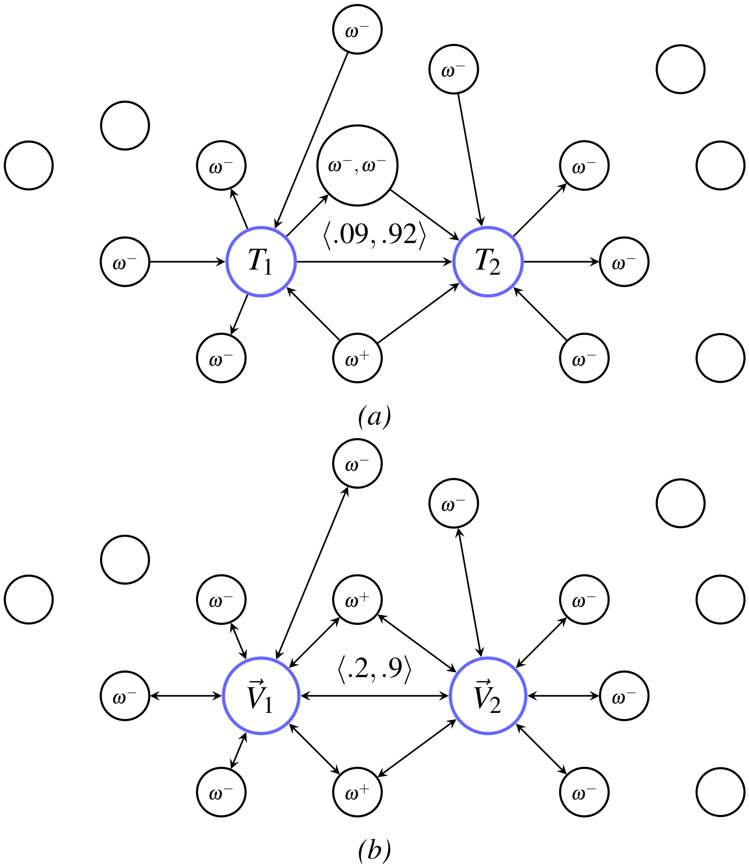
$$e^+ = \{\vec{V}_1^{\vartheta} \cap \vec{V}_2^{\vartheta}\} \tag{3.3.21}$$

$$e^- = \{\vec{V}_1^{\vartheta} \setminus \vec{V}_2^{\vartheta}\} \tag{3.3.22}$$

$$\omega^+ = |e^+| = |\vec{V}_1^{\vartheta} \cap \vec{V}_2^{\vartheta}| \tag{3.3.23}$$

$$\omega^- = |e^-| = |\vec{V}_1^{\vartheta} \setminus \vec{V}_2^{\vartheta}| \tag{3.3.24}$$

Finlay the Eq. 3.3.10 and 3.3.11 from NAL used to getting truth-values frequency f and confidence c. Fig. 3.11 presents a difference between the cardinality of positive and negative evidence in NAL and after adaptation and how they affect resulting truth-values.



(a)

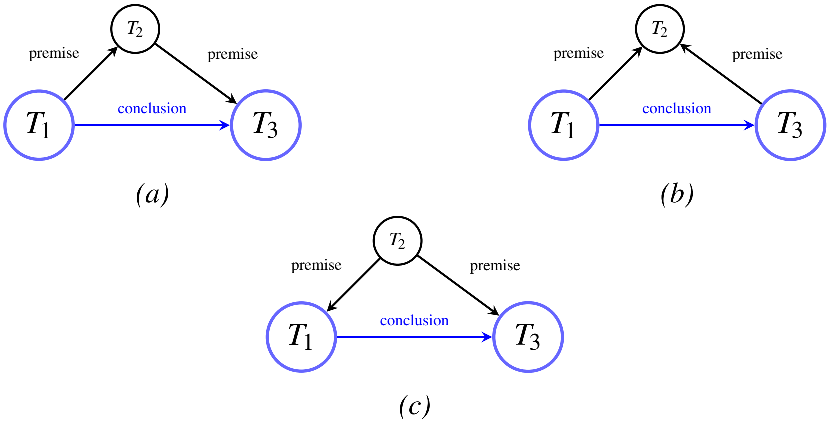
(b)

Figure 3.11: Influence of positive and negative evidence on resulting truth-values a) in NAL, and b) after adaptation.

3.3.3 Inference based on truth-values

The main rules of inference in NAL are extended syllogisms where two statements that share the same term can be used as premises for concluding a relation between their remaining two terms (P. Wang, 1994). Syllogisms can be used to expand the knowledge base, yet this thesis suggests an application to gather similarity information from other points of the distributed system. The forward inference is one of the types of reasoning rules in NAL that produce new statements in judgments. There are three combinations of premises and conclusions when applying forward inference: deduction, abduction, and induction.

Fig. 3.12 shows a graph presentation of inference rules. In Fig. 3.12a, it can be observed that a shared term in one statement is a subject, but in another predicate (inference by deduction). In contrast, a term shared by statements is predicate and subject in abduction and induction, respectively (Fig. 3.12b and Fig. 3.12c). The difference between the three is purely syntactic (P. Wang, 2000).



(a)

(b)

(c)

Figure 3.12: Rules of inference in NAL: a) deduction, b) abduction and c) induction.

If, for example, it is said that the person described by the HOG descriptor from which the extracted features in $\vec{V}_1$ is similar to the person $\vec{V}_2$ to the extent that they belong to the same person, and that the person $\vec{V}_2$ similar person $\vec{V}_3$ to the extent that they belong to the same person, then it can be concluded that person $\vec{V}_1$ is similar to person $\vec{V}_3$ to the extent that it can be claimed that they belong to the same person (show on Fig. 3.13). Semantically, the conclusion was reached by deduction, but the similarity in the structure of statements could go in the directions that represent induction and abduction. For the purposes of the reasoning system, the deduction will be applied in this thesis.

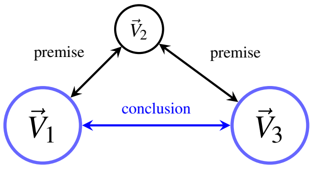


Figure 3.13: Applied inference method after NAL adaptation.

In NARS where each statement has the form $T_1 \rightarrow T_2\langle f, c\rangle$, based on truth-values of premises, the frequency and confidence of conclusion in the deduction process will be calculated using Eq. 3.3.25 and Eq. 3.3.26 (P. Wang, 2000).

$$f = \frac{f_1 f_2}{f_1 + f_2 - f_1 f_2} \tag{3.3.25}$$

$$c = (f_1 + f_2 - f_1 f_2)c1c2 \tag{3.3.26}$$

The calculation of frequency f and confidence c of truth-value in forward inference applied in the reasoning system discussed in this thesis is also performed using 3.3.25 and 3.3.26.

In addition to the rules of forward inference, some rules do not produce new statements. However, the content of the conclusion is equal to the content of the premise, and those rules are named local.

In NAL, there are two rules: the revision rule, which merges its premises into its conclusion, and the choice rule, which chooses one of the premises as its conclusion (P. Wang, 2013). Local rules are related to inconsistent beliefs, i.e., same judgments with different truth-values. Inconsistent beliefs are the result of the openness of the system to new experiences that may differ from existing ones, as well as the adoption of the same courts based on different evidential bases. This thesis uses revision when truth-values between two feature vectors are obtained from other points of the distributed system. Eq. 3.3.27 and Eq. 3.3.28 are used to calculate the resulting frequency values f and confidence c based on truth-values from different points of the system.

$$f = \frac{f_1 c_1 (1 - c_2) + f_2 c_2 (1 - c_1)}{c_1 (1 - c_2) + c_2 (1 - c_1)} \tag{3.3.27}$$

$$c = \frac{c_1 (1 - c_2) + c_2 (1 - c_1)}{c_1 (1 - c_2) + c_2 (1 - c_1) + (1 - c_1)(1 - c_2)}. \tag{3.3.28}$$

How different frequency and confidence input value combinations affect the revision result is shown in Tab. 3.2. What can be concluded is that the resulting frequency value is closer to that obtained with greater confidence. For example, on Tab. 3.2 could be seen two marked fields in the revision results, one of which is the result of opposing input values, a high value of frequency with low confidence, and vice versa. The value obtained by applying the Eq. 3.3.27 and Eq. 3.3.28 gives a result identical to the input with greater confidence. In the second marked field, the result is based on the same input values of frequency and confidence, which gives revision results with a frequency equal to the input values, but confidence increases.

Table 3.2: Examples of revision based on a combination of different values of frequency and confidence.

$\langle f,c\rangle$	$\langle 0.2,0.8\rangle$	$\langle 0.4,0.6\rangle$	$\langle 0.5,0.5\rangle$	$\langle 0.6,0.4\rangle$	$\langle 0.8,0.2\rangle$
$\langle 0.2,0.8\rangle$	$\langle 0.2,0.9\rangle$	$\langle 0.3,0.8\rangle$	$\langle 0.3,0.8\rangle$	$\langle 0.3,0.8\rangle$	$\langle 0.2,0.8\rangle$
$\langle 0.4,0.6\rangle$	$\langle 0.3,0.8\rangle$	$\langle 0.4,0.8\rangle$	$\langle 0.4,0.7\rangle$	$\langle 0.5,0.7\rangle$	$\langle 0.5,0.6\rangle$
$\langle 0.5,0.5\rangle$	$\langle 0.3,0.8\rangle$	$\langle 0.4,0.7\rangle$	$\langle 0.5,0.7\rangle$	$\langle 0.5,0.6\rangle$	$\langle 0.6,0.6\rangle$
$\langle 0.6,0.4\rangle$	$\langle 0.3,0.8\rangle$	$\langle 0.5,0.7\rangle$	$\langle 0.5,0.6\rangle$	$\langle 0.6,0.6\rangle$	$\langle 0.7,0.5\rangle$
$\langle 0.8,0.2\rangle$	$\langle 0.2,0.8\rangle$	$\langle 0.5,0.6\rangle$	$\langle 0.6,0.6\rangle$	$\langle 0.7,0.5\rangle$	$\langle 0.8,0.3\rangle$

3.4 Evaluation

In evaluating the results of experiments that will be performed, two different methods of pattern identification used in machine learning will be considered: classification and clustering. Due to the differences between them, different evaluation methods are required.

3.4.1 Evaluation methods for classification

Based on the confusion matrix elements, accuracy (Acc) is determined as the ratio of accurate predictions and the total number of predictions, i.e., matching (C.-C. Wu et al., 2019).

$$Acc = \frac{TP+TN}{TP+FP+TN+FN} \tag{3.4.1}$$

In unbalanced sets where the number of samples of one class is significantly higher than in another, accuracy can no longer be considered a sufficiently reliable measure. Accuracy too optimistically estimates classifier performance for the majority class (Chicco & Jurman, 2020). On a large scale of persons who need to be identified during the criminal investigation in actual conditions, a disproportionate number of different and the same persons are expected to favor different ones. Therefore, other classification quality measures have been applied in this thesis: $F1$ score and Matthews correlation coefficient (MCC). A standard measure used in unbalanced sets is $F1$, representing the harmonic mean of precision and recall and has the following form (Chicco & Jurman, 2020):

$$F_1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3.4.2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3.4.3}$$

$$Recall = \frac{TP}{TP+FN} \tag{3.4.4}$$

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP+FP+FN} \tag{3.4.5}$$

The minimum value of the measure, i.e., $F1$ = 0, is obtained when the number of correctly positively classified samples is equal to zero, $TP = 0$. In contrast, the perfect classification, $F1$ = 1, is considered when the number of incorrect negative and positive classifications is equal to zero, FN = FP = 0. What is immediately noticeable is that the $F1$ measure does not consider accurate negatively classified predictions TN.

Additional verification of the classification quality is performed by the MCC. MCC considers all confusion matrix elements and is applied in various research areas using binary classifiers, such as those presented in (Y. Wu et al., 2017; Y. Wang, Fang, & Hong, 2019). The MCC is described by definition 3.4.6.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN)}} \tag{3.4.6}$$

Suppose MCC = 0, classification result is equal to a random selection. A positive value indicates a correct classification (a value of +1 coefficient represents a perfect classification). In contrast, negative values indicate a worse classification than a random selection, where the value of a coefficient of -1 represents a wide discrepancy between prediction and concrete classes. *MCC* is the only measure of binary classification that generates a high score when a particular model performs a successful classification regardless of which class is dominant (Chicco & Jurman, 2020).

3.4.2 Evaluation methods for clustering

BCubed F-measure (B^3F) and Normalized Mutual Information (NMI) methods are mostly used in related work to confirm effectiveness, so they will be used in this thesis for comparison with other state-of-art solutions.

B^3F is the harmonic mean of BCubed precision ($B^3Precision$), a fraction of points of the same class in the same cluster, and BCubed recall ($B^3Recall$), which is a fraction of points of the same class that are assigned to the same cluster. B^3 is suitable in situations where the number of clusters is unknown (Lin, Chen, Castillo, & Chellappa, 2018). Taking into account both $B^3Precision$ and $B^3Recall$ makes the B^3F measure very practical (Z. Wang et al., 2019), and imposes it as a choice for the evaluation of the proposed algorithms in this thesis.

The relation between the two vectors $\vec{V_i}$ and $\vec{V_j}$ extracted from face image descriptors is determined by comparing whether they are in the same cluster C and whether they are labeled L in the same way.

$$Correct(\vec{V_i},\vec{V_j}) = \begin{cases} 1 & \text{if } C(\vec{V_i}){=}C(\vec{V_j}) \leftrightarrow L(\vec{V_i}){=}L(\vec{V_j}) \\ 0 & \text{otherwise} \end{cases} \tag{3.4.7}$$

Pairwise correctness is further used to determine the values of $B^3Precision$ and $B^3Recall$, which are then used in the recognizable F1-score calculation formula.

$$B^3Precision = Avg_{\vec{V_i}}[Avg_{\vec{V_j}:C(\vec{V_j})=C(\vec{V_i})}[Correct(\vec{V_i},\vec{V_j})]] \tag{3.4.8}$$

$$B^3Recall = Avg_{\vec{V_i}}[Avg_{\vec{V_j}:L(\vec{V_j})=L(\vec{V_i})}[Correct(\vec{V_i},\vec{V_j})]] \tag{3.4.9}$$

$$B^3F = \frac{2 \cdot B^3Precision \cdot B^3Recall}{B^3Precision + B^3Recall} \tag{3.4.10}$$

NMI measures the normalized similarity between the ground truth labels and the labels determined by the clustering algorithm. Although it is most suitable when the number of clusters is known and fails to penalize over-clustering, it is often used compared to other clustering methods (Lin, Chen, Castillo,

& Chellappa, 2018; Qi et al., 2021).

$$NMI(L(\vec{V}),C(\vec{V})) = \frac{I(L(\vec{V}),C(\vec{V}))}{\sqrt{H(L(\vec{V}))H(C(\vec{V}))}} \tag{3.4.11}$$

NMI is obtained by dividing mutual information $I(L(\vec{V}),C(\vec{V}))$ with entropy ground truth cluster set $H(L(\vec{V}))$ and the predicted cluster set $H(C(\vec{V}))$.

Chapter 4

Gathering digital evidence for presentation and identity resolution using open-source tools

Identity resolution can be defined as a form of classification where two or more profiles of a person, often from different databases, are compared based on the similarity of their features (Edwards, Wattam, Rayson, & Rashid, 2016). The definition emphasizes the term "profile of a person," pointing to one of the most common research subjects: identity resolution capability for social networking profiles, where the goal is to identify multiple profiles that relate to the same individual. The task of criminal investigators is to establish the identity and all illegal activities committed by the same person while using false and true identities. Essential for this process is that the sources themselves go beyond social networks and even the availability of profiles with specific attributes that determine an individual. Therefore, in general, identity resolution is also a process of determining whether a single identity is semantically the same when otherwise described, or it is a process of collecting, identifying, and comparing attributes of a person to obtain a consistent identity (Phillips et al., 2020).

When developing an identity resolution system, and even when it comes to a system that assists in that process, as is the case here, matching identity needs to be performed, which includes two critical phases: feature extraction and model construction (Srivastava & Roychoudhury, 2020). The models are based on different machine learning techniques. However, the model's success depends on selecting appropriate features and their extraction. Traditional resolution techniques rely on identification numbers, personal names, places, dates of birth, addresses, and other attributes mostly available in record management systems (J. Li & Wang, 2015). On the other hand, most current studies on identity resolution are focused mainly on information from social media profiles, the most desirable of which are currently unavailable due to privacy concerns (Srivastava & Roychoudhury, 2020). Still, during a police investigation, biometrics is essential in determining identity and identity resolution. When it comes to digital evidence, regardless of its sources, one of the available biometric methods is face recognition. Therefore, for this thesis, images of individuals are the main feature for resolving a person's identity and connecting and presenting digital evidence relating to the same person.

Images of faces, among other information, can be obtained in the everyday work of investigators and police in general, for example, by collecting photo and video material from public and private Closed-Circuit Television (CCTV) systems or photographing suspects during their arrest, also by physical and electronic surveillance. OSINT is another source of digital evidence, i.e., criminal intelligence, which, among other information, may contain images of a person. It originates from security agencies and LEA and refers to intelligence derived from publicly available information sources such as global media, web blogs, government reports, satellite pictures, academic papers, social media, and other forms of crowdsourcing (Quick & Choo, 2018).

However, due to the variety of materials, forensic reports obtained from analyzing mobile phones, personal computers, and similar devices stand out from all the above sources of digital evidence and police intelligence. In addition, technological advances and changes in consumer behavior have led to an increase in the "volume, variety, velocity, and veracity" of data analyzed by digital forensics (Quick & Choo, 2018).

4.1 Collecting digital evidence from forensic reports

Digital forensics has progressed in the last few decades of its development from "obscure tradecraft to an important part of many investigations" (Garfinkel, 2010), and today digital forensic tools are in everyday use. Forensic investigation is a complicated and time-consuming process that requires researchers' skills from multiple fields, such as information security, penetration testing, reverse engineering, programming, and behavior profiling, to reach the ultimate goal, which is to create a clear picture of the crime committed using a computer (Koen, 2009).

The two main areas in which digital forensics is applied most intensively are law enforcement agencies and corporate security, each having a different approach (Bassett, Bass, & O'Brien, 2006). In brief, corporate security focuses on adequate response and damage assessment (Cohen, Bilby, & Caronni, 2011), while law enforcement agencies seek to find traces of the crime, computer programs, or data that can indicate the link between the offense, the perpetrator, or victim in a large amount of data. Concerning both these areas, digital forensics can be defined as a process in which scientific principles are applied to analyze electronically stored information (artifacts) on one or more digital devices in order to determine the sequence of events that led to a particular incident, as well as the understanding and reconstructing events that must have transpired in generating the said artifacts (Raghavan, 2013).

Modern digital forensics faces numerous obstacles, even without anti-forensic techniques. The necessary picture of what happened in a particular case is often challenging due to inconsistencies and a lack of incriminating evidence (Koen, 2009). The forensic investigation process consists of two phases, Acquisition Phase and Analysis Phase (Carrier, 2002), each with its challenges. The data acquisition process is hampered by a large number of various devices from which it is necessary to collect digital records. A particular frustration for the forensic community has been generated by mobile phones since the beginning of their proliferation. As mobile phones develop faster than other technologies, a significant problem is developing the tool that would be considered an industry standard (Yates, 2010). The "internet of things" paradigm and development of even more devices containing forensic artifacts will further complicate the acquisition process. Also, the evidence needs to be collected from several technological zones, the devices themselves, the Network zone, and the Cloud zone (Miljković, Čabarkapa, Prokin, & Budimir, 2018). Another issue is that the protection of modern devices and data storage encryption is already a constant problem for forensic investigators. In order to overcome all issues, forensic investigators must continuously invest in new data collection tools and techniques.

The data analysis phase consists of searching for evidence among the collected data. The aim is to confirm the hypothesis of some event, find arguments to refute the hypothesis, or at least show that a particular system has been altered to conceal evidence (Carrier, 2002).

After phases of data analysis, the work of investigators results in the creation of forensic reports, which is of central interest in this thesis as input for system assisting criminal investigators in identity resolution and grouping of evidence.

4.1.1 Using open-source tools to collect evidence

Forensic reports are usually exported in HyperText Markup Language (HTML) or Portable Document Format (PDF) because those two types of files can be easily opened on most devices (personal computers or smartphones and similar devices). Still, depending on the case, i.e., the type of crime, files of many different formats can be found within exported content from the case attached to the report.

What is needed, for example, to extract face features from images and videos, which is part of the report, is to convert them into a suitable format for the face recognition process. Also, converting the digital evidence into a searchable text form or into forms that an internet browser can show is necessary to gather and present all digital evidence to investigators. Forms of data that could be used by web technology are images in Joint Photographic Expert Group (JPEG) and Portable Network Graphics (PNG) formats, plain text but also structured like JavaScript Object Notation (JSON) and Extensible Markup Language (XML), or PDF which is a type of document that internet browsers can show. Open-source applications are imposed as an easily accessible, usually free of charge, simple for usage over Command-Line Interface (CLI), and can be automated by scripts. An additional advantage of using this type of software is saving time on developing complex functionalities required for conversions between many different formats.

The free, open-source software development began in the computer science departments of Stanford University, Berkeley, Carnegie Mellon, and Massachusetts Institute of Technology in the 1960s and 1970s. In the mid-1980s, the first open-source licenses appeared. Essential features of open-source software are that it is developed in a public and collaborative manner, that the program code is available to users who can change or distribute it, that it can often be considered a viable alternative to proprietary software, and the community around open-source has become an innovative global movement (Menéndez-Caravaca, Bueno, & Gallego, 2021). Software licenses are defined as freedoms such as being used for any purpose, study of how the program works, freedom of redistribution, and distribution of the modified version of the source (Gallego, Bueno, Racero, & Noyes, 2015). Changing licenses is also part of freedom and choice. For example, the MongoDB NoSQL database used in this thesis is called open-source software because of its version dated prior to the end of 2018. This version is enough for research and created model, especially for hardware with limited resources. In 2018, they imposed some restrictions and used a new Server Side Public License. Although it is an open-source look-alike, it has not been approved by, for example, Open Source Initiative (*The SSPL is Not an Open Source License*, 2021).

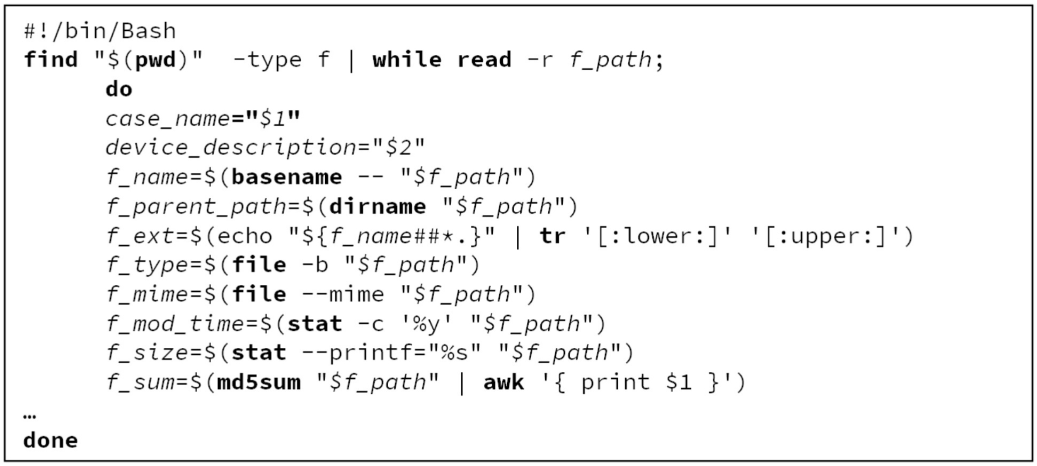
The motivation for creating open-source software is ethical because it is about knowledge and must be disseminated without hindrance, but also there is a practical side related to technical and economic advantages that open-source can generate (Menéndez-Caravaca et al., 2021). Some open-source solutions represent important business elements such as Linux operating systems on servers, Apache and Nginx web servers, MySQL/MariaDB and PostgreSQL databases, internet browsers such as Firefox Mozilla, and Chromium, Gimp image editor, and vector graphics editor Inkscape. Wider use of open-source software by non-technical users is limited due to the technical orientation of many open-source solutions (during development, priority is given to functionality opposite to the user experience). Some research has shown that training is vital for accepting open-source solutions (Gallego et al., 2015). Still, concerning the problem of open-source tool usage, the functional model built in this thesis will use them only in the system's background.

In addition to the advantages of free use of software, open-source software can also bring savings on hardware because the system requirements of these solutions are much lower than proprietary designs based on the technical capabilities of the latest hardware (Menéndez-Caravaca et al., 2021). Linking the functionality of particular tools was done using a Bash script that requires knowledge of programming basics (variables without too much consideration about the type, conditional statements, and control flow). The application of different tools and the Bash scripts allows building the system according to a wide variety of different needs or the needs of a particular case.

4.1.2 Collecting information about files

The data that needs to be collected from the forensic report refers to the system information about the files found in the analysis process: storage location in the report, name, size, and type. There are also metadata that represents file parameters (for example, a mark and the model of the camera which took the photo, if GPS is on, the spatial references can be found). Metadata is part of the file content, not the file system.

Many open-source applications are usually built-in as part of operating systems distributions based on the Linux kernel that can provide the necessary information. For example, with the **stat** command, we can learn the size and the last file modification time, the **baseman** can extract the file name from the absolute path, a **file** can obtain file type information. By installing packages for forensic purposes, the commands of these packages can be used to get all the available details mainly found in file systems such as NTFS (like a file creation time) and other non-Linux environments. After mounting the media that contains the digital forensic report, it is necessary to go through all the folders and find all the files. The **find** command can be used for this purpose, as it returns a list of absolute paths of all files. Command **read** gets each absolute file path which then can be used as the input parameter for other tools and commands. An example of using different commands for extracting file information data is shown in Fig. 4.1.



```
#!/bin/Bash
find "$(pwd)"  -type f | while read -r f_path;
      do
      case_name="$1"
      device_description="$2"
      f_name=$(basename -- "$f_path")
      f_parent_path=$(dirname "$f_path")
      f_ext=$(echo "${f_name##*.}" | tr '[:lower:]' '[:upper:]')
      f_type=$(file -b "$f_path")
      f_mime=$(file --mime "$f_path")
      f_mod_time=$(stat -c '%y' "$f_path")
      f_size=$(stat --printf="%s" "$f_path")
      f_sum=$(md5sum "$f_path" | awk '{ print $1 }')
…
done
```

Figure 4.1: Collecting file information from the file system.

In addition, to file information that various tools already collected from a file system, the following tool named **exiftool** extracts a set of metadata depending on the file type as shown in Fig. 4.2.

```
#!/bin/Bash
find "$(pwd)"  -type f | while read -r f_path;
      do
      …
      temp_dir="/home/eve/forensics/temp_dir/"
      case $f_ext in
            JPEG | JPG | PNG | PSD)
                  fmd_make=$(exiftool -s -s -s -make "$f_path")
                  fmd_model=$(exiftool -s -s -s -model "$f_path")
                  fmd_software=$(exiftool -s -s -s -software "$f_path")
                  fmd_lat=$(exiftool -s -s -s -gpslongitude "$f_path")
                  fmd_long=$(exiftool -s -s -s -gpslatitude "$f_path")
                  fmd_w=$(exiftool -s -s -s -imagewidth "$f_path")
                  fmd_h=$(exiftool -s -s -s -imageheight "$f_path")
                  fmd_mdate=$(exiftool -s -s -s -filemodifydate "$f_path")
                  f_md="metadata: [{make:'$fmd_make',
model:'$fmd_model',software:'$fmd_software',lat:'$fmd_lat',long:'$fmd_long',
w:'$fmd_w',h:'$fmd_h',mdate:'$fmd_mdate'}],"
                  ;;
            …
      esac
…
done
```

Figure 4.2: Collecting metadata with exiftool command.

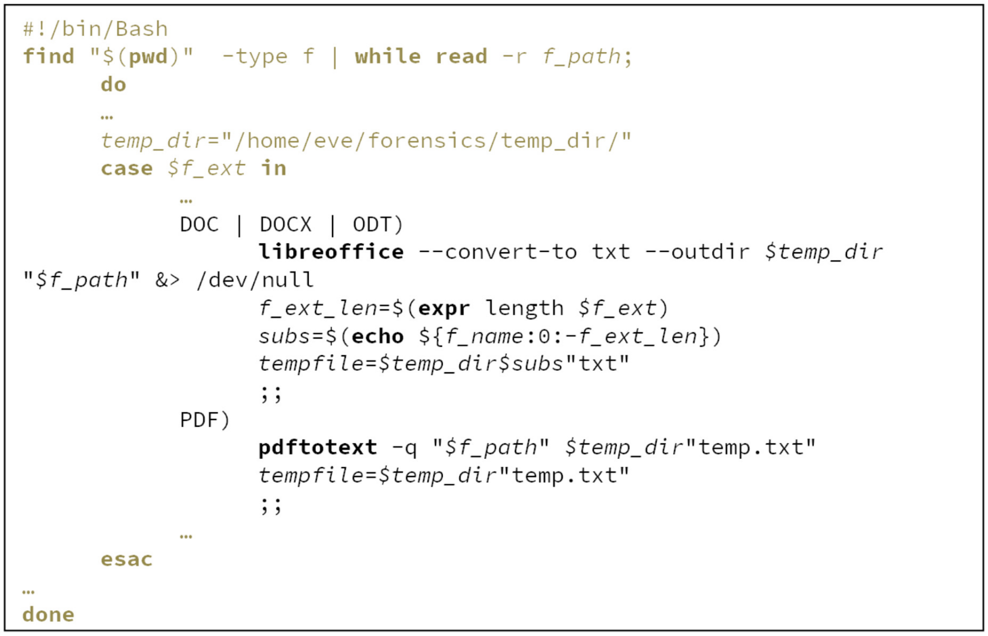
Each tool tries to display a more transparent list of results, which is also true for **exiftool**. To provide only the necessary data (image size, processing software, etc.), the command **exiftool** executing repeatedly several times with the different options (repeating the option -s tree times separates the value from a result, for example, image height).

4.1.3 Collecting textual content

The most important information for the forensic investigation is within the file's contents. Unlike the tools that collect data from the file system and metadata, one open-source tool is not enough to extract as much information from the file's contents. For each type of file or set of files with a similar purpose different one is needed. For example, the LibreOffice open-source package offers the ability to read the contents of the file it creates, as well as several other proprietaries file types like Microsoft Office files (Word, Excel, PowerPoint).

The way to use the LibreOffice application is usually over a Graphical User Interface (GUI). Important for automatization is that LibreOffice also could be executed from the Command-Line Interface (CLI). During execution within CLI necessary for LibreOffice application, it is an absolute path to the file and directory where the output will be saved, that is, the file with the textual content. For example, the output directory in the script shown in Fig. 4.3 is for temporarily storing content until its further processing.

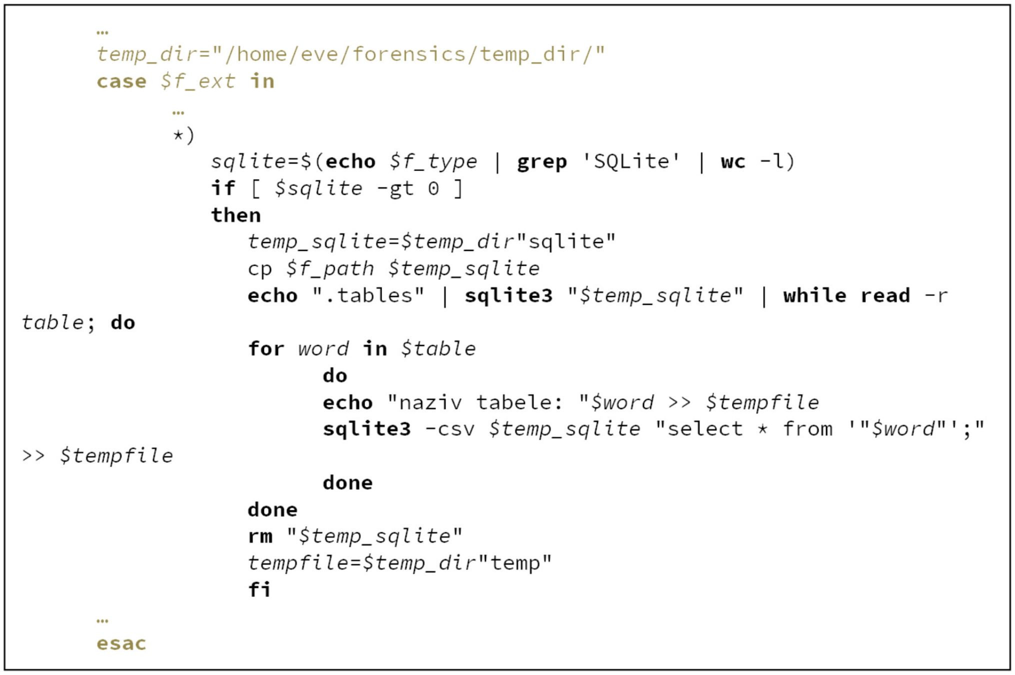
The same principle applies to the **pdftotext** command, which converts PDF file to a text file based on the absolute path. What is different is that for the **pdftotext** tool, the user defines the output file name of a temporary text file with the content of the PDF file. In the case of the **libreoffice** script need to be extended for two more lines to get the name of a temporary file. The name of the temporary files is required later during script execution to read the text content and temporary file deletion.



```
#!/bin/Bash
find "$(pwd)"  -type f | while read -r f_path;
      do
      …
      temp_dir="/home/eve/forensics/temp_dir/"
      case $f_ext in
            …
            DOC | DOCX | ODT)
                  libreoffice --convert-to txt --outdir $temp_dir
"$f_path" &> /dev/null
                  f_ext_len=$(expr length $f_ext)
                  subs=$(echo ${f_name:0:-f_ext_len})
                  tempfile=$temp_dir$subs"txt"
                  ;;
            PDF)
                  pdftotext -q "$f_path" $temp_dir"temp.txt"
                  tempfile=$temp_dir"temp.txt"
                  ;;
            …
      esac
…
done
```

Figure 4.3: Conversion of Office and PDF file contents into a text.

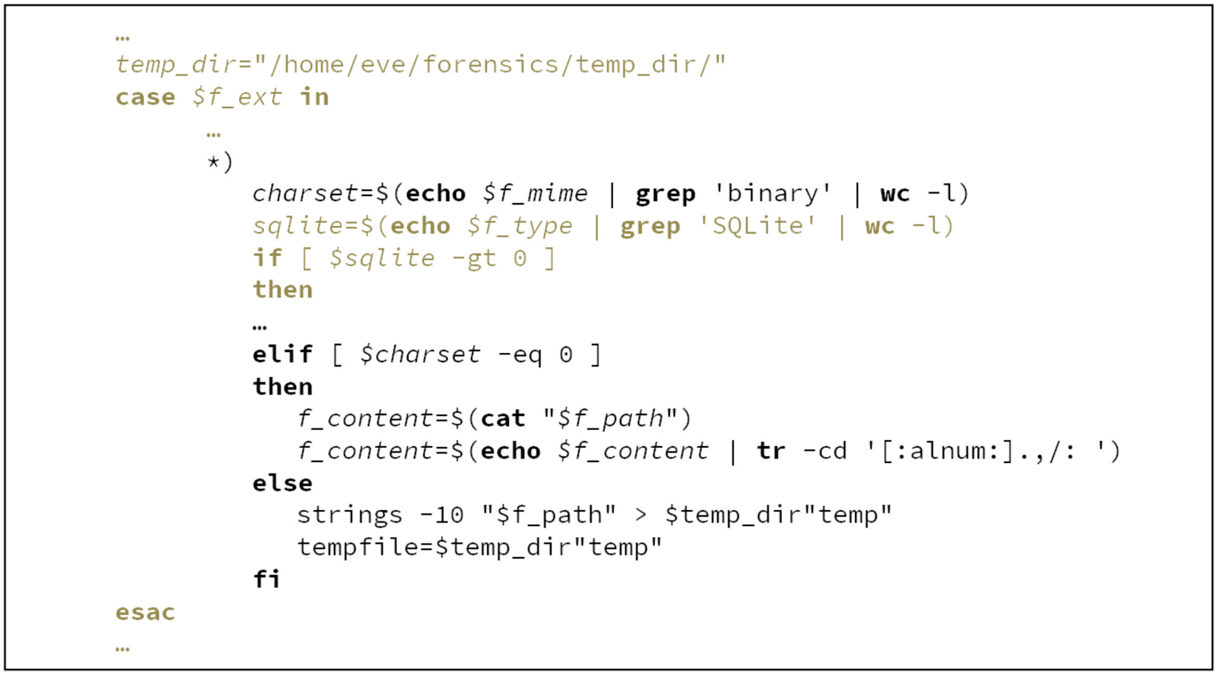
Microsoft Windows operating system, for example, uses extensions of file names to determine which type of file it is. Still, forensic investigators cannot rely on extensions. Users can deliberately replace them, making it harder to find specific types of files or applications that could exclude extensions entirely (for example, Firefox Mozilla puts the SQLite file name extension, while Google Chrome does not). Fig. 4.4 shows how to identify file type by using the **file** command. Applications often use SQLite as a database manager, so Fig. 4.4 shows how the **sqlite3** extracts tables and their contents stored in the SQLite file found in the case. Unlike most other files, the **sqlite3** application cannot access the SQLite file from the write-protected medium. Therefore, the SQLite file is copied to the temporary directory to access it, read content from all tables, and save it as a text for further processing.



```
      …
      temp_dir="/home/eve/forensics/temp_dir/"
      case $f_ext in
            …
            *)
               sqlite=$(echo $f_type | grep 'SQLite' | wc -l)
               if [ $sqlite -gt 0 ]
               then
                  temp_sqlite=$temp_dir"sqlite"
                  cp $f_path $temp_sqlite
                  echo ".tables" | sqlite3 "$temp_sqlite" | while read -r
table; do
                  for word in $table
                        do
                        echo "naziv tabele: "$word >> $tempfile
                        sqlite3 -csv $temp_sqlite "select * from '"$word"';"
>> $tempfile
                        done
                  done
                  rm "$temp_sqlite"
                  tempfile=$temp_dir"temp"
                  fi
      …
      esac
```

Figure 4.4: Extraction of SQLite tables content into a text file.

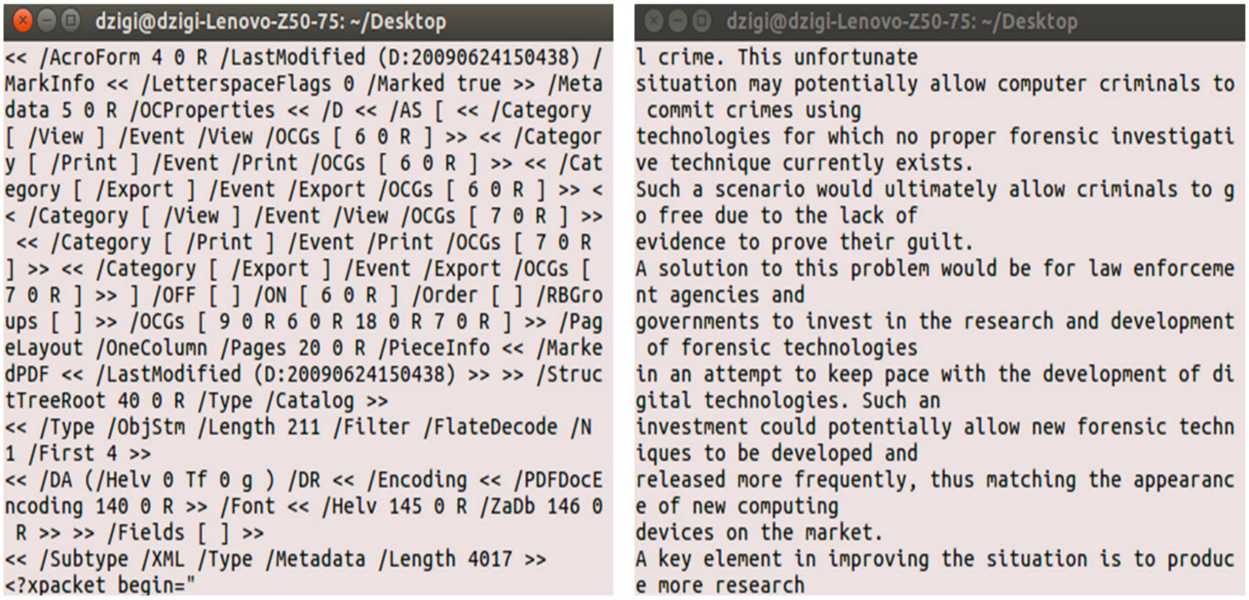
Using the –mime option of a command file, it is possible to determine whether it is a binary file. If this is not the case, the file's contents are textual and can be read instantly. In the case of a binary file, a group of printable characters that could be found in content of the file can be extracted with **strings** command as shown in Fig. 4.5. The goal is to use as many open-source applications as possible that can parse the binary structure instead of the **strings** command.



```
…
temp_dir="/home/eve/forensics/temp_dir/"
case $f_ext in
      …
      *)
         charset=$(echo $f_mime | grep 'binary' | wc -l)
         sqlite=$(echo $f_type | grep 'SQLite' | wc -l)
         if [ $sqlite -gt 0 ]
         then
         …
         elif [ $charset -eq 0 ]
         then
            f_content=$(cat "$f_path")
            f_content=$(echo $f_content | tr -cd '[:alnum:].,/: ')
         else
            strings -10 "$f_path" > $temp_dir"temp"
            tempfile=$temp_dir"temp"
         fi
esac
…
```

Figure 4.5: Reading text files and printable characters from binary files.

Differences in the results of using strings command and the application that can read the contents of a particular binary format are shown in Fig. 4.6. The text that represents the contents of the file on the right side of Fig. 4.6 can be found in a text composed of printable characters on the left side of the figure, meaning that **strings** command does not provide complete information about the content.



```
<< /AcroForm 4 0 R /LastModified (D:20090624150438) /
MarkInfo << /LetterspaceFlags 0 /Marked true >> /Meta
data 5 0 R /OCProperties << /D << /AS [ << /Category
[ /View ] /Event /View /OCGs [ 6 0 R ] >> << /Categor
y [ /Print ] /Event /Print /OCGs [ 6 0 R ] >> << /Cat
egory [ /Export ] /Event /Export /OCGs [ 6 0 R ] >> <
< /Category [ /View ] /Event /View /OCGs [ 7 0 R ] >>
 << /Category [ /Print ] /Event /Print /OCGs [ 7 0 R
] >> << /Category [ /Export ] /Event /Export /OCGs [
7 0 R ] >> ] /OFF [ ] /ON [ 6 0 R ] /Order [ ] /RBGro
ups [ ] >> /OCGs [ 9 0 R 6 0 R 18 0 R 7 0 R ] >> /Pag
eLayout /OneColumn /Pages 20 0 R /PieceInfo << /Marke
dPDF << /LastModified (D:20090624150438) >> >> /Struc
tTreeRoot 40 0 R /Type /Catalog >>
<< /Type /ObjStm /Length 211 /Filter /FlateDecode /N
1 /First 4 >>
<< /DA (/Helv 0 Tf 0 g ) /DR << /Encoding << /PDFDocE
ncoding 140 0 R >> /Font << /Helv 145 0 R /ZaDb 146 0
 R >> >> /Fields [ ] >>
<< /Subtype /XML /Type /Metadata /Length 4017 >>
<?xpacket begin="
```

```
l crime. This unfortunate
situation may potentially allow computer criminals to
 commit crimes using
technologies for which no proper forensic investigati
ve technique currently exists.
Such a scenario would ultimately allow criminals to g
o free due to the lack of
evidence to prove their guilt.
A solution to this problem would be for law enforceme
nt agencies and
governments to invest in the research and development
 of forensic technologies
in an attempt to keep pace with the development of di
gital technologies. Such an
investment could potentially allow new forensic techn
iques to be developed and
released more frequently, thus matching the appearanc
e of new computing
devices on the market.
A key element in improving the situation is to produc
e more research
```

Figure 4.6: Comparison of the results of the strings command and pdftotext command working on the same file.

4.1.4 Conversion of various formats containing face images

Open-source tools can convert files into formats that could be opened by widely available applications, like internet browsers. Otherwise, only to see the content will be necessary to use original proprietary software, such as Adobe Photoshop or Corel Draw.

Inkscape is a professional open-source vector graphics editor. Its natural working environment is a GUI. Still, with the **inkscape** command, it is possible to convert various formats used in graphic design within a script, as shown in Fig. 4.7 into a PNG format readable with any image preview application or internet browser.

```
#!/bin/Bash
find "$(pwd)"  -type f | while read -r f_path;
      do
      …
      temp_dir="/home/eve/forensics/temp_dir/"
      case $f_ext in
            …
            CDR | SVG | EPS | AI)
                  img_rep_path="$img_repository""$case_name"-$f_sum.png
                  inkscape "$f_path" --export-png="$img_rep_path"
                  img_rep_path=",'img_rep_path':'"$img_rep_path"'"
                  ;;
      …
      esac
…
done
```

Figure 4.7: Example of conversion vector graphics files into a raster-graphic file format.

Fig. 4.7 shows that along the script line used for the conversion that starts with the **inkscape** command, there are two more lines before and after. The first line variable *img_rep_path* is assigned a text value that contains the path to the image repository that the potential system will use, as well as an image name consisting of the case name and the hash value of the image. That information **inkscape** command use to store the output. Since this is a value that is temporarily needed, the same variable is used after the conversion to create structured text that will be recorded in the database.

Besides being compatible with various applications which show the content of images, PNG format can also be used for face recognition. Therefore, it is important that the format intended for editing vector graphics, in which personal documents are often falsified, be converted into a format used for face recognition. In addition to vector graphics editing applications, raster graphics processing applications are also widely used, ahead of all Adobe Photoshop. The command that convert the Photoshop Document (PSD) format of the application to PNG format is **mogrify** with the appropriate options. The script line **mogrify** -format png -path <*path to PNG file in the repository*> <*PSD file*> gives a set of PNG files whose number corresponds to the number layers created in Photoshop.The ability to recognize the face of each layer of Photoshop files allows the face, even if something obscuring it is added as a result of editing, to be successfully detected.

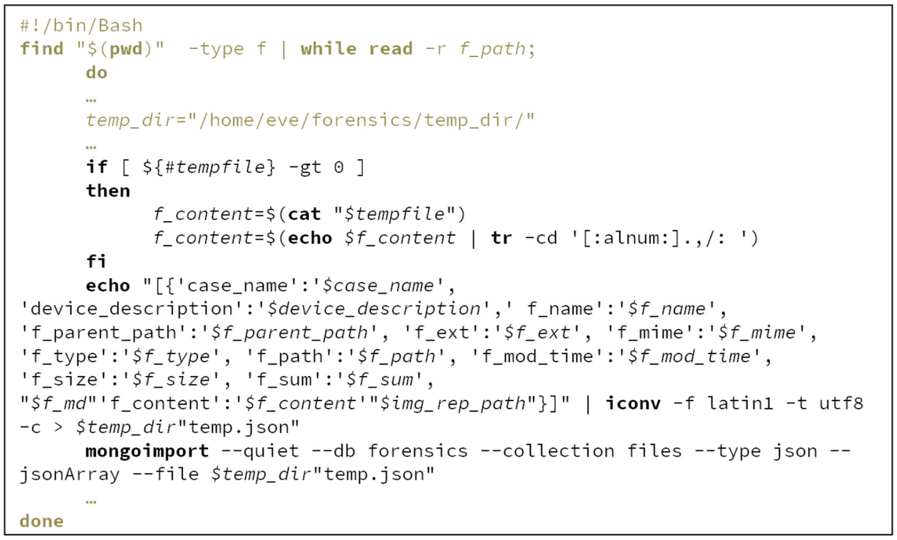
One of the formats in which forensic reports are generated, and therefore there is a possibility that there are a large number of images, including face images, is PDF. As with Photoshop files, PDF files can be converted to PNG format, with each page of the document represented by a separate PNG file. The command used for this purpose is **pdftoppm** as follows: **pdftoppm** -png <*PDF file*> <*name prefix PNG file*>.

When it comes to a command that can convert a large number of different file formats of images to, for example, the PNG format, the **convert** command, which is part of the same software package as **mogrify**, is very useful. The characteristic of this command is that, except conversion, it also enables the processing of a large number of images in ways defined by numerous different options and parameter values.

Extracting images from videos can be done using the **ffmpeg** command: **ffmpeg** -i <*video file*> -r 1 <*image name*> % 3d.png. The options and parameters applied with the **ffmpeg** command allow one frame per second to be extracted and three decimal places to follow the same prefix in the file name. Defining the designation of file names with three, but preferably more, decimals enables the preservation of the time sequence, i.e., chronological processing.

4.1.5 Data import, analysis and reporting

Fig. 4.8 shows how the script can import all the collected information into the MongoDB NoSQL open-source database. From temporary files where the script stored the textual content of a subject file, the script reads data, and then along with other information collected from the file system and metadata, they are entered as a document within a database collection. The **mongoimport** command imports data that the script previously saved in a file into a database as a JSON array.



```
#!/bin/Bash
find "$(pwd)"  -type f | while read -r f_path;
      do
      …
      temp_dir="/home/eve/forensics/temp_dir/"
      …
      if [ ${#tempfile} -gt 0 ]
      then
            f_content=$(cat "$tempfile")
            f_content=$(echo $f_content | tr -cd '[:alnum:].,/: ')
      fi
      echo "[{'case_name':'$case_name',
'device_description':'$device_description',' f_name':'$f_name',
'f_parent_path':'$f_parent_path', 'f_ext':'$f_ext', 'f_mime':'$f_mime',
'f_type':'$f_type', 'f_path':'$f_path', 'f_mod_time':'$f_mod_time',
'f_size':'$f_size', 'f_sum':'$f_sum',
"$f_md"'f_content':'$f_content'"$img_rep_path"}]" | iconv -f latin1 -t utf8
-c > $temp_dir"temp.json"
      mongoimport --quiet --db forensics --collection files --type json --
jsonArray --file $temp_dir"temp.json"
      …
done
```

Figure 4.8: Importing collected data into a NoSQL database.

Bash scripts and various commands can search over imported data and create reports in a context that the forensic investigator needs or chooses. First, the script obtains database search results in the JSON format. After that, command **jq** parses data into an array. Finally, the report function saves the data in a text file with name extension HTML that can be opened later in the internet browser. One of the ways to search the database and create a report is shown in Fig. 4.9.

```
report() { while read line; do echo $line>> report.html; done; }
echo "db.files.aggregate([
      {\$project:{_id:0,f_path:1,f_ext:1,img_rep_path:1,f_sum:1,f_content:1,
length:{\$strLenCP:'\$f_content'}}},
      {\$match:{f_ext:{\$in:['PDF','AI']}}}]).forEach(printjson);" >
$temp_dir"q.js"
mongo --quiet forensics '/home/eve/forensics/temp_dir/q.js' | jq -r
'(.f_path)+"\t<img src=\""+(.img_rep_path)+"\
width=\"300\"">\t"+(.length|tostring)' | while read -r row;
#+"\t"+(.length|tostring)'
do
      IFS=$'\t' read -r -a items <<< "$row"
      echo "<tr>" | report
      for item in "${items[@]}"
      do
            echo "<td bgcolor=\"#F5D0A9\">" | report
            echo $item | report
            echo "</td>" | report
      done;
      echo "</tr>" | report
done
```

Figure 4.9: Database search and report creation.

4.2 Subprocesses

Individual commands can process one or more files. Using a pipe can connect the functionality of multiple commands where the output of one of them is the input for another. At the same time, scripts allow commands to be connected in an even more advanced way very close to programming. However, what enables even more advanced application of commands is the management of them from programming languages by modules that allow the program to spawn new processes, connect to their input/output/error pipes, and obtain their return codes (*Python*, 2021). One such module is the *subprocess* of the Python programming language.

List. 4.1 shows an example of using a subprocess in the Python programming language. In the example, the Python script runs several CLI commands whose functionality is linked to achieving the same goal, including the **python3** command itself, which runs another Python script, then a shell script, and finally an application that uses GUI.

Listing 4.1: Example of subprocess application in Python programming language.

```
import subprocess
a='example'
def show_file_type_stat():
     command = "ls | cut -d'.' -f2 | sort | uniq -c"
     ret = subprocess.run(command, capture_output=True, shell=True)
     print(','.join(ret.stdout.decode().split('\n')))
command = [f"libreoffice --headless --convert-to pdf {a}.doc",
f"pdftoppm -png {a}.pdf {a} ",
"python3 face_detection.py ./",
"./to_html.sh",
"firefox test.html"]
for cmmnd in command:
    show_file_type_stat()
    ret = subprocess.run(cmmnd, capture_output=True, shell=True)
```

The goal of the script from the List. 4.1 is to convert a forensic report from a Microsoft Word file into a PDF document that can be viewed via an internet browser. The PDF file is then converted to a PNG format where a separate image represents each page. In the next step, the resulting PNG files are processed by focusing and normalizing using the **convert** command. The new Python script is used for face detection where found faces are extracted into new PNG files whose name starts with "k_...". Specific naming of face image files allows the Bash script *to_html.sh* to create a report with detected faces in HTML format, which is then displayed using an internet browser started by **firefox** command. The report is shown in Fig. 4.11.

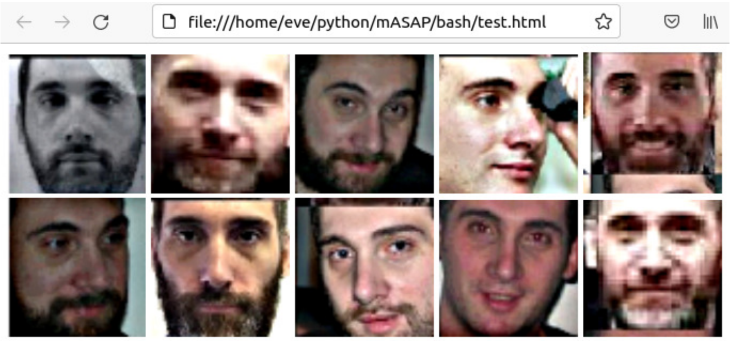
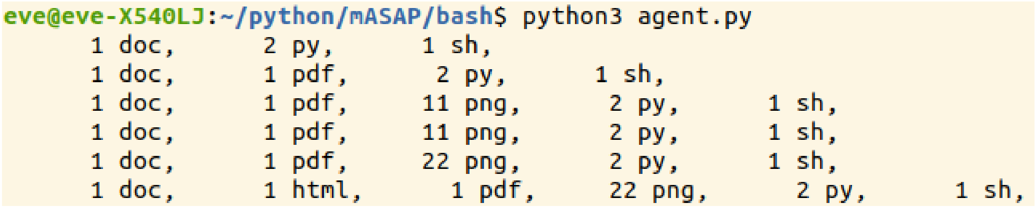


Figure 4.10: Detected faces report that is the result of the Python script execution.

In Fig. 4.11 function of the Python script *show_file_typ_stat()* at each step displays the number of different file types in the folder being created. It starts with a forensic report file *example.doc*, two Python scripts, and one Bash script. The next step shows files resulting from the conversion, face detection, and report creation. The two lines have the same number of files because when using the **convert** command for processing images, the command only modifies existing files.



```
eve@eve-X540LJ:~/python/mASAP/bash$ python3 agent.py
      1 doc,      2 py,      1 sh,
      1 doc,      1 pdf,      2 py,      1 sh,
      1 doc,      1 pdf,     11 png,      2 py,      1 sh,
      1 doc,      1 pdf,     11 png,      2 py,      1 sh,
      1 doc,      1 pdf,     22 png,      2 py,      1 sh,
      1 doc,      1 html,      1 pdf,     22 png,      2 py,      1 sh,
```

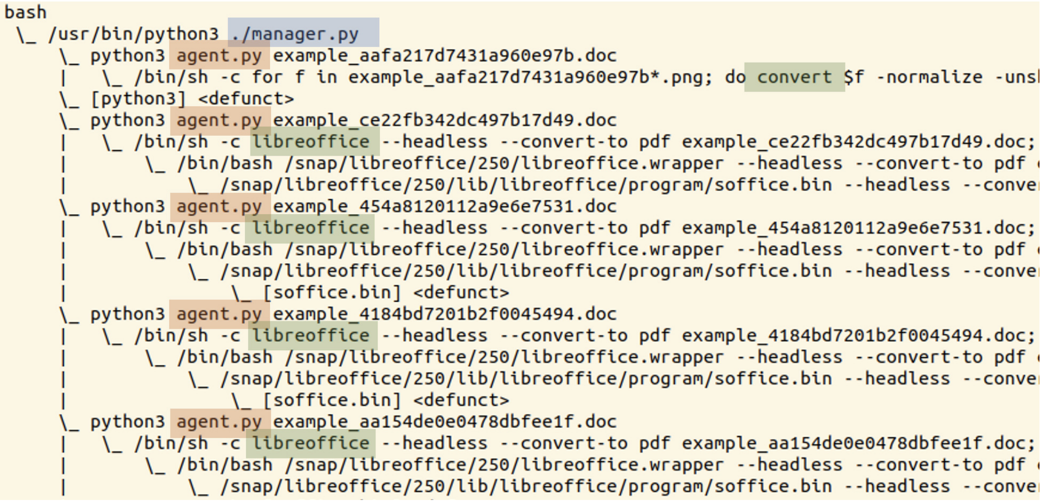
Figure 4.11: Output data obtained during Python script execution.

Running subprocesses in the background makes it possible to perform multiple data processing simultaneously. The Python script named *manager.py*, which content is displayed on List. 4.2 simulates concurrent data processing by running the script *agent.py* from List. 4.1 five times. In List. 4.2 the Popen function from the *subprocess* module starts the child processes.

Listing 4.2: Concurrent creation of subprocess from Python programing language.

```python
import subprocess
from subprocess import Popen
import secrets
import os
a='example.doc'
for i in range(5):
    b='example_'+secrets.token_hex(10)+'.doc'
    ret = subprocess.run(f"cp {a} {b}; sleep 5", capture_output=True, shell=True)
    Popen(['python3', 'agent.py', b])
```

With slightly modified code to use command line arguments, script *agent.py* is initiated five times so that after creating five copies of *example.doc* file with pseudo-random string suffix in the name, the conversion process starts again simultaneously. Fig. 4.12 displays a snapshot of the process on the system after running five instances of the *agent.py* script and several created child processes running competitively. In Fig. 4.12, it could be seen one **convert** and four **libreoffice** command subprocess that enhance images and convert the *example.doc* to PDF file, respectively.

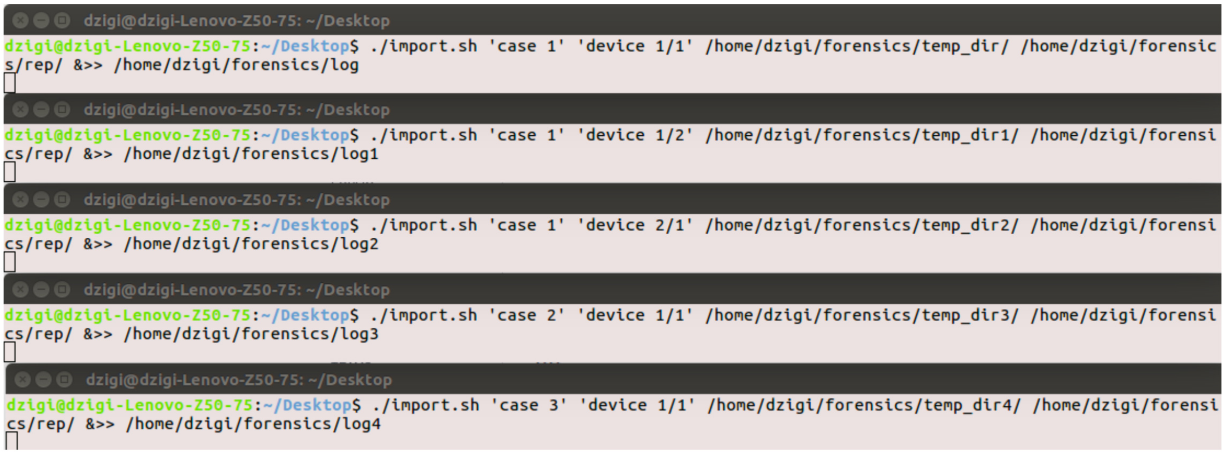


```
bash
 \_ /usr/bin/python3 ./manager.py
     \_ python3 agent.py example_aafa217d7431a960e97b.doc
     |   \_ /bin/sh -c for f in example_aafa217d7431a960e97b*.png; do convert $f -normalize -uns
     \_ [python3] <defunct>
     \_ python3 agent.py example_ce22fb342dc497b17d49.doc
     |   \_ /bin/sh -c libreoffice --headless --convert-to pdf example_ce22fb342dc497b17d49.doc;
     |       \_ /bin/bash /snap/libreoffice/250/libreoffice.wrapper --headless --convert-to pdf
     |           \_ /snap/libreoffice/250/lib/libreoffice/program/soffice.bin --headless --conve
     \_ python3 agent.py example_454a8120112a9e6e7531.doc
     |   \_ /bin/sh -c libreoffice --headless --convert-to pdf example_454a8120112a9e6e7531.doc;
     |       \_ /bin/bash /snap/libreoffice/250/libreoffice.wrapper --headless --convert-to pdf
     |           \_ /snap/libreoffice/250/lib/libreoffice/program/soffice.bin --headless --conve
     |               \_ [soffice.bin] <defunct>
     \_ python3 agent.py example_4184bd7201b2f0045494.doc
     |   \_ /bin/sh -c libreoffice --headless --convert-to pdf example_4184bd7201b2f0045494.doc;
     |       \_ /bin/bash /snap/libreoffice/250/libreoffice.wrapper --headless --convert-to pdf
     |           \_ /snap/libreoffice/250/lib/libreoffice/program/soffice.bin --headless --conve
     |               \_ [soffice.bin] <defunct>
     \_ python3 agent.py example_aa154de0e0478dbfee1f.doc
     |   \_ /bin/sh -c libreoffice --headless --convert-to pdf example_aa154de0e0478dbfee1f.doc;
     |       \_ /bin/bash /snap/libreoffice/250/libreoffice.wrapper --headless --convert-to pdf
     |           \_ /snap/libreoffice/250/lib/libreoffice/program/soffice.bin --headless --conve
```

Figure 4.12: A snapshot of the process showing concurrent command execution.

4.3 Results and discussion

Digital forensic cases often include several suspects. Each of them has several devices, which is why reports contain large amounts of data. In order to achieve the required efficiency in the work, parallel import of data from several reports from the same case or several different cases at the same time is needed (Vuković, 2019). Launching, for example, one script with various instances of the terminal application in Linux can concurrently import data from one computer or multiple computers in the network (Fig. 4.13).


dzigi@dzigi-Lenovo-Z50-75: ~/Desktop

```
dzigi@dzigi-Lenovo-Z50-75:~/Desktop$ ./import.sh 'case 1' 'device 1/1' /home/dzigi/forensics/temp_dir/ /home/dzigi/forensic
s/rep/ &>> /home/dzigi/forensics/log
```

dzigi@dzigi-Lenovo-Z50-75: ~/Desktop

```
dzigi@dzigi-Lenovo-Z50-75:~/Desktop$ ./import.sh 'case 1' 'device 1/2' /home/dzigi/forensics/temp_dir1/ /home/dzigi/forensi
cs/rep/ &>> /home/dzigi/forensics/log1
```

dzigi@dzigi-Lenovo-Z50-75: ~/Desktop

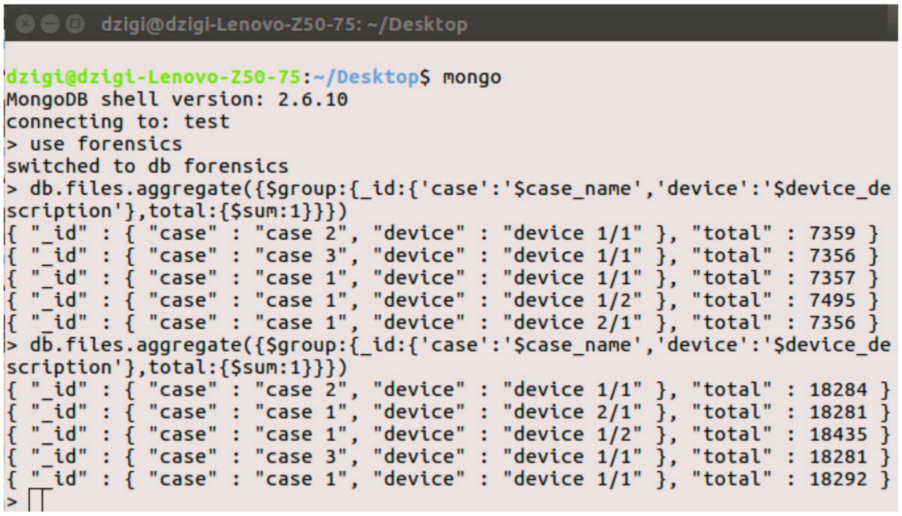
```
dzigi@dzigi-Lenovo-Z50-75:~/Desktop$ ./import.sh 'case 1' 'device 2/1' /home/dzigi/forensics/temp_dir2/ /home/dzigi/forensi
cs/rep/ &>> /home/dzigi/forensics/log2
```

```
dzigi@dzigi-Lenovo-Z50-75:~/Desktop$ ./import.sh 'case 2' 'device 1/1' /home/dzigi/forensics/temp_dir3/ /home/dzigi/forensi
cs/rep/ &>> /home/dzigi/forensics/log3
```

```
dzigi@dzigi-Lenovo-Z50-75:~/Desktop$ ./import.sh 'case 3' 'device 1/1' /home/dzigi/forensics/temp_dir4/ /home/dzigi/forensi
cs/rep/ &>> /home/dzigi/forensics/log4
```

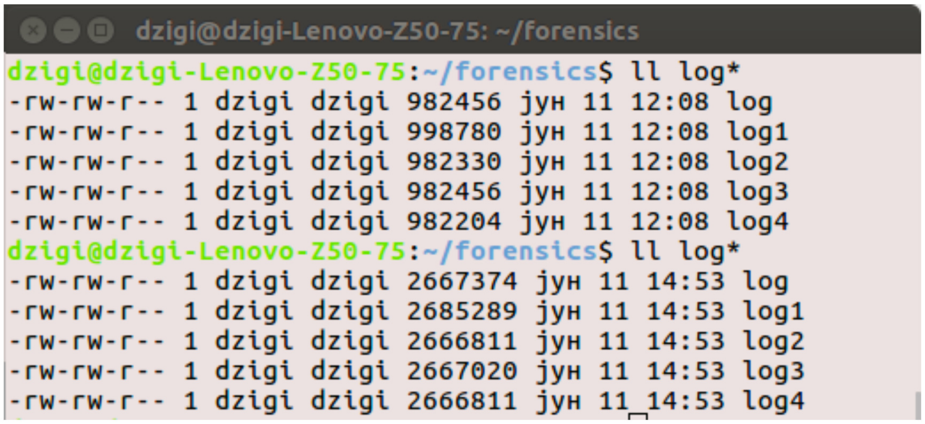
Figure 4.13: The five instances of the Terminal program use the same script and at the same time import data into the database.

Fig. 4.14 and 4.15 show the simultaneous import of data by recording two different states of the importing system, a change in the number of documents in the database, and the size of the log files.



```
dzigi@dzigi-Lenovo-Z50-75:~/Desktop$ mongo
MongoDB shell version: 2.6.10
connecting to: test
> use forensics
switched to db forensics
> db.files.aggregate({$group:{_id:{'case':'$case_name','device':'$device_de
scription'},total:{$sum:1}}})
{ "_id" : { "case" : "case 2", "device" : "device 1/1" }, "total" : 7359 }
{ "_id" : { "case" : "case 3", "device" : "device 1/1" }, "total" : 7356 }
{ "_id" : { "case" : "case 1", "device" : "device 1/1" }, "total" : 7357 }
{ "_id" : { "case" : "case 1", "device" : "device 1/2" }, "total" : 7495 }
{ "_id" : { "case" : "case 1", "device" : "device 2/1" }, "total" : 7356 }
> db.files.aggregate({$group:{_id:{'case':'$case_name','device':'$device_de
scription'},total:{$sum:1}}})
{ "_id" : { "case" : "case 2", "device" : "device 1/1" }, "total" : 18284 }
{ "_id" : { "case" : "case 1", "device" : "device 2/1" }, "total" : 18281 }
{ "_id" : { "case" : "case 1", "device" : "device 1/2" }, "total" : 18435 }
{ "_id" : { "case" : "case 3", "device" : "device 1/1" }, "total" : 18281 }
{ "_id" : { "case" : "case 1", "device" : "device 1/1" }, "total" : 18292 }
>
```

Figure 4.14: Change of the number of documents in the database during import.

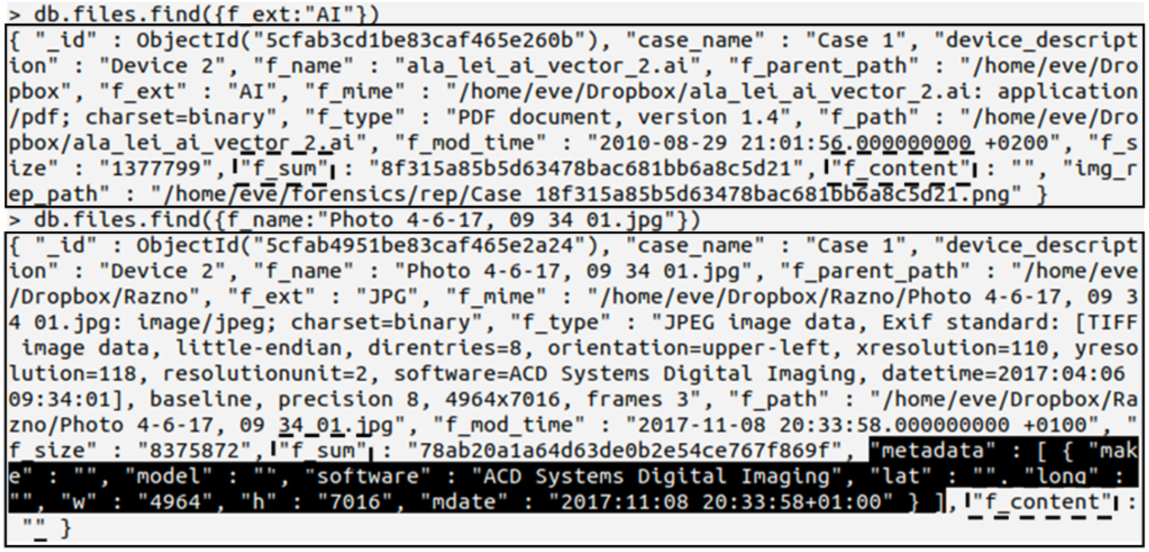


```
dzigi@dzigi-Lenovo-Z50-75:~/forensics$ ll log*
-rw-rw-r-- 1 dzigi dzigi 982456 јун 11 12:08 log
-rw-rw-r-- 1 dzigi dzigi 998780 јун 11 12:08 log1
-rw-rw-r-- 1 dzigi dzigi 982330 јун 11 12:08 log2
-rw-rw-r-- 1 dzigi dzigi 982456 јун 11 12:08 log3
-rw-rw-r-- 1 dzigi dzigi 982204 јун 11 12:08 log4
dzigi@dzigi-Lenovo-Z50-75:~/forensics$ ll log*
-rw-rw-r-- 1 dzigi dzigi 2667374 јун 11 14:53 log
-rw-rw-r-- 1 dzigi dzigi 2685289 јун 11 14:53 log1
-rw-rw-r-- 1 dzigi dzigi 2666811 јун 11 14:53 log2
-rw-rw-r-- 1 dzigi dzigi 2667020 јун 11 14:53 log3
-rw-rw-r-- 1 dzigi dzigi 2666811 јун 11 14:53 log4
```

Figure 4.15: Change of the log file size during the data import.

Open-source tools that can meet the requirements in terms of competitive data import and storage are database managers and client CLI applications of software packages of these managers (Vuković, 2019). There are two types of managers by the databases they manage: Relational and NoSQL (Could be Non-SQL or Not only SQL). The use of NoSQL databases for forensic purposes is suitable even though they do not provide the implementation of Atomicity, Consistency, Isolation, Durability (ACID) properties of a database transaction that should provide validity even in the event of errors, system failures for various reasons as relational managers do (Roussev, 2011). After the initial import of data, forensic investigators access data only by reading and not changing or deleting them, so consistency issues of the database are not expected (Federici, 2013).

The advantage of using NoSQL databases is that the concept of a document is much closer to file properties which ultimately differentiates by type, quantity, and content of its metadata and other parameters (Vuković, 2019). For example, in Fig. 4.16, there are two MongoDB collection documents, the first document contains data about a file that does not have metadata, and the other includes metadata without the need to explicitly change the structure as would be the case with relational databases. NoSQL databases solve this problem by offering the ability to store objects and matrices in the same field (Boicea, Radulescu, & Agapin, 2012).

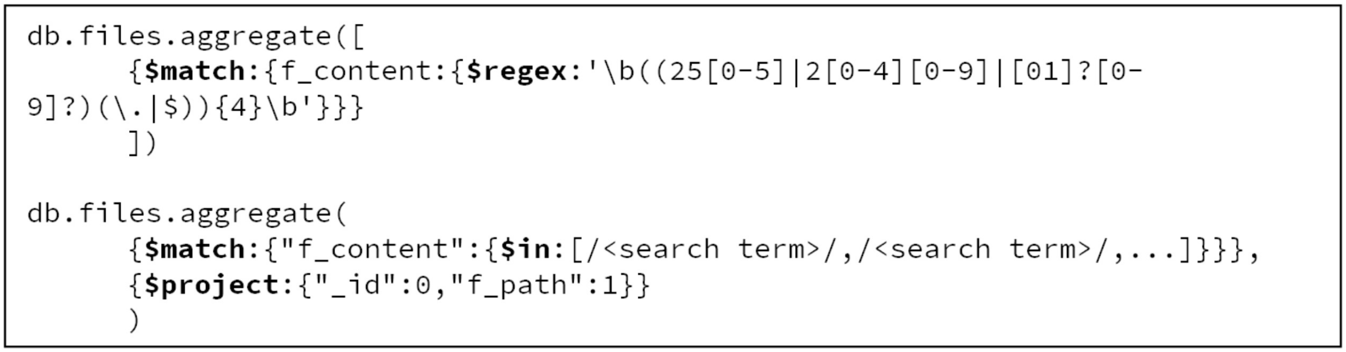


```
> db.files.find({f_ext:"AI"})
{ "_id" : ObjectId("5cfab3cd1be83caf465e260b"), "case_name" : "Case 1", "device_descript
ion" : "Device 2", "f_name" : "ala_lei_ai_vector_2.ai", "f_parent_path" : "/home/eve/Dro
pbox", "f_ext" : "AI", "f_mime" : "/home/eve/Dropbox/ala_lei_ai_vector_2.ai: application
/pdf; charset=binary", "f_type" : "PDF document, version 1.4", "f_path" : "/home/eve/Dro
pbox/ala_lei_ai_vector_2.ai", "f_mod_time" : "2010-08-29 21:01:56.000000000 +0200", "f_s
ize" : "1377799", "f_sum" : "8f315a85b5d63478bac681bb6a8c5d21", "f_content" : "", "img_r
ep_path" : "/home/eve/forensics/rep/Case 18f315a85b5d63478bac681bb6a8c5d21.png" }
> db.files.find({f_name:"Photo 4-6-17, 09 34 01.jpg"})
{ "_id" : ObjectId("5cfab4951be83caf465e2a24"), "case_name" : "Case 1", "device_descript
ion" : "Device 2", "f_name" : "Photo 4-6-17, 09 34 01.jpg", "f_parent_path" : "/home/eve
/Dropbox/Razno", "f_ext" : "JPG", "f_mime" : "/home/eve/Dropbox/Razno/Photo 4-6-17, 09 3
4 01.jpg: image/jpeg; charset=binary", "f_type" : "JPEG image data, Exif standard: [TIFF
 image data, little-endian, direntries=8, orientation=upper-left, xresolution=110, yreso
lution=118, resolutionunit=2, software=ACD Systems Digital Imaging, datetime=2017:04:06
09:34:01], baseline, precision 8, 4964x7016, frames 3", "f_path" : "/home/eve/Dropbox/Ra
zno/Photo 4-6-17, 09 34 01.jpg", "f_mod_time" : "2017-11-08 20:33:58.000000000 +0100", "
f_size" : "8375872", "f_sum" : "78ab20a1a64d63de0b2e54ce767f869f", "metadata" : [ { "mak
e" : "", "model" : "", "software" : "ACD Systems Digital Imaging", "lat" : "", "long" :
"", "w" : "4964", "h" : "7016", "mdate" : "2017:11:08 20:33:58+01:00" } ], "f_content" :
 "" }
```

Figure 4.16: An example of two documents from the same database collection with a difference in the structure

Scalability is another important feature of the NoSQL database manager. The amount of new information entering into potential analytical systems is significant, and it is difficult to anticipate what resources are sufficient to meet the needs of the job. The ability to expand storage capacity without disrupting the functioning of the working database gives priority to NoSQL systems relative to relational managers (Boicea et al., 2012).

In addition to storage space and collaboration requirements, the basis for each system that works with data like those from forensic reports is the ability to search to find information or digital evidence (Koen, 2009). In Fig. 4.17, there are two examples of search queries that use regular expressions for finding IP addresses, as well as searching multiple terms at the same time and displaying only the data from the corresponding fields (in this case, it is an absolute path).



```
db.files.aggregate([
      {$match:{f_content:{$regex:'\b((25[0-5]|2[0-4][0-9]|[01]?[0-
9]?)(\.|$)){4}\b'}}}
      ])

db.files.aggregate(
      {$match:{"f_content":{$in:[/<search term>/,/<search term>/,...]}}},
      {$project:{"_id":0,"f_path":1}}
      )
```

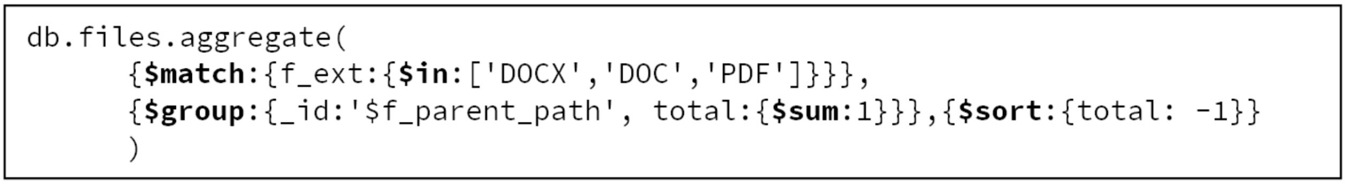
Figure 4.17: MongoDB queries enable different searches for forensic artifacts.

NoSQL databases are more efficient than relational when it comes to response rates. The same applies to open-source relational database managers (Győrödi, Győrödi, Pecherle, & Olah, 2015), as to the proprietary database managers, even for those which are considered an industry-standard (C. M. Wu, Huang, & Lee, 2015). That is why NoSQL databases are a recommended solution for intensive enrolment applications, and queries against massive amounts of data (Győrödi et al., 2015). Creating indexes for fields containing texts further accelerates the search.

Open-source software is often criticized for support and work on specific projects based on enthusiasm. In this research, scripts consist of open-source tools and commands whose primary purpose is not digital forensics but an everyday application in different areas of human work. The number of users of such applications exceeds the size of the forensic community gathered around a particular tool, allowing errors to be detected and repaired more efficiently (Vuković, 2019).

Customizing an analytical system that contains forensic report data by selecting only specific tools related to a particular context to improve the efficiency and effectiveness of the script leads to usability. Commands and Bash scripts give all needed functionality and offer a wide selection of different applications for different tasks.

Queries to the MongoDB database, which results in a static overview of the number of specific files, allow data triage. It represents a way to shorten the time of analyzing all the collected evidence by focusing on specific directories or file types instead of doing the keyword search of all files (Cohen et al., 2011). Saving, for example, only information about the file that the script collects from the file system into a database is a process that takes a relatively short time. After that, a query on a database can help with the triage by showing the locations of searched file types (also number of other criteria can be used, or their combination) sorted by their quantity in the specific location, as shown in Fig. 4.18. Importing only content of folders where files of interest are located could save time and resources.



```
db.files.aggregate(
      {$match:{f_ext:{$in:['DOCX','DOC','PDF']}}},
      {$group:{_id:'$f_parent_path', total:{$sum:1}}},{$sort:{total: -1}}
      )
```

Figure 4.18: An example of a query that helps during triage.

Using the script shown earlier in Fig. 4.9 for database search and report creation, after collecting information about files, it is possible to create a report in HTML format, which is convenient for submitting reports since all operating systems in use today contain internet browsers.

The application of subprocesses enables the management of commands and processing of their outputs directly from the programming language code. In this way, any program can be functionally expanded in the shortest time of design, development, testing, and implementation, which is especially important when there is a need to convert many different formats into those needed to search by content present data using web technologies. Also, any system built on this concept can be continuously improved by merely adding new open-source tools with a necessary set of options.

The use of open-source commands and scripts that combine their functionality is also suitable for system transparency and clarity, necessary for the development of the police system because it can be obtained a clearer insight into what a particular tool does and how, regardless of its specific application. Moreover, if further transparency is needed, insight into the selected tools can be achieved, which corresponds to the nature of this type of software.

4.4 Summary

The open-source concept speaks of freedom, such as using the software for any practical purpose; freedom to study an application's source code; free redistribution of the applications, and the improvement of the software for a particular purpose (Koen, 2009). Also, the fact is that such software is, in most cases, free of charge. Ubiquitous computing increases the volume and variety of data available for digital forensic analysis (Popović, Kuk, & Kovačević, 2018). Free tools developed under various open-source licenses allow reading, converting, storing, and searching a large number of different types of files found in forensic reports or collected in other ways in the work of the criminal police. The advantage of using different open-source tools for gathering digital evidence using Bash commands and scripts is that it shortens development time, and the system can continue to evolve gradually. Any new tool alone or combined with other tools can upgrade the script or be used to create a new one, thus improving the system.

Working with Bash scripts looks like programming. However, it is much closer to organizing the execution of various tools while choosing their options is the only condition for gaining the expected results. A combination of commands, scripts, Python subprocess for control, and NoSQL database storage capacity provides all elements need for developing a police system for working with a large amount of data.

Chapter 5

Influence of image enhancement techniques on effectiveness of unconstrained face detection and matching

In everyday work, police investigators routinely use different image processing software to obtain enhanced images that are more suitable for visual inspection and decision-making regarding identity resolution. It is tedious work requiring a lot of effort and time that they often do not have. The state-of-the-art solutions based on machine learning algorithms trained on massive databases to provide high face recognition effectiveness are expensive and unavailable for most police organizations. Therefore, to create a system that would perform the clustering of faces as accurately as possible and thus the grouping of evidence necessary for identity resolution based on pairwise matching feature vectors, it is necessary to examine the possibility of the impact of image enhancement on this process. The essential requirement is that image enhancement can work with limited hardware resources. A consumer laptop with an Intel i3-5005U dual-core 2.0 GHz base frequency processor and 4 GB of RAM was used in the experiment to simulate limited resources. The first step is to conduct an experiment to establish which one (if any) of the basic image enhancement techniques should be applied before the unconstrained face detection and matching to increase the effectiveness (Vukovic et al., 2021).

At the beginning of the experiment, copies were created with sharpened, normalized, resized images, and so on for each photo from the databases. Next, the HOG-based face detection is performed on the unprocessed images and all made copies, followed by the feature vectors extraction. The last step of the recognition process consists of creating a working set representing the intersection of the sets of all obtained feature vectors, which will be considered in the matching process. Only the faces detected by applying all the enhancement techniques were considered and further analyzed during the experiment.

5.1 Image enhancement

For the experiment, images are enhanced from four databases using several operations: contrast-stretching (normalization), sharpening, high-pass filters, reducing the speckles within a photo, and resizing, as well as a combination of these operations. Enhancement is automated using the *convert* application belonging to the ImageMagick 7.0 set of tools. It is free software that, over a command-line interface, creates, edits, composes, and converts raster graphics in many different formats, using multiple threads for a calculation to increase the performance (*ImageMagick*, 2020). Each enhancement starts with unprocessed images (UNP) of four databases.

One of the main characteristics of the databases is the different resolutions of the images used. Fig. 5.1 shows the resolution distribution of the photographs in the databases used in the experiment. The graph shows that the number of images with a resolution lower than 150 pixels is uneven within databases, but in total such images participate to a sufficient extent that the influence of resizing can be significant. Therefore, the *convert* application and -resize option were used to increase all face images whose size is less than 150 pixels. Data processed this way are marked with R, or if it is a combination of different preprocessing methods in question, R is at the beginning.

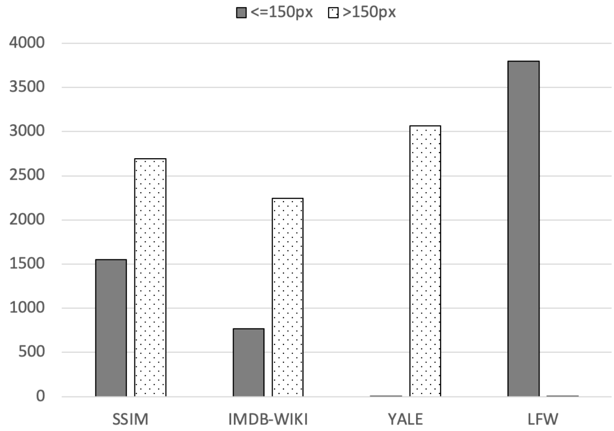
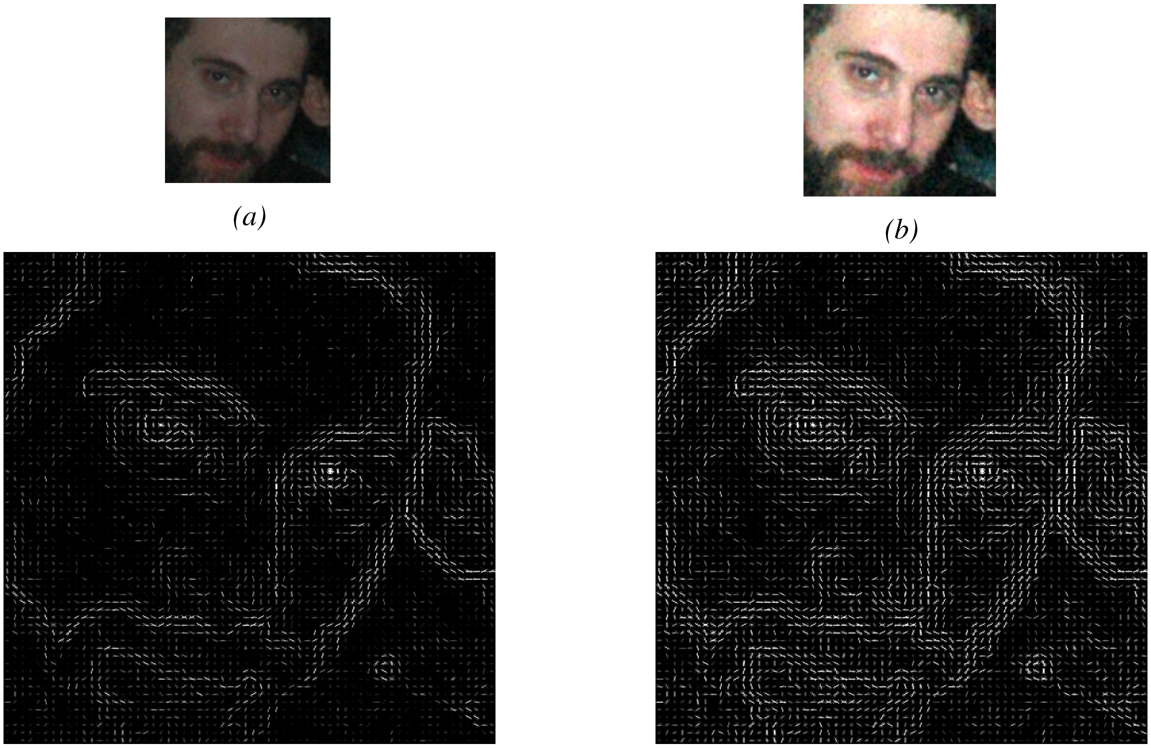


Figure 5.1: Resolution distribution in the databases used in the experiment.

The normalization (N) represents the increase in contrast in an image by stretching the range of intensity values. It is necessary to determine the lower and upper pixel value limits. Here, based on the previous setting of the operator of the *convert* command, the lowest pixel intensity value will be 2% of black-point values and the highest value 99% of white-point. The operator uses histogram bins to determine the range of color values that have to be stretched. Fig. 5.2a and 5.2b show the raw image and the image after applied preprocessing operation containing resizing and normalization. Fig. 5.2c and Fig. 5.2d show a visual representation of the HOG, created using the *skimage* Python library (Van der Walt et al., 2014), before and after enhancement. The first two figures show a visual improvement of the image, while the other two show a change in representation HOG. There are face contours with more information after normalization is applied.

Operator -unsharp (S) of the application *convert* sharpens an image by performing convolution with a Gaussian operator of the given radius and standard deviation (sigma) (*ImageMagick*, 2020). The selected standard deviation value is 5, and the larger number produces a sharper image. The radius is 0, which means that a suitable radius for the corresponding sigma value is chosen by *convert* application (*ImageMagick*, 2020).

A high-pass filter (HP) and resize and despeckle (RD) combines several different operators within the *convert* application to sharpen images and remove noise by blurring, respectively.



(a)

(b)

(c)

(d)

Figure 5.2: Comparison of a) unprocessed and b) enhancement image and representation of HOG descriptors c) before and d) after applied preprocessing.

5.2 Experimental settings

Fig. 5.3 shows the course of the experiment. Images from each database are firstly processed with different enhancement methods previously described. Subsequently, the HOG and ResNet functions from the Dlib library were used to perform the face detection and face-feature vectors' extraction.

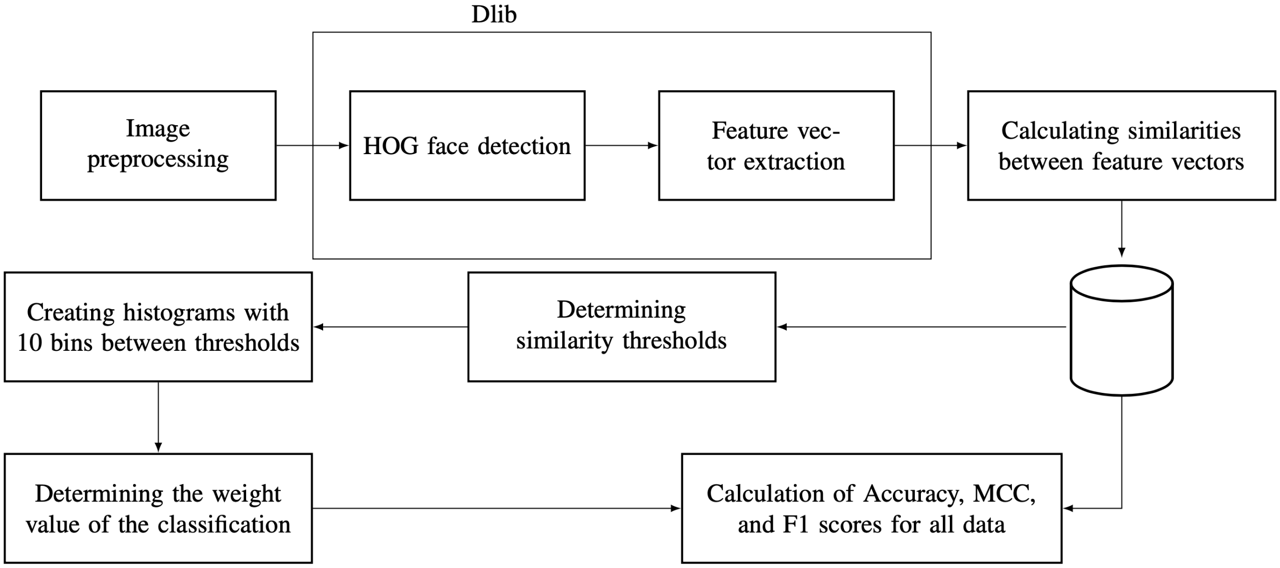

Dlib
Image preprocessing
HOG face detection
Feature vector extraction
Calculating similarities between feature vectors
Creating histograms with 10 bins between thresholds
Determining similarity thresholds
Determining the weight value of the classification
Calculation of Accuracy, MCC, and F1 scores for all data

Figure 5.3: The procedure of conducting the experiment

The working set V_{WS} is determined by comparing the vectors from the sets of images obtained after different enhancements, and the intersection of all sets created by preprocessing is taken. In other words, since ten data sets were used within the experiment, an individual working set (for example working set with sharpened face images) builds only the feature vector that appears ten times. Suppose a face is not detected only after normalization, but it is otherwise, his feature vector will still not be included in any individual working set. The result is testing on sets that are further reduced compared to the initial number of images in the face image database. However, this sets the same starting point for the face matching examination eliminating the influence of the face detection phase of the experiment on the face matching phase. This part of the experiment is presented in more detail in Alg. 1.

Finally, face matching is made using cosine similarity (distance), a measure of similarity based on each vector's component composition. This similarity measure between two non-zero vectors was chosen because it is a naturally normalized distance in which the outcome is bounded [0, 1].

The matching process consists of determining the similarity value (ϑ) between feature vectors representing the detected faces and belonging to the working set by a combination without repetition $\binom{|V_{WS}|}{2}$.

$$\vec{V} \in \mathbb{R}^{128} \tag{5.2.1}$$

$$V_{WS} = \{\vec{V}_1, \vec{V}_2, \vec{V}_3, ..., \vec{V}_i, \vec{V}_j, ..., \vec{V}_n\} \tag{5.2.2}$$

$$\{(\vec{V}_i, \vec{V}_j) \mid \vec{V}_i, \vec{V}_j \in V_{WS}, i \neq j\} \tag{5.2.3}$$

$$\vartheta = cos\Theta = \frac{\vec{V}_i \cdot \vec{V}_j}{||\vec{V}_i||\,||\vec{V}_j||} \tag{5.2.4}$$

Algorithm 1 Automation of image enhancement and creation of a working set.

```
    Input: Database with unprocessed face images for recognition phase
$I_{UNP} \leftarrow \{I_1, I_2, ...I_n\}$
    Output: A working set to be used for face matching composed of a face
feature vectors $V_{WS}$
  $V_S \leftarrow \{\}$ //S - sharpened images
  $V_N \leftarrow \{\}$ //N - normalized images
  $V_{NS} \leftarrow \{\}$ //NS - normalized and sharpened images
  $V_{RD} \leftarrow \{\}$ //RD - resizing and despeckle combination
  ...
  $V_{RHP} \leftarrow \{\}$ //RHP - resizing and high-pass filter
  for $i \leq n$ do
    $I_{Si} \leftarrow ($convert $I_i$ -unsharp $0x5)$
    $\vec{V_i} \leftarrow dlib.face\_recognition\_model\_v1().compute\_face\_descriptor(I_{shi})$
    $V_S \cup \{\vec{V_i} \in \mathbb{R}^{128}\}$
    $I_{Ni} \leftarrow ($convert $I_i$ -normalize$)$
    $\vec{V_i} \leftarrow dlib.face\_recognition\_model\_v1().compute\_face\_descriptor(I_{ni})$
    $V_N \cup \{\vec{V_i} \in \mathbb{R}^{128}\}$
    ...
    $I_{RHP} \leftarrow ($convert $I_i$ -resize $150x\backslash <$ -hp$^*)$
    $\vec{V_i} \leftarrow dlib.face\_recognition\_model\_v1().compute\_face\_descriptor(I_{rhp})$
    $V_{RHP} \cup \{\vec{V_i} \in \mathbb{R}^{128}\}$
  end for
  $V_{WS} \leftarrow V_S \cap V_N \cap ... \cap V_{RHP}$

  * the specified option in real application replaces a series of commands
```

In the experiment, the confusion matrix was used to determine the relationship between the prediction obtained by selecting the appropriate threshold and outcomes (C.-C. Wu et al., 2019).

Fig. 5.4 shows the confusion matrix for a binary classification.

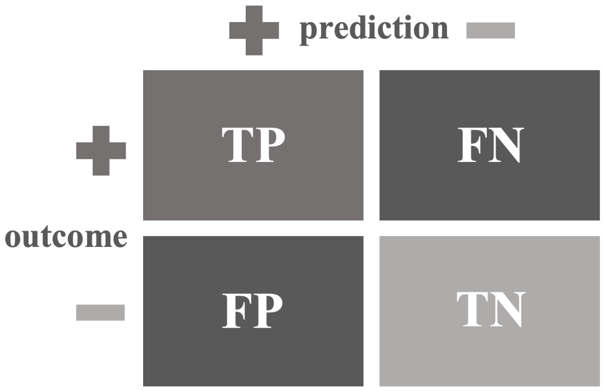


Figure 5.4: The confusion matrix for a binary classification.

When matching face images, TP or true positives are all classification results that correctly predict that two images belong to the same person, while TN or true negatives are all classification results that correctly suggest different individuals on images. Similarly, FP or false positives are the results of incorrect classification that there is the same person on both matched images, while FN or false negatives are all errors that the classification algorithm makes incorrectly assuming that faces on images belong to different persons.

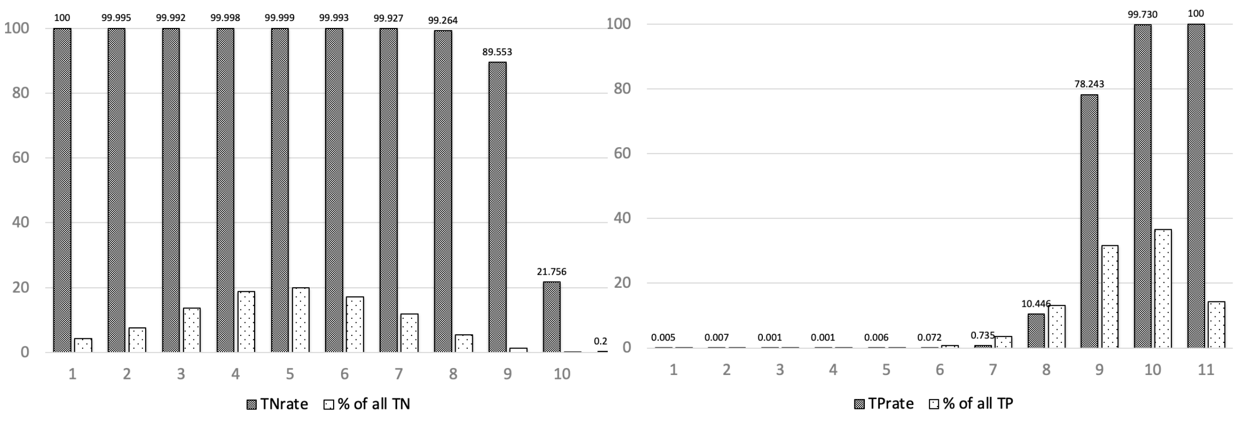
Related to the confusion matrix is the $TNrate$ or $Specificity$ in Fig. 5.5a illustrates the percentage of negative samples correctly classified, while $TPrate$ or $Recall$ ($Sensitivity$) shown in the example in Fig. 5.5b represents the percentage of positive samples correctly classified (Zou, Xie, Lin, Wu, & Ju, 2016).

$$TPrate\ (Recall) = \frac{TP}{TP+FN} \tag{5.2.5}$$

$$TNrate\ (Specificity) = \frac{TN}{FP+TN} \tag{5.2.6}$$

Before selecting the threshold to measure the impact of enhancement on the effectiveness of face matching on all databases, and after storing the similarity values in the database, the similarity thresholds are determined to represent the uncertainty interval's boundaries. Within the uncertainty interval, $TPrate$ and $TNrate$ are below 100%. In other words, the ϑ value cannot be determined whether it is the same or different individuals (Vukovic et al., 2021). The upper limit of the interval for each of the subsets of enhancement images is the maximum similarity value where the matching result is TN (ϑ^{-}_{max}). The lower limit of the interval is the minimum similarity value where the face matching result is TP (ϑ^{+}_{min}).

Histograms such as the examples shown in Fig. 5.5a and Fig. 5.5b are used during the analysis of the experiment's data.



(a)

(b)

Figure 5.5: Examples of histograms of the distribution of a) true negatives (TN) and b) true positives (TP).

The bin width h of the histogram represents the difference between the values of previously defined maximums and minimums similarity limits divided by k, the chosen number of bins.

$$h = (\vartheta^{-}_{max} - \vartheta^{+}_{min})/k \tag{5.2.7}$$

In Fig. 5.5a, the uncertainty interval is shown by bins 2 to 11, while in bin 1, TNrate is 100% ($\vartheta < \vartheta^{+}_{min}$). The uncertainty interval in 5.5b, where the focus is on true positives, is covered by bins 1 to 10, while in bin 11, TPrate is 100% ($\vartheta > \vartheta^{-}_{max}$). The unique threshold or weight value (w) for the set of similarities related to enhancement applied over all four databases were calculated using the mean value ($\overline{\vartheta}$) and standard deviation (s).

$$\overline{\vartheta} = \frac{1}{4}\sum_{i=1}^{4} \vartheta_i,\ TPrate\ > 0.9 \tag{5.2.8}$$

$$s = \sqrt{\frac{1}{3}\sum_{i=1}^{4} (\overline{\vartheta} - \vartheta_i)^2} \tag{5.2.9}$$

$$w = \overline{\vartheta} - s \tag{5.2.10}$$

The weight value w according to the rule by which all similarity values $\vartheta > w$ indicate the same person was used for classification. In contrast, in the case of $\vartheta <= w$, faces belong to different persons in the pictures.

5.3 Results and discussion

The experimental results suggest that applying different types of image enhancement impacts the total number of detected faces. By counting the feature vectors from each set of enhanced images, the data shown in Tab. 5.1 and Fig. 5.6 are obtained, representing the impact of enhancement on face detection effectiveness. The up arrow (↑) in Tab. 5.1 shows an increase in the number of detected faces by applying some visual enhancement methods compared to unprocessed images. The down arrow (↓) indicates that the number of detected faces decreased after preprocessing, while the equals sign (=) suggests that the number of detected images did not change. Based on the symbols in Tab. 5.1, it can be concluded that enhancement has a significant positive effect on the number of detected faces and that the best results we achieved by applying sharpening after enlarging images with a resolution of less than 150 pixels.

Table 5.1: The number of detected faces by databases and applied image enhancement methods.

	HP	N	S	NS	UNP	R	RHP	RN	RS	RD
SSIM	5,514 ↑	5,465 ↑	5,542 ↑	5,556 ↑	5,419 =	5,488 ↑	5,552 ↑	5,534 ↑	**5,615** ↑	5,421 ↑
IMDb-Wiki	9,618 ↓	9,611 ↓	**9,701** ↑	9,696 ↑	9,623 =	9,587 ↓	9,607 ↓	9,582 ↓	**9,701** ↑	9,623 =
Yale B	3,358 ↑	3,340 =	3,539 ↓	**3,542** ↑	3,340 =	3,340 =	3,358 ↑	3,340 =	3,539 ↓	3,129 ↓
LFW	3,956 ↑	3,949 ↑	3,984 ↑	3,975 ↑	3,940 =	3,941 ↑	3,959 ↑	3,948 ↑	**4,005** ↑	3,972 ↑

Fig. 5.6 gives more clarity than the table in the presentation of the overall impact of face detection's effectiveness concerning the samples from all databases used in the thesis. The chart in Fig. 5.6 shows the mean value of the number of detected persons for all databases after min-max normalization. The image sharpening gives the best results independently (S, RS) and with normalization (NS). Comparing the results of applying the same enhancement method with and without resizing in Fig. 5.6 shows that resizing images with a resolution of less than 150 pixels improves face detection. It should be noted that although the resizing of the image had an impact on the number of detected faces, it is still relevant only when it comes to laboratory conditions and images from face databases. In real conditions, the resolution of the face image cannot be determined before detection (Vukovic et al., 2021). Still, the results show that the effect of image sharpening still gives better results than other image preprocessing methods.

The differences in the number of detected faces in Tab. 5.1 emphasize the need to measure the impact of image enhancement on results of the matching process's effectiveness based on a previously created working set.

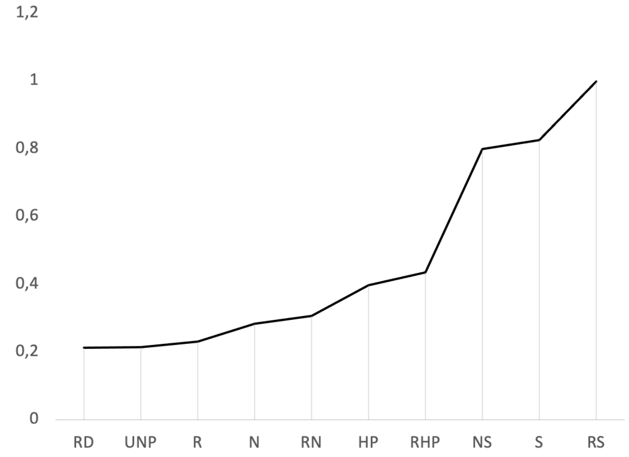
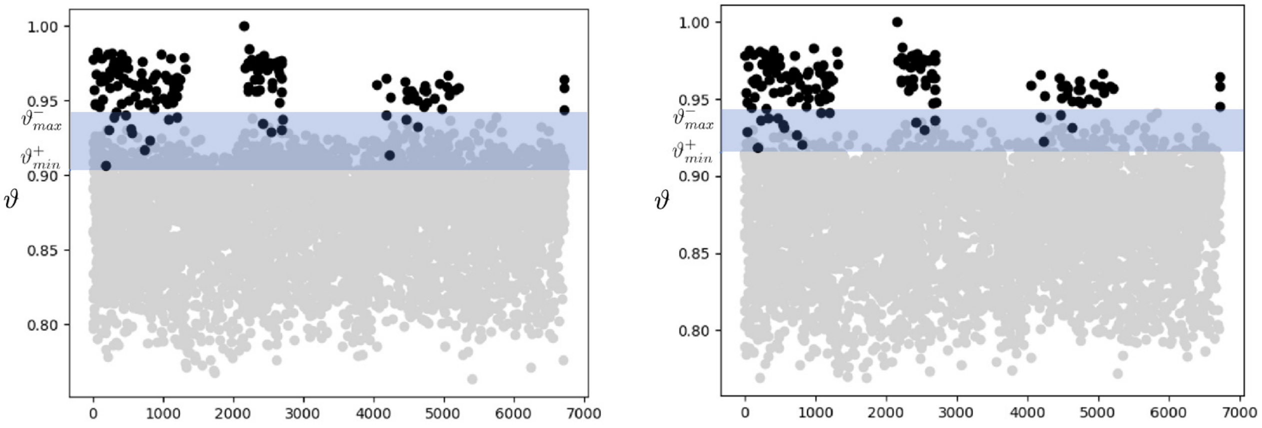


Figure 5.6: Impact of enhancement on the total number of detected persons.

In the simulation of the face matching applied in the experiment, a binary classification was performed by comparing the faces from one image with all other faces from the working set of a specific image enhancement method within the database. Then the procedure is repeated for each remaining subsequent image. Fig. 5.7a and 5.7b show the result of the first step of the described face matching process relating to the same face image, before and after enhancement, respectively.

The similarity value indicating the same person is represented by dark dots in Fig. 5.7a and Fig. 5.7b, whereas light dots represent different people's faces.

To determine the weight value w based on which the binary classification would be performed, there is a problem determining the most optimal similarity ϑ. The problem is visually presented as a shaded area in Fig. 5.7a and Fig. 5.7b, in this thesis named the uncertainty interval. The uncertainty interval is bordered by the threshold values above which all comparisons indicate the same individual (all dark spots) and below which it can be claimed that all matched faces belong to different persons (all bright spots). Also, the comparison of Fig. 5.7a and Fig. 5.7b shows how the uncertainty interval changes after image enhancement. In this case, the advantage of normalizing unprocessed images is noticeable. The continuing matching with other faces from the images produces different intervals within the same set. The goal is to make the overall interval as narrow as possible and with fewer similarity values within the uncertainty interval.



(a) *(b)*

Figure 5.7: Similarity values were obtained by matching one face with all others in the SSIM database with a) unprocessed images and b) normalization applied. The similarity value indicating the same person is represented by dark dots, while light dots represent different people's faces.

By measuring the results based on face matching from all sets by different image enhancement methods and then normalizing the obtained values for a unified representation on the chart, Fig. 5.8 is obtained. The chart shows the uncertainty interval's size ratio and the number of similarities whose values indicate the same and different individuals, i.e., located within the interval. In addition, the interval size affects the number of similarities within the interval. The best results are given by applying high-pass filters with and without resizing (RHP and HP, respectively). Comparing all enhancement methods combined with resizing of face images does not affect the results to the extent that the processing with which it is combined has impacted.

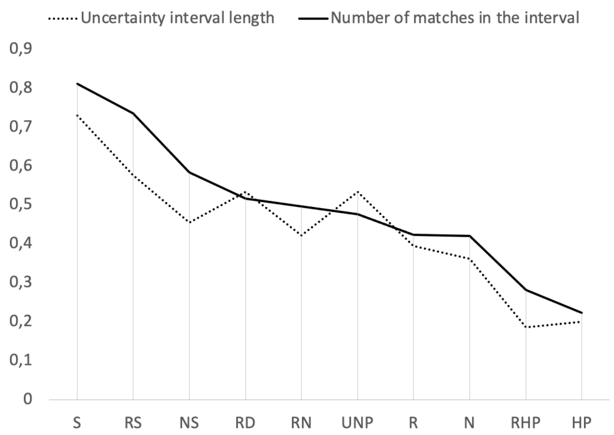


Figure 5.8: The relationship between the size of the uncertainty interval and the number of similarities within the interval after applying different image enhancement methods.

The face matching in the actual application related to a criminal investigation is characterized by a disproportionate number of comparisons that give positives and negatives. The results were analyzed from both perspectives separately in the experiment since sometimes it is crucial to be able to exclude possible suspects (TN) from further investigation, and in that manner, free occupied resources. The examples of histograms shown in Fig. 5.5a and 5.5b were used for this purpose. For example, in Fig. 5.5a, data from the bin in which $TNrate$ is equal to 100% were used, i.e., comparisons give correct results that they are different persons. Also, data from the first eight bins give a high degree of certainty that there are different persons on matched images, over 99%.

The data were first normalized within the same database, then the mean value was calculated for all four databases for each set of applied image enhancements. Fig. 5.9 shows a chart of the impact of enhancement on effectiveness in the matching process where it can be argued that these are different persons. The best results in claiming that these are different images with 100% certainty are obtained by sharpening using a high-pass filter (HP, RHP).

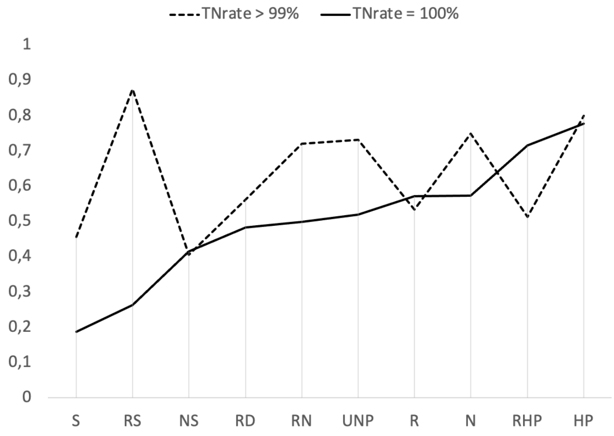


Figure 5.9: The total ratio of the number of correctly classified negatives through all databases and applied the image enhancement method with a high $TNrate$.

The influence of different image enhancement methods at $TNrate$ of 100% fully corresponds to the comparison of the size of the uncertainty interval. When it comes to the certainty greater than 99% that these are different individuals, which also covers most of the number of matching, the best results give sharpening (RS, HP) and normalization.

Similarly, the impact of different image enhancement methods on effectiveness in determining that it is most likely the same person is compared (shown in Fig. 5.10). In the classification where it can be claimed with complete certainty that the result of matching is the same person's face, the best results are obtained by applying sharpening with resizing face images of less than 150 pixels and then by normalization. The large number of comparisons in which it can be claimed with a certainty of over 90% that they are the same persons, including the most correctly performed classifications with positives, are obtained by applying normalization (RN, N).

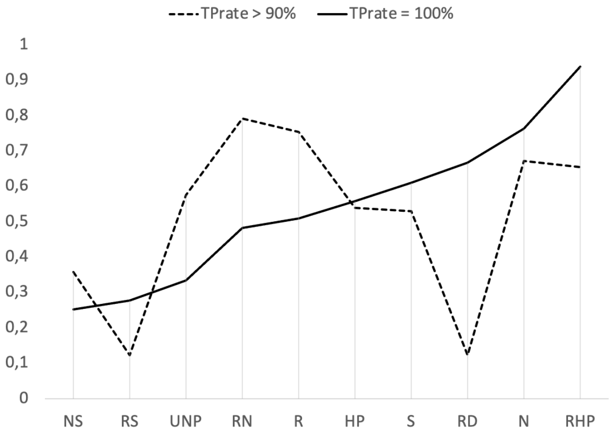


Figure 5.10: The total ratio of the number of correctly classified positives through all databases and applied the image enhancement method with a high $TPrate$.

Finally, data classification from all databases is performed using the appropriate unique weight value *w* for a particular image enhancement method. The scores for all three measures of effectiveness (Acc, F1, and MCC) are shown in Tab. 5.2.

Table 5.2: Scores of the classification performed using the weight value w.

	$\overline{\vartheta}$	s	Acc	F1	MCC
HP	0.9468	0.0046	0.9934 ↓	0.8156 ↓	0.8248 ↓
N	0.9457	0.0060	**0.9943** ↑	**0.8464** ↑	**0.8513** ↑
NS	0.9467	0.0058	0.9930 ↓	0.8035 ↓	0.8129 ↓
UNP	0.9460	0.0042	0.9940=	0.8336=	0.8406=
RD	0.9465	0.0031	0.9924 ↓	0.7784 ↓	0.7927 ↓
R	0.9468	0.0050	0.9940=	0.8344 ↑	0.8413 ↑
RHP	0.9452	0.0047	0.9939 ↓	0.8319 ↓	0.8385 ↓
RN	0.9456	0.0051	0.9942 ↑	0.8397 ↑	0.8457 ↑
RS	0.9448	0.0040	0.9933 ↓	0.8138 ↓	0.8222 ↓
S	0.9441	0.0056	0.9937 ↓	0.8267 ↓	0.8322 ↓

The relationship of the effect made by applied image enhancement methods with the performed normalization of the results data is shown in Fig. 5.11. The best score is obtained by applying normalization, first without and then with resizing. In contrast, although unprocessed images receive a high score at the applied weight value of *w*, resizing gives better results than purely unprocessed images. Regardless of the applied evaluation method, the scoring ratio is uniform.

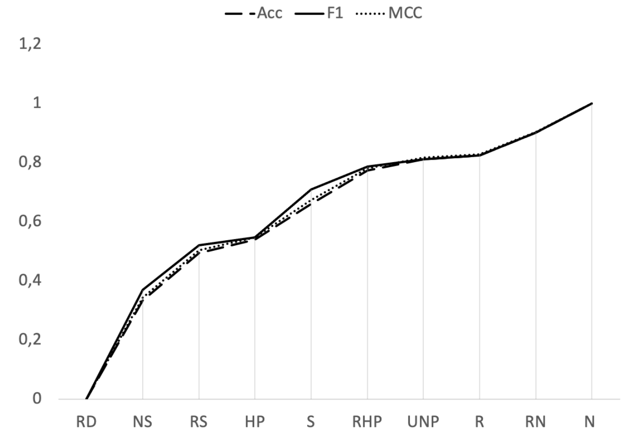


Figure 5.11: The ratio of scores for classification by applying weight values w.

5.4 Summary

This chapter has experimentally demonstrated the influence of image enhancement on unconstrained face detection and matching. The used databases are labeled, making it possible to identify persons and measure image enhancement effects. In addition, face databases simulate the conditions for analyzing unconstrained face images by criminal investigators.

The conclusions can be summarized as follows:

- Application of different image enhancement techniques improves the effectiveness of the used Dlib HOG solutions.
- Although the resizing in combination with the sharpening of the face images has the best results in face detection, this type of preprocessing is applicable only in experimental conditions of using face databases. What is important is that applying just sharpening gives nearly as good results, and therefore it should be applied to improve face detection accuracy.
- Evaluation using Acc, F1, and MMC scores after applying the same weight value *w* (classification threshold value) on all four databases shows that enhancing images with normalization gives the best face matching results.

Chapter 6

Truth-value unconstrained face clustering

Criminals try to mislead law enforcement agencies and hide their actions by changing identities, names and appearances. Providing insight into all collected digital evidence to identify the suspect, resolve his identity, provide the courts with only the necessary evidence, and reduce incidental findings requires an appropriate data processing and presentation method. Decisions and conclusions about which person is in question, i.e., who is the real person who falsely presents himself and what evidence corroborates that, ultimately need to be made by the criminal police officer himself. The system that supports this process must simplify its identity matching by grouping only the evidence relating to a particular person.

In their day-to-day work, the criminal police determine persons' identity and connection with the crime by applying various biometric data (fingerprints, face and voice recognition, and other data). Considering the great potential of biometric technologies for objective and quantitative evaluation of evidence (Champod & Tistarelli, 2017), it should be noted that one of the rare sources of physical biometric data that can be found in evidence collected by digital forensics, OSINT and similar methods are face features on images. Although it is a rare source of biometric data, the number of photos and videos found in evidence is significant, considering that cameras of increasing quality are present on all portable smart devices. Also, improved speeds and easy access to the internet in the last decade have made it possible to manipulate and distribute many images and videos.

Grouping of evidence by different criteria is naturally correlated with the clustering method. People use previously learned lineaments and possible variations of the same face to identify how many different subjects there are in a given collection of unknown face images (Lin, Chen, Castillo, & Chellappa, 2018). However, clustering images of faces belonging to the same person, creating "pseudo-classes," is not a trivial task for computers, especially regarding images created in unconstrained conditions.

6.1 Unconstrained face clustering

Face recognition can be considered a supervised classification problem, unlike face clustering, which is an unsupervised classification problem (Chang et al., 2019). The high precision of modern face recognition systems is based on large amounts of labeled training data (creating a ground-truth face database requires time-consuming, meticulous work). An alternative is sought in unsupervised or semi-supervised learning when sources are limited (Yang et al., 2019; Zhan et al., 2018).

Clustering, in general, can be applied in exploratory data mining, image analysis, information retrieval, data compression, pattern recognition and with the primary goal to group data points based on similarity, density, intervals of particular statistical distribution measures of the data space (Karim et al., 2021). Clustering can be divided into (Sinaga & Yang, 2020):

- Probabilistic model-based approaches, for example, the Expectation-Maximization algorithm.
- Nonparametric approaches which are based on similarity or dissimilarity measures and can be further divided into hierarchical and partial methods.

Probabilistic model-based approaches are also called soft clustering because data points are not assigned to exclusively one group, unlike hard clustering, i.e., nonparametric, where different objective functions maximize distances to connect elements only to the corresponding clusters. What is necessary for collecting information related to the criminal investigation is a method that will perform hard clustering of data about persons collected in real-world settings.

Hierarchical clustering is a recursive partitioning of a dataset into successively smaller clusters, for example, starting from a weighted graph with edge weight based on pairwise similarities or dissimilarities between data points (Cohen-Addad, Kanade, Mallmann-Trenn, & Mathieu, 2019). Depending on the process's direction, hierarchical clustering algorithms are divided into agglomerative and divisive, and the time complexity distinguishes them. Complexity is significantly higher regarding other nonparametric approaches, which is disadvantageous when processing large amounts of data.

In the case of partitional nonparametric methods, the primary is a determination of dissimilarity or distance between a point and cluster prototype, and the oldest and most popular method is k-means (Sinaga & Yang, 2020). The main problem is the initial determination of the number of clusters, which is mostly unknown, so different validity indices are applied. When it comes to face clustering, various nonparametric approaches are often combined with supervised machine learning techniques to overcome the shortcomings of this kind of clustering method.

Face clustering can have applications in various domains of everyday life and work, from grouping images in photo albums, organizing large photograph and video collections for fast retrieval in time-sensitive scenarios like forensic investigation, for social media imagery, as well as for providing (cleaning, labeling) large-scale data for deep convolution neural network training (Z. Wang et al., 2019; Lin, Chen, Ranjan, et al., 2018; Otto et al., 2017).

Face clustering, despite extensive research, remains a challenge when it comes to practical application with material characteristically for criminal evidence for two reasons: the problem of measuring similarity due to unconstrained conditions in which face images were created and knowing a number of clusters (individuals) necessary for the application of several well-established clustering algorithms such as k-means (Lin, Chen, Ranjan, et al., 2018). In addition, conventional methods make impractical assumptions on data distribution, k-means requires the clusters to be convex-shaped, spectral clustering requires that the cluster size be relatively balanced, and DBSCAN assumes different clusters to be in the same density (in the actual application density vary a lot), in summary, they cannot cope with complicated cluster structures (Yang et al., 2019; Guo et al., 2020). Unlike these methods, agglomerative hierarchical clustering is robust in grouping data with complex distribution because it makes no assumption (Z. Wang et al., 2019).

What is essential for this thesis is that different nonparametric approaches (hierarchical and partial) have examples of application from the similarity graph $G(V,E)$ defined for the given data, which indicates the importance of the approach itself. For example, spectral clustering finds the underlying structure based on the Laplacian graph (represent the difference of degree and adjacency matrix), while in hierarchical approaches, the data group is connected whenever the weight value of the edge is above some threshold (Lin, Chen, Ranjan, et al., 2018). Graph-based algorithms provide an easy way to represent both the objects and relations without limitation to the representation space, for example, feature vectors or the distances, which increases their practical application (Chang et al., 2019).

Graphs are also used to decide whether, for example, two data points belong to the same or different clusters. GCN extend CNN to process graph-structured data, which in existing research has shown advantages such as a strong capability of modeling complex graphical patterns, and good results in the process of classification nodes (whether similar or different) (Yang et al., 2019; Qi et al., 2021). GCN naturally uses local graph structure and can learn more discriminative features for classification purposes (Guo et al., 2020). However, this is supervised learning because the focus here is on finding a solution to achieve results without any model training.

One of the reasons why training should be avoided in unconstrained face clustering with limited resources can be seen through another clustering method. The method is a domain adaptation which is another way of face clustering and represents an attempt to transfer what has been learned from the source domain to the domain of the unknown target (Lin, Chen, Ranjan, et al., 2018). The problem during the domain shift between source and target is usually solved with a close-set (images of the source and target domain are from the same set), which is a problem in the actual application (M. Wang & Deng, 2020).

For the new approach proposed in this thesis, noteworthy are the results obtained by applying the Approximate Rank-Order (the solution is based on the assumption that images of persons belonging to the same person have the same first K_B nearest neighbors) distance achieve good clustering performance when it comes to unconstrained face datasets (Y. Shi et al., 2018).

6.1.1 Graph-based pairwise face clustering issues

The choice of the clustering algorithm and input data for examining its efficiency is challenging and depends on several factors. When it comes to data, authors of earlier scientific papers manually created facial features such as gradient and pixel intensity, allowing the use of classical methods like spectral clustering (Yang et al., 2019). However, data containing more distinctive characteristics and more suitable for biometric measurements, such as face landmarks, are needed for clustering when images are created in unconstrained conditions. Face landmark is a prominent feature that can play a discriminative role, such as the edges of the eyes and eye sockets, the root and tip of the nose, and the contour of the face, lips, and ears (Çeliktutan, Ulukaya, & Sankur, 2013). The precise location of these characteristic points on the face is an essential step in obtaining data whose further processing is not affected by the previous existence of unconstrained variables (lighting, poor image quality) and whose matching more accurately determines similarities and differences between individuals. Landmarks can be relatively easily detected using low-level image features such as histograms of gradients. A representation of this can be seen in Fig. 6.1.

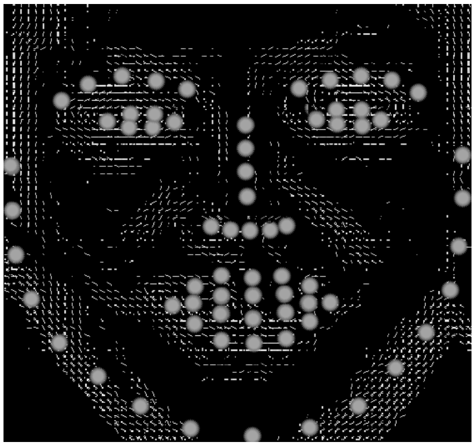


Figure 6.1: Representation of HOG face descriptors and 68-keypoints face landmarks.

The result of landmarking is a feature vector representing a reasonable amount of data about one person that allows the use of limited storage capacity (for example, it can be stored in database fields of textual type) and more efficient data processing with less demanding hardware resources. This thesis explores the possibilities of using a 128-dimensional feature vector obtained using HOG face descriptors and 68-keypoints landmarks. For that purpose off-the-shelf solution based on Dlib open source libraries can be used, which rationalizes the face detection process.

What is the primary problem when determining the similarity/distance threshold ϑ_t needed for classification of Dlib HOG descriptors, i.e, the feature vectors extracted with ResNet is shown in Fig. 6.2.

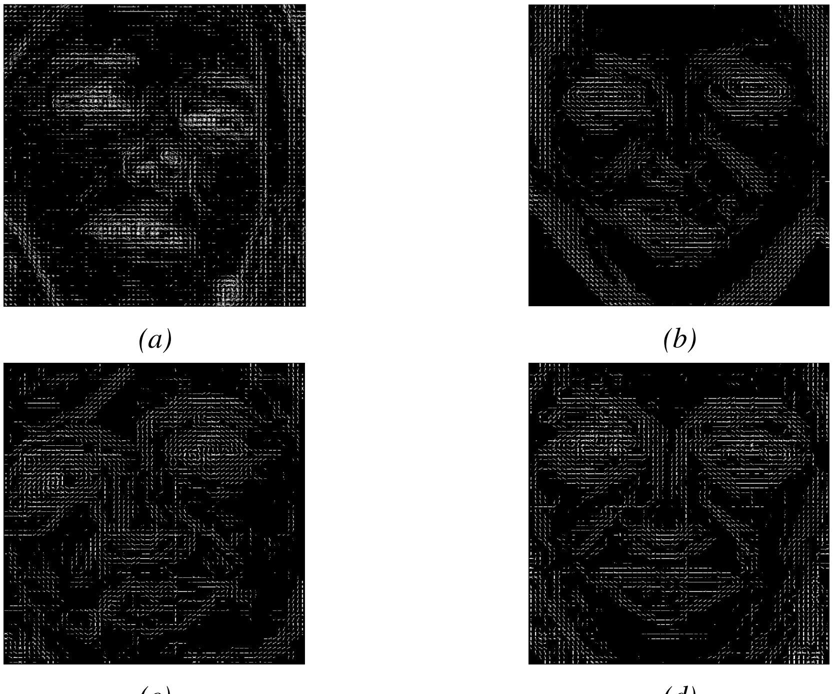


Figure 6.2: The calculation of the distance feature vectors obtained based on the HOG descriptors of the same person shown under a) and b) gives a lower value of cosine distance than the images under c) and d) belonging to different persons.

Fig. 6.2 shows a representation of the HOG descriptor in cases where the similarity value of ϑ between the faces of the same person (shown in Fig. 6.2a and 6.2b) is significantly lower than for the faces of different people (shown in Fig. 6.2c and 6.2d). If in the particular example, the similarity threshold was set $\vartheta_t = 0.94$, then the matching images in Fig. 6.2a and 6.2b would get false negatives as a result of the prediction. At the same time, in the case of face images on Fig. 6.2c and 6.2d result will be a false positive.

How the choice of the value ϑ_t affects the classification in such circumstances is shown in Fig. 6.3 where the image of one person from the SSIM database is compared to all other images in the set.

The similarity threshold ϑ_t in Fig. 6.3 is within the uncertainty interval $[\vartheta^{+}_{min}, \vartheta^{-}_{max}]$. The uncertainty interval is limited by the minimum value of similarity between face feature vectors belonging to the same person (ϑ^{+}_{min}) and the maximum value of similarity, indicating that these are vectors belonging to different individuals (ϑ^{-}_{max}).

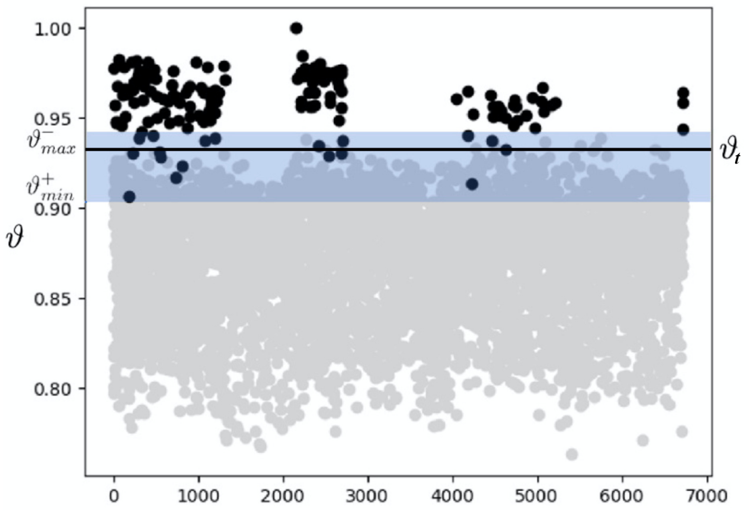


Figure 6.3: Result of determining the value of the similarity threshold ϑ_t within the uncertainty interval.

Fig. 6.3 shows 18 matchings which indicate that it is the same person and a larger number indicates there are different individuals solely within the interval of uncertainty.

Setting the threshold based on which the classification is performed to the value shown in Fig. 6.3 reduces the number of true positives that will be correctly classified, but reduces false negatives to the greatest extent. When evaluating the face classification, the applied ϑ_t would give satisfactory results. However, when it comes to clustering, especially graph-based, each false negative can represent a relationship between two clusters, as in the examples in Fig. 6.2 with very high values. In this way, the problem of precise clustering would be multiplied.

Clustering using the face feature vector can be done with a graph-based method and Euclidean distance calculation between two descriptors by choosing a single distance threshold value that gave the best possible results (Chang et al., 2019). Furthermore, given the problematic nature of data clustering (choice of similarity measure and a number of clusters), determination and analysis of distance itself, such as cosine similarity, can be extended with pairwise constraints such as "must-link" pairs and "cannot-link" pairs (Y. Shi et al., 2018). The novel approach of pairwise face clustering proposed in this thesis, named Truth-value clustering (TVC), uses NAL adaptation to solve the problems of determining pairwise similarities, primarily related to the existence of uncertainty intervals.

6.2 Experimental settings

An experiment was conducted to determine how the different values of the frequency f as part of the truth-value and the value of the similarity ϑ of feature vectors effects TVC unconstrained face clustering. The experiment used five different face databases, IMDb-Wiki, Extended Yale Face Database B, LFW, SSIM, and YouTube Faces database. The evaluation was performed using NMI, $B^3 Precision$, $B^3 Recall$ and $B^3 F$ score values.

6.2.1 System setup

The experiment is performed on eight identical virtual machines, assigned with four Intel Xeon Silver 4210R central processor units, running at 2.4 GHz clock speed, 32 GB working memory and 500 GB storage space. One virtual machine was created by installing the Ubuntu operating system (one of the major Linux distributions) and necessary software. After that, seven more identical copies were made.

The ArangoDB graph database was used for the purposes of the experiment. ArangoDB is also a NoSQL database, so it is horizontally extensible, can store different data structures, and required hardware is less demanding (Corbellini, Mateos, Zunino, Godoy, & Schiaffino, 2017). ArangoDB was chosen in this thesis as a multi-model database storing key-value pairs, documents, and graphs, and all data can be accessed in the same ArangoDB Query Language (AQL). Compared to other similar databases that specialize in graphing, Arangodb leads in performance, but its main advantage is still its multi-model architecture (Fernandes & Bernardino, 2018).

Python driver for ArangoDB with the following requirements (1) Python version 3.7+ and (2) ArangoDB version 3.7 + (*Python Driver for ArangoDB*, 2022), was used for automated database creation, reading, and writing to collections. A combination of different f and ϑ values resulted in 252 graph databases. It was intensively communicated with all databases for an extended period of over a half-year. As a result, the process went fluidly, indicating a stable connection and work with the database. Thus, the potential for applying the solution for practical purposes is evident.

6.2.2 Experiment run

The proposed face clustering framework based on the adaptation of non-axiomatic reasoning and graphs is shown in Fig. 6.4. Given an unlabeled face image gallery, face clustering is performed by (1) matching faces, (2) performing reasoning by determining the relationship of the cardinality of positive and negative evidence, and (3) collecting subgraphs representing clusters with faces. Fig. 6.4 shows a concept of the TVC clustering approach where, for a more straightforward presentation, the implementation of HOG descriptors and face feature vector extraction were not shown, as well as input parameters such as ϑ_t similarity threshold and truth-value frequency threshold f_t.

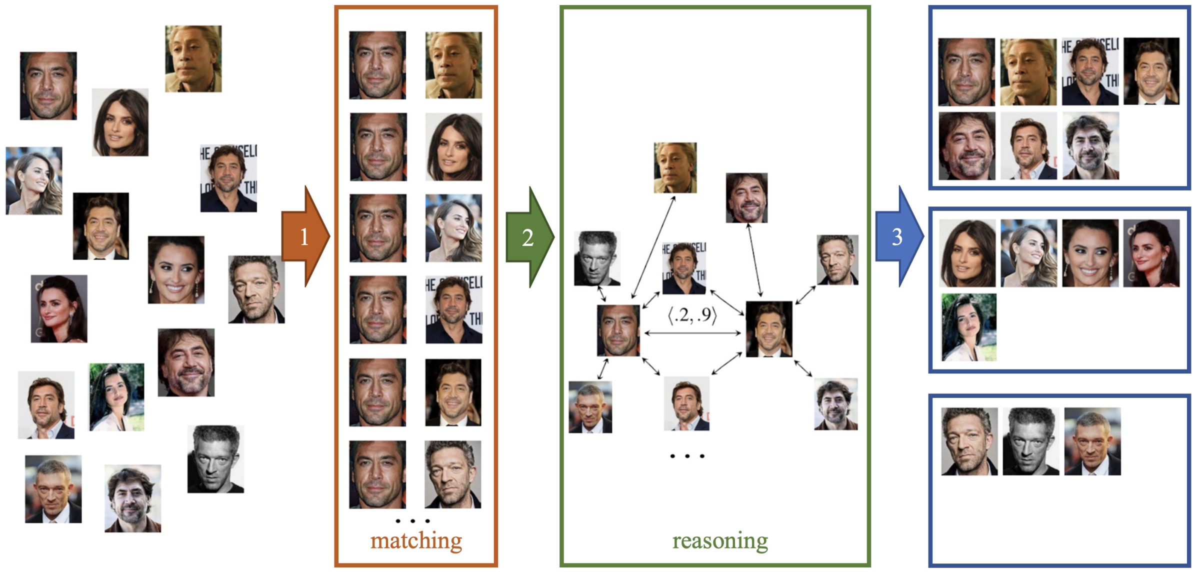


Figure 6.4: Proposed face clustering framework.

Based on the framework, the steps shown in Fig. 6.5 were determined by which the experiment was executed and where the implementation of the method was presented more precisely.

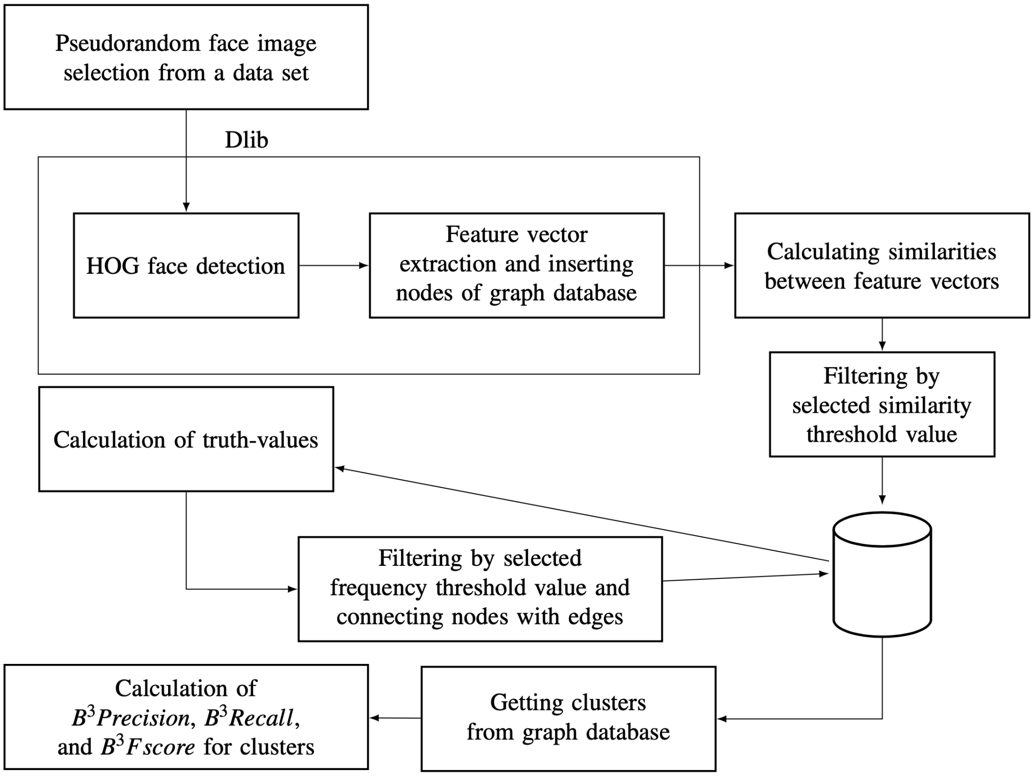

Pseudorandom face image selection from a data set
Dlib
HOG face detection
Feature vector extraction and inserting nodes of graph database
Calculating similarities between feature vectors
Filtering by selected similarity threshold value
Calculation of truth-values
Filtering by selected frequency threshold value and connecting nodes with edges
Calculation of $B^3Precision$, $B^3Recall$, and $B^3Fscore$ for clusters
Getting clusters from graph database

Figure 6.5: The steps of conducting the experiment.

Images of persons in used databases are grouped in separate directories based on the persons they belong to, i.e., they are sorted by names or codes that identify them. Therefore, the experiment begins with a pseudo-random selection of images from the face database to avoid the influence of the order on the amount of positive and negative evidence that determines whether two faces belong to the same or different faces. After selecting the image, face detection and feature vector extraction is performed using Dlib HOG and ResNet. Each feature vector represents a data point, i.e., a node in a new graph database created with each iteration of the experiment.

Creating nodes with a vector in the database, the distances ϑ between the feature vector and all recorded up to that point are calculated by reading from the database. The value of ϑ represents the edge's weight value between two vectors created and written to the database in the corresponding edge collection K_S. Within the experiment, the experience of the system K was divided into two collections of a graph database, similarity collection K_S and knowledge base K_B.The ϑ_t threshold is one of the input parameters of the experiment. It is determined before each iteration, and only graph edges are created in the database when the similarity value exceeds ϑ_t.

Algorithm 2 Creating a knowledge base.

```
  Input: Database with face image files $I \leftarrow \{I_1, I_2, ...I_n\}$
  Output: $V_A$ - verteces, $K_S$ - collection of edges with similarities, $K_B$ - collection of knowledge base
$V_A \leftarrow \{\}, K_S \leftarrow \{\}, K_B \leftarrow \{\}, i \leftarrow 1$
for $i \leq n$ do
    $\vec{V_i} \leftarrow Dlib.face\_recognition\_model\_v1().compute\_face\_descriptor(I_i)$
    $F \leftarrow \{"feature\_vec" : \vec{V_i} \in \mathbb{R}^{128}, "file\_name" : getFileName(I_i)"label" : getFaceLabel(I_i)\}$
    $V_A \cup \{F\}$
    if $i > 1$ then
        $j \leftarrow 1$
        for $j \leq |V_A|$ do
            $\vartheta_j \ = \ cos\Theta = \frac{F["feature\_vec"] \cdot V_A[\ j\ ]["feature\_vec"]}{\|F["feature\_vec"]\| \ \|V_A[\ j\ ]["feature\_vec"]\|}$
            if $\vartheta_j >= \vartheta_t$ then
                $E_\vartheta \leftarrow create\_edge(\vartheta_j, F["file\_name"], A_K[\ j\ ]["file\_name"])$
                $K_S \cup \{E_\vartheta\}$
            end if
        end for
        $z \leftarrow 1$
        for $z \leq |V_A|$ do
            $e_1 \leftarrow get\_pairs\_of\_F(F["file\_name"], K_S)$
            $e_2 \leftarrow get\_pairs\_of\_V_A[\ z\ ](V_A[\ z\ ]["file\_name"], K_S)$
            $f = \frac{|e_1 \cap e_2|}{|e_1 \cap e_2| + |e1 \setminus e2|}$
            if $f > f_t$ then
                $E_f \leftarrow create\_edge(f, F["file\_name"], A_K[\ j\ ]["file\_name"])$
                $K_B \cup \{E_f\}$
            end if
        end for
    end if
end for
```

The significance of the value ϑ_t selection is that the frequency f_t truth-value is further calculated based on it. The objective function that calculates f maximizes the distance between two feature vectors representing the faces of the different people and minimizes the distance between vectors of the same person. When calculating the frequency f, if the resulting value exceeds the threshold f_t

determined for the experiment iteration, a new edge will be created in a particular collection, the so-called knowledge base K_B. These are edges that form clusters with nodes at their ends. A more detailed representation of the first and second phases of the experiment, the calculation of similarities and frequencies, can be seen in Alg. 2.

The first two phases of the experiment execute sequentially for each extracted feature vector. In contrast, the last phase is performed after all data is entered into the database.

After calculating all distances and frequencies, queries are sent to obtain subgraphs from the knowledge base K_B for all data points. While collecting subgraphs, vertices contained in earlier subgraphs are removed from the data points list. In the practical application of the method, this step can be performed in parallel with the previous stages, whereby the result can be refreshed in decided intervals and allows the end-user, the criminal investigator, to react in a timely manner.

The last step of the evaluation is to calculate NMI, $B^3Precision$, $B^3Recall$ and B^3F score value, based on the cluster data obtained in the previous step. A more detailed representation of the last phase of the experiment can be seen in Alg. 3.

Algorithm 3 Collecting clusters.

```
    Input: $V_A \leftarrow \{F_1, F_2, ..., F_n\}, K_B \leftarrow \{E_f1, E_f2, ...E_fn\}$
    Output: $C \leftarrow \{C_1, C_2, ...C_n\}, NMI, B^3Precision, B^3Recall, B^3F$
  $C \leftarrow \{\}$
  $names \leftarrow \{\}$ //List of node names located in the cluster
  for $F$ in $V_A$ do
      if $F[''file\_name'']$ not in names then
          $C_t \leftarrow get\_subgraph($'FOR v, e, p IN 1..20 any F["file_name"] GRAPH $K_B$
  OPTIONS bfs: true, uniqueVertices: true, uniqueEdge: true RETURN distinct p.vertices[-1]._key')
          $C \cup \{C_t\}$
          $names \cup \{get\_list\_of\_names(C_t)\}$
      end if
  end for
  $(NMI, B^3Precision, B^3Recall, B^3F) \leftarrow get\_evaluation\_scores(C)$
```

6.3 Results and discussion

The proposed TVC method of clustering by adapting the truth-value concept from NAL from the perspective of time complexity consists of two main parts: (1) calculating distance, in this case, cosine similarity where only values above the corresponding threshold are recorded in the database, and (2) creating a knowledge base by determining truth-values. Runtime process of comparing face descriptors and creating a knowledge base K_B by applying the proposed method has quadratic growth with increasing the number of face images at the entrance.

Calculating similarities by big O notation has $O(n^2)$ complexity where n is the number of face images, and then the process of reasoning, i.e., creating a knowledge base, has $O(n\ log\ n)$ complexity, so overall time complexity is $O(n^2) + O(n\ log\ n) = O(n^2\ +\ n\ log\ n) = O(n^2)$. In a distributed system (which will be discussed in more detail in the next chapter), the reasoning process would also include turning to other nodes of the system. In the worst case, which is that the system itself does not have enough evidence for any of the data points, the time complexity of this part of the process would increase to $O(k\ n^2)$ where K_B would represent the number of data points of the node with the most extensive knowledge base (reasoning in nodes is performed simultaneously).

It should be noted that in the conditions of distributed system operation, when a request is sending only hash values of descriptors, i.e., where it is expected to find already existing relations in the knowledge base between two data points in adjacent nodes, then the complexity is increased by $O(k\ n\ log\ n)$. Although the worst-case scenario is considered when determining the time complexity, the complete absence of negative and positive evidence for all data points is unlikely, except at the beginning of the formation of the knowledge base (cold start). Regarding time complexity, Tab. 6.1 shows how specific system parameters in the clustering process affect the number of activities and thus the total runtime.

The total number of face matching could be obtained according to the formula $(n(n-1))/2$, where n is the number of images in the dataset. However, the number of distances/similarities used in the clustering process depends on the selected similarity threshold values ϑ_t and the frequency threshold value f_t. Selecting larger values of these two parameters reduces the number of items in the collection of edges with similarities K_S and the number of edges in the knowledge base K_B.

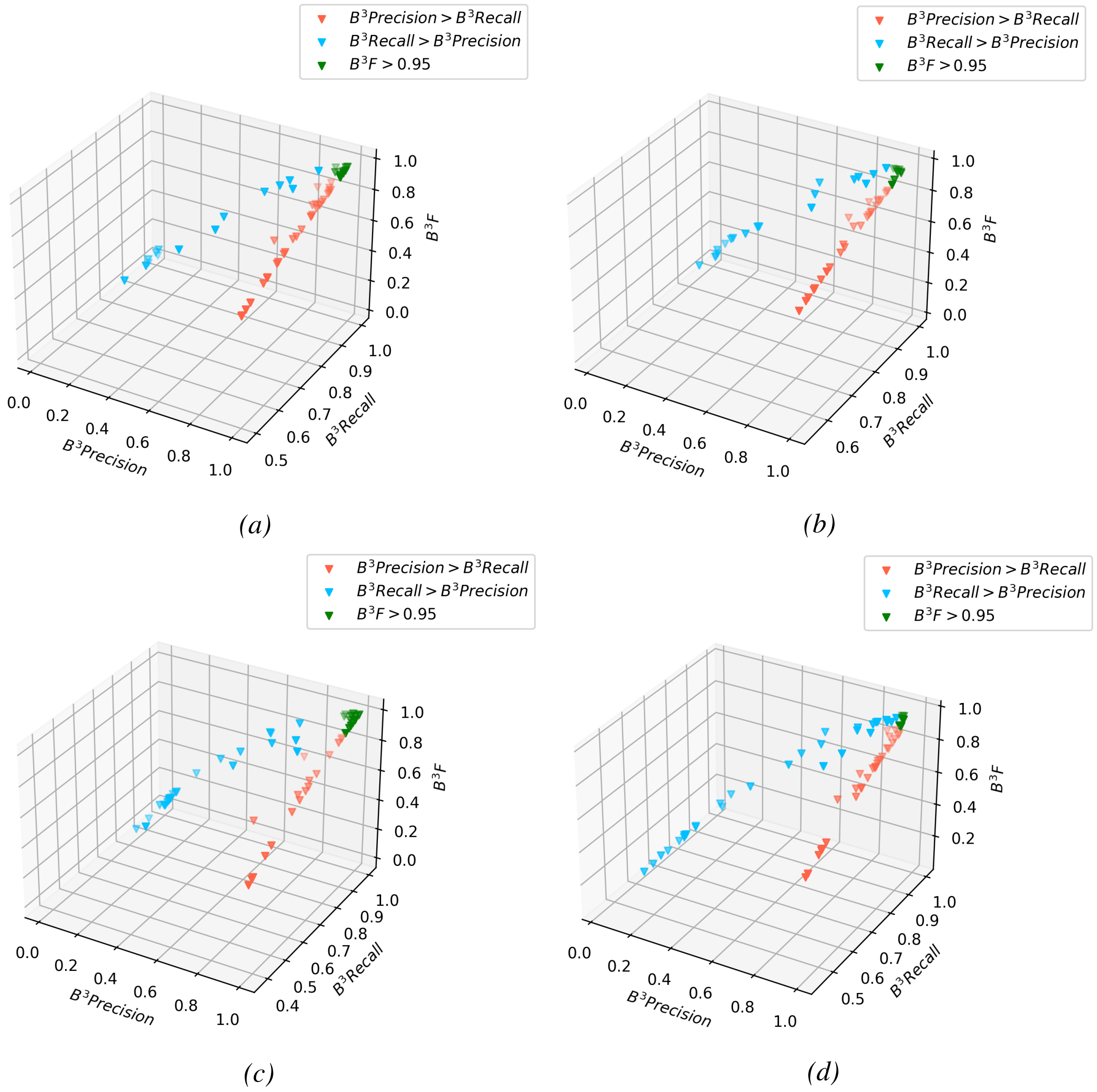
Runtime grows according to the established time complexity of the algorithm with an increasing amount of data. Still, communication with the graph database has a decisive influence on the duration of the process. In Tab. 6.1 the central emphasis shows the most intensive use of the database and the most time-consuming part of the process. The number of activities when creating edges in the similarity collection K_S is significantly higher than those with edges containing truth-value (frequency, i.e., knowledge base collection K_B), which crucially affects the duration of clustering. In other words, changes in f_t value from .100 to .700 in K_B collection affect the duration of the process by only a few minutes. Although the largest values in the table are the number of matching feature vectors, the duration of this part of the clustering process can compare with the time consumed by creating all the edges in K_S. Changes in the total duration of the process are noticeable only when the number of records in the collection with similarities K_S changes significantly.

Table 6.1: Performance comparison with different face image data sets.

	№ of img.	№ of matches	ϑ_t	$\lvert K_S \rvert$	f_t	$\lvert K_B \rvert$	Time (minutes)
IMDb-Wiki	9,581	45,892,990	.880	**5,616,098**	.100	319,042	2,711
					.700	171,180	2,764
			.900	**1,761,801**	.100	196,672	649
					.700	28,503	607
			.920	**461,569**	.100	23,497	312
					.700	697	306
SSIM	5,173	13,377,378	.880	**3,598,412**	.100	175,066	1,323
					.700	76,697	1,342
			.900	**1,375,829**	.100	105,874	354
					.700	31,010	354
			.920	**338,821**	.100	25,454	119
					.700	1,446	118
YTFaces	4,291	9,204,195	.880	**1,362,675**	.100	76,300	339
					.700	43,123	337
			.900	**427,589**	.100	48,286	119
					.700	27,993	118
			.920	**111,046**	.100	27,294	64
					.700	11,720	63
LFW	3,940	7,759,830	.880	**1,048,764**	.100	195,378	296
					.700	129,060	297
			.900	**388,376**	.100	138,793	122
					.700	6,744	111
			.920	**206,052**	.100	6,913	77
					.700	341	74
Yale-B	3,340	5,576,130	.880	**1,348,315**	.100	208,960	400
					.700	159,678	398
			.900	**562,217**	.100	171,592	152
					.700	79,885	141
			.920	**248,717**	.100	78,127	76
					.700	7,949	72

What is important to point out, as seen in Tab. 6.1, is that the uniform quality of face images in the Yale-B data set reflects values. The number of items in K_S is close to the results or greater than the two more numerous sets with corresponding f_t value. Furthermore, an increase in the number of records in the K_S collection has a positive effect on the knowledge base. Although the Yale-B set simulates many different unconstrained conditions, what is constant is the high resolution, i.e., faces on images are larger than the averages on other databases, and there all are in focus. Based on that, the importance of image enhancement before the detection and face matching process can be concluded.

Fig. 6.6 gives the experimental result, which shows that there is a stable combination of parameters of similarity threshold ϑ_t and frequency thresholds f_t, which results in high values B^3F score, as well as a uniform shift of values towards greater precision or better recall.



(a)

(b)

(c)

(d)

Figure 6.6: The experiment results of applying the proposed clustering approach with samples of a) IMDb-Wiki, b) SSIM, c) Yale B and d) LFW database.

Fig. 6.7 - 6.10 shows the effect of changing the input parameters (ϑ and f_t) on $B^3Precision$, $B^3Recall$ and B^3F when $B^3F\ score > 0.95$. To emphesize potencial pattern change, the values are normalized.

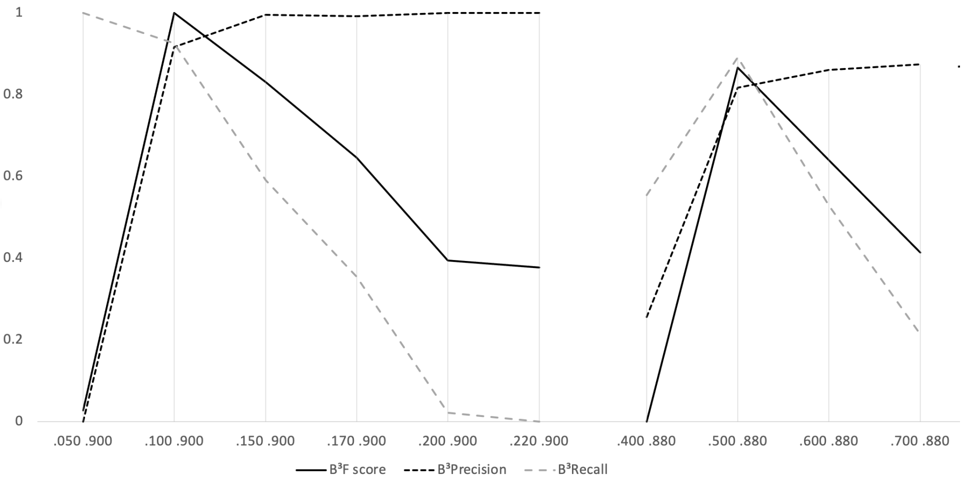


Figure 6.7: Influence of different f_t and ϑ_t values on B^3Precision and B^3Recall when B^3F score $>$ 0.95 and applying TVC clustering on IMDb-Wiki database face images.

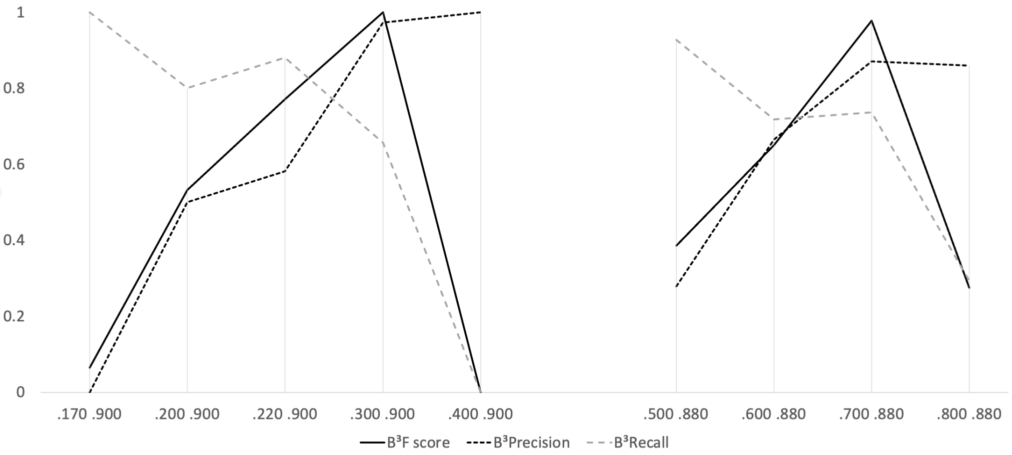


Figure 6.8: Influence of different f_t and ϑ_t values on B^3Precision and B^3Recall when B^3F score $>$ 0.95 and applying TVC clustering on SSIM database face images.

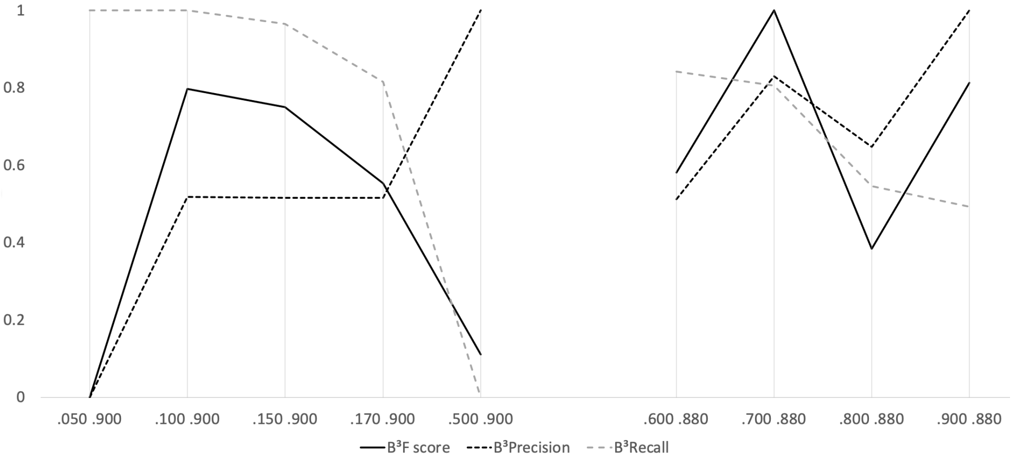


Figure 6.9: Influence of different f_t and ϑ_t values on B^3Precision and B^3Recall when B^3F score $>$ 0.95 and applying TVC clustering on Yale B database face images.

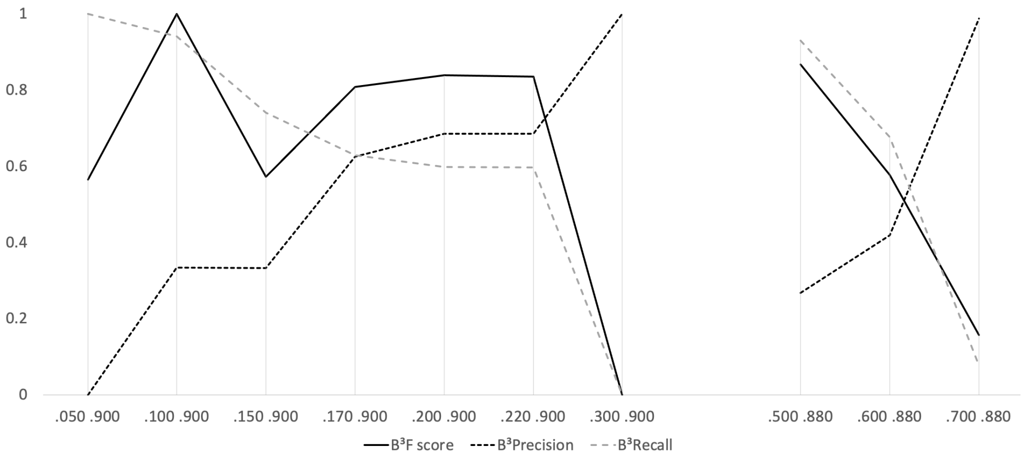
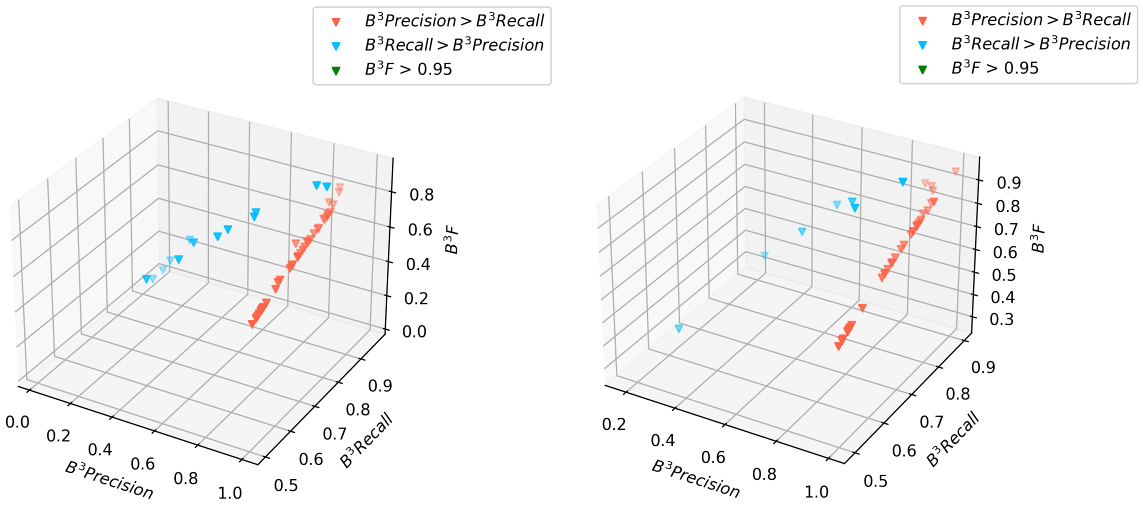


Figure 6.10: Influence of different f_t and ϑ_t values on B^3Precision and B^3Recall when B^3F score $>$ 0.95 and applying TVC clustering on LFW database face images.

It can be observed that the darker dashed line representing $B^3Precision$ increases with increasing frequency value f_t, regardless of the selected similarity threshold ϑ_t, while $B^3Recall$ decreases. Figures also show that among the combinations of input parameters, there are no examples of a similarity threshold $\vartheta_t = .920$. Tab. 6.1 clarifies these results with the numbers of records shown in the database collections with frequency values K_B. The number of records in the K_B is significantly smaller than applying other ϑ_t values, implying insufficient evidence to produce better evaluation results.

YTFaces database contains the most degraded images of all other data sets, which is reflected in the experiment results, i.e., the value of B^3F that does not exceed the limit of 0.95 as in previous sets of face images.



(a) *(b)*

Figure 6.11: Results of an experiment where TVC is applied on images set YTFaces database containing video frames a) shuffled and b) sorted.

Fig. 6.11 shows the result of an experiment with the same data set, whereas in Fig. 6.11a the results of the set with a random order of face images, while on Fig. 6.11b, images were used in the order that is in the YTFaces database. This change in experimental settings is significant because the images/frames from the YTFaces database are sorted in the order they were extracted from the video. Nevertheless, the result of the experiment showed that the distribution pattern is the same and that the difference in the random selection of images and when they are in order is not significant. These results indicate the possibility of using concurrent data set processing, which would potentially shorten the total time of the clustering process. In Tab. 6.2, clustering results obtained by the experiment performed on different data sets using B^3F measure and NMI can be compared. The best B^3F and NMI scores are marked in bold and bold-italic, respectively. Tab. 6.2 is separated into two parts. The first part lists the results of six state-of-the-art clustering algorithms: ResNet-29+ARO, ResNet-29+GGBC (Generalized Graph-Based Clustering), ConPaC, DDC and DDC-NEG. The clustering results in this part of the table are from the corresponding research (marked with asterisks in Tab. 6.2).

The second part of the Tab. 6.2 shows the results of the TVC approach proposed in this thesis with two different examples of the combination of input parameters.

Table 6.2: Results of different face clustering algorithms testing on multiple databases.

	LFW		YTFaces		Yale-B		SSIM		IMDb-Wiki	
	B^3F	NMI	B^3F	NMI	B^3F	NMI	B^3F	NMI	B^3F	NMI
ResNet-29 + ARO *	0.859	-	0.800	-	0.788	-	-	-	-	-
ResNet-29 + GGB *	0.934	-	0.854	-	0.888	-	-	-	-	-
ConPaC [2*]	0.922	-	-	-	-	-	-	-	-	-
PAHC [3*]	0.857	0.958	0.360	0.734	-	-	-	-	-	-
DDC [3*]	0.943	0.988	0.906	0.960	-	-	-	-	-	-
DDC-NEG [3*]	0.955	***0.991***	**0.919**	0.965	-	-	-	-	-	-
TVC 880 (Proposed)	0.955	0.981	0.904	***0.968***	**0.987**	***0.994***	**0.977**	***0.991***	0.968	0.990
TVC 900 (Proposed)	**0.962**	0.983	0.881	0.967	0.976	0.988	0.958	0.984	**0.975**	***0.991***

* (Chang et al., 2019), [2*] (Lin, Chen, Ranjan, et al., 2018), [3*] (Lin, Chen, Castillo, & Chellappa, 2018)

In Tab. 6.2, it can be observed that the TVC with both sets of parameters gives satisfying results. Moreover, the values obtained from both applied evaluation methods show that the TVC gave better results or is in line with other presented state-of-the-art algorithms.

Based on experimental results can be concluded that, in total, TVC 880 achieves the best clustering results. Still, it was slightly better than the TVC 900 combination of input parameters.

Further testing of the TVC approach extended with the IJB-B face image database, which within several of its evaluation protocols contains a protocol intended for testing clustering algorithms (*Test 7*). Tab. 6.3 shows the results of various algorithms applied with subsets IJB-B of the database.

Table 6.3: Results of different face clustering algorithms testing on IJB-B database.

	IJB-B 64		IJB-B 128		IJB-B 512		IJB-B 1024	
	B^3F	NMI	B^3F	NMI	B^3F	NMI	B^3F	NMI
DA-Net [*]	-	-	-	-	0.834	-	0.833	-
ECLIPSE [2*]	0.838	-	0.831	-	0.743	-	0.735	-
PAHC [3*]	-	-	0.695	0.863	-	-	0.639	0.890
DDC [3*]	-	-	0.810	0.918	-	-	0.723	0.913
DDC-NEG [3*]	-	-	0.829	0.927	-	-	0.751	0.922
GCN-A [4*]	-	-	-	-	0.833	0.936	0.833	0.942
RGCN-16 [5*]	-	-	-	-	0.878	0.945	0.885	0.956
ConPaC [6*]	0.772	-	0.769	-	0.656	-	0.641	-
TVC 880 (Proposed)	0.909	0.956	**0.956**	***0.979***	**0.926**	***0.973***	**0.921**	***0.974***
TVC 900 (Proposed)	**0.946**	***0.974***	0.940	0.974	0.911	0.966	0.857	0.939

[*](Guo et al., 2020), [2*](C. Li, Gunther, & Boult, 2018), [3*](Lin, Chen, Castillo, & Chellappa, 2018), [4*](Z. Wang et al., 2019), [5*](Qi et al., 2021), [6*](Y. Shi et al., 2018)

The results shown in the Tab. 6.3 confirm the state-of-the-art quality of the TVC algorithm. In addition, both combinations of input parameters TVC outperform the other algorithms with which the comparison was performed.

6.4 Summary

In order to resolve the identity of the person, all collected evidence must be available to the criminal investigator. The system that assists the criminal investigator in determining the actual identity of suspects and others they used while committing or concealing illegal activities must group all collected data related to suspects. The application of face biometrics in this process reflects everyday police work. However, images of persons collected from forensic, OSINT and other police reports are often created in unconstrained conditions, requiring an appropriate clustering method.

This chapter proposes a new approach of unconstrained pairwise face clustering TVC based on NAL adaptation, primarily the truth-value concept, and its capabilities are tested by experiment. Furthermore, the approach was applied to different sets of databases with face images, thus simulating an unconstrained condition.

These are the main conclusions:

- The proposed TVC algorithm performs clustering without prior knowledge of the number of clusters. Experiments have shown that the proposed clustering approach effectively achieves equal or better results than other state-of-the-art methods. However, the real advantage of the algorithm is in the application within a distributed environment, which will be discussed in the next chapter.

- Growth of processing is half time slower than what quadratic O notation of pairwise clustering suggests, and it is closer to $O(n(n+1)/2)$.
- Choosing larger values of ϑ_t significantly reduces the number of communications with the database, which decisively shortens the total time of the process, but ultimately reduces the cardinality of positive and negative evidence essential in calculating the frequency f.
- By choosing a higher value of the f_t, the system will perform reasoning more conservatively, while with a lower value, it will be more flexible but also more comprehensive.
- When it comes to frames extracted from the video, the experiment results did not vary significantly with pseudo-random against sequential order, which allows the concurrency in the clustering process.
- It is necessary to optimize the approach to clustering by implementing solutions that will make the process much more efficient while preserving effectiveness in future work.

Chapter 7

Multi-Agent System for Advance Policing - mASAP

Police systems are often independently developed across different organizational units, and even when they could be easily integrated with the development of computer networks due to the different levels at which these organizational units operate (for example, federal, state, or regional), and the sensitivity of collected data they remain separated. Increasing demands for personal data protection may cause new systems to be intentionally developed as separate entities to avoid any possibility of gaining insight into available (personal) data for the purpose they were not collected.

Contrary to stricter requirements for protecting personal and sensitive data, there is a need to identify the perpetrators of serious crimes, i.e., to resolve their identities, as soon as possible. It is also necessary to fully use all existing information in this process, which reduces the need to conduct other police activities that, although lawfully, further violate privacy. Connecting the separate independent units of the police system to appear to be a single unit is solved by creating a distributed system.

Distributed systems characterize independent computing elements and the impression that these elements form a single coherent system (Van Steen & Tanenbaum, 2017). The elements that a distributed system consists of, or nodes, can be different hardware, significantly different in specification, software process, and a distributed subsystem that works autonomously concerning the wider distributed whole (closer to the problem of police organizations). Particularly challenging task to connect the nodes so that end-users seem to work with the one system in this thesis is solving with a multi-agent system.

Software agents provide a solution for data exchange in distributed environments based on rules that apply to each node individually or to the extent that those rules allow, making the system functionally coherent.

7.1 mASAP system architecture

System mASAP, proposed in this thesis, enables by design the collection of data and information from various police reports to be grouped based on face clustering and other relations and then presented to criminal investigators in order to resolve suspects' identities. Furthermore, working in a distributed environment mASAP and applying revision, the inference method from NAL allows the knowledge that two face images belong to the same person who exists in different nodes using a node that does not have enough evidence. Fig. 7.1 shows the architecture of the mASAP system.

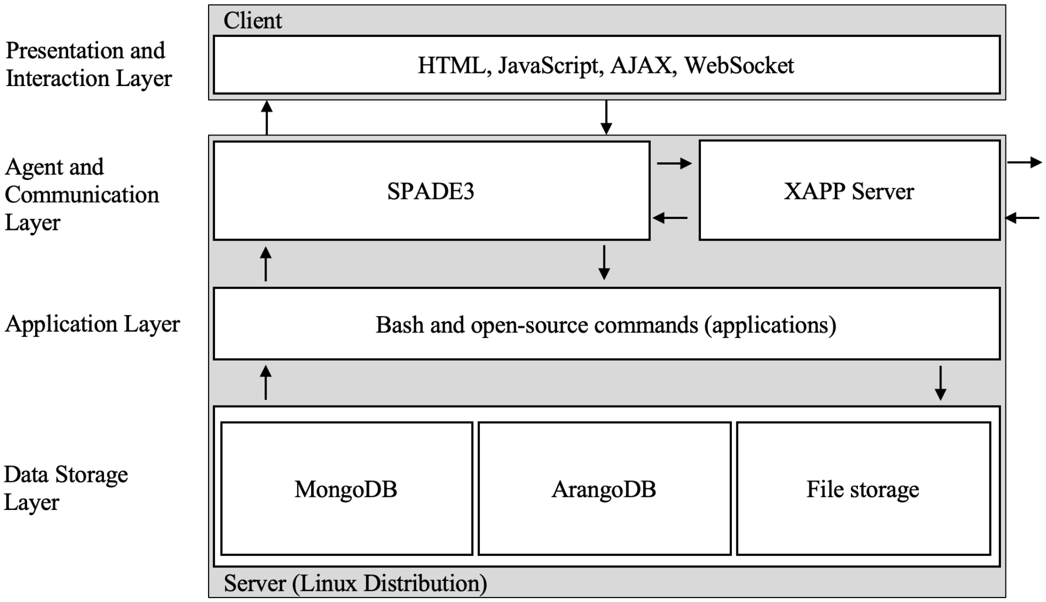

Presentation and Interaction Layer
Agent and Communication Layer
Application Layer
Data Storage Layer
Client
HTML, JavaScript, AJAX, WebSocket
SPADE3
XAPP Server
Bash and open-source commands (applications)
MongoDB
ArangoDB
File storage
Server (Linux Distribution)

Figure 7.1: The architecture of the mASAP system.

The architecture of the mASAP system consists of several layers. The first layer is the Presentation and Interaction Layer. It displays the results of the mASAP, alow interaction of agent and end-users and uploading of reports. Reports are uploaded using the archive file format (preferably ZIP or 7Z). Before uploading user can compress the file to shorten the period of uploading data to the server and reduce the system load. Report is upload to server using HTML elements. The agent informs the user about the progress during data processing using WebSocket technology.

If the user needs to analyze the data during import, the mASAP system enables this with Asynchronous JavaScript And XML (AJAX). Although the name AJAX suggests that XML is used to refresh the content of already loaded pages, which is what this technology is intended for, it can still be other content such as JSON or plain text. AJAX is essentially a combination of XMLHttpRequest (XHR) object which is built-in in the internet browser and serves to connect to the server, and JavaScript Document Object Model (DOM), a model that represents the structure of the website.

These web technologies suggest that there is a corresponding web server within the mASAP system. Since *http.server*, *cgi*, *socket*, and other Python modules will be used for this purpose, in Fig. 7.1 mASAP web server is not particularly emphasized but is implied within the Agent and Communication Layer. Central to this layer is the agent technology implemented through the Smart Python multi-Agent Development Environment (SPADE) platform, which will be discussed in more detail later in this chapter.

The next layer in the mASAP architecture is the Application Layer, in which the critical role is played by Bash, which automates the use and connects the functionality of open-source applications, as well as operating system utilities. Fig. 7.2 shows the relationship between Bash and applications and other vital elements of the operating system architecture.

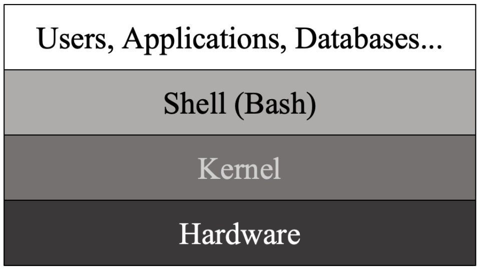


Figure 7.2: The relationship between Bash and applications within the operating system.

Shell is a macro processor that executes commands, a command interpreter, which provides a user interface to a set of operating system utilities, and a programming language (Ramey, 1994). Significant for its use are shell primitives such as reserved words, which make sense to the shell and its program language (for example **if** and **for** flow control constructs), operators such as "|" (vertical line called pipe) or ">" (Ramey, 2011). Everything else would make ordinary words. The main advantage of using shell, primarily by computer developers and scientists, is speed and efficiency over GUI because it allows the use of scripts to automate the processing of a large number of files and data at once (Kidwai et al., 2021).

In this thesis, mASAP system uses the Bash shell. Even more, the expressions such as Bash commands and scripts are also used. Some commands and scripts can execute without problems within different shells, while others cannot. However, all those listed in the thesis by Bash are executing for certainly. When it comes to the compatibility of Bash scripts with other shells, it exists primarily with the Z shell (developed on top of Bash), which is an example of a modern shell with many features. Although those features help users, they do not crucially influence how the shell is used in this thesis. What Z shell has, unlike Bash, is replaced by applications (for example, a Bash does not work with floating-point but **bc** CLI calculator could be used in that purpose). In addition, Bash is still the primary shell in Linux distributions (Kidwai et al., 2021).

The Data Storage Layer consists of MongoDB and ArangoDB databases that play a role in storing information from police reports and for face clustering, respectively. Finally, file storage is the repository for the files from reports that mASAP displaying using web technologies: primarily cropped images of faces and the original images, which represent sources of face images.

All system layers are equally important for the development and undisturbed functioning of mASAP, yet agent technology plays a key role.

7.2 Software agents

Software and intelligent agents are programs that perform tasks on behalf of users or other programs, but further of that, there is no agreed definition of the term "agent" (Obied & Abir, 2021). Different definitions define agents as entities that perceive their environment through sensors, obtain data about events in this environment and act upon it, or as autonomous, reactive, pro-active, and communicative system (Tarassov, 2018). The social ability of agents, from the simple exchange of messages of a predefined format, to more advanced where agents follow appropriate norms to coordinate their actions, cooperate, negotiate or compete, is one of their defining characteristics, especially when it comes to multi-agent systems (Mitrović, 2015). Along with agents, there are actors, "simpler versions" of agents, who are reactive in terms of responding to messages they receive but can also be used for complex simulations, such as perceiving the environment and initiating new actions when their internal state changes (Pal, Leon, Paprzycki, & Ganzha, 2020).

Agents have applications in various areas of life and work. Examples are information retrieval for automatic filtering, intrusion detection in computer networks, solving complex scheduling problems, modeling, and simulation of complex systems such as cities, hospitals, and others (Hillmann, Uhlig, Rodosek, & Rose, 2014). They can also be used for social simulations, mobility, physical entities, environment, and ecosystems simulations, as well as for economic studies, medicine, industry, and the military (Pal et al., 2020). Also, they are solving problems of energy optimization (González-Briones, De La Prieta, Mohamad, Omatu, & Corchado, 2018), have an important role in the research of the control and management of power systems (Melo, Sampaio, Leão, Barroso, & Bezerra, 2019), in transport (Burmeister, Haddadi, & Matylis, 1997; Teodorovic, 2003), construction and robotics, cloud computing, social networks (Dorri, Kanhere, & Jurdak, 2018), and in the learning platform or course management system (Vuković, Kuk, et al., 2021).

When it comes to the use of agents in criminal investigations, the agent can be used: for regulated information exchange of crime investigation data between police forces (Dijkstra, Bex, Prakken, & de Vey Mestdagh, 2005); to create a system for hierarchical cooperative analytics that utilizes various processes such as big data analytics, ontologies and metamodels in order to analyze data from different police sources (Alawneh, Said, & Al-Sharif, 2017); modeling criminal activity or crime patterns, for example spatial and temporal aspects of crime involving multiple offenders and multiple targets in urban areas (P. L. Brantingham, Glässer, Kinney, Singh, & Vajihollahi, 2005; P. Brantingham, Glässer, Jackson, & Vajihollahi, 2009); to model a specific scenario such as street robbery in combination with the expert system (Nowakowski, Jedrzejek, & Dutkiewicz, 2015); for modeling different theories like routine activity theory for example which emphasizes that crime occurs when three elements, which could be represent by agent for analasys and prediction, converge (Caskey, Wasek, & Franz, 2018); then in forensics where they can be used to correlate evidence, process distribution (based on the very nature of the Multi-Agent System (MAS) system) and reduce repetitive tasks in analysis, thereby reducing the number of pieces of evidence that must be personally reviewed by an expert. (Ganesh, 2017).

Intelligent agents are technologies that, as part of the general development of information technology and artificial intelligence, permeate both business and society (Elshan, Zierau, Engel, Janson, & Leimeister, 2022). Agents must achieve complex goals, like learning and reasoning, and adaptively perform effective actions within the environment to be considered intelligent (Dellermann, Ebel, Sollner, & Leimeister, 2019). Similarly, when it comes to AGI, an agent would be considered intelligent if he could adapt to the environment while working with insufficient knowledge and resources (P. Wang, 2007).

The three types of intelligent agent architecture that define their internal organization are most commonly used: reactive, Belief-Desire-Intention (BDI), and hybrid or layered architecture (Mitrović, 2015). Reactive agents can perceive their environment through sensors and react to changes that occur in the environment by performing appropriate actions (Garcia, Rodriguez-Aguilar, Alvarez-Cedillo, & Alvarez-Sanchez, 2019). BDI is an approach to building agents inspired by human reasoning based on three entities, beliefs that represent the informational state of the agent, desires that represent the motivational state of the agent and what the agent wants to achieve, and intentions that represent the deliberative state of the agent, or wich actions the agent used to achieve the goal (KC & Chodorowski, 2019). Hybrid agents combine reactive and reasoning components, thus reacting to environmental changes, but they also make decisions and take actions on their own (Mitrović, 2015). In addition to these architectures, there are other solutions and new trends in applied logic based on the features of human cognition, a process called Cognitive Logic (Tarassov, 2018). It is important to add that the circulation of the term started by Pei Wang, author of NAL a theory that this thesis uses.

When the work of multiple agents is required, then the appropriate infrastructure is needed in order to perform their task.

7.3 Multi-Agent System

Using multiple autonomous and cooperative agents achieves distributed intelligence of systems (Lyu, Fazlirad, & Brennan, 2020). MAS refers to the mechanism used to create goal-oriented agents in a shared environment, which have communication and coordination capabilities, giving efficiency in distributed problem solving (Qasem et al., 2022). It is widely used to develop intelligent distributed systems that manage sensitive data (Calvaresi, Dubovitskaya, Calbimonte, Taveter, & Schumacher, 2018). Except for autonomy and cooperation (communication), MAS characterize complexity produced by decision-making mechanisms, learning, and reasoning, adaptability of agents to a dynamic environment, competitive work of agents, distributed work and mobility of agents, security, and privacy are essential, but also openness by deciding dynamically about MAS participants (Oprea, 2004).

Agents within MAS can function independently, control their internal state and actions, and achieve individual or overall goals in cooperation with other agents. Based on the characteristics of agents MAS can be homogeneous (same characteristics and functionalities of agents) or heterogeneous, static (position and relations of agents do not change) or dynamic, can be triggered by time or events, as well as MAS of first, second, or higher-order based on the number of metrics (parameters about which agents in collaboration should agree) (Qasem et al., 2022). According to the Foundation for Intelligent Physical Agentss (FIPA) MAS platform consists of Agent Management System for registering and deregister agents, Directory Facilitator, indexing services performed by agents, Agent Communication Channel, which facilitates inter-agent communication, and Internal Platform Message Transport, message forwarding services.

The critical moment in the development of agent systems was the Workshop on Distributed Artificial Intelligence, held at the Massachusetts Institute of Technology in June 1980, after which the first attempts to define the central concept of agents domain started to appears (Pal et al., 2020). One of the oldest ways to advance a single agent concept into a multi-agent concept is by using mutual beliefs (Weiss, 1999). Since then, various MAS general or special purpose platforms have been developed, such as Java Agent Development Framework (JADE), GAma Modeling Language (GAMA), EMERALD (established on top of JADE), Cognitive Agent Architecture (Cougaar), ActressMas, AgentScript, Evoplex, MADKIT, MASS, Mesa and many others, some of which have been abandoned over time, while others are constantly evolving, and new versions are being released.

7.3.1 Smart Python multi-Agent Development Environment - SPADE

SPADE is a MAS platform based on the eXtensible Messaging and Presence Protocol (XMPP) and Python (Palanca, Terrasa, Rodriguez, Carrascosa, & Julian, 2021) programming language used to create the mASAP system. The first reason to choose SPADE as the basic platform for developing a multi-agent system is integration with XMPP. XMPP is an XML inspired protocol designed to be open and free, asynchronous, decentralized, secure, extensible (for example, SPADE uses extensions for remote procedure calls between agents and file sharing) and flexible (Palanca et al., 2021). The application of XMPP allows agents to easily participate in group communication (Mitrović, 2015). A feature that emphasizes SPADE, and is used using the XMPP protocol and provides the use of XMPP server is the possibility of an instant presence notification system, i.e., agents can be informed about the presence of another agent or change his status (Palanca et al., 2021).

Another reason to use the SPADE platform is that it is developed in Python. Python is a rapidly evolving programming language used for industrial and academic purposes (it should be noted that in addition to numerous data-centric libraries such as *scikit-learn* and *Pandas*, Python can be easily integrated with popular data science tools such as Jupyter notebooks) (Kazil, Masad, & Crooks, 2020). Python is also extensively used when it comes to machine learning frameworks (Lyu et al., 2020).

SPADE characterize high performance, stability, robustness, strong platform security, and the ability to work on all available operating systems (Qasem et al., 2022). When it comes to scalability SPADE allows the agent to run different processes on the same or different nodes of the distributed system, but also that a large number of agents run asynchronously within the same process in a single node (Palanca et al., 2021).

Within SPADE agents are modeled in terms of behaviors, which is the task that the agent performs (Mitrović, 2015). Each agent establishes a platform link by using a unique Jabber identifier (Qasem et al., 2022) but incorporates its strategy, implemented as a behavior.

SPADE allows interaction with each agent via a real-time web graphical interface and Uniform Resource Locator (URL) provided by the agent.

7.4 The roles of mASAP agents

Agent roles are a newer concept in the development of agents, originating from organizational theory and representing a set of behaviors *R* (Tomičić, Đurić, & Schatten, 2018).

$$R = \{b_1, b_2, b_3, ..., b_n\} \tag{7.4.1}$$

The SPADE platform supports five behavior types: Cyclic, One-Shot, Periodic, Time-Out, and Finite State Machine (Palanca, Terrasa, Julian, & Carrascosa, 2020). The roles represent a combination of these behaviors.

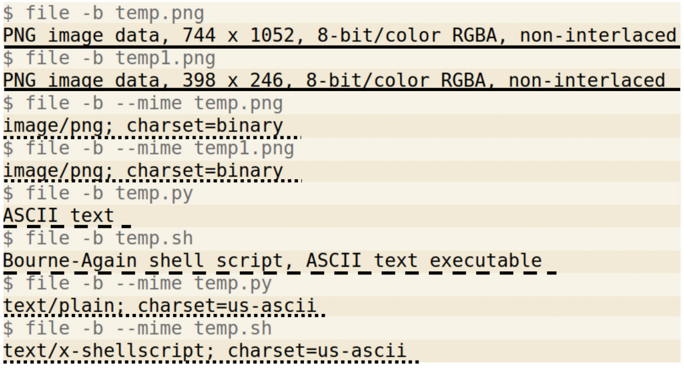
Agents within the mASAP system have a number of roles with often intertwined behaviors. One of the roles is to acquire new knowledge where agents search for vertices in the V_A graph database collection connected via a common vertex but not to each other. After finding such nodes, the values of the frequency f and confidence c are calculated based on Eq. 3.3.25 and 3.3.25, respectively. However, three key roles will be furtherly discussed. These are the (1) data preparation for import, (2) communication, and revision, presenting an extension of the clustering method (system creates new relationships between faces in the knowledge base based on information from other nodes), and finally, (3) grouping and presenting data.

7.4.1 Data preparation

Preparing data for import into the database consists of converting various formats into several that can be searched and presented using web technologies, primarily in internet browsers. To perform the task, the agent must recognize the file in question and perform one or more consecutive conversions using the Bash commands.

Information about the type of file can be obtained based on its extension. The part of the file name (hence the more precise name filename extension) is located behind the dot and usually consists of two to four characters. However, this is only relevant for the Windows operating system, which uses extensions to execute files, while for other operating systems, it does not play a crucial role. However, it should be emphasized that this is not mandatory for Windows either, mostly for user-run files, while system and application files do not have to have extensions.

Another way for an agent to know which file is in question is to use the Bash command **file**. For example, Fig. 7.3 shows the use of the **file** command on four files, two of which are with the same extension, PNG, and two with different ones, Bash script with SH extension and Python script with extension PY.



```
$ file -b temp.png
PNG image data, 744 x 1052, 8-bit/color RGBA, non-interlaced
$ file -b temp1.png
PNG image data, 398 x 246, 8-bit/color RGBA, non-interlaced
$ file -b --mime temp.png
image/png; charset=binary
$ file -b --mime temp1.png
image/png; charset=binary
$ file -b temp.py
ASCII text
$ file -b temp.sh
Bourne-Again shell script, ASCII text executable
$ file -b --mime temp.py
text/plain; charset=us-ascii
$ file -b --mime temp.sh
text/x-shellscript; charset=us-ascii
```

Figure 7.3: The relationship between file extensions and different file type information.

The command **file** is in the example in Fig. 7.3 applied in two ways, with and without the –mime option. Without using the –mime option, the command displays file types in a human-readable format (the results are marked on the figures with a solid line for images and a dashed line for scripts). Using the –mime option results in Multipurpose Internet Mail Extensions (MIME) file types, in Fig. 7.3 in all four cases are marked with dotted lines. By observing all eight results of the command **file**, it could be concluded that although the output when the –mime option is not used is clearer to the reader when deciding on the file type, the use of options is much more helpful. Firstly, the option helps with PNG files because the different outputs it gives in a human-readable format become uniform by applying the option and notifying that it is an image. In the second case, the outputs differ. However, they have in common the indication that there are text files, which is essential information concerning both files. Web pages can display the content of all four files considering their types. PNG is a format that can be displayed, but this does not apply to all files with MIME *image* type). At the same time, the contents of both scripts can be searched, and they are therefore ready for import. Tab. 7.1 shows a set of files and their MIME content types.

Table 7.1: Results of using the ***file*** *command with the –mime option on a set of files, in original and processed form.*

File ext.	**file -b –mime-type "file_path"**	**sed "s/\/ /g; s/[[:punct:]]/ /g; s/ \+/ /g; s/[[:space:]]/,/g"**
/T	**inode/x-empty**	inode,x,empty
7Z	application/x-7z-compressed	application,x,7z,compressed
AVI	video/x-msvideo	video,x,msvideo
DOCX	application/vnd.openxmlformats-officedocument. wordprocessingml.document	application,vnd,openxmlformats,officedocument, wordprocessingml,document
JPG	image/jpeg	image,jpeg
MP4	video/mp4	video,mp4
ODP	application/vnd.oasis.opendocument.presentation	application,vnd,oasis,opendocument,presentation
ODT	application/vnd.oasis.opendocument.text	application,vnd,oasis,opendocument,text
PDF	application/pdf	application,pdf
PNG	image/png	image,png
PPTX	application/octet-stream	application,octet,stream
PSD	image/vnd.adobe.photoshop	image,vnd,adobe,photoshop
PY	**inode/x-empty**	inode,x,empty
PY	text/plain	text,plain
SH	text/x-shellscript	text,x,shellscript
SQLITE	application/x-sqlite3	application,x,sqlite3
SVG	image/svg+xml	image,svg,xml
TXT	text/plain	text,plain,utf,8
VCF	text/vcard	text,vcard
XLS	application/vnd.ms-excel	application,vnd,ms,excel

Tab. 7.1 contains extensions in the first column, except for the first file that does not have an extension, so the name is displayed. The next two columns show the MIME content types obtained using the **file** command and the –mime-type option. The –mime-type option only displays the content type, while the encoding omits. The table shows the diversity of types, some of which are vaguer, while others speak clearly about the content. Unlike the type of content or general categories, such as

text, *image*, and *video*, the *application* category does not group data. In case when a general category is an *application*, the subtype of content can be the most diverse, from the database to the document for word processing, tables, and more. Due to the variety of categories, defining the conversion process for each type is time-consuming work, so in mASAP, it is chosen that the agent learns which type to convert in which way.

Algorithm 4 Creating an agent knowledge base.

```
    Input: $B_C \leftarrow \{(C_1, O_1), (C_2, O_2), ..., (C_n, O_n)\}$ - list of bash commands, $F \leftarrow \{F_1, F_2, ..., F_n\}$ - files from report
    Output: $K_A$- populated agent knowledge database

procedure RECOLLECTION($K_A, mime\_type$)
    $j \leftarrow 0$
    for $j < | K_A |$ do
        similarity=$\frac{| K_A[j][0] \cap mime\_type |}{| K_A[j][0] \cup mime\_type |}$
        if $similarity == 1$ then
            return $K_A[j][1]$
        end if
    end for
    return -1
end procedure
$K_A \leftarrow \{\}$
for $i < | F |$ do
    $mime\_type \leftarrow subprocessing($"file -b –mime-type $F[i]$ | sed ..."$)$
    switch $mime\_type[0]$ do
        case $text$
            $import(subprocessing($"cat $F[i]$"$))$
        case $image$
            $import(subprocessing($"file_name=$(basename $F[i]$); convert $F[i]$ $file_name'.png'"$))$
    if $mime\_type[1] ==$ "$empty$" then
        continue
    end if
    $k \leftarrow Recollection(K_A, mime\_type)$
    if $k == -1$ then
        for $z < | B_C |$ do
            $converted\_data \leftarrow subprocessing(B_C[z][0])$
            $output\_check \leftarrow subprocessing($"if test -f $B_C[z][1]$ ; then if [ -s $F[z][1]$ ]; then echo ok; fi; fi"$)$
            if $output\_check ==$ "$ok$" then
                $import(converted\_data)$
                $K_A \cup (mime\_type, z)$
                break
            else
                continue
            end if
        end for
    else
        $converted\_data \leftarrow subprocessing(B_C[k][0])$
        $import(converted\_data)$
    end if
end for
```

Two MIME content type, in Tab. 7.1 are shown in bold, indicating that these are empty files.

Information about empty files allows an agent not to spend resources trying to process such files but to continue moving on to the next file. The last column contains the processed data on the type used by the agent in the learning process. Alg. 4 describes the agent learning process in more detail.

The agent uses subprocesses, on the way shown in Chapter 4, to manage Bash commands. At the input of Alg. 4, there is a finite set of ordered pairs of Bash commands with corresponding parameters and options, as well as the expected output. In addition to the command set, there is also a set of report files. The report can be a single PDF file or a Microsoft Word file. However, here the assumption is a more complex situation, such as a forensic report with many exported files. The algorithm uses the **file** command to determine the MIME content type, which then is processed, and result is a list (third column in Tab. 7.1). The list is checked whether there are general categories that are always treated the same, namely *text* and *image*, or whether the file is empty so the process can continue to the next file. An easier way to compare categories and subcategories is one of the reasons why the MIME type is processed and essentially converted into a sequence. After these initial checks, the knowledge base K_A is consulted. Suppose the type exists in the knowledge base. In that case, the ordinal number of the element in the list Bash of the command B_C is immediately obtained. It is used for a particular command retrieval from the B_C, which is then executed using a subprocess. If the type is not in the knowledge base, the algorithm starts from the first command in the B_C and tries with everyone in the set, checking to see if the expected output is obtained.

The check is performed using the **test** command, which determines whether the file expected in the output exists and has content. Successful testing means that this is a new item in the agent's knowledge base K_A, while failure leads to the next attempt. The implementation is more complex than the algorithm suggests, as can be seen in the List. 7.1.

Listing 7.1: An example of code for implementation of the agent learning process

```
import subprocess
...
BC = [
(f"pdftoppm -png ’{file_name}’.pdf ’{file_name}’;
for f in ’{file_name}*.png’; do convert $f -normalize -unsharp 0x5 $f; done;
python3 face_clustering.py ’{file_name}’ &;
pdftotext {file_name}
",
f"result=’not ok’;
find . -name *.png; while read f_name; do if test -f $f_name then if [ -s
    $f_name ]; then result=’ok’; else result=’not ok’;fi;else result=’not ok’
    fi;
echo $result
"),...]
...
    resp = subprocess.run(BC[0][0], capture_output=True, shell=True)
    resp = subprocess.run(BC[0][1], capture_output=True, shell=True)
if(ret.stdout.decode()=="ok")
    ...
```

List. 7.1 shows one example of a command from the set B_C and an expected result, i.e., a test that checks whether what was desired has been achieved. The example shows that in the case of a PDF file, it is handled in two ways: (1) images are created from each page to detect faces, if any, then the clustering process starts, and (2) as well as extraction text from the document for import into the database and further search. After the command from B_C is executed, and it is more of a set of commands, not just one, the agent checks whether all the expected files have been created. The text file is not mandatory if it is a PDF file with scanned content. Hence, the agent only tests if there are images because an image must exist if there is even one page of the document. If non of the image files exist, it is not a regular PDF file. The value on standard output (stdout) will show the $result variable's initial value, which is 'not ok,' after which the agent learning process continues. It should be noted that one of the commands when working with a PDF document is **convert** whose purpose is to enhance images before detection and clustering.

The data preparation agent is also responsible for face detection, feature vector extraction, and TVC face clustering.

7.4.2 Communication and revision

The data for clustering collected by agents of mASAP system helps nodes of a distributed system to determine more accurately the relationships between images of persons they already possess and group data related to them, rather than retrieving personal data from other nodes in the distributed system. Fig. 7.4 shows the distributed system model and how data is collected based on two feature vectors.

Data request, which consists of two feature vectors and their hash values, is sent in the first step by the mASAP from the node @ *masap1* to the other nodes of the distributed system (shown in Fig. 7.4). In this way, each node agent that receives the message can decide whether to perform the entire TVC clustering procedure based on the vector or, due to the current resource usage, try to respond only based on the hash value in the second step.

Besides restrictions that create rules, such as which data may be shared between agent systems or agents, there are restrictions on making the resources of one node available to other elements of the distributed system. In step two, the selection of the input data, depending on the load of the agent system that received the request, affects the selection of the appropriate time complexity. The time complexity for using feature vectors according to O notation, in this case, is $O(2\ n^2)$, i.e., it is necessary to match two feature vectors with all others in the system that received the request. In the case of using hash values, complexity is $O(2\ log\ n)$, which is significantly more efficient because it only checks whether there are edges in the knowledge base K_B with the corresponding vertices at its end. Opposite situations are demonstrated with the stated complexities, i.e., maximum and minimum consuming time and resources. By using a hash value, each node can start the process and then move on to the more demanding use of the feature vector. A limitation for hash values is that the corresponding feature vector of the faces, which mutual relation has to be discovered, must exist in the system that received the request. There is also a situation where one image exists in the system, and all pairwise matching is already done, while only for the other feature vector similarities need to be calculated for all other face descriptors in the knowledge base K_B. Still, Tab. 6.1 has shown that in intensive communication with databases, these savings are significant.

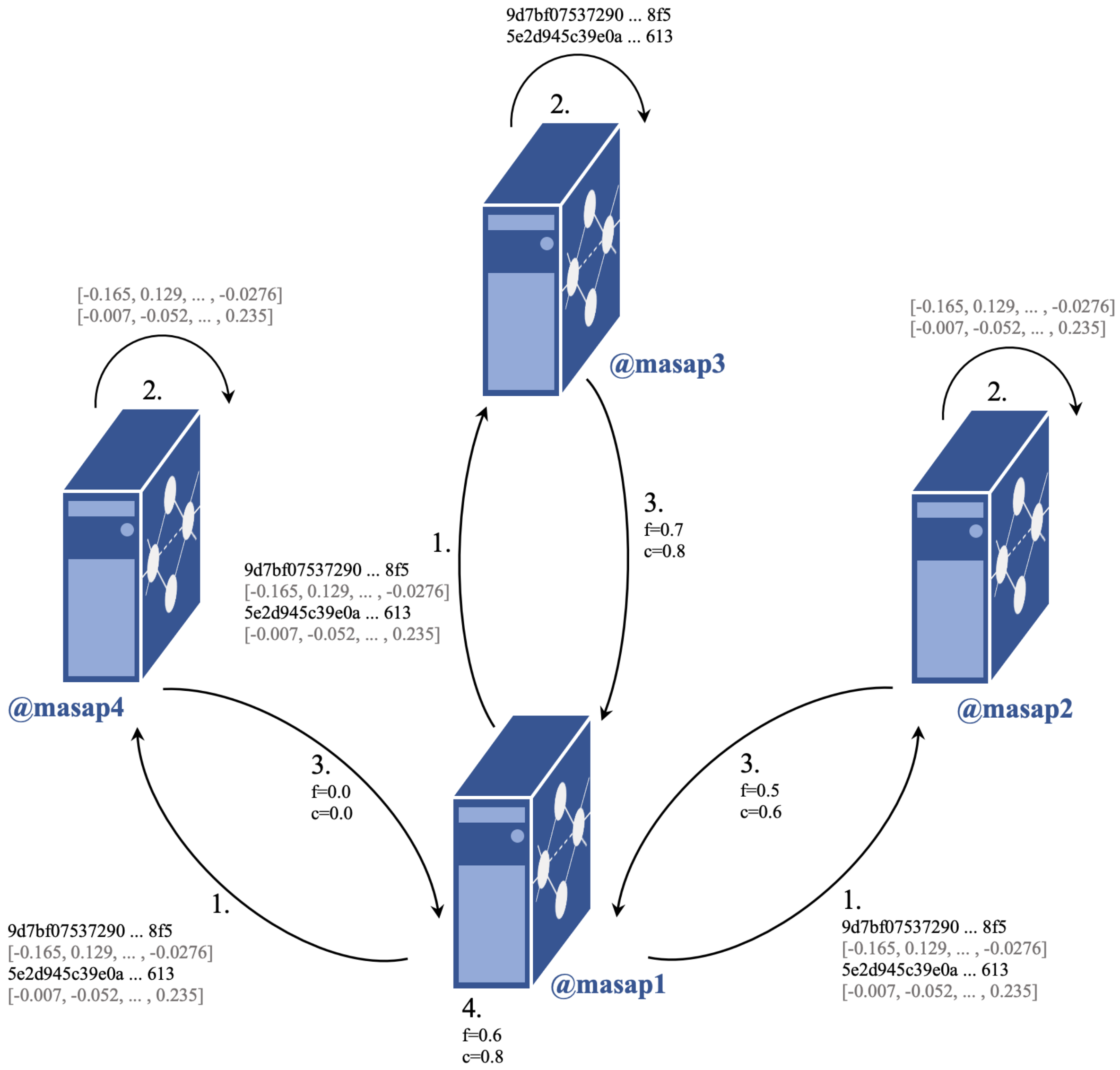


Figure 7.4: Collecting truth-values from other nodes of the distributed system.

It should be noted that the hash value of the feature vector could be calculated by the agent receiving the request without submitting it separately. However, in the example of Fig. 7.4, the idea is to demonstrate also one possibility of the agent system which requests assistance. It can send only hash values and choose how all others will respond if a prompt response or additional personal data protection is required. Following the protection of personal data for certain persons, in the third step in Fig. 7.4, it can be observed that one of the agent systems returns zero values for both the frequency f and the confidence c. Suppose the instance of mASAP does not want to reveal the existence of data on specific persons, it can return the same values as the system that does not contain data. In this particular case, the one node (@masap4) which received the request chose to work with feature vectors because, at that time, it did not know which person it was. After the process of TVC clustering and determining that it is a person whose data are restrictive, the system sends a regular response that does not indicate some restriction.

The other two agent systems (shown in Fig. 7.4) return truth-values so that the requesting agent system can retrieve the resulting values in step four. The resulting value showed that a pair of feature vectors belonged to the same person, although there was not enough evidence in the similarity collection K_S of the local system to calculate frequency and confidence values.

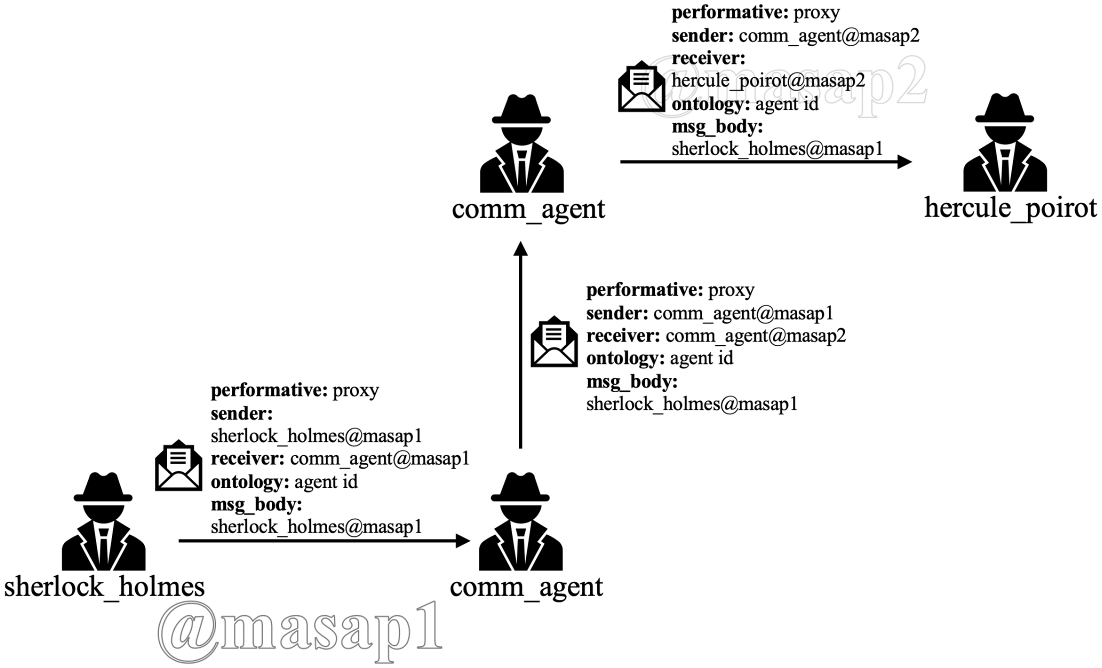
List. 7.2 shows how the frequency f and confidence c are calculated based on the truth-values provided. Using Python, two lambda functions implementing Eq. 3.3.27 and Eq. 3.3.28 and a sequence of truth-value tuples are sufficient.

Listing 7.2: Calculating the resulting truth-value

```
f= lambda f1,c1,f2,c2 : 0.0 if c1==0 and c2==0  else (f1*c1*(1-c2)+f2*c2*(1-c1))/(c1*(1-c2)+c2*(1-c1))
c= lambda f1,c1,f2,c2: (c1*(1-c2)+c2*(1-c1))/(c1*(1-c2)+c2*(1-c1)+(1-c1)*(1-c2))
truth_values_received=[(.0,.0),(.7,.8),(.5,.6)]
f_resulting_value=0      #there was not enough evidence to calculate the frequency for the pair of feature vector
c_resulting_value=0     #there was not enough evidence to calculate the confidence for the pair of feature vector
for truth_value in truth_values_received:
    f_resulting_value=round(f(truth_value[0],truth_value[1],f_resulting_value,c_resulting_value),1)
    c_resulting_value=round(c(truth_value[0],truth_value[1],f_resulting_value,c_resulting_value),1)
```

In Fig. 7.4 from the perspective of agents, an example of federated architecture is presented where several multiagent systems communicate with each other, for which facilitators are usually used (Ngolah & Far, n.d.). Agents communicate with each other (1) with the help of a blackboard, where information is available to everyone in a shared information space, (2) by direct messaging, and (3) with facilitation. However, since SPADE uses XMPP, which provides a federated, open server architecture with which an agent can communicate with any other agent, artifact or human can be said in the world (Palanca et al., 2020), for mASAP the solution with direct messaging was chosen.

On Fig. 7.5 - 7.7 are presented the way agents communicate within the federated architect of the mASAP system. Fig. 7.5 shows initiating communication by sending a message from an agent who needs assistance to a communication agent from the local mASAP. The message contains a FIPA "proxy" performative that signals that the sender expects the recipient to select target agents and forward the message to them. Then, the communication agent forwards the message to the communication agents in the other mASAP nodes of the distributed system, who pass the message on to free agents from their systems.



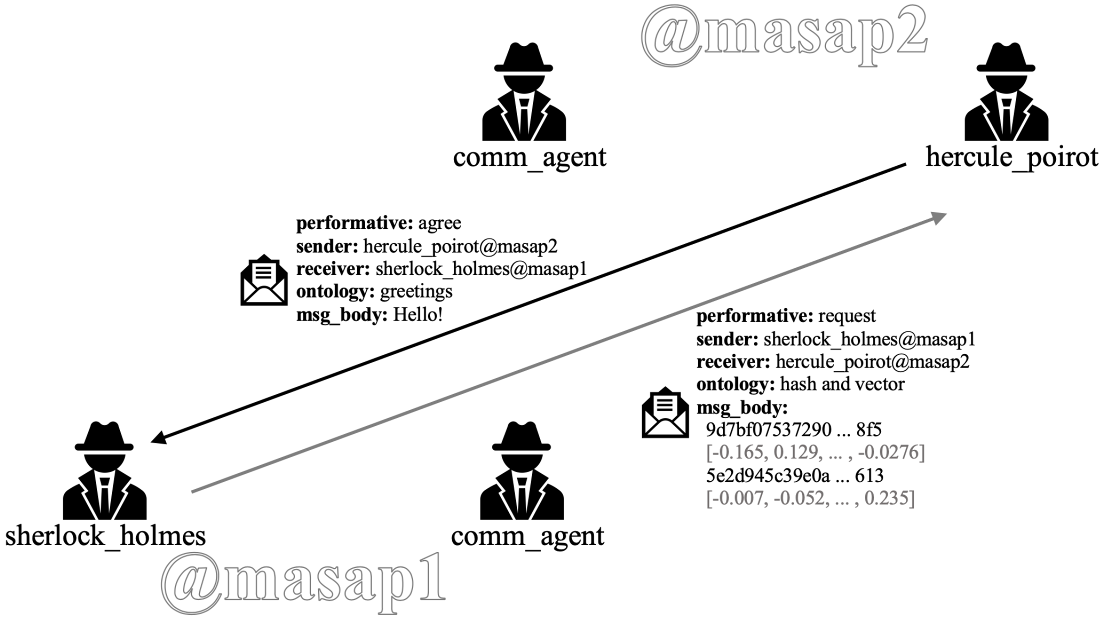
performative: proxy
sender: comm_agent@masap2
receiver: hercule_poirot@masap2
ontology: agent id
msg_body: sherlock_holmes@masap1

performative: proxy
sender: comm_agent@masap1
receiver: comm_agent@masap2
ontology: agent id
msg_body: sherlock_holmes@masap1

performative: proxy
sender: sherlock_holmes@masap1
receiver: comm_agent@masap1
ontology: agent id
msg_body: sherlock_holmes@masap1

Figure 7.5: Initiating communication between agents of two different mASAP.

The message structure exchanged by the mASAP system agents is based on the FIPA Agent Communication Language (ACL) specification where the mandatory parameter is performative, while the sender, receiver, and message content imply (*FIPA ACL Message Structure Specification*, 2022). The presence of other parameters is not mandatory and depends on the situation. Performative denotes the type of communicative act. Alonge with the "proxy" here will be used "agree," "request," and "inform." In addition to the mandatory parameters, the ontology "used to give a meaning to the symbols in the content expression" (*FIPA ACL Message Structure Specification*, 2022) is also used.



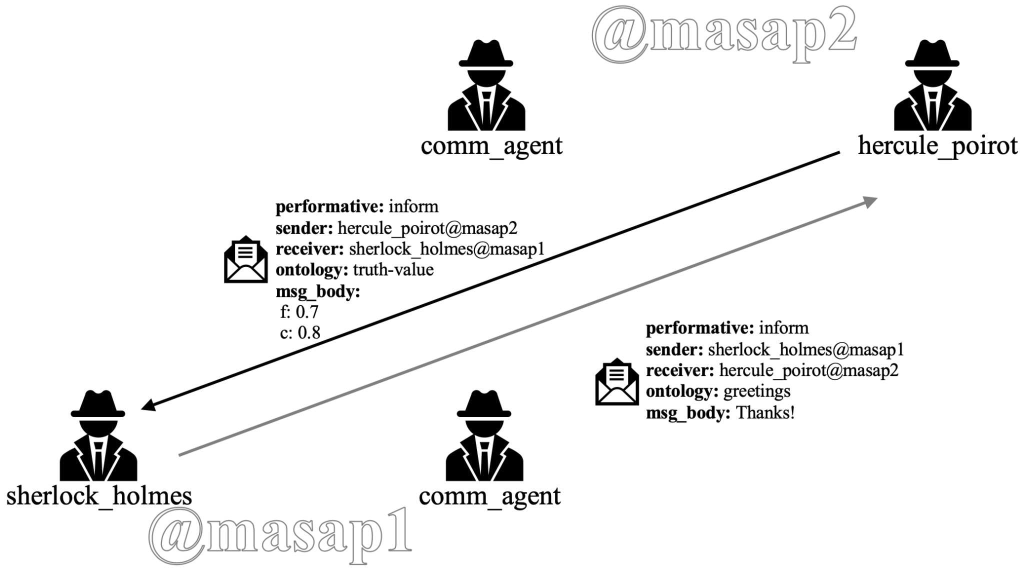
performative: agree
sender: hercule_poirot@masap2
receiver: sherlock_holmes@masap1
ontology: greetings
msg_body: Hello!

performative: request
sender: sherlock_holmes@masap1
receiver: hercule_poirot@masap2
ontology: hash and vector
msg_body:
9d7bf07537290 ... 8f5
[-0.165, 0.129, ... , -0.0276]
5e2d945c39e0a ... 613
[-0.007, -0.052, ... , 0.235]

Figure 7.6: Establishing communication between agents of two different mASAP and sending requests.

When the agent receives the initial message, it retrieves the address of the agent, Jabber Identity (JID), who needs assistance from the message body and sends a response using the performative "agree" immediately, which agrees in advance with potential requests. In this way, communication is established between agents who continue to exchange messages directly, the first of which contains a request with hash values and/or feature vectors (shown in Fig. 7.6).

The agent who received the request then acts in one of the previously discussed ways, selects the appropriate time complexity, and sends the response back, after which the agent who requested assistance can terminate the communication (shown in Fig. 7.7).



performative: inform
sender: hercule_poirot@masap2
receiver: sherlock_holmes@masap1
ontology: truth-value
msg_body:
f: 0.7
c: 0.8

performative: inform
sender: sherlock_holmes@masap1
receiver: hercule_poirot@masap2
ontology: greetings
msg_body: Thanks!

Figure 7.7: An agent receives responses from other mASAP node with truth-value and ends direct communication.

Communication between agents does not have to be finished immediately upon receiving the response, as suggested by Fig. 7.7. Instead, the same agents can communicate during the whole process of the TVC clustering. As present in recent figures, the establishment of communication, cooperation, and then completion of tasks essentially demonstrate the dynamic creation of teams composed of agents from different agent systems. The List. 7.3 shows a part of the Python code which, based on the SPADE documentation (*Agent communications*, 2021), implements sending one message from the example in the image 7.7.

Listing 7.3: An example of code implementing sending a message by an agent via the SPADE platform

```python
from spade.agent import Agent
from spade.behaviour import OneShotBehaviour
from spade.message import Message
class SenderAgent(Agent):
    class InformBehav(OneShotBehaviour):
        async def run(self):
            msg = Message(to="sherloc_holmes@masap1")
            msg.set_metadata("performative", "inform")
            msg.set_metadata("ontology", "truth-value")
            msg.body = "(0.7,0.8)"
            await self.send(msg)
            ...
```

An agent that receives a message using the built-in function *eval()* converts a string that looks like a tuple to a tuple and adds it as a new element of the sequence, using it to compute the resulting truth-value. The feature vector on the receiving side is converted similarly, after which it can be further used. The application of *eval()* does not affect the complexity but only further handling of the vectors.

It should be emphasized that only truth-value is returned as a message in the presented examples of communication. In the implementation of this mechanism in mASAP, which extends TVC clustering, hash values are returning too. Messages sent by assisting agents do not arrive simultaneously nor immediately after the request, so a hash value is used to determine which pair of face descriptors their content refers to. There are several ways in which a replay message can be identified using FIPA ACL parameters, such as *conversation-id* or *replay-with*, but this would mean introducing new additional values and their handling, which are not relevant to the process.

7.4.3 Grouping and presentation of data

In order to resolve the identity, the system must present to the criminal investigator all the information related to the suspects. The agent's strategy while working on grouping data and presenting them depends on how the query was made to the system. Two scenarios are considered here, the first when it comes to keyword search and the second when inserting images with faces as a query. Fig. 7.8 shows a graphical representation of data and information processed by the system mASAP. The image shows four clusters of faces and their connections to sources: images and videos. The following relation is different police reports in the form of a single document or a directory that contains many files with different types of data (tables, presentations, SQLite database files). Fig. 7.8 displays six reports, two forensic, one police, one analytical, one OSINT, and one made by a physical surveillance unit.

In the first scenario, the forensic investigator enters the keywords for the search. The search is performed through the MongoDB database, and terms are found in two reports, police and analytical. The agent then searches for data in ArangoDB in vertices collection V_A for those who belong to particular cases. After founding vertices agent search for clusters in which there are individuals only who belong to the case and prepare a presentation of these clusters.

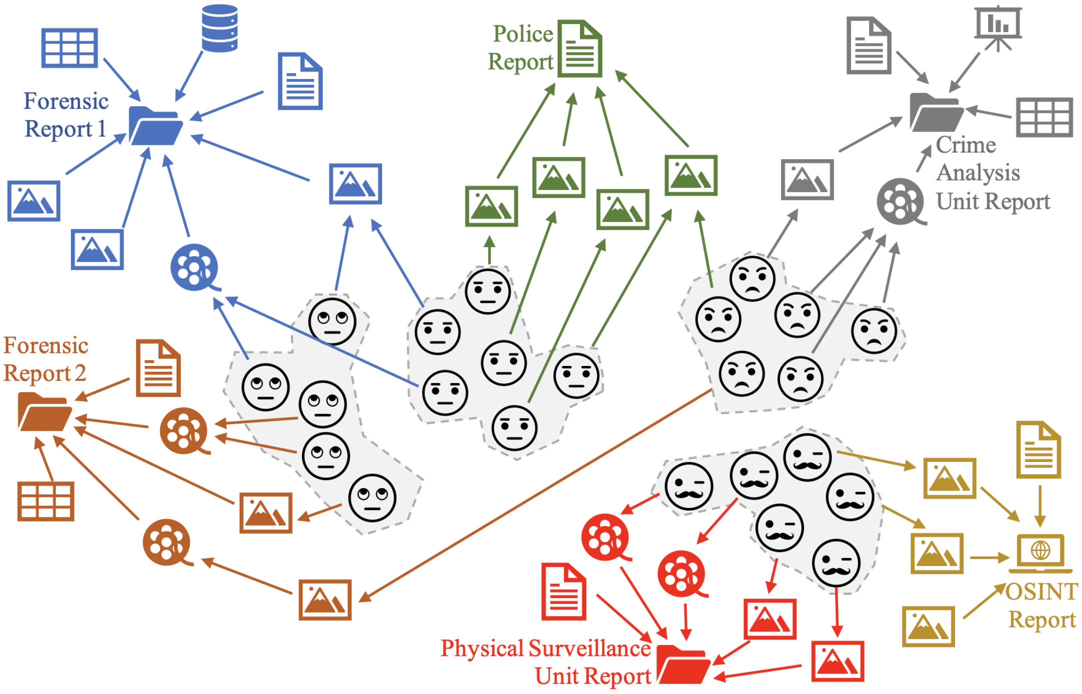


Figure 7.8: Representation of the set of data collected and processed by the mASAP and their relation.

After the cluster presentation, pairs of faces are shown according to the relation "on the same photo." Next is the presentation of pictures and video frames on which the faces from the cluster originated (sources of face images). The order of the presented sources is first photos with two or more faces in clusters, and then others. The analytical space available to the forensic analyst (for review and search) will be limited to the data from the report in which the keywords are found (shown in Fig. 7.9a).

At this stage, the user has an insight into all detected faces and the opportunity to correct the TVC clustering mechanism.

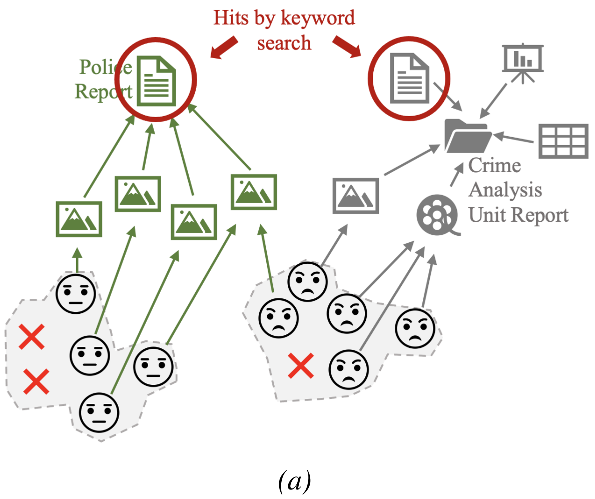


Figure 7.9: Grouping of data based on keyword search with limitation of analysis space only on data from a) reports in which query hit was found and b) with extension to all members of the face cluster.

Fig. 7.9b shows the following extension of the analytical space to all persons belonging to the clusters together with the faces of individuals from the initial reports. It is a step taken if not enough has been learned from the report files directly related to keyword search. In this case, three more face images are available to the criminal investigator.

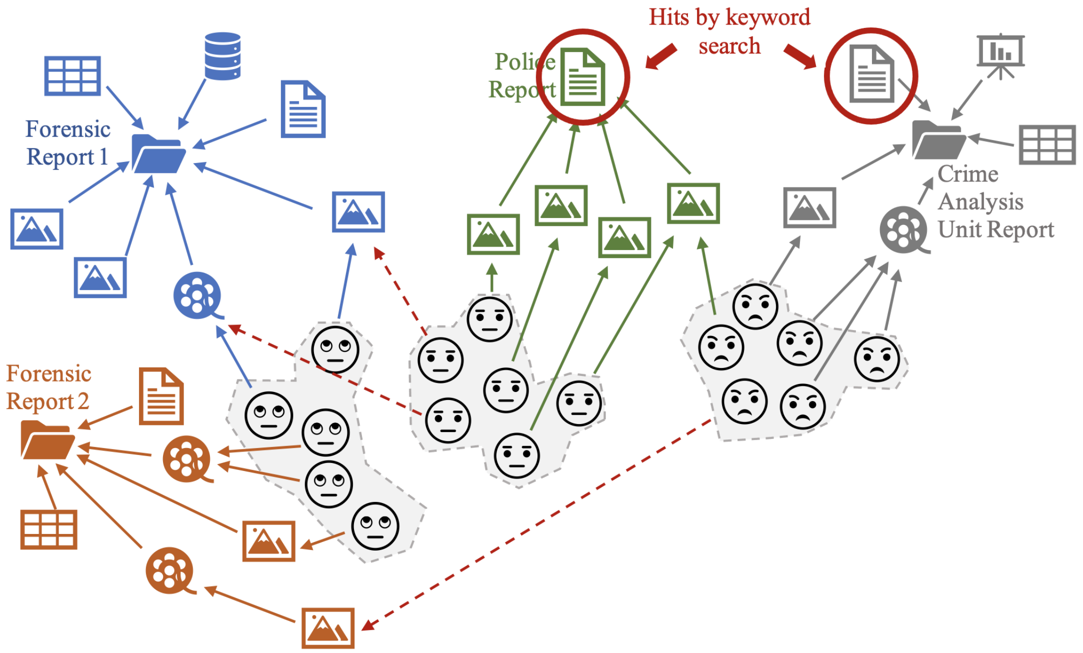
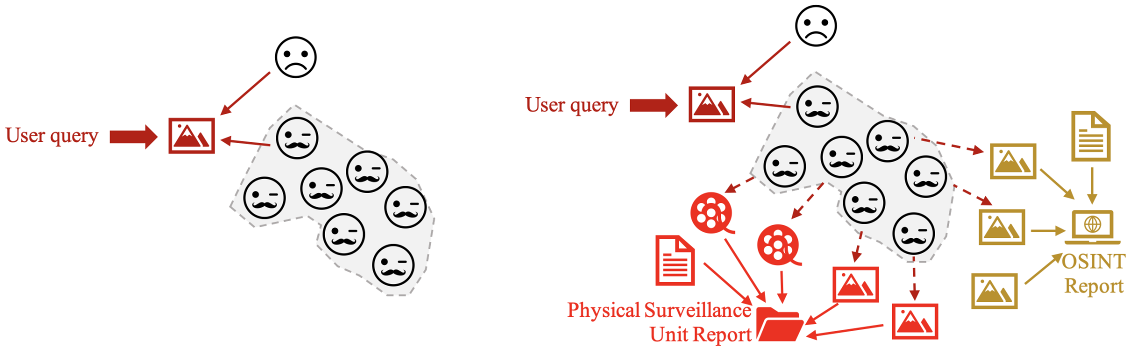


Figure 7.10: Grouping of data based on keyword search with an extension of analysis space to all data for which there are mutual relations in the mASAP.

The last step allows a criminal investigator to analyze data from the four police reports, as shown in 7.10. The faces from the cluster were found in images and video frames with files belonging to other than initial reports. However, two more reports from the whole set shown in Fig. 7.8 remain unavailable because there are no relations. In this way, the analytical space is maximally expanded, but only for data that have the specified relationships with the direct results of the keywords search.

The following scenario refers to a search that begins with a photograph of the suspects. The system detected two faces, one of which has no known relation in the system, while for the other, there is a cluster to which it belongs (shown in Fig. 7.11). In Fig. 7.11a, the first step is shown where only face images from the cluster are shown to the criminal investigator.

The expansion of the analytical space is enabled by the relations of other persons in the cluster (shown in Fig. 7.11b). Sources of images of faces (images and video frames) are available to the criminal investigator and other files in the reports to which the sources belong.



(a)

(b)

Figure 7.11: Grouping of data based on a query containing an image with a) limited space for analysis to the cluster to which one of the detected faces belongs and b) extension to all data related to other cluster members.

The links between the data in the system can also influence the decision made by an agent on how the data will be presented. The importance of visual data is at the forefront, while other data can be further presented according to different criteria. Within mASAP, two strategies are proposed (shown in Figs. 7.12).

Strategies suggest a way of grouping data from a narrower set, such as a cluster of face images from a particular report, to data from all report files that relate to the initial face cluster (shown in Fig. 7.12a). The choice of a criminal investigator can also be to request from the agent to reduce analysis space only to images of faces and their sources, based on all existing relations (from initial and all other connected reports). Although gaining insight into a set of images is faster than reading and searching documents and other files, the agent still needs to complete extensive work to achieve the requested presentation. So, for the agent, this is not a primary strategy.

In Fig. 7.12b, a sorting strategy is presented where data and information are ordered according to specific criteria within each of the different analysis spaces. For example, suppose only the clusters from the report in which the keyword was found are presented (case shown in Fig. 7.9 and Fig. 7.10), the cluster with the most images of the person from the report will be displayed first. Reports are used as frameworks beyond which additional information is required if necessary and only if there is a relationship with the data from the initial keyword search result.

Grouping data provided by mASAP also allow the exclusion of specific evidence. For example, if police reports contain images of children or persons not related to the case, in other words, incidental findings, all files related to them can be excluded by selecting a particular cluster.

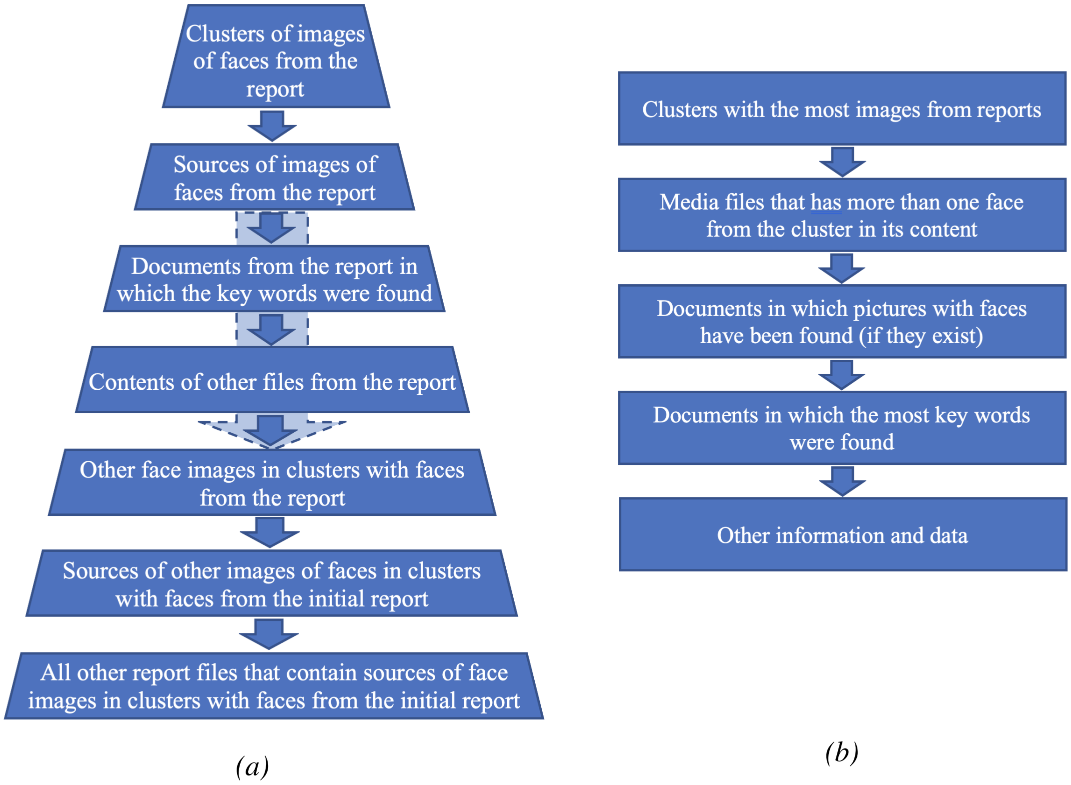

Clusters of images of faces from the report
Sources of images of faces from the report
Documents from the report in which the key words were found
Contents of other files from the report
Other face images in clusters with faces from the report
Sources of other images of faces in clusters with faces from the initial report
All other report files that contain sources of face images in clusters with faces from the initial report
(a)
Clusters with the most images from reports
Media files that has more than one face from the cluster in its content
Documents in which pictures with faces have been found (if they exist)
Documents in which the most key words were found
Other information and data
(b)

Figure 7.12: Model of a) data and information presentation and b) sorting strategy.

7.5 Conclusion remarks

For the needs of identity resolution in this chapter, a mASAP model is proposed, using various types of open-source solutions, from operating systems to shell and shell scripts, commands, databases, and multi-agent platforms, and off-the-shell solutions related to face recognition, for building an advanced system for processing large amounts of data and facilitates decision-making by criminal investigators. The system consists of four layers: Presentation and Interaction Layer, Agent and Communication Layer, Application Layer, and Data Storage Layer. The Agent and Communication Layer is particularly significant, and in this chapter, the most prominent roles of agents are presented.

The data preparation agent integrates the functionality of open-source applications using what is already available to accomplish his task. Based on the MIME content type, the agent can learn which files to process with which set of commands. Selecting a particular agent for a particular type of report allows its knowledge base not to be oversaturated, resulting in more efficient work. For example, police reports on financial crime contain many documents and spreadsheets, reports on forgeries, on the other hand, have more vector and raster graphics, while those on child pornography have a large number of images and videos. An agent who specializes would have a knowledge base to communicate more efficiently than an agent who imports all types of reports.

It should be noted that the data preparation agent cannot be part of the SPADE platform but must engage its extensions, which it periodically monitors to inform the end-user and follow progress. Parallel processing provided by the operating system is still more efficient and reliable than parallel work provided by the application, as demonstrated by the use of a data preparation agent in conducted TVC clustering experiments (while an agent from the SPADE platform would crash due to an error, an agent who works off the platform was processing data continuously for weeks without any problems). The workload of the agent can be seen in Tab. 6.1 in the previous chapter.

The ability to communicate using the XMPP server allows extension TVC clustering using the revision mechanism. The agent assigned for communication and revision provides nodes of the distributed system that do not have enough evidence with the new knowledge about the relation between two faces with appropriate values of frequency and confidence. Another way to provide new knowledge is the work of an agent who, by applying the deduction mechanism, creates direct relations between two vertices that are otherwise only indirectly connected to other vertices.

Data preparation and the clustering process requires the most resources and time. However, the key to the identity resolution process is grouping data and their presentation. There are various analytical spaces that a data grouping and presentation agent can provide to a criminal investigator. In this way, the end-user could be overburdened with an excessive amount of data or with unnecessary data. There is also a possibility of violation of data privacy. To avoid issues of too much data, the agent is guided by two strategies, (1) a strategy for presenting the data and (2) a strategy for sorting.

The proposed model is composed of functional elements, the applicability of which has been tested. It is essential to point out that the experiments in Chapter 6 related to the TVC clustering method were conducted using the raw form of agents for data preparation and grouping and presentation, incorporating needed commands from Chapter 5 using subprocesses.

Chapter 8

Conclusion

The constant attribute of police work in modern society is the public's interest in their monitoring, followed by great expectations for efficiency in fighting against crime and the symptomatic lack of human and material resources. In particular, the criminal police are expected to react as soon as possible when solving the most serious crimes. Their work is monitored through the media and various mechanisms of systemic control (by government and police internal control), non-governmental organizations, and citizens' organizations and groups. The criminal police are expected to be efficient and effective in identifying the perpetrators of criminal acts, providing the necessary evidence, and protecting personal data and other citizens' rights. Developing systems and methods to overcome resources gaps while helping criminal investigators make the right decisions in a timely manner under pressure is imperative.

This thesis proposes a method of grouping the collected evidence for identity resolution using the improved pairwise unconstrained face clustering. The proposed new approach to clustering, presented in Chapter 3 and tested in Chapter 6, allows the collected information and police reports to be grouped around unique biological characteristics, which are the most suitable means of identifying individuals. The most common biometrics available in digital forensics reports, OSINT and other police methods such as physical surveillance or data collection from various closed-circuit television systems, are face features on images.

The new approach to the unconstrained pairwise clustering method, which is the main contribution, is reflected in maximizing the distance between two feature vectors of face image descriptors by applying the concept of truth-value from NAL. In order to be able to calculate the frequency, one of two elements of truth-value, the NAL concept, need to be adapted since it originally was intended for working with terms. After the truth-value adaptation, other mechanisms from NAL can be used, such as revision and deduction.

The method of system reasoning when deciding if two face images belong to the same person can be explained to the end-user as a comparison of the number of positive evidence, which indicates that it is the same person, and the number of negative evidence, i.e., evidence that suggests there are different persons. Possibility of explaining to criminal investigators how the method works prevent the system from being a black box, which is one of the crucial factors in the problem of accepting similar solutions.

The problem with using all information within systems that store sensitive data, such as criminal police here, is also overcome by using truth-value, which is the second main contribution of this thesis. Due to the lack of evidence that two feature vectors represent the same or different persons, there is a possibility of addressing other points of the distributed system. The mechanism that enables this is revision. Implementation of revision allows that answers from several nodes of a distributed system can be compared, both frequencies f and confidence c, to be decided whether there is a connection between the two vectors. This mechanism is essential, especially when there is insufficient evidence in the node that sent the request.

It should be noted that TVC also exploit the advantages of pairwise clustering and the applied graph technology incorporated into the method. Pairwise clustering allows the method to remain unsupervised but does not require a number of clusters. The application of graphs enables cluster indexing and quick access to grouped data to be presented.

Clustering is based on matching feature vectors obtained with off-the-shelf solutions based on the HOG Dlib library and ResNet. The third main contribution of this thesis is that it has shown that applying different image enhancement techniques improves the effectiveness of face detection and matching. In Chapter 5, it was experimentally established that the application of image sharpening improves face detection. At the same time, normalization achieves better results in matching. Good image enhancement results, especially in detection, are essential for a clustering method that is not based just on descriptor similarity but on the amount of evidence (more face images detected, more evidence on whether two faces belong to the same or different individuals). An additional benefit is that automatic improvement of the quality of images and uniforming them makes it easier for criminal investigators to visually inspect evidence and make decisions when resolving suspects' identities. Due to all the above, image preprocessing is included in the clustering process.

For data preparation, clustering of face images, grouping, and presentation of data, in addition to the new approach to pairwise clustering, the mASAP system model was proposed, which could make significant financial savings by applying open source solutions controlling by the agents. Chapter 7 presents the important roles of the agents mASAP system, namely data preparation, communication and revision, and grouping and presentation. For data preparation, an agent specialization is proposed based on a particular data type. Agents can create their knowledge database using the MIME content type and set of open-source Bash commands while working. Specializing on particular data types, an agent can more efficiently communicate with the database. A similar input data set is expected for police reports related to certain crimes. The exchange of information between the nodes of the distributed systems is done by mutual communication between agents, which protects the data from unauthorized end-user access. How information is obtained from other points of the distributed system depends on what can be shared. The request from other nodes may contain vectors and/or hash values of the vector. Also, the response may be based on vectors that match others within a remote node or only based on hash values in more restrictive systems or systems with occupied resources. Information based on hash values depends on the existence of identical images at remote points.

Investigators' decision-making is mainly influenced by data grouping and presentation. Agents with this role use the relationships created during data preparation and, above all, the clustering process that represents the central point of grouping data from police reports. Two strategies for grouping and presentation have been proposed based on the relationships between the data presented in Chapter 7 and chosen query scenarios. The strategy of presenting data starts from images of people in a cluster of one or more reports in which keywords were found during searches. Then the space for analysis available to the criminal investigator expands to other data and reports only if relevant relations exist. The data is sorted by giving preference to specific characteristics, such as a cluster with most images of faces from a report where keyword search results were found.

Different police reports represent the sources of data and information that need to be grouped and images of persons over whom the clustering method is applied. Digital forensic, OSINT, surveillance, and other police reports produce a large amount of data and various file types. The research results presented in Chapter 4 showed that open-source software allows reading, converting, and storing such data in formats that can be subsequently searched or presented and distributed using web technologies. Of particular importance is the use of subprocesses that allow the control of scripts and open source commands/applications from programming languages. The data preparation agent of the proposed model mASAP system uses this feature when collecting file type information, converting data, importing data into the MongoDB database, and preprocessing images in order to improve them.

The proposed method of applying Bash commands and open-source commands, together with the multi-agent mASAP system, whose model is proposed and discussed in Chapter 7, gives building elements for an advanced but rational criminal police system. It also represents another main contribution of this thesis.

The thesis also left some questions that need to be answered through future work. One of the directions in future work is to reduce the time complexity of the clustering algorithm that affects the efficiency of the process. The second direction is finding and testing technological solutions which accelerate the application of the TVC method in its current form, i.e., to achieve the most optimal results. There is also space to improve assistance to criminal investigators through correlation or clustering of other data, such as textual content or other objects in images. Finally, within the text's content analysis, there is room for testing the possibility of applying NAL in its original form and the implementation of non-axiomatic reasoning more directly in the process of resolving the identity of the person.

8.1 Publications from the doctoral dissertation

- **Vukovic, I.**, Cisar, P., Kuk, K., Bandjur, M., & Popovic, B. (2021). Influence of image enhancement techniques on effectiveness of unconstrained face detection and identification. Elektronika ir Elektrotechnika, 27(5), 49–58., related to Chapter 5 of the thesis.
- **Vuković, I.**, Kuk, K., Čisar, P., Bandur, M., Banđur, Đ., Milić, N., & Popović, B.(2021). Multi-agent system observer: Intelligent support for engaged e-learning. Electronics, 10(12), 1370., related to the agent technology used in Chapter 7.
- **Vuković, I.** (2019). Centralized digital forensic analytical system: Rationalization using open-source software of general purposes. Archibald Reiss Days, 9(2), related to the collection of digital evidence in Chapter 4.
- **Vuković, I.**, Čisar, P., Kuk, K., & Popović, B. (2021). Challenges of contemporary predictive policing. Archibald Reiss Days, 9-10 (pp. 513–526), related to the notes on the challenges of police systems including artificial intelligence in the Introduction.

References

Agent communications. (2021). `https://spade-mas.readthedocs.io/en/latest/agents.html`. (Accessed: 2022-03-26)

Ahmed, N. K., Hemayed, E. E., & Fayek, M. B. (2020). Hybrid siamese network for unconstrained face verification and clustering under limited resources. *Big Data and Cognitive Computing*, *4*(3), 19.

Alawneh, L., Said, M., & Al-Sharif, Z. (2017). Towards hierarchical cooperative analytics architecture in law enforcement agencies. In *2017 8th international conference on information, intelligence, systems & applications (iisa)* (pp. 1–6).

Aleksić, v., & Škulić, M. (2010). *Kriminalistika* (Vol. 1). Pravni fakultet Beograd.

Asaro, P. M. (2019). Ai ethics in predictive policing: From models of threat to an ethics of care. *IEEE Technology and Society Magazine*, *38*(2), 40–53.

Bassett, R., Bass, L., & O'Brien, P. (2006). Computer forensics: An essential ingredient for cyber security. *Journal of Information Science & Technology*, *3*(1).

Boicea, A., Radulescu, F., & Agapin, L. I. (2012). Mongodb vs oracle–database comparison. In *2012 third international conference on emerging intelligent data and web technologies* (pp. 330–335).

Brantingham, P., Glässer, U., Jackson, P., & Vajihollahi, M. (2009). Modeling criminal activity in urban landscapes. In *Mathematical methods in counterterrorism* (pp. 9–31). Springer.

Brantingham, P. L., Glässer, U., Kinney, B., Singh, K., & Vajihollahi, M. (2005). Modeling urban crime patterns: Viewing multi-agent systems as abstract state machines. In *Abstract state machines* (pp. 101–118).

Burmeister, B., Haddadi, A., & Matylis, G. (1997). Application of multi-agent systems in traffic and transportation. *IEE Proceedings-Software*, *144*(1), 51–60.

Calvaresi, D., Dubovitskaya, A., Calbimonte, J. P., Taveter, K., & Schumacher, M. (2018). Multi-agent systems and blockchain: Results from a systematic literature review. In *International conference on practical applications of agents and multi-agent systems* (pp. 110–126).

Carrier, B. (2002). *Open source digital forensics tools: The legal argument.* Citeseer.

Casanovas, P., Morris, N., González-Conejero, J., Teodoro, E., & Adderley, R. (2018). Minimisation of incidental findings, and residual risks for security compliance: the spirit project. In *Terecom@ jurix* (pp. 97–110).

Caskey, T. R., Wasek, J. S., & Franz, A. Y. (2018). Deter and protect: crime modeling with multi-agent learning. *Complex & Intelligent Systems*, *4*(3), 155–169.

Çeliktutan, O., Ulukaya, S., & Sankur, B. (2013). A comparative study of face landmarking techniques. *EURASIP Journal on Image and Video Processing*, *2013*(1), 1–27.

Champod, C., & Tistarelli, M. (2017). Biometric technologies for forensic science and policing: State of the art. *Handbook of Biometrics for Forensic Science*, 1–15.

Chang, L., Pérez-Suárez, A., & González-Mendoza, M. (2019). Effective and generalizable graph-based clustering for faces in the wild. *Computational Intelligence and Neuroscience*, *2019*.

Chicco, D., & Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 1–13.

Cohen, M. I., Bilby, D., & Caronni, G. (2011). Distributed forensics and incident response in the enterprise. *digital investigation*, *8*, S101–S110.

Cohen-Addad, V., Kanade, V., Mallmann-Trenn, F., & Mathieu, C. (2019). Hierarchical clustering: Objective functions and algorithms. *Journal of the ACM (JACM)*, *66*(4), 1–42.

Corbellini, A., Mateos, C., Zunino, A., Godoy, D., & Schiaffino, S. (2017). Persisting big-data: The nosql landscape. *Information Systems*, *63*, 1–23.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 ieee computer society conference on computer vision and pattern recognition (cvpr'05)* (Vol. 1, pp. 886–893).

Degeling, M., & Berendt, B. (2018). What is wrong about robocops as consultants? a technology-centric critique of predictive policing. *Ai & Society*, *33*(3), 347–356.

Dellermann, D., Ebel, P., Sollner, M., & Leimeister, J. M. (2019). Hybrid intelligence. *Business & Information Systems Engineering*, *61*, 637–643.

Dijkstra, P., Bex, F., Prakken, H., & de Vey Mestdagh, K. (2005). Towards a multi-agent system for regulated information exchange in crime investigations. *Artificial Intelligence and Law*, *13*(1), 133–151.

Dong, X., Kim, S., Jin, Z., Hwang, J. Y., Cho, S., & Teoh, A. B. J. (2020). Open-set face identification with index-of-max hashing by learning. *Pattern Recognition*, *103*, 107277.

Dorri, A., Kanhere, S. S., & Jurdak, R. (2018). Multi-agent systems: A survey. *Ieee Access*, *6*, 28573–28593.

Edwards, M., Wattam, S., Rayson, P., & Rashid, A. (2016). Sampling labelled profile data for identity resolution. In *2016 ieee international conference on big data (big data)* (pp. 540–547).

Elshan, E., Zierau, N., Engel, C., Janson, A., & Leimeister, J. M. (2022). Understanding the design elements affecting user acceptance of intelligent agents: Past, present and future. *Information Systems Frontiers*, 1–32.

Federici, C. (2013). Almanebula: a computer forensics framework for the cloud. *Procedia Computer Science*, *19*, 139–146.

Fernandes, D., & Bernardino, J. (2018). Graph databases comparison: Allegrograph, arangodb, infinitegraph, neo4j, and orientdb. In *Data* (pp. 373–380).

Fipa acl message structure specification. (2022). `http://www.fipa.org/specs/fipa00061/SC00061G.html`. (Accessed: 2022-05-20)

Gallego, M. D., Bueno, S., Racero, F. J., & Noyes, J. (2015). Open source software: The effects of training on acceptance. *Computers in Human Behavior*, *49*, 390–399.

Ganesh, V. (2017). Artificial intelligence applied to computer forensics. *International Journal*, *5*(5).

Garcia, T. J. L., Rodriguez-Aguilar, R. M., Alvarez-Cedillo, J. A., & Alvarez-Sanchez, T. (2019). Development of software architecture for a 3d virtual environment with the incorporation of a reactive intelligent agent. *J. Theor. Appl. Inf. Technol*, *97*(17), 4589–4599.

Garfinkel, S. L. (2010). Digital forensics research: The next 10 years. *digital investigation*, *7*, S64–S73.

Georghiades, A. S., Belhumeur, P. N., & Kriegman, D. J. (2001). From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE transactions on pattern analysis and machine intelligence*, *23*(6), 643–660.

González-Briones, A., De La Prieta, F., Mohamad, M. S., Omatu, S., & Corchado, J. M. (2018). Multi-agent systems applications in energy optimization problems: A state-of-the-art review. *Energies*, *11*(8), 1928.

Guo, S., Xu, J., Chen, D., Zhang, C., Wang, X., & Zhao, R. (2020). Density-aware feature embedding for face clustering. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 6698–6706).

Győrödi, C., Győrödi, R., Pecherle, G., & Olah, A. (2015). A comparative study: Mongodb vs. mysql. In *2015 13th international conference on engineering of modern electric systems (emes)* (pp. 1–6).

Haghighat, M., Abdel-Mottaleb, M., & Alhalabi, W. (2016). Fully automatic face normalization and single sample face recognition in unconstrained environments. *Expert Systems with Applications*, *47*, 23–34.

Hammer, P., Lofthouse, T., & Wang, P. (2016). The opennars implementation of the non-axiomatic reasoning system. In *International conference on artificial general intelligence* (pp. 160–170).

Han, H., Shan, S., Chen, X., & Gao, W. (2013). A comparative study on illumination preprocessing in face recognition. *Pattern Recognition*, *46*(6), 1691–1699.

Hillmann, P., Uhlig, T., Rodosek, G. D., & Rose, O. (2014). A novel multi-agent system for complex scheduling problems. In *Proceedings of the winter simulation conference 2014* (pp. 231–241).

Huang, G. B., & Learned-Miller, E. (2014). Labeled faces in the wild: Updates and new reporting procedures. *Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep*, *14*(003).

Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008). Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'real-life'images: detection, alignment, and recognition.*

Imagemagick. (2020). `https://spade-mas.readthedocs.io/en/latest/agents.html`. (Accessed: 2022-05-25)

Jayaraman, U., Gupta, P., Gupta, S., Arora, G., & Tiwari, K. (2020). Recent development in face recognition. *Neurocomputing*, *408*, 231–245.

Johnston, B., & de Chazal, P. (2018). A review of image-based automatic facial landmark identification techniques. *EURASIP Journal on Image and Video Processing*, *2018*(1), 1–23.

Kapoor, R., Gupta, R., Jha, S., & Kumar, R. (2018). Detection of power quality event using histogram of oriented gradients and support vector machine. *Measurement*, *120*, 52–75.

Karim, M. R., Beyan, O., Zappa, A., Costa, I. G., Rebholz-Schuhmann, D., Cochez, M., & Decker, S. (2021). Deep learning-based clustering approaches for bioinformatics. *Briefings in Bioinformatics*, *22*(1), 393–415.

Kazil, J., Masad, D., & Crooks, A. (2020). Utilizing python for agent-based modeling: the mesa framework. In *International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation* (pp. 308–317).

KC, U., & Chodorowski, J. (2019). A case study of adding proactivity in indoor social robots using belief–desire–intention (bdi) model. *Biomimetics*, *4*(4), 74.

Kidwai, A., Arya, C., Singh, P., Diwakar, M., Singh, S., Sharma, K., & Kumar, N. (2021). A comparative study on shells in linux: A review. *Materials Today: Proceedings*, *37*, 2612–2616.

Koen, R. (2009). *The development of an open-source forensics platform* (Unpublished doctoral dissertation). University of Pretoria.

Kumar, P., Happy, S., & Routray, A. (2016). A real-time robust facial expression recognition system using hog features. In *2016 international conference on computing, analytics and security trends (cast)* (pp. 289–293).

Laucka, A., Adaskeviciute, V., & Andriukaitis, D. (2019). Research of the equipment self-calibration methods for different shape fertilizers particles distribution by size using image processing measurement method. *Symmetry*, *11*(7), 838.

Li, B., & Huo, G. (2016). Face recognition using locality sensitive histograms of oriented gradients. *Optik*, *127*(6), 3489–3494.

Li, C., Gunther, M., & Boult, T. E. (2018). Eclipse: Ensembles of centroids leveraging iteratively processed spatial eclipse clustering. In *2018 ieee winter conference on applications of computer vision (wacv)* (pp. 131–140).

Li, J., & Wang, A. G. (2015). A framework of identity resolution: evaluating identity attributes and matching algorithms. *Security Informatics*, *4*(1), 1–12.

Lin, W.-A., Chen, J.-C., Castillo, C. D., & Chellappa, R. (2018). Deep density clustering of unconstrained faces. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 8128–8137).

Lin, W.-A., Chen, J.-C., Ranjan, R., Bansal, A., Sankaranarayanan, S., Castillo, C. D., & Chellappa, R. (2018). Proximity-aware hierarchical clustering of unconstrained faces. *Image and Vision Computing*, *77*, 33–44.

Lyu, G., Fazlirad, A., & Brennan, R. W. (2020). Multi-agent modeling of cyber-physical systems for iec 61499 based distributed automation. *Procedia Manufacturing*, *51*, 1200–1206.

Manju, D., & Radha, V. (2020). A novel approach for pose invariant face recognition in surveillance videos. *Procedia Computer Science*, *167*, 890–899.

Meijer, A., & Wessels, M. (2019). Predictive policing: Review of benefits and drawbacks. *International Journal of Public Administration*, *42*(12), 1031–1039.

Melo, L. S., Sampaio, R. F., Leão, R. P. S., Barroso, G. C., & Bezerra, J. R. (2019). Python-based multi-agent platform for application on power grids. *International transactions on electrical energy systems*, *29*(6), e12012.

Menéndez-Caravaca, E., Bueno, S., & Gallego, M. D. (2021). Exploring the link between free and open source software and the collaborative economy: A delphi-based scenario for the year 2025. *Technological Forecasting and Social Change*, *173*, 121087.

Miljković, A., Čabarkapa, M., Prokin, M., & Budimir, Đ. (2018). The importance of iot and iot forensics. In *Proceedings of archibald reiss days* (Vol. 2, pp. 395–404).

Mitrović, D. (2015). Intelligent multiagent systems based on distributed non-axiomatic reasoning. *Univerzitet u Novom Sadu*.

Nanni, L., Lumini, A., & Brahnam, S. (2017). Ensemble of texture descriptors for face recognition obtained by varying feature transforms and preprocessing approaches. *Applied Soft Computing*, *61*, 8–16.

Ngolah, C. F., & Far, B. H. (n.d.). A tutorial on agent communication and knowledge sharing. *University of Calgary, SENG609*, *22*.

Nhat, H. T. M., & Hoang, V. T. (2019). Feature fusion by using lbp, hog, gist descriptors and canonical correlation analysis for face recognition. In *2019 26th international conference on telecommunications (ict)* (pp. 371–375).

Nowakowski, J., Jedrzejek, C., & Dutkiewicz, J. (2015). Assessing elements of crime based on an agent simulation of a street robbery. In *Challenge+ dc@ ruleml*.

Obied, A., & Abir, H. (2021). Intelligent software agent in e-health system: Review. *Journal of Al-Qadisiyah for computer science and mathematics*, *13*(1), Page–99.

Oprea, M. (2004). Applications of multi-agent systems. In *Information technology* (pp. 239–270). Springer.

Otto, C., Wang, D., & Jain, A. K. (2017). Clustering millions of faces by identity. *IEEE transactions on pattern analysis and machine intelligence*, *40*(2), 289–303.

Pal, C.-V., Leon, F., Paprzycki, M., & Ganzha, M. (2020). A review of platforms for the development of agent systems. *arXiv preprint arXiv:2007.08961*.

Palanca, J., Terrasa, A., Julian, V., & Carrascosa, C. (2020). Spade 3: Supporting the new generation of multi-agent systems. *IEEE Access*, *8*, 182537–182549.

Palanca, J., Terrasa, A., Rodriguez, S., Carrascosa, C., & Julian, V. (2021). An agent-based simulation framework for the study of urban delivery. *Neurocomputing*, *423*, 679–688.

Phillips, M., Amirhosseini, M. H., & Kazemian, H. B. (2020). A rule and graph-based approach for targeted identity resolution on policing data. In *2020 ieee symposium series on computational intelligence (ssci)* (pp. 2077–2083).

Popović, B., Kuk, K., & Kovačević, A. (2018). Comprehensive forensic examination with belkasoft evidence center. In *International scientific conference" archibald reiss days", belgrade, 2-3 october 2018. vol. 2* (pp. 419–433).

Python. (2021). `https://docs.python.org/3/library/subprocess.html`. (Accessed: 2022-04-05)

Python driver for arangodb. (2022). `https://github.com/ArangoDB-Community/python-arango`. (Accessed: 2022-04-26)

Qasem, M. H., Hudaib, A., Obeid, N., Almaiah, M. A., Almomani, O., & Al-Khasawneh, A. (2022). Multi-agent systems for distributed data mining techniques: An overview. *Big Data Intelligence for Smart Applications*, 57–92.

Qi, C., Zhang, J., Jia, H., Mao, Q., Wang, L., & Song, H. (2021). Deep face clustering using residual graph convolutional network. *Knowledge-Based Systems*, *211*, 106561.

Quick, D., & Choo, K.-K. R. (2018). Digital forensic intelligence: Data subsets and open source intelligence (dfint+ osint): A timely and cohesive mix. *Future Generation Computer Systems*, *78*, 558–567.

Raghavan, S. (2013). Digital forensic research: current state of the art. *CSI Transactions on ICT*, *1*(1), 91–114.

Ramey, C. (1994). Bash, the bourne- again shell. In *Proceedings of the romanian open systems conference & exhibition (rose 1994), the romanian unix user's group (guru)* (pp. 3–5).

Ramey, C. (2011). The bourne-again shell. *The Architecture of Open Source Applications*.

Rothe, R., Timofte, R., & Van Gool, L. (2018). Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, *126*(2), 144–157.

Roussev, V. (2011). Building open and scalable digital forensic tools. In *2011 sixth ieee international workshop on systematic approaches to digital forensic engineering* (pp. 1–6).

Saabia, A. A.-B., El-Hafeez, T., & Zaki, A. M. (2018). Face recognition based on grey wolf optimization for feature selection. In *International conference on advanced intelligent systems and informatics* (pp. 273–283).

Schmücker, R. (2016). Incidental findings: definition of the concept. In *Incidental radiological findings* (pp. 3–7). Springer.

Selbst, A. D. (2017). Disparate impact in big data policing. *Ga. L. Rev.*, *52*, 109.

Shi, X., Guo, Z., Xing, F., Cai, J., & Yang, L. (2018). Self-learning for face clustering. *Pattern Recognition*, *79*, 279–289.

Shi, Y., Otto, C., & Jain, A. K. (2018). Face clustering: representation and pairwise constraints. *IEEE Transactions on Information Forensics and Security*, *13*(7), 1626–1640.

Shu, K., Wang, S., Tang, J., Zafarani, R., & Liu, H. (2017). User identity linkage across online social networks: A review. *Acm Sigkdd Explorations Newsletter*, *18*(2), 5–17.

Sinaga, K. P., & Yang, M.-S. (2020). Unsupervised k-means clustering algorithm. *IEEE access*, *8*, 80716–80727.

Sredojević, D., Vidaković, M., Ivanović, M., & Mitrović, D. (2017). Extension of agent-oriented domain-specific language alas as a support to distributed non-axiomatic reasoning. In *International conference on information society and techology (icist 2017), kopaonik, serbia, march* (pp. 12–15).

Srivastava, D. K., & Roychoudhury, B. (2020). Words are important: A textual content based identity resolution scheme across multiple online social networks. *Knowledge-Based Systems*, *195*, 105624.

The sspl is not an open source license. (2021). `https://opensource.org/node/1099`. (Accessed: 2022-05-15)

Tapaswi, M., Law, M. T., & Fidler, S. (2019). Video face clustering with unknown number of clusters. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 5027–5036).

Tarassov, V. B. (2018). Information granulation, cognitive logic and natural pragmatics for intelligent agents. *Open semantic technologies of designing intelligent systems*(8), 45–55.

Taskiran, M., Kahraman, N., & Erdem, C. E. (2020). Face recognition: Past, present and future (a review). *Digital Signal Processing*, *106*, 102809.

Teodorovic, D. (2003). Transport modeling by multi-agent systems: a swarm intelligence approach. *Transportation planning and Technology*, *26*(4), 289–312.

Tomičić, I., Đurić, B. O., & Schatten, M. (2018). Implementing agent roles in massivley multi-player on-line role-playing games. In *Central european conference on information and intelligent systems* (pp. 17–21).

Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., ... Yu, T. (2014). scikit-image: image processing in python. *PeerJ*, *2*, e453.

Van Steen, M., & Tanenbaum, A. S. (2017). *Distributed systems*. Maarten van Steen Leiden, The Netherlands.

Vuković, I. (2019). Centralized digital forensic analytical system: Rationalization using open-source software of general purposes. *Archibald Reiss Days*, *9*(2).

Vukovic, I., Cisar, P., Kuk, K., Bandjur, M., & Popovic, B. (2021). Influence of image enhancement techniques on effectiveness of unconstrained face detection and identification. *Elektronika ir Elektrotechnika*, *27*(5), 49–58.

Vuković, I., Čisar, P., Kuk, K., & Popović, B. (2021). Challenges of contemporary predictive policing. In *Thematic conference proceedings of international significance [elektronski izvor]/international scientific conference" archibald reiss days", belgrade, 9-10 november 2021* (pp. 513–526).

Vuković, I., Kuk, K., Čisar, P., Banđur, M., Banđur, Đ., Milić, N., & Popović, B. (2021). Multi-agent system observer: Intelligent support for engaged e-learning. *Electronics*, *10*(12), 1370.

Wang, L., Ding, S., & Jia, H. (2019). An improvement of spectral clustering via message passing and density sensitive similarity. *IEEE access*, *7*, 101054–101062.

Wang, M., & Deng, W. (2020). Deep face recognition with clustering based domain adaptation. *Neurocomputing*, *393*, 1–14.

Wang, P. (1994). From inheritance relation to nonaxiomatic logic. *International Journal of Approximate Reasoning*, *11*(4), 281–319.

Wang, P. (2000). Unified inference in extended syllogism. In *Abduction and induction* (pp. 117–129). Springer.

Wang, P. (2006). *Rigid flexibility: the logic of intelligence* (Vol. 34). Springer Science & Business Media.

Wang, P. (2007). The logic of intelligence. In *Artificial general intelligence* (pp. 31–62). Springer.

Wang, P. (2009a). Analogy in a general-purpose reasoning system. *Cognitive Systems Research*, *10*(3), 286–296.

Wang, P. (2009b). Formalization of evidence: A comparative study. *Journal of Artificial General Intelligence*, *1*(1), 25.

Wang, P. (2010). Non-axiomatic logic (nal) specification. *University of Camerino, Piazza Cavour*, *19*.

Wang, P. (2013). *Non-axiomatic logic: A model of intelligent reasoning*. World Scientific.

Wang, Y., Fang, Z., & Hong, H. (2019). Comparison of convolutional neural networks for landslide susceptibility mapping in yanshan county, china. *Science of the total environment*, *666*, 975–993.

Wang, Z., Zheng, L., Li, Y., & Wang, S. (2019). Linkage based face clustering via graph convolution network. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 1117–1125).

Weiss, G. (1999). *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT press.

Whitelam, C., Taborsky, E., Blanton, A., Maze, B., Adams, J., Miller, T., … Allen, K. (2017). Iarpa janus benchmark-b face dataset. In *proceedings of the ieee conference on computer vision and pattern recognition workshops* (pp. 90–98).

Wolf, L., Hassner, T., & Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In *Cvpr 2011* (pp. 529–534).

Wu, C.-C., Yeh, W.-C., Hsu, W.-D., Islam, M. M., Nguyen, P. A. A., Poly, T. N., … Li, Y.-C. J. (2019). Prediction of fatty liver disease using machine learning algorithms. *Computer methods and programs in biomedicine*, *170*, 23–29.

Wu, C. M., Huang, Y. F., & Lee, J. (2015). Comparisons between mongodb and ms-sql databases on the twc website. *American Journal of Software Engineering and Applications*, *4*(2), 35–41.

Wu, Y., Chen, P., Luo, X., Wu, M., Liao, L., Yang, S., & Rangayyan, R. M. (2017). Measuring signal fluctuations in gait rhythm time series of patients with parkinson's disease using entropy parameters. *Biomedical Signal Processing and Control*, *31*, 265–271.

Xu, Y., Zhang, Z., Lu, G., & Yang, J. (2016). Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification. *Pattern Recognition*, *54*, 68–82.

Yang, L., Zhan, X., Chen, D., Yan, J., Loy, C. C., & Lin, D. (2019). Learning to cluster faces on an affinity graph. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 2298–2306).

Yates, M. (2010). Practical investigations of digital forensics tools for mobile devices. In *2010 information security curriculum development conference* (pp. 156–162).

Zhan, X., Liu, Z., Yan, J., Lin, D., & Loy, C. C. (2018). Consensus-driven propagation in massive unlabeled data for face recognition. In *proceedings of the european conference on computer vision (eccv)* (pp. 568–583).

Zou, Q., Xie, S., Lin, Z., Wu, M., & Ju, Y. (2016). Finding the best classification threshold in imbalanced classification. *Big Data Research*, *5*, 2–8.

Biography

Igor Vuković was born in Zrenjanin in 1978. He finished elementary school in Orlovat and then the Military Grammar School (Military Gymnasium) in Belgrade in 1997. He graduated from the Military Academy in 2001 as a graduate telecommunication military officer, defending his thesis on "Propagation of electromagnetic waves through waveguides." He completed his master's studies at the Technical Faculty "Mihajlo Pupin" in Zrenjanin, the University of Novi Sad, in the field of Informatics in 2010 with an average grade of 9.66 and defense of the thesis on "Development of an information system for collecting and criminal processing of digital evidence."

Employed in the Ministry of Interior of the Republic of Serbia since 2002.

Igor Vuković participated in the research project "Scalable Privacy-preserving Intelligence analysis for Resolving Identities" - SPIRIT within the framework program of the European Union for research and innovation - Horizon 2020 (Horizon, 2020) from 2018 to 2022. During doctoral studies, he has authored/co-authored seven scientific papers at journals and conferences.

Образац изјаве о ауторству

Изјава о ауторству

Име и презиме студента докторских студија: Игор Вуковић
Број индекса: 2P1/0002/18

Изјављујем

да је докторска дисертација под насловом:

”Разрешавање идентитета и груписање дигиталних доказа о осумњиченима применом технологија препознавања лица и система софтверских интелигентих агената заснованог на неаксиоматском резоновању”

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

У Београду, _______________ Потпис студента докторских студија

Образац изјаве о истоветности штампане и електронске верзије докторског рада

Изјава о истоветности штампане и електронске верзије докторске дисертације

Име и презиме студента докторских студија: Игор Вуковић
Број индекса: 2P1/0002/18
Студијски програм: Докторске студије информатике
Наслов дисертације: ”Разрешавање идентитета и груписање дигиталних доказа о осумњиченима применом технологија препознавања лица и система софтверских интелигентих агената заснованог на неаксиоматском резоновању”
Ментор: др Бранкица Поповић, редовни професор

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањивања у Дигиталном репозиторијуму Криминалистичко-полицијског универзитета.

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Криминалистичко-полицијског универзитета.

У Београду, _________________ **Потпис студента докторских студија**

Образац изјаве о коришћењу

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Криминалистичко-полицијског универзитета унесе моју докторску дисертацију под насловом:
”Разрешавање идентитета и груписање дигиталних доказа о осумњиченима применом технологија препознавања лица и система софтверских интелигентих агената заснованог на неаксиоматском резоновању”

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Криминалистичко-полицијског универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)
2. Ауторство – некомерцијално (CC BY-NC)
[✓ 3.] **Ауторство – некомерцијално – без прерада (CC BY-NC-ND)**
4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)
5. Ауторство – без прерада (CC BY-ND)
6. Ауторство – делити под истим условима (CC BY-SA) (Молимо да заокружите само једну од шест понуђених лиценци.

(Молимо да заокружите само једну од шест понуђених лиценци. Кратак опис лиценци је саставни део ове изјаве).

У Београду, ____________________ **Потпис студента докторских студија**

1. **Ауторство.** Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.
2. **Ауторство** – некомерцијално. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.
3. **Ауторство** – некомерцијално – без прерада. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.
4. **Ауторство** – некомерцијално – делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.
5. **Ауторство** – без прерада. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.
6. **Ауторство** – делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.