Funded by the Horizon 2020 Framework Programme of the European Union
PREVISION - Grant Agreement 833115

Deliverable D2.1
Title: Heterogeneous Data Streams Processing Tools (Initial Release)

Dissemination Level:	**PU**
Nature of the Deliverable:	**R**
Date:	**12/03/2020**
Distribution:	**WP2**
Editors:	**CERTH**
Reviewers:	**PPM, TRIL, SSP**
Contributors:	**CERTH, SPH, IOSB, CTL, ETRA, BPTI**

Abstract: This document describes the initial state of status regarding the WP2 - Extreme-scale Heterogeneous Data Streams Processing. It represents a first version of the developed functionality of each task to be integrated into the PREVISION platform. In addition, future steps regarding the functionalities that will be developed are also reported.

**** Dissemination Level:*** *PU= Public, RE= Restricted to a group specified by the Consortium, PP= Restricted to other program participants (including the Commission services), CO= Confidential, only for members of the Consortium (including the Commission services)*

***** Nature of the Deliverable:*** *P= Prototype, R= Report, S= Specification, T= Tool, O= Other*

Disclaimer

Revision History

Date	Rev.	Description	Partner
14/11/2019	v0.1	Initial version of TOC	CERTH
13/1/2020	v0.3	TOC version 0.3, Input from SPH, ETRA, IOSB, and BPTI has been incorporated	CERTH
20/1/2020	v0.4	TOC version 0.4, update contribution of ITTI and CTL	CERTH
30/1/2020	v0.5	The contribution received from SPH, CTL, ETRA, BPTI and CERTH has been integrated	CERTH, SPH, CTL, ETRA, BPTI
12/2/2020	v0.6	CERTH contribution regarding task 2.4 has been integrated	CERTH
13/2/2020	v0.7	Fix references of bibliography, figures and tables	CERTH
13/2/2020	v0.8	Initial version of the D2.1 uploaded to Gitlab	CERTH
14/2/2020	v0.9	IOSB & BPTI contribution (Section 6) has been integrated	IOSB, BPTI, CERTH
14/2/2020	v0.91	Updated description of pseudonymization	SPH
14/2/2020	v0.92	IOSB new version has been integrated	IOSB, CERTH
17/2/2020	v0.93	Update Introduction, conclusion, abstract and executive summary and glossary sections	CERTH
17/2/2020	v0.94	Minor corrections	CERTH
18/2/2020	v0.95	ETRA new version (section 3) has been integrated	ETRA
20/2/2020	v0.96	Feedback from PPM has been integrated	PPM, CERTH
23/2/2020	v0.961	Feedback from ethics and SPP's reviews have been integrated	TRIL, SPP, CERTH
24/2/2020	v0.962	Review comments have been resolved	CERTH, SPH, ETRA, IOSB, CTL, BPTI
05/03/2020	v0.963	Ethics comments have been resolved	CERTH, IOSB
10/03/2020	V0.964	Updated version	CERTH
12/03/2020	V0.965	Security review	PPM
12/03/2020	R1.0	Approve release version	CERTH

List of Authors

Partner	Author
CERTH	Konstantinos Gkountakos, Stefanos Tsolakidis, Stefanos Vrochidis, Ioannis Kompatsiaris
SPH	Konstantinos Tripolitis, Petros Panagopoulos, Alexandros Bartzas
ETRA	Antonio Moreno Borras, Luisa Perez Devesa
BPTI	Tomas Krilavičius, Monika Briedienė, Justina Mandravickaitė
CTL	Konstantinos Avgerinakis, Panos Mitzias
IOSB	Chengchao Qu, Andreas Specker

Table of Contents

Index of figures

Index of tables

Glossary

Acronym	Definition
ActEV	Activities in Extended Video Evaluation
API	Application Programming Interface
AUC	Area Under Curve
BC	Betweenness Centrality
BD-LSTM	Bi-Directional LSTM
BoW	Bag of Words
C3D	3D-convolutional architecture
CC	Closeness Centrality
CCTV	Closed-circuit television
CDF	Cumulative Distribution Function
CMD	Contour Mask Descriptor
CNNs	Convolutional Neural Networks
CPU	Central Processing Unit
DC	Degree Centrality
DCNN	Deep Convolutional Neural Networks
EC	Eigenvector Centrality
ECDF	Empirical Distribution Function
EM	Expectation Maximization
ESO	Event and Situation Ontology
ETL	Extract Transform Load
EU	European Union
EXIF	Exchangeable Image File
FC	Fully Connected
FCN	Fully Convolutional Networks
FFNN	Feed-Forward Neural Network
FR	Face Recognition
GB	Giga Byte
GFP	Global Feature Pooling
GMM	Gaussian Mixture Model
GTA	Grand Theft Auto
GUI	Graphical User Interface

HD	**High Definition**
HDFS	**Hadoop Distributed File System**
HOG	**Histograms of Oriented Gradients**
HoGP	**Histograms of Grassmannian Points**
HOOF	**Histograms of Oriented Optical Flow**
HTTPS	**Hypertext Transfer Protocol Secure**
I3D	**Inflated 3D**
IBAN	**International Bank Account Number**
ILP	**Integer Linear Programming**
JSON	**JavaScript Object Notation**
kNN	**k-Nearest Neighbors**
LBP	**Local Binary Patterns**
LDS	**Linear Dynamical Systems**
LDT	**Linear Dynamic Texture**
LEA	**Law Enforcement Agencies**
LFPW	**Labeled Face Parts in the Wild**
LFW	**Labeled Faces in the Wild**
LSTM	**Long Short-Term Memory**
MCL	**Markov Cluster**
NAF	**NewsReader Annotation Format**
NIST	**National Institute of Standards and Technology**
NLP	**Natural Language Processing**
PC	**PageRank Centrality**
POS	**Part of Speech**
REST	**Representational State Transfer**
RGB	**Red Green Blue**
ROC	**Receiver Operating characteristic Curve**
RoI	**Region of Interest**
SD	**Seed Distance**
SFTP	**Simple File Transfer Protocol**
shLDS	**stabilized higher order LDS**
SIFT	**Scale Invariant Feature Transform**
SoA	**State of the Art**

SRL	**Semantic Role Labelling**
SSD	**Solid State Drive**
SSH	**Single Shot Headless**
SSH	**Secure Shell**
SSL	**Secure Sockets Layer**
STOEF	**Spatio-Temporal Oriented Energy Features**
SUMO	**Suggested Upper Merged Ontology**
SURF	**Speeded-Up Robust Features**
SVM	**Support Vector Machine**
TB	**Terra Byte**
TCP/IP	**Transmission Control Protocol/Internet Protocol**
TLS	**Transport Layer Security**
Tor	**The onion router**
TPD	**Top Private Domain**
URL	**Uniform Resource Locator**
VLBP	**Volume Local Binary Patterns**
VPN	**Virtual Private Network**
WP	**Work Package**
XML	**eXtensible Markup Language**
PEP	**PREVISION complex Event Processor**

Executive Summary

The deliverable 2.1 (D2.1) “Heterogeneous Data Streams Processing Tools” defines the first steps that will be followed within the PREVISION platform regarding the processing of various data sources. The collection of the data that will be used by PREVISION’s platform includes datasets crawled from Dark and the Clear Web. These datasets are textual-based pseudonymized datasets. The initial release of the crawling tools is presented in D2.1. Furthermore, the initial techniques regarding the interoperability of traffic telecommunication and financial data are illustrated. Specifically, the Extraction, Transformation, and Loading (ETL) strategies are revealed.

In order to fulfill the end-user requirements related to the automatic analysis of visual content, a set of different and various technologies need to be evaluated and appropriately extended to meet the goals of PREVISION. The processing of visual content generated by CCTVs or single video files is tackled by four visual services:

- Activity recognition
- Person re-identification
- Face detection and recognition
- Crisis event detection

The collection of visual analysis tools has already been defined and in many cases, an initial version of these tools has already been proposed.

In addition, social network analytics services have also proposed. Community detection and key actor identification framework are planned to be one of the tools present in the PREVISION platform. Furthermore, the actor identity resolution framework has already been proposed. Moreover, linguistic analysis is also discussed. PREVISION linguistic analysis is based on multiple entities and multiple languages. Social analytics services could be able to consider proposed linguistic features, as the outcome of the deep linguistic analysis. The services regarding social network analysis can be summarized below:

- Community detection & key actor identification
- Actor identity resolution

Finally, it should be noted that this is an initial version and the final version of the document will be available in D2.1 (refined release) on Month 16.

1. Introduction

The necessity of analyzing big data coming from various sources such as camera devices, Dark and the Clear Web, etc. has become a big challenge in the security field, and especially in the European Union (EU). The protection of soft-targets –protection of a stadium-, the prevention of radicalization and terrorism-related threats, the investigation of financial crimes – detection of fraudulent companies-, the detection of terrorism-related cyber-crimes and finally, the investigation of illicit markets are the five use-cases that PREVISION aims to apply.

For most of the aforementioned cases, the data coming from a variety of sources are needed to analyze in order to give to LEAs more pieces of evidence. These data sources are:

- CCTV cameras ;
- Dark and the Clear Web;
- Social networks data;

These heterogeneous data sources had to be managed and analyzed in a short time, regardless of the high amount of data, in order to provide capabilities of LEAs and subsequently improve their investigation processes.

The objectives of PREVISION are summarized below:

- SO-1: Deliver an open, scalable and customizable toolset that provides support for extreme-scale data streams analytics;
- SO-2: Semantically integrate heterogeneous data streams delivering powerful knowledge graphs combined with advanced reasoning and machine learning engines;
- SO-3: Configure and tailor situation awareness enabling techniques and applications to meet specific operational needs of LEAs and address human factors;
- SO-4: Integrate and deploy the developed functions and capabilities into a common platform architecture, making it available to end-users for thorough validation (TRL-7);
- SO-5: Demonstrate and evaluate the developed technologies in realistic cases with the help of LEAs, organize relevant training activities and create a framework for the transfer of knowledge in the use of PREVISION tools from one LEA to another;
- SO-6: Ensure compliance with the legal, ethical, privacy, societal and court-acceptance guidelines and EU best practices;
- SO-7: Ensure the high multi-dimensional impact, continuity and business perspective of project results and allow for incremental investments;

In this deliverable, we describe five separate but also associated fields to address the aforementioned objectives. We report crawling techniques in order to build crawlers able to collect data that subsequently will be analysed by specific tools. The processing of visual streams, the analysis of social media data, the extraction of linguistic characteristics and the interoperability of traffic, financial and telecommunication data are part of these tools.

The deliverable structure is organized as follows:

In section 2 are described the crawling techniques, including both visual and textual anonymization approaches while in section 3 the interoperability of traffic, financial and telecommunication data are presented. Section 4 reports the four distinct visual analysis tools: activity recognition, person re-identification, face detection and recognition, and finally, crisis event detection is detailed. In section 5 the community detection, key actor identification and actor identity resolution problems are presented. Finally, the deep analysis of linguistic features is presented in section 6.

2. Crawling Tools

2.1 Introduction

The goal of the Crawling Tools, under the scope of the PREVISION project, is to collect open/public data (text, images and/or videos) from the Dark and the Clear Web (Social Media included) regarding various crime areas like cybercrime, terrorism, illicit trading of guns, explosives, drugs or art, human trafficking, credit card fraud, money laundering, etc. The collected data will be used for creating datasets available for search and analysis that will help LEAs in crime investigation and threat & risk assessment. In order to allow the collection and analysis of huge amounts of data, the Crawling Tools will be implemented over a Big Data infrastructure composed of the following components:

- *Hadoop (HDFS, HBase, YARN, MapReduce)*: A Hadoop cluster will be used to run crawling and analysis jobs. This allows it to be easily scaled up, in order to meet future needs, to support the storage and analysis of a very big amount of text, images and/or videos simply by adding additional hardware.
- *Elasticsearch*: In order to create searchable datasets available for analysis, crawled data will be indexed using *Elasticsearch*, an open-source search engine that has a distributed system architecture and is based on the Apache Lucene1 library. Elasticsearch provides full-text search capabilities through a Web API following the REST (Representational State Transfer) model and uses JSON files to store data.
- *Application nodes*: The application level of the Crawling Tools will be composed of various components like the Chrome-based crawlers, the API for search and graphs and the crawling and analysis engine. All of them run as Docker containers.

2.2 Entities

Data collection from the Dark Web will be performed taking into account the following proposed entities (search criteria) about an individual who is under investigation, but are not limited to:

- Age, City, Country, LastActive, E-mail, Phone Number (if any), Username, Number of Posts, Registration Date, FullName.

As far as the collection of data from the Clear Web (+ Social Media) is concerned, the following proposed entities will be used, but are not limited to:

- Age, City, Gender, Country, E-mail, Phone Number (if any), username, User Id, Profile (contains fields referred above), Posts, Comments, Events (attended by, created by, created at), FullName, Language, Friends, Followers, Following, Hashtags.

2.3 Pseudonymization

2.3.1 Textual Data

The purpose of textual data pseudonymization is to create a synthetic textual dataset from the text data that has been collected by the Crawling Tools, which will contain no personal data. Personal information retrieved from open/public data sources like names, e-mails or user IDs will be mapped

[1] https://lucene.apache.org/

to securely hash pseudonymized data so that PREVISION use cases can be demonstrated without compromising any information that is considered personal.

To achieve pseudonymization a fingerprinting technique using hashing (HMAC with a key) and lookup tables will be used, where the value of identifier fields will be replaced with a hashed representation (Figure 1). The original data and a generated key will be stored as a key-value pair in either a separate data store or separate index. The followed approach relies on the principle of storing every hash-value pair separately from the data in an identity store for use in a potential lookup, therefore allowing a pseudonymized value to be reversed by an authorized individual.

Since the datasets will be stored in **Elasticsearch**, there will be an **Elasticsearch Client** that will handle the data pseudonymization process. The Elasticsearch Client will retrieve data from the stored datasets and will pseudonymize the personal data. More specifically, the Elasticsearch Client will allow the consistent hashing of the appropriate fields, while it will keep the original value along with its hashed result in a document. Here the hash becomes the pseudonymized data and is used to overwrite the original identifier field value. The hashes and original identifier field values will be stored separately in new documents for lookup purposes, as shown below:

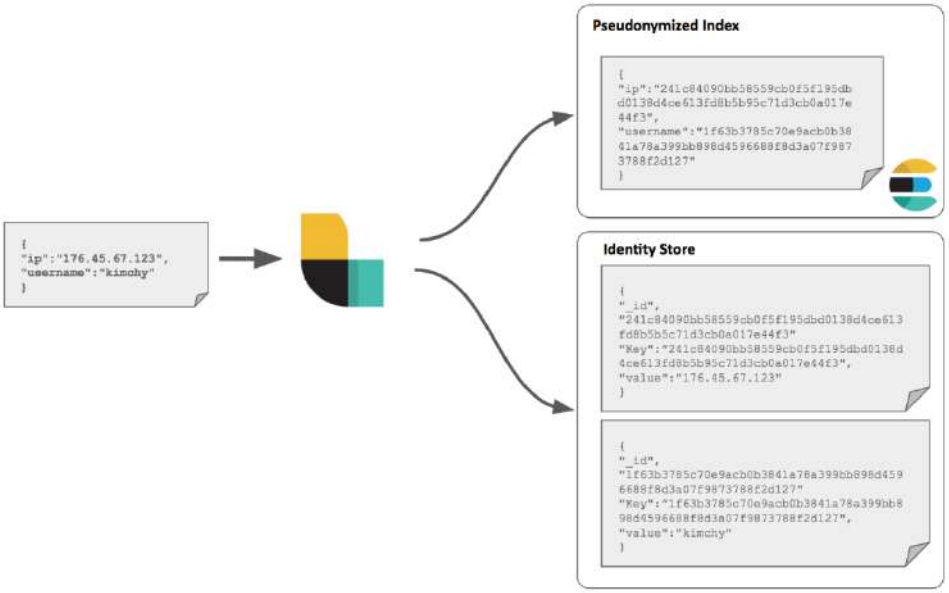


Figure 1.Illustration of the pseudonymization concept that will be used.

2.3.2 Visual Data

Anonymization and pseudonymization are methods used to comply with recital 26 of the GDPR, *"The principles of data protection should therefore not apply to anonymous information, namely information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable."*. In addition, anonymization and pseudonymization techniques are not only referred to textual data but also to visual data, such as videos and images [67].

Regarding anonymization/pseudonymization on visual data, many techniques have been proposed so far. Generally, the same techniques that can be applied to an image could be applied to a video,

allowing it to follow a frame-by-frame approach. Specifically, a video file is a sequence of images, called frames, presented during the time in order to create a movement in human eyes. Thus, the same as the technique for the anonymization of images can be applied to video files on the selected frames. The frames are selected using a fixed number to skip “non-required” information in order to reduce the information that should be processed due to the fact that the anonymization of all frames of a video is a time-consuming process. In the next sub-section, the techniques regarding image anonymization are presented.

2.3.2.1 *Anonymization techniques*

Visual anonymization techniques have been proposed in order to avoid the problem of the individuals' identification from visual content. These includes blurring faces, pixelization/mosaic, cropping, or using blackout bars. The goal is that, by removing information from the image, so that the people whom the data describe remain anonymous, this process also referred as De-identification in the recent literature [153].

The anonymization techniques can be categorized into two categories, transform-domain, and pixel-level. Techniques belong to the first category aim to anonymize the visual content by embedding errors during the encoding of information while on the decoding artifacts and other transformations will be shown. On the other hand, pixel-level techniques focus on pixel-level modifications after decoding to hide the characteristics (I.e. the part or whole persons’ face) that a person can be identified. Both approaches have advantages and disadvantages. The main disadvantage of transformation-domain methods is that they must be applied while the information is recorded. This disadvantage allows the transformation-domain methods feature to have the advantage to be reversible and usually are used by surveillance systems. Specifically, the monitoring system anonymizes the video footage, but law enforcement -if needed- can use specific tools to identify the anonymized information. On the other hand, the pixel level methods applied to the decoded content and have the advantage/disadvantage that cannot be reversible. It noted that some recent research papers introduce blurring filters that enable them to be reversible by adding watermarks [133], [20].

Figure 2 and Figure 3 show indicative examples of both anonymizations applied techniques. Regarding the PREVISION project, only techniques that can be applied are at the pixel-level, as the visual data that could be crawled have already been encoded and stored on the Dark and Clear Web. In the next sub-sections the pixel-level anonymization techniques are briefly presented.

Figure 2. Indicative examples of transform domain method using MPEG-4(left) and JPEG (right) [133].

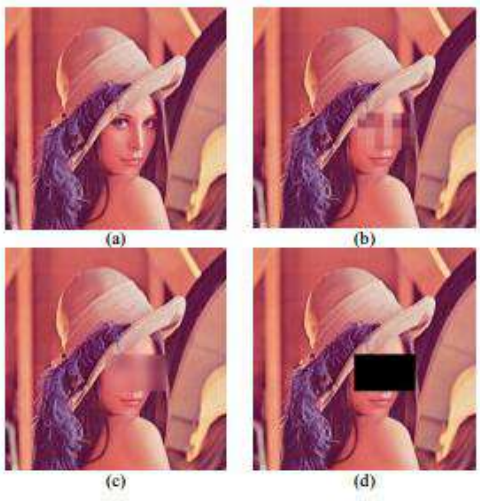


Figure 3. (a) Original image, (b) mosaic, (c) blurring and (d) cutting out examples of pixel-level techniques [20].

2.3.2.1.1 Blurring

Blurring technique is designed to blur the Region of Interest (RoI) by applying a Gaussian blur filter. After filtering, the degradation should be as small as possible in order to de-identify humans or faces and, simultaneously, to have the ability to distinguish image items. An indicative example of the application of a blurring filter is depicted in Figure 3 (c) while the original image is shown Figure 3 (a) [20].

2.3.2.1.2 Pixelization - Mosaic

The simplest pixel-level method - considering the computational cost- is the mosaic (also called pixelization). The basic idea of this approach is to split the RoI of an image into multiple non-overlapping regions with same dimensions and then apply a mean filter for each of the regions. Specifically, for each of these regions, all the pixels' values are replaced by the mean value of the pixels belong to this region [20]. An indicative example is presented in Figure 3 (b).

2.3.2.1.3 Cutting out - Blackout bars

Another simple technique in order to perform de-identification to images is the cutting-out technique. The goal of this technique is to remove entirely the identification information. To this end, for each RoI, the whole or part of the region is colored black. This also can be applied using blackout bars that can be drawn horizontally and/or vertically. The main disadvantage of this method is that the identified information is totally removed, making the rest of the content-unusable, as it does not contain information of interest [20]. Figure 3 (d) presents an example of the application of cutting out pixel-level method.

2.3.2.1.4 Masking

Masking is another method for anonymizing visual data. First of all, the silhouette of the target object is needs to be detected and then colored black. The coloring is not applied to a bounding box in contrast to cutting out, blurring and mosaic approaches, but over the silhouette of the object. Furthermore, a lot of techniques have been proposed in recent literature in order to extract the silhouettes of objects. An indicative example of masking anonymization technique is presented in Figure 4 [79].

Figure 4. Masking anonymization applied to a human silhouette [79].

2.3.2.2 *Summary*

A lot of techniques have been proposed so far regarding visual data anonymization. Mainly, the methods divided into categories, transform-domain, and pixel-level methods. The methods that belong to the pixel-level category are closer to PREVISION needs as can be applied to content that has already been generated. In addition, pixel-level methods include a set of approaches that give options to maintain a fair balance between the information to be discarded and information needed for processing.

2.4 Crawling tools

The Crawling Tools will consist of Crawlers that target and scrap the pages of interest from the Dark and the Clear Web. Possible targets could be darknet markets and forums, clear internet sites, blogs and social media that might contain information related to the crime areas under investigation. Depending on user needs, the crawlers could be configured to scrap even pages that require login, as long as a login account is provided by the user. The targets definition will be the responsibility of end-users.

Crawled data will be parsed and put into Datasets for further use. The information extracted from the crawled data into the datasets will include entities like the ones mentioned in Chapter 2.2, as well as:

- Bitcoin addresses and IBAN numbers found in documents;
- Geolocation of the servers that distribute the targeted pages;
- Geolocation of images (if EXIF data are available);
- Camera models used to take images (if EXIF data are available);
- Image hashes;
- Hits on predefined keywords and triggers;
- Timestamps;

The data stored in the datasets will be accessible for search and analysis through a REST-based API.

2.4.1 Crawlers

Each crawler will have a crawl database (**Apache HBase**), which will be used for storing the results of the crawling process (contents and meta information) and keep track of the crawling configuration (Figure 5). The crawl database will serve as an intermediate data store, which will not be searchable (i.e. exposed to external components). The actual searchable data store will be the dataset (described

in Section 2.4.2), where the crawled data will be indexed and searched. The crawlers will run in so-called crawl cycles. Each crawl cycle will consist of the following steps:

1. Inject new seeds into the crawl database (if needed).
2. Select URLs from the crawl database to fetch (filtering, sorting and limiting).
3. Fetch the selected URLs and write the results back to the crawl database.

The injection process (seeding) and the outlink extraction logic will determine which URLs end up in the crawl database. The URL filtering will determine which URLs are selected for fetching and scoring while limiting logic will determine the order of the URLs that are selected and fetched.

Before a crawler is started, at least one start URL, aka a seed, needs to be provided. There will be no limit in the number of seed URLs, apart from the available capacity of the system that will store the fetched data. New seed URLs can be added while a crawler is running.

The web is too big to crawl. Therefore, the parts of the web that the crawler should visit, as well as the order in which they will be visited, needs to be determined. The following principles will be followed for that matter:

- Filtering - Which URLs are allowed for fetching?
- Scoring - Which URLs should go first?
- Limiting - Keeping things efficient.

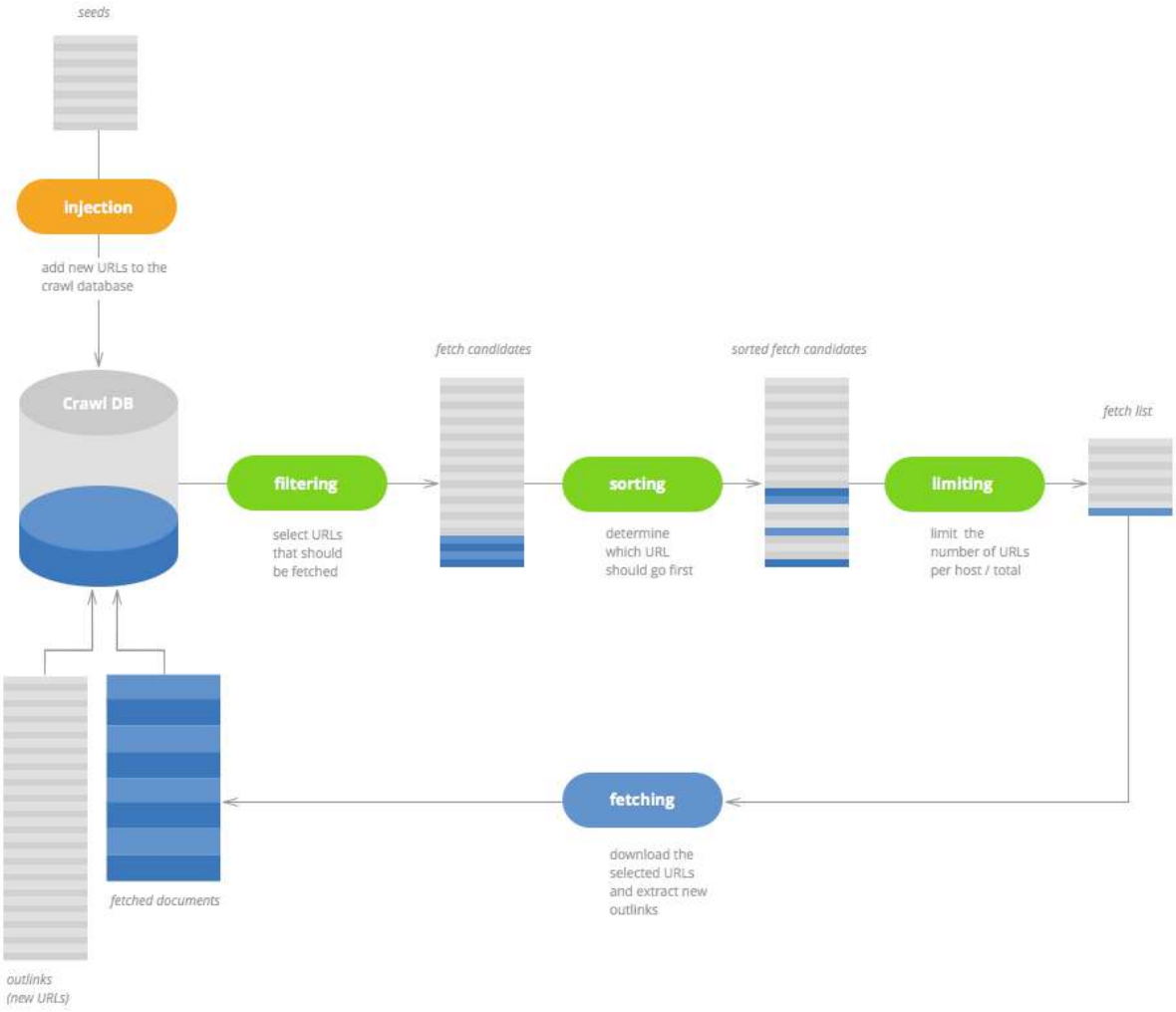


Figure 5. Illustration of the crawling functionality.

Only URLs that pass all filters below will be scheduled for fetching:

1. Is the URL within the maximum seed distances?
2. Does the URL match the specified URL patterns and host characteristics?
3. Does the URL pass the blacklist?
4. For URLs that have been fetched before: do the refresh policies allow a revisit?
5. Only select .onion URLs if the corresponding filter is set to true (applies for the dark web, i.e. Tor crawling)

All the above filters are part of the crawler configuration that is described in more detail below.

2.4.1.1 *Configuration*

Creating and configuring a crawler is an administrative task. The main configuration parameters of a crawler are the following:

- **Seeds**: These are the URLs (e.g. sites like markets or forums) targeted by the crawler. The targeted URL should be as specific as possible, in order to avoid fetching irrelevant data.
- **Distances:** By default, the crawler will only crawl the injected seeds and will not follow any outlinks. This can be changed by increasing the maximum seed distances:
  - **Seed distance (SD)**: this is the minimum number of link hops that is needed to get from one of the seed URLs to the given URL.
  - **TPD distance (TPDD)**: this is the minimum number of TPD (Top Private Domain) hops that is needed to get from one of the seed URLs to the given URL.
- **Crawl-delay**: This is the time to wait (in milliseconds) between successive requests on the same host. For each request, a random number will be taken from a range specified by min and max.
- **Robots ignore**: Web site owners can use the Robots Exclusion Protocol to specify which crawlers are allowed to crawl certain parts of their web site. This protocol is implemented by the robots.txt file in the root of a web site. Although a crawler is not forced to obey the protocol, it is good practice and polite to honour these rules (i.e., by respecting the limitations/rules provided within the robots.txt file). If a crawler does not honour the robots.txt rules, the website owner may decide to block it completely (based on the IP) or submit the crawler (name / IP) to public blacklists. If needed, however, e.g. when robots.txt totally prohibits crawlers, the crawler can be configured to ignore the robots.txt files. This is a global setting for all web sites in the given crawl. Ignoring robots.txt should be combined with high values in Crawl-delay. In that case, a crawler could be mistaken as a real user by a web site monitoring tool.
- **Tor proxy**: In order for a crawler to be able to crawl darknet sites, a tor proxy should be set in its configuration. Tor proxies will be part of the Crawling Tools application layer. In order to be used, they need to be enabled in the crawler's configuration.
- **Blacklist**: It can be used to exclude a list of sites and URL patterns, thus excluding "noise" data from crawling. Regex expressions (regular expression – RegEx – is a sequence of characters that define a search pattern) are allowed in the blacklist.
- **Refresh policy**: By default, the crawler only selects non-fetched URLs. URLs that have been downloaded once will never be fetched again. However, this can change by configuring

the refresh policy. This will allow the crawler to revisit specific URLs after a certain interval. This way crawlers can be configured to monitor certain web pages regularly (e.g. social media for updated posts/activity). Regular expressions (regex) are allowed also in refresh policy.

The above configuration parameters will be set in four kinds of configuration files per crawler:

1. Seeds will be set in one or more seed files. Example of a seed file's content:

```
http://lchudifyeqm4ldjj.onion/?page=0
http://lchudifyeqm4ldjj.onion/?page=1
http://lchudifyeqm4ldjj.onion/?page=2
http://lchudifyeqm4ldjj.onion/?page=3
http://lchudifyeqm4ldjj.onion/?page=4
http://lchudifyeqm4ldjj.onion/?page=5
http://lchudifyeqm4ldjj.onion/?page=6
http://lchudifyeqm4ldjj.onion/?page=7
http://lchudifyeqm4ldjj.onion/?page=8
http://lchudifyeqm4ldjj.onion/?page=9
http://lchudifyeqm4ldjj.onion/?page=10
http://lchudifyeqm4ldjj.onion/?page=11
http://lchudifyeqm4ldjj.onion/?page=12
http://lchudifyeqm4ldjj.onion/?page=13
http://lchudifyeqm4ldjj.onion/?page=14
http://lchudifyeqm4ldjj.onion/?page=15
http://lchudifyeqm4ldjj.onion/?page=16
http://lchudifyeqm4ldjj.onion/?page=17
http://lchudifyeqm4ldjj.onion/?page=18
http://lchudifyeqm4ldjj.onion/?page=19
```

2. Distances, Crawl-delay, Robots ignore and Tor proxy will be set in a configuration XML file. Example of an XML's content:

```
<?xml version="1.0"?>
<configuration>
  <property basic="false" cluster="false" analysis="false" crawl="false" none="false">
    <name>max.seedDistance</name>
    <label>Maximum seed distance</label>
    <value>2</value>
    <description>Constrain crawler with respect to distance of seeds. Use 1 to crawl a
single url, use &gt;1 to spider
            outlinks.</description>
    <category>Generator</category>
    <validator/>
    <source>defaults.xml</source>
    <defaultvalue>1</defaultvalue>
    <required/>
    <related/>
  </property>
  <property basic="false" cluster="false" analysis="false" crawl="false" none="false">
    <name>max.tpdDistance</name>
    <label>Maximum tpd(top private domain) distance</label>
    <value>1</value>
    <description>Constrain crawler with respect to distance of top private domains of
seeds. Use 1 to stay on the
            same top private domain.</description>
    <category>Generator</category>
    <validator/>
    <source>defaults.xml</source>
    <defaultvalue>1</defaultvalue>
```

```
    <required/>
    <related/>
  </property>
  <property basic="false" cluster="false" analysis="false" crawl="false" none="false">
    <name>fetcher.delay.min</name>
    <label>Minimum fetcher delay time</label>
    <value>10000</value>
    <description>Minimum time to wait (in milliseconds) between successive requests on
the same host. For each
            request, a random number will be taken from the range specified by min and
max.</description>
    <category>Fetcher</category>
    <validator/>
    <source>defaults.xml</source>
    <defaultvalue>2000</defaultvalue>
    <required/>
    <related/>
  </property>
  <property basic="false" cluster="false" analysis="false" crawl="false" none="false">
    <name>fetcher.delay.max</name>
    <label>Maximum fetcher delay time</label>
    <value>20000</value>
    <description>Maximum time to wait (in milliseconds) between successive requests on
the same host. For each
            request, a random number will be taken from the range specified by min and
max.</description>
    <category>Fetcher</category>
    <validator/>
    <source>defaults.xml</source>
    <defaultvalue>5000</defaultvalue>
    <required/>
    <related/>
  </property>
  <property basic="false" cluster="false" analysis="false" crawl="false" none="false">
    <name>fetcher.robots.ignore</name>
    <label>Ignore robots.txt</label>
    <value>true</value>
    <description>Instruct the crawler to ignore robots.txt instructions if
present.</description>
    <category>Fetcher</category>
    <validator/>
    <source>defaults.xml</source>
    <defaultvalue>false</defaultvalue>
    <required/>
    <related/>
  </property>
  <property basic="false" cluster="false" analysis="false" crawl="false" none="false">
    <name>generator.onion</name>
    <label>Tor onion</label>
    <value>true</value>
    <description>Schedule (i.e., generate) .onion hosts for fetching. Combine with
genlogic revtld:onion to stay on
            tor hidden services.</description>
    <category>Generator</category>
    <validator/>
    <source>defaults.xml</source>
    <defaultvalue>false</defaultvalue>
    <required/>
    <related/>
  </property>
  <property basic="false" cluster="false" analysis="false" crawl="false" none="false">
    <name>datadragon.fetcher.enabled</name>
    <label>Enable datadragon</label>
    <value>true</value>
    <description>Enable the dragon box service for browser fetching</description>
    <category>Fetcher</category>
    <validator/>
```

```
    <source>defaults.xml</source>
    <defaultvalue>false</defaultvalue>
    <required/>
    <related/>
  </property>
  <property basic="false" cluster="false" analysis="false" crawl="false" none="false">
    <name>datadragon.fetcher.chrome.env.extra_chrome_args</name>
    <label/>
    <value>--proxy-server="socks5://xxx.xxx.xx.xxz:9050" --host-resolver-rules="MAP *
0.0.0.0 , EXCLUDE xxx.xxx.xx.xxx" --user-agent="Mozilla/5.0 (Windows NT 6.1; rv:52.0)
Gecko/20100101 Firefox/52.0"</value>
    <description/>
    <category/>
    <validator/>
    <source/>
    <defaultvalue/>
    <required/>
    <related/>
  </property>
  <property basic="false" cluster="false" analysis="false" crawl="false" none="false">
    <name>legacy.crawler</name>
    <value>false</value>
  </property>
</configuration>
```

3. Blacklist will be set in one blacklist file with as many lines as required. Example of a blacklist file's content:

```
# Skip the mail.google.com host, but allow other google.com hosts
host:mail.google.com

# Skip all hosts under the google.com top private domain
site:google.com

# Skip all addresses containing the word search
regex:.*search.*

# Skip all google.com addresses containing the word search
and(site:google.com,regex:.*search.*)

# Skip all google.com addresses that containing either search or login
and(site:google.com,or(regex:.*search.*,regex:.*login.*))
```

4. Refresh policy will be set in one refresh file with as many lines as required. Example of a refresh file's content:

```
# Revisit only the profiles from plus.google.com every ten days
and(host:plus.google.com,regex:.*/profiles/.*) 10d 0h 10min

# Revisit the profiles and groups from plus.google.com every hour and a half
and(host:plus.google.com,or(regex:.*/profiles/.*,regex:.*/groups/.*)) 0d 1h 30min

# Try again URLs that were temporary unavailable at the previous request in a day
status:503 1d 0h 0min
```

2.4.1.2 *API*

The Crawling Tools will be installed with preconfigured crawlers that will be set up based on the use cases and the LEAs needs. Nevertheless, if a modification in the crawlers' configuration is required, a crawler management API (REST) will be provided that will allow authorized users to modify the behaviour of the crawlers. A short description of the crawler management API capabilities is given below.

- **Uploading a seeds file**: Uploading a seeds file could be done using a multipart request. Example:

```
curl -H "X-User-Email: user@example.com" -H "X-User-Token: [security_token]" -F
data=@seeds http://<host>:<port>/api_endpoint/v1/crawlers/<crawler_name>/upload_seeds
```

- **Getting the current crawl configuration**: The current crawl configuration xml could be downloaded. Example:

```
curl -H "X-User-Email: user@example.com" -H "X-User-Token: [security_token]"
http://<host>:<port>/api_endpoint/v1/crawlers/<crawler_name>/current_configuration
```

- **Setting the current crawl configuration**: The current crawl configuration xml could be replaced with a new one using a multipart request. Example:

```
curl -H "X-User-Email: user@example.com" -H "X-User-Token: [security_token]" -F
data=@config.xml -XPUT
http://<host>:<port>/api_endpoint/v1/crawlers/<crawler_name>/current_configuration
```

- **Getting the current blacklist**: The current crawl blacklist configuration could be downloaded. Example:

```
curl -H "X-User-Email: user@example.com" -H "X-User-Token: [security_token]"
http://<host>:<port>/api_endpoint/v1/crawlers/<crawler_name>/blacklist
```

- **Setting the current blacklist**: The current crawl blacklist could be replaced with a new one using a multipart request. Example:

```
curl -H "X-User-Email: user@example.com" -H "X-User-Token: [security_token]" -F
data=@blacklist -XPUT
http://<host>:<port>/api_endpoint/v1/crawlers/<crawler_name>/blacklist
```

- **Getting the current refresh policy**: The current crawl refresh policy could be downloaded. Example:

```
curl -H "X-User-Email: user@example.com" -H "X-User-Token: [security_token]"
http://<host>:<port>/api_endpoint/v1/crawlers/<crawler_name>/refresh
```

- **Setting the current refresh policy**: The current crawl refresh policy could be replaced with a new one using a multipart request. Example:

```
curl -H "X-User-Email: user@example.com" -H "X-User-Token: [security_token]" -F
data=@refresh -XPUT
http://<host>:<port>/api_endpoint/v1/crawlers/<crawler_name>/refresh
```

If required, the crawler management API can be enhanced so that it is possible to also create, start or stop a crawler.

2.4.1.3 *Data collection period*

In order for the collected data to be usable and reliable, it is advised to start crawling the desired targets at least 3 months before data analysis.

2.4.2 Datasets

The dataset will be the data store available for searching and performing analysis tasks. It will be based on **Elasticsearch** technology. A dataset will be created by parsing and analysing records, as well as extracting metadata and entities that are stored in one or more crawl databases (Figure 6).

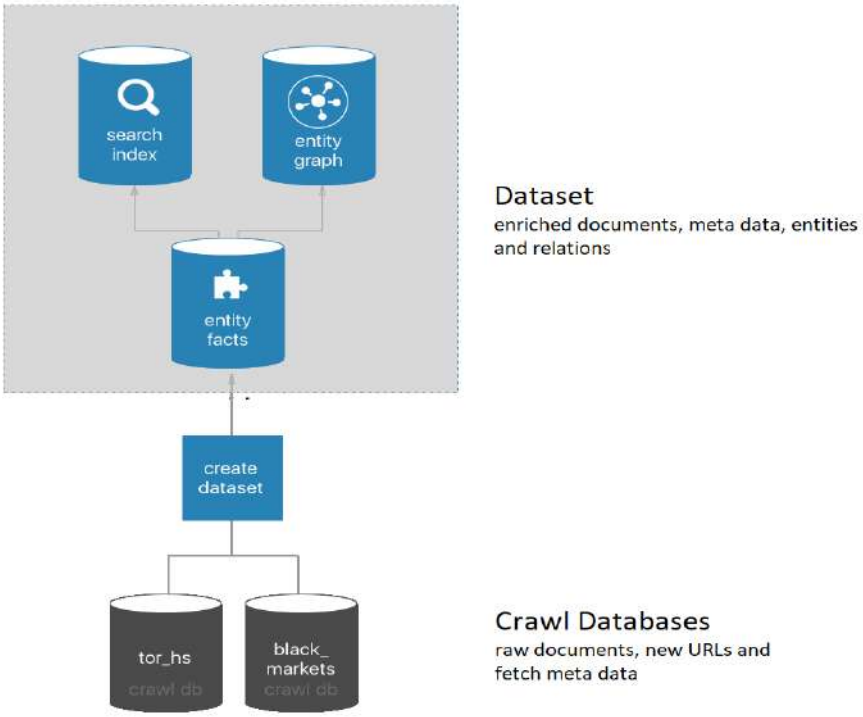


Figure 6. Dataset creation and utilization.

This process is separated from crawling for several reasons:

- Reproducibility: it allows a complete re-processing of all the source (crawled) data
- Combination: it allows combining records from different crawlers (i.e., targeting different dark markets) in one dataset
- Reusability: it allows creating different datasets (with different analysis) from the same sources (crawl databases)
- Security: it only exposes specific data to third parties (i.e. other PREVISION components).

2.4.2.1 *Configuration*

Creating and configuring a dataset is an administrative task. The main configuration parameters of a dataset are the following:

- **Sources**: The crawl databases that will be included in the dataset.
- **Plugins**: Through this parameter, the plugins that are required for the analysis of the dataset will be enabled. The Crawling Tools will provide a plugin system which will allow activating custom analysis components that will be used e.g. for site-specific entity extraction. For example, if data was collected from darknet markets, then the corresponding plugin will be activated to extract entities like advertisements, vendors, reviews, etc.

- **Dictionaries**: They are ways to document and use expert knowledge. Expert knowledge comes in many forms and is usually developed through experience. An example of expert knowledge is the recognition of drug trafficking. For such a domain, a list of drug names can be created to mark websites (i.e. marketplaces) revolving around drugs. The Crawling Tools will support the following types of Dictionaries.
    - **Labels**: These are lists of keywords. Labels can be created from a list of words from expert knowledge and used to tag the crawled sites based on their content. For example, if one or more keywords from a list of drug names are found on a crawled site, then this site could be labelled as a drug market site.
    - **Maps**: These are lists of predefined word mappings (i.e. one-word maps to another word). Sets of words could be defined that belong together in a specific context. Several different words could be mapped to the same word, like the words 'angry' and 'bad' to 'negative' or the words 'happy' and 'joy' to 'positive', thus creating a basic sentiment analysis. Another scenario suited for the use of maps is that of topic classification. Suppose there is interest in marketplaces selling all kinds of products, from weapons to drugs. To separate the different products from each other some mappings can be defined, such as 'weed' and 'cocaine' to 'drugs' and 'knife' and 'AK-47' to 'weapon'. These mappings can be used to group the crawled sites based on their topic.

The use of dictionaries will help with the classification of the crawled sites. For example, if an individual's information is found in sites that have negative classification, e.g. drug smuggling sites or weapon selling markets, then this individual could be considered as a potential threat. The above configuration parameters will be set in three kinds of configuration files per dataset:

1. Sources and Plugins will be set in a configuration XML file, where one or more crawler databases can be configured for the same dataset. Example of an XML's content:

```
<?xml version="1.0"?>
<configuration>
  <property basic="false" cluster="false" analysis="false" crawl="false" none="false">
    <name>fetchdbs</name>
    <label>Fetch database names</label>
    <value>crawl_db1,crawl_db2</value>
    <description>Comma separated list of fetch DB's (i.e., doc tables in HBase) to
include during analysis.</description>
    <category>Sources</category>
    <validator/>
    <source>programatically</source>
    <defaultvalue/>
    <required/>
    <related/>
  </property>
  <property basic="false" cluster="false" analysis="false" crawl="false" none="false">
    <name>plugins.files</name>
    <label>Plugin files</label>
    <value>/user/root/extensions/darkmarkets_bundle-1.jar</value>
    <description>List of all jar files (separated by ;)that should be scanned for
plugin implementations. When the plugin has
            @AutoScan annotations the scanning will be done automatically. When this
is not the case autoscan is
            disabled the package names should be listed in the sitespecific.packages
field.</description>
    <category>Plugins</category>
    <validator/>
```

```
    <source>programatically</source>
    <defaultvalue/>
    <required/>
    <related/>
  </property>
</configuration>
```

2. Labels will be set in one or more label files with as many lines as required per file. Example of a label file's content:

```
weed
cocaine
heroine
lsd
marihuana
```

3. Maps will be set in one or more map files with as many lines as required per file. Example of a map file's content:

```
machine gun    weapon
pistol  weapon
rpg     weapon
cannon  weapon
```

2.4.2.2 *API*

As in the case of crawlers, the Crawling Tools will be installed with preconfigured datasets. Again, if a modification in the datasets configuration is required, a management API (REST) will be provided to allow authorised users to modify the behaviour of the datasets. A short description of this management API's capabilities is given below.

- **Getting the current dataset configuration**: The current dataset configuration xml could be downloaded. Example:
- curl -H "X-User-Email: user@example.com" -H "X-User-Token: [security_token]" http://<host>:<port>/api_endpoint/v1/datasets/<dataset_name>/current_configuration
- **Setting the current dataset configuration**: The current dataset configuration xml could be replaced with a new one using a multipart request. Example:
- curl -H "X-User-Email: user@example.com" -H "X-User-Token: [security_token]" -F data=@config.xml -XPUT http://<host>:<port>/api_endpoint/v1/datasets/<dataset_name>/current_configuration
- **Uploading a label file**: Uploading a label file could be done using a multipart request. Example:
- curl -H "X-User-Email: user@example.com" -H "X-User-Token: [security_token]" -F data=@labels http://<host>:<port>/api_endpoint/v1/datasets/<dataset_name>/upload_label
- **Uploading a map file**: Uploading a map file could be done using a multipart request. Example:
- curl -H "X-User-Email: user@example.com" -H "X-User-Token: [security_token]" -F data=@labels http://<host>:<port>/api_endpoint/v1/datasets/<dataset_name>/upload_map

If required, the dataset management API can be enhanced so that it is possible to also create, start or stop a dataset.

2.4.2.3 *Data retention period*

As mentioned earlier, in order for the collected data (datasets) to be usable for investigation, they should be kept in the data store for a period of time that would be beneficial for the investigation (at the moment we are considering a time duration of at least three months, but this could be modified based on the needs of the LEAs and the use cases). Then they could be deleted, either upon completion

of the pilot operation of the PREVISION project or upon completion of the whole project, and at all times respecting the data retention policy of the project (all data will be deleted upon project completion).

However, when the system will be released for production use by LEAs (after the project's duration – not in scope of WP2/D2.1) it is recommended to keep data as long as possible, at least as long as storage capacity allows because old information can actually help to identify a potential threat.

2.4.3 Search API

Search and analysis tasks could be executed over a created dataset through the **REST API** provided by the Crawling Tools. This API will actually wrap the **Elasticsearch** REST API and allow external components to perform search queries and aggregations over the dataset. The Search API could be either accessed directly by another component (e.g. a Search GUI) or integrated with a publish - subscribe infrastructure, where search queries and aggregations would be published and consumed. More details on the latter implementation will be provided at a later stage of the project when PREVISION design is mature. Figure 7 presents an overview of the crawled data flow:

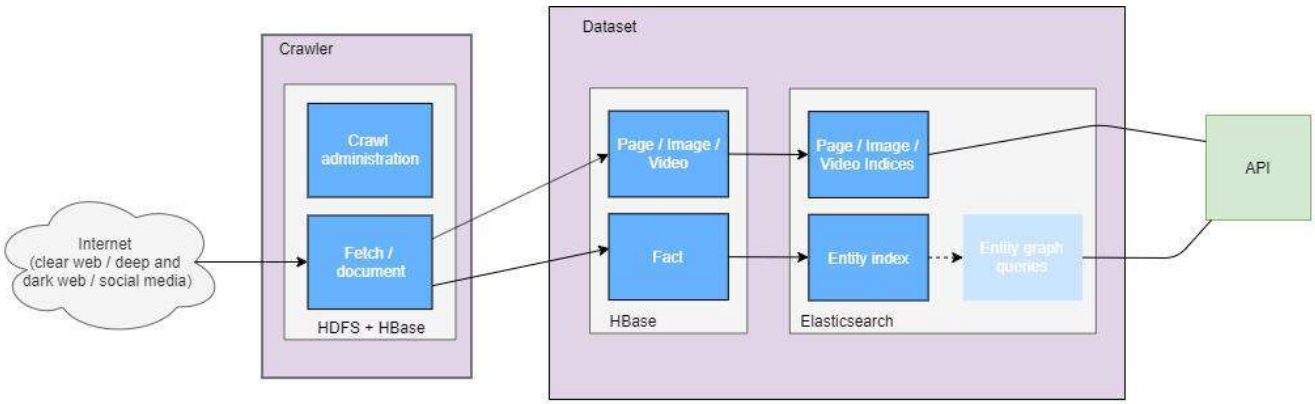


Figure 7. Crawling tools data flow.

Datasets will continuously be updated with new data as long as the crawlers are fetching new data from the Dark and Clear Web sources. Dataset updates will not interfere with searches, i.e. a search or an aggregation could still be performed over the dataset while it is being updated. However, such a search or aggregation might not return any results for an individual, if the information that has been fetched by the crawlers and is related to this individual has not been extracted into the dataset yet. This is why the crawling process should run for sufficient time (this would be defined by the LEAs) before the first search or analysis task is executed. It is crucial that the dataset has sufficient data for searching and analysing.

2.4.4 Test Environment

The Crawling Tools test environment could be deployed either on-premise or on a secure cloud infrastructure. The minimum hardware requirements to support crawling are:

- **Hadoop cluster of three nodes** (Apache Ambari, Zookeeper, HDFS, HBase, YARN, MapReduce2, Ambari Metrics) with two master and two data nodes (one data node serves also as a master node), with at least 4 CPUs and 32 GB of memory per node.

- **Elasticsearch cluster of three nodes** with two master and two data nodes (one data node serves also as a master node). For testing purposes, the Elasticsearch cluster could reside on the same machines with the Hadoop cluster, but it is advisable to separate the two clusters in production mode.
- The application will run on **Docker** containers orchestrated by **Rancher**, which is a complete software stack for managing containers. For testing purposes, the containers could reside with Rancher on the same nodes as the Hadoop master nodes, but it is better to run it on separate nodes in production mode.
- The amount of storage depends on the use cases, but for testing purposes, at least **1 TB of data storage (SSD)** should be available and at least **64 GB** should be dedicated to application needs.
- The internet connection speed also depends on the use cases. A recommended value for production is **100 Mbit/s**, but for testing purposes, lower speed, such as **24 Mbit/s,** is also adequate.
- A high-speed network between nodes is also recommended (**10GBit** preferably).

The above set-up could be scaled-up horizontally for production, meaning that the capacity of the cluster would grow by adding more nodes. Depending on the use case, if there will be a need to increase the number of crawlers and datasets that run in parallel, then an increase in CPU and memory per node might be required.

2.5 Conclusion and Future Steps

In this chapter, the utilization of the Crawling tools within PREVISION was presented along with the crawling techniques that will be used for targeting Dark and Clear Web data sources. A detailed description regarding dataset generation from crawled data was also provided. Furthermore, a draft proposal of the search functionality over the composed datasets was presented. Overall, this chapter shall act as a foundation for the capabilities of the Crawling tool within PREVISION that will enable stakeholders to perform complex crime investigations and threat risk assessments.

3. Interoperability with Traffic, Telecom and Financial Data Sources and Video Analysis Events

3.1 Introduction

The PREVISION system will require individual connections with available traffic, telecom, financial and video (the result of video and image events after having processed video surveillance data) data sources. The ETL component developed by ETRA will be essential to ingest this type of data, as this technology will allow the PREVISION system to receive relevant data from all the different data sources of PREVISION, e.g. darknet, LEA's file systems or Open datasets, transform the data applying specific business rules, keep the necessary data and load the transformed data into PREVISION's storage system.

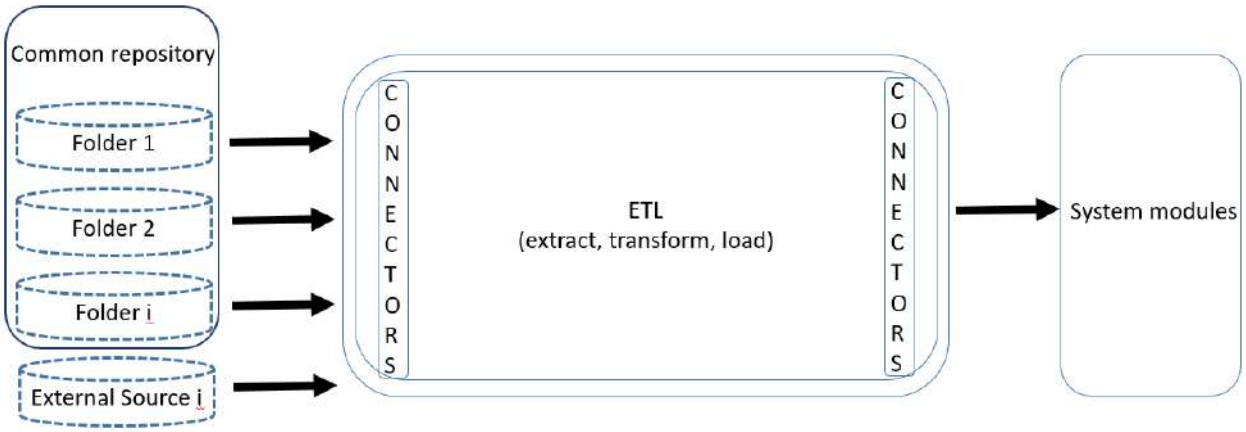


Figure 8 – Architecture of the ETL module

3.2 Communication Protocols

The connections established between the ETL component and the different remote data sources, including database systems and applications, will use encrypted TCP/IP communication protocols such as TLS, SSL, HTTPS or SFTP. The specific protocol to be used will depend on the type of access to the data source.

At this point of the project, the kind and number of endpoints are not confirmed, therefore the PREVISION system will probably require an ETL component for each one of the different datasets, connecting to them via VPN or SSH. However, if all the data suppliers upload their data to a local repository in the PREVISION system (as it is described in 3.3), the ETL would require a simple TCP connection, as ETLs have the flexibility to read any known file type and connect to any kind of repository.

3.2.1 Communication Protocols example

As the data sources are not yet decided, the connection to an external secure SFTP (secure ftp) server in which a LEA or other related organization leaves a daily document, e.g. an excel document, with data to be processed by the Prevision system will be simulated.

The ETL will have a component for the SFTP connection. When the ETL is started, it will connect to the configured SFTP, and once the connection is established another component will load the file to the ETL. Then, another component will read all the lines of the file, one at a time, and by means of a data

flow this component will send all the lines to another component with the obligation to read and possible modify some fields. Then, another data flow will send the transformed (or raw, if no transformation was required) data to the next component, which will establish the connection with the last module, e.g. kafka or Nats queue. This last component will send the data stream to the queue for the appropriate topic, and then all the read data will be sent to the entrance queue for the rest of the system.

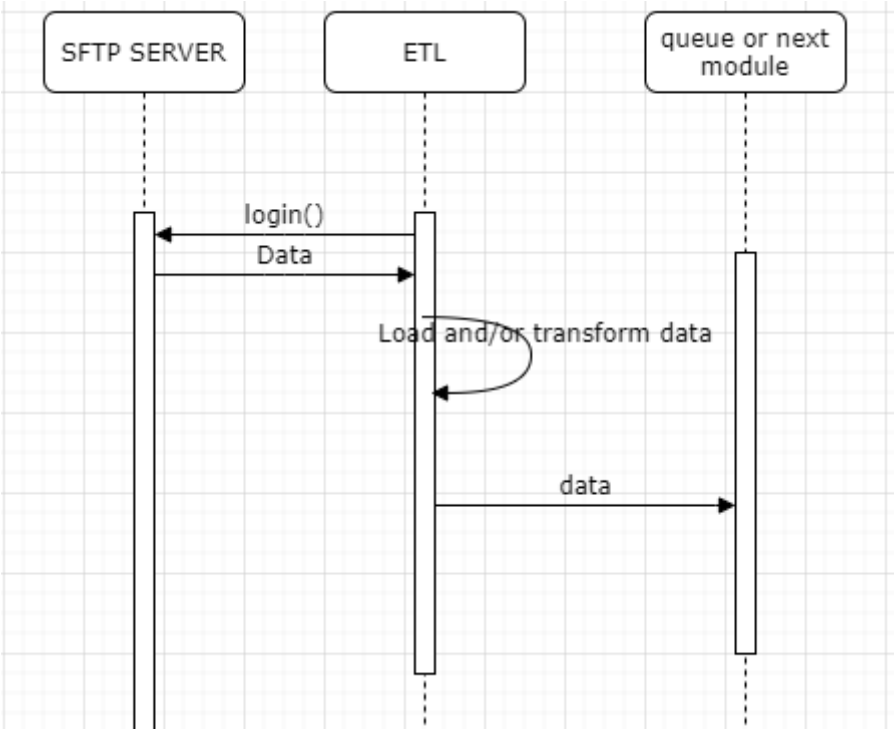


Figure 9. Example of a data flow between an SFTP server and the PREVISION system

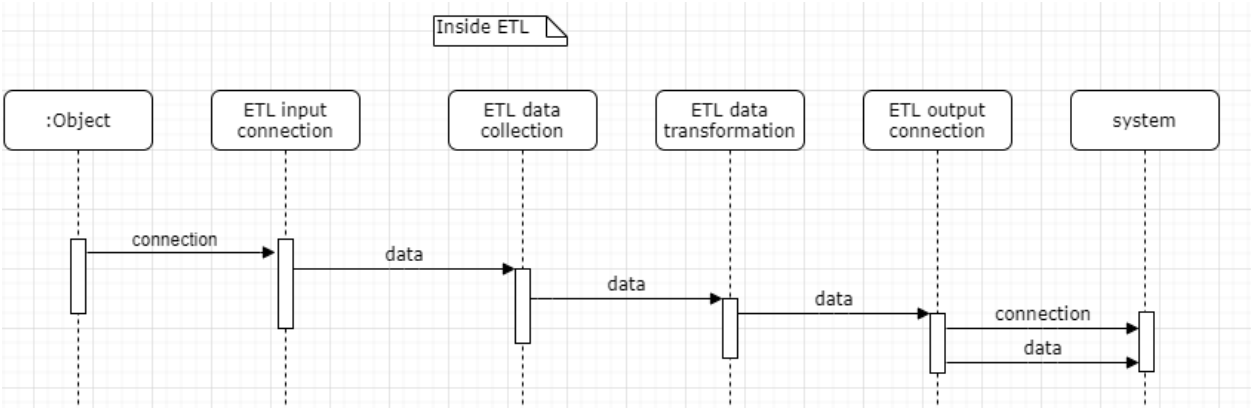


Figure 10. Internal data flow of the ETL

3.3 Extract, Transform and Load (ETL) of data sources

The different data streams used in PREVISION might not actually be streaming data. Besides, although the number of data sources will be later defined, it is expected to be numerous enough, thus requiring

connections to many different sources, which could not be properly operated. Hence, the recommendation is to have a common repository where each data supplier can send their data to, regardless of the format of each specific input. Currently, due to the sensitivity of the data, having a huge amount of high-quality data is becoming even more complex. The consortium has developed a questionnaire about the data sources used for refining the use-cases (D1.3). This questionnaire aims to develop a common methodology for sharing and creating datasets in a lawful manner.

In the development phase, to let the team develop and test the initial system, files with no personal data received via e-mail or private file transfers will be accepted, and although these upload methods will not be allowed in the integrated version. version

The received data will go through a process to transform it into the data format required by the next modules. This process involves the following phases.

3.3.1 Extraction

The extraction phase involves establishing connections to the common repository (assuming the use of a common repository can be used by the data suppliers) or the external data sources and extracting the data made available by the data suppliers. The session will be open while the data specific ETL is running.

The files will be read line by line to extract the fields needed for the subsequent processes.

At this stage, it is not yet decided whether the processed files will be moved to a processed files repository or deleted. Nevertheless, there is an identified need to distinguish what data has already been processed.

3.3.2 Transformation

When considered necessary, the data will be transformed or pre-processed to check its integrity and quality. The type of transformations that can be performed are not defined yet, as the LEA's have not confirmed what data can be provided. Some transformation examples are the management of null values (by replacing them with default values or simply deleting them), aggregations (mean, sum, min, max) or curation definitions.

In the data transformation stage, a series of rules or functions are applied to the extracted data in order to prepare it for loading into the end target.

An important function of the transformation is data cleansing, which aims to pass only "proper" data to the target. The challenge, when different systems interact, is the interface and communication between the relevant systems. Character sets available in one system may not be available in others.

In other cases, one or more transformation types may be required to meet the business and technical needs of the server or data warehouse. Some examples of standard transformations are:

- Select the columns to be loaded. Example: the source data has the following three columns (aka "attributes"): roll_no, age, and salary. If the age is not relevant for the end user, then the roll_no and salary attributes will be selected. In case no salary data is received, the selection mechanism will ignore all the records.
- Translate coded values when the target and source systems use different codes for the same the parameter. Example, the gender might be coded by the source system as "1" for

male and "2" for female, while the Prevision system might code male as "M" and female as "F".

- Encode free-form values, e.g. mapping of "Male" to "M".
- Derive a new calculated value, e.g. sale_amount = qty * unit_price.
- Sort the data in the columns, which helps improve the performance of a later search.
- Join data from multiple sources.
- Aggregate i.e. summarize multiple rows of data, e.g. total sales per store or region.
- Generate surrogate-key values.
- Transpose or pivot.
- Split a column into multiple columns.
- Disaggregate repeating columns.
- Look up and validate the relevant data from tables or referential files.

At the end of this phase, regardless of a transformation being applied to the received data, the processed data is ready for the next phase, the loading into PREVISION's system.

3.3.3 Loading

In this phase, the ETL will load the transformed data into the PREVISION Complex Event Processor (PEP), which will process the data and store the minimum data considered necessary, profitable and useful. The ETL will load the data into PREVISION's system or data warehouse to let the next modules analyze it. The load operation to send the processed data to the target database will require using a database-specific connector from the range of available connectors. Although the technical decisions regarding storage system are not yet made, considering that there are connectors for most technologies, no issue is expected regarding the connectors.

3.4 Connection between ETL and PEP

The design of the connection between the ETL and PEP to send the processed data (events) needs to take into account that there might be communication or other unknown or unexpected issues, resulting in the event processor not being able to process all the events at the rate they are raised by the ETL. To overcome this situation and prevent any loss of information, there will be an interposition of a data queue system that will receive the events coming from the ETL and send them to the PEP, when the PEP is ready to process them. Then, once the PEP has processed the event, the queue shall be acknowledged to remove the event data from the queue. Both the ETL and the PEP will work on an asynchronous mode and the queue will retry sending the event to the PEP if the acknowledge message is not received.

3.5 Conclusion and Future Steps

To conclude, the design of the interfaces for the ETL, PEP and queue system is therefore pending the datasets definition. Nevertheless, ETRA has performed some testing of an ETL and suggested PEP in their own laboratory and confirms that these components are suitable for the proposed architecture and application.

4. Batch and Near-real-time Video and Image Analysis

4.1 Introduction

The following subsections show a detailed examination of the four visual components. First, the activity identification component is presented and then the person re-identification component is mentioned. In the third subsection, the face detection component is presented while the crisis event detection component is briefly explained in the final subsection.

4.2 Activity Recognition

The activity recognition problem has attracted the interest of the research community over the last decades and especially in the last years since computer vision techniques are applied to more and more research problems. A variety of security applications for recognizing activities have been proposed so far. Most of them have been motivated by the needs of security systems and are basically applied to multimedia content and especially take the advantage of processing the video footage. The visual recognition of activities covers a wide range of scenarios, from the recognition of car accidents to the detection of anomalies on crowd-centered scenes. Considering the PREVISION's needs, firstly, the recognition and subsequently the detection will be performed. Both individuals and vehicles are taking part in the target activities, while the classification of activity as abnormally or normal depends on the environmental parameters in which the activity occurs. Several methods have been proposed so far, to tackle the problem of activity recognition, while a lot of approaches for finding them within videos of long duration also proposed. In the next section, a summary of activity recognition techniques is reported.

4.2.1 Related Work

The rise usage of deep neural networks has led to activity recognition algorithms to report promising results. This section presents the recent works based on deep neural networks regarding activity recognition problem. Furthermore, a brief report of methods using only handcrafted features is presented. In addition, methods take into account multiple modalities, such as depth [151], are presented. An abstract representation of methods categorization is presented in Figure 11.

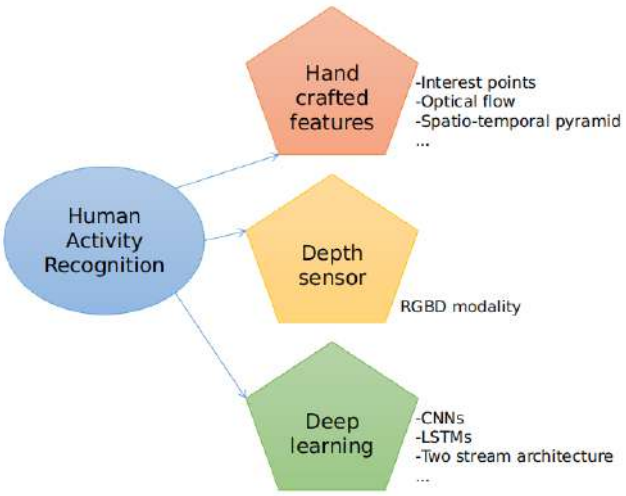


Figure 11. A categorization of human activity recognition methods.

4.2.2 Hand Crafted Methods

In hand crafted methods, a set of features are calculated using the raw pixel information and then the calculated features are taken into account to perform the recognition of the actions. The goal of the hand crafted methods is not only to extract visual features by a single figure, but also to examine the temporal information. To this end, many techniques have been proposed so far. The authors of [147] and [175] proposed key-frame based approaches that focus to detect the specific key-frames that could be indicative of performed activity. The authors of [128] and [43] proposed methods aimed at mapping the visual features of frames to words of vocabulary. The works such as [19] and [88] focused on techniques that aim to reduce the information of the extracted features. Finally, many approaches focused on the motion characteristics of the video frames. Specifically, the works such as [125], [152], [94] aim to estimate the motions among frames considering trajectory estimation, optical flow and histogram based techniques.

4.2.3 RGB-Depth Methods

The last decade, the wide range of the application of depth sensors such as Microsoft Kinect [181] has sparked the interest of the research community regarding the activity recognition problem. It is obvious that the hand crafted methods have a lack to capture activities that can be described by phrases such as “a person comes closer to the camera” due to the fact that these methods cannot capture the motion of the pixels. On the other side, RGB-Depth based methods not only take advantage of the processing of RGB information, but also exploit the depth information captured by corresponding devices. This led RGB-Depth methods generally to perform better compared to hand crafted approaches. An indicative method belongs to RGB-Depth category is [206] where the authors try to project the depth maps onto three orthogonal surfaces in order to generate depth motion maps before they extract the features using hand crafted techniques.

4.2.4 Deep Learning Methods

One of the first deep learning approaches regarding the activity recognition problem were proposed by the authors of [75]. They take advantage of the Convolutional Neural Networks (CNNs) and investigate the activity recognition problem from the scope of fusion of two visual streams, context, and fovea stream. Specifically, they examine four different fusion techniques: no fusion, late fusion, early and low fusion. They prove that the type of fusion between context and fovea information improves the accuracy of the baseline approach. The authors of [155] proposed a two-stream deep neural network architecture. The first stream is responsible for learning the visual appearance with RGB images given as input, while the second stream learns the motion taking as input the optical flow of two sequentially RGB images. The aforementioned methods are considered a breakthrough to activity recognition problem using deep neural networks.

In [44], the authors proposed a method regarding activity recognition problem using recurrent Long Short-Term Memory (LSTM) units. They make use both optical flow and RGB features - extracted using CNNs- and they prove that the usage of recurrent LSTM units are advantageous not only for activity recognition problems but on image and video description problems. Tran et al. [167] proposed a 3D-convolutional architecture (C3D) in order to give the ability to CNNs to capture both the spatial and temporal information without the necessity of optical flow information. Carreira et al. [22] presented a two-stream C3D architecture, called Two-Stream Inflated 3D ConvNet (I3D). Finally, Ullah et al. [169] proposed a bidirectional LSTM (BD-LSTM) network. Specifically, they process the video stream using

CNN for extracting features in order to reduce the complexity of the whole system and then these features are fed to a bidirectional LSTM network in order to learn the target activities.

4.2.5 Datasets for Activity Recognition

The number of datasets for evaluation activity recognition methods could be considered small due to the fact that they have a lot of effort to be annotated and complexity to be collected. The most used datasets for activity recognition problem are presented below:

UCF101 [157] is a dataset that consists of 101 activities described by 13,320 videos. The activities of the dataset are divided into 5 groups: Human-Object interaction, Body-Motion only, Human-Human interaction, Playing Musical Instruments and Sports.

HMDB51 [83] is a dataset was created from parts of movies, although videos from YouTube and other sources have been used. The total number of videos is 6,849 and each of them is mapped to one of 51 different actions. The minimum number of videos for each category was set to 101. Slimily to UCF101, the videos are divided into five categories: General facial actions, Facial actions with object manipulation, General body movements, Body movements with object interaction and Body movements with human interaction.

Kinetics 400 [22] is the largest dataset that was used on activity recognition domain. It consists of more than 306,245 videos divided into 400 classes. Each class consists of at least 400 videos.

ActEV [6] is a dataset that consists videos that describe interactions closer to the PREVISION needs. Human-Human interactions, Vehicle interactions and Human-Vehicle interaction are described. The dataset provided by National Institute of Standards and Technology (NIST), under the ActEV (Activities in Extended Video Evaluation 2019) series and is a part of the VIRAT-1 and VIRAT-2 datasets. For training and validation sets annotation is provided by the NIST team while the test set is still hidden. In Table 1 the number of videos and the extracted activities using the annotations provided by NIST is presented.

Table 1. ActEV dataset: training and validation sets properties.

	Training set	Validation set
Number of videos	64	54
Number of activities	1,338	1,128

The 18 activities of the ActEV dataset can be divided into four categories: Human-Vehicle or Human interaction (**HVH**), Human interaction (**H**), Vehicle interaction (**V**), and Human-Vehicle interaction (**HV**) as can be observed in Table 2. '**HVH**' consists activities that a person interacts with a vehicle or facility. '**H**' category describes the activities that performed only by a single person or among persons. Moreover, '**V** consists of interactions that performed by vehicles. Finally, the '**HV** describes the activities that a person interacts with a vehicle.

Table 2.ActEV activities official declaration.

Human-Vehicle or Human interaction	Human interaction	Vehicle interaction	Human-Vehicle interaction
Closing	**Activity carrying**	**Vehicle turning left**	**Closing trunk**
A person closing the door to a vehicle or facility	A person carrying an object up to half the size of the person	A vehicle turning left or right is determined from the POV of the driver of the vehicle	A person closing a trunk
Entering	**Specialized talking phone**	**Vehicle turning right**	**Loading**
A person entering (going into or getting into) a vehicle or facility	A person talking on a cell phone where the phone is being held on the side of the head	A vehicle turning left or right is determined from the POV of the driver of the vehicle	An object moving from person to vehicle
Exiting	**Specialized texting phone**	**Vehicle u turn**	**Open trunk**
A person exiting a vehicle or facility	A person texting on a cell phone	A vehicle making a u-turn is defined as a turn of 180 and should give the appearance of a "U"	A person opening a trunk
Opening	**Pull**		**Transport heavy carry**
A person opening the door to a vehicle or facility	A person exerting a force to cause motion toward		A person or multiple people carrying an oversized or heavy object
	Riding		**Unloading**
	A person riding a "bike"		An object moving from vehicle to person
	Talking		
	A person talking to another person in a face-to-face arrangement between n + 1 people		

4.2.6 Activity Recognition Framework

Regarding the PREVISION project, a supervised learning approach is implemented. We take the advantages of the 3D convolutional neural networks and deploy a 3D-ResNet neural network architecture in order to capture efficiently the video footage and subsequently recognize activities. Specifically, from the work of Hara et al. [62] we select the ResNet architecture consists of totally 50 layers as it has reduced complexity without sacrificing the performance ([62]:Table 2).

A 3D-ResNet architecture with 50 layers based on bottleneck blocks. Each bottleneck consists of: a convolutional layer with kernel size equal to 1, a batch normalization layer, a ReLU activation layer, a convolution layer with kernel size equal to 3, a batch normalization layer, a ReLU activation layer, a convolution layer with kernel size equal to 1, a batch normalization layer and finally a ReLU activation layer. The skip connection starts with the bottleneck block and merged before the final ReLU activation layer. The overall architecture of a bottleneck bloc is presented Figure 12.

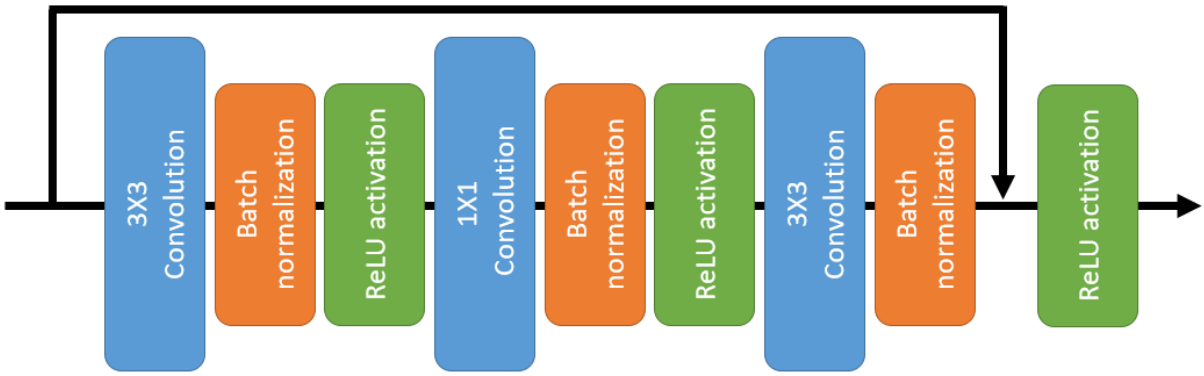


Figure 12. Bottleneck architecture of 3D-ResNet with 50 layers.

The deployed architecture makes use of the preloaded weights of Kinetics [22]. Regarding the training of the proposed model, the training set of ActEV [6] dataset was used. Specifically, for each video, we kept only the frames in which activities are annotated. In the selected frames, we performed a sampling every four frames. In order to ensure personal privacy (due to the fact that videos probably capture human faces in some cases) even if humans have been captured too far from the camera, we have applied an algorithm that detects faces, called TinyFace [232], and then a Gauss filter to blur them for each frame before saving is applied. These frames are used only for training and then are permanently deleted, while the weights (real-valued vectors) of the saved model are only kept. An indicative example is illustrated in Figure 13 where localized faces are blurred for a frame of an ActEV DataSet video. These frames are stored to a valid for processing format (.png). For training our model, we make use of the activities extracted only from the training test. The training parameters were tested and finally selected the following: batch size (32), total epochs (200), Stochastic Gradient Descent as optimizer, and reduce on plateau strategy was followed for reducing the initial learning rate (0.1). Moreover, data augmentation techniques were also performed. Specifically, five different scale-factors were used for generating a variety of cropping areas. More specifically, random crops were performed with scales to be set [1.0, ~0.84, ~0.70, ~0.59, ~0.49].

Figure 13. An indicative example of anonymization technique applied to a single frame that belongs to ActEV dataset

In order to verify the performance of the training process of the deployed model -before we demonstrate it using the test (non-annotated) data- we evaluate it using the validation data. The Precision@N was used as the evaluation criterion. In Figure 14 is illustrated the precision of recognized activities across the returned results, specifically the precision@1 exceeds 28%, while in cases of the target activity to be included in the top-3 predicted activities the proposed model reports 55% success.

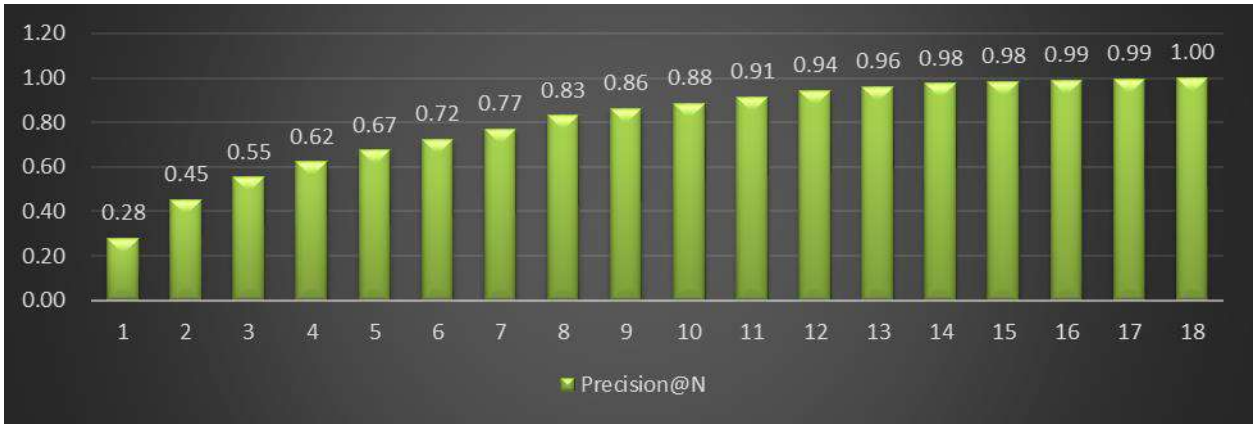


Figure 14. Precision@N, ActEV dataset evaluation using validation data.

4.2.7 Demonstration Tool

For demonstration purposes of the aforementioned framework, a tool for monitoring a video footage, selected from the test set, was implemented. The designed tool is a simple tool that shows to the end-user the real-time visual analysis of non-annotated data. In Figure 15 the visual footage that is fed to the activity recognition framework is depicted. In addition, the top-3 predicted activities during the time and the corresponding confidence scores are reported on the top left corner.

Figure 15. Snapshot of the visual activity recognition framework, video footage window.

The proposed tool consists of another window that simultaneously plots the outcome of the activity recognition framework. In Figure 16 the monitoring of probabilities is presented. On the right side, there is a list of the 18-predefined activities and the corresponding color. Axis y notes the predicted probabilities while on axis x each point represents the processing of 16 sequential frames (mini batches). Finally, during processing the predicted activities with a predicted confidence score more than a threshold equal to 10% are colored.

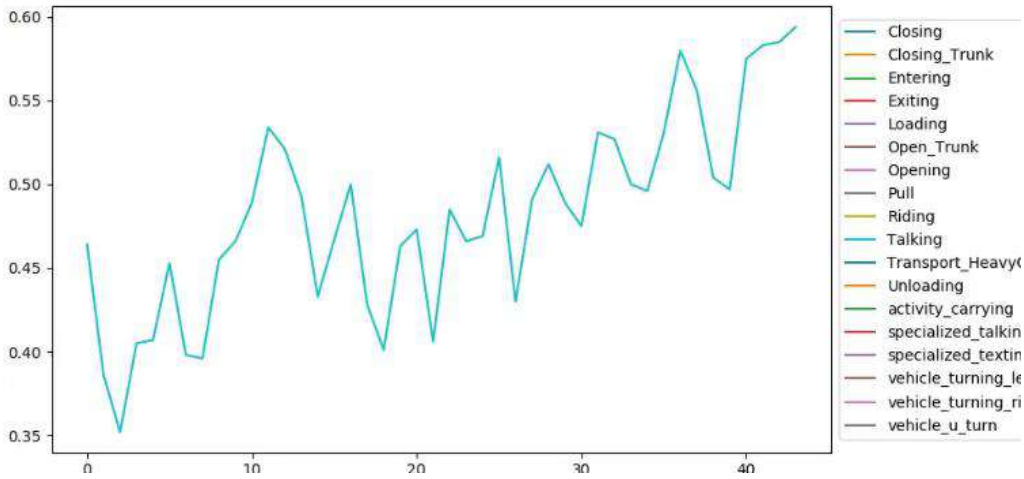


Figure 16. Snapshot of the visual activity recognition framework, probabilities monitoring.

A more convenient snapshot of the monitoring tool is illustrated in Figure 17. As can be observed, the video footage starts reporting the recognized activity “talking” and before the processing of the 300th mini batch the activity of “vehicle turn left/right” is captured. Furthermore, at the 400th mini batch and, approximately, 630th the same activities are recognized. Moreover, with blue are depicted the recognized activities at 660th and 780th proceed mini bathes where the activity “transport heavy carry” is predicted.

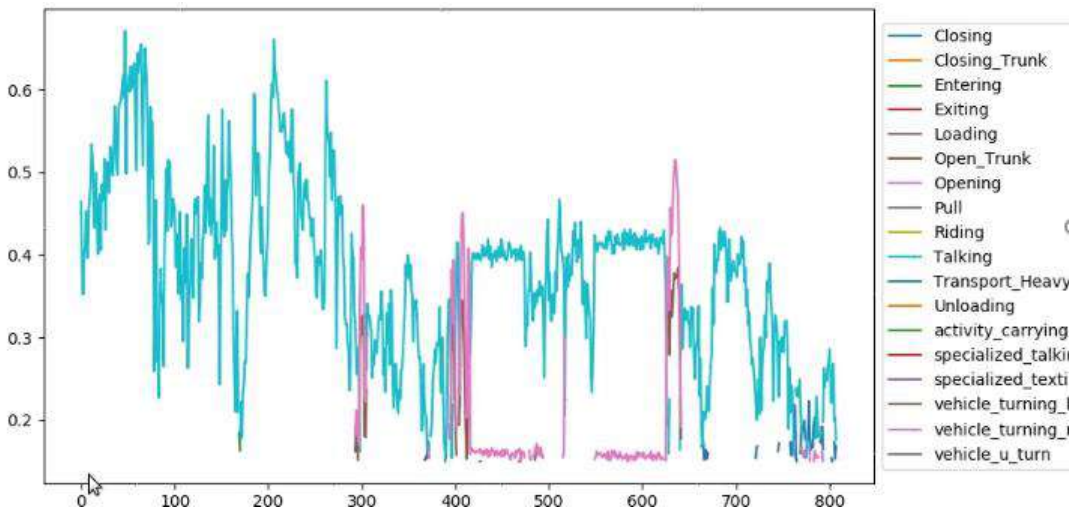


Figure 17. A snapshot of the monitoring probabilities is presented.

4.3 Person Re-Identification

In real-world surveillance scenarios, the automated re-identification of persons in a large amount of data recorded from surveillance cameras is an emerging topic. As more and more data are collected, it is not possible to manually search video and image mass data in order to find a specific person-of-interest. Especially when fast evaluation and reaction is required, this cannot be achieved by human effort.

Typically, so-called query images, which show the person to search, are used to find other occurrences of the same person in mass data, called gallery. However, this approach comes with the problem that a suitable image of the wanted person must be available. Otherwise, it would not be possible to perform an appearance-based retrieval. A further possibility to describe appearances of people is the use of semantic attributes. Such attributes can be the gender, descriptions of the clothing or information about accessories the person is carrying. A big benefit of this method is that semantic descriptions of persons can be extracted directly and easily from testimonies and then can serve as a search query. Therefore, witness statements are sufficient to start a retrieval and images are no longer required. Since this is an important task in practice, this work deals with this type of person re-identification which is referred to as attribute-based person re-identification in the following.

Various challenges influence the performance of such person re-identification systems negatively. Among the most challenging factors are occlusions, different view angles or low image resolution. Some example images are shown in Figure 18.



Figure 18. Examples for challenging factors that aggravate the task of person re-identification. From left to right: Occlusions, bad illumination and low image resolution.

Whereas poor illumination or low-resolution images affect both global image embeddings typically used in person re-identification and semantic attributes as retrieval features, semantic attributes are higher-level features and thus less dependent on e.g. viewing angles. In addition, some attributes may still be visible if a person is partly occluded.

Another aspect that should be considered to achieve a robust person re-identification system is that not just single images are recorded by surveillance cameras but videos instead. Persons are not only visible in one frame, so that whole tracklets of persons can be exploited and the influence of challenges like occlusions or different viewpoints can be minimized. So far, some work has been published on normal video-based person re-identification with image queries but only few regarding video-based pedestrian attribute recognition, which is the crucial part in an attribute-based person re-identification system. Therefore, in this project the inclusion of the temporal aspect for pedestrian attribute recognition is investigated in detail.

For this purpose, machine learning methods, more specifically convolutional neural networks (CNN), are applied. These are data-driven methods, so the availability of data is the critical factor in achieving good performance. For attribute-based person re-identification many images of many different people are needed, which leads to various problems in practice. Consent declarations of all persons appearing in the dataset are necessary to avoid privacy issues and great efforts must be made to manually annotate the data. All these problems can be overcome by generating and using synthetic data. No real persons occur in such simulated images and most annotations can be retrieved directly from the rendering engine. Furthermore, theoretically, infinite amounts of synthetic data can be generated. However, very realistic data is needed to be able to bridge the gap between synthetic and real-world domains.

Our contributions addressing the afore-mentioned points are the following:

- We develop a modification in GTA V that allows recording multi-camera surveillance scenarios;
- We create a synthetic dataset for video-based person re-identification;
- We thoroughly evaluate different strategies for video-based pedestrian attribute recognition;
- We compare these strategies with respect to their attribute- and video-based person retrieval performance;

First, related work with respect to pedestrian attribute recognition and re-identification is presented as well as an overview of datasets that can be used. Subsequently, our simulated dataset is described in Section 4.3.2, followed by Section 4.3.3 in which the person re-identification framework is detailed. Before the concluding summary, the results of the experiments are presented and discussed in Section 4.3.4. Note that for this initial report, experiments were conducted using real-world datasets only. The results presented here serve as a baseline for comparing methods that leverage synthetically generated data. Such experiments and evaluations will be part of the next project phase.

4.3.1 Related Work

This section presents literature on attribute-based re-identification of persons in videos (see Section 4.3.1.1) and datasets for this task (see Section 4.3.1.2).

4.3.1.1 *State of the Art Methods*

Nowadays, most image and video processing tasks are performed using CNNs because they achieve state-of-the-art accuracies in many domains. This also applies to pedestrian attribute recognition and person re-identification, respectively.

Generally, various approaches are possible to process video frames instead of single images in order to recognize person attributes for re-identification. A very straightforward way is to replace the convolutional backbone network that usually takes 2D images as input with 3D CNNs [166]. In contrast to 2D models, the convolutional kernels of 3D CNNs are expanded to three dimensions. Another possibility is the integration of so-called non-local blocks that can also learn temporal dependencies within a CNN [174]. These methods implicitly learn temporal aspects but require adaptions to the backbone architecture. In context of pedestrian attribute recognition, image-based 2D models can also be directly applied by processing each input frame separately and performing a temporal pooling or temporal attention operation on resulting features or attribute predictions. This is the procedure of the, so far, only published approach in literature to video-based attribute recognition [223]. The authors suggest extracting global features for each input frame first and then applying attention modules separately for each attribute to aggregate temporal information.

Since there is not much related literature available, our goal is to compare different approaches with respect to recognition and retrieval accuracy as well as efficiency to create a framework that can be used for near-real-time video analysis.

4.3.1.2 *Datasets for Person Re-Identification Methods*

The basic idea of attribute-based person re-identification in videos results in two main requirements for the datasets. On the one hand, image sequences must be given and on the other hand, both attributes and person id labels must be available. Existing datasets for pedestrian attribute recognition, which is the crucial task in attribute-based person re-identification, are mainly image-based and thus meet only the second requirement. Examples for such datasets are PETA [212], RAP [95] or Market-1501 [186] [99]. However, in the domain of person re-identification based on query images two large-scale datasets that provide image sequences instead of single person images are available, namely Motion Analysis and Re-identification Set (MARS) [185] and DukeMTMC-VideoReID [150]. Fortunately, attribute annotations are also available for each person identity contained in these datasets [223], which generally enables the use of these datasets in this work. Of these two datasets, however, only MARS can still be used since DukeMTMC-VideoReID was taken offline due to privacy issues.

MARS builds on the same data basis as the well-known person re-identification dataset Market-1501, but contains tracklets, i.e. short image sequences, instead of person images. In total, MARS consists of 20,478 person tracklets from 1,261 different people. The dataset was captured using a network of six surveillance cameras. Since originally only person ids were provided, the authors of [223] labeled 16 types of further annotations. These include motion and pose information as well as instance-related semantic person attributes like gender, clothing style or colors and information about accessories, e.g. backpacks. To ensure privacy and to make it impossible to identify persons contained

in the MARS dataset, we anonymized the images. For this purpose, faces were detected with the TinyFace [232] detector and then blurred with a Gaussian blur filter.

Since there is currently only one real-world dataset available for this topic and no synthetic dataset yet, we have decided to create our own dataset. Similar to [48] we wanted to simulate the data in order to avoid problems with privacy and to increase the diversity of data. Moreover, leveraging synthetic data may help improving the performance of person re-identification system in real-world applications.

4.3.2 Simulated person tracking and re-identification dataset

This section deals with the creation of our synthetic dataset for attribute-based person re-identification. First, the generation of the data base is described followed by the creation of the dataset with corresponding annotation.

Our goal was not only to produce a dataset, but instead to develop a generator that allows us to create datasets flexibly according to specific requirements. At the same time, the data generated should reflect reality as accurately as possible. To achieve these goals, we developed a modification in the video game GTA V (Rockstar North). The advantage of the use of an already existing video game is that person and scenes models already exist as well as natural behavior of e.g. persons or cars. Thus, effort for manual planning of scenes can be reduced to minimum and long-term recordings without additional effort can be done. We build our work on [48] in which a pose dataset was created but only with a single static camera. Since typical surveillance scenarios consist of camera networks with multiple different cameras and the task of person re-identification typically involves the cross-camera aspect, we extended their work and included the functionality to record frames of multiple cameras of the same scene simultaneously.

For this project, we decided to record a dataset in front of a shopping mall. In total, six different cameras were placed in the scene. The concrete camera setup is depicted in Figure 19.

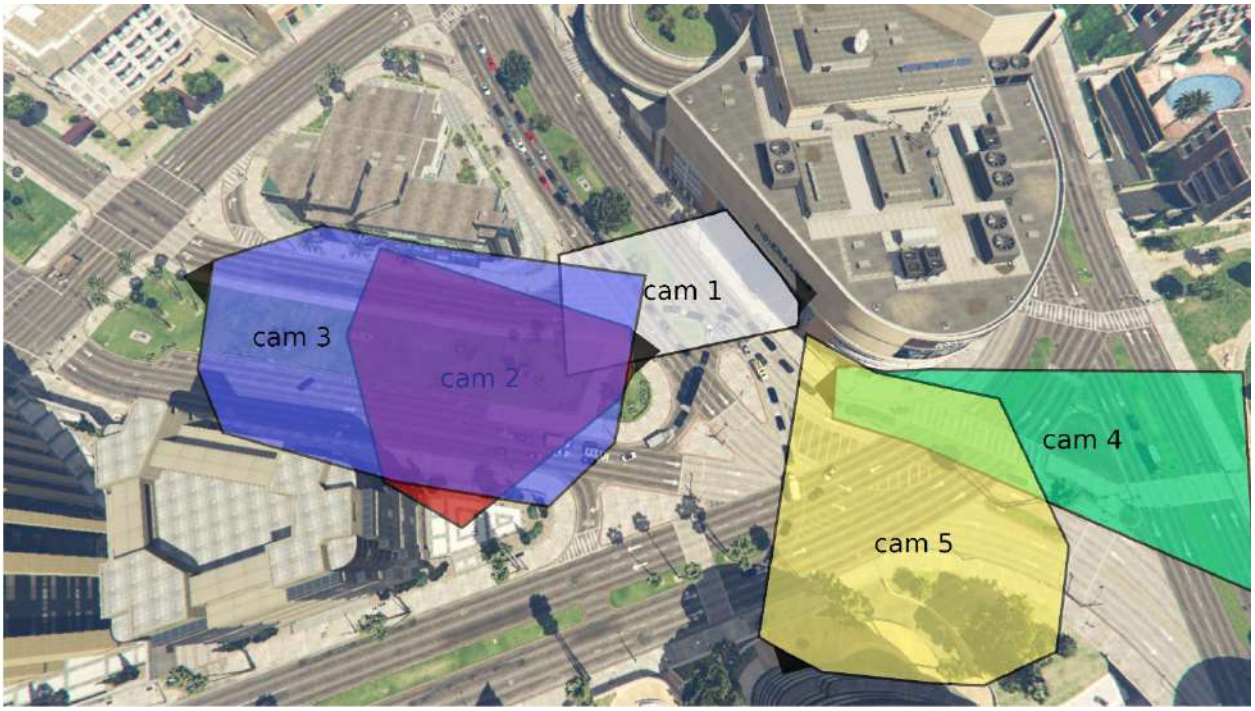


Figure 19. Scene overview. Since one camera is positioned inside a metro station, only five camera footprints are shown. It can be seen that there are overlapping as well as non-overlapping cameras.

In contrast to existing datasets, we used five outdoor and one indoor camera during the recording, which is why only the positions and footprints of five cameras are shown in Figure 19. This will help to develop less scene and lighting dependent approaches to evaluate methods regarding their robustness against these challenges. In addition, cameras were placed so that overlapping and non-overlapping field of views emerge. This is not directly relevant to the task of re-identification, but gets important if the dataset is used in other domains, e.g. if the paths of persons should be tracked across the camera network. Such design choices allow the created dataset to be applied to a wide range of methods and use cases.

In Figure 20, the views of the six cameras are displayed. The field of view of camera 0, which is shown in the top left image, records the inside of a metro station while other cameras show outdoor scenes from the metro station entry, park areas or sidewalks along roads.

Figure 20. Camera Views. From left to right and top to bottom: cameras 0-5

Because we wanted to use the dataset for person re-identification, it is necessary that the appearances of people in the game differ. Since the game produces persons with random appearances, and therefore the possibility exists that people with the same appearance but different identities appear, we decided to take control of the appearances of characters spawned. The person appearance system works with a base model of a person that has properties like clothing, beard, etc. that can be adapted. We treated one appearance as one person. Since the influence of certain editable components of the appearance system is not well documented, we captured automatically generated persons and selected a pre-selection consisting of 4,514 people. These appearances have been used to spawn people which walked tracks and were recorded for the dataset.

In order to obtain different tracks of people that go through multiple cameras, we defined creation and deletion spots outside the camera views and created a network graph. Besides these start and end positions, the graph contains nodes at intersections and edges between nodes if they are connected by a path. The track generation is done by generating all simple paths between a start node and an end node. A simple path contains no node twice, except the start node or end node. This means that there will be no loops contained in the generated tracks. As our graph consists of eleven start or end nodes, respectively, simple path algorithm resulted in 3,119 possible tracks. When a person is created during dataset recording, a track, the direction in which the track is passed and the velocity are randomly chosen. After a person reached the end of a track, it was deleted and not spawned again.

In the following, general facts about our resulting raw version of the dataset will be given. A brief overview is provided in Table 3.

Table 3. Facts and statistics about our MTA dataset.

MTA	
Cameras	6 (1 indoor + 5 outdoor)
Resolution & framerate	1920x1080 pixels @41 fps
Duration of videos	6x 122 minutes
No. of persons	3,221
Conditions	Sun & rain, day & night
Camera characteristics	Simulated lens distortion
Annotations	34 types (+ attributes)
Person bounding boxes	37,324,348

Video frames were recorded with Full HD resolution at a frame rate of 41 frames per second. The duration of the videos for each camera is slightly above two hours and in total 3,221 different persons walk through the scene during this time window. Note that not all persons were visible within the recorded time window and camera setup and thus not all pre-selected appearances occurred. In contrast to existing datasets, weather conditions changed during recording. Sunny periods as well as rainy phases alternate in the dataset videos. In addition, several day and night cycles were simulated, which is not the case in current datasets from literature. The course of the time of day is shown by the brightness in Figure 21.

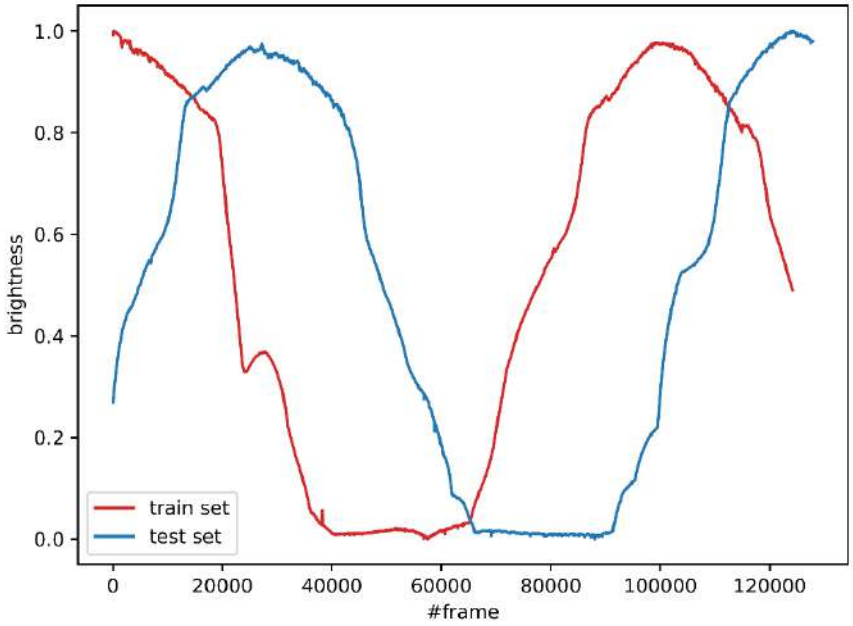


Figure 21. Brightness curve to measure the time of day over the train and test split of our dataset. Multiple day and night cycles were passed.

As mentioned before, simulating data has the advantage that many different annotations can be extracted directly and less manual labeling is required. In our case, 34 different types of annotations were recorded automatically. These include person identifiers, person pose key points or person bounding boxes. An overview is given in Table 4.

Table 4. Overview of automatically recorded annotations.

Category	Description
Frame number	Overall frame numbers and camera-wise frame numbers
ID numbers	Person identifiers and appearance identifiers
Pose key points	Information about the positions of 22 different body joints
Occlusion	Information about the occlusion of person body parts
Camera information	Information about global 3D camera positions, rotations and field of views
Person position	2D and 3D positions of persons
Person rotation	Rotation information about a person's yaw rotation
Semantic person attributes	Some attributes like gender or glasses

The large number of different annotations allows the dataset to be used in a wide range of applications. Since our focus is on attribute-based re-identification of persons, person bounding boxes are most important. In total, more than 37 million such boxes are available. As can be observed from Figure 22, the partly flat view of the cameras leads to both very large and very low-resolution images of persons. Additionally, Figure 23 visualizes the distribution of bounding box regarding their height in pixels. Note the logarithmic scale on the y-axis denoting the bounding box frequency. One can see that there are a lot of small person images with a bounding box height of less than 50 pixels.

Figure 22. Randomly selected person bounding boxes from our dataset. Original ratios of image sizes have been preserved.

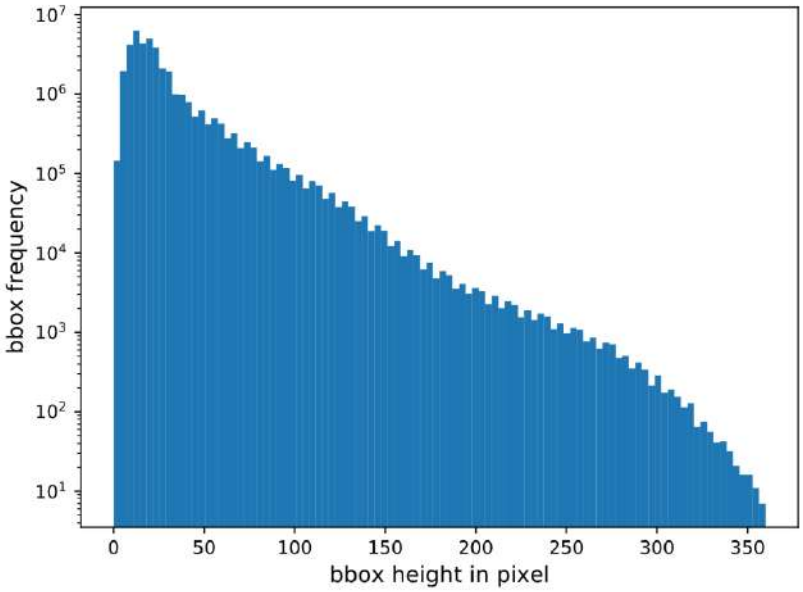


Figure 23. Distribution of bounding box heights.

In order to have a train and test set, we divided the videos and annotations in the middle. The first part should become the train set and the second part the test set. One problem which existed was that persons from the train set walked into the test set, which is not acceptable as the persons in the train and test set should be strictly separated. Therefore, we calculated the intersection of the person identifiers from train set and test set from all cameras and chopped out the time span in which such persons appeared. As a result, the train set has a length of 50:29 min and the test set has a length of 51:59 min.

To be able to use the person bounding boxes for attribute-based person re-identification, we had to prepare tracklets. Therefore, we extracted image patches using the bounding boxes of every frame and merged them into tracklets if persons were present in multiple consecutive frames. Since in practice we cannot assume perfect trackers and thus tracks, and since our goal was to create a dataset that is as realistic as possible, we closed tracklets as soon as a person was no longer visible for some frames, e.g. due to occlusions. We determined the level of occlusions or visibility, respectively, by evaluating the ratio of visible to invisible key points. In doing so, we only considered occlusions originating from other persons or objects and excluded self-occluded pose key points. After the processing of all video frames of training and test splits we took out 20% of the test set tracklets to create a query set. These preparations resulted in a video person re-identification dataset the statistics of which are presented in Table 5.

Table 5. Statistics of our MTA Person Re-identification dataset.

Subset	No. of IDs	No. of tracklets	No. of cameras	Duration (min)
Train	1,361	80,233	6	50:29
Test	1,475	76,871	6	51:59

To harden the task of person re-identification, so-called distractor images are typically included in datasets. Distractors are images that show non-relevant objects or only small parts of persons and thus images that are irrelevant for person retrieval. Distractors help to evaluate the robustness of

approaches against disturbing influences that occur in practice, such as poor person detections. We have determined our distractors by using those track images that were excluded during tracklet generation due to non-visibility of persons. Samples are shown in the following Figure 24.

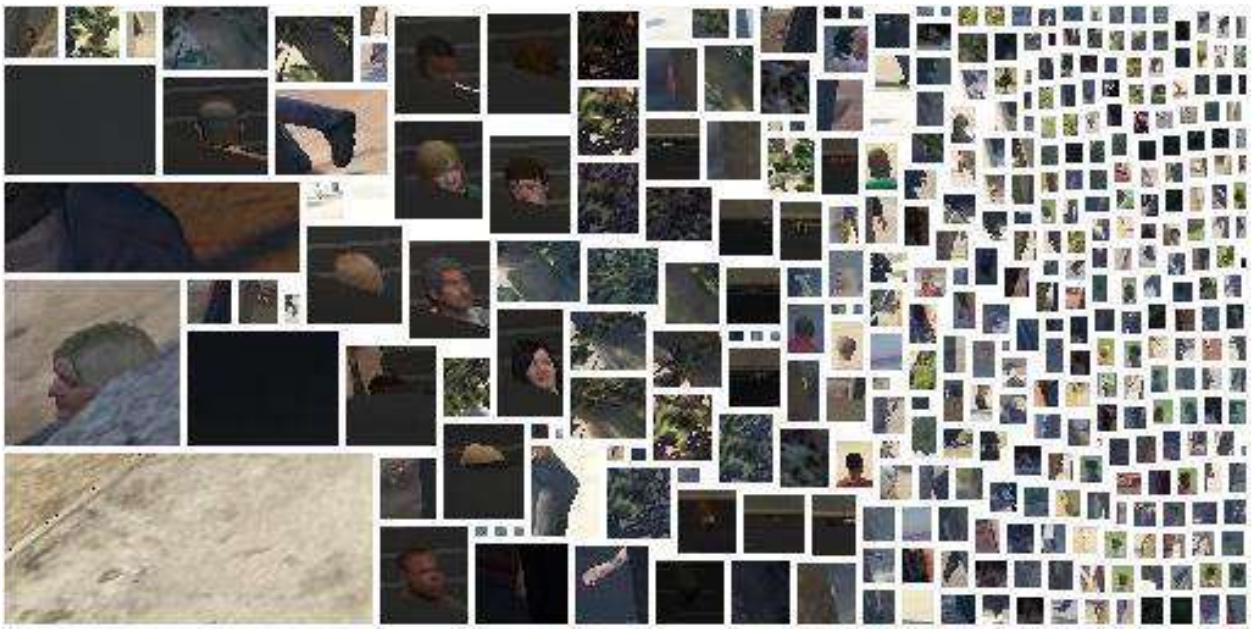


Figure 24. Distractors from our MTA Person Re-identification dataset.

Unfortunately, only two semantic attributes, namely gender and glasses, can be directly extracted from the engine. So, other attribute labels have to be annotated manually. Since people's appearances are fixed, they remain consistent in all tracks and across all cameras, and it is sufficient to provide attribute annotations on an instance-based level even if not all attributes are be visible in all frames. The temporal aspect minimizes possible negative influences, forcing CNNs to use information over time to correctly recognize all attributes. When selecting relevant attributes, we oriented ourselves on existing datasets and tried to combine them sensibly. Our final selection, which is not yet fully annotated, is shown in Table 6. Both global attributes such as gender or age as well as local attributes like hair length or shoe color are annotated.

Table 6. Attribute annotations.

Class	Attributes
Gender	Female / male
Age	Young / old
Body shape	Slim / fat
Hair	Length, color
Head and shoulder accessories	Hat, glasses, scarf
Upper-body clothing	Length, color, fit, type, pattern
Lower-body clothing	Length, color, fit, type, pattern
Shoes	Color, type
Attachments	Backpack, handbag, bag

In conclusion, we created a modification in GTA V that allows creating and recording multi-camera surveillance scenarios with maximum flexibility and maximum proximity to the real world. In addition,

the first synthetic dataset has been created that can be used for a variety of applications ranging from attribute recognition and re-identification of persons to multi-target multi-camera tracking approaches.

4.3.3 Person Re-Identification framework

In this section, the methodology of our attribute-based person re-identification framework is detailed. First, the general procedure of attribute-based person re-identification is described followed by the explanation of our approach.

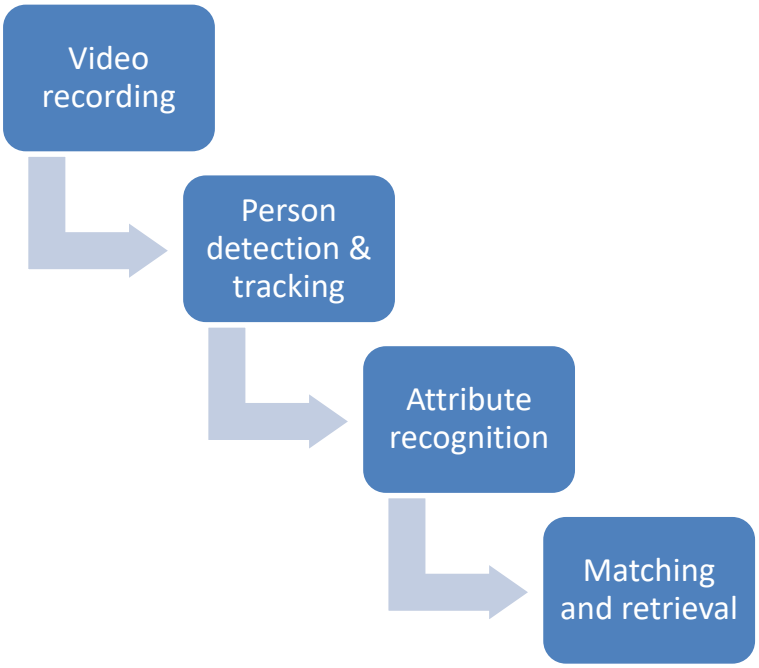


Figure 25. General pipeline for attribute-based person re-identification systems.

The general pipeline is shown in Figure 25 and includes three substantial processing steps. Input videos are fed into a person detector in order to find all persons present in the scene. For this task, many suitable detectors already exist which achieve detection accuracies beyond 90 %. For this reason, and because most datasets already contain cropped images of people, we do not focus on this step. If it should be necessary to detect persons first, we can rely on one of the state-of-the-art detectors that can be used off-the-shelf. If person re-identification should be video-based, tracking is also necessary. However, there are sufficient approaches in literature, e.g. DeepSORT [219], which can be used for this purpose. The next step is the most important in terms of retrieval accuracy, so our approach will focus on this. Attributes must be recognized for all persons detected in the previous step. Finally, yet importantly, the actual matching and retrieval is performed. Therefore, a metric is applied to compute distances between the query attributes and the extracted attributes of the gallery images or tracklets. By sorting them according to their distances, a rank list is created with best matches in early positions.

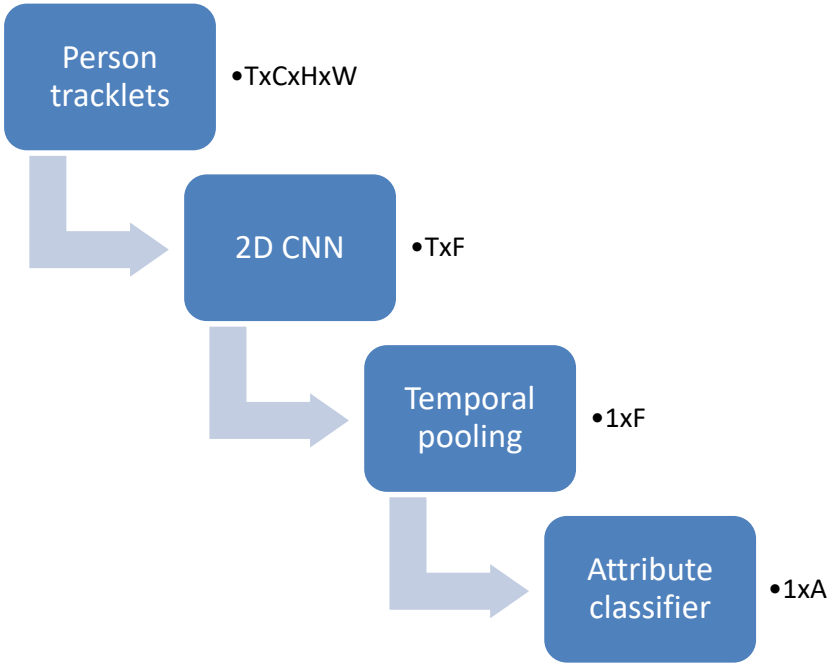


Figure 26. Our proposed approach for tracklet-based attribute classification.

In general, various approaches are possible to predict a person's attribute based on a tracklet. The naïve procedure would be to process each frame separately and to use the mean or maximum as tracklet-level predictions. However, this means that frames are considered completely independent of each other. Hence, we have decided to apply temporal pooling already to global frame features output from the 2D backbone CNN. Tracklets of size *T x C x H x W* are forwarded frame-wise through a 2D CNN resulting in *T* feature vectors of size *F*. *T* denotes the number of time steps i.e. frames of the tracklet. *C x H x W* stand for the size of the tracklet frames, i.e. number of channels (*C*), height (*H*) and width (*W*). We then pool the features along their temporal dimension and obtain a single feature vector for the entire input image sequence which in turn is used as input for the actual attribute classifier. The classifier consists of two fully connected layers, the size of the latter corresponding to the number of attributes *A*. For loss computation, sigmoid cross-entropy loss function with sample weighting [214], [214] is used. The weighting helps to deal with the problem of imbalanced distributions of attributes in the dataset. Person samples with very rare attributes receive a higher weight and thus, the training process is forced to focus on such training samples.

We will later compare this approach against other methods like 3D CNNs and the state-of-the-art in the evaluation section.

4.3.4 Experimental evaluation & results

In this section, experimental results are presented and discussed. First, information about the dataset used and some implementation details are given, followed by the quantitative results of our experiments.

4.3.4.1 *Evaluation dataset*

As dataset for evaluation of our person re-identification framework, MARS Attribute dataset (see Section 4.3.1.2) was chosen. Note that since not all attributes are annotated yet, only results for MARS

are reported in the following. Experiments using our simulated dataset will be carried out in the next steps.

Authors of the MARS Attribute dataset propose label-based mean accuracy and F1 score as evaluation metrics for pedestrian attribute recognition task. Since instance-based metrics are more common and more significant with respect to person re-identification, we additionally report these metrics. While label-based scores consider each attribute separately, instance-based scores focus on all predictions for a particular person.

For person re-identification, mean average precision (mAP) and ranking accuracies are usually given. We do the same and consider all persons that match the attribute query as matches.

4.3.4.2 *Implementation details and training parameters*

We conducted our experiments using the *PyTorch* framework. Unless otherwise specified, ResNet-50 model was used as the backbone architecture. The number of frames considered was set to 15, so tracklets were sampled to 15 frames during training. The learning rate was set to 1e-4 and reduced by a factor of 0.1 after every 20 epochs. In total, the networks were trained for 60 epochs.

4.3.4.3 *Evaluation results*

First, the difference between pooling strategies and pooling stages is evaluated. Results are presented in Table 7.

Table 7. Evaluation of temporal pooling.

Temporal Pooling	Pooling	mA (label)	mA (instance)	F1 (instance)
Global feature pooling (GFP)	Maximum	83.91	73.62	80.66
	Average	**87.53**	**79.03**	**84.79**
Prediction pooling (PP)	Maximum	81.97	76.31	80.59
	Average	86.70	74.43	83.75

The results show that pooling global features instead of predictions achieves higher scores in all metrics. While in regards of instance-based mA a 1.6 percentage points higher result is achieved, about 0.8 percentage points are gained in F1 measure. Another finding is that average pooling of the temporal dimension leads to better recognition of attributes. This is an intuitive result since the goal of temporal pooling is to combine the information from multiple frames in order to bridge gaps in which some attributes may not be visible. By computing the average feature vector, such small gaps are filtered out. In contrast, maximum pooling focuses on the most prominent global features. Thereby, features from such gaps may be chosen because the great difference to the other ones. While the influence would be suppressed by average pooling, maximum pooling could erroneously focus on these frames.

Table 8. 2D vs. 3D models

Model	mRA	mA	F1
Temporal Attention [223]	87.01	-	-
GFP	**87.53**	**79.03**	**84.79**
ResNet 3D	83.02	70.96	79.84
ResNet MC	86.34	73.88	83.57
ResNet 2+1D	86.54	75.22	83.84

Table 8 compares different 3D model architectures (for details see [166]) with the image-based global feature pooling approach and the current state-of-the-art approach. It can be observed that none of the 3D models can exceed the baseline established by the GFP approach. A reason for this is that attribute predictions are not dependent on the capturing of movements of persons, which is the strength of 3D models. For example, when considering classification accuracy for the motion attribute, the best 3D model achieves 94.7 % accuracy while the 2D pooling model scores 1 percentage point less. For the instance-related attributes, like gender or clothing, pooling approaches are superior. Moreover, the GFP model establishes a new state-of-the-art for video-based attribute recognition on the MARS dataset. In terms of computation time, it can be stated that the GFP approach requires only about half the FLOPS for the forward pass compared to the best 3D model while achieving better recognition performance. In addition, no attention modules are needed in contrast to the method described in [3] which also leads to fewer parameters and computation effort.

Finally yet importantly, we evaluate the approaches regarding their suitability for attribute-based person re-identification. Results are presented in Table 9.

Table 9. Attribute-based person retrieval results.

Approach	mAP	R-1	R-5	R-10	R-20
PP	**38.8**	40.5	**69.2**	**79.0**	**85.5**
GFP	37.2	**40.6**	66.8	76.3	83.5
ResNet 2+1D	32.7	38.1	64.9	74.2	80.9

The most important finding is that in contrast to attribute recognition accuracy, prediction pooling achieves the best retrieval results. Thus, if attribute-based person search is the main objective, it is beneficial to rely on prediction pooling rather than global feature pooling, although the latter achieves significantly higher attribute recognition performance. The reason for this is that attribute metrics are calculated after hard decisions have been made about the presence of attributes, whereas retrieval is based on distances between prediction confidences and query descriptions. Analogous to pedestrian attribute recognition results, 3D models cannot outperform the presented temporal pooling models.

4.4 Face Detection and Recognition

4.4.1 Related Work

In the sections below (4.4.1.1, 4.4.1.2) the related work of face detection and face recognition approaches is presented, correspondingly.

4.4.1.1 *Face Detection*

Face detection has always been a widely researched computer vision task. Even in early face detection techniques, handcrafted shallow representations, like the Haar cascades [248], or robust features like SURF (Speeded-Up Robust Features) [237] and Histograms of Oriented Gradients (HOG) [242] or Local Binary Patterns (LBP) [225], were manually designed in order to detect faces. However, those methods imply the use of exhaustive sliding window search techniques so as to discover candidate boxes of faces. This requirement made them extremely demanding in computational resources. Thus, later works focus on faster and more accurate techniques. To this end, efficient facial point localization methods were proposed, such as the mixtures-of-trees [254] and consensus of exemplars [226] that required much less computational time. Those methods, deploy several local detectors for parts of the face, or key-points, and then combine the results with fusion.

Soon thereafter, with the rise of deep neural networks, and especially in the tasks of image classification and object detection, a breakthrough in performance can be observed. The work of [245] proposed deep convolutional network cascade, which achieved high detection rates, on some popular challenging face detection benchmarks like LFPW [226] and YouTube faces [250]. Using facial key-point detection as a primary goal, they proposed a cascade of networks that refined facial key-point location in each level, fusing the output of multiple networks, whilst implicitly encoding geometric constraints between the points. Additionally, other methods were proposed that dealt with face detection as a generic object detection task, like [235] and [244]. These methods, obtained high-accuracy generic object detection CNNs and fed them with annotated face boxes for training, while at the same time, applied techniques like hard negative mining, feature concatenation and careful fine-tuning to improve their results.

More sophisticated works considered the spatial structure and arrangement of facial parts [251]. Several CNNs were trained in this work, each one dedicated to the detection of a single face part, while early convolution features maps were shared to improve the efficiency of the method. Moreover, the concept of “faceness” score was introduced, which was a measure of “what constitutes a face”. An object proposal ranking stage was calculated using “faceness” by determining how well each proposal met the structural constraints posed by the detected facial parts. Later, the framework proposed by [253], leveraged a cascaded architecture with three stages, each one deploying a deep CNN. Object proposals were extracted during the first stage using a Fully Connected Network (FCN), false positive candidates were filtered using another CNN in the second stage, and further refinements were performed based on facial landmarks during the final stage.

Simultaneously with the object detection works found in the literature at that time, single-shot architectures used in generic object detection found their way into the face detection research interests. For instance, the SSH (Single-Shot Headless) face detector [239], alleviated the need for a face bounding box proposal generation step. Different layers of an FCN were deployed, so as to predict various scales of faces in a single forward pass. The paper of [240], proposed a framework to deal with relevant tasks simultaneously, i.e. automatic facial landmark detection, pose estimation, gender recognition and face detection. This method was categorized as region-based, in essence working on patches of the image (candidates). Deep convolutional features taken from different layers were first fused and then fed to a five-headed output, where each "head" was a Fully Connected (FC) network dedicated to a task. The problem of detecting small objects, and thus faces as well, was the focus of [232]. Several key insights for small scale face detection, like the information in surrounding context

of boxes and the use of large receptive fields, were established. They found that training multiple detectors for various scales produced state-of-the-art results, while applying feature sharing between layers of the CNN hierarchy allowed maintained the efficiency. More recently, inspired by Feature Pyramid Networks, multi-scale features were extracted and high-level feature maps of various scales were aggregated to argument low level feature maps as contextual cues in an agglomeration manner with a hierarchical loss to train the pipeline in the work of [252].

4.4.1.2 *Face Recognition*

Face recognition (FR) has always been an extremely active topic in the computer vision domain. FR includes all the algorithms and techniques designed for face identification or verification. Face identification refers to the problem of classifying a face to a certain identity. Face verification, on the other hand, is the problem of determining whether or not a pair of faces belong to the same identity. Generally, the first step of FR is face feature extraction and representation. Then, the resulting vector is either classified to an identity, or the minimum distance to the gallery vectors is found and a match is confirmed. Thus, FR can be viewed as a classification problem or a metric learning problem depending on the testing settings. Specifically, in an open-set setting, test identities might not appear in the training set, therefore, FR resolves to metric learning. On the contrary, in a closed-set setting, where test identities exist in the training set, a classifier is the proper way to solve the task.

It is evident that face feature extraction and representation is the most crucial step in a FR framework. Early works built low-level descriptors, such as LBP, SIFT, or CMD, which were then combined with a shallow model for identification, such as SVMs, discriminative dimensionality reduction or Fisher encoding [228], [229], [233] and [243]. Later works depend heavily on deep learning methodologies to achieve significant boost in performance. In this class of algorithms, deep feature extractors are used to generate face representations, which are trained with the aim to acquire invariance to pose and illumination from the plethora of the available training data, rather than from low-level hand-crafted features.

Siamese networks for deep metric learning were proposed in the work of [230], which was one of the initial attempts to leverage deep learning. A Siamese network works by extracting features separately from two modes (inputs), with two identical CNNs, taking the distance between the outputs of the two CNNs as dissimilarity. In the work of [255] it was proposed to warp faces from arbitrary pose and transform them to frontal view with normal illumination using a trained deep neural network and then used the last hidden layer to get face representations. Other approaches, involving multi-stage networks, aligned the faces using 3D modeling first and then compared them in a multi-class network for identification [247].

In a similar fashion with Face Detection methods, facial parts were processed separately in cascade networks as in the work of [246]. Soon after, the focus shifted heavily towards approaches based on deep metric learning that led to significant performance improvement. Experimentation, therefore, with different metrics was the primary activity of those works. Discriminant face representations are characterized by smaller maximal intra-class distance and minimal inter-class distance in the embedding space. Therefore, recent works meticulously explore and experiment with several loss functions for CNNs, so as to find the most appropriate for this task [231], [238] and [249].

4.4.2 Face Detection and recognition framework

Our Face detection and recognition framework consists of a generic object detector that is able to detect faces amongst other objects, a baseline face recognition feature extraction network and an SVM that is used for identification as depicted in Figure 27. For our face detector, we chose to adopt the Faster R-CNN with Resnet V2 architecture pre-trained in the Open Images V4 (OI4) dataset [236]. This model achieves very good accuracy rates (i.e. on average 54% mAP across all classes[2]). The OI4 is a collection of images that contain bounding box annotations for 600 object categories. During a forward pass, from the 600 categories we select to only keep instances that are predicted with a "face" label above a threshold of 0.5, so as to detect candidate face bounding boxes in the image.

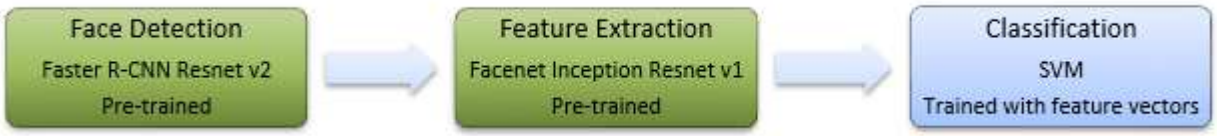


Figure 27. Face Recognition Pipeline

Several benchmark databases have been made publicly available for FR evaluation. The most popular database used in the literature is the "Labeled Faces in the Wild" (LFW) [234]. It contains nearly 13K images of over 5K people. The FaceNet embeddings proposed by [241] have shown very high accuracy for FR in this dataset. The authors propose a deep CNN feature extractor, coupled with L2 normalization, leads to an embedding which is trained in an end-to-end manner using a triplet loss function. Triplet loss works by minimizing the distance between anchors and their corresponding positive samples, (i.e. same identities), while maximizing the distance between anchors and their corresponding negative samples (i.e. different identities). The model can then be trained on a large-scale database, so as to learn meaningful face embeddings, using the above optimization goal for distances in Euclidian space, with the aim to minimize the distances of same identity instances in Euclidean space. Thus, for our FR module, we harness the FaceNet's discriminative power of embeddings.

In a real-world application an FR system cannot exist without incorporating the concept of the "unknown" identity, since there can only be a finite number of known identities to the system. Therefore, a small-scale database, or gallery, containing the faces of familiar identities must exist. Ideally, the gallery must include several images per person. Empirically, we set the minimum number of images per person to be 10, and we select 10 identities from the LFW that fulfill this requirement. The photos are taken from public figures and celebrities and they are shown in Table 10. In order to protect the identities, we anonymize our gallery, by assigning a unique ID number to each one instead of his/her real name. It is worth mentioning here that in order to increase the robustness of an FR system, the set of images for each identity should contain samples taken under various illumination settings and camera viewpoints. Hence, there is an inherent dependence of the performance of an FR system from the amount of variance in the gallery.

[2] https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md

Table 10: List of celebrities that are used from the LFW dataset.

Andre Agassi	Angelina Jolie	Arnold Schwarzenegger	Winona Ryder	Colin Powel
Hugo Chavez	Jennifer Lopez	Keanu Reeves	Kofi Annan	Serena Williams

We first obtain the FaceNet Inception ResNet v1 feature extractor pre-trained on the VGGFace2 database [227] which consists of 3.3M faces and 9000 classes. The feature extractor is trained on a large-scale dataset so as to optimize its weights, using the original work's suggested triplet loss and hard negative mining methods, with the aim to generate discriminative face representations, robust to viewpoint, illumination and scale variance. We use the pre-trained embedding layer in order to generate face representations for the subset of identities we have previously selected in the LFW database. This subset, which mostly contains instances of famous celebrities, will serve as a pool of random unknown identities. For each additional "known" identity that we want to recognize to produce alerts, we must add to the gallery another set of 20 images with that identity. The gallery should now be enhanced with additional "known" identities and some random persons (celebrities from the LFW dataset). Then, we generate face representations for all the identities in the full gallery. After that step, the images are no longer required to be stored, therefore we discard them from the system. Finally, we train a multi-class SVM classifier to recognize all the identities.

During inference time, a given instance is fed to FaceNet embedding layer first. Then, the resulting vector is passed through the SVM classifier and an identity class is predicted. In case the SVM predicts a "known" identity we produce an alert that a known identity was found. In the case of a test image with a random unknown identity that does not exist in the gallery, the SVM will forcefully predict one of the gallery classes nevertheless (known or unknown). However, the embedding layer will produce a representation similar to no known or unknown identity in the training set and the prediction will have to be made with a very low confidence due to the high dissimilarity of the embedding vector with the training samples of the gallery. Therefore, a confidence threshold is applied so as to properly interpret the SVM's classification result. If a known identity class is predicted and the confidence score surpasses the threshold, we accept the classification result. Otherwise, if the confidence score is low, or an unknown identity class is predicted we derive that the query instance belongs to an unknown person.

4.5 Crisis Event Detection

Crisis event detection is a quite intriguing computer vision problem that bother the scientific community for some decades now, due to its wide applicability in many real case scenarios that concern smart city security and citizen's safety. Its basic goal is to detect crisis events in a visual data in a spatio-temporal manner. In other words, this means that it tries to identify the appropriate techniques that will be needed in order to in order to localize smoke, fire and water particles inside image and video frames (i.e. spatial detection) as well as track them throughout time (i.e. temporal detection). Several techniques have been introduced to tackle these challenges, revolving both around deep learning and feature level approaches. Semantic segmentation, dynamic texture recognition and deep learning will also be designed and deployed in PREVISION project as well.

4.5.1 Related Work

Crisis event detection include the spatio-temporal recognition of fire and smoke events in visual content. The literature in this domain usually analyses the unpredictable nature of these events using both local features, such as LBP-Flow [73] and HoGP [42], as well as more generic ones, such as the spatio-temporal energy features.

As far as **spatial localization** is concerned, it is very usual in the literature to design and deploy deep Convolutional Neural Networks in order to initial perform a fast image classification and determine whether an image contains a crisis event, such as fire, flood or explosion. Semantic image segmentation can then follow in order to determine the image location of the specific event and if necessary, to be complemented by an object detection algorithm in order to define the number of people and vehicles affected in the surrounding area. Based on the result, spatial crisis event detection could provide a reliable and robust probability for defining the severity level of the crisis event.

- **Image classification** on the literature can be based on traditional Deep Convolutional Neural Networks (DCNN) [80] or prefer the exploitation of more recent deeper CNN models such as [117], where smaller stacked kernels were initially introduced, and improved later on by introducing more sophisticated and flexible architectures, such as Inception [161] and ResNet [4]. As far as security and surveillance domains are concerned, we can encounter flood classification in [117], [218] and fire classification in [65], [101].
- **Semantic image segmentation** literature has also the tendency to use deep CNNs by changing the goal of the classifier and classify each pixel in the image individually, leading to a classification mask for the whole image instead of a recognition class [86], [213]. As far as security and surveillance are concerned, there aren't so many techniques that work with the subject, as the features are too chaotic and very difficult to define a patternality amongst different environments and there aren't any annotated image samples (i.e. binary masks that will denote the location of the event) that could help on the training of these models. A worth-to-note technique which performs fire detection in social images with the use of color and texture attributes was presented [28] and [55].
- On the other end, **object detection** has numerous applications, in different domains, such as autonomous driving, smart video surveillance, facial detection, ambient assisted living and many others. Early works such as [144] included multi scale bounding box proposal generation techniques like Selective Search [69], as a feeding mechanism of candidate boxes to deep classifiers and then incorporate the outcome into single shot object detectors, using end-to-end deep architectures such [84], [86] and [70]. Those models achieved a better trade-off between accuracy and speed. A quite intriguing work was presented in [5] and [56] where, a quite novel and flexible scheme proposed to detect vehicles and pedestrians from traffic surveillance cameras using DCNN and Fisher encoding scheme, leveraging both the power of DCNNs as well as the one derived for high dense representation schemes. The same framework has also been deployed in UA-DETRAC vehicle detection dataset [80], achieving a really high detection rate.

On the other hand **spatio-temporal texture recognition** is amongst the most intriguing topics within computer vision society for some decades now. Dynamic texture typically refers to moving textures, such as the one monitored in fire, smoke and water, which undergo small, stochastic but unpredictable motions and are quite different to rigid motion, such as the one predicted by humans and vehicles.

The automatic recognition of such textures has recently attracted attention, as it can provide a significant contribution to many real-world outdoor applications, such as security applications focusing on the prevention of possible terrorist act and surveillance systems, which could be used for the avoidance of natural disasters. The main challenges for the analysis of dynamic textures and scenes consist of: (a) illumination changes, (b) complex and unpredictable motion patterns, (c) occlusions, (d) the presence of rigid and non-rigid objects in the same scene, (e) camera motion, and finally (f) significant intraclass differences among patterns of the same category. Computational efficiency is also a significant factor, as it must be kept within reasonable limits, so as to be used by real-world applications.

- Dynamic texture recognition methods can roughly be separated into two main categories according to their adopted underlying model. The first category refers to **Generative models** which involve the extraction of global features throughout video sequences and their modeling is based on some hidden parameters [52]. Recent works such as [192] use the spatiotemporal dynamics to train a Gauss-Markov recognition model, while [23] propose an expectation maximization (EM) algorithm to train the parameters of a statistical model. In [198] a Linear Dynamic Texture (LDT) scheme is proposed in order to represent a stochastic model of different appearance and motion dynamics. Lately, Linear Dynamical Systems (LDS) raised a lot of attention within this category, with the work of [118] being a representative example. In their work, a hierarchical EM algorithm is deployed in order to cluster and learn the statistical model of the motion dynamics. LDS has recently been extended into a stabilized higher order LDS (shLDS) in [42], who introduced Histograms of Grassmannian Points (HoGP). However, despite its high accuracy rates the method is computational costly, making it inappropriate for real-time applications.
- While generative models seem quite promising for representing dynamic textures, their application to classifying the wider set of motion patterns found in dynamic scenes has been shown to perform poorly [200]. The complex, stochastic character of dynamic textures makes their precise modeling very challenging, so a second category of dynamic texture representation, namely **Discriminative models** has been considered. This category is based on the extraction of local, spatio-temporal features to describe moving texture dynamics by estimating local variations and statistics of intensity and optical flow values. Early techniques involved the accumulation of local spatio-temporal features using appearance features like GIST [191], motion histograms, such as the Histograms of Oriented Optical Flow (HOOF) [26], swarm-intelligence [74], Spatio-Temporal Oriented Energy Features (STOEF) [41], and their successful and highly accurate Bag-of-Words (BoW) extension proposed in [209], named spatial energies. However, the coarse quantization of GIST and the rotation invariance of HOOF do not allow them to detect dynamic textures with accuracy, while on the other hand, the highly accurate STOEF, spatial energies and swarm dynamics suffer from computational efficiency making them inappropriate for real case implementations, such as surveillance and security scenarios.
- Accurate texture classification has been achieved in images using Local Binary Patterns (LBPs), whose promising results have led to a number of its extensions as a dynamic texture descriptor. Volume Local Binary Patterns (VLBP) [182] and LBP-TOP [202] are among the earlier methods, however they can easily reach a dimensionality of 2^{14} to 2^{26},

which is impractical in real-world applications involving large amounts of data that are to be processed in near real time. More recently in [111] a hybrid spatio-temporal extension of LBP was introduced, which stacks the descriptor in time to obtain temporal information. Even though, the method achieved very high accuracy rates when discriminating between water and non-water scenes, its highly tailored character to exclusively water class, makes it inappropriate for more general classification and localization scenarios.

4.5.1.1 *Datasets for Crisis Event Detection Methods*

In this section we compile all the visual data (images and videos) that have been identified as benchmark and are available on the web for research purposes only. The compilation of the data is based on our deep knowledge on the crisis scene detection and the deep scrutinization (filtering) that we deployed on the literature for the three crisis events scenarios that have been identified from the consortium partners, namely fire, smoke and flood and traffic scenes. The search and compilation of these publicly available datasets have been driven by the abovementioned requirements and are listed below category based:

Fire datasets

- The ***Mivia fire***[3] dataset contains several video samples (31) that depict fire cases that occur in both indoor (i.e. office scenario) and outdoor (i.e. forest fire scenario) environments. The dataset also contains non-fire situations in order to evaluate the discrimination power of the fire detector.
- The ***FIRESENSE***[4] *fire detection* dataset, which was compiled within the FIRESENSE project, includes both forest fires as well as human-induced ones, recorded by static cameras. The dataset was broadly used the last decade to evaluate fire detection algorithms. The demo fire clips[5] contain some further video samples, depicting fire situations recording with a static camera.
- The ***Rabot***[6] *2012* dataset is a rather simple dataset that contains, among others, video samples of fire scenarios in indoor environments.
- The **BowFire**[7] **dataset** is a fire dataset comprising of 226 fire images with various resolutions, from which: 119 images containing fire, and 107 images without fire. The fire images consist of emergency situations with different fire incidents, as buildings on fire, industrial fire, car accidents, and riots and are used to train the fire detection model. The remaining images consist of emergency situations with no visible fire and also images with fire-like regions, such as sunsets, and red or yellow objects.
- **UIA-CAIR**[8] Fire-Detection-Image-Dataset, which contains normal images and images with fire. It is highly unbalanced to reciprocate real world situations. It also consists of a variety of scenarios and different fire situations (intensity, luminosity, size, environment, etc.).

Smoke datasets

- The ***Mivia smoke***[9] dataset is composed by 149 videos, each lasting approximately 15 minutes and is widely used for smoke detection evaluation. It is a very challenging dataset,

[3] http://mivia.unisa.it/datasets/video-analysis-datasets/ fire-detection-dataset/
[4] http://signal.ee.bilkent.edu.tr/VisiFire/
[5] http://signal.ee.bilkent.edu.tr/ VisiFire/Demo /FireClips/
[6] http://multimedialab.elis.ugent.be/rabot2012/
[7] https://bitbucket.org/gbdi/bowfire-dataset/downloads/
[8] https://github.com/cair/Fire-Detection-Image-Dataset
[9] http://mivia.unisa.it/datasets/video-analysis-datasets/smoke-detection-dataset/

since it contains scenes and elements red houses in a wide valley, mountains at sunset, sun reflections in the camera, and clouds.

- The ***FIRESENSE***[10] *smoke detection* dataset, also compiled within the FIRESENSE project.
- The **Visor**[11] smoke detection dataset contains a wide number of smoke video samples and is used to evaluate the discrimination power of smoke detectors. Several motions also exist in the videos, so the separation between smoke and non-smoke regions is determined as a quite challenging task for state-of-the-art smoke detectors.
- The ***VisiFire***[12] ***smoke*** and *forest smoke*[13] datasets were recorded by the Bilkent University and are broadly used to evaluate smoke detection algorithms. The video samples are recorder by using static cameras in both forest and office environments.

Flood datasets

- *The* ***Video Water database***[14] consists of 260 high-quality videos that contain water depictions of predominantly 7 subcategories, namely canals, fountains, lakes, oceans, ponds, rivers, and streams, as well as non-water depicting samples that contain objects with similar spatial and temporal characteristics, such as clouds/steam, fire, flags, trees, and vegetation. The dataset is used for modeling water dynamics and can be used to evaluate flood detection algorithms.
- *The* ***Dyntex***[15] database is a diverse collection of high-quality dynamic texture videos that consists of more than 650 sequences, which can be used in order to model water, smoke, fire and other texture dynamics. It is broadly used for evaluating water, fire and smoke detection algorithms.
- The ***Moving Vistas***[16] dataset, broadly used for dynamic scene understanding, comprises 10 videos for each of the following 13 categories: Avalanche, Iceberg Collapse, Landslide, Volcano eruption, Chaotic traffic, Smooth traffic, Tornado, Forest fire, Waterfall, Boiling water, Fountain, Waves and Whirlpool, thus making it relevant for several PREVISION pertinent detection purposes, such as traffic, water, and fire recognition. Large variations in the background, illumination, scale and view render it a very challenging dataset that can be used for modeling and evaluating dynamic texture algorithms.
- **3F-emergency dataset** [55], is a collection of 12K social media images for fire and flood emergency scenario. The images contain scenarios of a 'flood', 'flooded street', 'fire', 'burning vehicle', 'fire accident', 'flood emergency', 'fire explosion', 'wildfire', etc. and other classes so as to identify the negative classification cluster comprising of classes such as 'busy street', 'forests', 'mountains', 'sunrise', 'beach', etc.

4.5.2 Crisis Event Detection Framework

4.5.2.1 *Spatial crisis event detection framework*

For the spatial localization of crisis events in video frames and images, we intend to deploy a multimodal approach that will be responsible for detecting crisis event particles, such as fire, smoke and flood in them. For these purposes we will design and deploy an **image classification** algorithm to determine whether an image contains a crisis event or not, then deploy appropriate **semantic segmentation** techniques to localize the image particles where a crisis event exist and finally and

[10] http://signal.ee.bilkent.edu.tr/VisiFire/
[11] http://www.openvisor.org/video_videosIn Category.asp?idcategory=8
[12] http://signal.ee.bilkent.edu.tr/VisiFire/Demo/Smoke Clips/
[13] http://signal.ee.bilkent.edu.tr/VisiFire/Demo/Forest Smoke/
[14] https://staff.fnwi.uva.nl/p.s.m.mettes/
[15] http://dyntex.univ-lr.fr/
[16] http://www.umiacs.umd.edu/users/nshroff/ DynamicScene.html

object detector to determine, whether there are people or vehicles in the surrounding area that could be in danger. The overall diagram is depicted of this framework could be found in Figure 28.

Image classification will be based on fine-tuning the pre-trained parameters of the ***VGG-16*** *[117]* on ***Places365*** dataset [187] in order to leverage the several distinctions that appear in this dataset between various visual clues that relate to generic scenery images, contrary to the ones that appear in crisis event scenes. Contrary to the initial VGG16 framework, in our case the final Fully Connected (FC) layer was removed and replaced with a new FC layer with a width of 3 nodes freezing the weights up to the previous layer and also deployed a softmax classifier in order to enable multi-class crisis event detection, such as "Fire", "Flood" and "Other". The outcome indicates the existence of fire and flood events in static visual content such as images and video frames in a holistic manner.

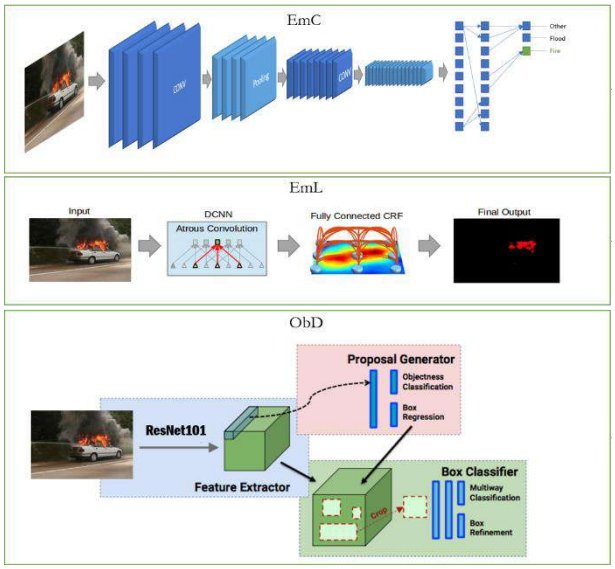


Figure 28: Crisis event detection in images

In the case of a positive outcome, semantic segmentation start analyzing the images in order to identify the regions where fire and flood pixels exist by providing the appropriate crisis event binary mask. Inspired from the recent success that semantic image segmentation achieved by [27], the DeepLab architecture of "atrous convolution", which uses convolution with up-sampled filters is adopted. Atrous convolution allows a wider reception field of the convolution filters, leading to richer context representations, while it also combines the result feature vectors of the final convolutional layer with a fully connected Conditional Random Field (CRF) which provides refined segmentation masks as it includes neighboring context on its calculations.

Object Detector also analyzes the images when image classification provides a positive outcome in order to identify persons and vehicles in the immediate surroundings by providing a set of bounding boxes. Groups of people or individuals are detected as persons, while vehicles may contain one of the following categories: cars, trucks, buses, bicycles and motorcycles. The basis of our object detection component is inspired from Faster R-CNN [148], pre-trained on COCO dataset [98], with some alterations so as to make it fit to our crisis event detection purposes. More specifically, based on [68], the ***ResNet101*** feature extractor was deployed for the extraction of deep features and then a Region of Interest (RoI) pooling scheme was used to classify candidate boxes. The model was trained in COCO dataset and only the relevant object classes were kept as valid predictions (e.g. vehicles, people).

4.5.2.2 *Spatio-temporal crisis event detection framework*

In order to effectively deal with the challenging nature of videos containing outdoors unconstrained environments, their representation should be firstly carefully examined and determined. The stochastic movements of the ensembles comprising dynamic textures in combination with their non-rigid nature, require the adoption of general descriptors, capable of managing highly unpredictable and ambiguous types of videos. To this end, the LBP-flow descriptor was adopted, which is then encoded by Fisher vectors resulting in an informative mid-level descriptor. The process is shown to be able to accurately classify dynamic scenes whose complex motion patterns are difficult to separate otherwise.

LBP-flow was initially introduced in [3] and was further adapted in PREVISION in order to accurately describe videos' underlying structure, as it has proven to effectively encode both appearance and motion induced variations, present in dynamic textures. LBP-flow builds upon the original LBP and extends it over time providing a powerful shallow spatio-temporal descriptor. In classic LBP, the LBP value of a particular pixel is computed by comparing its intensity value with that of its neighboring pixels. LBP-flow extends this definition to also include the values of the optical flow around the pixel, so as to embed motion information. The representation of motion as a temporal texture is introduced by calculating LBP over the optical flow values in the x and y directions, x-t and y-t respectively. This inclusion of motion information in the LBP-flow representation enriches the descriptor's spatio-temporal characteristics leading to a more robust and efficient shallow representation.

LBP-flow includes rich spatio-temporal information as a low-level local representation, but also allows for redundancies, such as intra-class pattern deviations and noise-induced artifacts. In order to constrain this noise and subsequently increase the discriminative ability of our descriptor, the **Fisher Vector** representation is adopted, transforming initial LBP-flow vectors of each video sample into a mid-level single vector representation, based on the detected most discriminating features (visual vocabulary) of a training video database. In this way, the size of the descriptor is significantly reduced, while at the same time recognition accuracy is increased. The computation of the most discriminating samples is performed by applying unsupervised clustering (Gaussian Mixture Model (GMM)) in the shallow representation hyperspace, as formed by the LBP-flow feature collection of the dynamic texture dataset.

Given the aforementioned powerful descriptor, a framework for dynamic texture recognition and localization is built. Fisher vectors are either used to train a binary/multi-class **Support Vector Machine** (SVM) classifier or a Neural Network (NN), in order to learn to discriminate between two or more classes. The framework including the NN can be characterized as a hybrid representation scheme, as it leverages both shallow and deep parameters to train a final classification model. Dynamic texture localization follows, to spatio-temporally localize the selected dynamic texture inside, and throughout, sequential video samples. The scheme exploits the resulting binary model of the recognition process and based on a superpixel clustering procedure leads to an accurate and computationally efficient localization framework. The overall diagram of this framework could be found in Figure 29

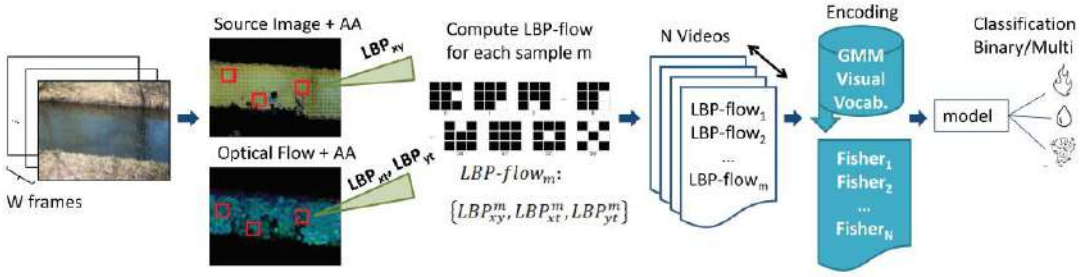


Figure 29: Block diagram of the spatio-temporal crisis event detection framework.

4.6 Conclusion and Future Steps

Regarding activity recognition, our plan includes the evaluation of the proposed framework using a dataset that should be closer to the LEAs that are participating in PREVISION project. Furthermore, the automatic localization of the activities during the time will be also investigated. Finally, updated versions of the demonstration tool will give advantages for the selection of the activities that need to be recognized.

Regarding person re-identification problem, a large-scale synthetic dataset that can be used for many different tasks has been created, e.g. attribute-based person re-identification. So far, the creation of the dataset has been completed but the process of annotating attributes is still ongoing. Moreover, we thoroughly evaluated different strategies to use temporal image sequences instead of single images to achieve a more robust pedestrian attribute recognition. Based on these results, we compared those approaches in terms of their suitability for attribute-based person re-identification. Next steps will be the completion of annotations, as well as an investigation of strategies to use synthetic data. The focus will be on reducing the amount of real-world data required and improving description-based person retrieval. In addition, in the next phase of the project, we will also address the topic of vehicle re-identification.

For the modules related to spatial crisis event detection, namely fire, smoke and flood detection in static visual samples such as images and video frames, we envisage to continue training our algorithms in deeper and more compressed CNN architectures, so as to better tailor our algorithms to the PREVISION use case needs. As far as spatio-temporal crisis event detection, namely fire, smoke and flood detection in dynamic image sequences and video samples, we plan to explore the option of evaluating the deeper architectures and modify them accordingly for video analysis. Key frame extraction would greatly augment the computational efficiency of the output, while the algorithms are expected to get a speed boost as well.

In this version, we have deployed SoA face detection and recognition algorithms into a unified pipeline to process input images and recognize known faces. We have tackled the task of face recognition as a classification problem. For our future steps, we will examine deep metric learning techniques so as to expand the applicability of our method and alleviate the need of retraining the SVM classifier once new entries are added to the gallery.

5. DarkNet, Web and Social Networks Data Analysis

5.1 Community Detection and Key Actor Identification

Due to the massive use of social networking platforms, the research community, as well as other communities, have put a lot of effort towards understanding the way the information is spread online and at a very large scale. A set of individuals, known as key actors, are often responsible for the vast cascading of information across the network. Especially, the identification and monitoring of key user accounts who may be considered potential threat to a society due to the spread of non-objective, misleading, or even destructive information (e.g., propaganda), could be of vital importance for Law Enforcement Agencies who will be then in position to prevent the further spreading of such "malicious" information. Thus, here, we present PREVISION key actor identification framework, which is able to identify key actors on multidimensional social networks, by considering several relationship types among users. The framework transforms the multidimensional network to a weighted single-layer network based on various mapping methods, performs then community detection to identify communities of users whose members share the same ideas and interests, to apply finally a set of centrality measures capable of detecting key actors within such communities.

The rest of this section is organised as follows. Section 5.1.1 reviews related work. Section 5.1.2 presents PREVISION community detection and key actor identification framework, while finally this section concludes with a summary (Section 5.1.3).

5.1.1 Related Work

This section reviews related work. Specifically, Section 5.1.1.1 focuses on state-of-the-art community detection methods, while Section 5.1.1.2 discusses popular key actor identification methods.

5.1.1.1 *Community Detection*

In general, within a network, community can be defined as *a group of entities more densely connected to each other compared to the rest of the network and which usually share common properties*. Community detection (or graph clustering) has been extensible applied in social media user networks [196], [204], [162], and [9]. A network can be seen as a graph, $G = (V, E)$, where the set of nodes V represents the network actors and the set of edges E represents the links between the actors, i.e., G can be represented by the set of triplets (v_i, v_j, w_{ij}) where $v_i, v_j \in V$ are the edge nodes and w_{ijk} is the weight of the relationship (e.g., reflecting its strength) between these two nodes. In this sense, community detection is applied on a graph.

In the context of social media, community detection is often applied to friendship networks which are generated by the declared users' affiliations, e.g., based on follower/followee Twitter relations [115]. Social interactions (e.g., mention to a user, tag or response to social content) can be also exploited to construct user communities since interactions among their members may indicate both awareness of each other as well as interest in common topics. The exploitation of content information could be a way to detect hidden relationships between users [217], where the similarity of the content posted by social network users is estimated. Thus, due to the multiple interaction types that exist in online social networks, heterogeneous information networks are constructed [139], [140].

A popular community detection algorithm is SCAN [197], which builds on the density-based clustering algorithm DBSCAN [190]. While DBSCAN has been widely used for clustering spatial points based on their density distribution, SCAN operates on graphs based on a structural similarity measure. An alternative constitutes the Markov Cluster (MCL) algorithm, i.e., a fast and scalable unsupervised cluster algorithm for graphs based on simulation of (stochastic) flow in graphs [45]. FastGreedy [32], WalkTrap [194], and Louvain [16] methods are popular community detection methods due to their applicability to very large social media graphs. Finally, the latest advancements in neural networks and neural representation learning have also been applied in the context of graph analytics in the form of *network embeddings* where the objective is to learn latent representation of nodes on a network, thus allowing then the detection of coherent communities [119].

5.1.1.2 *Key Actor Identification*

Due to the importance of identifying key actors within online communities, key actor identification has attracted the interest of the research community. For instance, authors in [134] have studied a wide range of social media (i.e., Twitter, Facebook, and Livejournal) alongside scientific publishing in the American Physical Society to detect the most influential spreaders of information. Moreover, focusing on Twitter, Canada's political communities [46] and the most influential candidates of the European Elections 2014 [7] have been studied. Finally, other well-known social networks, such as Delicious (a social bookmarking web site), Epinions (a product review web site), and Slashdot (a technological news web site) have also been analysed to find out the more influential spreaders [96].

Users' direct relationship (such as follower and friends relationships) are often exploited to identify key actors within social networks. For instance, to detect the most influencing users on Twitter communities, authors in [46] build initially a friendship network to employ then various commonly used measures, such as in-degree, eigenvector centrality, and clustering coefficient. Apart from friendship relationships, indirect association among users are also considered, such as likes or retweets of a post, mentions of users, etc., to construct a network of users and thus find hidden associations between them. Towards this direction, authors in [205] build a network considering three relationship types on Twitter, i.e., following, retweets, and mentions, while then three algorithms are proposed (i.e., InfRank, LeadRank, DiscussRank) to identify influencers, leaders, and discussers. Similarly, authors in [72] consider mentions, replies, and retweets to detect then the top terrorism-related key actors on Twitter based on various centrality (e.g., degree, betweenness, closeness, and eigenvector centrality) measures, while in the same context, entropy-based centrality measures have been proposed to identify key actors in terrorism-related Twitter accounts (identified through the use of Arabic keywords related to terrorists' propaganda) [54]. Finally, retweet, reply, reintroduce (when a person reintroduces tweets instead of retweeting), and read (probability of reading tweets) relationship types have been considered to measure the influence of users on Twitter based on random walks [184].

5.1.2 Multidimensional Key Actor Identification Framework

Here, we present PREVISION framework for identifying key actors in social networks, with a particular focus on Twitter. This framework involves the following steps: (i) weighted multidimensional social network building, (ii) weighted single-layer social network building, (iii) community detection, and (iv) key actor identification. This initial version of PREVISION framework builds upon the multidimensional

key actor identification framework that has been developed in the EU-funded H2020 TENSOR project (Grant Agreement ID: 700024).

5.1.2.1 *Weighted Multidimensional Social Network*

To construct a weighted multidimensional social network, the interactions among users are considered. Specifically, to quantify the interactions between users, three relationship types are examined, i.e., mentions, replies, and retweets. In the resulting network, each user is represented by a node, while an edge is created between two users if one or more interactions are detected between them. In the edge created between two users a weight can be attached, which reflects the strength of the interaction.

5.1.2.2 *Weighted Single-layer Social Network*

In order to transform the aforementioned weighted multidimensional social network into a weighted single-layer social network (needed due to the centrality measures employed in the PREVISION framework) a set of mapping methods is used. Specifically, five mapping functions are employed:

- Binary Network: a single-layer network is produced, where two users are linked to each other if they have interacted with each other at least once (regardless of the relationship type).
- Multi-binary Network: here, the total weight of each edge between two users corresponds to the total number of relationship types that exist between such two users.
- Multi-weighted network: the main difference between Multi-binary and Multi-weighted mapping functions is that here instead of simply checking whether exists an interaction between two users for the three relationship types, we consider the number of times that such interactions have been performed.

- Multi-binary Network with Relationship-based Importance: this mapping function considers the importance of each relation type based on two approaches.

  Approach 1. The importance of each relationship type is a fraction of the total weight between all network edges of this relationship type when compared to the total weights between all network edges for all the relationship types.

  Approach 2. The importance of each relationship type is defined as the fraction of the total weights between all network edges for all the relationship types when compared to the total weights between all network edges of the considered relationship type; is the inverse fraction of the previous approach (Approach 1).

- Multi-weighted Network with Relationship-based Importance: this mapping method transforms the multidimensional social network to a single-layer social network similarly to the Multi-binary network with relationship-based importance mapping method with the difference being the consideration of the Multi-weighted network instead of the Multi-binary one.

5.1.2.3 *Community Detection*

The initial version of PREVISION key actor identification framework supports two very popular community detection methods: (i) FastGreedy and (ii) Louvain.

FastGreedy method. The FastGreedy method [32] is a hierarchical approach for detecting communities, where the objective is to optimise a quality function, known as modularity (how densely connected the nodes within a cluster are). Initially, every node in a graph belongs to a separate community, while then communities are iteratively merged so that each merge to produce the largest possible increase in the modularity. The algorithm terminates when it is not possible to increase further the modularity. The advantage of this method is that it is quite fast and there is no need for parameters tuning.

Louvain method. The Louvain method [16] is suitable for identifying groups on large networks as, similar to FastGreedy method, attempts to optimise the modularity measure of a network by moving nodes from one cluster to another. Specifically, the optimisation is performed on two steps. First, the method searches for small communities by optimising modularity locally on all nodes, while then it groups nodes belonging to the same community and builds a new network where its nodes represent these communities. These steps are repeated iteratively until maximum modularity is achieved and a hierarchy of communities is generated.

5.1.2.4 *Key Actor Identification*

Various centrality measures have been considered in literature to identify key actors in online communities. In the end, the top key actors are ranked in descending order based on their respective centrality score. In PREVISION key actor identification framework five centrality measures are employed: (i) Degree Centrality, which counts the number of neighbors a user has [51], (ii) Betweenness Centrality, which quantifies the number of times a user acts as a bridge along the shortest path between two other users [51], (iii) Eigenvector Centrality, which measures the influence of a user in a network [17], (iv) PageRank Centrality, which measures the importance of a user in a network (a user is important if they linked with other important users or if they are highly linked) [21], and (v) Closeness Centrality, which indicates how close a user is to all other users in a network [51].

5.1.3 Summary

The rise of social networks has enabled great advances in communication platforms, and specifically in the way that the information is spread among people. Similar to the offline world, the extent to which the information will be diffused in a network, as well as its propagation velocity, importantly depends on the position that the transmitter has within it. The detection of such key transmitters (i.e., key actors) could be of utmost importance, especially in cases where the objective is the early stopping of the diffusion of misleading, abusive, or even destructive information. In this context, we presented PREVISION community detection and key actor identification framework, which exploits a set of popular relationship types that take place among Twitter users, i.e., mentions, replies, and retweets, as well as the strength of such relationships to detect communities of users, along with the key actors of such communities.

5.2 Actor Identity Resolution

In its somewhat more than 20 years of existence, social media constitutes an integral part of the life of more than 2.6B people around the globe. Originally envisaged as a means to stay connected with friends, get informed, or be entertained, it has become a very powerful instrument for public opinion formation and dissemination of all kinds of not always harmless content. Actor identity resolution (or

otherwise, user identity linkage) within a single social network can offer improved understanding of the networks formation, while also can set the ground for mitigating abusive and/or illegal activities on large scale. Users in an effort to outspread their thoughts, ideas, and perspectives often hold several accounts in a social network. Especially, when non-legitimate, or even illegal, activities take place, users tend to create multiple accounts to bypass social media combating measures, i.e., retain their online identity even though an (set of) account(s) gets offline by force. In alignment to PREVISION needs, user identity linkage is studied here as a means to reveal accounts that are likely to belong to the same natural person in an effort to prevent the spread of criminal or terrorism-related behaviours on a large scale.

The rest of this section is organised as follows. Section 5.2.1 reviews related work. Section 5.2.2 focuses on the PREVISION actor identity resolution framework and presents the employed dataset, the extracted features, as well as the techniques for modelling the data and predicting possible user linkage. Then, Section 5.2.3 describes the process for building the ground truth, the feature selection process, the experimental methodology, and the classification results. This section concludes with a summary (Section 5.2.4).

5.2.1 Related Work

Numerous studies have examined user identity linkage *across* online social networks [71], [207], and [168]. Malhotra et al. [71] proposed to disambiguate profiles of the same user based on their digital footprint in both Twitter and LinkedIn. Twitter has also been jointly considered in many works as one of the studied platforms in relation to other social networks, e.g., Yelp [136], Flickr [136], Foursquare [168], Instagram [168], and Facebook [207]. For instance, authors in [168] proposed a method that examines whether two accounts belong to the same mobile user, through the exploitation of location information, when acting on Twitter and Instagram. Identity linkage on a single social network has also been explored. For instance, an Irish forum was studied [116] to first unmask authors' identities and then detect matching aliases. The so-called "sockpuppetry" (i.e., blocked users initiating new accounts) has been considerably studied on Wikipedia [127], [215]. Finally, user identity linkage has been explored on popular online news sites, such as *The Guardian* and the *SPIEGEL ONLINE*, to assist their providers detect attacks on public opinion [208].

To build a model for identifying actors' identity features of different types are considered, such as profile (e.g., username and biography [120]), content (e.g., temporal and spatial information [168], [24]), or network based (e.g., based on a user's friendship network [24], [106]). Stylometric features (e.g., part-of-speech tags, n-grams, word length distribution, etc.) are also widely employed [116], [127], and [208]. Based on such features, then supervised, unsupervised, and semi-supervised methods are employed to detect user identity. For instance, authors in [76] proceed with a probabilistic classification, based on Naive Bayes, to map identities of individuals across social media sites. In addition to Naïve Bayes, decision trees, SVM, and kNN algorithms have also been tested [71]. On the other hand, an alignment algorithm has been used, where an affinity score is computed based on timestamped location-based properties to find the most likely matching identities using a weighing scheme [168]. Regarding semi-supervised models, a multi-objective framework has been built for modelling heterogeneous behaviours and structural consistency maximisation [207].

The first version of PREVISION actor identity resolution framework aims to detect account that are likely to belong to the same natural person within a single social network (i.e., on Twitter). Various

activity, content, and network features are considered, while different traditional machine learning techniques are tested and evaluated.

5.2.2 Actor Identity Resolution Framework

PREVISION framework for the detection of the possible linkage of user accounts consists of three main modules: (i) data collection, (ii) feature extraction, and (iii) classification. The initial version of PREVISION framework acts as a baseline compared to the framework developed in the EU-funded H2020 CONNEXIONs project (Grant Agreement ID: 786731). Compared to CONNEXIONs, here, the emphasis is more on the feature selection process (Section 5.2.3.2), aiming at an in-depth understanding of those attributes that help significantly in identifying accounts likely to belong to the same natural person. In later iterations, special emphasis will also be given to users' writing style by extracting and analysing a wide range of linguistic features (as described in Section A.1).

5.2.2.1 *Data Collection*

The first step is to collect the necessary content from Twitter. For this study we use an abusive-related dataset obtained from Twitter, since it is likely to involve users with multiple accounts[17]. Specifically, we use a dataset provided by [134], which was created for studying abusive activities on Twitter; consists of 600k tweets in English and 312k users. It should be noted that personal data (i.e., usernames and ids) were pseudonymised upon collection using the MD5 hashing technique, i.e., a cryptographic hash function which randomly generates a hash sequence that is 128 bits in length.

5.2.2.2 *Feature Extraction*

Activity, content, and network features have been examined to model each individual user account. The features from each category are summarised in Table 11. In next iterations additional features will be considered selected from Section A.1.

Table 11. Considered Features

Category	Description	Features
Activity	They consider a user's posting behaviour	avg. # hashtags, avg. # mentions, posts' inter-arrival time
Content	The focus, here, is on users' posted content (tweets)	avg. characters per sentence, avg. characters per word, standard deviation (STD) characters length, STD words length, difference in length between the longest and shortest words, avg. # digits, avg. # punctuation marks, posts' similarity, uppercase ratio, part-of-speech tags (i.e., noun, verbs, adjectives, adverbs)
Network	They measure the users' connectivity in the network	clustering coefficient, eigenvector, pagerank, authority, hub, # triangles

Focusing on the content features, and specifically on the posts' similarity, the Levenshtein distance [164] is used, namely a measure that estimates the differences between two posts, by counting the minimum number of single-character edits needed to convert one string into another. As for the part-of-speech (POS) tags, i.e., the extraction of the average number of nouns, verbs, adjectives, and

[17] https://www.cnet.com/news/facebook-pulls-down-fake-accounts-from-the-uk-and-romania/

adverbs out of the available textual resources, we build upon the POS tagger provided by the Tweet NLP library[18].

To study the connectivity of users within a network, at first we construct a network based on the number of mentions, replies, and retweets between each pair of users. Users in such a network can have a varying degree of connectivity with different parts of the network, influence in their neighborhood, etc. To overview users' position within their network, six measures are estimated, i.e., Clustering Coefficient, Eigenvector and PageRank centrality measures, hub and authority scores, and the number of triangles that a node is a member of. To extract these features, Gephi[19] is used, i.e., a network analysis and visualisation software.

Joint Representation. The aforementioned features represent each individual user account; since though the objective is to detect whether two accounts are likely to belong to the same natural person, we need to jointly represent each user pair in order to determine their possible association. Towards this direction, we jointly represent the behaviour of each pair of users u_i and u_j, $\forall i, j$, where $i \neq j$, by the absolute difference between u_i and u_j for every feature under consideration. The absolute difference is employed here, since it typically refers to the distance between two numbers, which in our case can be considered to reflect the differentiation in users' behaviour.

5.2.2.3 *Classification*

The final step involves the classification based on the extracted joint representations. Different machine learning techniques are tested, such as probabilistic (e.g., Naive Bayes, BayesNet), tree-based (e.g., J48, LADTree, LMT), and Random Forest as an ensemble classifier. To build the Random Forest classifier, we tune the number of trees to be generated as 100, and the maximum depth unlimited.

5.2.3 Experiments and Results

This section presents our evaluation experiments on data collected from Twitter using the extracted features, the corresponding joint representations, and the ground truth. As already stated, we consider various machine learning algorithms, either probabilistic, tree-based, or ensemble classifiers; only the best results for each family of classifiers are presented. For evaluation purposes, we examine standard machine learning performance metrics: precision (prec), recall (rec), and weighted area under the ROC curve (AUC). For all experiments, we use the WEKA data mining toolkit and repeated (5 times) 10-fold cross validation.

5.2.3.1 *Ground truth*

To perform classification based on the extracted features and the corresponding joint representations, a ground truth dataset (depicts whether two user accounts belong to the same natural person within a dataset) is necessary. Due to the absence of an already annotated dataset, here, we create a ground truth as follows. From the abusive dataset described in Section 0, initially we randomly select 200 user accounts with more than 10 posts each to ensure that sufficient evidence will be available. Specifically, a stratified random sampling approach is employed, which initially involves the division of the entire population into smaller sub-groups based on the number of the posted tweets. We vary this number between 10 to 60 posts with step 5. The final sub-group consists of all users who have posted more

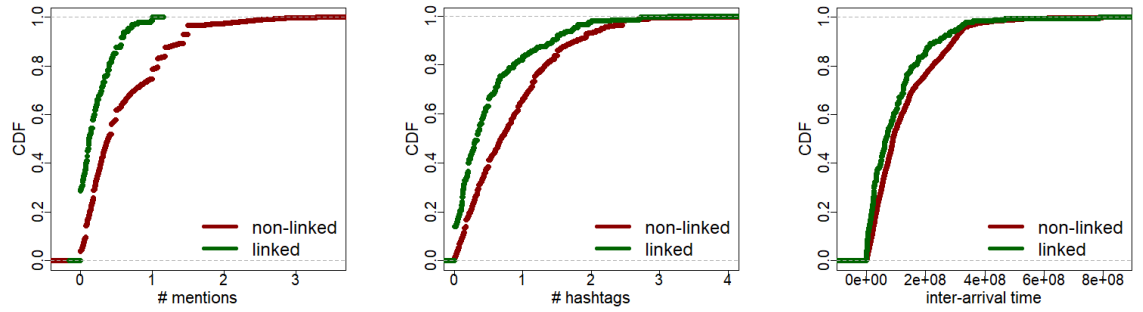
[18] http://www.cs.cmu.edu/~ark/TweetNLP/
[19] https://gephi.org/

than 60 tweets. Then, a random sample from each sub-group is taken in a number proportional to the sub-group's size when compared to the entire population.

To create the ground truth, similar to already existing works [116] [208], first we split each of the 200 selected users into two separate users (i.e., user u_i was split to u_{ia} and u_{ib}), resulting in this way to a set of known linked accounts. To split the tweets of the original accounts (e.g., u_i) into linked users (e.g., u_{ia} and u_{ib}) we proceed with a random assignment of an equals number of posts to each. This way, in the end we result to two sets of users, i.e., $A = \{u_{1a}, u_{2a}, \dots, u_{200a}\}$ and $B = \{u_{1b}, u_{2b}, \dots, u_{200b}\}$. Comparing each user from set A, one at a time, with all users in the set B, we result to overall 39,800 non-linked accounts, respectively. In the end, we maintain a proportion of 10% of linked and 90% of non-linked accounts; we opt for this selection given that previous works have indicated that almost 9% of users tend to exhibit bad behaviour within a dataset [7]. Overall, the final ground truth dataset consists of 200 and 1,800 linked and non-linked accounts, respectively.

5.2.3.2 *Features Selection*

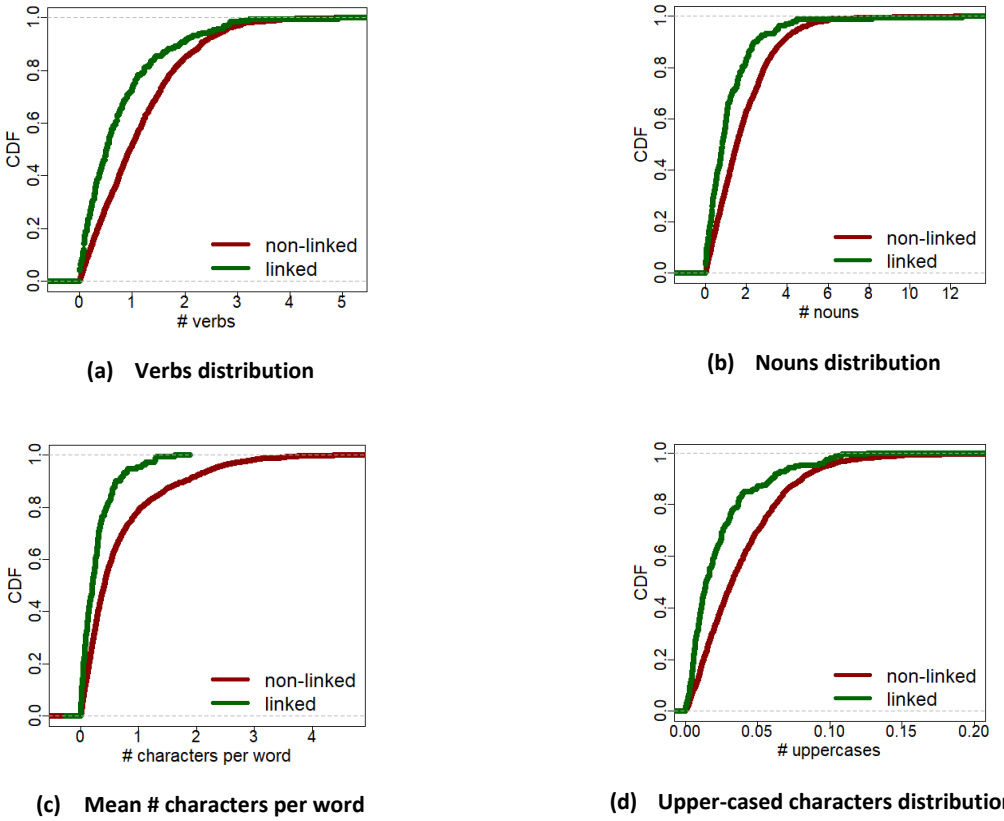
Section 5.2.2.2 described various features that could be considered for exploring whether two accounts is likely to belong to the same natural person. As expected, some features could be more distinguishable and could assist more in the classification. Here, we examine the significance of differences between the distributions of the linked and non-linked user accounts. To proceed with such an analysis we use the two-sample Kolmogorov-Smirnov test, i.e., a non-parametric statistical test, which enables assessing whether two samples come from the same distribution based on their empirical distribution function (ECDF). We consider as statistically significant all cases with $p < 0.01$.



(a) Mentions distribution **(b) Hashtags distribution** **(c) Inter-arrival time distribution**

Figure 30. ECDF plots for (a) Mentions, (b) Hashtags, and (c) Posts' inter-arrival time

Activity Features. Figure 30a-2b plot the ECDF for the number of mentions and hashtags for the linked and non-linked users ($p < 0.01$ with D=0.32961 and D=0.26301, respectively). We observe that the non-linked users tend to have a higher difference in relation to the number of mentions and hashtags compared to the linked user accounts. Concerning the inter-arrival time between the posted tweets (Figure 30c), we observe that the linked accounts tend to have less waiting time in their posting activity compared to the non-linked accounts, with the difference in their distributions being statistically significant (D=0.15849).



(a) Verbs distribution

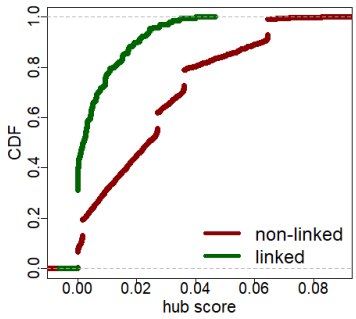
(b) Nouns distribution

(c) Mean # characters per word

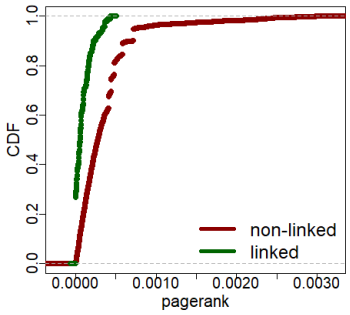
(d) Upper-cased characters distribution

Figure 31. ECDF plots for (a) Verbs, (b) Nouns, (c) Mean # characters per word, and (d) Upper-cased characters

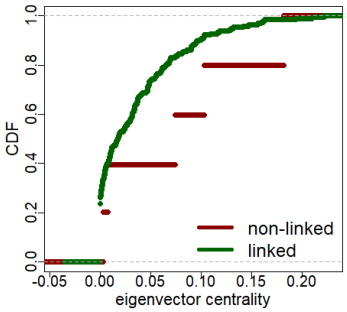
Content Features. To identify the linkage of two or more accounts we consider a set of various content attributes extracted from the available textual material. Indicatively, Figure 31a-3d depict the CDFs for the frequency of verbs, nouns, mean number of characters per word, and uppercased characters features. Specifically, from Figure 31a-b we observe that the linked accounts tend to use a lower number of nouns and verbs in their posts in relation to the non-linked ones. A similar pattern is observed in the case of mean number of used characters per word and uppercased letters (Figure 31c-d). Overall, comparing the distributions among the linked and non-linked accounts, we observe that the differences are statistically significant with D=0.25181, D=0.29595, D=0.30405, and D=0.29209, respectively. Considering the rest content features the difference in their distributions is also statistically significant. Specifically, for the average number of used adjectives and adverbs, D=0.26506 and D=0.34573, respectively, while, for the average characters per sentence, STD characters length, STD words length, difference in length between the longest and shortest words, average number of digits, average number of punctuation marks, as well as posts' similarity D equals as follows: 0.34149, 0.26708, 0.37006, 0.29999, 0.27521, 0.30742, 0.1832, respectively.



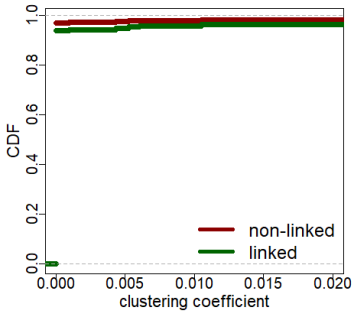
(e) Hubs distribution



(f) PageRank distribution



(g) Eigenvector distribution



(h) Clustering coefficient distribution

Figure 32. ECDF plots for (a) Hubs, (b) Pagerank, (c) Eigenvector, and (d) Clustering Coefficient

Network Features. Figure 32 depicts the CDF plots for the hub score (similar distribution is observed for the authority score), the PageRank and Eigenvector centralities, as well as the clustering coefficient. For the hub and authority scores the difference in distributions is statistically significant with mean (STD) values for the authority score to be equal to 0.00587 (0.00837) and 0.02383 (0.01965) for the linked and non-linked accounts, respectively, and for the hub score to be equal to 0.00612 (0.00873) and 0.02487 (0.02051) for the linked and non-linked accounts, respectively. Specifically, from Figure 32a we observe that linked accounts have lower value in their hub score (similar to the authority score), which indicates that they are not so popular in their networks**.** Concerning the PageRank and Eigenvector centrality measures the difference is statistically significant (D= 0.49974, D= 0.43939, respectively), which is not the case for the clustering coefficient and the number of triangles where we cannot reject the null hypothesis that the distributions are different.

5.2.3.3 *Experimental Methodology*

The analysis presented in Section 5.2.3.2 indicated that most of the features presented in Table 11 are useful (statistically significant) in discriminating between the two classes (i.e., linked and non-linked user accounts). However, some are not useful and are excluded from the modelling analysis to avoid adding noise. Specifically, the clustering coefficient and the number of triangles features are excluded from the following analysis.

5.2.3.4 *Classification Results*

Table 12 overviews the results obtained with BayesNet (BN), J48, and Random Forest (RF).

Table 12. Classification Results

	Activity Features			Content Features			Network Features			All		
	Prec	Rec	AUC	Prec	Rec	AUC	Prec	Rec	AUC	Prec	Rec	AUC
BN	**88.94**	**91.28**	74.20	90.14	86.36	85.10	97.58	97.60	96.78	95.96	95.60	97.80
J48	88.70	91.14	59.34	89.46	91.34	70.94	**99.08**	**99.10**	94.92	**99.24**	**99.24**	95.64
RF	88.56	91.02	**74.68**	**91.80**	**92.38**	**86.02**	97.80	97.80	**98.48**	96.28	96.14	**98.56**

From Table 12 we observe that the J48 based model with all types of features considered succeeds in detecting **99.24%** (recall) of both the linked and non-linked accounts, with the Random Forest following behind with **96.14%**. Overall, we succeed the best performance in terms of AUC with the Random Forest classifier (**98.56%**). Focusing on each feature category (i.e., activity, content, and network features) we observe that we result to the best performance (**98.48%** AUC) when only the network features are employed.

Focusing more on the applied features, the top five based on the information gain are the Eigenvector centrality, the hub and authority scores, the PageRank centrality, and the number of mentions (preserving the order of contribution). Thus, the most contributing features are the network-based; this is also depicted in the results obtained during the classification (i.e., higher AUC values).

Overall, our models perform well, particularly if we take into consideration the overall AUC of the ROC curves, which are typically used to evaluate the performance of machine learning algorithms by testing the system on different points and getting pairs of true positive against false positive rates indicating the sensitivity of the model. The high ROC area for the overall classification indicates that the corresponding models can quite successfully discriminate between linked and non-linked accounts.

5.2.4 Summary

Detecting accounts of the same user poses several difficulties, since often users alternate their behavioural patterns in an effort to stay under the radar of social media platforms. In this section, we presented PREVISION actor identity resolution framework, which is able to detect whether two accounts are likely to belong to the same natural person. Different types of features were tested, i.e., activity, content, and network based, to detect linked accounts on Twitter, while also traditional machine learning algorithms were evaluated. The results showed that the followed method is able to effectively detect linked accounts created in an effort to maintain and spread over time non-legitimate, or even illegal activities.

6. Deep Linguistic Analysis

6.1 Information Extraction with a Two-step Approach

The PREVISION Text Mining service realizes a component, which performs deep NLP analysis on texts which are written in English natural language. The linguistic analysis includes e.g. part-of-speech tagging, semantic role labelling (shallow parsing), named entity recognition, and in the end tries to capture the meaning of a sentence in intermediate linguistic model. This analysis step is the main and most expensive part in terms of runtime of this service. This component is mainly based on an already existing service called PIKES partly developed in the context of the former EU project NewsReader (FP7 2011.4.4). PIKES focus lies on the extraction of knowledge from textual resources (see http://pikes.fbk.eu/ for further information). The result of the processing is an annotated document from which the relevant information can be extracted in form of a conceptual graph of pre-processed linguistic information. Pikes performs the aforementioned tasks such as named entity recognition, semantic role labelling and word sense disambiguation by using techniques provided by already existing frameworks e.g. CoreNLP from Stanford University. Following the analysis step, which extracts information and instantiates an intermediate linguistic model, is the mapping step. This step maps the linguistic aspects to the domain model of PREVISION.

6.1.1 Parsing English Text into Intermediate Linguistic Model

The Semantic Role Labelling (SRL) is a technique to assign labels to words or sub-phrases in a sentence that indicate their semantic role, such as that of an agent, goal, location or time. A key factor for understanding the sentence and instantiating an appropriate event in the ontology is the identification of the (semantic) roles of a verb. SRL is based on the result of a dependency parser, which generates the syntactic structure called parser tree, where the top level (i.e. root of the parser tree) is a verb. Semantic roles for verbs are listed in an online accessible dictionary called Proposition Bank (see https://propbank.github.io), which contains about 10000 entries.

An example of a graph generated by Pikes for the sentence:

“The bomb explodes in Maghreb marketplace on 01/25/10.” is given in Figure 33, which shows the dependency information above and the SRL structure below the sentence.

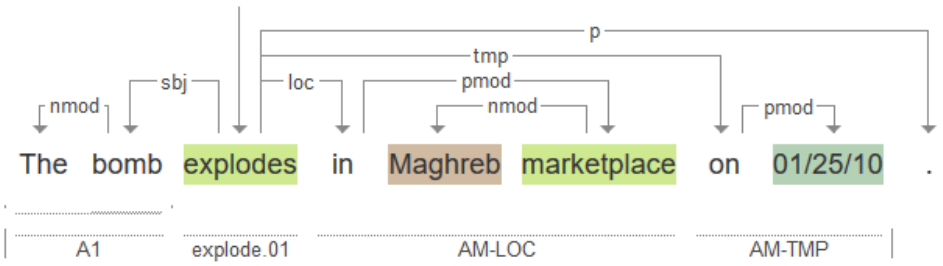


Figure 33: Example sentence and SRL structure.

The colors in the sentence, highlighting certain words, are references to an RDF tree, which is not further considered in the context of PREVISION.

Table 13 shows the semantic roles with their descriptions and the values assigned from the example shown in Figure 33.

Table 13. Semantic roles of explode.01

Argument	Role	Description	Value
A0	Agent	bomber, agent/cause	--
A1	Patient, Theme	bomb, thing exploding	The bomb
A2	Predicative Complement	attribute, end state	--
AM-LOC	Location	location	in Maghreb marketplace
AM-TMP	Time	time	on 01/25/10

6.1.2 Mapping Intermediate Model to PREVISION Domain Model

The information from the natural language analysis is collected in a structure of annotations, which models an intermediate linguistic language model. In a second step a mapping must be realized from this linguistic model to the target domain model, which in case of PREVISION is the PREVISION ontology model e.g. in the LEA domain. This model is based on the Intelligence Pentagram (see Figure 34) coming from the military domain, but is also suitable for the investigation tasks of an analyst in the LEA domain. The Intelligence Pentagram introduces five top-level knowledge categories, depicted as the corners of a pentagram (*Event*, *Bios,* etc.), and additionally a concept for time, depicted in between the corners. All corners are fully connected.

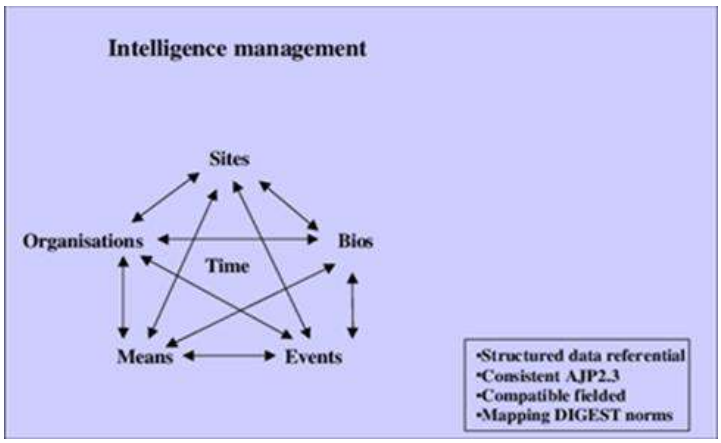


Figure 34: Intelligence Pentagram [13]

The linguistic parser generates an XML document in the NAF (NewsReader Annotation Format), where a part of it is describing the SRL information (besides dependency tree, named entities) extracted from the sentence. The main task now is to map the information contained in the NAF (seen as intermediate language) to the Intelligence Pentagram structure coded in the PREVISION ontology. Table 14 shows the mapping of the main semantic roles to the corresponding pentagram concepts. The mapping of a verb, which is looked up in the Proposition Bank, to an Event is a more involving process, which is described in the following chapter. The instances and their connections are finally written to the knowledge store, which is realized in PREVISION by an Apache Fuseki Triple store.

Table 14. Mapping of pentagram to SRL concept.

Pentagram	Bios	Sites	Organisations	Means	Events	Time
SRL	NER	AM-LOC	NER	AM-MNR	Proposition Bank	AM-TMP

6.1.2.1 *Event Detection*

The task of Event detection is a little bit more involving than the direct mappings of the other pentagram concepts. The main problem is to identify, if a certain verb (e.g. drive … away) has a correspondence to an *EventCategory* (like Movement) of the PREVISION ontology model. Moreover, an Event should be something like an activity or process. To decide, if a verb fall into this category, we use additional ontologies, which model processes or activities. These models are SUMO (Suggested Upper Merged Ontology) and ESO (Event and Situation Ontology). In a first step we access all synonyms of the verb with the help of Wordnet database. This list is than looked up in ESO and SUMO and if matches occur, we know, that the verb is a candidate for the mapping to an Event Category modelled in the PREVISION ontology.

"… GTK Boxer drove away."

1. synset: drive away -> leave
2. progressive form: leaving
3. searching in SUMO taxonomy:
   Leaving
   ↳Translocation
   ↳Motion
   ↳Process ☑

6.1.2.2 *State Description*

As described in the last sub chapter the text mining system is Event centered as e.g. an investigation tries to reconstruct the sequence of events e.g. along the time scale. But there is other descriptive information which is not *Event* based, this is the case for all state descriptions or possessive relations. For PREVISION the software has to be extended to capture such descriptive information, which will be introduced by the auxiliary verbs mainly *be* and *have*.

6.1.2.3 *A View to the Knowledge Base*

The visualization of and navigation in the knowledge store is at the time of writing realized by an IOSB product called the instance editor. This tool is IOSB background and will be replaced by a new solution, which can be integrated in the Web framework of PREVISION. For the example sentence (Figure 33) the IOSB tool allows the visualization and editing of the generated pentagram structure (*Event* with *EventCategory*, *Place*, *Time* and *Means*). Additionally, an instance of the concept *Resource* is generated, which references the original text document to support the chain of custody (Figure 35).

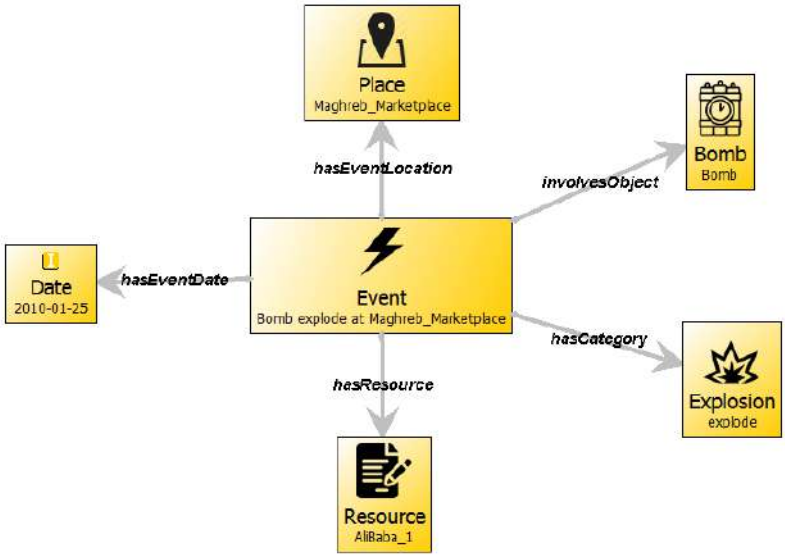


Figure 35. Semantic network of the example shown by IOSB tool (IPR background).

6.2 Extended Coreference Resolution

6.2.1 State-of-the-art of Reference Resolution Algorithms

6.2.1.1 *Rule-based Coreference Resolution*

Hobb's naïve algorithm [66] was one of the first algorithms developed for anaphora resolution. This algorithm is rule-based, to search for an antecedent it parses the syntactic tree of a sentence left to right by traversing breadth-first. Another well-known algorithm was the Lappin and Leass algorithm [14] for pronominal anaphora resolution. This algorithm was based on the salience assignment principle. BFP algorithm [20] was developed in order to exploit discourse properties for pronoun resolution. This algorithm motivated the centering theory [59] which used discourse structure to explain such phenomena as anaphora and coreference.

Most of the rule-based algorithms were knowledge-rich, however, there were some, e.g. [8], [63], [61], [89], [180], that aimed to reduce dependency of rules on external knowledge – the so called "knowledge-poor algorithms". CogNIAC [8] was a high precision coreference resolver is an example of such algorithms. The core rules defining CogNIAC were picking a unique or single existent antecedent in current or prior discourse, the nearest antecedent for a reflexive anaphor, picking exact prior or current string match for possessive pronoun, etc.

The COCKTAIL system [64] was one of the systems which took a knowledge-based approach to mine coreference rules. It used WordNet for evidence of semantic consistency and was based on principles of structural coherence and cohesion. Rule-based algorithm by [97] also took a knowledge-based approach for pronominal anaphora resolution. It used WordNet ontology and heuristic rules to develop an engine for both intra-sentential and inter-sentential antecedent resolution. This model also constructed a finite state machine with the aim of identifying noun phrases and checked for occurrences of anaphoric references and pleonastic *it*.

A widely used rule-based baseline of coreference resolution was a deterministic, called "H and K model" [61]. It proposed a strong baseline by modularizing syntactic, semantic and discourse constraints and outperformed all the unsupervised and most of the supervised algorithms proposed till then. This model motivated the use of multiple hierarchical sieves for coreference resolution.

Stanford CoreNLP deterministic coreference resolution system is a good example of such coreference resolution solutions [145], [89], [90]. It uses a multi-sieve approach based on a sieve that applied tiers of deterministic rules ordered from high precision to lowest precision one be one. Each sieve built on the result of the previous cluster output. An extension of this multi-sieve approach was presented at the CoNLL 2011 shared task. The major modifications made to the earlier system were addition of five more sieves, a mention detection module at the beginning and, finally, a post-processing module at the end to provide the result in OntoNotes format [142].

A recent rule-based algorithm developed by [180] also used dependency syntax as input. It targeted the coreference types which were not annotated by the CoNLL 2012 shared task, e.g. cataphora, compound modifier, *i*-within-*i*, etc.

6.2.1.2 *Statistical and Machine Learning based Resolution*

Machine learning-based coreference models can be classified into mention-pair models, entity-mention models and ranking models [160]. The mention-pair model treats coreference as a collection of pairwise links. A classifier is used to decide whether two noun phrases are co-referent. This stage was followed by reconciling the links with greedy partitioning or clustering. The most well-known algorithm for mentioning instance creation was heuristic mention creation method created by [158]. For instance, creation it only considered annotated noun phrases. A modified approach by [124] enforced another constraint, i.e. that a positive instance between a non-pronominal instance and antecedent could only be created if antecedent was non-pronominal too.

The next stage of mention-pair models was the training a classifier. Decision trees and random forests were widely used for this task, e.g. [2], [109], [91]. Statistical learners (e.g., [12], [53]), memory learners, e.g. TiMBL [37], and rule-based learners, e.g., [33] were also popular.

The following phase of the mention-pair model was generating a noun phrase partition. The model, trained on an annotated corpus, could be tested on a test-set in order to obtain the coreference chains. Multiple clustering techniques were used for this task. Some of the well-known ones were best-first clustering [123], closest-first clustering [158], correlational clustering [108], Bell Tree beam search [103] and graph partitioning algorithms [126], [107].

In the closest first clustering [158] all possible mentions before the mention under consideration were processed from right to left, processing the nearest antecedent first. A modified approach by [123] linked the current instance instead with the antecedent which is classified as true and has the maximum likelihood. Correlational clustering algorithm [108] measured the degree of inconsistency by including a node in a partition and making repairs. Thus, the assignment to the partition was not only dependent on the distance measure to the node but on a distance among all the nodes in a partition. In graph-partitioning the nodes of the graph represented the mentions and the edge weights represented the likelihood of assignment of the pairs [126], [107].

For combining the phases of classification and effective partitioning in mention-pair models, Integer Linear Programming (ILP) [39], [50] was used as well. According to [39], this task was suitable for ILP as coreference resolution required to take into consideration the likelihood of two mentions being co-referent during pairwise classification and final cluster assignment. Also, [49] proposed a model which eliminated the classification phase entirely. Their model had only two phases of mention detection and clustering.

One of disadvantages of mention-pair models was the constraint of transitivity which did not always hold true and another – it only determined how good an antecedent was with respect to the anaphoric noun phrase and not how good it was with respect to other available antecedents [160]. The entity-mention models and the mention-ranking models were proposed with the aim of overcoming these

disadvantages. Entity mention model aimed to classify whether a noun phrase was co-referent with a preceding partially formed cluster instead of an antecedent [122]. Instances were represented as cluster-level features instead of pairwise features. The cluster-level features, e.g., gender and number, were defined over subsets of clusters using the “ANY”, “ALL”, “MOST”, etc. predicates [179], [103]. The first order probabilistic model by [36] also attempted to use cluster-level features for coreference resolution as well as most recent models [29], [31].

Mention-pair models used a binary classifier to decide whether an antecedent was co-referent with the mention by providing only a “YES” or “NO” result but giving no information on how good one antecedent was compared to the other. The ranking models circumvented this disadvantage by ranking the mentions and choosing the best candidate antecedent [160]. In mention-ranking algorithm by [40] the classification function was replaced by a ranking loss. Another mention ranking model used surface features [47] as well as log-linear model for antecedent selection. It outperformed the Stanford system [90] which was the winner of CoNLL 2011 shared task [142].

The mentioned rankers were not able to effectively exploit past decisions for current decisions [160]. This motivated the “cluster ranking” algorithms. The cluster ranking approaches aimed at combining the best of the entity-mention models and the ranking models. Recent deep learning models, e.g [29], have also used a combination of mention ranker and cluster ranker for coreference resolution. Also, the mention-ranking model is not able to differentiate between anaphoric and non-anaphoric noun phrases. Recent deep learning-based mention ranking models, such as [30], [31] and [176], [177] are able to overcome this disadvantage by learning anaphoricity together with mention ranking.

6.2.1.3 *Deep Learning Models for Coreference Resolution*

One of the first non-linear mention ranking models for coreference resolution aimed at learning different feature representations for anaphoricity detection and antecedent ranking by pre-training on these two individual subtasks [176]. This approach addressed 2 major issues in coreference resolution – the identification of non-anaphoric references in the text and complicated feature conjunction in linear models. This model handled the above issues by introducing a new neural network model with raw unconjoined features as inputs in order to learn intermediate representations.

The non-linear coreference model by [177] showed that the coreference task could benefit from modelling global features regarding entity clusters by augmenting the neural network based mention-ranking model [61]. It was executed by incorporating entity-level information produced by a Recurrent Neural Network (RNN) that ran over the candidate antecedent-cluster. The idea was to capture the history of previous decisions along with the mention-antecedent compatibility.

Roughly during the same time algorithm was proposed by [31] which instead defined a different cluster ranking model to induce global information. Their approach was based on the idea of incorporating entity-level information. The architecture of this neural network consisted of mainly four sub-parts: the mention-pair encoder which passes features through a Feed-Forward Neural Network (FFNN), a cluster-pair encoder which uses pooling over mention pairs to produce distributed representations of cluster pairs, a mention ranking model and the cluster ranking module to score pairs of clusters.

The end-to-end neural model [91] is jointly modelled for mention detection and coreference resolution. This model begins with the construction of high-dimensional word embeddings (a concatenation of Glove, Turian and character embeddings) to represent the words of an annotated document. During inference the best scoring antecedent is chosen as the most probable antecedent and coreference chains are formed using the property of transitivity. This model used a very large deep neural network that is difficult to maintain.

Deep learning coreference resolution systems such as [29], [31] and [91] use word vectors which depict semantic relationships between words. These systems also implicitly capture the dependencies between mentions (using RNN, LSTMs and gated recurrent units (GRUs)). However, these systems are difficult to maintain. On the other hand, regarding the deep learning-based coreference resolution algorithms, the dependency on features decreased over time. This was mainly due to the pre-trained word embeddings. Unlike the Stanford CoreNLP deterministic coreference resolution system [29], [31], system by [91] used minimal mention-level and mention-antecedent pair features. On the other hand, the latter model is still a mention ranking model which chooses the highest scoring antecedent without using any cluster-level information. According to earlier deep learning works which used cluster-level information, such as [31], [177], etc., this information is necessary to avoid linking incompatible mentions to partially formed coreference chains.

6.2.1.4 *Coreference Resolution for Morphologically Rich Languages*

Only first steps of research have been performed to solve coreferences in Lithuanian. Rule-based solution of coreference resolution in Lithuanian medical records was proposed by [188]. The rules constructed are based on POS, NER information and external databases and evaluation was performed by analyzing 100 articles that have been pre-annotated in Lithuanian Language Coreference Corpus [189] in addition to the transcribed records of medical reception.

Considering languages that are more grammatically similar to Lithuanian than English, related work on coreference resolution for Latvian (the only other Baltic language beside Lithuanian) and Slavic languages (Polish, Russian, and Czech) is reviewed.

For Latvian, the only model is LVCoref [211], which is a rule-based system that uses an entity-centric model. It focuses on named entity matches and uses Hobbs' algorithm for pronoun resolution.

For Polish, there is a rule-based Ruler [131] which specifically targets pronouns. Also, BARTEK [203] is an adaptation of BART (originally for English) to Polish. Mixed Polish coreferences resolution approach combines neural networks architecture with the sieve-based approach [221].

For Russian, RU-EVAL-2014 [165] was an evaluation campaign of anaphora and coreference resolution tools that used a wide variety of approaches. The evaluation was performed on Russian Coreference Corpus (RuCur). Machine learning approaches, e.g. [220], were also used.

For Czech, coreferences are annotated on the tectogrammatical layer of Prague Dependency Treebank (PDT). Their first coreference resolution model was rule based [193]: all possible candidates are collected, and their list is narrowed down with 8 filters, then an antecedent is selected from remaining ones based on the closest distance to corefering object. It was adapted two older English language models to Czech language and used Decision Tree C5 for the classifier-based approach, while the ranker-based approach employed the averaged perceptron algorithm [199]. The latter approach provided better results. Treex CR [129] was developed for the Czech language and adapted to English, Russian, and German, although in this case coreference labels were projected, which negatively affected the results [216].

In summary, the rule-based solutions are easier in terms of adaptability, they provide comparable results when good training data is not available [211]. Many of more advanced solutions cannot be fully adapted for other (e.g., under-resourced) languages due to the lack of available linguistic resources. For example, BART supported 64 feature extractors, but because of lack of language-specific resources for Polish, only 13 could be utilized [211].

6.2.2 Issues

Coreference resolution includes many different types of references. Some of these references are rare and some types are not labelled by current coreference datasets [180]. This has led to research targeting specific types of references like *multi-antecedent references* [172], *abstract anaphora* [105]

and *one anaphora* [58]. Also, some types of references (e.g. split anaphora [113] are extremely hard to resolve automatically mainly because they require external world knowledge. Though the usefulness of world knowledge for a coreference system has been known, early mention-pair models, e.g. [158], [123], [178], did not incorporate any form of world knowledge into the system. As knowledge resources became less noisy and more available, some researchers included them into coreference resolution in variety of forms: web-based encyclopedias [170], unannotated data [38], coreference annotated data [10], and knowledge bases like YAGO, FrameNet and Wordnet [47]. World knowledge was mainly incorporated as features into the mentioned pair models and cluster ranking models with mixed success, e.g. [47] reported only minor performance gains using world knowledge. Thus, instead of representing commonsense knowledge as features, some models used predicates to encode commonsense relations [135].

6.2.3 Extended Coreference Resolution: Selected Problem

6.2.3.1 *Multiple Antecedence Coreference (MAC) Resolution*

Coreference resolution research is mostly focused on single antecedence and multiple antecedence for cases where the antecedences are in a simple conjunctive (e.g. in the same position) form as in:

- **Bob** and his friend **Paul** are driving to school in the morning. They are still very tired.

State-of-the-Art coreference resolution software struggles with split anaphora (discontinuous sets) in general cases, see the example: “Bob is driving to school with his friend Paul in the morning. They are both very tired.” The pronoun “they” is not resolved as a reference to **Paul** and **Bob** by *any* of the state-of-the-art tools. There are two options to enhance the performance of the coreference algorithms:

- Enhance the performance of a state-of-the-art coreference tool by adding additional rules in case of rule-based tool or retrain or transfer learn in case of machine learning based tools. The training approach is a good option, if there is enough training data (in a supported format), which targets split antecedents, and the used method or software supports the training.
- Rewrite (derive) the sentence containing the split antecedents to its conjunctive trivial form (if the form exists) and right shift a relative clause to a separate sentence as all tools are managing the resulting construct very well:
    - **Bob** is driving to school with his friend **Paul** in the morning. => **Bob** and his friend **Paul** are driving to school in the morning.
    - When **Paul** helps **Bob** and **Bob** helps **Paul**, … => When **Paul** and **Bob** help each other (reciprocal, reflexive form), …
    - **Paul** told **Bob** to attend the party. => ??? No conjunctive form is available here without changing the meaning of the sentence. This is true for split antecedents in different positions with relative clauses, where the relative clause may be in its infinitive form (like in the example) or is introduced by a relative or personal pronoun.
    - **Bob** and **Paul**, who are still very tired, are driving to school. => **Bob** and **Paul** are driving to school. They are very tired. Even the two antecedents are in a single conjunctive position, the relative clause in its canonical position must be right shifted to a separate sentence exchanging the relative pronoun “who” with the personal pronoun “they”.

IOSB will contribute to the second option, which implies, that the existing linguistic parser Pikes may be used with no changes. The approach is as follows:

- Detect the correct reference forms (pronouns we, both, they).
- Analyze the antecedents in scope and test, if they are candidates for non-trivial split antecedents.
- Rewrite the sentence, if possible.

6.2.3.2 *Tools and Models to apply for extended coreference resolution*

We are planning to use the following tools and models for the coreference:

1. SpaCy as a main framework;
2. NER -- from SpaCy + additional resources (if needed);
3. Coreference resolution:
    a. for Lithuanian language:
        1. Try reusing models of grammatically similar languages;
        2. Adapting models of grammatically similar languages;
        3. Using approaches less dependent on linguistic resources.
    b. English & German:
        1. Many approaches, tools and resources to choose from; for multiple antecedent coreference resolution (per IOSB suggestion) consider [172] (see section 6.2.1).

6.3 Extended Named Entities Resolution

6.3.1 State-of-the-Art of Entity Coreference Resolution

Entity Coreference Resolution is the task of resolving all the mentions in a document that refer to the same real world entity and is considered as one of the most difficult tasks in natural language understanding. However, named entity resolution as separate case is usually attempted but not necessarily described individually. As it is usually part of general coreference resolution and thus the same methods are used, named entity coreference resolution there is introduced in a compact way. For more details, please see section 6.2.1.

Coreference resolution has been targeted with four different approaches (Mention-Pair models, Mention-Ranking models, Entity-Based models, Latent Structured models), that were essentially built on top of each other in a hierarchical way [159]. Essentially, Mention-Pair and Mention-Ranking models set the foundation required for the Entity-Based and Latent-Structure models. A detailed review of all 4 approaches described above can be found in [124]. The deep learning approaches are introduced next. As previously introduced approaches, deep learning models start with mention-ranking approaches and gradually move to Entity-Based and Latent-structure approaches.

Following the methodologies used in early coreference resolution, neural approaches can be identified in similar categories: Mention-Ranking models, Entity-Based models, Latent-Structure models and Language Modeling models. While mention-pair models are very simplistic and have been proven useful in the past, they also have disadvantages that are solved through Mention-Ranking. Hence, these approaches separately were never attempted in neural implementation. However, Mention-Ranking models are essential part of Entity-Based models as they utilize the scoring functions to prune possible antecedents [159]. Deep learning allowed the effective implementation of Latent-

structure models combined with graphs and clusters. Finally, the best approaches for coreference resolution have been identified through Language-Modeling models, even though they are not directly aimed at this task.

6.3.2 Entity Coreference Resolution: Best Performance

According to the results of coreference resolution it is apparent that the best overall results are from either language modelling or latent entity approaches. Model by [91] is considered a baseline. However, [60] performed worse than this baseline, because model by [91] is based on a 5-model ensemble. A lot of the proposed models are built on top of previous baseline ones, thus the results are directly related and changes in performance scores comes from a parameter fine tuning, e.g. in [29] the fine tuning is done with the reinforcement learning. All the approaches, except for [176] and [177], used dropout to avoid overfitting as well as some type of word embeddings. For example, in [29], [31] they used pre-trained word2vec embeddings [112] on the Gigaword corpus and Polyglot embeddings [1]. In [91] and all related realizations use a combination of Glove [137] and CNN character embeddings [138] which are extended by the use of ELMo embeddings in [93]. Also, [29] and [31] used RMSprop [224] for the parameter optimization during learning stage, while in other approaches Adam [77] was mainly used. While Feed-Forward Neural Network remains the same in all mentioned implementations, in [93] they used Highway BiLSTMs instead of simple BiLSTMS.

6.3.3 Extended named entities resolution: Possible Contributions to Consider

1. Gender bias resolution [124]: see details in Figure 36
2. Out-of-domain performance [124] or generalization problem: model performance decrease, when named entities are changed to names that do not occur in the training set.
3. It is not clear regarding performance and results "…they do not penalize systems for not identifying any mentions by name to an entity and they reward systems even if systems find correctly mentions to the same entity but fail to link these to a proper name (she–the student–no name)" [222]
4. Incorporation of sophisticated knowledge sources: "Recent results suggest that the performance of coreference models that do not employ sophisticated knowledge is plateauing" [124]

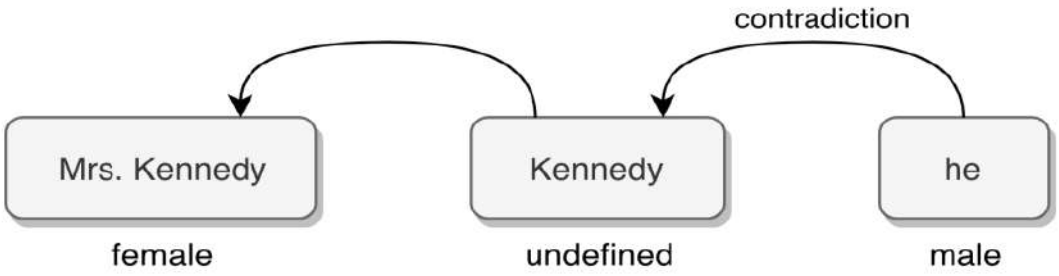


Figure 36. Example of contradictions in the linking process: arrows represent positive coreference links.

6.3.4 Tools and Models to Apply for Extended Named Entity Coreference Resolution

We are planning to use the following tools and models for the coreference:

1. SpaCy as a main framework;
2. NER -- from SpaCy + additional resources (if needed);
3. Named entity coreference resolution:
    1. Reuse / adapt general purpose methods, e.g. [183], to make coreference models more robust

2. Reuse / adapt cross-lingual coreference methods, e.g. [85]
3. Reuse / adapt projection-based coreference resolution methods, e.g. [216]

6.4 Deploy tools for languages with weak machine translation support

6.4.1 State-of-the-Art

MT engines automate the transfer of text from one language to another. MT is broken up into three primary methodologies: rules-based, statistical, and neural (which is the new player). The most widespread MT methodology is statistical, which (in very brief terms) draws conclusions about the interconnectedness of a pair of languages by running statistical analyses over annotated bilingual corpus data using n-gram models. When a new source language phrase is introduced to the engine for translation, it looks within its analyzed corpus data to find statistically relevant equivalents, which it produces in the target language. MT can be useful as a productivity aid to translators, changing their primary task from translating a source text to a target text to post-editing the MT engine's target language output. It is not recommended to use raw MT output in localizations, but if the working community is trained in the art of post-editing, MT can be a useful tool to help them make large volumes of contributions.

The machine translation systems currently developed and available are based on two translation methods: rules-based or statistical. The first is the analysis of texts using special algorithms, while the statistical method is based on analogous texts translated into a foreign language. The more bilingual texts are collected, the smoother and better the statistical translation will be. However, there are far fewer unpublished texts in the world (including Lithuanian) translated into English than, for example, German. Therefore, it should not be surprising that the machine translation system, which is based on a statistical method, translates from, for example, German to a much higher quality than from Lithuanian. German-English is still the most popular language pair when it comes to online and offline machine translations.

Statistical machine translation is trained in two types of data: human translated parallel bilingual texts and monolingual text in the target language. Also, dictionaries, terminology, ontologies, databases of named entities are needed to construct the model. The more text (data for training) is presented, the better results will be achieved (the higher translation accuracy will be obtained). Also important are the quality of the texts (misspellings, punctuation, etc.), the relevance of their field, and the accuracy of the alignment. It is worth mentioning that statistical machine translation translates better to the same area of the text on which it was trained.

Machine translation models learn from available data to recognize which sentence to translate into (sentence level alignment), to recognize which word to translate into which (word alignment and translation probabilities), how the translated sentence (sentence structure) should look (use of existing language model).

After analyzing the machine translation tools (see Table 15), it can be said that the MOSES tool is noteworthy. Universal open source software machine translation software MOSES (see Figure 37) developed by EuroMatrix, a project supported by the European Commission. A set of translated texts (parallel body) is needed. Once a model is ready (after training), an efficient search algorithm quickly determines the maximum (the highest) probability translation among the exponential number of choices.

Table 15. Machine translation tools, toolkits and frameworks

No	Title	Cloud vs local	Supported languages (from/to)	Free vs commercial	Comments	Links
			Translation tools			
1	DeepL	Cloud / Local	DE/EN, DE/EN, EN/ LT, LT/EN (in experiments did not work with LT)	Free / Paid	In order to allow customers to make use of the services, customers are granted access to the DeepL application programming Interface ("API") and to an extended version of the DeepL web translator.	https://www.deepl.com/en/home
2	SDL Machine Translation		DE/EN, EN/DE, EN/LT, LT/EN	Paid	An enterprise-grade solution for those looking to apply the latest in neural machine translation to automatically translate content.	https://www.sdl.com/
3	Omniscien Technologies Language Studio	Cloud / Local	DE/EN, EN/DE, EN/LT, LT/EN	Paid	Offers a broad range of integration options with CAT tools and TM's as well as desktop integration and API options for bespoke solutions.	https://omniscien.com/language-studio/
4	IBM Watson Language Translator	Cloud / Local	DE/EN, EN/DE, EN/LT, LT/EN	Free / Paid	By default all language pairs leverage neural machine translation. Both rule-based and statistical models developed by IBM Research. Neural Machine Translation models available through the Watson Language Translator API for developers.	https://www.ibm.com/watson/services/language-translator/
5	PROMT	Cloud / Local	DE/EN, EN/DE	Free / Paid	System uses Hybrid, rules-based, SMT and neural methods. Risks: Russian company.	https://www.promt.com/
6	Lucy LT	Local	DE/EN, EN/DE	Free / Paid	The Machine Translation Solution is fully integrated in leading Translation Management tools such as SDL Trados Studio.	https://www.lucysoftware.com/english/machine-translation/integration-capabilities/
7	Babylon NG	Cloud	DE/EN, EN/DE	Free / Paid	Desktop tool/translator. Prompts to install the Babylon Toolbar, a browser hijacker	https://www.babylon-software.com/

					which is difficult to remove.	
8	OpenLogos Machine Translation	Cloud / Local	DE/EN, EN/DE	Free / Paid	Various text documents in different formats can be submitted to the system and within a short amount of time are translated into different target languages. System is rule-based, deep transfer.	http://logos-os.dfki.de/
9	Systran	Local	DE/EN, EN/DE	Paid	System uses Hybrid rules-based and SMT methods.	https://www.systransoft.com/
Translation frameworks						
1	Moses (mosesdecoder, moses-smt, Moses for Mere Mortals)	Local	should be trained	free	A statistical machine translation system that allows you to automatically train translation models for any language pair. Drop-in replacement for Pharaoh, features factored translation models and decoding of confusion networks.	https://github.com/moses-smt/mosesdecoder
2	Open-NMT	Local	should be trained	free	Provides greatly documented, modular and readable code for fast training and efficient performance of the models.	https://opennmt.net/
3	Apertium	Cloud / Local	should be trained	free	A free/open-source machine translation platform. A toolbox to build open-source shallow-transfer machine translation systems, especially suitable for related language pairs: it includes the engine, maintenance tools, and open linguistic data for several language pairs. Rule-based, shallow transfer; all programs and language data are free and open source.	https://github.com/apertium
Translation toolkits						
1	cdec	Local	should be trained	free	Support SMT method.	https://github.com/redpony/cdec

2	giza-pp	Local	should be trained	free	Support SMT method. Are used for IBM.	https://github.com/moses-smt/giza-pp
3	teny	Local	should be trained	free	For low-resource machine translation.	https://github.com/vchahun/teny
4	nematus	Local	should be trained	free	Support NMT method.	https://github.com/EdinburghNLP/nematus
5	mtrain	Local	should be trained	free	Support SMT and NMT methods. Are used for Moses and Nematus.	https://github.com/ZurichNLP/mtrain
6	thumt	Local	should be trained	free	Support NMT method.	https://github.com/THUNLP-MT/THUMT

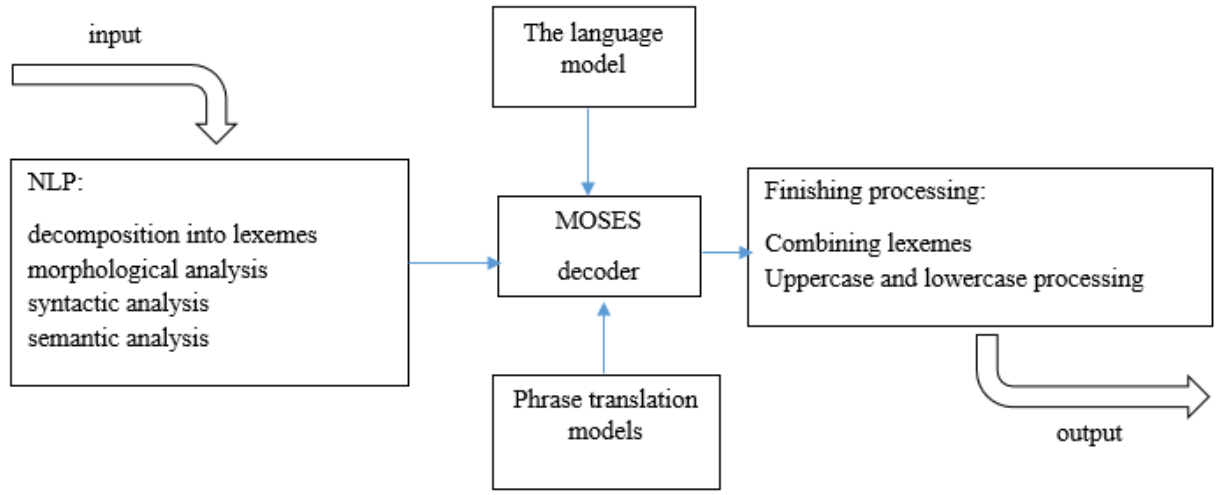


Figure 37. Machine translation workflow in MOSES.

6.4.2 Plans for machine translation deployment (in case, when pipeline, which includes MT, is used)

1. The tool to deploy should be chosen, further testing is required. The following types of local (non-cloud) tools will be investigated.
    1. MT tools supporting required languages, e.g. DE/EN, LT/EN.
    2. MT platforms and possibility to train our own MT frameworks, e.g. using EU parallel corpora (could be too time consuming, and is not directly part of the project).
2. Deployment of tools for pipeline with MT, see concise description of the pipeline:
    1. translate to English;
    2. preprocessing;
    3. Event detection (contains coreference resolution, named entity resolution, other types of deep linguistic analyses).

7. Summary and conclusions

This deliverable defines the basic strategy that will be followed regarding heterogeneous data stream processing. Specifically, the Crawling tools that will be a part of the PREVISION tools are presented. A detailed description regarding dataset generation from crawled data was also provided. Finally, a draft proposal of the functionality of the crawling tools over the composed datasets were presented. The diverse data sources such as data sources of traffic, telecommunication, and financial data will be managed by developed interfaces able to Extract Transform and Load (ETL). The design of ETL approaches has tested and verified that it is suitable for the PREVISION architecture and subsequently the PREVISION's application.

Regarding the analysis of the data sources consists of visual streams four different services reported within PREVISION. Firstly, the activity recognition service that will be able to detect activities and filter them according to the LEA needs. Secondly, the Person re-identification service that will be able to recognize persons deploying an attribute-based searching functionality. Furthermore, a synthetic dataset has already been reported in order to train the PREVISION model into different environmental (weather, time, etc.) conditions. The detection of faces and subsequently the recognition of them is also a part of PREVISION's visual analytics tools. The deployment of detection and sequentially the recognition of target faces using publicly available data is reported while future steps include incorporation of innovative augmented techniques. Finally, the crisis event detection and especially the detection of fire, smoke and flood deep learning-based approaches will be used in order to efficiently detect the key frames that describe a crisis event during the video stream.

The analysis of social data is another important task of PREVISION's platform. Specifically, the strategies for the detection of communities and the identification of the key actors are presented. Additionally, a framework called "actor identity resolution framework" is also proposed. Specifically, the detection of the accounts belong to the same natural person is the target of the proposed framework. The experimental results show that the proposed framework is able to detect effectively linked accounts using Twitter's dataset. Future steps include experiments of deferent features to be considered for the analysis. These features could be extracted by deep linguistic models. The processing of feature extraction includes the features of different natural languages and a variety of entities to be considered.

8. References

[1] Al-Rfou' R, Perozzi B, Skiena S (2013) Polyglot: Distributed word representations for multilingual NLP. In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, Sofia, Bulgaria, pp 183–192.

[2] Aone C, Bennett SW (1995) Evaluating automated and manual acquisition of anaphora resolution strategies. In: Proceedings of the 33rd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp 122–129.

[3] Avgerinakis, K., Giannakeris, P., Briassouli, A., Karakostas, A., Vrochidis, S., & Kompatsiaris, I. (2017). LBP-flow and hybrid encoding for real-time water and fire classification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (pp. 412-418).

[4] Avgerinakis, K., Giannakeris, P., Briassouli, A., Karakostas, A., Vrochidis, S., & Kompatsiaris, I. (2017). Intelligent traffic city management from surveillance systems (CERTH-ITI). NVIDIA AI City Challenge 2017.

[5] Avgerinakis, K., Moumtzidou, A., Andreadis, S., Michail, E., Gialampoukidis, I., Vrochidis, S., & Kompatsiaris, I. (2017, September). Visual and Textual Analysis of Social Media and Satellite Images for Flood Detection@ Multimedia Satellite Task MediaEval 2017. In MediaEval.

[6] Awad, G., Butt, A., Curtis, K., Lee, Y., Fiscus, J., Godil, A., ... & Quénot, G. (2019, November). Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. In Proceedings of TRECVID (Vol. 2019).

[7] Azaza, L., Kirgizov, S., Savonnet, M., Leclercq, E., & Faiz, R. (2015, November). Influence assessment in twitter multi-relational network. In 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) (pp. 436-443). IEEE.

[8] Baldwin B (1997) Cogniac: high precision coreference with limited knowledge and linguistic resources. In: Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts, Association for Computational Linguistics, pp 38–45.

[9] Bello-Orgaz, G., Hernandez-Castro, J., & Camacho, D. (2017). Detecting discussion communities on vaccination in twitter. Future Generation Computer Systems, 66, 125-136.

[10] Bengtson E, Roth D (2008) Understanding the value of features for coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp 294–303.

[11] Benoit, K., Watanabe, K., Wang, H., Nulty, P., Obeng, A., Müller, S., & Matsuo, A. (2018). quanteda: An R package for the quantitative analysis of textual data. Journal of Open Source Software, 3(30), 774.

[12] Berger AL, Pietra VJD, Pietra SAD (1996) A maximum entropy approach to natural language processing. Computational linguistics 22(1):39–71.

[13] Biermann J. et al., "From Unstructured to Structured Information in Military Intelligence: Some Steps to Improve Information Fusion", online available, 2004.

[14] Biermann J. et al., "From Unstructured to Structured Information in Military Intelligence: Some Steps to Improve Information Fusion", online available, 2004.

[15] Bischke, B., Bhardwaj, P., Gautam, A., Helber, P., Borth, D., & Dengel, A. (2017, September). Detection of Flooding Events in Social Multimedia and Satellite Imagery using Deep Neural Networks. In MediaEval.

[16] Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment, 2008(10), P10008.

[17] Bonacich, P., & Lloyd, P. (2001). Eigenvector-like measures of centrality for asymmetric relations. Social networks, 23(3), 191-201.

[18] Brandes, U. (2001). A faster algorithm for betweenness centrality. Journal of mathematical sociology, 25(2), 163-177.

[19] Bregonzio, M.; Gong, S.; Xiang, T. Recognising action as clouds of space-time interest points. In Proceedings of the CVPR 2009, Miami Beach, FL, USA, 20–25 June 2009; Volume 9, pp. 1948–1955.

[20] Brennan SE, Friedman MW, Pollard CJ (1987) A centering approach to pronouns. In: Proceedings of the 25th annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp 155–162.

[21] Brin, S., & Page, L. (2012). Reprint of: The anatomy of a large-scale hypertextual web search engine. Computer networks, 56(18), 3825-3833.

[22] Carreira, J.; Zisserman, A. Quo vadis, action recognition? A new model and the Kinetics dataset. In Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4724–4733.

[23] Chan, A. B., & Vasconcelos, N. (2009). Layered dynamic textures. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31(10), 1862-1879.

[24] Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., & Vakali, A. (2017, April). Measuring# gamergate: A tale of hate, sexism, and bullying. In Proceedings of the 26th international conference on world wide web companion (pp. 1285-1290).

[25] Chaudhry, R.; Ravichandran, A.; Hager, G.; Vidal, R. Histograms of oriented optical flow and Binet–Cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 1932–1939.

[26] Chen, J., Zhao, G., Salo, M., Rahtu, E., & Pietikainen, M. (2013). Automatic dynamic texture segmentation using local descriptors and optical flow. IEEE Transactions on Image Processing, 22(1), 326-339.

[27] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. CoRR, abs/1606.00915. Retrieved from http://arxiv.org/abs/1606.00915

[28] Chino, D. Y., Avalhais, L. P., Rodrigues, J. F., & Traina, A. J. (2015, August). Bowfire: detection of fire in still images by integrating pixel color and texture analysis. In 2015 28th SIBGRAPI Conference on Graphics, Patterns and Images (pp. 95-102). IEEE.

[29] Clark K, Manning CD (2015) Entity-centric coreference resolution with model stacking. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol 1, pp 1405–1415.

[30] Clark K, Manning CD (2016a) Deep reinforcement learning for mention-ranking coreference models. arXiv preprint arXiv:160908667.

[31] Clark K, Manning CD (2016b) Improving coreference resolution by learning entity-level distributed representations. arXiv preprint arXiv:160601323.

[32] Clauset, A., Newman, M. E., & Moore, C. (2004). Finding community structure in very large networks. Physical review E, 70(6), 066111.
[33] Cohen WW, Singer Y (1999) A simple, fast, and effective rule learner. AAAI/IAAI 99:335–342.
[34] Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 809–819.
[35] Covington, M. A., & McFall, J. D. (2010). Cutting the Gordian knot: The moving-average type–token ratio (MATTR). Journal of quantitative linguistics, 17(2), 94-100.
[36] Culotta A, Wick M, McCallum A (2007) First-order probabilistic models for coreference resolution. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pp 81–88.
[37] Daelemans W, Zavrel J, Van Der Sloot K, Van den Bosch A (2004) Timbl: Tilburg memory-based learner. Tilburg University.
[38] Daumé III H, Marcu D (2005) A large-scale exploration of effective global features for a joint entity detection and tracking model. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp 97–104.
[39] Denis P, Baldridge J (2007) Joint determination of anaphoricity and coreference resolution using integer programming. In: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pp 236–243.
[40] Denis P, Baldridge J (2008) Specialized models and ranking for coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp 660–669.
[41] Derpanis, K. G., Lecce, M., Daniilidis, K., & Wildes, R. P. (2012, June). Dynamic scene understanding: The role of orientation features in space and time in scene classification. 2012 IEEE Conference on Computer Vision and Pattern Recognition, (pp. 1306-1313).
[42] Dimitropoulos, K., Barmpoutis, P., Kitsikidis, A., & Grammalidis, N. (2017). Classification of Multidimensional Time-Evolving Data using Histograms of Grassmannian Points. IEEE Transactions on Circuits and Systems for Video Technology, PP(99), 1-1.
[43] Dollár, P.; Rabaud, V.; Cottrell, G.; Belongie, S. Behavior recognition via sparse spatio-temporal features. In Proceedings of the 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Beijing, China, 15–16 October 2005; pp. 65–72.
[44] Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2625–2634
[45] Dongen, S. (2000). A cluster algorithm for graphs.
[46] Dubois, E., & Gaffney, D. (2014). The multiple facets of influence: Identifying political influentials and opinion leaders on Twitter. American Behavioral Scientist, 58(10), 1260-1277.
[47] Durrett G, Klein D (2013) Easy victories and uphill battles in coreference resolution. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp 1971–1982.

[48] Fabbri, M., Lanzi, F., Calderara, S., Palazzi, A., Vezzani, R., & Cucchiara, R. (2018). Learning to detect and track visible and occluded body joints in a virtual world. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 430-446).

[49] Fernandes ER, Dos Santos CN, Milidi´u RL (2012) Latent structure perceptron with feature induction for unrestricted coreference resolution. In: Joint Conference on EMNLP and CoNLL-Shared Task, Association for Computational Linguistics, pp 41–48.

[50] Finkel JR, Manning CD (2008) Enforcing transitivity in coreference resolution. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, Association for Computational Linguistics, pp 45–48.

[51] Freeman, L. C. (1978). Centrality in social networks conceptual clarification. Social networks, 1(3), 215-239.

[52] Fritz, M., Leibe, B., Caputo, B., & Schiele, B. (2005). Integrating representative and discriminant models for object category detection. Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, 2, pp. 1363-1370.

[53] Ge N, Hale J, Charniak E (1998) A statistical approach to anaphora resolution. In: Sixth Workshop on Very Large Corpora.

[54] Gialampoukidis, I., Kalpakis, G., Tsikrika, T., Vrochidis, S., & Kompatsiaris, I. (2016, August). Key player identification in terrorism-related social media networks using centrality measures. In 2016 European Intelligence and Security Informatics Conference (EISIC) (pp. 112-115). IEEE.

[55] Giannakeris, P., Avgerinakis, K., Karakostas, A., Vrochidis, S., & Kompatsiaris, I. (2018, June). People and vehicles in danger-A fire and flood detection system in social media. In 2018 IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP) (pp. 1-5). IEEE.

[56] Giannakeris, P., Kaltsa, V., Avgerinakis, K., Briassouli, A., Vrochidis, S., & Kompatsiaris, I. (2018). Speed estimation and abnormality detection from surveillance cameras. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 93-99).

[57] Goga, O., Lei, H., Parthasarathi, S. H. K., Friedland, G., Sommer, R., & Teixeira, R. (2013, May). Exploiting innocuous activity for correlating users across sites. In Proceedings of the 22nd international conference on World Wide Web (pp. 447-458).

[58] Goldberg AE, Michaelis LA (2017) One among many: Anaphoric one and its relationship with numeral one. Cognitive science 41(S2):233–258.

[59] Grosz BJ, Weinstein S, Joshi AK (1995) Centering: A framework for modeling the local coherence of discourse. Computational linguistics 21(2):203–225.

[60] Gu JC, Ling ZH, Indurkhya N (2018) A study on improving end-to-end neural coreference resolution. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, Springer, pp 159–169.

[61] Haghighi A, Klein D (2009) Simple coreference resolution with rich syntactic and semantic features. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3, Association for Computational Linguistics, pp 1152–1161.

[62] Hara, K., Kataoka, H., & Satoh, Y. (2018). Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 6546-6555).

[63] Harabagiu SM, Bunescu RC, Maiorano SJ (2001) Text and knowledge mining for coreference resolution. In: Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies, Association for Computational Linguistics, pp 1–8.

[64] Harabagiu SM, Maiorano SJ (1999) Knowledge-lean coreference resolution and its relation to textual cohesion and coherence. The Relation of Discourse/Dialogue Structure and Reference.

[65] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[66] Hobbs JR (1978) Resolving pronoun references. Lingua 44(4):311–338.

[67] https://www.privacy-regulation.eu/en/recital-26-GDPR.htm

[68] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., . . . others. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. IEEE CVPR.

[69] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," International journal of computer vision, vol. 104, no. 2, pp. 154–171, 2013

[70] Jiang, F., Yuan, J., Tsaftaris, S. A., & Katsaggelos, A. K. (2011). Anomalous video event detection using spatiotemporal context. Computer Vision and Image Understanding, 115(3), 323-333.

[71] Johansson, F., Kaati, L., & Shrestha, A. (2015). Timeprints for identifying social media users with multiple aliases. Security Informatics, 4(1), 7.

[72] Kalpakis, G., Tsikrika, T., Vrochidis, S., & Kompatsiaris, I. (2019, January). Identifying Terrorism-Related Key Actors in Multidimensional Social Networks. In International Conference on Multimedia Modeling (pp. 93-105). Springer, Cham.

[73] Kaltsa, V., Avgerinakis, K., Briassouli, A., Kompatsiaris, I., & Strintzis, M. G. (2018). Dynamic texture recognition and localization in machine vision for outdoor environments. Computers in Industry, 98, 1-13.

[74] Kaltsa, V., Briassouli, A., Kompatsiaris, I., Hadjileontiadis, L. J., & Strintzis, M. G. (2015, July). Swarm Intelligence for Detecting Interesting Events in Crowded Environments. IEEE Transactions on Image Processing, 24(7), 2153-2166.

[75] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).

[76] Kayes, I., Kourtellis, N., Quercia, D., Iamnitchi, A., & Bonchi, F. (2015, May). The social world of content abusers in community question answering. In Proceedings of the 24th International Conference on World Wide Web (pp. 570-580).

[77] Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980.

[78] Klausen, J. (2015). Tweeting the Jihad: Social media networks of Western foreign fighters in Syria and Iraq. Studies in Conflict & Terrorism, 38(1), 1-22.

[79] Korshunov, P., Araimo, C., De Simone, F., Velardo, C., Dugelay, J. L., & Ebrahimi, T. (2012, September). Subjective study of privacy filters in video surveillance. In 2012 IEEE 14th International Workshop on Multimedia Signal Processing (MMSP) (pp. 378-382). Ieee.

[80] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[81] Kubát, M., & Milička, J. (2013). Vocabulary richness measure in genres. Journal of Quantitative Linguistics, 20(4), 339-349.

[82] Kubát, M., Matlach, V., & Čech, R. (2014). QUITA. Quantitative Index Text Analyzer. Lüdenscheid: RAM-Verlag.

[83] Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; Serre, T. HMDB: A large video database for human motion recognition. In Proceedings of the International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011.

[84] Kuettel, D., Breitenstein, M. D., Van Gool, L., & Ferrari, V. (2010, June). What's going on? Discovering spatio-temporal dependencies in dynamic scenes. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 1951-1958). IEEE.

[85] Kundu, G., Sil, A., Florian, R., & Hamza, W. (2018). Neural cross-lingual coreference resolution and its application to entity linking. arXiv preprint arXiv:1806.10201.

[86] Lan, J., Li, J., Hu, G., Ran, B., & Wang, L. (2014). Vehicle speed measurement based on gray constraint optical flow algorithm. Optik, 125(1), 289-295.
Language Processing: Issues and Approaches, IGI Global, pp 185–201.

[87] Lappin S, Leass HJ (1994) An algorithm for pronominal anaphora resolution. Computational linguistics 20(4):535–561.

[88] Laptev, I.; Marszalek, M.; Schmid, C.; Rozenfeld, B. Learning realistic human actions from movies. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008, pp. 1–8.

[89] Lee H, Chang A, Peirsman Y, Chambers N, Surdeanu M, Jurafsky D (2013) Deterministic coreference resolution based on entity-centric, precision-ranked rules. Computational Linguistics 39(4):885–916.

[90] Lee H, Peirsman Y, Chang A, Chambers N, Surdeanu M, Jurafsky D (2011) Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In: Proceedings of the fifteenth conference on computational natural language learning: Shared task, Association for Computational Linguistics, pp 28–34.

[91] Lee H, Surdeanu M, Jurafsky D (2017a) A scaffolding approach to coreference resolution integrating statistical and rule-based models. Natural Language Engineering 23(5):733–762.

[92] Lee K, He L, Lewis M, Zettlemoyer L (2017b) End-to-end neural coreference resolution. arXiv preprint arXiv:170707045.

[93] Lee K, He L, Zettlemoyer L (2018) Higher-order coreference resolution with coarse to-fine inference. arXiv preprint arXiv:180405392.

[94] Lertniphonphan, K.; Aramvith, S.; Chalidabhongse, T.H. Human action recognition using direction histograms of optical flow. In Proceedings of the 2011 11th International Symposium on Communications & Information Technologies (ISCIT), Hangzhou, China, 12–14 October 2011; pp. 574–579.

[95] Li, D., Zhang, Z., Chen, X., Ling, H., & Huang, K. (2016). A richly annotated dataset for pedestrian attribute recognition. arXiv preprint arXiv:1603.07054.

[96] Li, Q., Zhou, T., Lü, L., & Chen, D. (2014). Identifying influential spreaders by weighted LeaderRank. Physica A: Statistical Mechanics and its Applications, 404, 47-55.

[97] Liang T, Wu DS (2004) Automatic pronominal anaphora resolution in english texts. International Journal of Computational Linguistics & Chinese Language Processing, Volume 9, Number 1, February 2004: Special Issue on Selected Papers from ROCLING XV 9(1):21–40.

[98] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., . . . Zitnick, C. L. (2014). Microsoft coco: Common objects in context. European conference on computer vision, (pp. 740-755).

[99] Lin, Y., Zheng, L., Zheng, Z., Wu, Y., Hu, Z., Yan, C., & Yang, Y. (2019). Improving person re-identification by attribute and identity learning. Pattern Recognition, 95, 151-161.

[100] Liu, L., Zhao, L., Long, Y., Kuang, G., & Fieguth, P. (2012). Extended local binary patterns for texture classification. Image and Vision Computing, 30(2), 86-99.

[101] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[102] Lu, X. (2012). The relationship of lexical richness to the quality of ESL learners' oral narratives. The Modern Language Journal, 96(2), 190-208.

[103] Luo X (2005) On coreference resolution performance metrics. In: Proceedings of the conference on human language technology and empirical methods in natural language processing, Association for Computational Linguistics, pp 25–32.

[104] Malhotra, A., Totti, L., Meira Jr, W., Kumaraguru, P., & Almeida, V. (2012, August). Studying user footprints in different online social networks. In 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (pp. 1065-1070). IEEE.

[105] Marasović A, Born L, Opitz J, Frank A (2017) A mention-ranking model for abstract anaphora resolution. arXiv preprint arXiv:170602256.

[106] Massanari, A. (2017). # Gamergate and The Fappening: How Reddit's algorithm, governance, and culture support toxic technocultures. New Media & Society, 19(3), 329-346.

[107] McCallum A, Wellner B (2003) Object consolidation by graph partitioning with a conditionally-trained distance metric. In: KDD Workshop on Data Cleaning, Record Linkage and Object Consolidation, Citeseer.

[108] McCallum A, Wellner B (2005) Conditional models of identity uncertainty with application to noun coreference. In: Advances in neural information processing systems, pp 905–912.

[109] McCarthy JF, Lehnert WG (1995) Using decision trees for coreference resolution. arXiv preprint cmplg/9505043.

[110] McNamara, D. S., Graesser, A. C., McCarthy, P. M., & Cai, Z. (2014). Automated evaluation of text and discourse with Coh-Metrix. Cambridge University Press.

[111] Mettes, P., Tan, R. T., & Veltkamp, R. C. (2017). Water detection through spatio-temporal invariant descriptors. Computer Vision and Image Understanding, 154, 182-191.

[112] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp 3111– 3119.

[113] Mitkov R (2014) Anaphora resolution. Routledge.

[114] Modeling unrestricted coreference in ontonotes. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, Association for Computational Linguistics, pp 1–27.

[115] Moosavi, S. A., Jalali, M., Misaghian, N., Shamshirband, S., & Anisi, M. H. (2017). Community detection in social networks using user frequent pattern mining. Knowledge and Information Systems, 51(1), 159-186.

[116] Mu, X., Zhu, F., Lim, E. P., Xiao, J., Wang, J., & Zhou, Z. H. (2016, August). User identity linkage by latent user space modelling. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1775-1784).

[117] Muhammad, K., Ahmad, J., & Baik, S. W. (2018). Early fire detection using convolutional neural networks during surveillance for effective disaster management. Neurocomputing, 288, 30-42.

[118] Mumtaz, A., Coviello, E., Lanckriet, G. R., & Chan, A. B. (2013, July). Clustering Dynamic Textures with the Hierarchical EM Algorithm for Modeling Video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(7), 1606-1621.

[119] Narayanan, A. et al. (2017). graph2vec: Learning distributed representations of graphs. arXiv preprint arXiv:1707.05005.

[120] Navarro, G. (2001). A guided tour to approximate string matching. ACM computing surveys (CSUR), 33(1), 31-88.

[121] Newman, M. E. (2008). The mathematics of networks. The new palgrave encyclopedia of economics, 2(2008), 1-12.

[122] Ng V (2010) Supervised noun phrase coreference research: The first fifteen years. In: Proceedings of the 48th annual meeting of the association for computational linguistics, Association for Computational Linguistics, pp 1396–1411.

[123] Ng V, Cardie C (2002b) Improving machine learning approaches to coreference resolution. In: Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics, pp 104–111.

[124] Ng, V. (2016). Entity Coreference Resolution. IEEE Intelligent Informatics Bulletin, 17(1), 7-13.

[125] Ngo, C.W.; Pong, T.C.; Zhang, H.J. Motion-based video representation for scene change detection. Int. J. Comput. Vis. 2002, 50, 127–142. [CrossRef]

[126] Nicolae C, Nicolae G (2006) Bestcut: A graph algorithm for coreference resolution. In: Proceedings of the 2006 conference on empirical methods in natural language processing, Association for Computational Linguistics, pp 275–283.

[127] Nie, Y., Jia, Y., Li, S., Zhu, X., Li, A., & Zhou, B. (2016). Identifying users across social networks based on dynamic core interests. Neurocomputing, 210, 107-115.

[128] Niebles, J.C.; Wang, H.; Fei-Fei, L. Unsupervised learning of human action categories using spatial-temporal words. Int. J. Comput. Vis. 2008, 79, 299–318.

[129] Novák, M. (2017). Coreference resolution system not only for Czech. In Proceedings of the 17th conference ITAT (pp. 193-200).

[130] Nurhadiyatna, A., Hardjono, B., Wibisono, A., Sina, I., Jatmiko, W., Ma'Sum, M. A., & Mursanto, P. (2013, September). Improved vehicle speed estimation using gaussian mixture model and hole filling algorithm. In 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (pp. 451-456). IEEE.

[131] Ogrodniczuk, M., & Kopeć, M. (2011). Rule-based coreference resolution module for Polish. In Proceedings of the 8th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2011) (pp. 191-200).

[132] Opsahl, T., Agneessens, F., & Skvoretz, J. (2010). Node centrality in weighted networks: Generalizing degree and shortest paths. Social networks, 32(3), 245-251.

[133] Pantoja, Cesar and Fernandez Arguedas, Virginia and Izquierdo, Ebroul “Anonymization and De-identification of Personal Surveillance Visual Information: A Review” Proceedings of the 5th Latin-American Conference on Networked and Electronic Media (LACNEM 2013). 2013, p.1--6.

[134] Pei, S., Muchnik, L., Andrade Jr, J. S., Zheng, Z., & Makse, H. A. (2014). Searching for superspreaders of information in real-world social media. Scientific reports, 4, 5547.

[135] Peng H, Khashabi D, Roth D (2015) Solving hard coreference problems. In: Proceedings of the 2015

[136] Pennekamp, J., Henze, M., Hohlfeld, O., & Panchenko, A. (2019, May). Hi Doppelgänger: Towards Detecting Manipulation in News Comments. In Companion Proceedings of The 2019 World Wide Web Conference (pp. 197-205).

[137] Pennington J, Socher R, Manning C (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543.

[138] Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Proc. of NAACL.

[139] Petkos, G. et al. (2017). Graph-based multimodal clustering for social multimedia. Multimedia Tools & Applications, 76(6):7897–7919.

[140] Pio, G. et al. (2018). Multi-type clustering and classification from heterogeneous networks. Information Sciences, 425, 107-126.

[141] Popescu, I. I. (2009). Word frequency studies (Vol. 64). Walter de Gruyter.

[142] Pradhan S, Ramshaw L, Marcus M, Palmer M, Weischedel R, Xue N (2011) Conll-2011 shared task:

[143] Qian, X., Hua, X.-S., Chen, P., & Ke, L. (2011). PLBP: An effective local binary patterns texture descriptor with pyramid representation. Pattern Recognition, 44(10), 2502-2515.

[144] R. Girshick, "Fast r-cnn," arXiv preprint arXiv:1504.08083, 2015.

[145] Raghunathan K, Lee H, Rangarajan S, Chambers N, Surdeanu M, Jurafsky D, Manning C (2010) A multi-pass sieve for coreference resolution. In: Proceedings of 2010 Conference on Empirical Methods in NAturalLanguage Processing, Association for Computational Linguistics, pp 492–501.

[146] Rahman A, Ng V (2011) Coreference resolution with world knowledge. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, pp 814–824.

[147] Raptis, M.; Sigal, L. Poselet key-framing: A model for human activity recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2650–2657

[148] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, (pp. 91-99).

[149] Riederer, C., Kim, Y., Chaintreau, A., Korula, N., & Lattanzi, S. (2016, April). Linking users across domains with location data: Theory and validation. In Proceedings of the 25th International Conference on World Wide Web (pp. 707-719).

[150] Ristani, E., Solera, F., Zou, R., Cucchiara, R., & Tomasi, C. (2016, October). Performance measures and a data set for multi-target, multi-camera tracking. In European Conference on Computer Vision (pp. 17-35). Springer, Cham.

[151] Rodríguez-Moreno, I., Martínez-Otzeta, J. M., Sierra, B., Rodriguez, I., & Jauregi, E. (2019). Video Activity Recognition: State-of-the-Art. Sensors, 19(14), 3160.

[152] Sand, P.; Teller, S. Particle video: Long-range motion estimation using point trajectories. Int. J. Comput. Vis. 2008, 80, 72. [CrossRef]

[153] Sara R. Jordan (2014) Research integrity, image manipulation, and anonymizing photographs in visual social science research, International Journal of Social Research Methodology, 17:4, 441-454, DOI: 10.1080/13645579.2012.759333

[154] Shan, C., Gong, S., & McOwan, P. W. (2009). Facial expression recognition based on Local Binary Patterns: A comprehensive study. Image and Vision Computing, 27(6), 803-816.

[155] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In Advances in neural information processing systems (pp. 568-576).

[156] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556. Retrieved from http://arxiv.org/abs/1409.1556

[157] Soomro, K.; Zamir, A.R.; Shah, M. UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv 2012, arXiv:1212.0402

[158] Soon WM, Ng HT, Lim DCY (2001) A machine learning approach to coreference resolution of noun phrases. Computational linguistics 27(4):521–544.

[159] Stylianou, N., & Vlahavas, I. (2019). A Neural Entity Coreference Resolution Review. arXiv preprint arXiv:1910.09329.

[160] Sukthanker, R., Poria, S., Cambria, E., & Thirunavukarasu, R. (2020). Anaphora and coreference resolution: A review. Information Fusion.

[161] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

[162] Tang, L., Wang, X., & Liu, H. (2012). Community detection via heterogeneous interaction analysis. Data mining and knowledge discovery, 25(1), 1-33.

[163] Těšitelová, M. (1992). Quantitative linguistics. John Benjamins.

[164] Thompson, R. (2011). Radicalization and the use of social media. Journal of strategic security, 4(4), 167-190.

[165] Toldova, S., Azerkovich, I., Ladygina, A., Roitberg, A., & Vasilyeva, M. (2016, June). Error analysis for anaphora resolution in Russian: new challenging issues for anaphora resolution task in a morphologically rich language. In Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016) (pp. 74-83).

[166] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 6450-6459).

[167] Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.

[168] Tsikerdekis, M., & Zeadally, S. (2014). Multiple account identity deception detection in social media using nonverbal behavior. IEEE Transactions on Information Forensics and Security, 9(8), 1311-1321.

[169] Ullah, A.; Ahmad, J.; Muhammad, K.; Sajjad, M.; Baik, S.W. Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features. IEEE Access 2018, 6, 1155–1166. [CrossRef]

[170] Uryupina O, Poesio M, Giuliano C, Tymoshenko K (2012) Disambiguation and filtering methods in using web knowledge for coreference resolution. In: Cross-Disciplinary Advances in Applied Natural

[171] Vajjala, S. (2015). Analyzing text complexity and text simplification: connecting linguistics, processing and educational applications (Doctoral dissertation, Ph. D. thesis, University of Tübingen).

[172] Vala H, Piper A, Ruths D (2016) The more antecedents, the merrier: Resolving multi-antecedent anaphors. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol 1, pp 2287–2296.

[173] Wang, L., & He, D.-C. (1990). Texture classification using texture spectrum. Pattern Recognition, 23(8), 905-910.

[174] Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7794-7803).

[175] Wang, Y.; Sun, S.; Ding, X. A self-adaptive weighted affinity propagation clustering for key frames extraction on human action recognition. J. Vis. Commun. Image Represent. 2015, 33, 193–202. [CrossRef]

[176] Wiseman S, Rush AM, Shieber S, Weston J (2015) Learning anaphoricity and antecedent ranking features for coreference resolution. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol 1, pp 1416–1426.

[177] Wiseman S, Rush AM, Shieber SM (2016) Learning global features for coreference resolution. arXiv preprint arXiv:160403035.

[178] Yang X, Zhou G, Su J, Tan CL (2003) Coreference resolution using competition learning approach. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1, Association for Computational Linguistics, pp 176–183.

[179] Yang X, Zhou G, Su J, Tan CL (2004) Improving noun phrase coreference resolution by matching strings. In: International Conference on Natural Language Processing, Springer, pp 22–31.

[180] Zeldes A, Zhang S (2016) When annotation schemes change rules help: A configurable approach to coreference resolution beyond ontonotes. In: Proceedings of the Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2016), pp 92–101.

[181] Zhang, Z. Microsoft kinect sensor and its effect. IEEE Multimed. 2012, 19, 4–10

[182] Zhao, G., & Pietikainen, M. (2006). Local Binary Pattern Descriptors for Dynamic Texture Recognition. 18th International Conference on Pattern Recognition (ICPR'06), 2, pp. 211-214.

[183] Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K. W. (2018). Gender bias in coreference resolution: Evaluation and debiasing methods. arXiv preprint arXiv:1804.06876.

[184] Zhaoyun, D., Yan, J., Bin, Z., & Yi, H. (2013). Mining topical influencers based on the multi-relational network in micro-blogging sites. China Communications, 10(1), 93-104.

[185] Zheng, L., Bie, Z., Sun, Y., Wang, J., Su, C., Wang, S., & Tian, Q. (2016, October). Mars: A video benchmark for large-scale person re-identification. In European Conference on Computer Vision (pp. 868-884). Springer, Cham.

[186] Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., & Tian, Q. (2015). Scalable person re-identification: A benchmark. In Proceedings of the IEEE international conference on computer vision (pp. 1116-1124).

[187] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. Advances in neural information processing systems, (pp. 487-495).

[188] Žitkus, V., Butkienė, R., Butleris, R., Maskeliūnas, R., Damaševičius, R., & Woźniak, M. (2019). Minimalistic Approach to Coreference Resolution in Lithuanian Medical Records. Computational and mathematical methods in medicine, 2019.

[189] Žitkus, Voldemaras, 2018, Lithuanian Coreference Corpus, CLARIN-LT digital library in the Republic of Lithuania, http://hdl.handle.net/20.500.11821/19.

[190] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd (Vol. 96, No. 34, pp. 226-231).

[191] Oliva, A., & Torralba, A. (2001, May). Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. International Journal of Computer Vision, 42(3), 145-175.

[192] Doretto, G., Chiuso, A., Wu, Y. N., & Soatto, S. (2003, Feb). Dynamic Textures. International Journal of Computer Vision, 51(2), 91-109.

[193] Kučová, L., & Žabokrtský, Z. (2005, September). Anaphora in Czech: Large data and experiments with automatic anaphora resolution. In International Conference on Text, Speech and Dialogue (pp. 93-98). Springer, Berlin, Heidelberg.

[194] Pons, P., & Latapy, M. (2005, October). Computing communities in large networks using random walks. In International symposium on computer and information sciences (pp. 284-293). Springer, Berlin, Heidelberg.

[195] Ahonen, T., Hadid, A., & Pietikainen, M. (2006, Dec). Face Description with Local Binary Patterns: Application to Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(12), 2037-2041.

[196] Zhao, Q., Mitra, P., & Chen, B. (2007, July). Temporal and information flow based event detection from social text streams. In AAAI (Vol. 7, pp. 1501-1506).

[197] Xu, X., Yuruk, N., Feng, Z., & Schweiger, T. A. (2007, August). Scan: a structural clustering algorithm for networks. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 824-833).

[198] Chan, A. B., & Vasconcelos, N. (2008, May). Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(5), 909-926.

[199] Nguy, G. L., Novák, V., & Žabokrtský, Z. (2009, September). Comparison of Classification and Ranking Approaches to Pronominal Anaphora Resolution in Czech. In Proceedings of the SIGDIAL 2009 Conference (pp. 276-285).

[200] Shroff, N., Turaga, P., & Chellappa, R. (2010, June). Moving vistas: Exploiting motion for describing scenes. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (pp. 1911-1918).

[201] Cichowski, J., & Czyzewski, A. (2011, November). Reversible video stream anonymization for video surveillance systems based on pixels relocation and watermarking. In 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops) (pp. 1971-1977). IEEE.

[202] Zhao, G., Ahonen, T., Matas, J., & Pietikainen, M. (2012, April). Rotation-Invariant Image and Video Description With Local Binary Pattern Features. IEEE Transactions on Image Processing, 21(4), 1465-1477.

[203] Kopec, M., & Ogrodniczuk, M. (2012, May). Creating a Coreference Resolution System for Polish. In LREC (pp. 192-195).

[204] Morrison, D., McLoughlin, I., Hogan, A., & Hayes, C. (2012, May). Evolutionary clustering and analysis of user behaviour in online forums. In Sixth International AAAI Conference on Weblogs and Social Media.

[205] Jabeur, L. B., Tamine, L., & Boughanem, M. (2012, October). Active microbloggers: identifying influencers, leaders and discussers in microblogging networks. In International Symposium on String Processing and Information Retrieval (pp. 111-117). Springer, Berlin, Heidelberg.

[206] Yang, X., Zhang, C., & Tian, Y. (2012, October). Recognizing actions using depth motion maps-based histograms of oriented gradients. In Proceedings of the 20th ACM international conference on Multimedia (pp. 1057-1060).

[207] Solorio, T., Hasan, R., & Mizan, M. (2013, June). A case study of sockpuppet detection in wikipedia. In Proceedings of the Workshop on Language Analysis in Social Media (pp. 59-68).

[208] Zafarani, R., & Liu, H. (2013, August). Connecting users across social media sites: a behavioral-modeling approach. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 41-49).

[209] Feichtenhofer, C., Pinz, A., & Wildes, R. P. (2014, June). Bags of Spacetime Energies for Dynamic Scene Recognition. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[210] Liu, S., Wang, S., Zhu, F., Zhang, J., & Krishnan, R. (2014, June). Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In Proceedings of the 2014 ACM SIGMOD international conference on Management of data (pp. 51-62).

[211] Znotiņš, A. (2014, September). Coreference Resolution in Latvian. In Human Language Technologies-The Baltic Perspective: Proceedings of the Sixth International Conference Baltic HLT 2014 (Vol. 268, p. 153). IOS Press.

[212] Deng, Y., Luo, P., Loy, C. C., & Tang, X. (2014, November). Pedestrian attribute recognition at far distance. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 789-792).

[213] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[214] Li, D., Chen, X., & Huang, K. (2015, November). Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios. In 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR) (pp. 111-115). IEEE.

[215] Liu, L., Cheung, W. K., Li, X., & Liao, L. (2016, July). Aligning Users across Social Networks Using Network Embedding. In Ijcai (pp. 1774-1780).

[216] Novák, M., Nedoluzhko, A., & Žabokrtský, Z. (2017, April). Projection-based coreference resolution using deep syntax. In Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017) (pp. 56-64).

[217] Jiang, L., Li, H., Wang, L., & Wu, J. (2017, June). Finding overlapping communities based on information fusion in social network. In 2017 International Conference on Service Systems and Service Management (pp. 1-6). IEEE.

[218] Sharma, J., Granmo, O. C., Goodwin, M., & Fidje, J. T. (2017, August). Deep convolutional neural networks for fire detection in images. In International Conference on Engineering Applications of Neural Networks (pp. 183-193). Springer, Cham.

[219] Wojke, N., Bewley, A., & Paulus, D. (2017, September). Simple online and realtime tracking with a deep association metric. In 2017 IEEE international conference on image processing (ICIP) (pp. 3645-3649). IEEE.

[220] Khadzhiiskaia, A., & Sysoev, A. (2017, November). Coreference resolution for Russian: taking stock and moving forward. In 2017 Ivannikov ISPRAS Open Conference (ISPRAS) (pp. 70-75). IEEE.

[221] Nitoń, B., Morawiecki, P., & Ogrodniczuk, M. (2018, May). Deep neural networks for coreference resolution for Polish. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).

[222] Agarwal, O., Subramanian, S., Nenkova, A., & Roth, D. (2019, June). Evaluation of named entity coreference. In Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference (pp. 1-7).

[223] Chen, Z., Li, A., & Wang, Y. (2019, November). A temporal attentive approach for video-based pedestrian attribute recognition. In Chinese Conference on Pattern Recognition and Computer Vision (PRCV) (pp. 209-220). Springer, Cham.

[224] Tieleman T, Hinton G (2012) Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning 4(2):26–31.

[225] Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence, (12), 2037–2041.

[226] Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., & Kumar, N. (2013). Localizing parts of faces using a consensus of exemplars. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(12), 2930–2940.

[227] Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018). Vggface2: A dataset for recognising faces across pose and age. 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), 67–74.

[228] Cao, Z., Yin, Q., Tang, X., & Sun, J. (2010). Face recognition with learning-based descriptor. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2707–2714.

[229] Chen, D., Cao, X., Wen, F., & Sun, J. (2013). Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3025–3032.

[230] Chopra, S., Hadsell, R., LeCun, Y., & others. (2005). Learning a similarity metric discriminatively, with application to face verification. CVPR (1), 539–546.

[231] Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4690–4699.

[232] Hu, P., & Ramanan, D. (2017). Finding tiny faces. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 951–959.

[233] Huang, C., Zhu, S., & Yu, K. (2012). Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval. ArXiv Preprint ArXiv:1212.6094.

[234] Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008). Labeled faces in the wild: A database forstudying face recognition in unconstrained environments.

[235] Jiang, H., & Learned-Miller, E. (2017). Face detection with the faster R-CNN. Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference On, 650–657.

[236] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., … others. (2018). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. ArXiv Preprint ArXiv:1811.00982.

[237] Li, J., Wang, T., & Zhang, Y. (2011). Face detection using surf cascade. Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference On, 2183–2190.

[238] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 212–220.

[239] Najibi, M., Samangouei, P., Chellappa, R., & Davis, L. S. (2017). SSH: Single Stage Headless Face Detector. ICCV, 4885–4894.

[240] Ranjan, R., Patel, V. M., & Chellappa, R. (2017). Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[241] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 815–823.

[242] Shu, C., Ding, X., & Fang, C. (2011). Histogram of the oriented gradient for face recognition. Tsinghua Science and Technology, 16(2), 216–224.

[243] Simonyan, K., Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2013). Fisher vector faces in the wild. BMVC, 2(3), 4.

[244] Sun, X., Wu, P., & Hoi, S. C. H. (2018). Face detection using deep learning: An improved faster RCNN approach. Neurocomputing, 299, 42–50.

[245] Sun, Y., Wang, X., & Tang, X. (2013). Deep convolutional network cascade for facial point detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3476–3483.

[246] Sun, Y., Wang, X., & Tang, X. (2015). Deeply learned face representations are sparse, selective, and robust. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2892–2900.

[247] Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1701–1708.

[248] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference On, 1, I--I.

[249] Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., … Liu, W. (2018). Cosface: Large margin cosine loss for deep face recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5265–5274.

[250] Wolf, L., Hassner, T., & Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference On, 529–534.

[251] Yang, S., Luo, P., Loy, C.-C., & Tang, X. (2015). From facial parts responses to face detection: A deep learning approach. Proceedings of the IEEE International Conference on Computer Vision, 3676–3684.

[252] Zhang, J., Wu, X., Hoi, S. C. H., & Zhu, J. (2020). Feature agglomeration networks for single stage face detection. Neurocomputing, 380, 180–189.

[253] Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters, 23(10), 1499–1503.

[254] Zhu, X., & Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference On, 2879–2886.

[255] Zhu, Z., Luo, P., Wang, X., & Tang, X. (2014). Recover canonical-view faces in the wild with deep neural networks. ArXiv Preprint ArXiv:1404.3543.

Appendix A

A.1 Deep Linguistic Features

Table 16. Linguistic features

No	Feature (indicator)	Explanation	Type
1	Number of sentences in a text [110]		Descriptive
2	Number of words in a text [110]		Descriptive
3	Standard deviation of the mean length of sentences [110]	A large standard deviation indicates that the text has large variation in terms of the lengths of its sentences.	Descriptive
4	Mean number of syllables (length) in words [110]	Shorter words are easier to read and the estimate of word length serves as a common proxy for word frequency.	Descriptive
5	Standard deviation of the mean number of syllables in words [110]	A large standard deviation indicates that the text has large variation in terms of the lengths of its words, such that it may have both short and long words.	Descriptive
6	Mean number of letters (length) in words [110]	Longer words tend to be lower in frequency or familiarity to a reader.	Descriptive
7	Standard deviation of the mean number of letter in words [110]	A large standard deviation indicates that the text has large variation in terms of the lengths of its words.	Descriptive
8	TTR (Type-Token Ratio) [35]	Ratio of distinct words and all words in the text.	Frequency structure indicator / lexical diversity
9	h-Point [141]	Fuzzy border between high frequency and lower frequency words.	Frequency structure indicator / lexical diversity
10	Entropy (H) [82]	In linguistics, entropy expresses the degree	Frequency structure indicator / lexical diversity
11	Token Length Frequency Spectrum [82]	List of all token lengths in a text with their frequency.	Frequency structure indicator
12	Λ (Lambda) [82]	Describes frequency structure of text, i.e. it is related to vocabulary richness, but also considers the relationship between neighbouring frequencies.	Frequency structure indicator
13	Adjusted Modulus (A) [82]	Frequency structure indicator, independent of text length.	Frequency structure indicator
14	Curve Length (L) [82]	As a lot of vocabulary richness measures are based on the curve of rank-frequency distribution, L is defined as the sum of the Euclidean distances between all neighbouring points on the curve.	Frequency structure indicator
15	a [141]	Text lenght independent variation of h-Point (see above).	Frequency structure indicator
16	R1 [82]	Vocabulary richness indicator (focus on lower frequency or lexical/content words).	Lexical diversity
17	RR (Repeat Rate) [82]	Shows the degree of vocabulary concentration in a text, i.e. inverse measure of vocabulary richness.	Lexical diversity
18	RRmc [82]	Relative RR for better comparison with the other indices.	Lexical diversity
19	Gini coefficient [82]	In linguistics G is used as a measure for vocabulary richness.	Lexical diversity
20	R4 [82]	The reversed Gini coefficient.	Lexical diversity
21	Hapax percentage (HL) [82]	Ratio between the number of hapax legomena, i.e. words that occur only once, in a text, and number of all words.	Lexical diversity

No	Feature (indicator)	Explanation	Type
22	Curve Length R Index [82]	Indicator of vocabulary richness derived from the curve length (L) (see above).	Lexical diversity
23	MATTR (Moving Average Type-Token Ratio) [81]	Calculates TTRs for a moving window of tokens from the first to the last token, computing a TTR for each window. The MATTR is the mean of the TTRs of each window; text length independent.	Lexical diversity
24	MWTTR (Moving Window Type-Token Ratio) [81]	Defined as the series of Vi (or by another words, each Vi is mapped to its i).	Lexical diversity
25	MWTTRD (Moving Window Type-Token Ratio Distribution) [81]	The distribution of MWTTR values.	Lexical diversity
26	Yule's K [11]	Measure of vocabulary repetitiveness.	Lexical diversity
27	Guirad's R [163]	Measure of vocabulary richness, i.e. relation between the length of a text N and its vocabulary V.	Lexical diversity
28	Herdan's C [11]	LogTTR	Lexical diversity
29	Guiraud's Root TTR (R) [11]	Simple try to lessen effect of TTR dependence on text length.	Lexical diversity
30	Carroll's Corrected TTR (CTTR) [11]	Simple try to lessen effect of TTR dependence on text length.	Lexical diversity
31	Dugast's Uber Index (U) [11]	Simple try to lessen effect of TTR dependence on text length.	Lexical diversity
32	Summer's index (S) [11]	Simple try to lessen effect of TTR dependence on text length.	Lexical diversity
33	Yule's I (I) [11]	Inverse Yule's K.	Lexical diversity
34	Simpson's D [11]	Measure of vocabulary repetitiveness.	Lexical diversity
35	Herdan's Vm [11]	Measure of vocabulary repetitiveness.	Lexical diversity
36	Maas' index a2 [11]	Measure of relative vocabulary growth while the text progresses.	Lexical diversity
37	Maas' index logV0 [11]	Measure of relative vocabulary growth while the text progresses.	Lexical diversity
38	MSTTR (Mean Segmental Type-Token Ratio) [11]	It splits the tokens (words) into segments of the given size, TTR for each segment is calculated and the mean of these values returned; independent of text length.	Lexical diversity
39	Number of Different Words (NDW) [102]		Lexical diversity
40	NDW (first 50 words) [102]		Lexical diversity
41	Writer's View [82]	Indicator that is defined by the angle between the h-Point (see above) and the ends of the rank-frequency distribution, i.e. the golden ratio.	Other
42	Average Token (Word) Length (ATL) [82]	Simple readability measure.	Readability
43	Automated Readability Index (ARI) [11]	Based on average sentence length and average word length.	Readability
44	ARI.Simple [11]	Based on average sentence length and average word length.	Readability
45	Coleman's (1971) Readability Formula 1. [11]	Based on number of 1-syllable words in 100 words.	Readability
46	Coleman's (1971) Readability Formula 2. [11]	Based on number of 1-syllable words in 100 words.	Readability
47	Coleman-Liau Estimated Cloze Percent (ECP) (Coleman and Liau 1975) [11]	Based on average word length, number of words and number of sentences in a text.	Readability
48	Coleman-Liau Grade Level (Coleman and Liau 1975) [11]	Based on Coleman-Liau Estimated Cloze Percent (ECP) (see above).	Readability
49	Coleman-Liau Index (Coleman and Liau 1975) [11]	Relies on characters instead of syllables per word.	Readability
50	Dickes-Steiwer Index (Dicks and Steiwer 1977) [11]	Based on average word length, average sentence length and TTR (see above).	Readability
51	Easy Listening Formula (Fang 1966) (ELF) [11]	Ratio of number of words with 2 syllables or more and number of sentences ina text.	Readability

No	Feature (indicator)	Explanation	Type
52	Farr-Jenkins-Paterson's Simplification of Flesch's Reading Ease Score(Farr, Jenkins and Paterson 1951) [11]	Based on number of one-syllable words per 100 words and average sentence length in words.	Readability
53	Flesch's Reading Ease Score (Flesch 1948) [11]	Based on average sentence length and ratio of number of syllables and number of words.	Readability
54	The Powers-Sumner-Kearl's Variation of Flesch Reading Ease Score (Powers, Sumner and Kearl, 1958) [11]	It calculate the US grade level of a text sample based on sentence length and number of syllables.	Readability
55	Flesch-Kincaid Readability Score (Flesch and Kincaid 1975) [11]	Based on number of words per sentences and syllables per words in a text.	Readability
56	Gunning's Fog Index (Gunning 1952) [11]	Readability evaluated using the mean sentence length and a hard-word (>3 syllables) factor.	Readability
57	The Navy's Adaptation of Gunning's Fog Index (Kincaid, Fishburne, Rogers and Chissom 1975) [11]	Based on the number of words with less than 3 syllables and the number of 3-syllable words.	Readability
58	FORCAST (Simplified Version of FORCAST.RGL) (Caylor and Sticht 1973) [11]	Based on number of single-syllable words in a 150-word sample.	Readability
59	FORCAST.RGL (Caylor and Sticht 1973) [11]	Based on number of single-syllable words in a 150-word sample.	Readability
60	Fucks' (1955) Stilcharakteristik (Style Characteristic) [11]	Based on average word length and average sentence length.	Readability
61	Linsear Write (Klare 1975) [11]	Specifically designed to calculate the US grade level of a text sample based on sentence length and the number of words used that have three or more syllables	Readability
62	Neue Wiener Sachtextformeln 1 (Bamberger and Vanecek 1984) [11]	Based on these characteristics of the text: number of words with 3 syllables or more, average sentence length, number of words with 6 characters or more and number of 1-syllable words.	Readability
63	Neue Wiener Sachtextformeln 2 (Bamberger and Vanecek 1984) [11]	Based on these characteristics of the text: number of words with 3 syllables or more, average sentence length, number of words with 6 characters or more.	Readability
64	Neue Wiener Sachtextformeln 3 (Bamberger and Vanecek 1984) [11]	Based on these characteristics of the text: number of words with 3 syllables or more and average sentence length.	Readability
65	Neue Wiener Sachtextformeln 4 (Bamberger and Vanecek 1984) [11]	Based on these characteristics of the text: number of words with 3 syllables or more and average sentence length.	Readability
66	Anderson's (1983) Readability Index [11]	Ratio of words with 7 syllables or more and number of sentences in the text.	Readability
67	Simple Measure of Gobbledygook (SMOG) (McLaughlin 1969) [11]	Based on the number of sentences of the text and the number of words with three or more syllables.	Readability
68	SMOG (Regression Equation C) (McLaughlin's 1969) [11]	Based on the number of sentences of the text and the number of words with three or more syllables.	Readability
69	Simplified Version of McLaughlin's (1969) SMOG Measure [11]	Based on the number of sentences of the text and the number of words with three or more syllables.	Readability
70	Adaptation of McLaughlin's (1969) SMOG Measure for German Texts [11]	Based on the number of sentences of the text and the number of words with three or more syllables.	Readability
71	Strain Index (Solomon 2006) [11]	Based on number of syllables and number of sentences on the text.	Readability
72	Wheeler & Smith's (1954) Readability Measure [11]	Based on average sentence length and the number of words with 2 syllables or more.	Readability
73	Average sentence length [171]	Simple measure of syntactic complexity of the text.	Syntactic complexity
74	Ratio of Commas and sentences [171]	Simple measure of syntactic complexity of the text.	Syntactic complexity