UNIVERSITÀ DEGLI STUDI DI MILANO

FACOLTÀ DI SCIENZE POLITICHE,
ECONOMICHE E SOCIALI

Master's Degree of Data Science and Economics

Thesis

Face Recognition models implementation and impacts on individual's privacy

Advisor
Ch. Prof.ssa Silvia Salini

Graduating student
Anna Errichiello

Matriculation number
943897

Academic year
2020 / 2021

Contents

Introduction 4

1. Fields of implementation of Face Recognition 5

1.1 Application of face recognition technologies 5

1.2 Process of Face Recognition Technology 7

1.3 Possible attacks 8

2. Face Recognition Technology and General Data Protection Regulation 10

2.1 Rules for Face Recognition Technology 10

2.2 Main Principles related to the processing of personal data 12

3. Rights of the data subjects 16

3.1 Surveillance Systems 16

3.2 Commercial Applications and Rights of the data subjects 17

3.3 Legal Grounds for Face Recognition Technologies 19

4. Data Security 22

4.1 Prevention and Mitigation measures 22

4.2 Codes of conduct and certifications 24

4.3 Face Recognition Technologies vulnerabilities 25

4.4 Pseudonymization of face images 26

5. Face Verification 29

5.1 Face images used for Face Verification 29

5.2 Preparation of data 30

5.3 Euclidean Distance and Mean Squared Error 32

6. Face Identification 33

6.1 Methodology used for Face Identification 33

6.2 Face Detection 33

6.3 Face Identification 35

7. Face clustering and Euclidean distance 38

7.1 Clustering method proposed....38
7.2 Hierarchical clustering....39
7.3 Hierarchical clustering on the data....40
7.4 K-means clustering....40
7.5 Results over the K-means clustering....41
7.6 Euclidean distance....43
7.7 Binary Logistic Regression and Euclidean distances....44
8. Face Clustering with Mahalanobis distance....48
8.1 Mahalanobis distance and outliers' detection....48
8.2 Impact of Outliers....49
8.3 Binary Logistic Regression for Mahalanobis Distance....50
Concluding Remarks....54
Appendix....56
Face verification with Euclidean Distance....56
Face verification with Mean Squared Error....57
Data preparation for Face Identification and Face Clustering....58
Face Detection....59
Face Identification....60
Face Clustering and Euclidean distance....63
Hierarchical Clustering....64
K-means Clustering....65
Euclidean distances....68
Mahalanobis distance....73
Bibliography....79

Introduction

Face recognition is one of the most challenging technologies and is affecting our daily routines, being implemented in many areas of application. The main task of these technologies is to match a face to a person's identity.

The objective of the study is to analyse the juridical and legal aspects that regulate these technologies, since their use implies the acquisition of sensitive data, identified as biometric data. The regulatory environment is mainly described through the normative reported in the GDPR (General Data Protection Regulation). The images captured through Face Recognition Technologies are subject to such regulations in order to protect the data subjects themselves, however another fundamental role of such technologies, as in case of video surveillance, implies the protection of public interest to overcome the interest of the single individual.

Following the delineation of the normative framework, the main passes of Face Recognition are put in practice by implementing different methods and models for Face Verification and Face Identification. Face Verification is a one-to-one approach used to compare one target image with another one, whereas Face Identification is a one-to-many approach, through which a target image is associated to multiple images that ideally show the same person. Furthermore, the study proposes different methods to group images inferred as showing the same person, without starting from a single target face image.

The methods used for face clustering are:

- Hierarchical clustering and K-means clustering,
- Euclidean distance,
- Mahalanobis distance.

Finally, the study proposes a comparison between the two distance methods used.

1. Fields of implementation of Face Recognition

Abstract

Face Recognition is a very challenging technology, and it is widely used in many fields. This Chapter introduce the topic of the study by delineating the contexts of application and implementation.

1.1 Application of face recognition technologies

There are many fields in which Face Recognition has become vital in order to identify people for different reasons: from marketing purposes to the prevention of violence and recognition of criminals through video surveillance. Each of these applications have the same purpose: give an identity to the person or more specifically associate a face to many other similar faces and, thanks to the availability of identification databases, such as the ANPR (Anagrafe Nazionale Popolazione Residente) in Italy that has digitalized many specific information about each citizen. These information allow the identification of a person. Some of the fields of the Face Recognition Technology are reported in the following table.

Security	Investigations Licensed drivers' managing	HCI
• Access control to buildings • Airports/seaports • ATM machines • Border checkpoints • Computer/network security • Smart card • Video indexing • Surveillance • Labelling faces in video • Forensics • Criminal justice systems • Mug shot/booking systems • Post-event analysis • Image database	• Benefit recipients • Missing children • Immigrants and police bookings • Witness face reconstruction • General identity verification • Electoral registration • Banking • Electronic commerce • Identifying newborns • National IDs • Passports • Drivers' licenses	• Ubiquitous aware • Behaviour monitoring • Customer assessing

Table 1.1

The three areas of applications reported in the table are: security, investigation and HCI (Human-Computer Interaction).

One of the most common security applications is accessible from every smartphone, where it is possible to unlock the device just by placing the smartphone owner's face in front of the camera.

Face recognition is used at the entrance of buildings for access control, such as offices. This technology can be also used in combination with others.

Another interesting application is the temperature scanners that can be found at the entrance of hospitals or surgeries, especially during the period of pandemic. These devices can potentially capture individuals' images and store it.

The use of passports and identity cards in case of airport security checks implies image-based Face Recognition, since it is checked whether the person on the identity card is the same person that present the document.

Another important application is biometric authentication: Face Recognition, in fact, has many advantages compared to other biometric identification procedures that imply the use of fingerprints or iris.

Face Recognition Technologies are consistently used in investigations, such as in the research of criminals, in case of missing children or simply when a general identity verification takes place.

Another important use of Face Recognition is surveillance, which is widely used, especially to protect the public interest.

Face recognition Technology works with different type of inputs:

- image-based Face Recognition,
- video-based Face Recognition,
- 3D-based Face Recognition.

Images and videos can be captured and processed in order detect a human face and give identity to that face: however, there are different type of systems used to solve this kind of issues. In fact, there are many technologies using bidimensional Face Recognition Systems, but also others that try to reach a better accuracy by exploiting the three-dimensional geometry of human faces.

1.2 Process of Face Recognition Technology

Face Recognition Technology is characterized by two main components: Face Detection and Face Recognition. The former consists in detecting the presence of a human face in the input transmitted to the Face Recognition Technology, the latter consists in giving an identity to the face detected in the input. There is another distinction to consider, already mentioned in the Introduction, which is given by two different terms: Face Verification and Face Identification. Face Verification implies the analysis of two images and the understanding of whether the two faces belong to the same person or not, which means it consists in a one-

to-one comparison. However, Face Identification implies a one-to-many recognition process.

In general, Face Recognition Processes are marked by the following steps:

- data acquisition,
- pre-processing: it consists in elaborating the data,
- feature extraction,
- Face Verification or Identification.

Data acquisition consists in capturing the data, which can be images or videos. These data are then subject to pre-processing, that in case of Face Recognition, consists in detecting human faces in the image. The feature extraction is a fundamental step performed to obtain specific information about the human face found in the image. The features can be described as a summary of the main information of the human face detected. The last step of Face Verification or Identification is performed by using many algorithms that can potentially identify whether two faces belong to the same person or whether in a database of a great number of images, there are other images that represent the same person shown in the target image. In some identification procedures, only those images that are inferred as similar are used to effectively identify an individual.

1.3 Possible attacks

The fields of implementation of Face Recognition Technologies reported in the table above may possibly incur in the risk of attacks. An article published in the Forbes reports the main industries prone to attacks:

- Surveillance: the main goal is to identify offenders or criminals in the city for instance,
- Banking: the ATMs associate the card to the owner's face and the system can help in preventing frauds and securing transactions,
- Internet and Search Engines,
- Self-Payment Systems: for instance, the FamilyMart supermarket, originally located in Shanghai and then spread all around the world, has introduced the use of Face Recognition Technologies while paying without the assistance of cashiers,

- Airports and Custom Offices,
- Smart Homes: Face Recognition can be used for access control, to check whether the person corresponds to one of the people registered in the database.

2. Face Recognition Technology and General Data Protection Regulation

Abstract

Face Recognition Technologies are subject to laws and regulations since they work with sensitive information. The improper disclosure of sensitive information violates individual's rights and privacy. This Chapter briefly describes the conditions and principles that influence the treatment of such sensitive personal data.

2.1 Rules for Face Recognition Technology

Face Recognition Technology is a controversial technology and there are both positive and negative aspects to consider. In many cases it allows people to apply security blocks, preventing from different types of attacks. On the other side, it involves and affects people's privacy.

Face Recognition Technology implies the acquisition of images, videos or other types of inputs used to identify the person in the image. This technology has many applications mentioned in Chapter 1: the main ones are security and monitoring.

More specifically, Face Recognition Technology is a tool that allows the capturing of biometric images and its main task is to give an identity to such person or find other images or other types of inputs to facilitate the person's identification.

The use of Face Recognition Technology is not explicitly mentioned within the GDPR, however these technologies are subject to its rules, since they collect and process personal biometric data, that are indicated in the GDPR as one of the sensitive categories. Article 9 of the GDPR states that usually the processing of sensitive data is not allowed, except for specific situations:

1. *Processing of personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person's sex life or sexual orientation shall be prohibited.*
2. *Paragraph 1 shall not apply if one of the following applies:*

a. *the data subject has given explicit consent to the processing of those personal data for one or more specified purposes, except where Union or Member State law provide that the prohibition referred to in paragraph 1 may not be lifted by the data subject,*
b. *processing is necessary for the purposes of carrying out the obligations and exercising specific rights of the controller or of the data subject in the field of employment and social security and social protection law in so far as it is authorised by Union or Member State law or a collective agreement pursuant to Member State law providing for appropriate safeguards for the fundamental rights and the interests of the data subject,*
c. *processing is necessary to protect the vital interests of the data subject or of another natural person where the data subject is physically or legally incapable of giving consent,*
d. *processing is carried out in the course of its legitimate activities with appropriate safeguards by a foundation, association or any other not-for-profit body with a political, philosophical, religious or trade union aim and on condition that the processing relates solely to the members or to former members of the body or to persons who have regular contact with it in connection with its purposes and that the personal data are not disclosed outside that body without the consent of the data subjects,*
e. *processing relates to personal data which are manifestly made public by the data subject,*
f. *processing is necessary for the establishment, exercise or defence of legal claims or whenever courts are acting in their judicial capacity,*
g. *processing is necessary for reasons of substantial public interest, on the basis of Union or Member State law which shall be proportionate to the aim pursued, respect the essence of the right to data protection and provide for suitable and specific measures to safeguard the fundamental rights and the interests of the data subject,*
h. *processing is necessary for the purposes of preventive or occupational medicine, for the assessment of the working capacity of the employee, medical diagnosis, the provision of health or social care or treatment or the management of health or social care systems and services on the basis of*

Union or Member State law or pursuant to contract with a health professional and subject to the conditions and safeguards referred to in paragraph 3;

i. *processing is necessary for reasons of public interest in the area of public health, such as protecting against serious cross-border threats to health or ensuring high standards of quality and safety of health care and of medicinal products or medical devices, on the basis of Union or Member State law which provides for suitable and specific measures to safeguard the rights and freedoms of the data subject, in particular professional secrecy,*

j. *processing is necessary for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes in accordance with Article 89(1) based on Union or Member State law which shall be proportionate to the aim pursued, respect the essence of the right to data protection and provide for suitable and specific measures to safeguard the fundamental rights and the interests of the data subject.*

3. *Personal data referred to in paragraph 1 may be processed for the purposes referred to in point (h) of paragraph 2 when those data are processed by or under the responsibility of a professional subject to the obligation of professional secrecy under Union or Member State law or rules established by national competent bodies or by another person also subject to an obligation of secrecy under Union or Member State law or rules established by national competent bodies.*
4. *Member States may maintain or introduce further conditions, including limitations, with regard to the processing of genetic data, biometric data or data concerning health.*

The fourth part of Recital 51 specifically refers to photographs that should not be considered as part of special categories of data unless its entity falls under the definition of biometric data and if processed through specific measures can possibly lead to the unique identification or authentication of a natural person.

2.2 Main Principles related to the processing of personal data

There are many aspects whose risk is to incur in privacy and confidentiality breaches. The main risk is to incur in unauthorised capturing, storing and disclosure of images. That is the

reason why, for instance, in case of clinical photography of the patients, the medical staff has the obligation of declaring the scope of such photography. In this case, the data subject is the patient itself and the medical staff has the role of controller, whose main obligations are to determine the purposes and the means of the processing of personal data. The declaration of the purposes and the means must be compliant with the principles of Lawfulness, Fairness and Transparency reported in the GDPR:

- The principle of Lawfulness requires the processing of personal data to take place in a lawful manner. Lawful processing of data needs the consent of the data subject. The first paragraph of article 6 of the GDPR reports the main aspects of this principle:

    1. *Processing shall be lawful only if and to the extent that at least one of the following applies:*
        a. *the data subject has given consent to the processing of his or her personal data for one or more specific purposes,*
        b. *processing is necessary for the performance of a contract to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract,*
        c. *processing is necessary for compliance with a legal obligation to which the controller is subject,*
        d. *processing is necessary in order to protect the vital interests of the data subject or of another natural person,*
        e. *processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller,*
        f. *processing is necessary for the purposes of the legitimate interests pursued by the controller or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, in particular where the data subject is a child.*

- The principle of Fairness requires the processing of personal data to be implemented in a fair manner. The controller, in fact, should notify the data subject and the public that the data will be processed in a lawful and transparent way. They have also the duty of proving the compliance of data processing with the GDPR.

- The principle of Transparency requires the data processing to be carried on in a transparent manner. The controllers, in fact, have the obligation of communicating to the data subject how and what reasons personal data are used for. Recital 58 of the GDPR reports the main implications of such principle:

    1. *The principle of transparency requires that any information addressed to the public or to the data subject be concise, easily accessible and easy to understand, and that clear and plain language and, additionally, where appropriate, visualisation be used.*
    2. *Such information could be provided in electronic form, for example, when addressed to the public, through a website.*
    3. *This is of particular relevance in situations where the proliferation of actors and the technological complexity of practice make it difficult for the data subject to know and understand whether, by whom and for what purpose personal data relating to him or her are being collected, such as in the case of online advertising.*
    4. *Given that children merit specific protection, any information and communication, where processing is addressed to a child, should be in such a clear and plain language that the child can easily understand.*

These principles must be applied in any case that involves the processing of personal data. Face Recognition Technology needs to respect the rules of the GDPR since the images produced are data that can bring to the identification of the data subject. Personal data, in fact, are defined in Article 4 of the GDPR:

"personal data means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person."

Since pictures of data subject contain confidential information, Face Recognition Technology must be regulated, and the purpose of the processing of images of any kind must fall under the scope of the GDPR. Furthermore, the processing of personal images requires definition of purposes and it must be limited to it. There cannot be any further

processing that are not directly correlated with the original one without having contacted and asked for another specific consent to the data subject. Consequently, any processing of personal images must be done for a specific and well-defined scope or other purposes that are aligned with the original one. This principle, called as “The principle of purpose limitation”, is another fundamental principle of the GDPR and it is strongly connected with the other principles of the European Data Protection Law, such as transparency, predictability and user control. Recital 50 of the GDPR says:

“The processing of personal data for purposes other than those for which the personal data were initially collected should be allowed only where the processing is compatible with the purposes for which the personal data were initially collected.”

Another principle mentioned in the European Data Protection Law is the ‘Data Minimisation principle’, which requires the processing to be limited to what is necessary to fulfil the pre-defined purpose.

3. Rights of the data subjects

Abstract

Surveillance Systems and Commercial applications exploit the use of Face Recognition to pursue different goals. These two applications of Face Recognition Technologies may incur in damages to individual's privacy. That is the reason why the GDPR has stated different rights exercisable by individuals whose sensitive data are processed or are improperly disclosed. Finally, there is a focus on the legal grounds for these technologies.

3.1 Surveillance Systems

As already mentioned in the Introduction, Face Recognition Technology has many applications and one of them is the Surveillance and Monitoring systems. This application can interfere with individuals' privacy and it may get into contrast with situations of public interest. Individuals, in this case, are subject to face recognition processes without being totally aware.

Surveillance Systems are located in many places such as stations, airports or even supermarkets and they have the capability of collecting large amounts of images or videos, without effectively having the consent of the data subjects. Consequently, individuals are subject to deprivation of choice since they cannot choose whether to be filmed or not. However, the GDPR justify the processing of personal data in case the task is carried out in the public interest or in the *"exercise of official authority vested in the controller"*. The Recital 45 reports:

"Where processing is carried out in accordance with a legal obligation to which the controller is subject or where processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority, the processing should have a basis in Union or Member State law."

For instance, surveillance may have an important role in case of dangerous situations such as terroristic attacks. This would be the case in which the use of surveillance and the images retrieved of a suspect individual can have a decisive role during investigations.

3.2 Commercial Applications and Rights of the data subjects

Commercial applications can take advantage of Face Recognition Technology. Many companies, in fact, use Face Recognition to improve costumers' experience and to maximize profits. For instance, Facebook tags people in the pictures by prompting users to identify their friends, with whom they communicate the most. This practice impacts the systems through which it is suggested which people to tag in a certain photograph. Facebook in fact groups pictures based on the presence of similar faces in them. This activity avoids the consent request of the data subjects and it seems unethical, even if there is the possibility of changing the privacy settings.

The consent given by the data subject is the legal basis for any kind of processing, except for some cases mentioned in Chapter 2. Furthermore, the data subject can exercise rights over his or her personal data, which in this specific case are personal identifiable images.

An example of right that correspond to an obligation of the controller is:

- the right to be informed: controllers have the obligation to inform the data subject about the purposes, how and which personal data are being collected. Under the first paragraph of Article 8 of the Modernised Convention 108, the content of the information includes:
  1. *Each Party shall provide that the controller informs the data subjects of:*
     a. *his or her identity and habitual residence or establishment,*
     b. *the legal basis and the purposes of the intended processing,*
     c. *the categories of personal data processed,*
     d. *the recipients or categories of recipients of the personal data, if any; and*
     e. *the means of exercising the rights set out in Article 9,*
     f. *as well as any necessary additional information in order to ensure fair and transparent processing of the personal data.*
  2. *Paragraph 1 shall not apply where the data subject already has the relevant information.*
  3. *Where the personal data are not collected from the data subjects, the controller shall not be required to provide such information where the processing is expressly prescribed by law or this proves to be impossible or involves disproportionate efforts.*

The data subject, in the context of Face Recognition Technologies, is empowered by many other rights such as:

- Right to rectification,

- Right to erasure,
- Right to restriction of processing,
- Right to object,
- Right to lodge a complaint.

Under EU law and Council of Europe law, data subjects can exercise the right to have their personal data rectified. This right is strongly connected with the notion of accuracy of personal data. The accuracy principle of the GDPR states that inaccurate data must be deleted without delay and the data must be checked out regularly to ensure the compliance with such principle. According to Article 5 (1), personal data must be:

1. *accurate and, where necessary, kept up to date; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay ('accuracy').*

The debate held on the 28th of January 2021 of Consultative Committee of the Convention for the protection of individuals with regard to automatic processing of personal data revealed that:

"Entities have to ensure that the biometric templates and digital images are accurate and updated. For instance, the quality of images and biometric templates inserted in watchlists must be checked to prevent potential false matches since low quality images can cause an increase in the number of errors. This is directly linked to the sources of the images compiled in the watchlist, which require strict respect of the data protection principles such as the principle of purpose limitation. In case of false matches, the entities will take all reasonable steps to correct future occurrences and ensure the accuracy of digital images and biometric templates", i.e., in case of false matches, which means that the individual is wrongly associated to a face, the data subject can ask for the rectification of such information in order to avoid the repetition of this kind of mistakes.

The right to erasure, also known as the right to be forgotten, provides the data subject the ability of asking for the deletion of their personal information. Article 17 of the GDPR explains in what kind of situations data can be effectively erased. Data need to be erased without delay for the following reasons:

- the data are not necessary to pursue the goal for which they were collected or processed,
- the data subject withdraws the consent that initially allowed the data processing,
- the purposes of the processing are not the ones for which the data subject gave consent to,
- the personal data have been processed in an unlawful manner.

The controllers must prove at any time that there is a legal basis for the processing of personal data. There are many exceptions to the right to erasure and the most relevant one in case of face recognition applications is related to reasons of public interests in the area of public health.

The rights to restriction of processing and to object empowers the data subject in limiting the processing of personal data. In these two cases there are exceptions too.

The debate of Consultative Committee of the Convention for the protection of individuals with regard to automatic processing of personal data held on the 28th of January 2021 underlined that "*in the case of limitation of the rights of data subjects, law enforcement authorities have to inform data subjects inter alia about their right to lodge a complaint with supervisory authorities, and about their general right to remedy*". Individuals have the right to complaint with the Supervisory Authority of competence in case they think that the data processing is taking place without being in accordance with the law.

3.3 Legal Grounds for Face Recognition Technologies

As previously reported, sensitive personal data should not be subject to processing unless there is a specific reason for it. These technologies are particularly intrusive, and they may represent a threat to individual's privacy: it depends on situations and there are cases where the processing implemented by such technologies can be subject to limitations or to total prohibition. During the discussion of the Consultative Committee held on January 2021, it has been stated that:

- *"The use of live Facial Recognition Technologies in uncontrolled environments, in light of the intrusiveness it bares upon the right to privacy and the dignity of individuals, coupled with a risk of adverse impact on other human rights and*

fundamental freedoms, should be subject to a democratic debate on its use and the possibility of a moratorium pending complete analysis",

- *"The use of Facial Recognition for the sole purpose of determining a person's skin colour, religious or other beliefs, sex, racial or ethnic origin, age, health condition or social condition should be prohibited unless appropriate safeguards are provided for by law to avoid any risk of discrimination".*

The term 'uncontrolled environment' refers to those cases in which public area are subject to video surveillance and individuals can pass through without being aware.

Furthermore, another very important aspect is the legal framework under which Face Recognition Technologies must be regulated. As already mentioned in paragraph 1.2, the main steps in Face Recognition are the following: acquisition, pre-processing, feature extraction, classification, verification/identification. The legal framework must regulate not only these aspects of the process applied in this kind of technology, but also the sector of application and the intrusiveness.

Legislators must ensure that the digital images captured through Face Recognition Processes are not subject to biometric feature extraction, unless they are processed for legal purposes stated before the start of the processing itself.

Extracting biometric templates, which are the unique information that would lead to the identification of the individual, implies sensitive data processing and that is the reason why it is fundamental to set a legal basis that in most cases is not the mere consent of the data subject. However, if the personal identifiable images are retrieved in controlled environments, such as in context where private entities are involved, the consent will represent the legal basis for the extraction of biometric information. Most of Face Recognition Technologies are implemented in video-surveillance, in areas where individuals may be not aware of the presence of such systems. The extraction of biometric features takes place in case of necessity, such as for reasons of public interest. Therefore, the legal basis for Face Recognition Technologies used by public authorities cannot be the consent of the data subject. The same happens in case Face Recognition Technologies are used by private entities.

It is necessary then to set specific rules. It emerged from the debate of the Consultative Committee that:

“Biometric data processing by Facial Recognition Technologies for identification purposes in a controlled or uncontrolled environment should be restricted, in general, to law enforcement purposes. It should be carried out solely by the competent authorities in the area of security. Laws can provide different necessity and proportionality tests depending on whether the purpose is verification or identification, considering the potential risks to fundamental rights and as long as individuals' images are lawfully collected. For identification purposes, the strict necessity and proportionality must be observed both in the setting-up of the database (watchlist) and deployment of (live) Facial Recognition Technologies in an uncontrolled environment. Laws should provide clear parameters and criteria that law enforcement authorities should adhere to, when creating databases (watchlists) for specific, legitimate and explicit law enforcement purposes (for example suspicion of severe offences or risk to public security). Considering the intrusiveness of these technologies, in the deployment phase of the live Facial Recognition Technologies in uncontrolled environments, the law shall ensure that law enforcement authorities demonstrate that a variety of factors, including the place and timing of deployment of these technologies, justify the strict necessity and proportionality of the uses”.

4. Data Security

Abstract

Face Recognition implies the acquisition of personal identifiable images that are defined by the GDPR as sensitive biometric information. This Chapter faces the issues of prevention and reaction to possible impostor's attacks and the importance of codes of conduct and certifications in which the information about all the mitigating measures must be reported. Finally, possible vulnerabilities of these technologies are mentioned.

4.1 Prevention and Mitigation measures

The Consultative Committee of the Convention for the protection of individuals with regard to automatic processing of personal data give also strong direction with regard to the security measures to be adopted. The consequences of failures in the systems may cause data breaches, which are even more severe than other type of data since personal identifiable images are considered as sensitive information. Entities that make use of Face Recognition Technology must implement strong security measures to avoid the loss of sensitive data or unauthorized access. The security system assessment is composed by two main elements:

- prevention system,
- reaction system.

The former consists in implementing measures in order to prevent and react to "technology-specific" attacks. The reaction system is composed by all those measures that help in mitigating the negative consequences of a possible data breach.

In case of violation of data protection, it is necessary to inform the Supervisory Authority and the data subject, according to Article 33 of the GDPR. The text of the Article says:

1. *In the case of a personal data breach, the controller shall without undue delay and, where feasible, not later than 72 hours after having become aware of it, notify the personal data breach to the Supervisory Authority competent in accordance with Article 55* (regarding the competences of the Supervisory Authority)*, unless the personal data breach is unlikely to result in a risk to the rights and freedoms of natural persons. Where the notification to the Supervisory Authority is not made within 72 hours, it shall be accompanied by reasons for the delay.*

2. *The processor shall notify the controller without undue delay after becoming aware of a personal data breach.*

3. *The notification referred to in paragraph 1 shall at least:*

    a. *describe the nature of the personal data breach including where possible, the categories and approximate number of data subjects concerned and the categories and approximate number of personal data records concerned,*

    b. *communicate the name and contact details of the data protection officer or other contact point where more information can be obtained,*

    c. *describe the likely consequences of the personal data breach,*

    d. *describe the measures taken or proposed to be taken by the controller to address the personal data breach, including, where appropriate, measures to mitigate its possible adverse effects.*

4. *Where, and in so far as, it is not possible to provide the information at the same time, the information may be provided in phases without undue further delay.*

5. *The controller shall document any personal data breaches, comprising the facts relating to the personal data breach, its effects and the remedial action taken. That documentation shall enable the Supervisory Authority to verify compliance with this Article.*

Furthermore, the Consultative Committee outlined that *"Security measures should evolve over time and in response to changing threats and identified vulnerabilities. They should also be proportionate to the sensitivity of the data, to the context in which a specific Facial Recognition Technology is used and its purposes, to the likelihood of harm to individuals and other relevant factors. Strict retention and disposal practices - through safe procedures - for Facial Recognition data, with the shortest possible retention periods, also contribute to reducing security exposures".*

The typology of security measures used must be reported in the Data Protection Impact Assessment (DPIA). In fact, the entities that make use of Facial Recognition Technologies must:

- implement transparent policies, procedures and practices,
- publish reports aligned with the transparency principle,
- promote training program every time there is Facial Recognition Technologies' update,
- nominate an internal committee of experts that evaluate the procedures involving Face Recognition Technologies.

According to Article 35 of the GDPR, the DPIA must be necessarily carried out in case the processing of personal data may incur in high risks to the rights and freedoms of data subjects. In case of involvement of Facial Recognition Technologies, the DPIA must be carried out since the personal data processed are considered sensitive data. This assessment must focus not only on the risks to the fundamental rights at stake, but also on the measures to prevent data breaches and mitigating measures to diminish the negative effects of such breaches. If the data subject, whose personal identifiable images processed, is a child, the measures must be even stronger, since minors, according to the GDPR, need a special protection, as reported in Recital 38:

"Children merit specific protection with regard to their personal data, as they may be less aware of the risks, consequences and safeguards concerned and their rights in relation to the processing of personal data.

Such specific protection should, in particular, apply to the use of personal data of children for the purposes of marketing or creating personality or user profiles and the collection of personal data with regard to children when using services offered directly to a child.

The consent of the holder of parental responsibility should not be necessary in the context of preventive or counselling services offered directly to a child."

4.2 Codes of conduct and certifications

Codes of conduct should be encouraged by the Member States, Supervisory Authority, the Board and the Commission. They are designed to define more specific guidelines for data protection and privacy in specific sectors, looking also at the specific needs of micro, small and medium sized enterprises. Face Recognition is applicable and implementable in many sectors and it needs a more focused regularization (Article 40 of the GDPR).

The Member States, Supervisory Authority, the Board and the Commission should encourage also the definition of certification mechanisms *"for the purpose of demonstrating compliance with this Regulation of processing operations by controllers and processors"* (Article 42 of the GDPR).

The Consultative Committee underlined the importance of certification mechanisms by saying that:

"Legislators and decision-makers should use different mechanisms to ensure the accountability of the developers, manufacturers, service providers or entities using these technologies. The setting up of independent and qualified certification mechanism for facial recognition and data protection to demonstrate full compliance of the processing operations carried out would be an essential element in building users' confidence. Such a certification could be implemented according to the application of artificial intelligence used by the Facial Recognition Technology: one type of certification to categorise structures (design of algorithm, integration of algorithm, etc) and another to categorise algorithms (computer recognition, intelligent search, etc.)".

4.3 Face Recognition Technologies vulnerabilities

Despite all the prevention measures applied to Face Recognition Technologies, they are still vulnerable to different types of attacks. These technologies are based on algorithms that can incur in error rates, whose consequences are either false positives, which consist in making wrong matches, or false negatives where true matches are not extracted through the algorithms. There are two rates then: False Accordance Rate and False Rejection Rate. The former relates to the incorrect acceptance of a match (false positive), the latter indicates the incorrectly rejected result (false negative). These two rates need to be balanced: the processor needs to choose whether she prefers to have a higher possibility of incurring in false positives or in false negatives.

The vulnerability of a Face Recognition System is to be circumvented through the action of spoofing an identity of registered user for instance. The use of a picture to circumvent the system is one of the most common spoofing practices. The picture can be taken by directly photographing the user or by looking for pictures on the social network. The advent of 3D

printers impacted Face Recognition Technologies, since an impostor can fake face images by wearing a mask produced by a 3D printer. That increases the probabilities of incurring in system attacks.

Another issue to consider when dealing with the choice of False Accordance Rate and False Rejection Rate is that if, for instance, during an investigation, too many false positives are retrieved, innocent people are accused or if there are too many false negatives, the true person is not identified.

A perfect system does not exist, and the acceptable false/positive rate is decided by the processors themselves.

In case of face spoofing, the prevention measures to adopt are systems capable of understanding whether the face is real or fake by implementing for instance:

- Texture Analysis, which consists in detecting whether the skin is natural, or it is composed by an unnatural texture.
- Motion Analysis, which consists in detecting whether the movements made are discernible in real.
- Liveness Analysis, which allows the capturing of signs of life, like the eye-blinks.

4.4 Pseudonymization of face images

Images, once captured and associated to an identity, that can possibly include name, surname, date of birth etc., are stored to have readily accessible information. These information are accessible only by entities of competence, such as the controllers or the processors, who act on behalf of the controller.

Sensitive information must be subject to de-identification measures, in such a way that an impostor would not be able to get to the identity of the person looked for. Article 25 of the GDPR states that:

"Taking into account the state of the art, the cost of implementation and the nature, scope, context and purposes of processing as well as the risks of varying likelihood and severity for rights and freedoms of natural persons posed by the processing, the controller shall, both at

the time of the determination of the means for processing and at the time of the processing itself, implement appropriate technical and organisational measures, such as pseudonymisation, which are designed to implement data-protection principles, such as data minimisation, in an effective manner and to integrate the necessary safeguards into the processing in order to meet the requirements of this Regulation and protect the rights of data subjects".

Article 25 of the GDPR underlined then the importance of adopting measures that bring to effective protection against attacks that can possibly incur in dispersion of sensitive information of individuals. The ideal protection would be anonymisation, however it is not possible since Face Recognition Technologies are specifically designed to associate a face to a person, given a set of people's identities in a database. Furthermore, the GDPR would not be applicable to completely anonymised face images, as Recital 26 states:

"The principles of data protection should apply to any information concerning an identified or identifiable natural person.

Personal data which have undergone pseudonymisation, which could be attributed to a natural person by the use of additional information should be considered to be information on an identifiable natural person.

To determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly.

To ascertain whether means are reasonably likely to be used to identify the natural person, account should be taken of all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments.

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.

This Regulation does not therefore concern the processing of such anonymous information, including for statistical or research purposes."

Pseudonymization is a de-identification procedure that Article 4 of the GDPR defines as a method or set of methods applied to personal data in such a way that they cannot be attributed to the data subject without using additional information*, "provided that such additional information is kept separately and is subject to technical and organisational measures to ensure that the personal data are not attributed to an identified or identifiable natural person".*

An example of de-identification method for face images is to cover the eyes, in such a way that many facial features cannot be retrieved, whereas the others remain to make the picture associable to an identity thanks to the use of additional information.

The extreme solution would be to cover the whole image, but this would lead to anonymisation, whose consequence is the impossibility of identifying the person, and such covert image would be of no use, since all facial characteristics are obscured. Furthermore, as previously highlighted, anonymous data are not subject to the GDPR.

5. Face Verification

Abstract

Face Recognition is a very challenging task through which it is possible to detect human faces in an image and give an identity to those faces. In general, as mentioned many times in the previous chapters, the goal of Face Recognition Technology is to associate an unknown face to other pictures in the database. Machine Learning and Deep Learning Algorithms are used to analyse whether images contain human faces and whether faces in the different images belong to the same person. Sometimes the combination of Machine Learning and Deep Learning in Face Recognition can bring to many advantages, that will be exploited in the analytical part of this study. This Chapter proposes two methodologies for Face Verification.

5.1 Face images used for Face Verification

The first part of the project is dedicated to Face Verification. As mentioned in the previous chapters, there is a distinction between Face Verification and Face Identification: the former implies a one-to-one comparison, the latter consists in a one-to-many approach.
Sometimes there is the need of verifying whether a person is one or another. The analysis proposed here consists in comparing one target image to two other pictures and trying to understand to which image the target one is more similar. The target image considered is the following.

Figure 5.1: Target Image

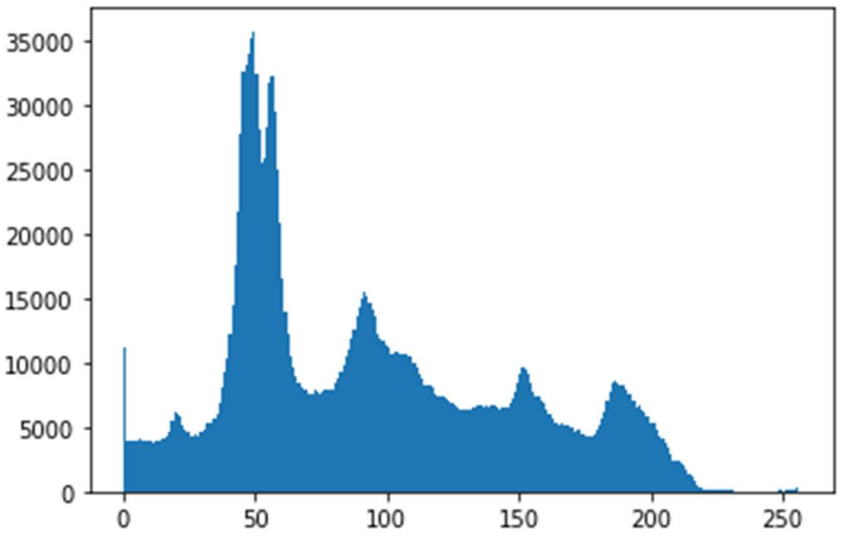
The two images over which the comparison is performed are reported below

Figure 5.2

Figure 5.3

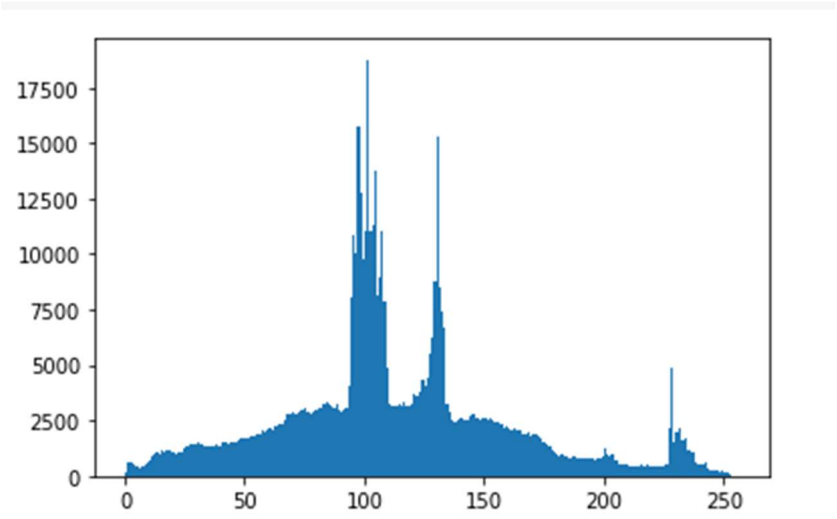
5.2 Preparation of data

The images need to be converted into grayscale first by using the 'open-cv' library. This library allows the creation of histograms, which give information about the pixels in the image. The resulting histogram for the target image is the following.

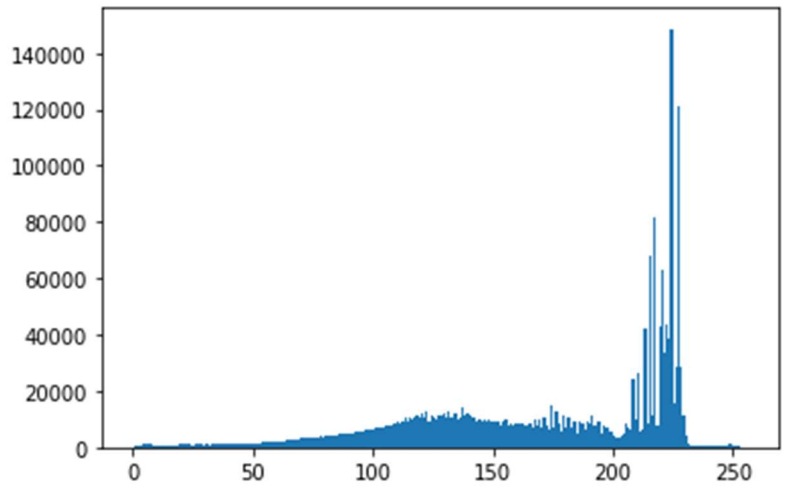


Histogram 5.1

The histograms for Figure 5.2 and Figure 5.3 are reported below.



Histogram 5.2



Histogram 5.3

The three histograms give an idea of the intensity distribution in each image. These are plots where on the X-axis there are the values of the pixels, usually from 0 to 255. On the Y-axis the number of pixels in the image are reported. These plots inform on the brightness, contrast, and other aspects of the image. The left side of the histogram plot reports the number of darker pixels, whereas the right side reports the brighter pixels. The conditions of lights and shadows may be given also by the characteristics of the face itself. That is the reason why this technique is used to compare human faces.

The Histogram 5.1, associated to the target image, shows that there is a higher number of values associated to darker pixels. Looking at Histogram 5.2 and Histogram 5.3, the former shows more pixels in the central area, whereas the latter one shows a higher number of pixels in the brighter area. The histograms are represented by and array of numbers, each of which indicates how many pixels for a certain pixels' type are present in the image.

5.3 Euclidean Distance and Mean Squared Error

The approach proposed consists in the computation of the distance between arrays. The distance used is the Euclidean one. This computation leads to the expected result. The target image is more similar to the Figure 5.2 rather than Figure 5.3.

It is also possible to use another approach which consists in the application of Mean Squared Error. This measure is indicated as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

where *f*(*xi*) is the prediction that *f* gives for the *i*th observation. It is usually applied to evaluate the performance of a statistical method, measuring the average of the squares of the errors. In case of Face Verification, it allows the computation of the overall errors computed over the pixels' values. The target image is in fact is converted to grayscale and the numbers for each type of pixel is found. All the needed information is set and it is now possible to compute the Mean Squared Error. The result is again what expected, since the value of the Mean Squared Error between the target image and the first 'test' image, indicated as Figure 5.2 is strictly lower than the Mean Squared Error computed between the target image and the second 'test' image.

This kind of evaluations are based only on the characteristics of pixels, that depend on the pose, the illumination, facial expression and so on. That is why this methodology may be not so precise. This methodology should be improved by the research of facial features that can increase the performance and better classify the images.

6. Face Identification

Abstract

Face Identification includes all those methods and procedures that, starting from a single image, aims at finding images that ideally show the same person. It is not a one-to-one mechanism as in Face Verification anymore. This methodology involves a higher number of images and most of the times it deals with large-scale data issues.

6.1 Methodology used for Face Identification

As mentioned before, the complexity of Face Recognition tasks starts by preparing and pre-processing the data, that in this case are images. It is not possible to use the images as they are at the beginning: they need to be converted into arrays. In this project the module 'skimage.io' with function 'imread' Is used and it converts the picture into a NumPy array of pixels. This is not enough for Face Recognition since pixels reports only information about the colours, their intensity, the brightness, and shadow. Two images may look similar if you consider two arrays of pixels, as shown in the previous part, however if you have people in the image, many more steps should be performed.

The part of the project reported in this Chapter is divided into two parts:

- the first faces the problem of face detection,
- the second focuses on the problem of face recognition, starting from a single image.

The dataset used is taken form Kaggle platform and it contains pictures of fourteen different celebrities (Ben Afflek, Arnold Schwarzenegger, Lauren Cohan, Mindy Kaling, Simon Pegg, Jerry Seinfeld, Keanu Reeves, Madonna, Dwayne Johnson, Will Smith, Kate Beckinsale, Sofia Vergara, Elton John, Anne Hathaway) in different position and different situation.

6.2 Face Detection

As mentioned in the previous Chapters, Face Detection is the step performed before Face Recognition or Verification.

The main steps performed to detect the presence of human faces in the image are:

- grayscale conversion,
- creation of a Cascade Classifier,

- construction of a rectangle around the human face detected.

Images are converted to greyscale first through the library ‘OpenCV’, which is designed to solve this kind of problems. This library is based on the use of Machine Learning algorithms for computer vision. The problem of Face Detection is that it consists in analysing each block in a picture in order to understand whether that block contains a human face or not. That is the reason why ‘OpenCV’ is used: it faces the detection problem by using cascades. The project uses an XML file that contain OpenCV data used to perform human face detection. In this case the project uses ‘haarcascade_frontalface_default.xml’. Consequently the ‘faceCascade’ is generated and it is applied to all the images in the dataset. The next step is to create a rectangle around the face to prove that the face has been effectively detected. The following pictures are reported as an example.

Figure 6.1: Original image

Figure 6.2: Face detected

Figure 6.3: Original image

Figure 6.4: Face detected

6.3 Face Identification

The next part focuses on the research of pictures that show the same person, starting from a single image. The approach proposed uses the ‘face_recognition’ library, that allows the recognition and manipulation of images from Python. It is based on the library ‘dlib’, which contains Machine Learning and Deep Learning algorithms. Deep Learning algorithms are the one that characterize ‘face_recognition’ module. The images, already saved as array through the ‘imread’ function, are converted to ‘rgb’ (red, green and blue) colours. This data preparation is needed in order to perform the features extraction, which consists in finding the ‘face encodings’ that are retrieved using the pre-trained ‘face_recognition’ library.

In computer vision and especially in Face Recognition tasks, it is necessary to extract the features if you want to compare images and detect the presence of the same person in dataset. The function used in the project maps faces to vectors of 128 features, which can be defined also as face descriptors. These are the most relevant information about the face detected. These vectors are called embeddings since the image is compressed in a vectorial representation of 128 features, reducing the dimensionality of the data. In fact, if first the image is read and converted to an array of thousands of pixels, with face embeddings it is possible to reduce the number of information to only 128. It is the result of Deep Learning, especially Neural Networks. Two images, showing the same person, are ideally associated to similar face encodings vectors, whereas if you take as input two pictures of different people, the resulting vectors have very different values.

Figure 6.5: Original image

```
[array([-0.01666883,  0.14427619, -0.03182324, -0.0826207 , -0.15808246,
        -0.04623568,  0.05356185, -0.10417124,  0.25942305, -0.10354534,
         0.18379301, -0.10144293, -0.25204176,  0.05676649, -0.0724926 ,
         0.14668152, -0.27197969, -0.1811347 , -0.07873083, -0.00782898,
         0.09381774, -0.05078962,  0.09137599,  0.20281518, -0.08012062,
        -0.40024054, -0.10748583, -0.07248709,  0.05429133, -0.17046064,
         0.10718809,  0.07718392, -0.23278058, -0.02885288, -0.03983937,
         0.07037222, -0.06553915, -0.19661599,  0.20698856, -0.05395769,
        -0.22694282, -0.03419458,  0.06182137,  0.27670568,  0.14340988,
         0.02893937,  0.04563285, -0.10915135, -0.00069422, -0.20317867,
         0.0311112 ,  0.18551382,  0.01689524,  0.04898968,  0.02374539,
        -0.17342184,  0.03162076,  0.11724138, -0.21268986,  0.01277341,
         0.04644708, -0.02042129,  0.00306374, -0.09595769,  0.25992081,
         0.11552978, -0.09438765, -0.15523106,  0.19828971, -0.22626731,
        -0.06181563,  0.09688771, -0.13321178, -0.21630014, -0.20807301,
        -0.0641041 ,  0.46524903,  0.13783687, -0.08732112,  0.05972096,
        -0.12303077, -0.03173143, -0.00629845,  0.17084265, -0.02513519,
         0.00115793, -0.15581256,  0.14266503,  0.15807766, -0.04919042,
        -0.0965535 ,  0.21284199,  0.00800138, -0.09251457,  0.00747328,
         0.02164488, -0.11530905,  0.10308737, -0.15314785, -0.03008012,
         0.02484012,  0.01282218, -0.10075361,  0.08894454, -0.15112537,
         0.09251556,  0.03067736, -0.09137182, -0.01090664,  0.02966839,
        -0.14878874, -0.09992647,  0.17870307, -0.3454054 ,  0.18563288,
         0.19467561,  0.01701706,  0.17060077, -0.02502396,  0.15161607,
         0.06479508, -0.10234892, -0.19160624, -0.01697564,  0.05067183,
         0.0097712 ,  0.05529388,  0.00080448])]
```

Figure 6.6: Face encodings of Figure 6.5

The example reported above shows the transformation of the original image into an array of numbers that are generated by the Neural Networks in the trained module ‘face_encodings’. There are many micro distances such as the distance between the eyes, length of the nose, length and distance between the eyebrows, dimensions of the mouth, and so on. Face encodings are retrieved for each image in the dataset. Sometimes there are other faces in the picture and the algorithm detect them too and generate face encodings for all the faces in the image. That is why when the module ‘compare_faces’ is run to compare one picture to all the others returns different ‘True’ or ‘False’ on the same line, meaning that many faces in the same picture are detected. After having implemented this module, it is possible to retrieve those images that are effectively inferred as similar to the chosen one. The following pictures are inferred as showing the same person represented in the target image (in this case Sofia Vergara’s picture) by the algorithm.

Figure 6.7: Set of images showing the same person (Sofia Vergara)

The following set reports the images inferred as Lauren Cohan's photos.

Figure 6.8: Set of images inferred as showing the same person (Lauren Cohan)

Looking at the two sets of images (the one of Sofia Vergara's pictures and the one for Lauren Cohan's pictures) it is possible to see that they have one picture in common, the last photo in the first set and the second-last in the second set. This picture shows another person and that is the reason why it is inferred as false positive. The same happens for the last picture of the second set.

7. Face clustering and Euclidean distance

Abstract

Face Verification and Face Identification imply the use of a single image as a starting point. Face Verification implies then the comparison of the image with another one, whereas Face Identification implies the comparison of such image with many others, with the goal of finding which photos show the same person. However, it is possible to directly group images, ideally showing the same individual. The research of faces belonging to the same person is based on distance computation. A distance in fact is a measure that enables to classify how close two items are. The distances are computed over the arrays of 128 features used to identify the characteristic of each face. This Chapter proposes two different methods to group pictures on the basis of distance measures:

- clustering,
- computation of Euclidean Distance.

This analysis aims at using first clustering methods based on the computation of the Euclidean distance, then the Euclidean distances between couples of faces are calculated and the optimal threshold is found using binary logistic regression classifier and the ROC curve.

7.1 Clustering method proposed

Thanks to the extraction of the features, it is possible to use different methodologies to find images of the same person. One possible approach is clustering.

Clustering refers to a very wide area of Machine Learning techniques used to find groups or clusters in a dataset. The idea is to find subgroups of elements in the dataset that are inferred as similar. Consequently, it is needed a definition of what is similar and what is different. The approach suggested in this project uses the Hierarchical clustering. The initial assumption, in fact, is that there is no information about how many people there are in the dataset. It means that the information about the K value that is normally used in K-means clustering is still not available.

The idea is to use first the Hierarchical clustering to determine K value that will be then used in the performance of the K-means clustering method.

7.2 Hierarchical clustering

The Hierarchical clustering has a bottom-up approach: the visual representation is a dendrogram (which can be described as an upside-down tree) which is built by starting from the leaves that are combined into different fusions.

How it is possible to determine whether two items should be detected into one fusion and not another? This deals with the concept of dissimilarity that need to be discussed with the definition of pairs of groups of observation, not only pair of observation. In particular the notion of linkage needs to be considered: it determines the similarity of the observations in two different groups. There are four types of linkage: single, complete, average and centroid. The following table from the book 'An Introduction to Statistical Learning' summarize the differences and characteristics of the different types of linkages.

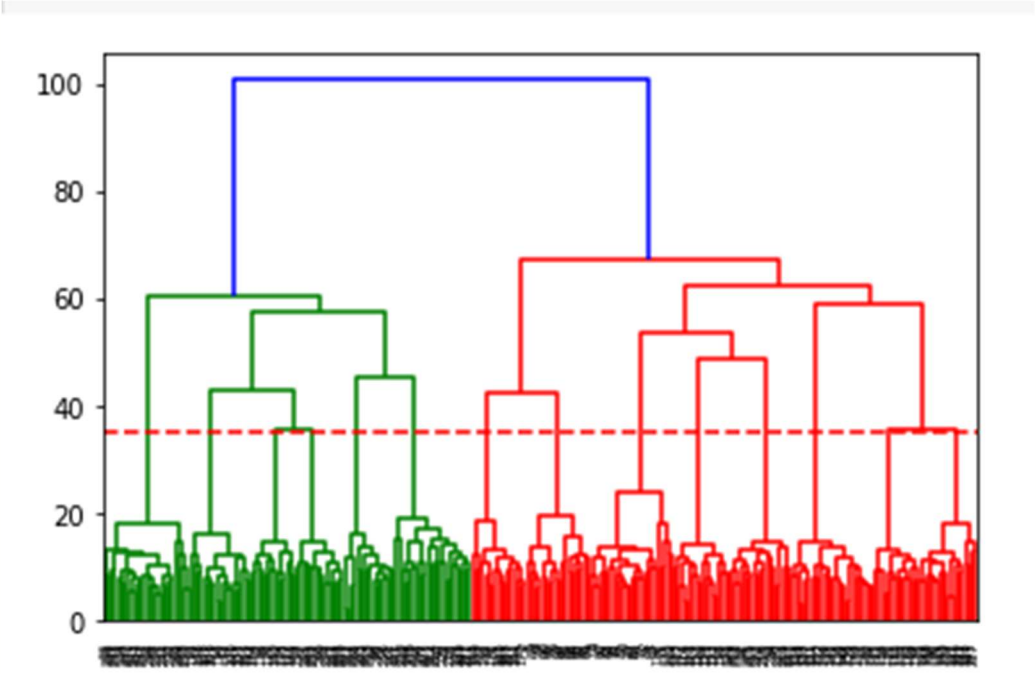
Linkage	*Description*
Complete	Maximal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B and record the *largest* of these dissimilarities.
Single	Minimal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length *p*) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Table 7.1

Furthermore, there is not a specific way to choose the number of clusters looking at the dendrogram. However, you must notice that the more you go upper along the dendrogram, a lower number of clusters and less specific clusters you have.

7.3 Hierarchical clustering on the data

The data on which the clustering is performed are composed by all the encodings previously extracted. These data are rescaled by applying the ‘whiten’ function, through which each observation is divided by its standard deviation to give to it a unique variance. The standard deviation is 1.001837 for all the features. The linkage is built by using the ‘ward’ method, which apply a recursively merging of clusters, minimally increasing the variance, and the ‘euclidean’ metric. The dendrogram is then displayed using the ‘scipy.cluster.hierarchy’ module. The following graph shows the dendrogram.



Graph 7.1

The dotted red line corresponds to y = 35. At this level, the number of clusters is equal to 14. This value is used as K value in the K-means clustering.

7.4 K-means clustering

The approach of K-means clustering implies the partition of the dataset into multiple subgroups, or clusters. In this case, the number of partitions is pre-defined. K-means clustering is based on two properties:

- for each of the K clusters, a certain number of observations is assigned,

- each observation is mapped to exactly one cluster, which lead us to the conclusion that there are no overlapping clusters.

A good performance of K-means clustering results when there is a little variance within the elements in one cluster. The problem the algorithm wants to solve is the following:

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

C_k is the cluster numbered k and $W(C_k)$ is the amount of which the elements in that cluster differ. The goal is then to minimize the variance between the element in the cluster. Most of the time the amount for which the observations in a cluster differ is represented by Euclidean distance. It is possible to rewrite $W(C_k)$ as:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

where the cardinality of C_k indicates the number of observations in the cluster k. The approach of clustering is then to minimize the sum of all the Euclidean distances between pairs of observations divided by the total number of observations in cluster k.

7.5 Results over the K-means clustering

K-means clustering is performed by setting the number of clusters equal to 14, value retrieved through the Hierarchical clustering. The following sets of pictures are associated to a specific cluster. The following set of pictures is associated to cluster 1.

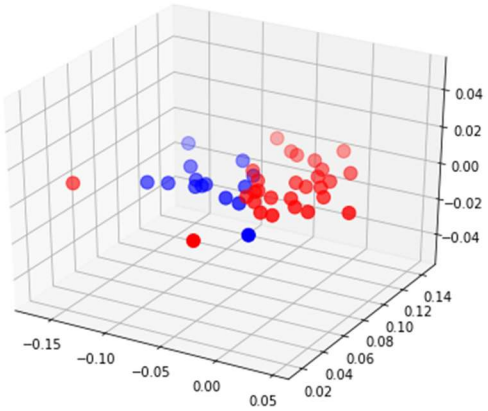
Figure 7.1: Set of pictures associated to cluster 1

The results are quite satisfactory since only four pictures represents someone else.
The following set of pictures are associated to the cluster 5.

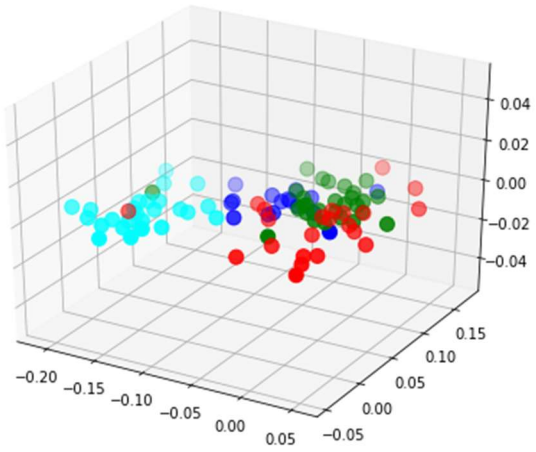
Figure 7.2: Set of pictures associated to cluster 5

As in the previous set of pictures, there are four false positives, due to similar values in the encodings vector. The clusters are determined on the basis of the Euclidean distances and these false positives may be due to different position and consequently there may be a distortion of the real measurements.
The graphs reported below represent respectively:

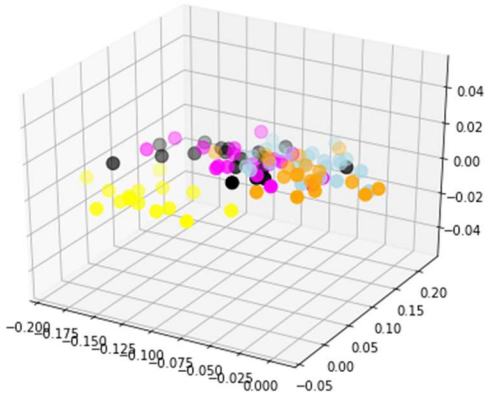
- The clusters 3 and 14.
- The clusters 1, 2, 3, 4.
- The clusters 5, 6, 7, 8, 9.
- The clusters 10, 11, 12, 13.



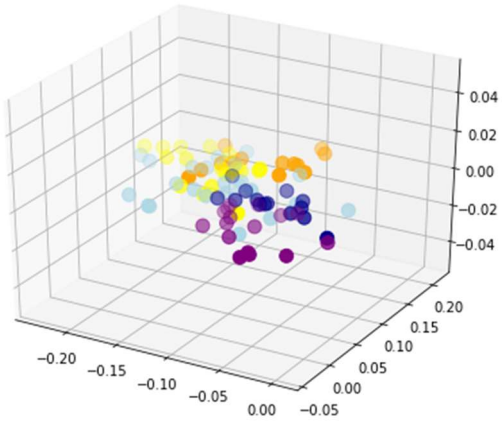
Graph 7.2



Graph 7.3



Graph 7.4



Graph 7.5

In all of them it is possible to see that there are very close clusters.

7.6 Euclidean distance

The Euclidean distance is one of the most common distance measures and this analysis aims at computing this distance between pairs of images and group those images that have the lowest distance as possible.

A pandas Data Frame is built: the column reports the index of the first picture, the index of the second picture, and the Euclidean distance computed over the vectors reporting the 128 features of each image. The first 5 rows of the data frame are reported.

Id1	Id2	Euclidean distances
0	0	0.000000
0	1	0.445862
0	2	0.549662
0	3	0.570183
0	4	0.886661

Table 7.2

The next part is structured as a classification problem, since the goal is to classify whether the two images, identified by 'id1' and 'id2', show the same person. A Binary Logistic Regression is performed.

The Binary Logistic Regression can be described as an extension of linear regression. The difference is that the dependent variable is dichotomous. The independent variables are used to determine whether a certain item is associated to one or the other class. Logistic Regression computes the probability of 'success' or failure'. In this case the 'success' refers to those situations where the two images are inferred as similar (representing the same person) and since the binary version is used, if the prediction is set to value 1 then the two images are likely to represent the same person, otherwise it is set to 0.

The factors used to predict whether two images show the same person are the Euclidean distances, all the features of the image indicated by 'id1' and the difference between the first feature of the encodings of the image identified with 'id1' and the first feature of the encodings of the image identified with 'id2'. This value is named 'Score_diff'. The dependent variable y is built: when the 'Score_diff' is lower than the half of the mean of all differences computed for each couple of images, the two image are likely to be showing the same person.

7.7 Binary Logistic Regression and Euclidean distances

The dependent and independent variables are prepared for the Binary Logistic Regression. Fist the data partition is performed using the 'sklearn.model_selection' module: the training set represents the 70% of the data, while the test set corresponds to the 30%.

The Logistic Regression is fitted over 'X_train' and 'y_train' and the accuracy of Logistic Regression classifier on test set is 0.89. The confusion matrix is computed:

[[14.504 1.049]
[1.348 5.458]].

It shows that the classifier predicts correctly 14.504 + 1.049 observations. The remaining 1.348 + 5.458 observations are incorrect predictions.

Finally the ‘classification_report‘ is built in order to understand better how the classifier works.

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.91	0.93	0.92	15553
1	0.84	0.80	0.82	6806
accuracy			0.89	22359
macro avg	0.88	0.87	0.87	22359
weighted avg	0.89	0.89	0.89	22359

Table 7.3: Classification report for classifier based on Euclidean distances

The Binary Logistic Regression is evaluated by considering the precision, recall and f1-score.

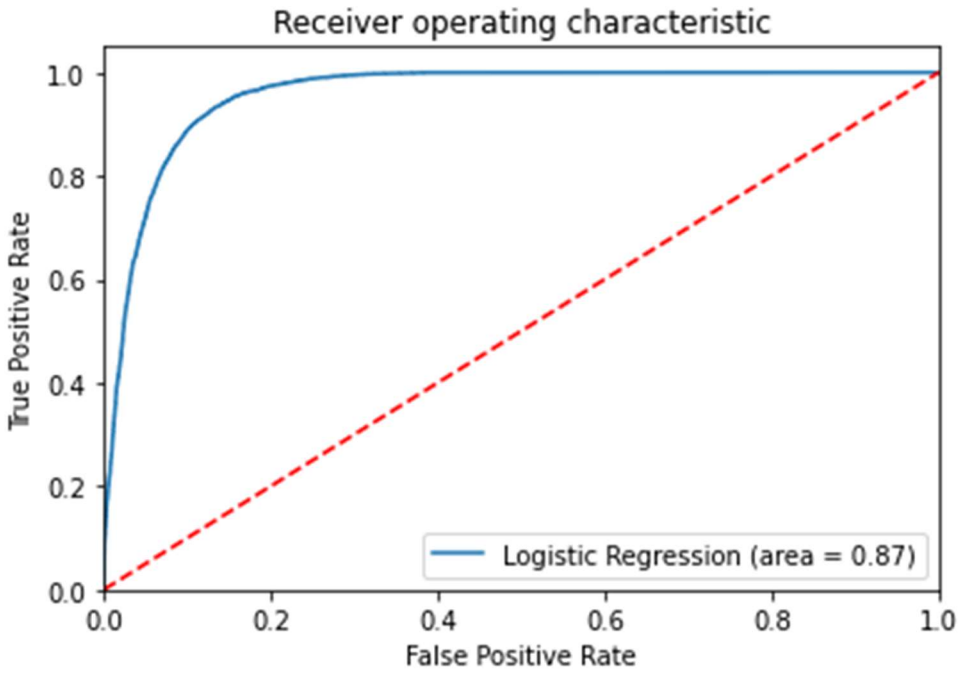
The precision score is defined as the ratio TP/(TP+FP) where TP is the number of true positives and FP is the number of false positives. It indicates how well the classifier predicts each class. In this case, the classifier is better at classifying class 0.

The recall score measures the ability of finding positive samples and it is defined by the ratio FP/(FP+FN). The recall scores are respectively 0.93 and 0.80 for class = and class 1. They are both good values, however it is confirmed again that the classier has a better performance when dealing with class 0. Finally, the f1-score is the weighted mean of precision and recall. The best f1-score is 1.0, the worst is 0.0.

The support indicates the number of occurrences for each class in the test set.

The last two lines report the macro average and the weighted average of the precision, recall and f1-score values of the two classes.

The performance of the classifier is evaluated also through the ROC curve.



Graph 7.6: ROC curve

The Receiver operating characteristic curve is another common tool to evaluate the performance of the classifier. It plots the False Positive Rate on the x-axis and the True Positive Rate on the y-axis. This graph confirms once more the good performance of the classifier. The ROC curve computed over the predicted values and the Euclidean distances returns values of the false positive rate and false negative rate for each possible threshold. There are many thresholds that can be chosen.

The lower the threshold, the lower number of photos will be grouped together. For these reasons, the threshold chosen is 0.4013106. The result is structured as a list of lists, one for each picture. Since there are lists with common values, a merge is needed. That is why the lists with common elements are merged between them. There are many sets with just one value, corresponding to one picture and it means that in the dataset there is no other image showing a person with encodings Euclidean distance less than 0.4013106. These sets are not considered in the following part. It may be possible to increase the threshold, but there is the risk to incur in more false positives. Two examples are reported below.

Set 1:

Figure 7.3

Set 2:

Figure 7.4

In both sets there are false positives, however the greatest part of the images shows the same person. An option to decrease the number of false positives may be to lower even more the threshold, however this would lead to higher probability of incurring in false negatives. A compromised is needed.

8. Face Clustering with Mahalanobis distance

Abstract

This Chapter proposes another method to search for similar images: the use of Mahalanobis distance. This measure allows the detection of outliers. However, the main purpose of this part of the analysis is to understand how the Mahalanobis distance influences the results and to make a comparison with the Euclidean Distance.

8.1 Mahalanobis distance and outliers' detection

The Mahalanobis distance is the second distance measure used in this project. This distance has the ability of detecting multivariate outliers. The outliers in Face Recognition Technology are given especially by those images inferred as similar even if they are not. As previously mentioned, these images are called false positives.

The Mahalanobis distance measures how close a point, that in this case is represented by an encodings vector, is to a distribution D, whereas the Euclidean distance is the distance between two points. The Mahalanobis distance differs from the Euclidean one since:

- the variables are transformed into uncorrelated ones,
- the variance of the features is set to 1,
- the Euclidean distance is then computed.

Mahalanobis distance is defined as:

$$D^2 = (x - m)^T \cdot C^{-1} \cdot (x - m)$$

Where:

- D^2 is the square of the Mahalanobis distance,
- X is the vector of the observations,
- M is the vector of mean values of independent variables (mean of each column),
- C^{-1} is the inverse covariance matrix of independent variables.

The term (x – m) tells us how close the vector is to the mean and it is divided by C^{-1}, which is the inverse of the covariance matrix of the independent variables. If the variables in the data are strongly correlated, the values of the covariance will be high, and the distances will

be reduced. However, if the values in the covariance are low, the variables are not correlated, and the distances will be not reduced.
The first step performed in this part of the project consists in building a data frame where the two ids of the images considered and their Mahalanobis distance are reported.
The Mahalanobis distance is computed using the ‘scipy’ package.

8.2 Impact of Outliers

The research of the outliers takes place. The Mahalanobis distances are computed over the variables in the dataset. The number of total outliers in the observations is quite elevated, however it is something expected since we are dealing with human faces and there are a lot of possible variants in the picture that can influence the final result.
The dataset used in fact contains many images where there are many factors that generate a degradation in the Face Recognition system. Face Recognition systems, in fact, usually deal with variations in face images such as the position of the person, expression, but also the conditions of acquisition of the image. All these factors that cause a decrease in performance can be called noisy factors, whose consequence is to generate noisy faces. The dataset used contains many pictures where people are not in front of the camera, but they are photographed from different angulations, with different illumination, different hair looks, with or without glasses, with different facial expressions and so on.
That is the reason why, there are so many images that are detected as outliers. The following set of pictures reports the first ten noisy faces found through the outliers’ detection.

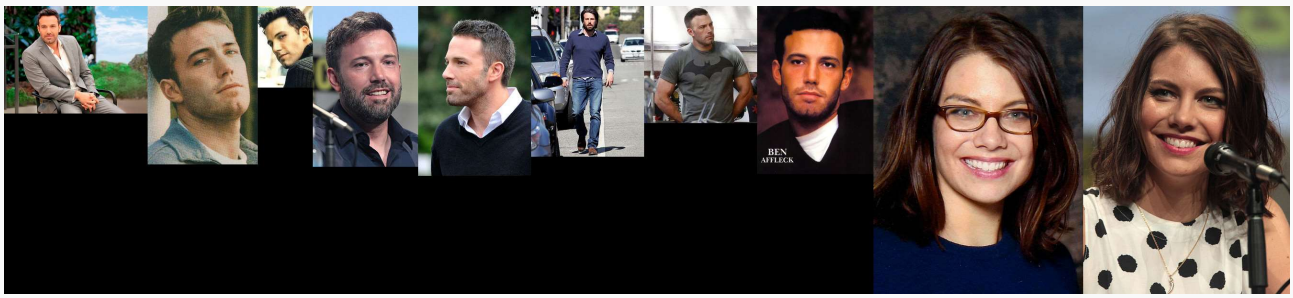
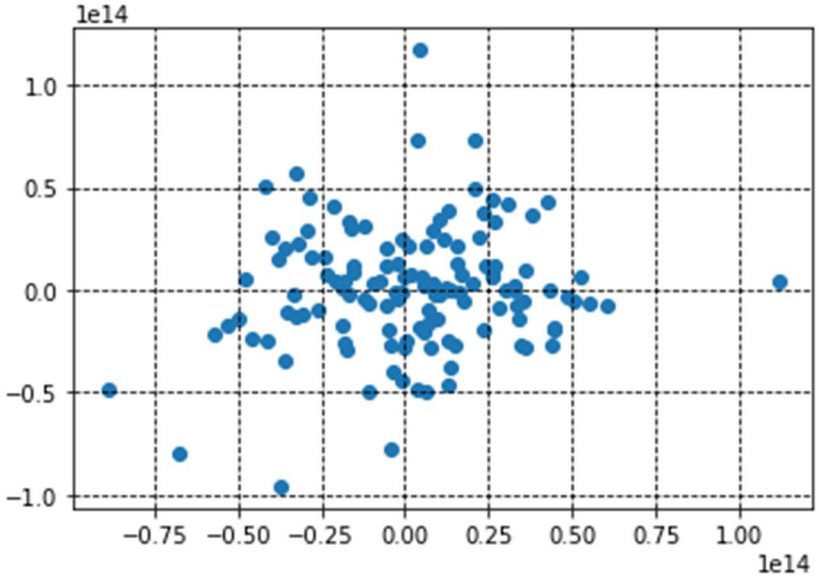


Figure 8.1

Each face is not exactly in front of the camera, there are many facial expressions (smiling, serious) and many photos do not show only the face of the person and this can have

consequence on the accuracy of the features' evaluation, used to compute the distances between pictures.

When a face contains noisy elements, the features are impacted and there may be variables that are further away from all the others. This is shown in the following graph.



Graph 8.1

The graph plots the variables used in the Face Recognition system, considering especially the first and the second row in the covariance matrix. The grid shows that most of the points are contained between -0.5 and 0.5 on the y-axis, and between -0.5 and 0.5 on the X-axis. There are many other points that are further away and can be possibly detected as outliers. For this analysis it is chosen to not remove the noisy faces, in order to see the effect of the outliers in the final results.

8.3 Binary Logistic Regression for Mahalanobis Distance

The next part is analogous to what is reported in the previous Chapter about the use of Euclidean distance.

The factors contained in the independent variable X are the Mahalanobis distances computed between two images, the 128 features of the image indexed with 'id1', the score of the first feature for the image indexed with id1, the score of the same feature for the image indexed with 'id2' and the difference between the last two values.

The encodings are split into training and test set that are respectively 70% and 30%.

The binary dependent variable y is built by assuming that if the difference score computed is lower than half the mean of all the differences in the data, then the two human faces encoded are inferred as reporting the same person.

The Logistic Regression model is built on 'X_train' and 'y_train'. The accuracy over the test set is 0.95, which is slightly higher than the accuracy score of the Binary Logistic Regression based on the data frame reporting the Euclidean distances.

The confusion matrix in this second case reports that 15.077 + 476 predictions are correct, whereas 555 + 6.251 are wrong. The classification report is the following:

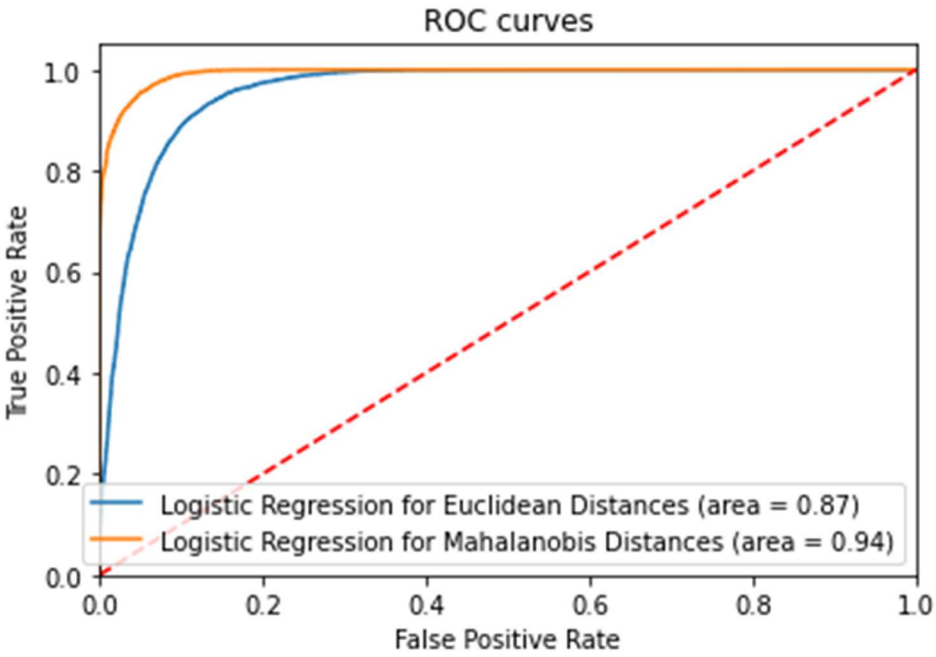
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.96	0.97	0.97	15553
1	0.93	0.92	0.92	6806
accuracy			0.95	22359
macro avg	0.95	0.94	0.95	22359
weighted avg	0.95	0.95	0.95	22359

Table 8.1

The precision is respectively 0.96 and 0.93 for class 0 and class 1. The recall measure shows a higher value in the identification of class 0, however both classes are found with high accuracy. Finally, the f1-score reflects the previous results, since once again the value for class 0 is slightly higher than the score reported for class 1. The overall accuracy of the classifier based on Mahalanobis distance is 0.95. The last two lines in the classification reports shows the macro average and the weighted average for each of the evaluation scores.

The ROC curve is then plotted, considering the ROC curve of the Binary Logistic Regression computed over the Euclidean Distances.



Graph 8.2

Both curves show a good performance for the classifiers, however it is evident that the Mahalanobis distance improves it.

There are many possible threshold values for the last part: the chosen one is 12.54878819. The result reports the couples of images whose distance is less than the value chosen. This is grouped by the identification number of the first image. A list of similar images is set for each image considered and since there is a list for all the images in the dataset, the lists with common elements are merged.

The result is composed by many sets with few elements and one set that includes many people, and it proves the degradation that the outliers bring to the Face Recognition System. However, it would be possible to make a further selection by comparing the images in the set. The set mentioned is the following.

Figure 8.2

It is evident that the set of images in Figure 8.2 shows many different people, and the variations in pose, illumination, facial expressions, and other elements are visible. However, this may be a good base where to start from and apply other techniques to prevent such combinations.

Concluding Remarks

This job of thesis aims specifically at studying the main legal and regulatory aspects, the impacts on individuals' privacy, the exercisable rights, prevention and mitigation measures to be applied by entities that use Face Recognition Technologies. The analytical part aims at proposing different models to identify pictures of the same person.

The study proposes a comparison between two distance measures, Euclidean and Mahalanobis, computed between vectors of face embeddings. It is possible to conclude that the presence of outliers causes a degradation when using the Mahalanobis distance. This is a very common issue Face Recognition Technologies have to deal with: in fact, as mentioned before, there are many factors that influence the performance of such technologies and each of them need to be consider when building models and algorithms to identify human faces. Surveillance Systems, for instance, are less likely to capture images or other type of inputs where the individual is positioned exactly in front of the camera lens, however, it is more likely to retrieve images from different angulations or where the person is in different poses, with or without glasses or with different facial expressions. All these factors are present in the so-called outliers.

The analysis revealed that the use Euclidean distance measure has a better performance compared to the results obtained through the use of Mahalanobis distance, by considering in both cases those images inferred as outliers, even if the classifier build on the basis of Mahalanobis distances has apparently a higher accuracy.

However, the results obtained on the basis of the Euclidean distances computation show the presence of false positives as well, even if in reduced form. This is very common in Face Recognition and, as reported in Paragraph 4.3, it is up to the developers of these technologies to decide which false/positive rate to choose. In particular, the choice consists in balancing the need of finding the highest number of images that lead to a certain person and the need of finding only images that show a specific individual. In the first case, it is more likely to incur in false positives, whereas in the second case, it is more probable to have false negatives. This choice may have possible impacts in case of investigations: the presence of false positives induces to wrong face-identity matches, whereas the presence of false negatives induces to not retrieve the identity of the individual the authority is looking for. In both cases the consequence may be to reach incorrect information.

For these reasons, it is necessary to strengthen the rights of the individuals in extreme cases and the entities that produce technologies where Face Recognition is used must always act

on the basis of Article 25, which underlines the importance of *“data protection through technology design”*.

Appendix

Abstract

The Appendix reports the scripts written in Python 3, in which all the models and methods proposed are developed with related comments.

Face verification with Euclidean Distance

```
import cv2
#Load the target image (Figure 5.1)
from skimage import io
img = io.imread('https://images.vanityfair.it/wpcontent/uploads/2021/02/
08190901/Jennifer-Aniston-950x684.jpg')

#Gray conversion
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

#Creation of histogram of the target image
Histogram = cv2.calcHist([img_gray], [0], None, [256], [0, 256])

#Plot the histogram
from matplotlib import pyplot as plt
plt.hist(img_gray.ravel(), 256, [0,256])

#Load first test image (Figure 5.2), convert to gray scale and plot the histogram
Test = io.imread('https://assets.afcdn.com/album/D20200207/
phalbm25895000_w980h638c1.jpg')
imgg_test = cv2.cvtColor(test, cv2.COLOR_BGR2GRAY)
histogram1 = cv2.calcHist([imgg_test], [0], None, [256], [0, 256])
plt.hist(imgg_test.ravel(),256,[0,256])

#Load second test image (Figure 5.3), convert to gray scale and plot the histogram
Image = io.imread('https://www.cinematographe.it/wpcontent/uploads/2020/12/
jennifer-lawrence-1.jpg')
gray_image2 = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

```
histogram2 = cv2.calcHist([gray_image2], [0],
                          None, [256], [0, 256])
plt.hist(gray_image2.ravel(),256,[0,256])


#Euclidean Distance between the target image (Figure 5.1) and first test image
#(Figure 5.2)
c1, c2 = 0, 0
i = 0
while i<len(histogram) and i<len(histogram1):
    c1 += (histogram[i]-histogram1[i])**2
    i += 1
c1 = c1**(1 / 2)

#Euclidean Distance between the target image (Figure 5.1) and sencond test image
#(Figure 5.3)
I = 0
while i<len(histogram) and i<len(histogram2):
    c2 += (histogram[i]-histogram2[i])**2
    i += 1
c2 = c2**(1 / 2)

if(c1<c2):
    print("data1.jpg is more similar to test.jpg as compare to data2.jpg")
else:
    print("data2.jpg is more similar to test.jpg as compare to data1.jpg")
```

Face verification with Mean Squared Error

```
#Definition of the Mean Squared Error function that is applied to the gray
#conversion of the two images considered

import numpy as np
def mse(imageA, imageB):
    err = np.sum((imageA.astype("float") - imageB.astype("float")) ** 2)
    err /= float(imageA.shape[0] * imageA.shape[1])
    return err
```

Data preparation for Face Identification and Face Clustering

```
#Import the needed libraries
import cv2
import pickle
import os

#Import of data from Kaggle
from google.colab import files
files.upload() #upload kaggle.json

!pip install -q kaggle
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!ls ~/.kaggle
!chmod 600 /root/.kaggle/kaggle.json

!kaggle datasets download -d danupnelson/14-celebrity-faces-dataset
!unzip /content/14-celebrity-faces-dataset.zip

#Organize data
import glob
data = glob.glob('/content/14-celebrity-faces-dataset/data/*/*/*.jpg')

#Research the identities of people in the dataset
Identities = []
for file in glob.glob('/content/14-celebrity-faces-dataset/data/*/*'):
        identity = os.path.splitext(os.path.basename(file))[0]
        identities.append(identity)

#Drop duplicates
list(set(identities))

#Read each image in the dataset through ‘skimage.io’
import numpy as np
from skimage import io
```

```
array = []
for j in range(len(data)):
  array.append(io.imread(data[j]))


#Read each image in the dataset through ‘openCV’
array_cv = []
for i in range(len(data)):
  array_cv.append(cv2.imread(data[i]))

#Convert to gray scale each image in the dataset read through ‘openCV’
gray = []
for i in range(len(array_cv)):
  gray.append(cv2.cvtColor(array_cv[i], cv2.COLOR_BGR2GRAY))

#Convert to rgb (red, green and blue) the images read through ‘skimage.io’
rgb = []
for i in range(len(array)):
  rgb.append(cv2.cvtColor(array[i], cv2.COLOR_BGR2RGB))

#Convert to rgb the images read through ‘openCV’
rgb_1 = []
for i in range(len(array_cv)):
  rgb_1.append(cv2.cvtColor(array_cv[i], cv2.COLOR_BGR2RGB))
```

Face Detection

```
#Creation of the cascades for Face Detection
import sys
faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_
frontalface_default.xml')
imagePath = sys.argv[1]

casc = []
for c in range(len(gray)):
  casc.append(faceCascade.detectMultiScale(gray[c], 1.3, 5))
```

```
#Open images in Google Colaboratory
from google.colab.patches import cv2_imshow


#Apply Face Detection to all images in the dataset
#Create rectangle around human faces detected
Images = []
filename = []
for i in range(len(casc)):
  for (x, y, w, h) in casc[i]:
      images.append(cv2.rectangle(array_cv[i],(x,y),(x+w,y+h),(255,0,0),2))
      filename.append(data[i])
#This function returns 4 values: the x and y location of the rectangle, and the
#rectangle’s width and height (w, h)

#Function that allows to see the image x in 'images'
for n in range(len(images)):
  def show(n):
    cv2_imshow(images[n])

#Perform Face Detection on one image
img = io.imread('/content/14-celebrity-faces-
dataset/data/train/kate_beckinsale/321px-Kate_Beckinsale_2006.jpg')
#Convert to gray scale
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = faceCascade.detectMultiScale(gray, 1.3, 5)

for (x, y, w,h) in faces:
    img_1 = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
```

Face Identification

```
#Install and import the needed packages
!pip install dlib
```

```
!pip install face_recognition

import dlib
import face_recognition
import numpy as np
#Extract the encodings (face embeddings) for each image in the dataset through
#‘face_recognition’ library
enc=[]
for i in range(len(rgb_1)):
  enc.append(face_recognition.face_encodings(rgb_1[i]))

encodings=[]
for i in range(len(enc)):
  encodings.append(np.array(enc[i]))

#Perform Face Identification for image 10 of the dataset
similar_faces=[]
for i in range(len(encodings)):
 face_recog=face_recognition.compare_faces(encodings[i], encodings[10])
 similar_faces.append(face_recog)

#Create data frame where for each human face in each picture it is reported a
#‘False’ in case the face does not correspond to the one in image 10, otherwise
#‘True’ is reported
import pandas as pd
df = pd.DataFrame(similar_faces)

s = df.squeeze()
#There are some images in which more than one face is identified

#List of faces similar to the one in image 10
list_similarfaces10 = [i for i, e in enumerate(s[0]) if e == [True]]

#How to open images
from PIL import Image
image_list = []
```

```
for filename in data:
  im=Image.open(filename)
  image_list.append(im)

sim_list= []
for i in list_similarfaces10:
  im=image_list[i]
  sim_list.append(im)

image = []
for i in list_similarfaces10:
   image.append(data[i])

#Create of set of images inferred as showing the same person as in image 10
images = [Image.open(x) for x in image]
widths, heights = zip(*(i.size for i in images))

total_width = sum(widths)
max_height = max(heights)

new_im = Image.new('RGB', (total_width, max_height))

x_offset = 0
for im in images:
  new_im.paste(im, (x_offset,0))
  x_offset += im.size[0]

#Set of images saved
new_im.save('test.jpg')

#Second example of Face Identification
similar_faces2 = []
for i in range(len(encodings)):
 face_recog = face_recognition.compare_faces(encodings[i], encodings[38])
 similar_faces2.append(face_recog)
```

```
df = pd.DataFrame(similar_faces2)
s2 = df.squeeze()

list_similarfaces = [i for i, e in enumerate(s2[0]) if e == [True]]

sim_list2= []
for i in list_similarfaces:
  im=image_list[i]
  sim_list2.append(im)

image2 = []
for i in list_similarfaces:
   image2.append(data[i])

#Create set of images inferred as showing the same person as in image 38
images = [Image.open(x) for x in image2]
widths, heights = zip(*(i.size for i in images))

total_width = sum(widths)
max_height = max(heights)

new_im = Image.new('RGB', (total_width, max_height))

x_offset = 0
for im in images:
  new_im.paste(im, (x_offset,0))
  x_offset += im.size[0]

#Set of images saved
new_im.save('test.jpg')
```

Face Clustering and Euclidean distance

```
#Create array of encodings for each face
enco = [y for y in encodings if 0 not in y.shape]
```

```
from sklearn.metrics.pairwise import euclidean_distances
#Example of distance computation between encodings of image 0 and encodings of
#image 1
euclidean_distances(enco[0], enco[1])

#List of encodings
list1 = []
for i in range(len(enco)):
  for n in range(len(enco[i])):
    e = enco[i][n]
    list1.append(e)

#In case the picture contains multiple human faces, only the first array of
#encodings is considered
list1 = pd.DataFrame([i[0] for i in enco])
```

Hierarchical Clustering

```
#Import needed packages for Hierarchical clustering
#Scale the data
from scipy.cluster.vq import whiten
scaled_data = whiten(list1.to_numpy())
pd.DataFrame(scaled_data).describe()
#The result shows that all the features' variance is set to 1

#Create dendrogram
from scipy.cluster.hierarchy import linkage
link = linkage(scaled_data, method = 'ward', metric = 'euclidean')

import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import dendrogram

dendrogram = dendrogram(link)

#Plot the dendroram
plt.axhline(c='red',linestyle='--', y=35)
```

```
#For y = 35, number of clusters=14
```

K-means Clustering

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=14)

kmeans.fit(list1)

#Retrieve the 14 clusters
clusters_kmeans = list(zip(kmeans.fit_predict(list1), data))

dataframe_kmeans = pd.DataFrame(clusters_kmeans)
cluster_id = dataframe_kmeans.groupby(0).agg(lambda x: list(x))
cluster_id.values[1]

image_kmeans = []
for i in cluster_id.index:
   image_kmeans.append(cluster_id.values[i])

image_kmeans[0][0]

#Set of images in Cluster 0
images = [Image.open(x) for x in image_kmeans[5][0]]
widths, heights = zip(*(i.size for i in images))

total_width = sum(widths)
max_height = max(heights)

new_im1 = Image.new('RGB', (total_width, max_height))

x_offset = 0
for im in images:
  new_im1.paste(im, (x_offset,0))
  x_offset += im.size[0]
```

```
new_im1.save('test1.jpg')

#Set of images in Cluster 1
images = [Image.open(x) for x in image_kmeans[1][0]]
widths, heights = zip(*(i.size for i in images))

total_width = sum(widths)
max_height = max(heights)

new_im2 = Image.new('RGB', (total_width, max_height))

x_offset = 0
for im in images:
  new_im2.paste(im, (x_offset,0))
  x_offset += im.size[0]

new_im2.save('test2.jpg')

#Data preparation for plots
#Graphs are both bidimensional and tridimensional
list1= np.array(list1)

clust=kmeans.fit_predict(list1)

#Tridimensional plot for clusters 3 and 14 (Graph 7.2, Paragraph 7.5)
%matplotlib inline
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(26,6))
ax = fig.add_subplot(131, projection='3d')
plt.scatter(list1[clust==2, 0], list1[clust==2, 1], s=100, c='red', label
='Cluster 3')
plt.scatter(list1[clust==13, 0], list1[clust==13, 1], s=100, c='blue', label
='Cluster 14')

#Bidimensional representation for clusters 3 and 14
```

```
%matplotlib inline
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(26,6))
ax = fig.add_subplot(131)
plt.scatter(list1[clust==2, 0], list1[clust==2, 1], s=100, c='red', label
='Cluster 3')
plt.scatter(list1[clust==13, 0], list1[clust==13, 1], s=100, c='blue', label
='Cluster 14')

#Tridimensional plot for clusters 1, 2, 3, 4 (Graph 7.3, Paragraph 7.5)
%matplotlib inline
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(26,6))
ax = fig.add_subplot(131, projection = '3d')
ax.scatter(list1[clust==0, 0], list1[clust==0, 1], s=100, c='red', label
='Cluster 1')
ax.scatter(list1[clust==1, 0], list1[clust==1, 1], s=100, c='blue', label
='Cluster 2')
ax.scatter(list1[clust==2, 0], list1[clust==2, 1], s=100, c='green', label
='Cluster 3')
ax.scatter(list1[clust==3, 0], list1[clust==3, 1], s=100, c='cyan', label
='Cluster 4')

#Tridimensional plot for clusters 5, 6, 7, 8, 9 (Graph 7.4, Paragraph 7.5)
%matplotlib inline
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(26,6))
ax = fig.add_subplot(131, projection='3d')
ax.scatter(list1[clust==4, 0], list1[clust==4, 1], s=100, c='magenta', label
='Cluster 5')
ax.scatter(list1[clust==5, 0], list1[clust==5, 1], s=100, c='black', label
='Cluster 6')
```

```
ax.scatter(list1[clust==6, 0], list1[clust==6, 1], s=100, c='yellow', label
='Cluster 7')
ax.scatter(list1[clust==7, 0], list1[clust==7, 1], s=100, c='lightblue', label
='Cluster 8')
ax.scatter(list1[clust==8, 0], list1[clust==8, 1], s=100, c='orange', label
='Cluster 9')
#Tridimensional plot for clusters 10, 11, 12, 13, 14 (Graph 7.5, Paragraph 7.5)
%matplotlib inline
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure(figsize=(26,6))
ax = fig.add_subplot(131, projection = '3d')
ax.scatter(list1[clust==9, 0], list1[clust==9, 1], s=100, c='purple', label
='Cluster 10')
ax.scatter(list1[clust==10, 0], list1[clust==10, 1], s=100, c='yellow', label
='Cluster 11')
ax.scatter(list1[clust==11, 0], list1[clust==11, 1], s=100, c='lightblue',
label ='Cluster 12')
ax.scatter(list1[clust==12, 0], list1[clust==12, 1], s=100, c='orange', label
='Cluster 13')
ax.scatter(list1[clust==13, 0], list1[clust==13, 1], s=100, c='darkblue', label
='Cluster 14')
```

Euclidean distances

```
#Compute Euclidean distances for each pair of images in the dataset
euclidean_dist = []
for i in range(len(enco)):
  euclid = euclidean_distances(enco[i], enco[i])
  euclidean_dist.append(euclid)

from scipy.spatial import distance
euclid = []
distances = []
for i in range(len(list1)):
  for e in range(len(list1)):
```

```
    dist = distance.euclidean(list1[i], list1[e])
    euclid.append([i, e, dist])
    distances.append(dist)

#Data frame where Euclidean distances for each couple of pictures are reported
df_euclid = pd.DataFrame(euclid, columns=['id1', 'id2', 'euclidean_distances'])

#Data Frame reporting for each human face all the 128 features
rows = pd.DataFrame(list1)

#Merge the data frame in order to have ‘id1’, ‘id2’, the Euclidean distance and
#all the 128 features of the image identified with ‘id1’
merge = (pd.merge(left=df_euclid, right=rows[0], left_on='id1',
right_on=rows.index))

merge2 = (pd.merge(left=merge, right=rows, left_on='id2', right_on=rows.index))

#Get the Difference between the value of the first feature of the image identified
#with ‘id1’ and the same value for image identified with ‘id2’
merge2['Score_diff'] = abs(merge2['0_x'] - merge2['0_y'])

merge2['Score_diff'].mean()/2

#Binary Logistic Regression for Euclidean Distances
#Creation of the dependent variable for Binary Logistic Regression
merge2['assumption'] = np.where((merge2['Score_diff']<merge2['Score_diff']
.mean()/2), 1, 0)

y = merge2['assumption']

#Independent variable
X = merge2.drop('assumption', axis='columns')

#Array of Euclidean distances
array = np.array(merge2['euclidean_distances'])
```

```
#Fit the model
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(X_train, y_train)

y_pred = logreg.predict(X_test)
#Evaluate the performance
print('Accuracy of logistic regression classifier on test set:
{:.2f}'.format(logreg.score(X_test, y_test)))

from sklearn.metrics import roc_curve
#Research of the optimal threshold between the ones proposed by the following
#function (ROC curve)
fpr1, tpr1, thresholds1 = roc_curve(y, array)

#Confusion matrix (Paragraph 7.7)
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)

#Classification report (Table 7.3, Paragraph 7.7)
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))

#Plot the ROC curve for the classifier built on the basis of Euclidean distances
#(Graph 7.6, Paragraph 7.7)
from sklearn.metrics import roc_auc_score
logit_roc_auc1 = roc_auc_score(y_test, logreg.predict(X_test))
fpr1, tpr1, thresholds1 = roc_curve(y_test, logreg.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr1, tpr1, label='Logistic Regression (area = %0.2f)' %
logit_roc_auc1)
```

```
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
plt.show()

df_id = X.groupby('id1').agg(lambda x: list(x))

#For each image extract the images distant for less than 0.4013106
lists = []
for i in range(len(df_id['euclidean_distances'])):
  d = df_id['euclidean_distances'][i]
  list_=[index for index,value in enumerate(d) if value <= 0.4013106]
  lists.append(list_)

l=lists

#Merge lists with common images
out = []
while len(l)>0:
    first, *rest = l
    first = set(first)

    lf = -1
    while len(first)>lf:
        lf = len(first)

        rest2 = []
        for r in rest:
            if len(first.intersection(set(r)))>0:
                first |= set(r)
            else:
```

```
                rest2.append(r)
        rest = rest2

    out.append(first)
    l = rest
print(out)
#Remove lists with less than 2 images
for i in out:
    if(len(i) <= 2):
        out.remove(i)

v1 = []
for v in out:
    if v != []:
        v1.append(v)

#Example of grouping based on Euclidean distance (Figure 7.3, Paragraph 7.4)
x = []
for i in v1[12]:
  im=data[i]
  x.append(im)

images = [Image.open(x) for x in x]
widths, heights = zip(*(i.size for i in images))

total_width = sum(widths)
max_height = max(heights)

new_im4 = Image.new('RGB', (total_width, max_height))

x_offset = 0
for im in images:
  new_im4.paste(im, (x_offset,0))
  x_offset += im.size[0]

new_im4.save('test4.jpg')
```

```
#Second example of grouping based on Euclidean Distance (Figure 7.4, Paragraph
#7.7)
image = []
for i in v1[12]: #4
    image.append(data[i])

images = [Image.open(x) for x in image]
widths, heights = zip(*(i.size for i in images))

total_width = sum(widths)
max_height = max(heights)

new_im5 = Image.new('RGB', (total_width, max_height))

x_offset = 0
for im in images:
  new_im5.paste(im, (x_offset,0))
  x_offset += im.size[0]

new_im5.save('test5.jpg')
```

Mahalanobis distance

```
enco_ = [i[0] for i in enco]

from scipy.spatial import distance

cov = np.cov(enco_, rowvar=False)
inv = np.linalg.inv(cov)

#Compute Mahalanobis using ‘scipy’ package distance for each pair of images in
#the dataset
mahal = []
mahalanobis_dist = []
for i in range(len(list1)):
```

```
  for e in range(len(list1)):
    dist = distance.mahalanobis(list1[i], list1[e], inv)
    mahal.append([i, e, dist])
    mahalanobis_dist.append(dist)


#Create inverse of the covariance matrix needed for the manual computation of
#Mahalanobis distance
df= pd.DataFrame(inv)
centerpoint = np.mean(enco_, axis=0)
#Research of the outliers
distances = []
for i, val in enumerate(enco_):
      x = val
      m = centerpoint
      distance = (x-m).T.dot(df).dot(x-m)
      distances.append(distance)
distances = np.array(distances)

cutoff=distances.mean()

#Index of outliers
outlierIndexes = np.where(distances > cutoff)

print('Index of Outliers')
print(outlierIndexes)

print('Observations found as outlier')
print(df[ distances > cutoff])


#Plot outliers (Graph 8.1, Paragraph 8.2)
%matplotlib inline
fig = plt.figure()
ax = plt.subplot()
ax.scatter(df[0], df[1])
t = np.linspace(0, 2*pi, 100)
plt.grid(color='black',linestyle='--')
plt.show()


df = pd.DataFrame(mahal, columns=['id1', 'id2', 'mahalanobis_distances'])


merge = (pd.merge(left=df, right=rows[0], left_on='id1', right_on=rows.index))
```

```
merge_mahal = (pd.merge(left=merge, right=rows, left_on='id2',
right_on=rows.index))

#Create new variable, as done in case of Euclidean distances computation
merge_mahal['Score_diff'] = abs(merge_mahal['0_x'] - merge_mahal['0_y'])

#Build the dependent variable y
merge_mahal['assumption'] = np.where((merge_mahal['Score_diff'] < merge_mahal
['Score_diff'].mean()/2), 1, 0)

y = merge_mahal['assumption']

#Independent variable
X = merge_mahal.drop('assumption',axis='columns')

#Array of Mahalanobis distances
array = np.array(X['mahalanobis_distances'])

#Fit the model
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(X_train, y_train)

pred = logreg.predict(X)

y_pred = logreg.predict(X_test)

#Evaluate accuracy of classifier built on the basis of Mahalanobis distance
#computation
print('Accuracy of logistic regression classifier on test set:
{:.2f}'.format(logreg.score(X_test, y_test)))
```

```
#Reasearch the optimal threshold
from sklearn.metrics import roc_curve
fpr, tpr, thresholds = roc_curve(y, array)


#Confusion matrix
from sklearn.metrics import confusion_matrix
confusion_matrix = confusion_matrix(y_test, y_pred)
print(confusion_matrix)

#Classification report (Table 8.1)
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))

#Plot ROC curves for both Euclidean distances and Mahalanobis distances (Graph
#8.2, Paragraph 8.3)
import matplotlib.pyplot as plt
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(y_test, logreg.predict(X_test))
fpr, tpr, thresholds = roc_curve(y_test, logreg.predict_proba(X_test)[:,1])
plt.figure()
plt.plot(fpr1, tpr1, label='Logistic Regression for Euclidean Distances (area =
%0.2f)' % logit_roc_auc1)
plt.plot(fpr, tpr, label='Logistic Regression for Mahalanobis Distances (area =
%0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC curves')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
plt.show()
```

```
#For each image extract the images distant for less than 12.67074448
lists_ = []
for i in range(len(df_id_mahal['mahalanobis_distances'])):
  d = df_id_mahal['mahalanobis_distances'][i]
  list_= [index for index,value in enumerate(d) if value <= 12.67074448]
  lists_.append(list_)

l = lists_
out = []
while len(l)>0:
    first, *rest = l
    first = set(first)

    lf = -1
    while len(first)>lf:
        lf = len(first)

        rest2 = []
        for r in rest:
            if len(first.intersection(set(r)))>0:
                first |= set(r)
            else:
                rest2.append(r)
        rest = rest2

    out.append(first)
    l = rest

for i in out:
    if(len(i) == 1):
        out.remove(i)

v2 = []
for v in out:
    if v != []:
        v2.append(v)
```

```
#Example of grouping based on Mahalanobis distance (Figure 8.2, Paragraph 8.3)
x = []
for i in v2[3]:
  im=data[i]
  x.append(im)

images = [Image.open(x) for x in x]
widths, heights = zip(*(i.size for i in images))

total_width = sum(widths)
max_height = max(heights)

new_im6 = Image.new('RGB', (total_width, max_height))

x_offset = 0
for im in images:
  new_im6.paste(im, (x_offset,0))
  x_offset += im.size[0]

new_im6.save('test6.jpg')
```

Bibliography

https://machinelearningmastery.com/introduction-to-deep-learning-for-face-recognition/

https://www.computer.org/csdl/pds/api/csdl/proceedings/download-article/12OmNAHmOrk/pdf

https://arxiv.org/pdf/1803.11556.pdf

https://uk.practicallaw.thomsonreuters.com/Glossary/UKPracticalLaw/Ibcdae8145a0c11e89bf199c0ee06c731?transitionType=Default&contextData=(sc.Default)&firstPage=true

https://www.spglobal.com/marketintelligence/en/news-insights/latest-news-headlines/facial-recognition-tech-tests-the-limits-of-europe-s-data-privacy-laws-59476415

GDPR

https://trucchifacebook.com/facebook/guida/foto-eliminate-cancellate-su-facebook-sono-accessibili/

https://search.coe.int/cm/Pages/result_details.aspx?ObjectId=09000016807c65bf

https://www.cnil.fr/sites/default/files/atoms/files/facial-recognition.pdf

https://www.safetysecuritymagazine.com/articoli/facial-recognition-system-aspetti-tecnici-e-di-privacy/

https://www.facefirst.com/blog/brief-history-of-face-recognition-software/

https://towardsdatascience.com/facial-recognition-types-of-attacks-and-anti-spoofing-techniques-9d732080f91e#:~:text=Attacking%20Methods&text=Typically%2C%20face%20recognition%20systems%20can,use%20makeup%20or%20plastic%20surgery.

https://www.forbes.com/sites/forbestechcouncil/2020/06/23/facial-recognition-systems-security/?sh=294e727d2324

https://rm.coe.int/guidelines-on-facial-recognition/1680a134f3

https://www.machinelearningplus.com/statistics/mahalanobis-distance/

https://arsfutura.com/magazine/face-recognition-with-facenet-and-mtcnn/

https://onezero.medium.com/those-covid-19-temperature-scanning-kiosks-use-scary-powerful-facial-recognition-8cc8ada0c595

https://manivannan-ai.medium.com/face-encodings-from-image-e6015b07bd91

https://stackabuse.com/autoencoders-for-image-reconstruction-in-python-and-keras/

https://arxiv.org/abs/1512.03385

https://www.idiap.ch/~marcel/labs/faceverif.php#:~:text=Face%20recognition%2C%20verification%20and%20identification,verification%20(also%20called%20authenticationù

How to Extract Image Metadata in Python - Python Code

S&#xFFFD;bastien Marcel - Lab: Face Verification / Face Authentication

https://www.vidiemme.it/riconoscimento-facciale/

sklearn.metrics.roc_curve — scikit-learn 0.24.1 documentation

Face clustering with Python - PyImageSearch

Face recognition with OpenCV, Python, and deep learning - PyImageSearch

Research on face feature extraction based on K-mean algorithm | EURASIP Journal on Image and Video Processing | Full Text

In the blink of AI: How facial recognition technology is capitalising on the COVID-19 crisis | View | Euronews

Book Title: Face Recognition Technology

Book Subtitle: Compulsory Visibility and Its Impact on Privacy and the Confidentiality of Personal Identifiable Images

Authors: Ian Berle

Book Title: Handbook of Face Recognition,

Editors: Li, Stan Z., Jain, Anil K.

Book Title: An Introduction to Statistical Learning

Authors: Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Book Title: