AGH
Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
Wydział Fizyki i Informatyki Stosowanej

Master thesis

Zofia Pieńkowska

Major: **Medical Physics**

Specialisation: **Imaging techniques and biometry**

Impact of childhood photos on face recognition classification accuracy

Supervisor: **dr inż. Tomasz Pięciak**

Cracow, September 2020

Plagiarism statement

Aware of criminal liability for making untrue statements, I declare that the following thesis was written personally by myself and that I did not use any sources but the ones mentioned in the dissertation itself.

....

(legible signature)

Składam serdeczne podziękowania opiekunowi naukowemu pracy,
Panu dr. inż. Tomaszowi Pięciakowi, za poświęcony czas oraz
wskazówki udzielone podczas powstawania niniejszej pracy.

Dziękuję również Pani dr inż. Joannie Świebockiej-Więk
za wprowadzenie mnie w świat biometrii oraz
ogromny wkład w powstanie niniejszej pracy.

Kraków 2020

Merytoryczna ocena pracy przez promotora:

Końcowa ocena pracy przez promotora: ....
Data: .... Podpis: ....

Skala ocen: 5.0 - bardzo dobra, 4.5 - plus dobra, 4.0 - dobra, 3.5 - plus dostateczna, 3.0 - dostateczna, 2.0 - niedostateczna

Merytoryczna ocena pracy przez recenzenta:

Końcowa ocena pracy przez recenzenta: ....
Data: .... Podpis: ....

Skala ocen: 5.0 - bardzo dobra, 4.5 - plus dobra, 4.0 - dobra, 3.5 - plus dostateczna, 3.0 - dostateczna, 2.0 - niedostateczna

Contents

List of abbreviations

AI	–	Artificial Intelligence
CEO	–	Chief Executive Officer
CIA	–	Central Intelligence Agency
FNN	–	Feedforward Neural Network
KNN	–	k-Nearest Neighbours
LBP	–	Local Binary Patterns
LDA	–	Linear Discriminant Analysis
NSA	–	National Security Agency
PCA	–	Principal Component Analysis
SVM	–	Support Vector Machine
USA	–	United States of America

Chapter 1

Introduction

1.1 Introduction

Face recognition is a process of recognizing or verifying the identity of a person by analyzing the features extracted from the image of its face. It is currently considered the most natural and convenient biometrical technique, and is already visible in many areas of life, inevitably causing controversies over its applications and privacy issues [1].

The concept of automated face recognition dates back to 1960s, when it was still a new and highly experimental field of science [2]. But as its popularity was rising and ideas for its applications were accumulating, it became a fascinating new trend; while sci-fi culture was still picturing face recognition as a technology from the future, tech giants were implementing its algorithms both for surveillance and personalization of services [3]. Now, face recognition is as widely available as it is controversial.

Nowadays, face recognition is present in security systems, public surveillance, access control and social applications. It is used to verify the identity of people crossing the border or boarding a plane, to verify whether known criminals try to enter a mass event, to unlock systems containing sensitive data, and to automatically tag our friends on the photos posted on social networks [4]. While public security has its own share of controversies over the use of biometry, companies like Facebook and Google have invented exceptionally effective algorithms for facial recognition [5, 6]. They are often criticized for not being fully transparent about the use of those algorithms and the huge collection of faces labeled with names that they own.

1.2 Motivation of this thesis

The main objective of this thesis is to examine whether adding childhood photos to training datasets of machine learning algorithms can affect their accuracy when recognizing people present on the photos. Additionally, the effectiveness of recognizing adult people by algorithms trained using only their childhood photos is measured.

This master thesis is divided into two parts: **background** and **contributions**. The first part concerns historical background of face recognition, discussion of its ethical controversies and currently used algorithms. Both historical and state-of-the-art methods are addressed, along with the mathematics behind them and their applications. The second part presents the results of the research conducted using various classification methods, with the aim of determining whether the presence of childhood photos in the learning set could affect the face recognition accuracy. Additionally, a learning set consisting only of childhood photos has been constructed, which allows to determine the accuracy of recognizing adults after training the algorithm solely on childhood photos.

Chapter 2

Face recognition – theoretical background and controversies

2.1 Introduction

This chapter covers historical background of face recognition, its fundamentals, currently used algorithms and surrounding controversies.

Firstly, the history of face recognition is summarized. The evolution of approach and techniques used is described to provide the background of this area of technology, its milestones and difficulties met. Next, the basic concepts and universal principles of face recognition systems are discussed. A few algorithms of feature extraction and classification are addressed, and their mathematical grounds are provided to a small extent. Although not all of those methods have been implemented in the research, many approaches are described to present the variety of ways, similarities and differences between them, and their interrelations. Finally, controversies arising from their applications are discussed.

2.2 Historical background

The first accomplishments in the field of face recognition are dated to 1964 and attributed to Woodrow Bledsoe, Charles Bisson and Helen Chan Wolf, who then worked at Panoramic Research, Inc. [7] Using a mug book as a dataset, they aimed to automatize the process of searching the book for a particular face, which was depicted on another photo. Little of the results has been published due to private funding of this research (some sources claim CIA to be the founder [8]), but it is known that they relied on a form of feature extraction – mainly the coordinates of landmarks such as pupils or outer corners of the eyes. Bledsoe described some of the difficulties encountered, e.g. aging or variability of face angle, lightning conditions and facial expression – the same obstacles that bother researchers to this day [2, 7].

After Bledsoe left Panoramic in 1966 [2], the work has been continued at Stanford Research Institute by Peter Hart in collaboration with Bledsoe. Their reports mention slight modifications to the algorithms, e.g. using Bayes decision theory, and overall Hart achieved considerable success compared to Bledsoe [7, 8].

In 1971, A. Jay Goldstein, Leon Harmon and Ann Lesk conducted an experiment in which human jurors rated individual face features on a scale from 1 to 3 (minimum) or 5 (maximum). Goldstein et al. used the word “vector” to describe a set of such features ratings for a particular face – a term that would soon become crucial in automated face recognition.

They used Euclidean distance to compare the faces with each other, and concluded that such a model can be useful in reducing the number of similar faces to choose from [9].

A hypothesis has been brought up that the features useful in automated recognition are the same features that human brain looks for while recognising faces; however a possibility has been allowed for that this might not be the case. “It is of interest but not of immediate concern whether humans, either consciously or unconsciously, use »features« like ear length, hair texture, and lip thickness in recognizing a face”, they wrote [9].

What is claimed to be the next breaking point was using PCA on face images by Lawrence Sirovich and Michael Kirby in 1986 [10]. They were already viewing the photo of the face as a matrix, elements of which marked various levels of grayscale, or a vector, constructed by concatenating the rows of the matrix. By applying PCA, they extracted what they called *eigenpictures* – matrices that could be linearly combined to replicate original images.

Sirovich and Kirby only proposed a theoretical approach, not being sure whether it might have practical applications; “we offer no experimental procedure for verifying or refuting that our method bears in any way on our faculties for face recognition”, they wrote [10]. However, in 1991, Matthew Turk and Alex Pentland improved this algorithm and presented a fully functioning system for face detection and recognition [11]. Their research also allowed for quicker image decomposion, thus being the first step to face recognition being fully automated and used *en masse*.

Then, throughout the years, automated face recognition kept expanding into further areas of life, such as surveillance systems, advertising, but also personal access control to smartphones, banking applications or other sensitive data holders. Those companies which were in possession of huge amounts of their users’ photos developed exceptionally effective algorithms, most notable being DeepFace by Facebook with its 97.35% accuracy [6] and FaceNet by Google with 99,3% accuracy [5].

The development of face recognition has raised many controversies. Of course, similar technologies can be, and are, used to improve the quality of our lives. They contribute to improving the safety of various systems, and allow for evaluation whether our photo has been posted online without our consent; they can also help disabled people in unexpected ways – like a smartphone application by Listerine which lets blind people know when others smile at them [12]. However, many privacy-oriented organizations have been pointing out the risks for many years, and the term “face recognition” is nowadays often associated with invigilation and loss of privacy [13].

2.3 Overview of face recognition stages

The process of face recognition consists of two stages: feature extraction and classification. Every image can be represented as a vector, each of its elements corresponding to the following pixel. A naïve way to compare the vectors representing two images would be to calculate the Euclidean distance between them; however, the number of elements in such vector can be enormous. Also, image details – background, the person wearing sunglasses, or having their hair painted – would disrupt the comparison greatly. For this reason, feature extraction is an essential step in face recognition; it aims both at extracting distinctive features of the face and reducing the vector dimensionality. Achieving the first goal increases the accuracy of recognition, while the second makes the computations more memory-efficient.

In face recognition, supervised machine learning is usually used. This means the data is **labeled**: every training sample has a label assigned, stating which class it belongs to. In supervised machine learning, unknown samples can be classified on the basis of their similarity to known specimens.

After representing the face as a feature vector, it needs to be classified to an appropriate category – the person it belongs to; this means that similarity of some kind must be stated between the vector under classification and a known vector labeled with a person's name.

The following section provides an overview of currently used feature extraction algorithms and classifiers.

2.4 State-of-the-art

2.4.1 Feature extraction

Eigenfaces (PCA)

PCA proves useful in feature extraction task. It aims to maximize the variance between high-dimensional data, thus facilitating the classification. Minimization of dimensionality is achieved by calculating so called **eigenfaces** – images that, when combined linearly, closely approximate the images from the initial dataset. The images are represented as vectors, created by concatenation of rows of pixels. Let M face images in the training dataset be $\mathbf{\Gamma}_1, \mathbf{\Gamma}_2, ..., \mathbf{\Gamma}_M$. An eigenface is calculated as follows [11]:

1. An average face $\mathbf{\Psi}$ is calculated:

$$\mathbf{\Psi} = \frac{1}{M} \sum_{n=1}^{M} \mathbf{\Gamma}_n. \tag{2.1}$$

2. The average face is subtracted from each of the faces in the training dataset. Every face differs from the average face by vector $\mathbf{\Phi}$:

$$\mathbf{\Phi}_{\boldsymbol{n}} = \mathbf{\Gamma}_n - \mathbf{\Psi}. \tag{2.2}$$

3. PCA is performed over the set of vectors $\{\mathbf{\Phi}_1, \mathbf{\Phi}_2, ..., \mathbf{\Phi}_M\}$. It aims to calculate the eigenvectors and corresponding eigenvalues Λ of the covariance matrix $\mathbf{C}$:

$$\mathbf{C} = \frac{1}{M} \sum_{n=1}^{M} \mathbf{\Phi}_n \mathbf{\Phi}_n^T. \tag{2.3}$$

Should the eigenvectors be calculated over the whole training dataset, each of the original face vectors can be represented as their linear combination. These eigenvectors have the same size as original images, so they can be viewed as images (eigenfaces) themselves. From all the eigenfaces generated, k vectors with the greatest variance are chosen. Many approaches to calculating k exist, but one of the most popular include assuming a treshold of variance ε. The eigenvalues are arranged in descending order $\Lambda_1, \Lambda_2, ..., \Lambda_n$. Then, k is calculated as the smallest number to satisfy the equation [14]:

$$\frac{\Lambda_1 + \Lambda_2 + ... + \Lambda_k}{\Lambda_1 + \Lambda_2 + ... + \Lambda_n} > \varepsilon, \qquad k < n. \tag{2.4}$$

The more eigenfaces are generated, the smaller is the reconstruction error; however, a large number of eigenfaces results in higher dimensionality of the data and, consequentially, longer computation time.

PCA is an unsupervised learning technique, and as such it does not take any classes into consideration; it only aims to maximize the variance between all the data [15, 16].

Fisherfaces (LDA)

The LDA algorithm has been invented by Sir R. A. Fisher – hence the name "fisherfaces", in reference to Eigenfaces – and implemented in face recognition by Peter Belhumeur, João Hespanha, and David Kriegman in 1997 [18]. It is a supervised learning method and extends PCA, maximizing the between-class scatter to within-class scatter ratio. PCA only maximized the variance scatter between all the data, which yields especially bad results if the background, light of facial expression is not homogenous within the dataset. LDA allows preserving some valuable data that would otherwise be lost, and thanks to that performs well in the presence of light and facial expression variations [17, 18].

If we consider a set of N images $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ and assume that each image belongs to one of c classes $\{X_1, X_2, ..., X_c\}$, the between-class scatter S_B can be defined as:

$$S_B = \sum_{i=1}^{c} N_i(\boldsymbol{\mu}_i - \boldsymbol{\mu}u)(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \tag{2.5}$$

and the within-class scatter S_W is defined as:

$$S_W = \sum_{i=1}^{c} \sum_{\mathbf{x}_k \in X_i} (\mathbf{x}_k - \boldsymbol{\mu}_i)(\mathbf{x}_k - \boldsymbol{\mu}_i)^T, \tag{2.6}$$
$$k = 1, 2, ..., N,$$

where:

- $\boldsymbol{\mu}_i$ is the mean image of class X_i,
- $\boldsymbol{\mu}$ is the mean image of all samples,
- N_i is the number of samples in class X_i.

By maximizing the between-class scatter and minimizing the within class scatter, Fisherfaces method facilitates face recognition under varying light conditions and face expression, which remains its main advantage. However, the algorithm performs worse in presence of extreme lightning conditions, or when shadowing dominates the image [18].

LBP

The LBP method has been introduced in 2004 by T. Ahonen, A. Hadid and M. Pietikainen [19]. The algorithm is claimed by its authors to be both fast and efficient. It is all the more promising since it is not just another statistical method like PCA or LDA, but rather a texture recognition system, and its authors had face recognition in mind from the very beginning of their work. The calculations are performed as follows:

1. The image is divided into smaller chunks, usually cells of $n \times n$ pixels;
2. The value of each pixel in the cell is compared to the value its eight nearest neighbours, clockwise or counter-clockwise. If the neighbour is darker than the center pixel, value of 1 is assigned to it, and 0 otherwise. This creates a binary number of eight digits, which can be converted to a decimal number in the range of 0 to 255;
3. A histogram of the decimal numbers is calculated, which results in a 256-element vector. Its elements are the frequencies of value combinations around the center pixel;
4. The histograms of all the cells in the picture are concatenated, resulting in a feature vector of the image.

Such a histogram contains information about the image micropatterns of the whole image. Such vectors can reach significant sizes, which is their main disadvantage, but despite that they can be successfully used for further classification, for example by SVMs, neural networks or other machine learning techniques [19].

FaceNet

FaceNet is an algorithm developed by Google researchers in 2015; it uses a convolutional neural network to provide optimal mapping of the faces in the dataset [5]. It maps the images onto an Euclidean space, and its output is a 128-element floating numbers vector called an embedding. The embeddings can be used as feature vectors compared with each other using Euclidean distance; however, the use of convolutional neural network allows the mapping rather to be learnt from the images than calculated by a static algorithm.

FaceNet implements triplet loss function; its key feature is taking three images as an input, namely “anchor”, “positive” and “negative”, where the “positive” image should depict the same object (or face) as “anchor”, and “negative” should picture some other. The neural network aims to minimize the Euclidean distance between the embeddings of the same person’s photos, and maximize the distance if the photos depict two different people. The triplet loss function can be defined as follows:

$$\mathcal{L}(\mathbf{A}, \mathbf{P}, \mathbf{N}) = ||f(\mathbf{A}) - f(\mathbf{P})||^2 - ||f(\mathbf{A}) - f(\mathbf{N})||^2 + \alpha, \tag{2.7}$$

where:

- $||\mathbf{q} - \mathbf{p}||$ – ℓ_2 norm;
- $\mathbf{A}$ – anchor image;
- $\mathbf{P}$ – positive image;
- $\mathbf{N}$ – negative image;
- f – the embedding of an image;
- α – set margin between positive and negative elements.

The cost function is calculated as a sum of loss function values over all n triplets:

$$\mathcal{J} = \sum_{i=1}^{n} \mathcal{L}(\mathbf{A}_i, \mathbf{P}_i, \mathbf{N}_i). \tag{2.8}$$

The cost function is then minimized in order to achieve optimal classification results.

2.4.2 Classification

Euclidean distance classifier [20, 21]

The Euclidean distance simply measures how far two n-dimensional vectors are from each other in an n-dimensional space. The distance is calculated as follows:

$$d = \sum_{i=1}^{n} (q_i - p_i)^2, \tag{2.9}$$

where $\mathbf{q}$ and $\mathbf{p}$ are the vectors in question, and q_i and p_i are their subsequent components.

Despite being possibly the simplest comparison between two feature vectors, this classifier is successfully used as a part of more advanced face recognition algorithms. Its advantages include being computationally simple and intuitive, but in its basic form it can only compare one vector to another one by one. It has no capability to learn nor is it memory-efficient. However, the Euclidean distance is used in more complex classifiers, such as KNN or SVM, which are described below.

KNN [22, 23]

The KNN method uses the Euclidean distance classifier, but is more resistant towards outliers. The goal of the algorithm is to find k vectors located nearest to the sample in

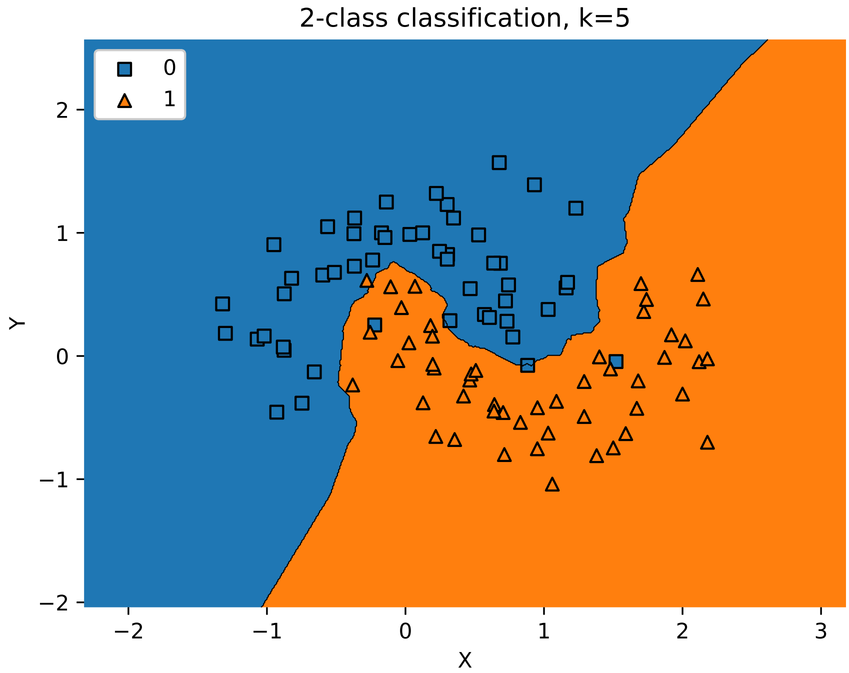

Figure 2.1: Graphical representation of KNN classification algorithm with $k = 5$ on example data. The points represent training data, and the colored planes represent areas of belonging to various classes. When a sample is to be classified, it is assigned to the class most of its 5 nearest neighbours belong to.

question. The sample is then assigned to the most popular class in this k-element set. The process has been illustrated in figure 2.1.

The key phase of the algorithm is finding optimal value of parameter k. If its value is too small, the classification might be disrupted by the presence of one class representants in another class' clusters, so called outliers. If the sample in question is located too close to such an anomal element, it might be classified to its subset despite belonging to another. On the other hand, too big value of k does not allow to reflect the complexity of the feature space. This can lead to ignoring smaller data clusters and, subsequently, to erratic classification.

The KNN algorithm relies on calculating the distances between the element under classification and every other vector, and data analysis in the next step. This makes such an implementation rather computationally demanding and memory-uneffcient, especially for large datasets. To remedy this issue, the data space is usually divided into smaller parts which are indexed in some way. Then, each sample is complemented with the index of the part it belongs to; when the distances are computed, only the elements which belong to the same part as the sample under classification are taken into consideration. Example methods using this approach include **bucketing** or ***k*-dimensional trees.**

SVM

The SVM algorithm was invented by Vladimir Vapnik and his colleagues in 1997 [25]. The input data is represented as n-dimensional vectors, which are then separated with a $(n-1)$-dimensional hyperplane. The algorithm chooses such a hyperplane that its distance to the nearest data vectors is maximized.

SVM works best if the input data can be easily classified into subsets which are distant from each other. Such datasets are sometimes called *extreme*. The more data is placed

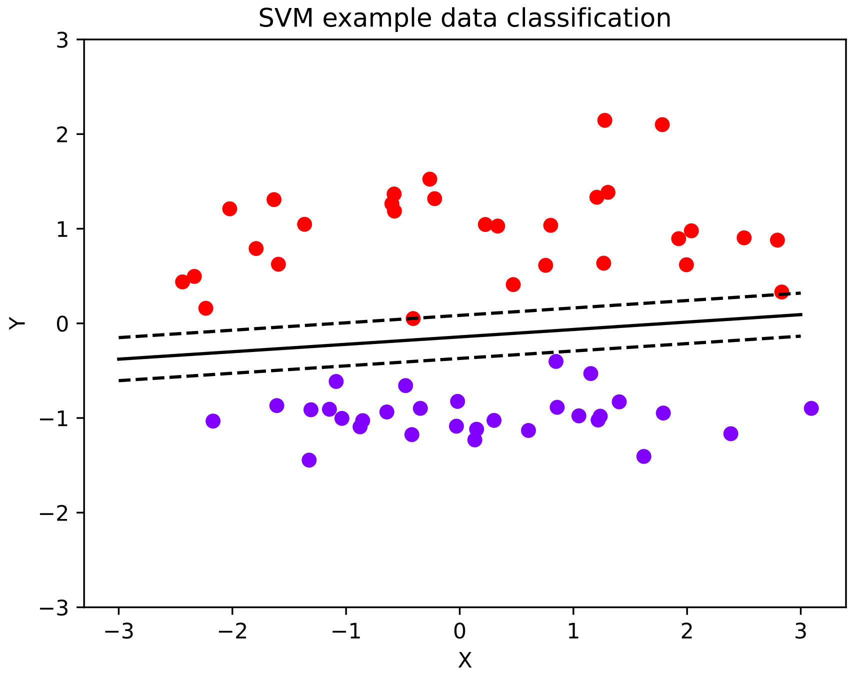

Figure 2.2: Example data separated by SVM. Solid line is the separation boundary, and dotted lines are the support vectors.

somewhere between the subsets clusters, the harder is the task of finding the separating hyperplane.

The title *support vectors* are the data which lie closest to the hyperplane; only these vectors are a part of the computations. The others are ignored, as they do not have any impact on the final hyperplane equation. This increases the memory efficiency of the SVM algorithm.

If the data cannot be easily separated with a hyperplane, it is transformed to a higher dimension space. The transformations should make the data separable by a linear function. Finding a proper transform function is a non-trivial and computationally intensive task; there are, however, numerous kernel functions (also known as kernel tricks) that can often be used to obtain good results. An example of projecting the data onto a higher dimension space to separate it is illustrated in fig. 2.2.

Advantages of SVM include the aforementioned memory-efficiency and possibility to choose between various kernel functions (or to define custom ones). What is more, SVM obtains good classification results in cases when the space dimension is greater than the number of samples.

It might not be the best choice, however, if the number of data features extends the number of samples. Another major disadvantage of SVM is that the algorithm gives no information about probability of a sample belonging to a given dataset. Such information can only be obtained by using **k-fold cross check**.

Applications of SVM include image classification in medicine and web pages ranking in web search engines.

FNN

An FNN is the simplest case of a neural network [26]. It is a supervised machine learning algorithm, so for every input training data $\mathbf{x}$ there exists a true label y. The main goal of

FNN is to estimate a function $f(\mathbf{x})$, such that $f(\mathbf{x}) = y$ for as many testing data as possible. The function f has some parameters, which are randomly assigned at first; then, the function is run over the test dataset, and its accuracy is checked, defined as the percent of correctly assigned labels. Parameters are then modified in each iteration in such a way that maximizes the function accuracy – this process is called **learning**.

The error rate of a neural network is calculated by a **loss function**. It quantifies the difference between the real labels and the predicted ones, and presents it as a number; the precise algorithm to calculate this number depends on the loss function used, but a general rule is that the smaller the loss function value, the greater the accuracy of the neural network.

The parameters having the most impact on the neural network accuracy – thus, the parameters worth changing in the next iterations – are found by the means of an **optimizer**. The optimizer uses gradient values of the loss function with respect to the function parameters, which allows for estimating how the parameters should be modified to achieve the highest accuracy.

The resulting function $f(\mathbf{x})$ is, in fact, a combination of functions. Each of them is represented by a **layer** of a neural network, first of which is called the **input layer**, the last – **output layer**, and all the layers in between are called **hidden layers**. The output of one layer is passed directly as the input of the next one, hence the name "feedforward neural network". By completing subsequent learning cycles, called **epochs**, the neural network optimizes its parameters of each component function, with the aim to estimate a model which predicts the data labels most accurately.

2.5 Ethical dimension

2.5.1 Risks and notable controversies

Numerous risks related to automated face recognition have been identified so far; while some are associated with the vulnerability of any databases or computer systems themselves, others point out questionable and often confidential motives of corporations.

Firstly, every face data used – for example – for access control has to be stored in some kind of database. Should it be compromised, it creates the possibility of accessing vulnerable data by an unauthorized person. Although the same risk occurs for every other method of access control, many people claim that the thought of their face metrics being compromised occurs to them as especially unsettling.

Another controversy caused by imperfection of systems regards the possibility of making a false positive or false negative mistake. In case of automated searching for known criminals, that could lead either to people being falsely accused or charged or to overlooking the criminals the system was initially looking for. Both of these cases pose an obvious risk. The event most often brought up in this respect is the Super Bowl XXXV in Tampa, which took place in 2001 [27]. Unbeknownst to the the participants, their faces were scanned and compared to the faces of known criminals to test the use of face recognition on public events. Although it is claimed that this system helped identify 19 people with outstanding warrants, there were many other matches that were false positives; but what probably caused the greatest indignation was the fact of using face recognition technology without people's knowledge or consent.

In some situations, face recognition algorithms behave differently depending on people's race. There are cases known when the faces of black people were not detected, unlike their white friends' in the same picture [28]; or worse, they were labeled by Google as "gorillas" [29]. The systems are also more likely to mismatch dark-skinned faces [30], which can lead to disastrous results, not to mention the resulting feeling of exclusion among people of color.

What remains the subject of a heated debate is the possibility of using the biometrical recognition by the government against the citizens’ will or without their knowledge at all. From the Super Bowl XXXV case mentioned above, through the documents disclosed by Edward Snowden in 2014, stating that the NSA was secretly collecting millions of peoples’ faces photos per day, to China installing face recognition systems in classes, pharmacies and to track ethnic minorities [31] – many governments have shown multiple times that they do not always inform their citizens about the extent to which their biometrics are being used.

Finally, the biggest tech companies also have proven not to be fully transparent in respect to biometrics privacy. Facebook first introduced their DeepFace algorithm in 2011 by adding tag suggestions to the photos the users posted. It required other users’ faces to be analyzed and stored, which was firstly an opt-out feature, meaning that it was turned on by default and required user action to be turned off. The Facebook CEO, Mark Zuckerberg, did not conceal the company’s plan of using the photos to improve their AI algorithms. Only after numerous protests and lawsuits have they made the feature opt-in [32, 33]. However, together with other controversies surrounding Facebook, this became a repeatedly mentioned argument in the discussion on the company’s privacy policy.

2.5.2 Recent regulations and bans in USA

As of May 2019, many major cities in the USA began banning the use of face recognition by the police and city agencies, beginning with San Francisco. The ban has been justified by potential abuse and breach of citizens’ privacy by the government [34]. Oakland, Somerville and Brookline in Massachusetts soon followed, as did San Diego in December [35]. However, the last straw was probably the first case of wrongfully arresting a man based on a false indication by face recognition system – in June 2020, a Black man in Detroit has been detained for a crime he had not commited [36, 37]. Combined with recent protests against the police brutality towards Black people, this led to a law being proposed that would ban the use of face recognition by federal law enforcement agencies. Tech giants such as Amazon, IBM and Microsoft declared their support, deciding to quit selling their face recognition systems to the police [38]. In the meantime, many more cities voted on banning the use of face recognition. As of September 2020, the proposed bill is undergoing the legislative process.

2.6 The “10-year challenge” controversies

In 2019, a new trend appeared on social media such as Twitter, Facebook or Instagram (owned by Facebook), consisting in posting one’s current photo and another one from ten years ago. The meme quickly went viral and many people started posting their photos, up-to-date and from 2009, next to each other; it raised many suspicions about the company using the photos to train their face recognition algorithms. Aging has always been a great challenge for automated face recognition, but in the age of neural networks – DeepFace being a neural network as well [6] – such a big dataset could be easily used to enhance its accuracy. Facebook itself denied initiating the challenge, emphasizing the possibility to opt out of face recognition on the social network [39]. However, no research has been done so far that would confirm or exclude the usefulness of such a dataset.

Chapter 3

Contributions

3.1 Introduction

This chapter describes the research and experiments conducted for this thesis, and summarizes their results. Firstly, materials and methods are discussed, with the aim to provide a full description of the images and tools used in the research. Next, the dataset construction is explained. Finally, the conducted experiments are described, and their results are discussed.

3.2 Materials and methods

3.2.1 Materials

A total of 3,720 photos of 31 people have been gathered for this experiment. The simplest way to gather hundreds of photos of the same people as children and adults was to gather film stills or other publicly available photos of well-known actors; as a result, the images in the dataset depict popular movie stars.

The photos have been split into two categories, containing photos of adults (2,480) and children (1,240). Each of the two catalogs has been split into further 31, according to the identity of the person and labeled respectively. Each catalogue contained 80 photos of the person as an adult (age varying from 25 to 57), or 40 photos of the person as a child (age varying from 6 to 14). The age of the person within a catalogue varied, and was not determined by any pattern.

Firstly, 250 photos of each person as an adult and 150 photos of this person as a child have been downloaded automatically. The photos have then been reviewed with regard to whether they show the face of the correct person, and whether the face is depicted in front view. From all the photos satisfying these conditions, 80 photos from adulthood and 40 photos from childhood per person have been selected to form the final dataset. All the photos have been resized to 150 pixels width for greater computation speed. Example photos have been shown in figure 3.1.

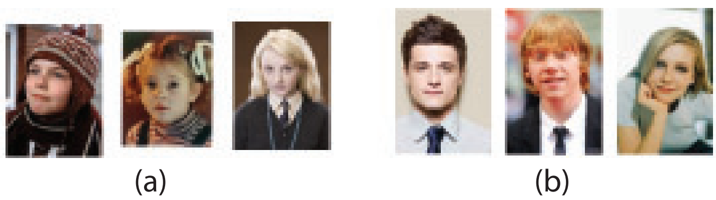

Figure 3.1: Example photos used in the experiments, depicting children (a) or adults (b).

3.2.2 Methods

1. To bulk download the photos as the result of a web search, **Bing Search API**[1] by Microsoft has been used. It has been implemented in Python language, and it allowed for downloading hundreds of images at once (results of a search query).

2. To perform feature extraction and obtain face encodings, **face_recognition**[2] library by Adam Geitgey has been used. It embeds Google's FaceNet feature extraction algorithm and allows for generating the face encodings from photos. Face_recognition methods are called over a catalog, which contains subcatalogs of photos. Each of subcatalogs should contain photos of one person only, and be named accordingly, for example by the person's name. Face_recognition generates a Python dictionary, its keys being the subcatalogs names and values being lists of encodings generated from the photos found in this subcatalog.

3. **PyTorch**[3], a Python framework based on the library Torch, has been used to construct a feedforward neural network. There are other frameworks offering similar values, most notable being TensorFlow[4]; however, PyTorch has been chosen due to its similarity to Python, developed community support, and popularity among researchers. The neural network built for the experiment has three linear layers and two ReLu layers appearing alternately. For each sample under tests, it produces a probability vector, which elements are the probabilities of the element belonging to subsequent classes. The neural network uses Cross Entropy Loss as the loss function. The loss is calculated as [40]:

$$\mathcal{L} = -\mathbf{y} \cdot \log(\hat{\mathbf{y}}), \tag{3.1}$$

   where:
   - $\mathbf{y}$ is the true probability vector, which elements are equal either to 0 or 1,
   - $\hat{\mathbf{y}}$ is the estimated probability vector.

4. **Scikit-learn**[5] is a Python module build on SciPy used for machine learning. It facilitates performing various machine learning methods, e.g. SVM.

5. KNN and Euclidean distance have been implemented in Python.

3.3 Experiments and results

In order to verify the impact of childhood photos in the training dataset and the training dataset size on recognition accuracy, nine sets of images have been constructed and three experiments have been conducted. In what follows, a brief description of the dataset construction is provided.

[1] www.azure.microsoft.com/pl-pl/services/cognitive-services/bing-web-search-api/
[2] www.github.com/ageitgey/face_recognition
[3] www.pytorch.org
[4] www.tensorflow.org
[5] www.scikit-learn.org

3.3.1 Dataset construction

Nine datasets, consisting of childhood and adulthood photos of 31 people, have been constructed:

1. Three training datasets (A, B and C) of 80 images per person, in which childhood photos made up sequentially 0% (0 images), 25% (20 images), and 50% (40 images) of the dataset, and the rest of the images were adulthood photos;

2. One training dataset (D) consisting only of childhood photos, in the number of 40 images per person;

3. Four training datasets (E, F, G and H) consisting only of adulthood photos, containing sequentially 5, 20, 40 and 60 images per person;

4. A test dataset (I), consisting only of photos from adulthood, in the number of 30 images per person. The photos used in this dataset have not been used in any of training datasets.

The datasets have been then encoded by FaceNet algorithm, resulting in nine sets of labeled vectors representing the faces, which have been used in three experiments.

3.3.2 The impact of childhood photos on recognition accuracy

In the first experiment, datasets A, B, C, D and I have been used to determine the dependence of recognition accuracy from the percentage of childhood photos in the training dataset. The encodings were subjected to classification by FNN, SVM, Euclidean distance classifier and KNN algorithm with $k = 3$. In case of a neural network, the accuracy has been slightly different in each iteration, so the accuracies from the last 25 iterations have been averaged as the final result. Experiment results have been collected in table 3.1 and illustrated in figure 3.2.

Percent of children in the training dataset [%]	**Accuracy of recognition [%]**			
	FNN	**SVM**	**Euclidean distance**	**KNN, k=3**
0 (dataset A)	89.59(1.99)	**99.77**	99.55	99.43
25 (dataset B)	83.85(8.66)	**99.09**	**99.09**	**99.09**
50 (dataset C)	78.67(7.65)	99.21	99.21	98.98
100 (dataset D)	36.96(6.97)	87.29	88.53	**88.76**

Table 3.1: Accuracy of recognizing adults as a function of childhood photos percentage in the training dataset. In case of the neural network, the mean from last 25 iterations is given, and the standard deviation is given in brackets.

3.3.3 The impact of k value on KNN accuracy as a function of childhood photos percentage in the training dataset

In the second experiment, datasets A, B, C, D and I have been used to determine the accuracy of recognition by KNN algorithm as a function of k value, and the percentage of childhood photos in the training dataset. All the encodings have been subjected to KNN classification, but the k parameter takes the value either 1, 3, 5, 7 or 8. The experiment results have been collected in table 3.2 and illustrated in figure 3.3.

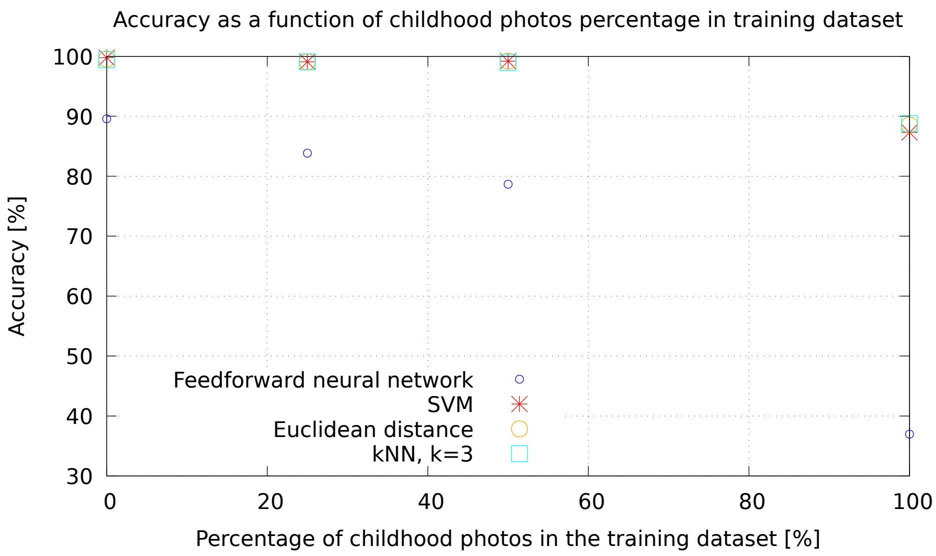

Figure 3.2: Accuracy of recognizing adults as a function of children percentage in the training dataset.

Percent of childhood photos in the training dataset [%]	**Accuracy of recognition** [%]				
	$k = 1$	$k = 3$	$k = 5$	$k = 7$	$k = 8$
0 (dataset A)	**99.55**	99.43	99.31	99.21	99.09
25 (dataset B)	**99.09**	**99.09**	**99.09**	98,98	98.98
50 (dataset C)	**99.21**	98.98	98.86	98.75	98.75
100 (dataset D)	88.53	88.76	88.88	**88.99**	88.53

Table 3.2: Accuracy of recognizing adults as a function of childhood photos percentage in the training dataset with various k values.

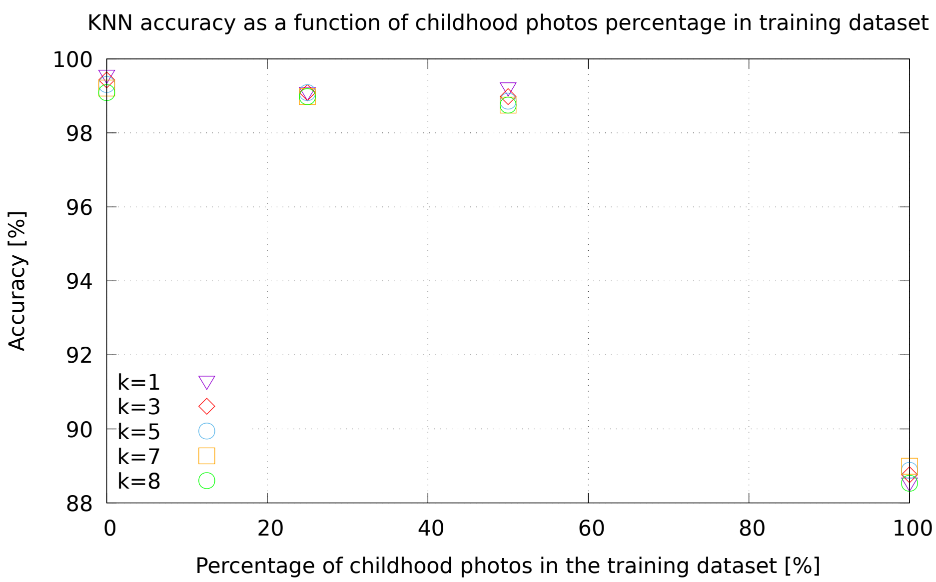

Figure 3.3: Accuracy of KNN algorithm as a function of children percentage in the training dataset with various k values.

3.3.4 The impact of training dataset size on recognition accuracy

In the third experiment, datasets E, F, G, H, A and I have been used to determine the dependence of recognition accuracy from the size of the training dataset. The encodings have been classified by the same classifiers as previously: FNN, SVM, Euclidean distance classifier and KNN with $k = 3$. Analogically to the previous experiment, the accuracy of the neural network has been calculated as mean accuracy from 25 last iterations. The logarithmic curve $a \cdot \ln(x) + b$ has been fitted to the FNN accuracy. The resulting function takes the form of $27.40 \cdot \ln(x) - 30.55$. Experiment results have been collected in table 3.3 and illustrated in figure 3.4.

Number of images in the training dataset	**Accuracy of recognition [%]**			
	FNN	**SVM**	**Euclidean**	**KNN, *k=3***
5 (dataset E)	10.70(3.31)	99.09	**99.20**	99.09
20 (dataset F)	57.44(7.22)	**99.43**	98.98	98.98
40 (dataset G)	71.67(7.65)	**99.43**	99.43	99.09
60 (dataset H)	77.70(7.64)	**99.66**	99.66	99.32
80 (dataset A)	89.19(4.83)	**99.77**	99.55	99.43

Table 3.3: Accuracy of recognizing adults as a function of training dataset size. In case of the neural network, the mean from last 25 iterations is given, and the standard deviation is given in brackets.

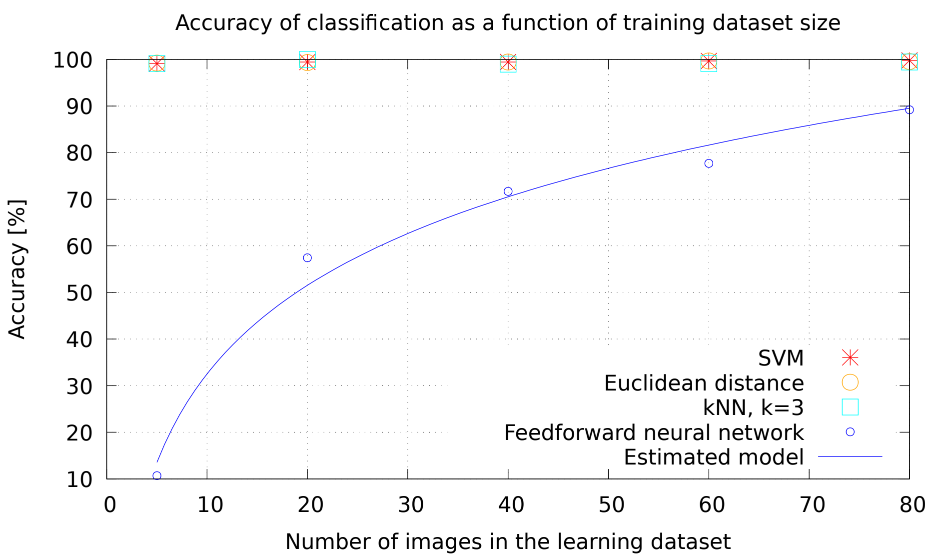

Figure 3.4: Accuracy of recognition as a function of the training dataset size.

3.4 Discussion of results

3.4.1 The impact of childhood photos on recognition accuracy

From the results of the experiment, it can be concluded that adding childhood photos to the training dataset does not increase the accuracy of recognizing adults. Rather, it decreases the accuracy, having only slight impact on it in case of SVM, Euclidean distance and KNN; it had significantly greater impact on feedforward neural network classifier.

In case of Euclidean distance and KNN classifier, the slight impact can be easily explained – the FaceNet encoding has been invented with a view to the Euclidean distance classifier, so the more similarity exists between two encoded faces, the closer their encoding lie in the Euclidean space. That means both the Euclidean distance and KNN classifiers look for one or three encoding vectors that are closest to the encoding under test, and are not be affected by other, more distant vectors. For this reason, adding more "insignificant" encodings to the dataset affected the accuracy only slightly. In contrast to that, a neural network takes all the data into consideration and all are equally significant, so adding more childhood photos to the training dataset had a greater impact on recognizing adults.

It should be noted that the accuracy of recognizing adults on the basis of childhood photos only is quite high; while the accuracy of the neural network dropped to 36.93% in this case, the accuracy of KNN with $k = 3$ was as high as 88.76%.

3.4.2 The impact of k value on KNN accuracy as a function of childhood photos percentage in the training dataset

Changing the k value did affect the accuracy of KNN algorithm for almost all datasets, however only slightly. For the dataset with 25% content of childhood photos, the change was minimal – computations with k value equal to 1, 3 and 5 achieved the same accuracy, and computations with k value equal to 7 and 8 achieved the same – slightly lower – accuracy. For the datasets with 0% and 50% content of childhood photos, increasing the k value lowered the recognition accuracy. However, for the dataset consisting of childhood photos only, increasing the k value resulted in better recognition accuracy.

The probable explanation of this exception is the same as in the first experiment: the FaceNet encodings are made for Euclidean classification, which means the photos depicting the same person lie close to each other as vectors in the Euclidean space. This means that if the training dataset includes any truly similar image to the image under classification – in this example, an adult person – other vectors do not have significant impact on the classification. However, if the dataset consists only of childhood photos, increasing the margin of error by higher k values makes the classification more accurate.

3.4.3 The impact of training dataset size on recognition accuracy

Changing the training dataset size did not affect the SVM, Euclidean and KNN classifiers much. The classification accuracy did not drop or increase by as little as 1% in any case. However, the training dataset size has a great impact on the accuracy of FNN, which success rate was as little as 10.70% for five photos per person while other classifiers maintained the accuracy of over 98%. However, the accuracy of FNN shows a logarithmic increase as the training dataset size increases.

The accuracy increasing with the training dataset size is a well-known property of most neural networks [41]. The fitted function, $27.40 \cdot \ln(x) - 30.55$, reaches 95.63% for number of photos in the training dataset equal to 100. This suggests that for even bigger training datasets the FNN could achieve the success rate equal to those of other classifiers, or even better.

Chapter 4

Conclusion

4.1 Objective of the thesis and summary

The main objective of this thesis was to determine whether the presence of childhood photos in the training dataset could facilitate face recognition of adults. The impact of training dataset size on classification accuracy has also been examined, as well as the impact of k parameter value in KNN classification for various percentages of childhood photos in the training dataset.

It can be concluded that adding childhood photos to the training dataset does not increase the classification accuracy for faces encoded by FaceNet feature extraction algorithm. However, for the training datasets consisting only of childhood photos, the accuracy of recognizing adults can be as high as 88.99% for KNN classification algorithm.

Another conclusion is that the size of the training dataset has an impact on classification accuracy; this impact is tremendous for classification by a neural network, but minimal for every other classifier examined. The classification accuracy reaches over 99% for SVM, Euclidean distance and KNN classifiers for as few as 5 images in the training dataset per person, while the accuracy of the FNN showed a logarithmic growth as a function of the training dataset size, and achieved lower accuracy than other classifiers even for 80 images per person in the training dataset. It can also be stated that neural networks are not optimal classifiers in case of FaceNet feature extraction approach, in contrast to linear or Euclidean-based classifiers.

The final conclusion is that the k parameter value has an impact on the success rate of KNN classifier, which proved to be highly accurate itself. Higher k values lower the classification if the training dataset contains any adulthood photos; however, if the training dataset consists only of childhood photos, the proportionality is direct – higher k values increase the accuracy of KNN classification.

4.2 Future work

To draw further conclusions, similar research should be conducted for images encoded with different feature extraction algorithm. FaceNet has been invented with the purpose of Euclidean distance classification; for this reason, the Euclidean distance and KNN classificators performed well in the conducted experiment. For other feature extraction algorithms, different classificators might prove to be more useful.

References

[1] Ropek L, 2020: "Could National Unrest Derail the Future of Facial Recognition?", retrieved on August 7th from govtech.com/policy/Could-National-Unrest-Derail-the-Future-of-Facial-Recognition.html. (cited at page 8)

[2] Ballantyne M, Boyer R, Hines L, 1996: "Woody Bledsoe. His Life and Legacy", *AI Magazine*, 17(1), p. 7. (cited at pages 8 and 9)

[3] Jackson N, 2010: "Facebook Will Start Using Facial Recognition Next Week", retrieved on August 7th from theatlantic.com/technology/archive/2010/12/facebook-will-start-using-facial-recognition-next-week/68121. (cited at page 8)

[4] Parmar DN, Mehta BB, 2014: "Face Recognition Methods & Applications", *International Journal of Computer Technology & Applications*, Vol 4 (1), pp. 84-86. (cited at page 8)

[5] Schroff F, Kalenichenko D, Philbin B, 2015: "FaceNet: A Unified Embedding for Face Recognition and Clustering", *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815-823. (cited at pages 8, 10 and 13)

[6] Taigman Y, Yang M, Ranzato M, Wolf L, 2014: "DeepFace: Closing the Gap to Human-Level Performance in Face Verification", *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701-1708. (cited at pages 8, 10 and 17)

[7] Boyer RS (Ed.), "Automated Reasoning: Essays in Honor of Woody Bledsoe", Kluwer Academic Publishers, 1991. (cited at page 9)

[8] Kalat D, 2018: "Nervous System: Taking Biometrics At Face Value", retrieved on June 28th from law.com/legaltechnews/2018/11/02/nervous-system-taking-biometrics-at-face-value/. (cited at page 9)

[9] Goldstein AJ, Harmon LD, Lesk AB, 1971: "Identification of human faces", *Proceedings of the IEEE*, 59(5), pp. 748–760. (cited at page 10)

[10] Sirovich L, Kirby M, 1987: "Low-dimensional procedure for the characterization of human faces", *Journal of the Optical Society of America*, 4, pp. 519-524. (cited at page 10)

[11] Turk MA, Pentland AP, 1991: "Face Recognition Using Eigenfaces", *Journal of Cognitive Neuroscience*, 3(1), pp. 71-86. (cited at pages 10 and 11)

[12] Jack L, 2015: "A New Facial Recognition Mobile App From Listerine Helps Blind People See Smiles", retrieved on August 7th from fastcompany.com/3050622/a-new-facial-recognition-mobile-app-from-listerine-helps-blind-people-see-smiles. (cited at page 10)

[13] Ada Lovelace Institute, 2019: "Beyond face value: public attitudes to facial recognition technology", retrieved on August 7th from adalovelaceinstitute.org/wp-content/uploads/2019/09/Public-attitudes-to-facial-recognition-technology_v.FINAL_.pdf. (cited at page 10)

[14] Abdi H, Williams LJ, 2010: "Principal component analysis", *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4), 433–459. (cited at page 11)

[15] Bartlett M, Movellan J, Sejnowski T, 2002: "Face Recognition by Independent Component Analysis", *IEEE Transactions on Neural Networks*, 13(6), pp. 1450–1464. (cited at page 11)

[16] Draper B, Baek K, Bartlett M, Beveridge J, 2002: "Recognizing faces with PCA and ICA", *Computer Vision and Image Understanding*, 91(2003), pp. 115–137. (cited at page 11)

[17] Wagner P, 2012: "Fisherfaces", retrieved on June 12th from bytefish.de/blog/fisherfaces. (cited at page 12)

[18] Belhumeur P, Hespanha J, Kriegman D, 1997: "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), pp. 711-720. (cited at page 12)

[19] Ahonen T, Hadid A, Pietikainen M, 2004: "Face Recognition with Local Binary Patterns", *2004 European Conference on Computer Vision*, pp. 469–481. (cited at page 12)

[20] Duda R, 2008: "Linear Discriminant", retrieved on June 2nd from cs.princeton.edu/courses/archive/fall08/cos436/Duda/PR-simp/lin-disc.htm. (cited at page 13)

[21] Rajalakshmi R, Jeyakumar M, 2012: "A Review on Classifiers Used in Face Recognition Methods Under Pose And Illumination Variation", *International Journal of Computer Science Issues*, 9(6), pp. 474-485. (cited at page 13)

[22] Ślot K, "Wybrane zagadnienia biometrii", WKŁ, 2005. (cited at page 13)

[23] Hou W, Li D, Xu C, Zhang H, 2018: "An Advanced k Nearest Neighbor Classification Algorithm Based on KD-tree", *IEEE International Conference of Safety Produce Informatization*, pp. 902-905. (cited at page 13)

[24] Kim E, 2013: "Everything You Wanted to Know about the Kernel Trick", retrieved on September 12nd from eric-kim.net/eric-kim-net/posts/1/kernel_trick.html. (Not cited.)

[25] Vapnik V, Golowich S, Smola A, 1997: "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing", *Advances in Neural Information Processing Systems*, 9, pp. 281–287. (cited at page 14)

[26] Bagheri R, 2020: "An Introduction to Deep Feedforward Neural Networks", retrieved on September 22nd from towardsdatascience.com/an-introduction-to-deep-feedforward-neural-networks-1af281e306cd. (cited at page 15)

[27] Woodward JD, 2001: "Super Bowl Surveillance: Facing Up to Biometrics", Santa Monica, CA: RAND Corporation, 2001. (cited at page 16)

[28] Simon M, 2009: "HP looking into claim webcams can't see black people", retrieved on July 15th from edition.cnn.com/2009/TECH/12/22/hp.webcams. (cited at page 16)

[29] Guynn J, 2015: "Google Photos labeled black people »gorillas«", retrieved on July 15th from eu.usatoday.com/story/tech/2015/07/01/google-apologizes-after-photos-identify-black-people-as-gorillas/29567465. (cited at page 16)

[30] Simonite T, 2019: "The Best Algorithms Struggle to Recognize Black Faces Equally", retrieved on July 15th from wired.com/story/best-algorithms-struggle-recognize-black-faces-equally. (cited at page 16)

[31] Byler D, 2019: "China's hi-tech war on its Muslim minority", retrieved on July 15th from theguardian.com/news/2019/apr/11/china-hi-tech-war-on-muslim-minority-xinjiang-uighurs-surveillance-face-recognition. (cited at page 17)

[32] Rimm H, 2019: "Facebook is giving your face some privacy", retrieved on July 15th from refinery29.com/en-us/2019/09/8378597/facebook-face-recognition-opt-in-update. (cited at page 17)

[33] Samuel S, 2019: "Facebook will finally ask permission before using facial recognition on you", retrieved on July 15th from vox.com/future-perfect/2019/9/4/20849307/facebook-facial-recognition-privacy-zuckerberg. (cited at page 17)

[34] Conger K, Fausset R, Kovalesky SF, 2019: "San Francisco Bans Facial Recognition Technology", retrieved on August 31st from nytimes.com/2019/05/14/us/facial-recognition-ban-san-francisco.html. (cited at page 17)

[35] Schneider B, 2020: "We're Banning Facial Recognition. We're Missing the Point", retrieved on September 1st from nytimes.com/2020/01/20/opinion/facial-recognition-ban-privacy.html. (cited at page 17)

[36] Hill K, 2020: "Wrongfully accused by an algorythm", retrieved on September 1st from nytimes.com/2020/06/24/technology/facial-recognition-arrest.html. (cited at page 17)

[37] Rahal S, Hicks M, 2020: "Detroit police work to expunge record of man wrongfully accused with facial recognition", retrieved on September 1st from eu.detroitnews.com/story/news/local/detroit-city/2020/06/26/detroit-police-clear-record-man-wrongfully-accused-facial-recognition-software/3259651001/. (cited at page 17)

[38] Shepardson D, 2020: "Two U.S. senators seek ban on collecting customer biometric data without consent", retrieved on September 1st from reuters.com/article/us-usa-congress-facial-recognition/two-u-s-senators-seek-ban-on-collecting-customer-biometric-data-without-consent-idUSKCN2520CN. (cited at page 17)

[39] Martin N, 2019: "Was The Facebook »10 Year Challenge« A Way To Mine Data For Facial Recognition AI?", retrieved on July 15th from forbes.com/sites/nicolemartin1/2019/01/17/was-the-facebook-10-year-challenge-a-way-to-mine-data-for-facial-recognition-ai. (cited at page 17)

[40] Martinek V, 2020: "Cross-entropy for classification", retrieved on September 22nd from towardsdatascience.com/cross-entropy-for-classification-d98e7f974451. (cited at page 19)

[41] Brownlee J, 2017: "How Much Training Data is Required for Machine Learning?", retrieved on August 31st from machinelearningmastery.com/much-training-data-required-machine-learning. (cited at page 23)