COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS

FEW-SHOT FACE RECOGNITION USING ARTIFICIAL NEURAL NETWORKS

Master's Thesis

2020 Ing. Igor Slovak

COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS

FEW-SHOT FACE RECOGNITION USING ARTIFICIAL NEURAL NETWORKS

Master's Thesis

Study Programme: Cognitive Science
Field of Study: 2503 Cognitive Science
Department: Department of Applied Informatics
Supervisor: prof. Ing. Igor Farkaš, Dr.
Consultant: Mgr. Ing. Matúš Tuna

2020 **Ing. Igor Slovak**

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname: Ing. Igor Slovák
Study programme: Cognitive Science (Single degree study, master II. deg., full time form)
Field of Study: Cognitive Science
Type of Thesis: Diploma Thesis
Language of Thesis: English
Secondary language: Slovak

Title: Few-shot face recognition using artificial neural networks

Annotation: Artificial neural networks achieve high accuracy in image recognition tasks. However, this requires a high number of labelled learning examples per class, which is a problem for the majority of applications based on face recognition. This is because acquiring a large number of training examples for every user is usually impractical. The majority of face recognition tasks, for example user identification for security purposes, require learning to identify the user using only a small number of training examples.

Aim:
1. Create an overview of state-of-the-art face recognition research and few-shot learning research. Assess the sociological impact of this technology.
2. Empirically compare the accuracy of state-of-art few-shot learning methods, such as Siamese neural networks or Prototypical networks, in both the face verification task and the face classification task.

Literature:
Koch, G., Zemel, R., Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In ICML Deep Learning Workshop, vol. 2.
Schroff, F., Kalenichenko, D., Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 815-823).
Snell, J., Swersky, K., Zemel, R. (2017). Prototypical networks for few-shot learning. In: Advances in NIPS, pp. 4080-4090.

Supervisor: prof. Ing. Igor Farkaš, Dr.
Consultant: Mgr. Ing. Matúš Tuna
Department: FMFI.KAI - Department of Applied Informatics
Head of department: prof. Ing. Igor Farkaš, Dr.

Assigned: 27.02.2019

Approved: 27.02.2019 prof. Ing. Igor Farkaš, Dr.
Guarantor of Study Programme

.....
Student

.....
Supervisor

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Ing. Igor Slovák
Študijný program: kognitívna veda (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Few-shot face recognition using artificial neural networks
Rozpoznávanie tváre s použitím metód rýchleho učenia v neurónových sieťach

Anotácia: Umelé neurónové siete dosahujú mimoriadne výsledky v oblasti rozpoznávania obrazu. Avšak na dosiahnutie vysokej klasifikačnej úspešnosti je potrebné veľké množstvo učiacich príkladov, čo predstavuje problém pre väčšinu aplikácií rozpoznávania tváre, nakoľko zber veľkého množstva príkladov tváre pre každého používateľa je nepraktický. Väčšina aplikácií rozpoznávania tváre si vyžaduje prístup, ktorý umožňuje naučenie rozpoznávania tváre iba na základe pár učiacich príkladov.

Cieľ:
1. Urobte prehľad o súčasnom výskume rozpoznávania tváre a vo výskumemetód rýchleho učenia v neurónových sieťach. Zhodnoťte potenciálne spoločenské dopady tejto technológie.
2. Empiricky porovnajte úspešnosť súčasných metód rýchleho učenia, akonapríklad Siamské neurónové siete alebo Prototypické siete, v úlohách verifikácie a indentifikácie tváre.

Literatúra:
Koch, G., Zemel, R., Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In ICML Deep Learning Workshop, vol. 2.
Schroff, F., Kalenichenko, D., Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 815-823).
Snell, J., Swersky, K., Zemel, R. (2017). Prototypical networks for few-shot learning. In: Advances in NIPS, pp. 4080-4090.

Vedúci: prof. Ing. Igor Farkaš, Dr.
Konzultant: Mgr. Ing. Matúš Tuna
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.

Dátum zadania: 27.02.2019

Dátum schválenia: 27.02.2019

prof. Ing. Igor Farkaš, Dr.
garant študijného programu

.....
študent

.....
vedúci práce

Declaration

I hereby declare that I elaborated this diploma thesis independently using the cited literature.

Ing. Igor Slovak

Bratislava, 2020

Acknowledgment

I would like to express my gratitude to my supervisor, prof. Ing. Igor Farkaš, Dr., not only for his counsel concerning the present thesis but also for his continuous support and mentoring throughout my entire journey in the world of cognitive science.

I would also like to thank my consultant, Mgr. Ing. Matúš Tuna, for allowing me to be a part of his neural networks research and his close guidance thought the past year, not forgetting his excessive experience with tuning neural networks. Next, I would like to thank my teacher RNDr. Kristína Malinovská, PhD., for her endless support and aid with my thesis.

Finally, I owe thanks to all the people that supported me, especially my wife, family and those that have contributed to this project.

ABSTRACT

Face recognition is an essential part of visual perception. Few-shot learning is inspired by the unique ability of humans to recognize objects only after one or a few presentations. We adopt an interdisciplinary approach by combining the research from neuroscience for face perception as well as computer science research for using neural networks in the face perception domain. We focused on the implementation of few-shot learning models for face recognition, namely the Prototypical networks and the Siamese networks. We have used Labeled Faces in the Wild dataset (Huang et al., 2007). We provide an analysis showing that few-shot learning models do not provide sufficient classification accuracy compared to humans. However, with more computational power allowing a more thorough exploration of the deep learning models for few-shot face recognition is a very promising area.

Keywords: Deep Learning, Few-shot learning, Prototypical Network, Siamese Network, Face recognition

ABSTRAKT

Rozpoznanie tváre je nevyhnutnou súčasťou vizuálneho vnímania. Few-shot učenie sa inšpiruje jedinečnou schopnosťou ľudí rozoznať objekty už po jednej alebo niekoľkých prezentáciách. Interdisciplinárny prístup aplikujeme kombináciou výskumu neurovedy pre vnímanie tváre, ako aj výskumu počítačovej vedy používaním neurónových sietí v oblasti vnímania tváre. Zamerali sme sa na implementáciu few-shot učiacich modelov pre rozpoznávanie tváre, konkrétne Prototypických sietí a Siamských sietí. Použili sme dataset Labeled Faces in the Wild (Huang a spol., 2007). Prezentujeme analýzu, ktorá ukazuje, že modely s few-shot učením neposkytujú dostatočnú presnosť klasifikácie v porovnaní s ľuďmi. S väčšou výpočtovou silou, ktorá umožňuje dôkladnejšie skúmanie hlbokých modelov učenia je few-shot rozoznavanie ludí veľmi sľubná oblast .

Keywords: Deep Learning, Few-shot learning, Prototypical Network, Siamese Network, Face recognition

Table of Contents

List of Figures

List of Tables

Introduction

Face recognition is an essential part of visual perception. We use this skill during social interaction between humans. Face recognition is an important part of artificial intelligence with a wide application in the fields of recognition systems, surveillance applications, and social robotics.

Deep learning is the leading technology for computer vision, and deep learning is inspired by cognitive science and neuroscience. Few-shot learning is inspired by the unique ability of humans to recognize objects only after one or a few presentations. This works well for recognizing faces in humans; however, there are not many such applications in deep learning. In this thesis, we will focus on few-shot learning models that can be used for face recognition. This thesis adopts an interdisciplinary approach by combining the research from neuroscience for face perception as well as computer science research for using neural networks in the face perception domain.

In this master thesis, we will explain Face recognition from the neuroscience perspective in chapter 1. We outline some neuroscience details, e.g., the Fusiform Face Area (FFA) that is responsible for face perception and some research in the area of face recognition and theories of perception. Then we continue with chapter 2 Artificial neural networks, where we will explain the basics about Artificial neural networks and more details of Convolutional neural networks (CNN's). In chapter 3, we will mention some research in face recognition using CNN and discuss the ethical concerns of using face recognition technologies around the world. Next, we will explain few-shot learning principles and describe in detail the Prototypical neural networks and Siamese neural networks. In chapter 5, we will focus on our work on few-shot learning models for face classification and verification. Then we will describe the dataset that we used to train our models. Finally, we present our architectures used in our experiments. In the following chapter 6, we will present our implementation and experimental results. After that, we discuss the results and provide some conclusions to the experimental part and suggestions for future work.

1. Face perception – neuroscience view

Face perception and recognition of its individual characteristics are an integral part of cognitive processes. Facial perception is different from the perception of other objects and is accompanied by specific neurophysiological processes. Thus, during the evolution of the brain and human social life, specialized brain structures, and a specific neurocognitive network were created. Modernization of brain imaging methods has made it possible to observe the activation of specific areas of the brain in specific activities, including the perception of the human face. Significant technological advances have brought much new knowledge that has led to the formulation of new theories of perception, which also have practical benefits. (Blažek & Trnka, 2009).

For purposes of this chapter, which deals with the specificities of facial perception, we will mention Rakic's theory of radial units (Rakic, 1995), which explains the process of forming the cladding of the terminal brain during the early years of human life. The basis of the theory is to increase the number of cells in mature mammals and especially the possibility of increased mitotic division, where one of the two daughter cells travels according to the protrusion of radial glia into the emerging cortex, and the other can divide. The frequency of this division affects the resulting number of neurons in the cortex. We are increasing the cortex volume results in finer differentiation of cortical fields and more complex brain structures (including enlargement).

Damasio's model of somatic markers for experiencing and awareness of internal states (Damasio, et al., 2000) points out that body signals also increase the accuracy and efficiency of the decision-making process in recognizing human emotionality. The central idea is that body marker signals influence the stimulus-response process. Markers reflect, but are not limited to, bioregulatory processes, such as emotions, as well as physical states, and regulatory processes or their manifestations in the central nervous system. (Damasio, Everitt, & Bishop, 1996).

The action of the marker takes place at the conscious as well as unconscious level. Instead of calculating all behavioral alternatives associated with stimulus-response, the brain only focuses on those associated with a positively experienced somatic marker.

Markers that are experienced negatively are excluded from decision making. In this way, the response to the stimulus is reduced to a lower number, which can then be processed at a conscious level, for example, by searching for logical connections and relationships (Damasio, Everitt, & Bishop, 1996).

According to Mesulam's theory of neurocognitive networks, the various functional areas of the cerebral cortex are networked to form a neurocognitive network for certain processes that are closely related or interconnected. The neuro-cognitive network for recognizing the face and its components, including the expression, is quite complex. In this facial network, in addition to the primary visual cortex and the region of the lower occipital lobe, fusiform gyrus participates on the borderline between the temporal and occipital lobes, which probably play a crucial role in facial recognition), the amygdala and the prefrontal region. (see Figure 1) This network performs several tasks in parallel, in particular facial expression, emotional tuning of others, sex determination, the distinction of relatives, etc. Cognitively oriented theories of mental representation formation are also important for perception and facial recognition.

1.1 Neurophysiological mechanisms of facial perception

Neurophysiological processes of facial perception are specific, although many functional areas of the terminal brain cortex, which are crucial for facial recognition, may also be involved in other cognitive processes. In this context, a "facial specificity hypothesis" has been developed, which is linked to the existence of a neural network with a crucial special area on the border of the temporal and occipital lobes in the lower part - the gyrus fusiform (Blažek & Trnka, 2009). The uniqueness of this area has been known since the 1990s, and its existence and relationship to face recognition is proven mainly by the fact that we perceive faces as a special category of observed objects and attribute uniqueness to them. (Gautier & Nelson, 2001).

Koukolík & Drtinová (2006) demonstrate that, when the brain is exposed to visual stimuli in the form of faces, functional imaging methods show considerable activation of the Fusiform Face Area (FFA). Although this area is also active in the visual presentation of other objects - faces, it reacts to the face twice as intensely. The experiment of Rhodes

(2006) examined the function and essence of this area and was to decide among three hypotheses:

- FFA is specialized in processing visual stimuli related to faces
- FFA is specialized in the individualization of visually similar items within one category of a stimulus. This hypothesis is based on the fact that although human faces are important to us, it may not be the only category of visual stimuli that are very similar in absolute terms, and we must distinguish among them. It would be useful for the brain to have the ability to adapt (immediately after a series of sensations) to a series of stimuli and be able to differentiate them. This hypothesis does not attach to the crucial role of experience.
- FFA is specialized in individualization within a category with which an individual has experience. This hypothesis is similar to the previous one with the difference that instead of rapid brain adaptation, it is gradually gaining expert experience. Once the brain has gathered enough data to process the perception at a "deeper" level, the facial area of the gyrus fusiform takes up this activity.

Rhodes' research consisted in presenting human faces, objects with which the participants had no experience (a series of objects that were specified for the second hypothesis, for example, butterflies of one species differ so slightly that only an individual familiar with the category can differentiate between them) or objects with which they had one that could be considered an expert.

If hypothesis 1 were true, the FFA would only be activated in the presentation of human faces; if hypothesis 2, it would be activated in the presentation of all the objects they encountered, and its activation would continuously increase. The results of the experiment confirmed the hypothesis 3. People became active in the FFA area when they perceived human faces and if they perceived objects with which they had expert experience (such as butterfly connoisseurs).

Koukolík & Drtinová (2006) add that the results showed the existence of groups of FFA neurons that are tuned to the features necessary to distinguish members of different classes of visual objects. The facial function of FFA has been confirmed, among other

things, by other studies of injured patients (Goffaux, Jemel, Jacques, Rossion, & Schyns, 2003).

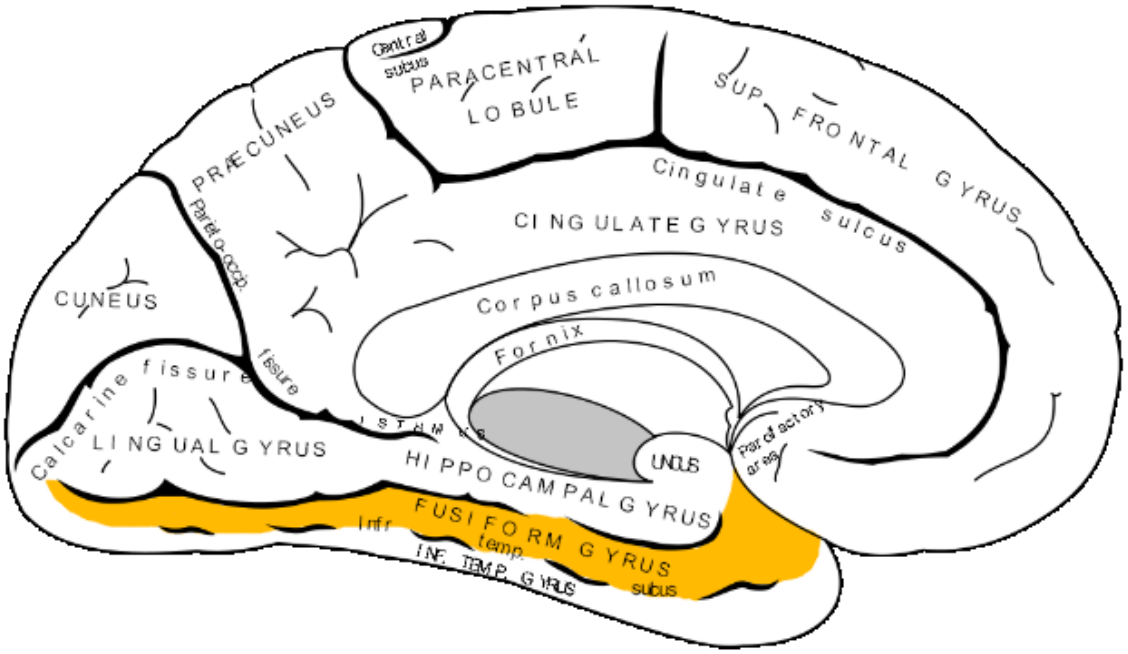

Figure 1: Localization of the fusiform face area – fusiform gyrus (Mysid, 2010)

1.2 Facial neurocognitive network

The technological boom in medicine has provided the finding that facial recognition is a complex process involving many areas in the brain. The facial neurocognitive network includes the following nodal areas (Blažek & Trnka, 2009):

a) The lower occipital lobe, which belongs to the visual cortex,
b) gyrus fusiform,
c) amygdala,
d) certain areas of the prefrontal cortex at the front of the frontal lobe. Furthermore, hearing centers in the temporal lobe are also connected to the network.

The facial network works as follows (Blažek & Trnka, 2009):

1. The perceived face is treated in the first step in the primary visual cortex as any other visual stimulus.
2. Visual analysis is performed, which recognizes partial elements of perceived objects. At this point, the object is identified as a human face.
3. Fusiform facial area (FFA) processes further information (already on the face).

4. It continues, together with the occipital face area, to the differentiation of other facial characteristics (gender, age, race, acquaintance) and individual face identification.
5. Identification is further fixed to the memory in the crown lobe with the participation of prefrontal areas (asymmetrically differentiated based on whether facial information is stored or retrieved). Through the limbic system, information is processed in the amygdala with the participation of some areas of the temporal and parietal lobes to recognize mimic expressions.
6. Amygdala also transmits information to various parts of the prefrontal area, which are the basis for assessing and shaping attitudes, for assessing the attractiveness and associating facial characteristics with anticipated personality traits, anticipating certain reactions in the context of social relationships.

Face recognition itself takes place in prefrontal areas where the background information flows from the fusiform facial area (FFA) (Blažek & Trnka, 2009). The facial neurocognitive network evolves during the ontogenesis process and forms a complex that is functionally interconnected with other areas, not only with the visual analyzer but also with the hearing analyzer districts for speech signal recognition and others. (In blind people, the dominant visual perception is replaced by increased susceptibility to auditory, tactile, and odor sensations.) (Blažek & Trnka, 2009)

1.3 Facial image processing

The retina of the eye, which senses and preprocesses the light signals coming into it through the lens, includes, in addition to the luminous cells (rods and cones), neurons originating in the proximal process. The optic nerve transmits information from the retina to the thalamus in the middle and then further to the visual centers, especially the occipital lobe of the cerebral cortex (see above). The perception of the object is based on the rapid shift of the view from one point to another (for the face, especially between the left and right eye, the tip of the nose, lips, the contour of the face), with the overall image reconstructed by the visual analyzer in the brain. The visual analyzer itself is further divided into two to three dozen districts, which are specialized and provide partial parts of the analysis. Each of them specializes in different aspects of visual perception, such as the position of the

object in space, movement of the object, contrast, etc. In the lower occipital lobe, there is an area that activates when facial is perceived. It is part of the lower occipital lobe, which is responsible for recognizing the individual physical characteristics and qualities of the face rather than recognizing the face itself. It is also referred to as the occipital region. (Blažek & Trnka, 2009)

Pre-adaptations in the field of visual information processing were important for the development of human cognitive abilities. In the first place, it is the formation of the so-called parvocellular cells, which are based on about 80% of ganglion neurons with smaller dimensions, in addition to the previously formed (mammalian) magnocellular cells consisting of 10% of ganglion cells, which are larger.

Magnocellular cells are characterized by low color resolution, high contrast sensitivity, fast resolution over time, and low resolution of spatial characteristics. The creation of the parvocellular cells has allowed higher primates to enhance visual perception, namely overall spatial orientation, object recognition and location and spatial orientation, three-dimensional vision, fine color differentiation, and speed motion resolution. (Koukolík F. , 2002)

Facial surface features, such as eye size and shape, eyebrow density, skin pigmentation, as well as the three-dimensional structure of the face and the location of the individual elements in it, are important for face recognition. (Rolls, & Ekman, 1992)

At facial perception, functional asymmetry of the brain is already traceable at the level of area activation in the visual cortex. Face perception is associated with the right half of the brain, but asymmetric activation also occurs in the primary V1 region. When observing other objects, i.e., objects that are not identified as a human face activate areas on the border of the temporal and occipital lobes in the areas following the secondary visual cortex.

1.4 Specifics of face perception

Once the perceived object is recognized as a human face, areas of the cortex are activated in the brain, which is specifically designed to analyze the face and its individual characteristics. This activation occurs even under difficult or unusual conditions for

identification, such as insufficient light, blurred image, a different angle of view, inverse face (rotated 180 °), etc.

The identification of an object as a face takes about 120 ms, and this time interval is considered to be evidence of a two-step process of facial recognition since a longer time is required to identify a person (Kato & Nakamura, 2004). There are several differences in facial perception compared to other objects, as the face is identified as a face based on simple criteria. It has been shown experimentally that the size of the face does not affect its perception (it plays a role in other objects). Observing the face as a complex with all characteristics (photographs) and as a simpler representation (e.g., line drawing) does not lead to a difference in the functioning of the relevant cortical areas (Allison, Puce, Spencer, & McCarthy, 1999). Manipulation of facial image (rotation, blur, negative image, moving different parts of the face, combinations of different faces) is not an obstacle to categorizing the face into the face category and activating areas for cognitive processing as a face (Farah, 1998).

There is a neurological disorder of prosopagnosia in which patients can classify an object as a face, but are unable to identify a face, even if it is demonstrably known to them (they are not able to identify their own face and associate it with themselves). The main symptom of prosopagnosia is the inability to analyze the face and its features, resulting from impaired or impaired gyrus fusiform (Grüter, Grüter, & Christia, 2008). Autism patients also achieve poor face recognition results. While a healthy individual is generally able to better distinguish and fix faces in memory than other objects, autists do not distinguish between faces and other objects and thus have much worse scores in face recognition (Hauck, 1998). Autists use centers other than people without this diagnosis to recognize and distinguish faces, leading to different results.

In summary, in addition to the involvement of the areas of the visual cortex, especially in the occipitalis inferior gyrus, gyrus fusiform and other areas of the occipital and crown lobes are decisive for the perception of the face. At the same time it is connected with other parts of the brain: especially the prefrontal cortex (in front of the frontal lobe) and its precincts, which are focused on self-knowledge, memories of perceived and identified person, or attribution of evaluation aspects; furthermore, the fusiform region also

has links to the limbic system and the temporal and temporal lobe. In addition to identifying the individual, gender differentiation concerning sexual signaling, and differentiating facial expressions in communication, these cognitive functions are also important for forming attitudes towards others (Koukolík & Drtinová, 2006). Recognition of belonging of individuals to more distant, different population groups, associated with assignment to the so-called "races" (i.e., categories based on a typological approach).

1.5 The specificity of perception of different races

What motivated the development of research activities on the perception of various human races was the inaccurate identification of eyewitnesses. The reason for the misconception of the court was misinformation provided by witnesses who, moreover, probably acted in good faith in the correctness of their identification. (Loftus, 1976) (Lickson, 1974) described a case from Florida: In 1971, a group of blacks were arrested and charged with murdering a man during a robbery. No clues were found in the scene of the crime that could withstand the trial and, at the same time, prove directly or indirectly that the detainees were real perpetrators. However, even five independent testimonies spoke against the detainees. The prosecution's language was eloquent: "What better than identifying a biased witness? Moreover, if there are five such witnesses? This is evidence that leaves no room for justified doubt." The court condemned the defendants. As a witness of the defense, Dr. Werner was called to court. Haythorn, a specialist psychologist who was supposed to disprove testimony by pointing out the effect of another race affecting identification, but since there were not enough conclusive studies at the time, his testimony was not considered relevant. The ability to better remember the faces of one's own race is due to the different facial image processing processes, namely the increased response in the left fusiform cortex and the right hippocampus. Besides, asymmetries in amygdala activity in the perception of faces associated with a race other than that of the observer were also observed when investigating the effect of another race. Other experiments (Eberhardt, 2005) have also shown a different activity of the prefrontal regions, which they relate to attitudes and beliefs gained in the social context since the activity of the prefrontal regions is also recorded in the investigation of racial prejudice (Richeson & Shelton, 2003).

However, some voices attribute different amygdala activity to learning and cultural influences (Lieberman, Hariri, Jarcho, & Eisenberger, 2005).

2. Artificial neural networks

Artificial neural networks (ANN) are computer systems inspired by biological neural networks, the brains of living organisms (Williams & Zipser, 1989). Generally, the brain works by sending individual impulses between many interconnected cells called neurons. Components of the ANN are neurons, connections with weights, activation function, and the learning rule. There are different approaches to training the ANN: supervised learning, unsupervised learning, reinforcement learning (Lippmann, 1988).

2.1 Introduction to Artificial neural networks

The neural network itself consists of mathematical models of neurons, where each neuron has its own set of weights and through these generates an output as a product of the input and the synaptic weight strength, which is subsequently processed by the activation function. These neurons are topologically assembled into a structure communicating through oriented evaluated junctions. All networks can be different. Very often, it depends on the nature of the neuron, the topological arrangement, and, last but not least, the learning strategy.

In Figure 2, we can see that the neurons are assembled into predetermined layers, and this structure is called the multilayer perceptron (MLP). The neurons in this model are usually not connected within a layer, but there is a full connection between the layers. Each individual connection is given a weight, which controls the strength of the connections between neurons. In this arrangement, there is always one input layer and one output, but there may be more than one hidden layer. The most standard architecture, especially in deep learning is the feed-forward neural network. It is called feed-forward because the signal propagates in only one direction, from input to output. Neural networks are characteristic of robustness and resistance to damage. It means they provide relatively correct outputs even if some parts of the input or the weights themselves are damaged.

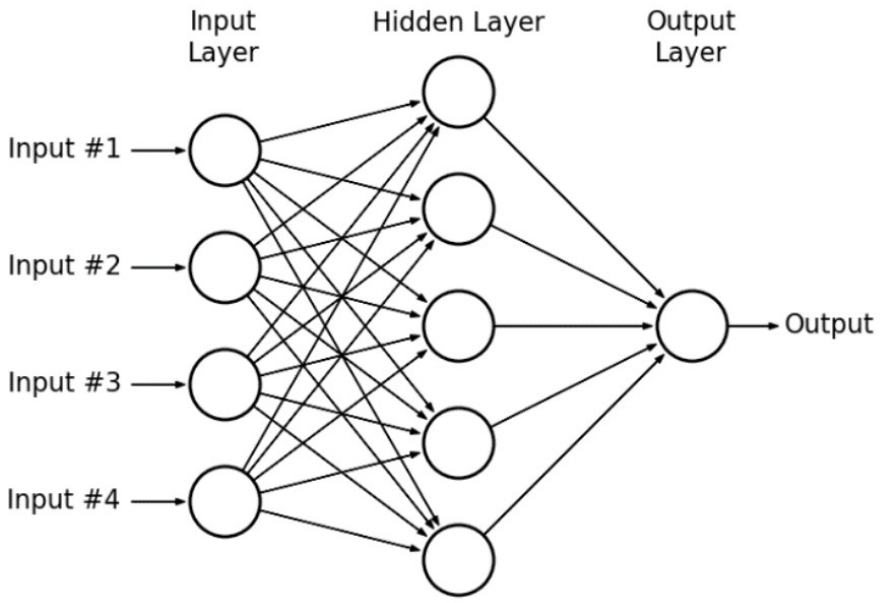

Figure 2: Multilayer perceptron with a hidden layer (Hassan et al., 2015)

The life of a neural network comprises two phases. The first phase is always learning or so-called *training* when the network learns its parameters based on the training dataset. Subsequently, it enters the *testing* phase, in which the network no longer changes, but it is tested on different inputs and evaluated.

2.1.1 Learning principles

The input weights *W* represent trainable parameters of the neuron. Learning is provided by an adaptation algorithm that sets the weights. This process takes place iteratively during the training phase when the algorithm has a set of input data and the corresponding output. The learning procedure finds a set of weights t alhat maximize the measure of correctness (performance) of the network. The great advantage of neural networks is the ability to find the right set of weights, even in those cases where the solution is very hard to find analytically. Usually, a large amount of training data is needed for this, on which the given network can learn correctly. Generally, there are three approaches that differ in how the learning objective is formulated, unsupervised learning, i.e., without a teacher, supervised learning, i.e., with the teacher and reinforcement learning, which is similar to supervised and we will not include it here due to space limitations of this thesis.

Learning without a teacher

Unsupervised learning is, in general, a machine learning method that looks for a hidden structure in data. It works on the principle of cluster analysis, so it looks for similar elements in the input data, which are sorted into groups (the number of groups can be known in advance), in which these objects have similar properties. We do not interfere in learning in any way, so the whole learning is based only on the information obtained from the input data set. The most famous neural network for unsupervised learning is the so-called Kohonen network, or it is also called the self-organizing map (Kohonen, 1982).

Learning with the teacher

In this case, the adaptation algorithm has a sufficient set of input pairs to the corresponding output. It, therefore, has specific examples of the correct outputs, which it uses to adapt the weights. Usually, the dataset is divided into three parts, training, testing, and validation parts. The ratio is not exactly given; it must be chosen according to the nature of the task. Typically, a training set contains about 60% - 80% of the total data set.

The algorithm gradually presents the individual elements and determines the deviation from the expected output and then performs the weight correction. This procedure, after it goes through the whole training set, is called an epoch. Hundreds to thousands of epochs are usually needed to train a network. The moment of stopping the learning is most often chosen by reaching a certain value of the total error when we declare the network learned. Learning can also stop if the error has stabilized at a value and no longer decreases. We then verify the performance of the learned network using a test set. There are more criteria for correct learning, but most often, it is the mean square error (MSE) between the outputs of the network and the ground truth values. If the performance over the test set is good, we assume that it will be approximately as good for inputs outside our set.

Given the context of this thesis, supervised learning is usually used on such datasets that contain image data and classifications of each item (e.g., the face of George W. Bush). ANN learns for many epochs in order to increase the accuracy of the representation needed for the classification task.

2.1.2 Multilayer perceptron

A perceptron network is a multilayered neural network with forward connections shown in Figure 2. Neurons in one layer are connected to all neurons in the previous layer. There are usually no connections between distant layers or between neurons within a single layer. Thus, each neuron has exactly as many inputs as there are neurons in the lower layer. The network input layer is only used to distribute input values. For the activation function of a multilayered perceptron, neither a simple stepwise nor a linear function is suitable. So, the nonlinear activation functions are used in perceptrons. These functions usually "squash" the input values into some small predefined range (for example, sigmoid, tanh) or set the negative values of the input to zero or close to zero (rectified linear unit - ReLU, leaky ReLU). In the classical multilayer perceptron, the sigmoid function is most often used. A hyperbolic tangent is also a popular option, and the ReLU is typical in deep learning. See Figure 3 with the equations and graphs of the activation functions:

$$sigmoid: \sigma(x) = \frac{1}{1 + e^{-x}}$$

$$tanh: tanh\ (x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

$$ReLU: relu\ (x) = \ max\ (0, x)$$

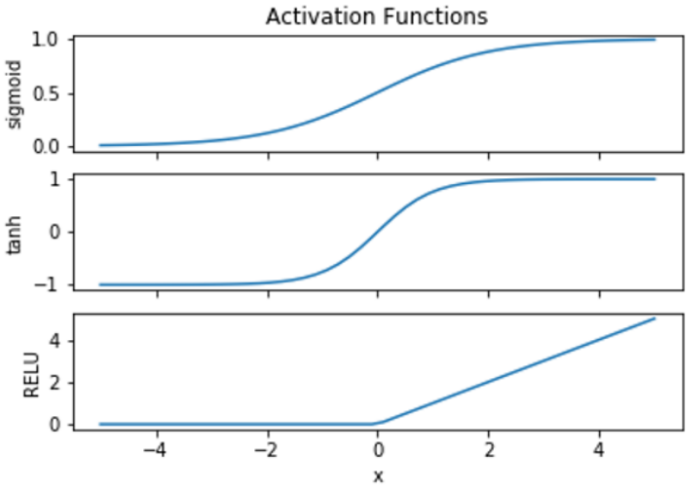

Figure 3: Activation functions: sigmoid, tanh, ReLU (Edvinsson, 2017)

A multilayer perceptron network with one hidden layer, i.e., the layer between the input layer and the output layer, is generally taken as a universal approximator. We assume that any continuous function can be represented with such architecture. Each neuron in the layer can divide the space into two parts. A neural network with no hidden layer can be used to represent simple binary logical operations such as disjunction or conjunction. The

hidden layer already allows us to learn a linearly inseparable problem. The more neurons in a given layer, the more complex function can be represented with a given layer.

2.1.3 Error backpropagation

The error backpropagation algorithm is the most important and used algorithm for supervised neural network learning. Therefore, the input and output value pairs must be known. The error propagates back across all layers to the first layer. Learning, according to this algorithm, takes place in three phases.

- In the first phase, the input is presented. The neurons of the individual layers of the network respond to this input, gradually from the input layer to the output layer. Once the network returns the output values, an output error can be computed.
- In the second phase, error information is propagated back from the output layer. The error of neurons in the hidden layer is determined by the sum of the errors of the neurons of the next layer multiplied by the corresponding weights. There is no need to consider the error for the input layer because the input layer only distributes the input values.
- In the third phase, when the error is already known for each neuron, it is possible to adapt the weights according to the learning rule. First, we need to compute the gradient of the output error (cost function) with respect to all weights (∇J). Then, we can use these gradients to update the weights of the network.

$$\text{Loss function: } \nabla \mathrm{J} = \left(\frac{\partial \mathrm{J}}{\partial w_1}, \frac{\partial \mathrm{J}}{\partial w_2}, \dots, \frac{\partial \mathrm{J}}{\partial w_l}\right) \quad \text{Weight update: } \Delta w_l = -\alpha \frac{\partial J}{\partial w_l}$$

The learning cycle consists of the individual iterations described above. Each pattern is submitted to the network exactly once during the cycle. Experience has shown that where training patterns are independent of each other, it is not appropriate, as already mentioned, to present the patterns in the same order. This is because the network could find unwanted dependencies in repetitive sequences.

Unfortunately, backpropagation has several unpleasant features. Above all, it is a fact that the error function is dependent on all weights, and thanks to that, it is a very complex function, i.e., it has many local minima. The gradient method can lead to the nearest minimum, which may not be global. The second problem is the number of learning

parameters, also called the hyperparameters. The algorithm does not determine that, and the successful convergence of the error function depends on them. The appropriate setting of these parameters can significantly affect the learning performance. This method converges relatively slowly, especially for large weights, where nonlinear activation function-based changes are very small.

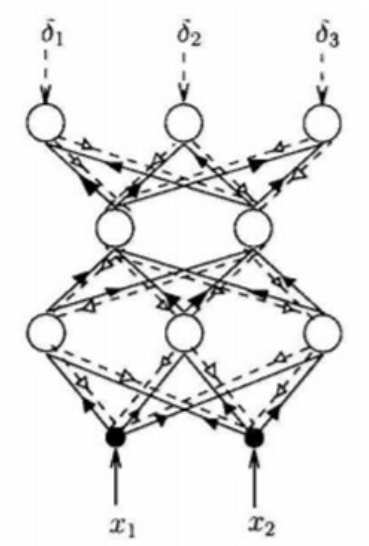

Figure 4: Error backpropagation (Rumelhart, Hinton, & Williams, 1985)

In Figure 4, there are the schematics of the error backpropagation algorithm. The solid arrows indicate the direction of signal propagation, while the dashed direction of error propagation. The improvement of the algorithm with backward propagation of the error consists in the introduction of "inertia" in the changes of the weights, where the change of the weight also depends on the size of the previous change of the weight. Originally, the stochastic gradient descent (SGD) method was used for the update of the weights.

2.1.4 Optimizers

In order to update the weights of the network, we can use various optimization algorithms. Nowadays, modern optimizers are used instead of classical stochastic gradient descent. One of the most popular ones is Adam (Kingma & Ba, 2014). Because it achieves higher accuracy than stochastic gradient descent across multiple tasks, Adam is very popular in the field of Deep Learning.

Unlike the SGD, which maintains a single learning rate (alpha -) during the whole course of training. Adam adapts the learning rate as learning unfolds. It is a combination of older extensions to the SGD, namely the AdaGrad (Duchi, Hazan, & Singer, 2011) and RMSProp (Tieleman & Hinton, 2012).

The Adaptive Gradient Algorithm (AdaGrad), maintains a specific learning rate for each parameter (weight). This brings improvement in problems with sparse gradients (e.g., computer vision problems and natural language).

The Root Mean Square Propagation (RMSProp) also maintains per-parameter learning rates. These separate learning rates are adapted as the average of recent magnitudes of the gradients for the particular weight, so the learning rate depends on how quickly the concrete weight is changing. Like this, the algorithm can do well in case of the online and noisy non-stationary problems.

2.1.5 Image processing

For image recognition and classification, fully connected neural networks described above may not be the best solution. In order to be able to work effectively with a large set of different input images, it would be necessary to make the network very deep, with an unreasonable number of neurons. Although it is possible to arrange this, we still encounter the fundamental limitations of classical multilayer perceptron - individual neurons learn in isolation from other neurons, while at the input, we have a real-life image where pixels are semantically grouped. Thus, it would be more advantageous to focus on improving the architecture of the neural network itself, specialized in having a bitmap at the input, and thus neighboring neurons should somehow share their weights at the inputs. Modern trends are in the use of convolutional neural networks, which were designed primarily for working with image data. We devote the latter part of this chapter to deep and convolutional neural networks.

2.2 Deep learning and Convolutional neural networks

Deep learning is often applied in very deep feed-forward neural networks (DNNs), and it is used as an alternative to standard machine learning techniques for working with large data. The main aim of using deep networks is to discover representations that can help feature detection and classification of the data. Deep learning seems to be better than conventional algorithms or other machine learning techniques at discovering high-level structures in multi-dimensional data. Therefore, it can outperform task-specific algorithms in the area of image recognition (Krizhevsky, Sutskever, & Hinton, 2012), speech recognition, natural language processing systems, drug design, and other (Deng & Yu, 2014). Steinkraus, Simard, & Buck (2005) proposed to use the graphical processing unit (GPU) instead of a central processing unit (CPU) for machine learning tasks. DNNs implemented on a GPU provided incredible performance results compared to CPU processing (Ciresan, Meier, Masci, Gambardella, & Schmidhuber, 2011).

Further progress in the area of image processing and ANN has led to developing Convolutional Neural Networks (LeCun et al., 1989). CNN's were inspired by biological processes in the brain related to processing visual content. From the visual neuroscience, there are simple cells and complex cells (Hubel & Wiesel, 1962). Therefore, the CNN structure is divided into various layers that can have different architecture. CNN tries to mimic the architecture of the visual cortex, which is defined by areas in the brain: V1, V2, V4, and IT (Felleman & Van, 1991). Each area is specific and serves a slightly different role in the whole hierarchical process of image recognition, which is well illustrated in Figure 5. Similarly, CNN's are designed to process 2D data in many neural network layers that perform hierarchical detection of image features.

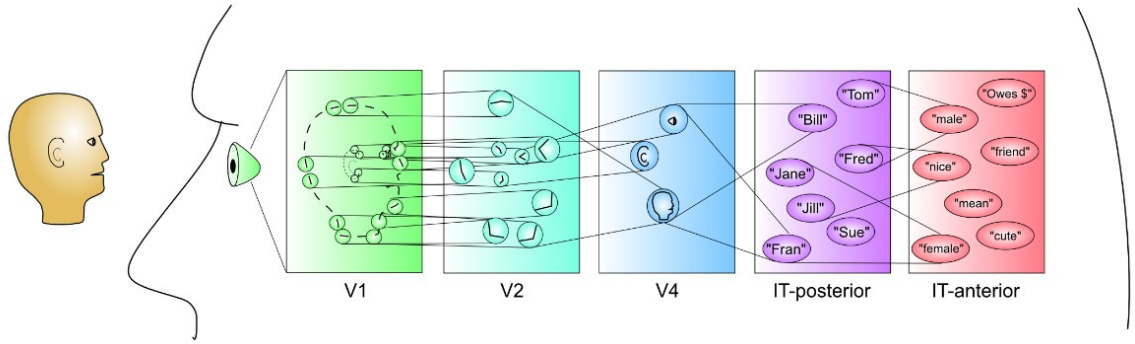

Figure 5: Image processing by humans (O'Reilly & Munakata, 2000)

2.2.1 Convolutional neural networks

As well as the standard feed-forward neural networks, CNN consists of an input and an output layer and many hidden layers. The difference with CNN is that it is designed for the extraction of features. Hidden layers of the CNN consist of convolutional layers (CONV), pooling layers, and some layers that can be fully connected (FC). The convolutional layer, which has sparser connectivity, then the standard FC layer, processes the input and passes it to another layer. The pooling layer consolidates the output from separately processed parts of the previous layer. Typically, the ReLU is used as the activation function for these special layers in order to minimize computational demand. Fully connected layers are interconnecting every neuron from one layer to another layer and usually are placed at the output of the whole architecture to perform classification or another task. Here, standard activation functions such as sigmoid or softmax are used.

According to Deng & Yu (2014), the main ideas behind CNN are that they "take advantage of the properties of natural signals: local connections, shared weights, pooling, and the use of many layers". Due to the economy of the convolutional layers which have much fewer weights than the fully connected ones, we can process much bigger images with less computational effort. For example, if we process only 22-by-22 pixel colorful images, in case of fully connected networks with the same amount of neurons in each layer, every layer will have 1452 neurons and exponentially more weights. In the case of real image processing with medium and high-resolution images, CNN's have become a necessity.

Based on Figure 6 below, it is clear that the convolutional neural network is made up of individual layers, each with different properties and functions. The main building blocks unique to CNN's are convolution layer, pooling layer, and batch-normalization layer.

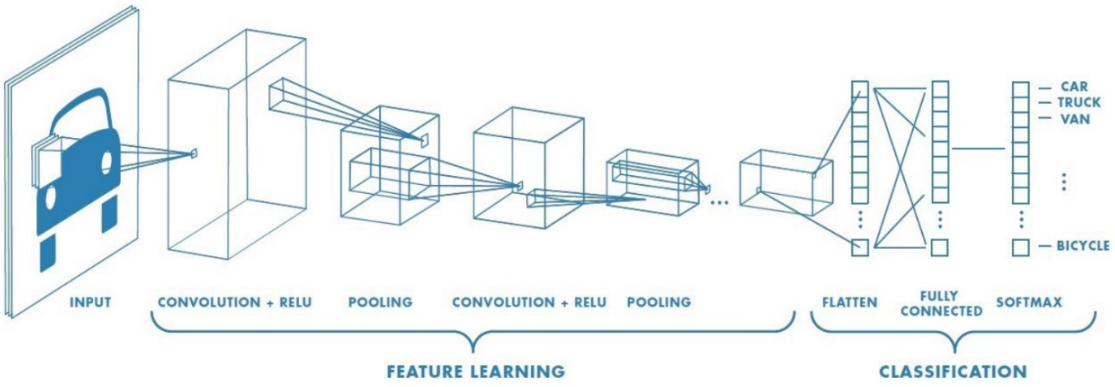

Figure 6: Convolutional neural network architecture (Saha, 2018)

2.2.2 Convolution layer

The convolution layer (CONV) applies a filter to the input image. We can define the size of the filter in advance, for example, 6 x 6 x 3. The first two values refer to the spatial size of the filter, and the third value refers to the number of neurons or "depth" of the filter. The first two values can be thought of as the size of the receptive field of the neuron. The filter moves over the input space, and at each location, it outputs some activation value that is passed to the next layer. For ease of understanding, this procedure is illustrated in the figure below. How much each convolutional filter moves during the convolution procedure is controlled by a parameter called stride. If the stride parameter is set to a number greater than 1, the spatial size of the input to the convolutional layer will be reduced when it passes through the convolutional layer.

For the most part, however, the input is passed through multiple filters simultaneously. For example, there are several different filters, and each one forms a two-dimensional activation map. Thanks to the Figure 7: CNN sequence to classify handwritten digits , we can observe how the number of activation maps in individual layers increases, and at the same time, the maps decrease. The reason is both the stride parameter of the filter and the use of a pooling layer.

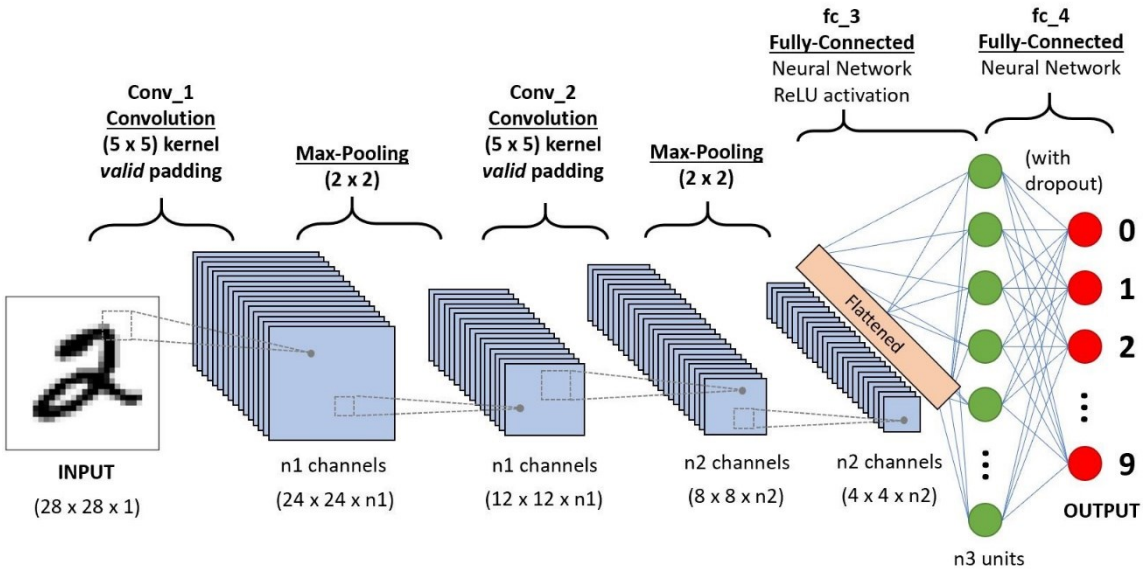

Figure 7: CNN sequence to classify handwritten digits (Saha, 2018)

Many convolutional layers stacked together are the basic building blocks of modern neural networks used in Deep Learning.

2.2.3 Pooling layer

The task of the pooling layer is to progressively reduce the size (spatial dimension) of represented data and thus reduce the number of activations passed to the next layer. The pooling layer works independently of the input depth. The most often used size of the pooling layer is 2x2. Up to 75% of the input data is reduced. The pooling operation is presented in Figure 8, whereby applying the pooling, the size of the input is reduced by 75%.

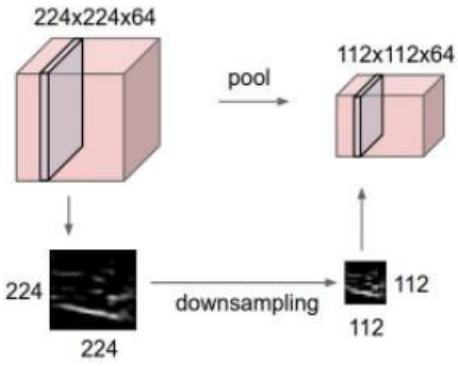

Figure 8: Pooling layer of the image

This layer works similarly to the convolution layer, where a filter travels over the entire image, which selects a suitable output value from a given matrix of points. The output value is determined based on the required function, either the maximum is selected from the given values (max-pooling) or the mean value (average-pooling) as shown in Figure 9.

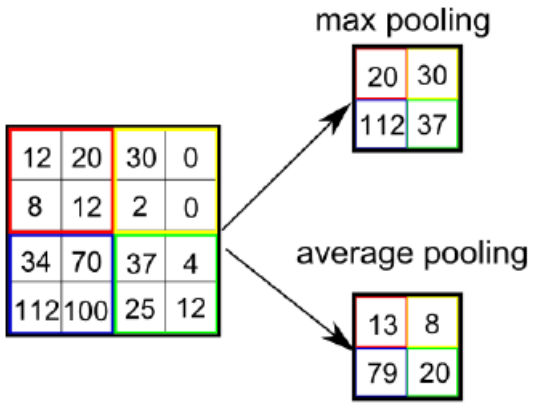

Figure 9: Pooling function: max pooling or average pooling (Saha, 2018)

2.2.4 Batch normalization layer

The batch normalization layer is normalizing the input layer, usually hidden layers in the CNN to speed up the learning and increase the stability of the neural network (Ioffe & Szegedy, 2015). Batch normalization reduces the covariance shift applied to the hidden layer values. Regularization is a side effect of this algorithm.

First, the batch normalization algorithm computes the minibatch mean and minibatch standard deviation. The batch normalization updates the output of a previous activation layer by subtracting the minibatch mean and dividing by the minibatch standard deviation. The batch normalization adds two trainable parameters gamma and beta to each layer. Then, the normalized output is multiplied by parameter gamma and add parameter beta.

3. Convolutional neural networks for face recognition

CNN's are widely used for face recognition problems. These include identification of people, segmentation in images, classification of emotions, detection of gender and age, and many more.

3.1 Face recognition

The conventional face recognition process consists of 4 stages: face detection, face alignment, feature extraction, and classification. According to (Hu, et al., 2015), the most critical stage of face recognition is feature extraction. It remains an open problem to find common facial features in an image, which are improving the accuracy of face recognition in unconstrained environments. This problem is mainly related to the diversity of the images and the background environments on them. If we focus on the facial features on the image, where colors are bright, it is easy to recognize the common features like eyes, nose, and mouth. In more challenging conditions and improper contrast, the common facial features cannot be clearly identified, while other features can be used to categorize or identify the face on the image.

The most challenging database for benchmarking of automated recognition systems is Labeled Faces in the Wild (LFW), which contains images of faces in various background environments (Huang et al., 2007).

Hu et al. (2015) compared the performance and accuracy of several CNN models and tried to invent a better design of CNN for face recognition problems. Sun, Wang, & Tang (2014) exploited a novel approach to fuse multiple networks which have been named DeepID. Hybrid or multiple networks have emerged, and they lead to impressive results with accuracy better than 90%, e.g., High Dimension Local Binary Patterns (Chen, Cao, Wen, & Sun, 2013).

In recent years, improvements in deep learning techniques, computing power utilizing GPUs, and accumulating large training datasets lead to the evolution of CNN architecture complexity, speed of response, and accuracy of the results. Naïve Deep Face Recognition (Zhou, Cao, & Yin, 2015) presented Megvii Face Recognition System, which

achieved 99.50% accuracy on the LFW benchmark dataset. The main reason why they provided such remarkable accuracy was due to the high amount of training images that were gathered from the internet. New social media; Facebook, Instagram contains many images of known persons, and they are correctly categorized.

3.2 Age and gender classification

Another area within the face recognition domain is age and gender classification. The problem of automatically extracting age-related attributes from images with faces has received increasing attention in recent years. The first methods of age estimation were based on calculating ratios between different measurements of facial features. Common facial features (eyes, nose, mouth, chin, ears, etc.) need to be located on the image, and their size and distance between them computed. Then, predefined rules are used to estimate age from ratios between facial features.

One of the early methods for gender classification (Golomb, Lawrence, & Sejnowski, 1990) used a neural network trained on a small set of face images. More recently, Ullah et al. (2012) used the webers local texture descriptor (Golomb & Sejnowski, 1995), (Jabid, T, Kabir, & Chae, 2010) for gender recognition, demonstrating near-perfect performance.

On top of the recent achievements, there have been created a very comprehensive model for image recognition (Ranjan et al., 2017), which provided age, gender estimation, smile detection, and face recognition. It was supported by the US government agencies and provided very accurate results. The processing time of one image with the trained model took on average 3.5s. The main bottleneck in the model is the process of generating region proposals and passing each of them through the CNN - first stage.

3.3 Ethical concerns of using face recognition technologies

Earlier on 18th January 2020, information spread around the world about the inconspicuous Clearview AI start-up, which collected several billion photos of people from social networks, which they offered to police officers in search of facial identification technology. The United States government has been enthusiastic about this information, as it makes

their work much easier and faster. For example, when they look for a thief from a store whose face is recorded on an industrial camera but does not match any records in official databases, Clearview AI will help authorities to determine who it is. It can find his face in pictures from YouTube, Facebook, or Twitter and connect him with profiles on these networks (Hill, 2020).

According to Martin Urban, CEO of Eyedea Recognition "I cannot imagine such a thing in Europe yet, the protection of personal data, including a photo, is at a different level here. The police would not even be able to legally obtain such a tool here" (Urban & Zandla, 2018).

Facial recognition has significantly improved and accelerated over the last three years, due to the development of deep neural network technology. As stated by Hill (2020), It would be technically extremely demanding and expensive that cameras will be watching us everywhere and comparing online footage with databases and our profiles on Facebook. It was not mentioned that the most probably we will face this problem in the near future.

On the other hand, it is important to note that face recognition is nothing new, people have been commonly encountered at airports for many years, and the police have been trying to pair industrial camera footage with their footage for criminals or lost children for at least two decades. Nevertheless, with the development of artificial intelligence and high-resolution cameras equipped with an internet connection, the question of ethics and ubiquitous snooping is fully opening up. A warning example is China, which makes no secret of the fact that it is building nationwide camera surveillance, and which, according to some estimates, has already ahead of the US in the development of artificial intelligence.

The highest deployment rate of cameras with facial recognition technology is in a Chinese province in the west of the country inhabited mainly by the Uyghur minority. In the Xinjiang region, there are the same number of cameras per ten thousand inhabitants as in other regions of tens of millions of people. According to calculations by the analytical company IHS Markit, China holds 46% of the global CCTV market. "Today, there are 176 million industrial cameras in China. In comparison, there are 50 million in the US. There are already 200 million in China by 2020. A large part of them will be located in Xinjiang," says Russian sinologist Leonid Kovacic (Light & Kovachich, 2020).

The police databases contain photographs of all the inhabitants of the province, and the register is connected to the facial recognition system. The system can monitor everyone within range of the cameras and watch when someone changes their daily routine.

Bloomberg reported that artificial intelligence itself would alert the police if a person identified as a "person of interest" deviated from his route home to work by more than three hundred meters. Moreover, they do not have to be criminals or recidivists, and they can also be human rights activists or practicing Muslims (Drozdia, 2020).

At the end of 2018, the Karel Čapek Research Center for the Study of Values in Science and Technology, established by Charles University and the Academy of Sciences, was established precisely because of ethical and legal issues associated with the development of modern technologies.

There is an agreement in professional circles on three basic rules for dealing with it. The first is the principle of commitment in the public interest and respect for human rights. The second is the principle of the least possible interference, especially in people's privacy, and the last is the principle of proportionality. Here it is thought that once there are civilian victims, it must be redeemed by some important strategic goal. By analogy, if we are already interfering with someone's privacy, there must be a profound reason (Jirouš, 2019).

The principles that Černý mentions are ignored in places where civil society is either not present or the regime is working to weaken it. In addition to China, these are India, the states of the former Soviet Union, or several countries in Africa and Latin America. Developers in Russia are very active. In June 2016, visitors to the Alfa Future People dance festival in Moscow received a message from the organizers to view the photos on which they are displayed. It was one of the first public presentations of FindFace technology developed by NtechLab. She examined tens of thousands of photos from the festival created by organizers or visitors, recognized faces, and then linked them to the profiles of users of Russia's largest social network VKontaktě. The creators of FindFace technology won the 2015 The MegaFace Benchmark competition organized by the University of Washington, and a year later, they took first place in the Facial Recognition Vendor Test, compiled by the American National Institute of Standards and Technology. In the Kremlin, they want to fight against unauthorized demonstrations with the help of facial recognition (Light & Kovachich, 2020).

This is a key issue because of the spread of artificial intelligence is closely linked to security. And not just because facial recognition technology helps find terrorists or prevents thugs from entering sporting venues. It is also a question of national security - Chinese companies, such as Huawei, are leading suppliers of computer technology and, at the same time, are closely linked to the local communist government and play a leading role in

China's state-sponsored development of artificial intelligence. For Beijing, this is one way to spread its influence around the world (Thompson, 2019).

A positive example is Ecuador, where crime has fallen by 24 percent, making it one of the safest countries in Latin America, China's Xinhua agency quoted deputy chief of staff os system ECU 911. The system ECU 911 was developed by CEIEC and donated 14 million US dollars free of charge in 2016 to the Ecuadorian government. CEIEC is a Chinese government corporation that manufactures electronics for military and security use. In recent years, it has been actively offering its face recognition equipment around the world.

Nevertheless, CEIEC does not succeed everywhere; they met with resistance in Kyrgyzstan, for example. Last spring, CEIEC signed an agreement with the local Ministry of the Interior to supply eighty cameras, and it was later revealed that CEIEC had donated the system to Bishkek. Given that Kyrgyzstan borders China's Xinjiang Province, where China has built a police "state" to monitor Uyghurs, local people have a good idea of how the use of a facial recognition system can turn out (Hill, 2020).

4. Few-shot learning

Few-shot learning was first formulated as a classification or a verification task in computer vision in which we have only a limited amount of images per each class. There is a subset one-shot learning where we have only one image per category. In few-shot learning, we usually use 3 or 5 images per category. These tasks are challenging to accomplish with a high level of accuracy. On the other hand, humans can perform a one-shot task with high precision. (Lake, Salakhutdinov, Gross, & Tenenbaum, 2011). In this thesis, we focus only on the classification and verification of images. However, modern few-shot learning research also involves tasks such as image segmentation or object detection.

4.1 Few-shot learning models

We can approach the few-shot learning problem in different ways. Usually, deep learning networks with backpropagation mechanisms using gradient descent learning mechanisms are used. This was explained in the previous chapter. In order to classify new classes not seen during training based only on a few examples, an existing neural network architecture must be taken and adapted to accommodate this new task (Snell, Swersky, & Zemel, 2017). The model can be trained on a particular task with a selected set of data that will not be used for testing, or it can even be trained on a different task. This approach is referred to as transfer learning, and it is a very common practice in the image classification domain. Transfer learning can be done with the same pre-trained model just by adapting the last layers of the architecture or by so-called model fine-tuning in which the pre-trained model is further trained on the current dataset.

In order to classify a *query* image into one of the classes, a model is given a *support set* of example images (few-shot) or a single example (one-shot) for every class. The number of examples per class is usually denoted as *the shot* and the number of classes as *the way* (e.g., 1-shot 15-way training). For the image classification task, we split the dataset based on classes into distinct datasets. One for training and one for testing. The testing dataset cannot contain images from classes in the training dataset and vice versa.

In the following sections, we are describing the two chosen neural network models for the few-shot learning task. There are many more models. For instance well known are the Matching networks model which was proposed by Vinyals et al. (2016), MAML networks (Finn, Abbeel, & Levine, 2017), REPTILE networks (Nichol, Achiam, & Schulman, 2018) and Relational networks (Sung, Yang, Zhang, Xiang, & Torr, 2017).

4.2 Prototypical neural networks

The Prototypical neural networks (PNN) were designed by Snell and colleagues (Snell, Swersky, & Zemel, 2017). From the interdisciplinary point of view, they can be seen as an implementation of the prototype theory of categorization (Rosch, 1973). According to this theory, categories are formed around some more typical members than others (see Figure 10). In this principle, the classification of an example can be presented as a single number showing the degree of membership to the class. The PNN model learns an embedding function using a sub-network. Then the PNN builds a prototype out of the given example set. Then it computes the distance between the embedding of the query image and the prototype of the class. Based on these distances, the best-matching class is selected.

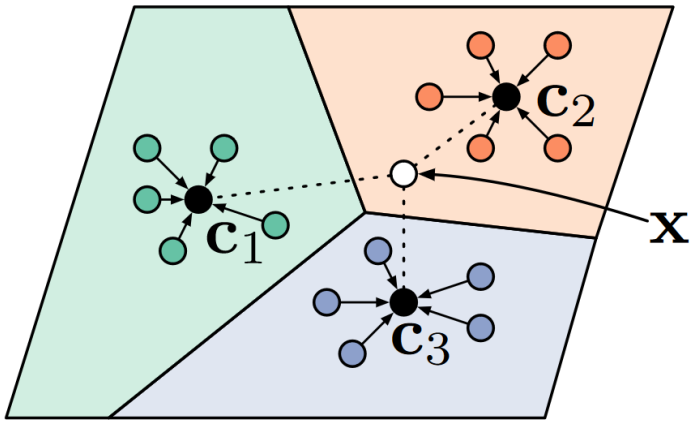

Figure 10: The Prototypical network classification principle (Snell et al., 2017)

As visualized in Figure 10, the Prototypical networks classification principle from Snell et al. (2017). C_1, C_2, and C_3 represent prototypes for a given category. These prototypes are constructed as a mean of the embeddings of support examples. Query example X is then classified by calculating an Euclidean distance to the prototypes and then setting the predicted category to the category of the closest prototype.

As described by Snell et al. (2017), Prototypical networks are computing the so-called prototypes from the output of the embedding part of the architecture $f_{\emptyset}$, which is a convolutional neural network. Prototypes denoted as c_k are computed as means of embeddings from a small set of examples S_k from the particular class k:

$$c_k = \frac{1}{|S_k|} \sum_{(x_i) \in S_k} f_{\emptyset}(x_i)$$

The training algorithm (pseudocode), which was presented by Snell et al. (2017), is outlined in Figure 11, where N denotes the number of examples in the training set, K denotes the number of classes in the training set, $N_C \leq K$ denotes the number of classes per episode, N_S denotes the number of support examples per class, N_Q denotes the number of query examples per class. RANDOMSAMPLE(S, N) is a function that creates a set of N elements randomly chosen from set S, without repeating items.

Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where each $y_i \in \{1, \ldots, K\}$. $\mathcal{D}_k$ denotes the subset of $\mathcal{D}$ containing all elements $(\mathbf{x}_i, y_i)$ such that $y_i = k$.
Output: The loss J for a randomly generated training episode.

$V \leftarrow$ RANDOMSAMPLE$(\{1, \ldots, K\}, N_C)$ ▷ Select class indices for episode
for k in $\{1, \ldots, N_C\}$ **do**
  $S_k \leftarrow$ RANDOMSAMPLE$(\mathcal{D}_{V_k}, N_S)$ ▷ Select support examples
  $Q_k \leftarrow$ RANDOMSAMPLE$(\mathcal{D}_{V_k} \setminus S_k, N_Q)$ ▷ Select query examples
  $\mathbf{c}_k \leftarrow \frac{1}{N_C} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_{\phi}(\mathbf{x}_i)$ ▷ Compute prototype from support examples
end for
$J \leftarrow 0$ ▷ Initialize loss
for k in $\{1, \ldots, N_C\}$ **do**
  for $(\mathbf{x}, y)$ in Q_k **do**
    $J \leftarrow J + \frac{1}{N_C N_Q} \left[d(f_{\phi}(\mathbf{x}), \mathbf{c}_k)) + \log \sum_{k'} \exp(-d(f_{\phi}(\mathbf{x}), \mathbf{c}_k)) \right]$ ▷ Update loss
  end for
end for

Figure 11: Pseudocode for the Prototypical networks training procedure (Snell et al., 2017)

Classification is performed in the few-shot scenario by finding the nearest prototype class for an embedded *query* point. In practice, it means that we have to go through every query item and compute Euclidean distance to each prototype class and then choose a class that has the lowest distance between its prototype and our query item embedding.

4.3 Siamese neural networks

A Siamese Neural Network (Bromley, Guyon, LeCun, Sackinger, & Shah, 1993) is a class of neural network architectures that contain two identical sub-networks. Identical here means they have the same configuration with the same parameters and weights. Parameter updating is mirrored across all sub-networks. Two sub-networks output feature vectors that are concatenated to form an input to subsequent classification layers that will output the final output of the network. The final output of the network represents the degree of class similarity between the two inputs to the network. If the final output of the network is close to 1, that means that network "judges" the two inputs as belonging to the same class. If the output is close to 0, the inputs should belong to different classes. See in Figure 12.

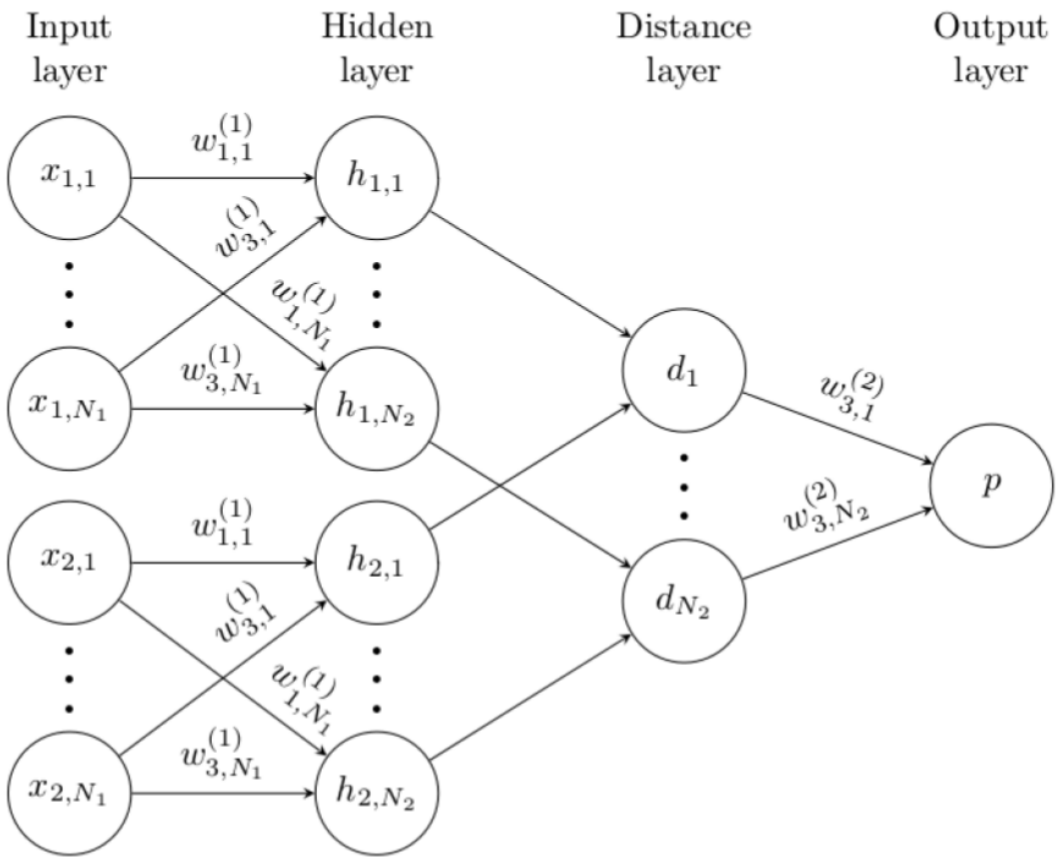

Figure 12: Schema of the Siamese neural network. The output is representing a probability of inputs from the same class (Koch et al., 2015)

The Siamese network learning task is, therefore, reduced to a binary classification learning task where the pairs of the input data are classified either as belonging to the same

class or not. Unfortunately, classification within many classes is not implemented in the Siamese network; therefore, we need to perform the classification by finding the similarity between the support examples of different classes and the query example that we want to classify. Then we assign the query example the category of support example that has the highest similarity to query example.

With the Siamese network, we can perform the verification task by comparing two images, and the output is the probability that the images are from the same class.

5. Few-shot face recognition

We have focused our experiments on the few-shot learning problem utilizing Prototypical neural networks (PNN) and Siamese neural networks (SNN). We have directed mainly on the classification task, and later we have performed a verification task using SNN. We have created a dataset for few-shot tasks as it is described later in this chapter. In order to compare the result, we have used the same architecture for embedding sub-network of both PNN and SNN.

We have performed a classification task and a verification task. The classification task was measured if the *query* example was properly assigned to the class from the list of classes we have provided. The verification task was much more straightforward as we need to choose if the query image is from the same class as the supported image, which is a binary classification.

5.1 Dataset

There are multiple datasets in the face recognition task. Among the most popular face recognition datasets are the Labeled Faces in the Wild (LFW) dataset created by Huang et al. in 2007 and the CelebA dataset (Liu, Luo, Wang, & Tang, 2018).

Finally, we have chosen to use the Labeled Faces in the Wild (LFW). This dataset contains a large number of categories, which is ideal for few-shot image recognition and is considered a benchmark dataset in the image recognition domain (Schroff, Kalenichenko, & Philbin, 2015).

Within the LFW dataset, there are several preprocessing steps applied to the images. We have chosen the dataset where the images were preprocessed by the automatic alignment that is called deep funneling (Huang, Mattar, Lee, & Learned-Miller, 2012) - LFW deep funneled images. For our task, where we will introduce to the network only a few images as support images in order to provide proper results. Below is a short example of images from the dataset which have been preprocessed by deep funneling see Figure 13.

Figure 13: Images from LFW dataset - deep funneled images

The resolution of the images in the LFW dataset is 250x250 pixels and are in the RGB color space. For training purposes, the images were resized to 84x84 resolution while preserving all three color channels. The dataset contains many categories; each category represents a different person. We removed all categories with less than six images so that we can select at least three queries and three support images for every category. Our reduced dataset contains 311 categories and 5425 unique images. We have divided the dataset into two distinct datasets. The training dataset contains 277 classes with 4679 images, and the testing dataset contains 34 classes with 746 images. The important fact of the datasets is that the testing dataset was separated from training one, and testing images were never used during training.

5.2 Prototypical network for face recognition

We tested the application of prototypical networks to few-shot face recognition using various numbers of classes ("way") and a various number of support examples per class ("shot"). We have chosen 1, 3, and 5 shots for our experiments and 5, 10, 15, 30, 60, 90 for a number of classes – ways.

An important part of the training process is the preparation of the training batches. In each training episode, we randomly select a certain number of training classes from all available training classes. For example, if we are training our network in the 5-way setting, we randomly select five classes from all 277 training classes. Then we select a certain number of support examples for every training class. For example, in the 5-shot setting, we select five support images for every training class. Subsequently, for every training class, we select some query examples that we wish to classify. These query examples should naturally be different from the support examples. Finally, we construct a label for all query examples in this training episode. We do this by simply assigning a random label according to the number of training classes in the episode. For example, if we have ten training classes, the classes will be assigned labels from 0 to 9. The same batch preparation procedure is applied to testing batches, but the classes are sampled from testing classes. Example of the expected label vector for 3-shot 10-way:

Output Label = [0, 0, 0, 1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5, 5, 6, 6, 6, 7, 7, 7, 8, 8, 8, 9, 9, 9]

During the training, we periodically tested and logged the performance of our network. Testing was performed in a 30-way setting (30 classes per testing episode), and it was repeated 100 times.

5.3 Siamese network for face recognition

We have implemented the Siamese network from the Koch et al. (2015) proposal, and the process for training was as follows: For every training episode, we randomly select several images from the training part of the dataset regardless of the category of the images. Then for every selected image, we select another image from the training part of the dataset. With a 50% probability, we select an image from the same category, and with a 50% probability, we select an image from a different category. The goal of the model is then to classify the pairs of images into two classes: pairs of images belonging to the same class and pairs of images belonging to a different class. The labels reflect this binary classification task. Below we provide an example of a label vector for 3-shot and 10-way.

Output Label = [1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0]

During the training, we tested and logged the performance of our network periodically. Verification accuracy is computed by selecting batches using the process described above, but we sample the data from the testing part of our dataset. The output of the network is rounded to the nearest integer, i.e., to 0 or 1. When the network outputs 0, the network "thinks" that the pair of input images belong to different categories. When the network outputs 1, the network "thinks" that the pair of images belong to the same categories. We compare the labels to the rounded output of the network to get the verification accuracy.

Classification testing is much more complicated. First, we have to create *support* images, and *query* images list the same way as in prototypical networks. E.g., 3-shot 30-way. The query and support lists cannot contain the same images from the same class. Then the classification is performed by comparing the query image (the image that we want to classify) to every support image. The support-query image pair with the highest output is used to determine the class of the query image. If we use more than one support image per class, we average the outputs of the network for support images that belong to the same category. This process needs to be repeated for every query image separately.

The Siamese classification testing procedure is much more computational extensive that in the Prototypical networks.

5.4 Architecture

We have used the same architecture for embedding sub-network for both models. The architecture is presented in Table 1. Siamese network architecture contains a fully connected network which has input from above embedding sub-network. The architecture of the Siamese fully connected network, which creates output value, is in Table 2.
Prototypical network computes from embedding sub-network euclidian distance layer for every query and support. Then the softmax layer is choosing the prototype with the lowest distance to the query input. The number of outputs is the same size as the number of classification categories.

Table 1: Architecture specification for embedding sub-network

Layer name	Type	Number of neurons	Kernel size	Stride	Activation function
conv_1	Convolutional	64	3x3	1	ReLU
bn_1	Batch norm	64	-	-	-
mp_1	Max Pooling		2x2		
conv_2	Convolutional	64	3x3	1	ReLU
bn_2	Batch norm	64	-	-	-
mp_2	Max Pooling		2x2		
conv_3	Convolutional	64	3x3	1	ReLU
bn_3	Batch norm	64	-	-	-
mp_3	Max Pooling		2x2		
conv_4	Convolutional	64	3x3	1	ReLU
bn_4	Batch norm	64	-	-	-
mp_4	Max Pooling		2x2		

Table 2: Architecture specification for the Siamese fully connected network

Layer name	**Type**	**Number of neurons**	**Activation function**
fc_1	Fully connected	512	ReLU
fc_out	Fully connected	1	Sigmoid

6. Experiments and results

6.1 Implementation

The experimental part of the work was performed on the LFW dataset with deep funneling and used the Prototypical networks and the Siamese networks models, as it was described in chapter 6. Both models were implemented in the Python programming language using the Pytorch library. Pytorch library is a Deep Learning library that utilizes the CUDA toolkit to accelerate both inference and learning computations using graphical processing units (GPUs).

6.2 Hyperparameters

As described in the previous chapter architecture of embedding sub-network were shared between Prototypical and Siamese network. Siamese network has a difference that the output from sub-network, which is creating the feature vectors per each image. This is then used as input to a fully-connected linear layer, which provides the final output value.

The following hyperparameters were used for the training of the network.

- Learning rate - α = 0,0001

  We have tested with various values e.g. 0,001 , 0,0001 and 0,00001, but in final experiments we used value: $\alpha = 0{,}0001$.

  Adam optimizer was used in all experiments, and it was initiated by the proposed learning rate while Adam was updating the learning rate during the training (Kingma & Ba, 2014)

  Hyperparameters:

  $\beta_1 = 0{,}9$ $\beta_2 = 0{,}999$ $\varepsilon = 10^{-8}$

- Number of episodes = 20 000 or 60 000

  We have used this number of episodes based on training results. For Prototypical networks, there was enough to train in 20 000 episodes while in Siamese networks, we had to train more; therefore, we have used 60 000 training episodes.

- Classes per testing = 30

  We have used this parameter as the main parameter to keep the results from both networks in a comparable form. During testing, we have used that amount of test classes for classification of the query images from Support images.

6.3 Prototypical network results

We have used 30 classes (30-way) for testing in all graphs presented here.

In Figure 14, we present the learning curves of the Prototypical network model based on the number of training classes per batch during the training.

Figure 14: Classification accuracy using a different number of training classes

From Figure 14, it is apparent that test accuracy saturates when we are using ten or more train classes per training batch.

We also performed experiments, where we vary the number of support examples per test batch (shot). We examine the relationship between the number of classes per train batch and the number of support examples per test batch and results are in Figure 15.

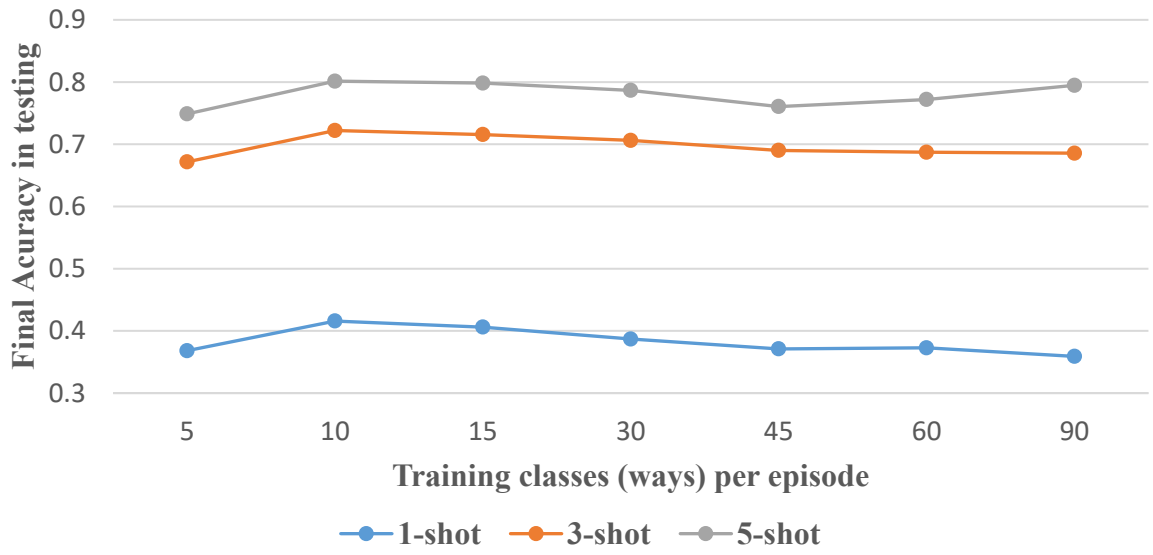

Figure 15: Final accuracy in testing for a different amount of training classes and different final test shots

In Table 3, Table 4, and Table 5, we present the accuracy for each training case 3-shot, 5-shot, 1-shot. The bold result is highlighting the best result in that category.

Table 3: Test accuracy for 3-shots during training using a different number of train classes -ways and different number of testing shots

Number of classes during train - way	**Support during the test – Shot**		
	1-shot	**3-shot**	**5-shot**
5	0.368	0.672	0.749
10	0.416	**0.722**	**0.801**
15	0.406	0.716	0.798
30	0.387	0.706	0.787
45	0.371	0.690	0.761
60	0.373	0.687	0.772
90	0.359	0.686	0.795

Table 4: Test accuracy for 5-shot during training using a different number of train classes - ways and different number of testing shots

Number of classes during train - way	**Support during the test – Shot**		
	1-shot	**3-shot**	**5-shot**
5	0.333	0.649	0.750
10	0.340	0.679	0.756
15	0.347	0.696	0.787
30	0.334	0.681	0.781
45	0.310	0.674	0.773
60	0.301	0.666	0.768

Table 5: Test accuracy for 1-shot during training using a different number of train classes -ways and different number of testing shots

Number of classes during train - way	Support during the test – Shot		
	1-shot	3-shot	5-shot
5	0.449	0.682	0.730
10	**0.473**	0.699	0.752
15	0.469	0.694	0.766
30	0.436	0.673	0.747
45	0.433	0.697	0.766
60	0.440	0.682	0.754
90	0.412	0.644	0.725
120	0.438	0.692	0.768

From the tables above, we can compose Table 6 with the best classification accuracy parameters per each shot category.

Table 6: Best classification accuracy results with hyperparameters

Number of classes during train – way	1-shot train	3-shot train	3-shot train
	1-shot test	3-shot test	5-shot test
10	0.473	0.722	0.801

6.4 Siamese network results

We have been testing the Siamese networks, and we have found that we are not able to achieve the same level of accuracy compared to prototypical networks. Therefore, we have increased the number of episodes.

The graph in Figure 16 shows the progress of test classification accuracy over the time of training using more options for the number of shots.

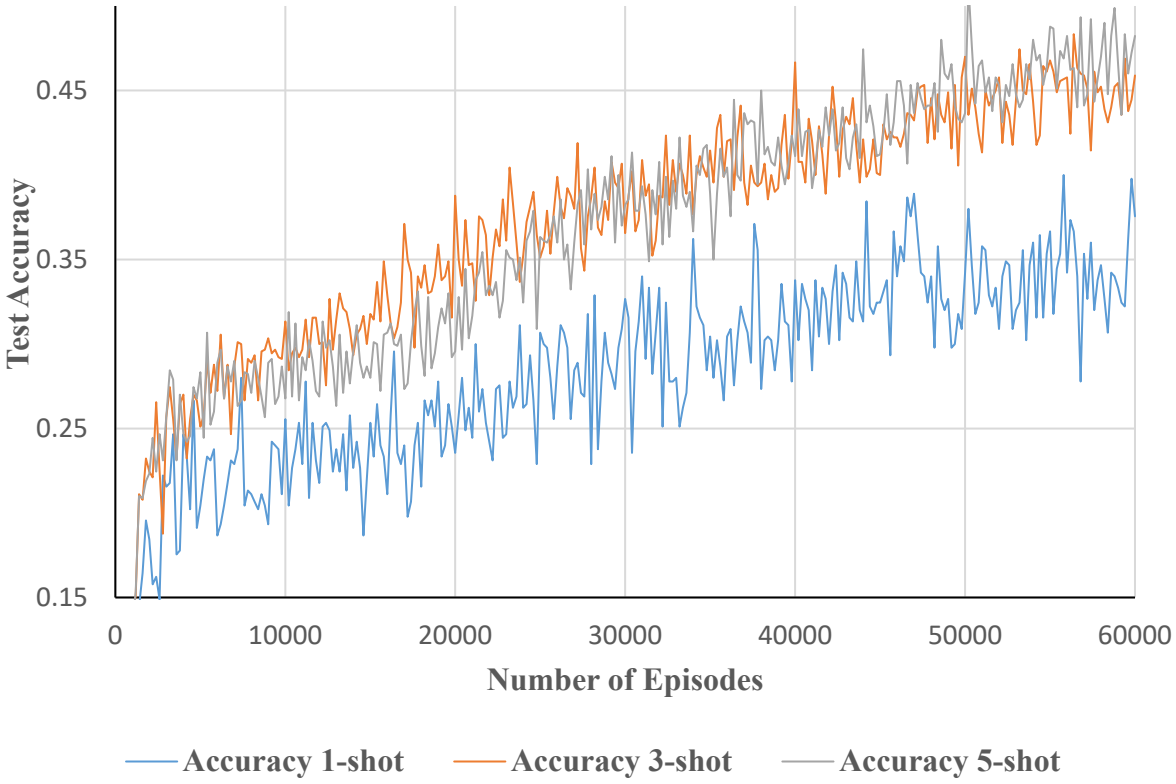

Figure 16: Classification accuracy using a different number of shots – training classes

Additionally, in Figure 17 is a graph representing testing loss over the time of training.

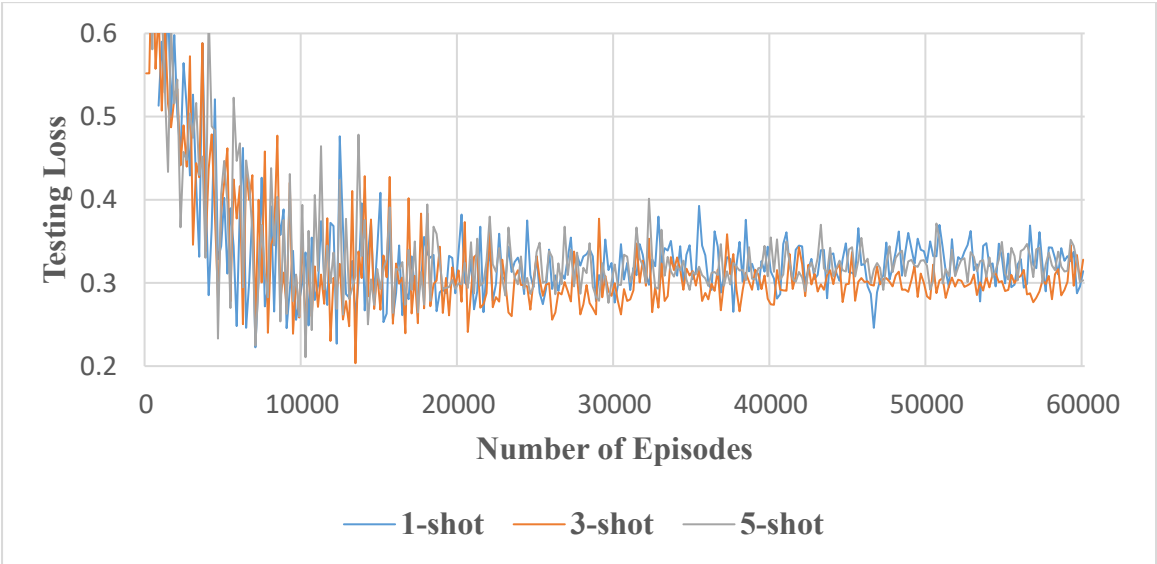

Figure 17: Testing classification loss using a different number of shots – training classes

Then we have recorded and compared the verification accuracy with a different number of support examples – shots. The graph is shown in Figure 18.

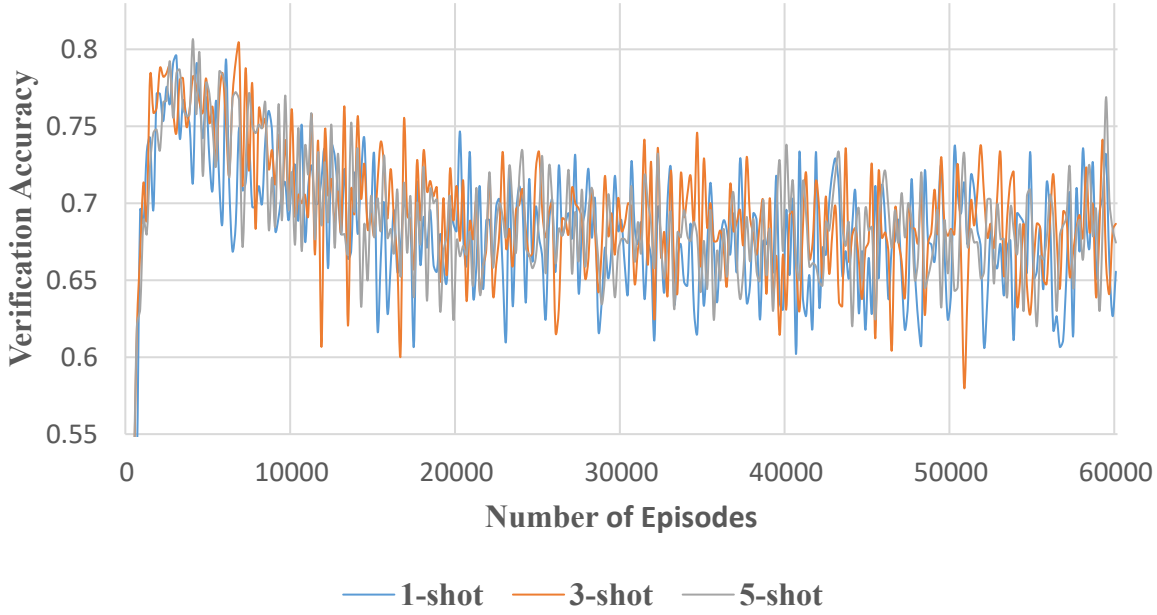

Figure 18: Verification accuracy using a different number of shots

The final testing accuracy in the Siamese network after 60 000 episodes is in Table 7.

Table 7: Final testing accuracy for classification and verification task after 60 000 episodes

	Classification	**Verification**
1-shot	39.78%	66.22%
3-shot	48.33%	74.11%
5-shot	51.22%	76.89%

6.5 Performance

We have had limited resources for computation, and during our experiments, we have utilized 185 hours of computational time for Prototypical networks and 194 hours of computational time for Siamese networks. We do not have the space to present all experiments we have executed and all the variables we have tested, but we have tried to tune the Siamese networks as best as we could. Nevertheless, the accuracy of the Siamese networks is not very impressive.

Table 8 shows computational time per each final test we have performed using the Prototypical networks.

Table 8: Duration of training for the Prototypical networks

Training Classes per Episode	Support to train - shot		
	1-shot	3-shot	5-shot
5	1:30:04	4:17:24	4:22:15
10	5:12:30	5:19:58	5:29:38
15	6:10:55	6:23:00	6:36:29
30	8:48:21	6:37:59	10:00:21
45	8:33:41	12:47:35	13:27:38
60	15:05:16	11:56:59	7:48:28
90	6:10:55	22:24:31	--
120	16:41:52	--	--
TOTAL	185:45:49	**Apx. 7.2 day**	

7. Discussion and future work

In this master thesis, we have prepared experiments for a few-shot learning face classification and verification. We have trained the Prototypical neural networks and the Siamese neural networks.

7.1 Discussion of results

Prototypical networks

In Figure 14, we present the training curve when we use a different amount of training classes (ways) in a batch during the training. This graph tells us that the final accuracy is better then 10-way (class) batches are used during the training.

In Tables 3, 4, and 5, we explore the relationship between the number of classes per batch during the training, the number of train shots, and the number of test shots. The best combinations of these parameters are summarized in Table 6. The highest accuracy is achieved using 10-way training batches with 3 training examples per class (shots) and 5 testing examples (shots). We have achieved the highest classification accuracy 80,14% in 5-shot testing, 72,21% in 3-shot testing, and 47,27% in 1-shot testing.

Siamese networks

The presented results show that Siamese networks need more training episodes compared to Prototypical networks. The limitation of the Siamese networks model was already outlined in chapter 5.3. The main limitation of Siamese networks is that the testing task (30-way classification) is different from the training task (binary verification task). Therefore Siamese network does not directly learn the few-shot classification task, as Prototypical networks do, but learn a related task (binary image verification), which can be leveraged for classification during the testing. This can lead to lower classification accuracy compared to Prototypical networks, which was the case in our experiments.

We have presented the progress of testing accuracy on the Siamese networks for the classification task in Figure 16. It is clearly visible that the network continues to learn after 60 000 episodes.

Best results for classification accuracy in Siamese networks are presented in Table 6, and the highest classification accuracy is 51.22% in 5-shot testing, 48.33% in 3-shot testing, and 47,27% in 1-shot testing.

As we mentioned before, Siamese networks are trained on a verification task. In Figure 18, we show the learning curves for a different amount of examples per testing batch (shots). In Table 7, we show the final results of the verification accuracy, which is 76.89% for 5-shot testing, 74.11% for 3-shot testing, and 66.22% for 1-shot testing.

Interestingly, as seen in Figure 18, the highest verification accuracy is achieved around episode 5000. After 5000 episodes, verification accuracy tends to go down while the classification accuracy continues to rise.

We have to admit that the classification accuracy of the presented models, namely Prototypical networks and Siamese networks, cannot be compared with the one-shot learning capabilities of humans. We think that with advancements in computational resources and algorithms, few-shot learning algorithms might close the gap between the few-shot classification accuracy of current Deep Learning algorithms and humans. This means not only in the classification accuracy, but as well match the speed of performing the classification task.

At the moment, only neural network models trained using large face-recognition datasets with a large number of examples per class can come close to face classification accuracy of the human brain.

7.2 Limitations of our models and future work

GPU memory is the main limiting factor in Prototypical networks. More concretely, the GPU memory requirements rise sharply with a higher number of output classes and the number of shots per batch. In Table 4, where we have used 5-shot for training, we have been able to train with a maximum of 60-ways (classes). In Table 5, where we have used 1-shot training, we have been able to compute with 120-ways (classes). We were limited by the GPU memory, which is in our case of Nvidia GTX1080 8 GB.

The number of computational resources was another limitation. We have reduced the picture size to 84 by 84 pixels, and even then, the length of our experiments was quite high.

It was mentioned in chapter 6.5 that total computational time was more than 15 days of continuous processing. This time does not include testing time and running incorrect configuration etc. If we have more computational power and resources, we could explore deeper models, and do more extensive hyperparameter search. We could use the full-size images for training and testing and train the model with more episodes. Then this could lead to much better results than we are presenting here.

The most surprising result was that the Siamese network results were not capable of providing classification accuracy comparable to prototypical networks. We hypothesize that this is because the classification procedure is not explicitly built-in the Siamese network training procedure. Instead, the Siamese network is trained on the verification task, and the classification procedure is done during the testing phase by comparing support examples with the query examples.

We think that the main area of focus for future work is to check the possibility to improve the classification accuracy of the Siamese networks and to modify the Prototypical networks to improve the classification accuracy.

Conclusion

In this master thesis, we focused on the implementation of few-shot learning models for face recognition, namely the Prototypical networks and the Siamese networks. The main contribution of this thesis is a novelty in the combination of face recognition using well known few-shot learning models.

In the theoretical part, we have explained face recognition functionality from a neurological perspective, then we have explained the basics of neural networks and, in more detail, how the convolutional networks work. Later we have focused on current research in face recognition using convolutional neural networks and ethical challenges with using these technologies. Then we have elaborated few-shot learning, specifically the Prototypical networks and Siamese networks. Currently, we cannot determine bio plausibility of few-shot networks, but the Prototypical networks derive from cognitive science because they use the prototype theory which was described in chapter 4.2

Finally, we have presented the implementation of our experiments, including architecture and a detailed description of our models. We present the results of our experiments in chapter 6, and we concluded that the few-shot learning models do not provide sufficient classification accuracy compared to humans. However, with more computational power allowing a more thorough exploration of the deep learning models, few-shot face recognition is a promising avenue for future research.

References

Allison, T., Puce, A., Spencer, D. D., & McCarthy, G. (1999). Electrophysiological studies of human face perception. I: Potentials generated in occipitotemporal cortex by face and non-face stimuli. *Cerebral cortex 9, no. 5*, 415-430.

Andreasen et al. (1996). Neural substrates of facial recognition. *Neurosciences 8*, 139-146.

Blažek, V., & Trnka, R. (2009). *Lidský obličej: Vnímání tváře z pohledu kognitivních, behaviorálních a sociálních věd.* Praha: Karolinum.

Bromley, J., Guyon, I., LeCun, Y., Sackinger, E., & Shah, R. (1993). Signature verification using a siamese time delay neural network. *Advances in Neural Information Processing Systems, volume 6*.

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

Ciresan, D., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, High Performance Convolutional Neural Networks for Image Classification. *Proceedings of the Twenty-Second International Joint Conference on Artitificial Intelligence*, *Volume Two. 2*, pp. 1237–1242.

Damasio, A. R., Everitt, B. J., & Bishop, D. (1996). The somatic marker hypothesis and the possible functions of the.

Damasio, A. R., Grabowski, T. J., Bechara, A., Damasio, H., Ponto, L. L., Parvizi, J., & Hichwa, R. D. (2000). Subcortical and cortical brain activity during the feeling of self-generated emotions. *Nature neuroscience*, pp. 1049-1056.

Deng, L., & Yu, D. (2014). Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing*, pp. 1-199. doi:10.1561/2000000039

Drozdia, N. (2020). *facial-recognition-coming-to-europe-terms-and-conditions-apply*. Retrieved from Bloomberg [online].: https://www.bloomberg.com/news/articles/ 2020-02-01/facial-recognition-coming-to-europe-terms-and-conditions-apply

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochasticoptimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Eberhardt, J. L. (2005). Imaging race. *American Psychologist 60, no. 2*, p. 181.

Edvinsson, J. (2017). *Machine Learning at Condé Nast, Part 1: A Neural Network Primer*. Retrieved from https://technology.condenast.com/story/a-neural-network-primer

Farah, I. (1998). Approximate homomorphisms. *Combinatorica 18.3*, 335-348.

Felleman, D. J., & Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex*, 1-47.

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *ICML*, pp. 1126-1135.

Goffaux, V., Jemel, B., Jacques, C., Rossion, B., & Schyns, P. G. (2003). ERP evidence for task modulations on face perceptual processing at different spatial scales. *Cognitive Science, 27(2)*, 313-325.

Golomb, B. A., Lawrence, D. T., & Sejnowski, T. J. (1990). SEXNET: A Neural Network Identifies Sex From Human Faces. *NIPS*, (pp. Vol. 1, p. 2).

Golomb, B., & Sejnowski, T. (1995). Sex recognition from faces using neural networks. *Applications of neural networks*, 71-92.

Grüter, T., Grüter, M., & Christia, C. (2008). Neural and genetic foundations of face recognition and prosopagnosia. *Journal of neuropsychology 2.1*, 79-97.

Hassan, H., Abdelazim, N., Mohamed, Z., & Oliver, S. (2015). Assessment of artificial neural network for bathymetry estimation using High Resolution Satellite imagery in Shallow Lakes: case study El Burullus Lake. *International Water Technology Journal 5*.

Hauck, S. (1998). The roles of FPGAs in reprogrammable systems. *Proceedings of the IEEE 86.4*, (pp. 615-638).

Hill, K. (2020). *The Secretive Company That Might End Privacy as We Know It.* Retrieved from The New York Times [online]: https://www.nytimes.com/2020/01/18/ technology/clearview-privacy-facial-recognition.html

Hu, G., Yang, Y., Yi, D., Kittler, J., Christmas, W., Li, S. Z., & Hospedales, T. (2015). When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. *IEEE International Conference on Computer Vision Workshops*, (pp. 142-150).

Huang, G. B., Mattar, M., Lee, H., & Learned-Miller, E. (2012). *Learning to Align from Scratch. LFW deep funneled images.*

Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (2007). *Labeled faces in the wild: A database for studying face recognition in unconstrained environments.* Technical Report 07-49, University of Massachusetts, Amherst.

Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (2007). *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.* Amherst: University of Massachusetts.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology, 160(1),. 106-154.

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift.* arXiv preprint arXiv:1502.03167.

Jabid, T, T., Kabir, M. H., & Chae, O. (2010). Gender classification using local directional pattern (LDP). *20th International Conference on Pattern Recognition* (pp. 2162-2165). IEEE.

Jirouš, F. (2019). *ČÍNA DNEŠKA: KAMERY VÁS POZNAJÍ PODLE TVÁŘE I STYLU CHŮZE.* Retrieved from Centrum Karla Čapka [online]: https://www.cevast.org/cz/news/33-cina-dneska-kamery-vas-poznaji-podle-tvare-i-stylu-chuze-za-sedm-minut-vas-maji

Kato, Y., & Nakamura, O. (2004). On the isolation of spectacles and the extraction of faces for personal identification. *Canadian Conference on Electrical and Computer Engineering (IEEE Cat. No. 04CH37513), vol. 2,*, (pp. 999-1003).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint*, p. arXiv:1412.6980.

Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. *ICML deep learning workshop, 2*.

Kohonen, T. (1982). Analysis of a simple self-organizing process. *Biological cybernetics 44, no. 2*, pp. 135-140.

Koukolík, F. (2002). Funkční systémy lidského mozku. *Psychiatrie*, pp. 60-65.

Koukolík, F., & Drtinová, J. (2006). *Vzpoura deprivantů.* Praha: Galen.

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *NIPS 2012, Neural Information Processing Systems.* Lake Tahoe, Nevada.

Lake, B. M., Salakhutdinov, R., Gross, J., & Tenenbaum, J. (2011). One shot learning of simple visual concepts. *Proceedings of the annual meeting of the cognitive science society, vol. 33*, 33.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation, vol. 1, no. 4,*, 541-551.

Lickson, J. (1974). David Charles: The story of the Quincy Five. *Mockingbird Press*.

Lieberman, M. D., Hariri, A., Jarcho, J. M., & Eisenberger, N. I. (2005). An fMRI investigation of race-related amygdala activity in African–American and Caucasian – American individuals. *Nature Neuroscience, 8,*, pp. 720–722.

Light, F., & Kovachich, L. (2020). *Coronavirus Outbreak Is Major Test for Russia's Facial Recognition Network.* Retrieved from The Moscow Times [online]: https://www.themoscowtimes.com/2020/03/25/coronavirus-outbreak-is-major-test-for-russias-facial-recognition-network-a69736

Lippmann, R. P. (1988). An introduction to computing with neural nets. *ACM SIGARCH Computer Architecture News 16, no. 1*, 7-25.

Liu, Z., Luo, P., Wang, X., & Tang, X. (2018). *Large-scale CelebFaces attributes (CelebA) dataset.*

Loftus, E. F. (1976). Unconscious transference in eyewitness identification. *Law & Psychol. Rev. 2*, p. 93.

Mysid. (2010). *fusiform gyrus (occipito-temporal gyrus)*. Retrieved from https://en.wikipedia.org/wiki/Fusiform_gyrus#/media/File:Gray727_fusiform_gyrus.png

Nichol, A., Achiam, J., & Schulman, J. (2018). On first-order meta-learning algorithms. *arXiv preprint*, p. 1803.02999.

O'Reilly, R. C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain.* MIT press.

Rakic, P. (1995). A small step for the cell, a giant leap for mankind: a hypothesis of neocortical expansion during evolution. *Trends in neurosciences*, pp. 383-388.

Ranjan, R., Sankaranarayanan, S., Castillo, C. D., & Chellappa, R. (2017). An all-in-one convolutional neural network for face analysis. *12th IEEE International Conference on Automatic Face & Gesture Recognition*, (pp. 17-24).

Rhodes, G. (2006). The evolutionary psychology of facial beauty. *Annu. Rev. Psychol. 57*, 199-226.

Richeson, J. A., & Shelton, N. J. (2003). When prejudice does not pay: Effects of interracial contact on executive function. *Psychological Science, 14(3),*, pp. 287-290.

Rolls,, E. T., & Ekman, P. (1992). Facial Expressions of Emotion: An Old Controversy and New Findings: Discussion. . *Philosophical Transactions of the Royal Society of London Series B,* , 335,69.

Rosch, E. H. (1973). Natural categories. *Cognitive psychology 4(no 3)*, pp. 328-350.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation. . *No. ICS-8506. California Univ San Diego La Jolla Inst for Cognitive Science*.

Saha, S. (2018). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Retrieved from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 815-823).

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in neural information processing systems*, pp. 4077-4087.

Steinkraus, D., Simard, P., & Buck, I. (2005). Using GPUs for Machine Learning Algorithms. *12th International Conference on Document Analysis and Recognition*, (pp. 1115–1119).

Sun, Y., Wang, X., & Tang, X. (2014). Deep learning face representation from predicting 10,000 classes. In. *Computer Vision and Pattern Recognition*, (pp. 1891–1898).

Sung, F., Yang, Y., Zhang, L., Xiang, T., & Torr, P. (2017). Learning to compare: Relation network for few-shot learning. . *arXiv preprint* , p. 1711.06025.

Thompson, E. (2019). *Federal study finds race, gender bias in facial recognition technology.* Retrieved from USA TODAY [online].: https://eu.usatoday.com/ story/tech/2019/12/19/facial-recognition-study-finds-results-biased-race-gender-and-age/2704291001/

Tieleman, T., & Hinton, G. (2012). *Lecture 6.5 - RMSProp,.* COURSERA: Neural Networks for Machine Learning.

Ullah, I., Hussain, M., Muhammad, G., & Aboalsamh, H. (2012). Gender recognition from face images with local wld descriptor. *19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, (pp. 417-420).

Urban, M., & Zandla, P. (2018). *Poznají nás stroje na každém kroku? Pokrok jde rychle, říká odborník na identifikaci tváří.* Retrieved from Lupa.cz [online]: https://www.lupa.cz/clanky/eyedea-rozpoznavani-obliceju-z-verejnych-kamer/

Vinyals, O., Blundell, C., Lillicrap, T., & Wierstra, D. (2016). Matching networks for one shotlearning. *Advances in Neural Information Processing Systems*, 3630–3638.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation 1, no. 2*, pp. 270-280.

Zhou, E., Cao, Z., & Yin, Q. (2015). Naive-deep face recognition: Touching the limit of LFW benchmark or not?. arXiv preprint arXiv:1501.04690.

Appendix

Appendix includes the source code of the Prototypical and Siamese networks used in experiments.