DEEP LEARNING FOR FACE RECOGNITION IN SURVEILLANCE VIDEOS

by

Paul-Darius Sarmadi

A dissertation submitted in partial fulfillment of the requirements for the degree of
Master of Engineering in Computer Science

Examination Committee:	Dr. Matthew Dailey (Chairperson) Dr. Mongkol Ekpanyapong Dr. Manukid Parnichkun
Nationality:	French
Previous Degree:	Baccalauréat scientifique Lycée Descartes, Tours, France

Asian Institute of Technology
School of Engineering and Technology
Thailand
May 2016

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Cameras are everywhere, in front of any store, any business and any streets. They are used for several purposes :
-After a crime has been commited as clues or proofs. The goal is to get more information on the person or on the events happened.
-In the next minutes after a crime, to intercept the person who committed it. Some teams in the police are always scanning surveillance videos, monitoring the subways, streets etc to be able to catch the perpettrator in the corridors, before they could escape the station.
-Before a crime. Typically, to dissuade potential thieves from stealing anything. Sometimes, the cameras used are not working, not recording, or pure fakes.

In the context of global terrorism, the problem of recognizing a previously identified terrorist, hoping to follow him on video cameras saddly appears to be a key issue. More generally, following the path of any criminal using surveillance videos sounds like a main concern.

In this work we can easily assume that the police faces several difficulties:
-The first one is the number of cameras they may have access to -depending on the country's policy-. This number is often very high, and is growing everywhere very fast.
-The second one is the low quality of the videos. It is sometimes very hard to recognize a person on a video with a low resolution.
-The third one is the crowd. It takes a second to recognize a previously identified thief on a video where he is alone. What if there are thirty other persons on the video, in a crowded street for example?

Recognizing a previously identified criminals on these video surveillance cameras adds on to the above mentioned difficulties which requires a lot of human resources. These issues lead to a simple conclusion. It is an expensive work in terms of time, money and human resources, nevertheless extermely important to upheld the national security of any country.

Automatizing the face recognition process in surveillance video seems to be an interesting answer to address the problem.

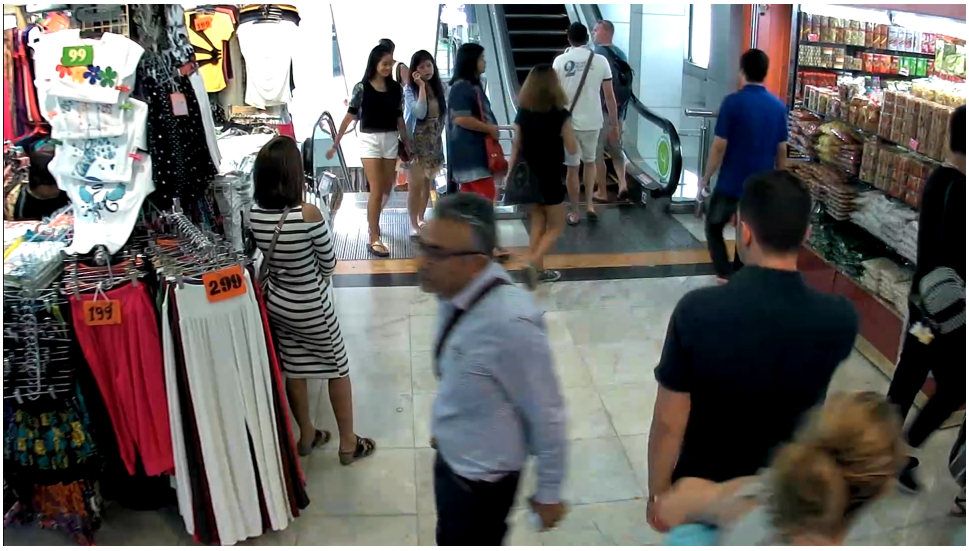


Figure 1.1: A screenshot of a video from the MBK dataset.

1.2 Problem Statement

Face identification is a key machine learning issue at present. The best results were obtained using deep neural network -any artificial neural network with more than one hidden layer. In 2014, DeepFace reached an accuracy of 97.35% for face verification (Taigman, Yang, Ranzato, Wolf, 2014). In 2015, FaceNet reached a 99.63% accuracy on the "Labeled Faces in the Wild" database for identification (Schroff, Kalenichenko, Philbin, 2015).

In surveillance video, face recognition faces issues such as blurriness, low resolution and unexpected poses of faces. Though the problem of face recognition in surveillance video has already been studied, deep learning techniques have a lot more to add on this issue.

The goal of this research is to build a deep neural network for face recognition on surveillance videos. This model should be based on the latest algorithms deep learning offers.

1.3 Objectives

The objectives of this research are formed on a database already provided for this study. This database contains recorded videos from a surveillance system of the MBK mall in Bangkok. The figure 1 shows a frame of one of these videos. Three of our researchers appear during several seconds on some of the videos. They are walking like anyone else in the mall.

Choosing this database serves one goal: to be placed in the theoretical situation of policemen looking

for one or several criminals in the streets or in the corridors of public places, using few sample pictures of them to try to find them out.

Our researchers are playing the role of the criminals we are looking for. The main objective is to use deep learning techniques to build an automated solution to this task of finding our researchers in the MBK mall.

More precisely, considering this database, there are three main objectives in this research study:
1. Create a database of images which are the faces extracted from the surveillance videos.
2. Build a deep neural network for face recognition. The general idea is that the network learns to recognize the three researchers on a training set, and will try to find them on a testing set. The
3. Testing the resulting model. Compare its accuracy with other experiments which used different techniques.

1.4 Limitations and Scope

There are two main limitations to this research:
-Time. This project is three months long. There are many deep learning techniques existing however, due to the limits of time unfortunately, not all could be explored.
-Material limitations. The laboratory provides an NVIDIA GeForce 780 GTX GPU card for the computation. This implies some limitations on the mini-batch size for stochastic gradient descent. Preliminary results show that the size will be limited to 20 samples, while the usual size is 128.

Deep learning gives astonishing results in the task of face recognition. That is why despite those limitations, we can get an interesting model in the context of surveillance videos.

1.5 Research Outline

I organize the rest of this dissertation as follows.

In Chapter 2, I describe the literature review.

In Chapter 3, I propose my methodology.

Chapter 2

Literature Review

As said in the previous section, the goal of this study is to exploit *deep learning* algorithms for face recognition in surveillance videos.

2.1 About Deep Learning

According to "Deep Learning Methods and Applications" (Deng, Yu, 2014):

> Deep learning is a set of algorithms in machine learning that attempt to learn in multiple levels, corresponding to different levels of abstraction. It typically uses artificial neural networks. The levels in these learned statistical models correspond to distinct levels of concepts, where higher-level concepts are defined from lower-level ones, and the same lowerlevel concepts can help to define many higher-level concepts.

One of the most well-known deep learning algorithms is the *Convolutional Neural Network* (LeCun, Bottou, Bengio and Haffner, 1998), a variant of the Multilayer Perceptron (Rosenblatt, 1961) which involves the use of convolutional layers (see Figure 2-1). Biologically inspired by animals visual mechanisms, it is historically known for an application called *LeNet*, able to recognize hand-written digits (LeCun et al., 1989). For ten years, the interest for convolutional networks has grown fast. In fact, with the evolution of technology -basically massive usage of always more powerful GPU cards-, it has been possible to use these networks more widely and for various tasks. They have been used widely for human face classification tasks and have given impressive results.

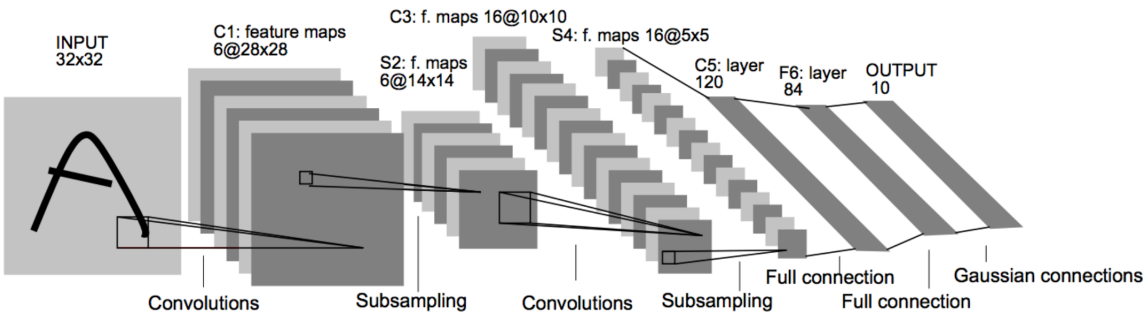


Figure 2.1: Architecture of LeNet-5 for digits recognition. Extracted from LeCun, Bottou, Bengio and Haffner, 1998.

2.2 Previous Work

2.2.1 Deep Learning for face Recognition

There are two main schemes for face recognition :
-Recognition by comparison. Those are the Same/Not Same" algorithms. The basic idea is that the training dataset is made of pairs of image linked to the label "1" if they are representing the same object -in our case, the face of a person, and 0 otherwise. In deep learning, the reference algorithm is the Siamese Network (Chopra, Hadsell, LeCun, 2005). The siamese network is built out of two convolutional networks. The input of the first one is the first image of the pair and the input of the second one is the second image. The two networks share the same parameters, and return two values each. Those values are used to compute an energy. If the energy is high, the two images are considered as "very different". Else they are considered as similar. This energy is a variable of a loss function which will be the origin of an update on the parameters of the two convolutional networks by contrastive gradient descent (Figure 2-2). Intuitively, this computed energy is similar to the gravitational potential energy. If a mass is far from Earth, its GPE will be high, and the mass will be considered as not belonging to the planet. If the mass is stuck to Earth, its GPE will be low and the mass will be considered as a part of the planet itself.

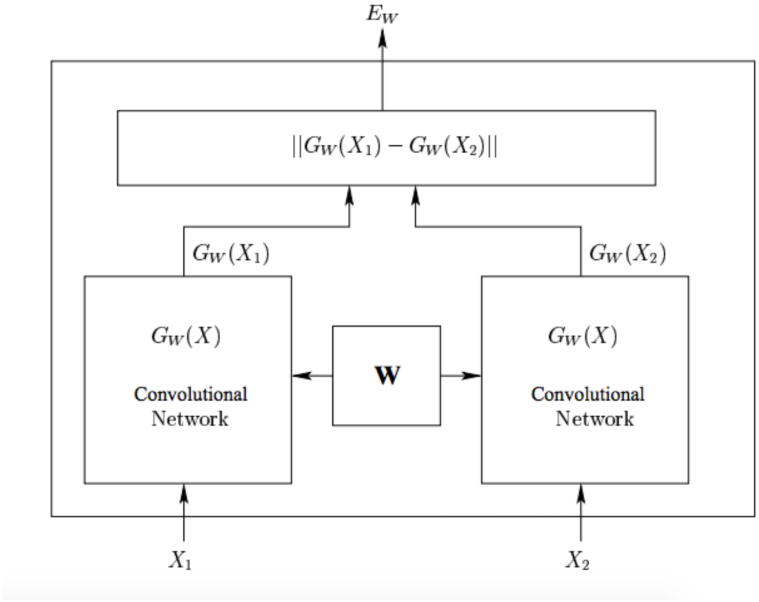


Figure 2.2: Architecture of a siamese network. Extracted from Chopra, Hadsell, LeCun, 2005.

-Recognition by person identification. A network takes an image as an input and returns a label which identifies one and only one person as an output. The literature on this type of architectures is extremely abundant. On face verification, DeepFace (Taigman, Yang, Ranzato, Wolf, 2014) reached a 97.35% accuracy on the Labeled Faces in the Wild (LFW). The state of the art for face identification is FaceNet (Schroff, Kalenichenko, Philbin, 2015).

2.2.2 Video-Based Face Recognition

On automated face recognition for surveillance video, a number of articles have been published. They are using a wide range of methods. (Liu and Chen, 2003) used a Hidden Markov Model for video-based face recognition. (Le An, Kafai, Bhanu, 2012) used a Dynamic Bayesian Network. (Goswami, Bhardwaj, Singh, Vatsa, 2014) proposed an interesting methodology. First, an algorithm extracts the most "memorable" frames in the video. Then, the chosen frame is applied to a deep learning network performing face recognition. This idea is the state-of-the-art for low false accepts rates.

2.3 Conclusion

Video-based face recognition is a very important area of research. Though many articles are published on this subject, some of the latest deep neural networks have still to be applied to the context of surveillance video-extracted images.

Chapter 3

Methodology

3.1 System Design

The following figure shows the design of the final product.

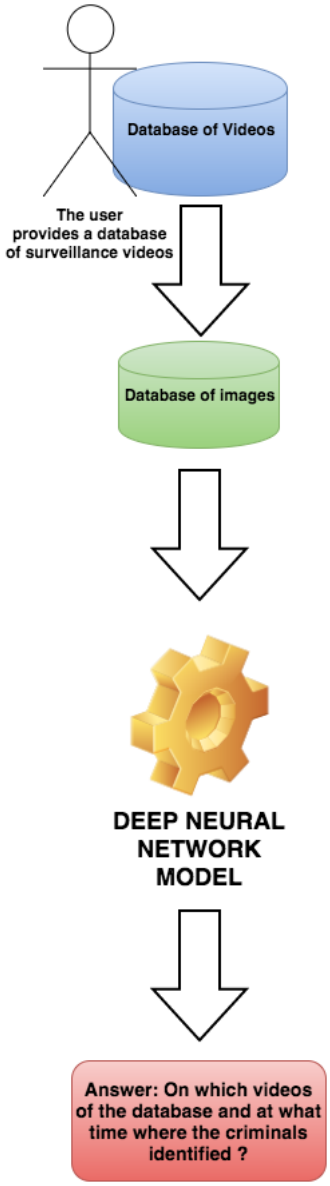


Figure 3.1: Design of the final product.

3.2 Solution overview

The goal of this research is to build an algorithm which can detect our three researchers in the surveillance videos of a mall. There are five steps in this process. 1. Initially, we have a database of videos.
2. Then, these videos are processed to extract the faces of every person appearing in them. We have now a database of faces.
3. From this database are generated some files which are necessery for the learning process.
4. A model is learnt to recognize our researchers.
5. The model is tested.

3.3 Solution Design

The following figure presents two main ideas. In blue, the steps described in the above section are shown. In green, the solutions used to go from a step to the next one are described.

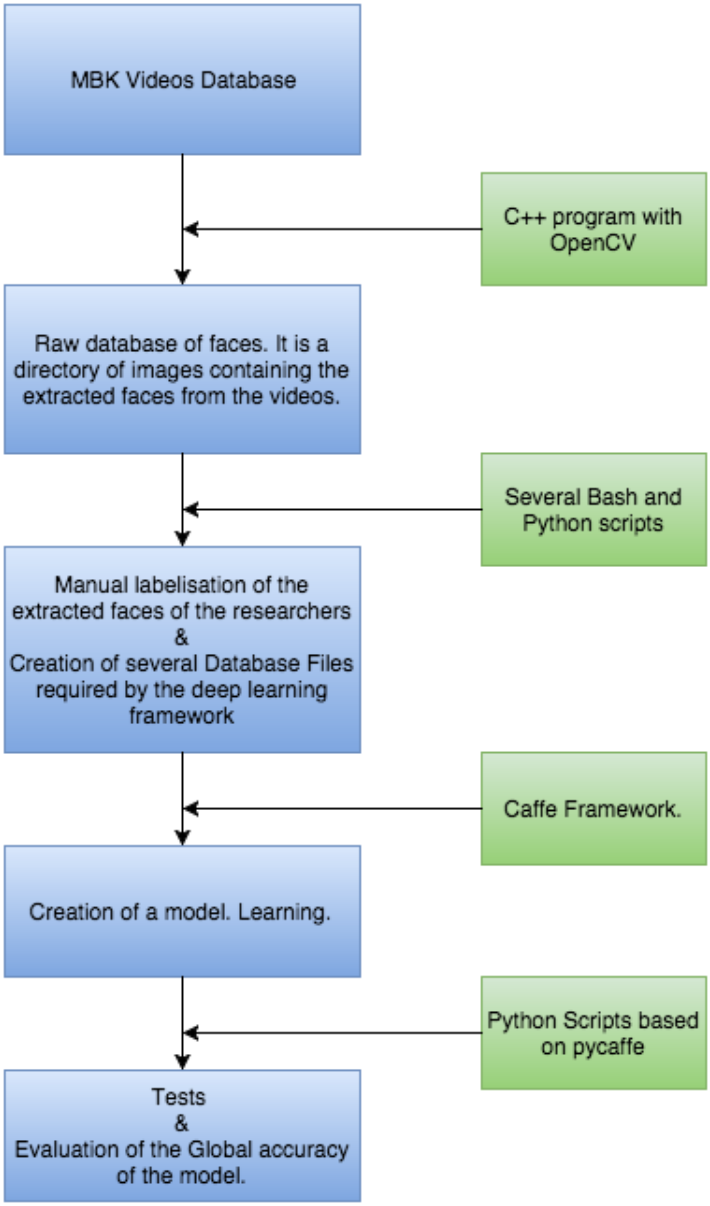


Figure 3.2: An overview of the global design of the study.

3.4 Database

As said in introduction, deep learning algorithms will be applied to face recognition in video surveillance system. A database of surveillance videos is required to generate a training set and a testing set for our model. The database which will be used for the learning process is a set of 14 videos recorded in the MBK Shopping Center of Bangkok. The duration of the videos is variable, from a minute to around three minutes and thirty seconds. Three of the researchers of our laboratory appear on the videos, walking in the mall like any other person.

3.5 Raw Database of Faces

A C++ algorithm using OpenCV (Bradski, G. 2000) will process like so:

for *each video in the database* **do**
    for *each frame of the current video* **do**
        Detect all the faces of the current frame N in the current video;
        Save P-th detected in "Database/video/FrameNFaceP.jpg";
    end
end

Algorithm 1: Face detection Algorithm

The algorithm used for face detection uses a machine learning process called Haar feature-based cascade classifiers, described in (Viola, Jones, 2001).

Once the process is over, a "Database" directory is created one directory for each video. In each directory, all the faces are saved as .jpg files.

Figure 3.3: An example of two faces extracted from the surveillance system

3.6 Creation of database files

The framework which will be presented in the next section requires basically two files to work: a train.txt file and a test.txt file. Their role is trivially linked to their name in a supervised learning process. Each of those files share the same structure:

/adress/of/the/training/image1.jpg label
/adress/of/the/training/image2.jpg label2
...
/adress/of/the/training/imageN.jpg labelN

The label being 1 for the first of our researchers appearing in the surveillance system, 2 for the second one, 3 for the third one, and 0 for any other person.

The labelisation has to be done manually. The chosen method consists in modifying the name of the files where a researcher appears in this way:

"filename.jpg becomes Kfilename.jpg"

K being the label of the researcher.

It is a feasible task to create a serie of python scripts which will run one after the other to create the required train.txt and test.txt files. These python scripts will be indirectly launched through bash scripts. A README file will be provided to explain which commands to type and which options to select to generate the required files from the database.

The purpose of the designed architecture is to make those scripts reusable for any new database, and generalizable for an arbitrary number of labels. If a user provides an other database of videos, and follows the process described in the README file, the required train.txt and test.txt files will be generated, and the models presented in the next section will can be used for learning or testing directly on his own data.

3.7 Model

3.7.1 Framework

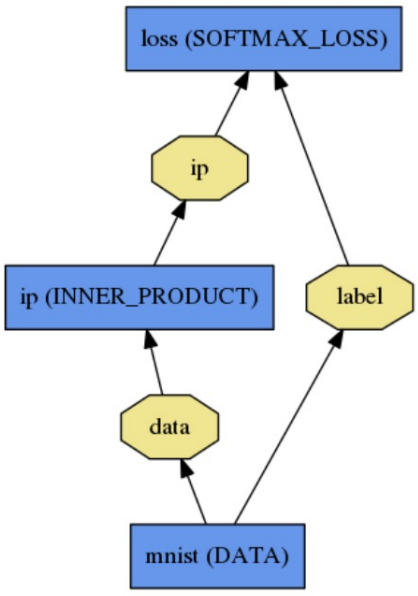
The chosen deep learning framework for this study is Caffe (Jia et al., 2014). It is possible that Torch (Collobert, Kavukcuoglu, Farabet, 2011) will be used also, to some extent.

3.7.2 Strategy

Different strategies have been considered for this modelisation. As explained in the previous chapter, there are two usual schemes for face recognition.
-The first one is the “same/not same” scheme. The idea is to readjust the Siamese Network described by (Lecun et al., 2005) to our particular dataset.
-The second scheme is direct face identification. For this purpose, the idea is to modify the last layer of a state-of-the-art range deep neural network architecture for face identification. The output of this last modified layer should be binary. Either the input image represents one of our researchers -in practice, a criminal-, or either it is not. The previous layers should already be trained, and the training should be done on the last layer only.

3.7.3 The learning process with Caffe

The Caffe framework requires several files to learn the model.
-First, a train.txt and a test.txt files which give the path to all the images with their corresponding label.
-Second, a train_test.prototxt file which describes the architecture of the network. This file is written with protobuf. According to the README file available on the project’s GitHub, “Protocol Buffers (a.k.a., protobuf) are Google’s language-neutral, platform-neutral, extensible mechanism for serializing structured data.”. The figure 3.4 shows how a logistic regression classifier is easily defined in a train_test.prototxt file with Caffe.




```
name: "LogReg"
layer {
  name: "mnist"
  type: "Data"
  top: "data"
  top: "label"
  data_param {
    source: "input_leveldb"
    batch_size: 64
  }
}
layer {
  name: "ip"
  type: "InnerProduct"
  bottom: "data"
  top: "ip"
  inner_product_param {
    num_output: 2
  }
}
layer {
  name: "loss"
  type: "SoftmaxWithLoss"
  bottom: "ip"
  bottom: "label"
  top: "loss"
}
```

Figure 3.4: Logistic regression classifier definition with Caffe. Extracted from the official website of the framework.

-Third, a solver.prototxt file which contains informations on the batch size or on the variables related to the used loss function.

The learning process produces two files. A .caffemodel and a .solverstate. Those files are used to store the value of the parameters of the designed model after a number of steps of learning chosen in the .solverstate file.

3.8 Testing

As we just said, the output of a face identification model should be binary. A python script to test the accuracy of the model can be written with no difficulty. On the contrary, the input of a Siamese Network is two images and the output is an energy. This energy is high for two images representing two different persons and low else. A threshold on this energy has to be determined to make classification possible with the network. Thus, before any test, a script has to be written, determining a good threshold. This script should be written using pycaffe, the caffe model for python. Then, and only then can the tests be done.

Bibliography

[1] Le An, Mehran Kafai, and Bir Bhanu. Face recognition in multi-camera surveillance videos using dynamic bayesian network. In *Distributed Smart Cameras (ICDSC), 2012 Sixth International Conference on*, pages 1–6. IEEE, 2012.

[2] G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000.

[3] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[4] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

[5] Li Deng and Dong Yu. Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387, 2014.

[6] Gaurav Goswami, Romil Bhardwaj, Rajdeep Singh, and Mayank Vatsa. Mdlface: Memorability augmented deep learning for video face recognition. In *Biometrics (IJCB), 2014 IEEE International Joint Conference on*, pages 1–7. IEEE, 2014.

[7] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[8] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[9] Xiaoming Liu and Tsuhan Chen. Video-based face recognition using adaptive hidden markov models. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–340. IEEE, 2003.

[10] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, DTIC Document, 1961.

[11] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[12] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. pages 1701–1708, 2014.