UNIVERSITY OF TARTU

Faculty of Science and Technology

Institute of Technology

Gülce Naz Mert

Artificial Intelligence Based Profession Prediction Using Facial Analysis

Bachelor's Thesis (12 ECTS)

Curriculum Science and Technology

Supervisors

Professor, PhD Gholamreza Anbarjafari

Data Scientist, M.A. Doğuş Karabulut

Tartu 2020

Artificial Intelligence Based Profession Prediction Using Facial Analysis

Abstract

Youth unemployment is a global problem which affects millions of young people. One of the reasons for this is that young people are often misguided, or have adopted professions that are not a good fit for them. If an association between facial features and certain professions can be established using artificial intelligence, it is possible to guide young people into suitable career paths, providing them a better future with more satisfying jobs. In order to achieve this goal, different neural network models that employ deep learning and transfer learning were built, alongside with a dataset consisting of face images of people who are professionals in their fields. This data was then fed into these neural networks, testing effects of different networks and their parameters on the accuracy of predicting professions based on face images. The experiments however, did not lead to high accuracy rates. The results and networks are then analyzed and limitations are brought up. The possible solutions to what could have caused low accuracy rates are discussed.

Keywords

Machine Learning, Computer Vision, Deep Learning, Transfer Learning

CERCS

T111 Image Processing, T112 Signal Processing

Tehisintellektipõhine Elukutse Ennustamine Näoanalüüsi Abil

Lühikokkuvõte

Noorte tööpuudus on globaalne probleem mis mõjutab miljoneid noori. Üks põhjustest on kuna noori inimesi on tihti valesti juhitud või nad on omastanud ameteid mis pole neile sobilikud. Kui on võimalik leida assotsiatsioone näojoonte ja kindlate ametite vahel kasutades tehisintellekti, kas siis on võimalik juhtida noori inimesi parematele ametikohtadele, varustades neid parema tulevikuga, kus on rohkem rahuldavad töökohti. Et sellise saavutusega hakkama saada, ehitati erinevaid närvivõrgud mudeleid mis kasutavad süvaõpet ja ülekandmise õpe koos andmetega, mis koosnevad inimeste näo piltidest kes on oma ala professionaalid. See informatsioon siis sisestati närvi võrkudesse, katsetades erinevate võrkude efekte ja nende parameetreid näo järgi ameti valimise täpsuses. See katse kahjuks ei viinud kõrge täpsusega tulemusteni. Tulemused ja võrgud siis analüüsiti ja leiti limiidid. Võimalike lahendusi arutatakse selle üle mis võiksid tekitada vähese täpsusega tulemusi.

Võtmesõnad:

Masinõpe, Arvuti Nägemine, Sügav Õppimine, Ülekandmise Õpe

CERCS

T111 Pilditehnika, T112 Signaalitöötlus

TABLE OF CONTENTS

1 Introduction

1.1 Problem Definition

Youth unemployment is a major global problem. Long term unemployment is shown to be associated with depression, loss of confidence and other psychological effects in young people (Morrell et al., 1998). With almost 13% of the youth population being out of jobs around the world (ILOSTAT Database), this makes up to roughly a hundred and fifty six millions of young people searching for a job. Considering the employed young population, not all of them are happy with their jobs. This again is a problem, because a study found that young people who are not happy with their job still relatively lack self esteem and struggle with depression (Winefeld et al., 1988), showing that having a job that is not suitable for the person can also have upsetting outcomes, which might lead to people quitting their jobs hence increasing unemployment rate.

Growing up, we all had our dream jobs and when we dream about such things, we do not know what the future actually holds for us. Our interests may change, our abilities may differ, or we may face other difficulties. There are a few ways that are claimed to match people to professions, such as performing personality tests, following one's interests, speaking to a counselor or family. However, these sources might be subjective or not very accurate. One might end up with a job that they like, but do not have the talent, or worse, end up with a job that they hate. Perhaps, it is time to come up with hard evidence based ways to find the right job, which will in turn increase people's satisfaction from their jobs and hence the employment rate.

One of the things that can be used for such a task is people's faces. Excluding some exceptions, we all have the same structure of aligning of the sense organs on our faces. We can utilize this uniform basis to create a method to match certain facial features and patterns to professions. However, the human eye is often not powerful enough to recognize very minor patterns that might exist in a group of people that belongs to the same profession. But we know one thing that is useful at extracting features and patterns and features from images: artificial intelligence.

Artificial intelligence is often used to solve everyday problems and is very good at it. If we can use it for medical imaging, security surveillance, ID verification, and many more, why not use it to find suitable job matches?

1.2 Objective of the Thesis

In this thesis, we are using various machine learning and image processing methods to successfully predict individual's professions by training algorithms on face images of people who are professionals in their fields. We have combined various machine learning concepts, such as deep learning and transfer learning to discover what would be the most optimal and precise way to predict professions based on face images. We have a classifier with sixteen different classes (professions), which the algorithm needs to sort the images into. The algorithms have been trained on two types of images: images as whole (with their backgrounds) and images with faces cropped from their backgrounds. With this diversion, we have tried to detect if there is any effect of existing background on the accuracy of the algorithms.

Merging machine learning and computer vision, we have looked for hidden features or patterns that might exist in faces, indicating the profession one can belong to. For instance, farm workers mostly work outside, where they have to be under the sun for long hours and they might have wrinkles near their eyes as a result of squinting them for a long time. There might be many more features like this that go unnoticed by us, which might be an indicator to what someone does for a living.

Humans are generally very good at processing and extracting information from faces, but our eyes are seemingly not powerful enough to make such distinction between faces to classify other people's jobs based on their faces. If an association between facial features and profession fields can be established with the help of artificial intelligence, we can guide young people towards a career path that is well suited for them, providing them a better future with a satisfying job.

There are many studies that use artificial intelligence to extract facial features, nevertheless, they do not associate these features with profession groups. On the other hand, in one study (Olivola et al., 2014) scientists have tried to associate facial features with leadership roles of a few professions, however, they used mere human eyes instead of artificial intelligence. In that sense, this study is novel and has tried to expand the horizons of the usage of artificial intelligence for extracting information based on the face.

2 Background

Literature Review

The literature review will present theory and the machine learning concepts which are relevant to the aim of the thesis. The first section of the literature review will give an overview of disciplines and theory behind them, while the second part will introduce concepts which are used in experiments

1.1 Research Background

There are two main fields which are keys for understanding and solving the challenges of this thesis. Computer vision is essential to extract and process the dataset, i.e. faces of hundreds of people. Machine learning and neural networks help utilize this dataset and detect patterns and features that are difficult to recognize for humans.

1.1.1 Computer Vision

Computer vision is a field where the computers are enabled to make sense of the visual world and ”see" like humans. This process is actually a transformation, in which the input data gets transferred into either a new representation or a decision and such transformations are done to serve a beneficial purpose. The input data in this process is either a still picture or a video, and these data can be transferred to decisions such as "is there a human face in this picture" or "do tumour cells exist in this tissue" (Bradski and Kaehler, 2013). The ability to answer the question of whether there is a face in the picture or not is beneficial to get data ready for the next stages in this research. The data preparation was automated, meaning that the images to be used in neural networks were obtained from google images using a script instead of manually downloading each picture. Hence, the initial database contained not only the images of faces but also the images of unrelated objects. Since the neural networks need to be trained with face images only, elimination of these unrelated pictures was a necessity. The time required to look at every single picture to determine if it contains a face was considerably large, hence, another automation was required.

Using face detection, computer vision was utilized to write a Python script that deletes pictures without faces, leaving the suitable pictures only.

1.1.2 Machine Learning

What is Machine Learning

It has always been a matter of curiosity if the computer would be able to learn ever since they were invented. Now we know that, even though it might not be as efficient as how humans learn, the algorithms invented are efficient for certain types of learning tasks (Mitchell, 1997).

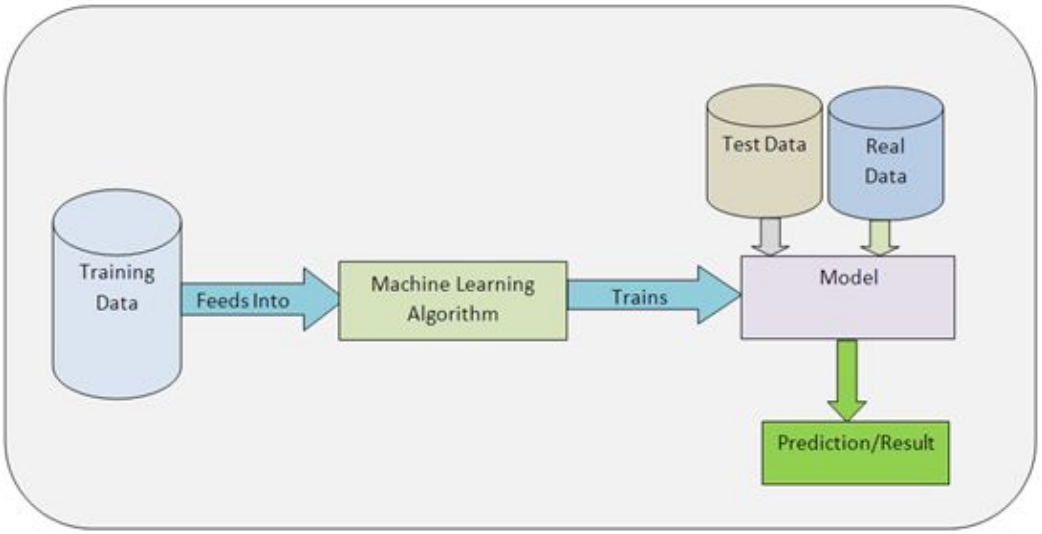

Figure 1. In machine learning, we have an algorithm, which we feed the training data into, which then trains the model with this data and outputs predictions. This means that this provides systems with the ability to learn and improve from experience without being explicitly programmed. [1]

These algorithms based on machine learning are now in use for a vast amount of tasks, such as medical diagnosis, data clustering, virtual assistance, face recognition, and many more aspects of everyday life.

Learning Approaches

There are different approaches to learning problems, namely Unsupervised Learning, Supervised Learning and Reinforcement Learning.

In supervised learning, as the name states, the algorithm is "supervised", meaning that a set with labelled data and their output is given to the algorithm to train on, and it uses its "experience" from training dataset to predict features in other datasets. Supervised Learning can be further divided into 2 sub-categories; classification and regression. In classification, the inputs are assigned to a finite number of discrete categories, tasks such as risk-classification of a banks' loan applicants. However in regression, the output has one or more continuous variables, the algorithm uses its previous information to predict a value for the input (Bishop,2006). An example of the regression would be to predict how much snow would fall to a town in that year.

In unsupervised learning, the algorithm is given a dataset without the labels, and the task for the algorithm is to find hidden patterns underlying in this data. Such a task is done with a clustering method, in which the data points that have similar features are grouped together (Mehrotra et al., 1999).

In reinforcement learning, the algorithm is not given data or examples, instead it learns via the "rewards" from the environment and focuses on maximizing the reward by taking suitable actions (Bishop,2006). The tasks suitable for reinforcement learning would be chess games, path finding, etc.

Under-Over Fitting

When evaluating our model, we measure how well it performs on the unseen data. In other words, how well it can *generalize*. There are 2 causes why our algorithm might have poor generalization ability: overfitting and underfitting.

When the algorithm learns the training data too well, that is, there are too many hypotheses that fit the training data, overfitting occurs (Bradski and Kaehler, 2013). In this case, the algorithm cannot generalize well and fails to fit well the unseen data.

Sometimes algorithms fail to model both training and the unseen data, which is called underfitting. However, this problem is not as significant as overfitting, as it is easy to detect.

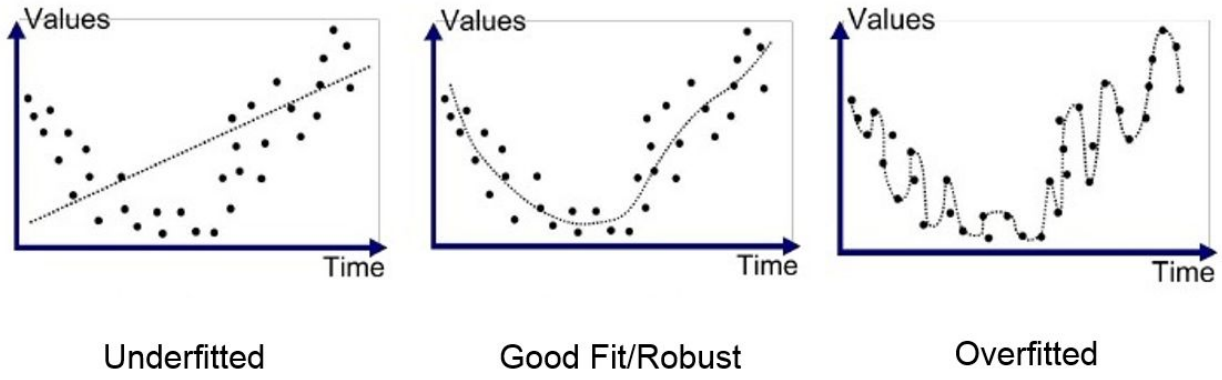

Figure 2. These diagrams show three different models for a given dataset. In the first diagram, it is predictable that the line does not cover all the data points, which shows that underfitting occurs. In the third diagram, it can be seen that the algorithm works too well on that dataset and will perform poorly on the unseen data. In the second diagram, however, the line is predicted quite well and it covers the majority of the training data. [2]

Artificial Neural Networks

Some tasks, such as recognizing faces, are easy for humans to do and yet very complex for computers. To make these tasks easier for computers, the initial development of artificial neural networks(ANN) took place and it was inspired by the urge to build a structure that imitates the human brain (Nilsson, n.d.). With this approach, instead of dealing with an immense amount of rules to solve a problem, a computer is given a model to assess the examples, and brief instructions to improve itself by modifying the model when the error rate is high. As a result, a well-suited model would be expected to solve the problem remarkably accurately.

Perceptron

A *perceptron* (Rosenblatt, 1958), represents a single neuron in the neural network. Multiple inputs(x_1,x_2,...,x_n) and a bias input(x_0) are fed to the perceptron and one output(y) is received in return.

A linear combination(z) of weights and inputs are derived and summed with the summation function, shown with the equation:

$$z = \sum_{k=1}^{n} x_k.w_k + bias \qquad (1)$$

(Mehrotra et al., 1999)

The weight for each input can change in the training process, which will be described later.

After the summation process, the z is then fed to an activation function($\sigma(z)$). The output y is determined by this function, as follows:

$$y = \begin{cases} 0 \text{ if } z \leq threshold \\ 1 \text{ if } z < threshold \end{cases} \qquad (2)$$

(Gallant, 1990)

One such perceptron can work as a binary classifier, which is useful for a linear problem. However, most of the problems machine learning deals with is not linear, and this leads to usage of multi-layer perceptrons, which enables scientists to solve nonlinear problems.

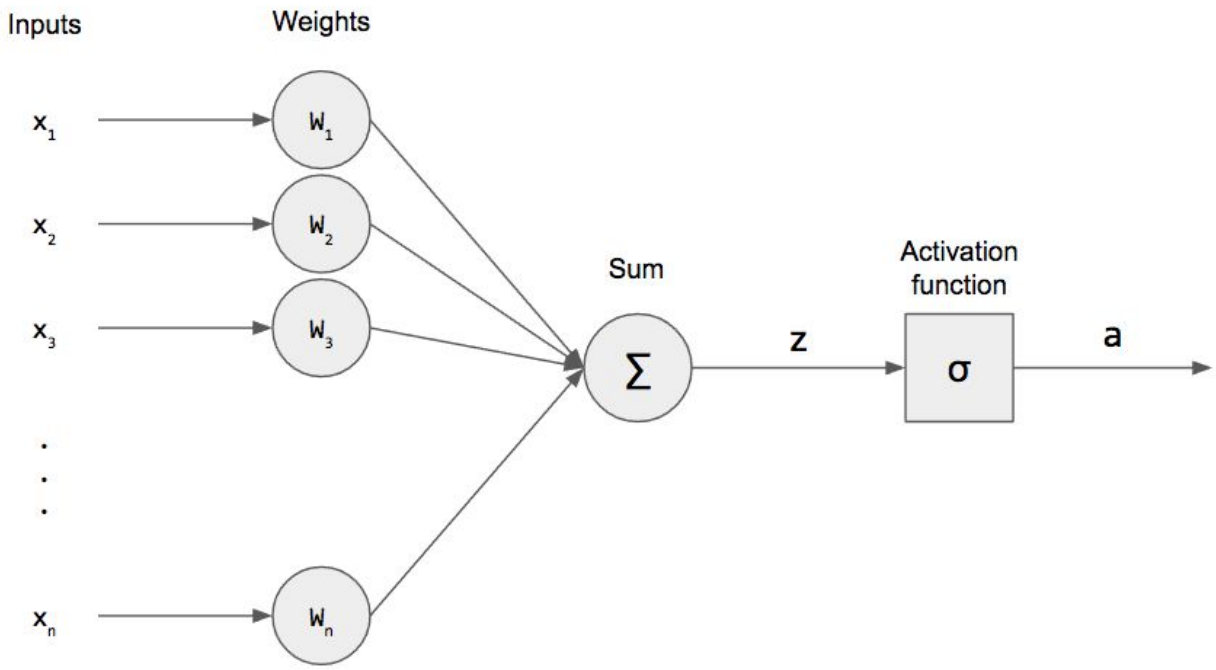

Figure 3. A perceptron shown with n inputs, their weights and summation & activation functions.[3]

Feed-Forward Neural Network

For complicated tasks, one neuron or one perceptron is not enough. For this reason, ANN consists of layers of neurons; input and output layers with hidden layers in between. Data passes through each layer one by one, and in each layer there are neurons(nods) where the computations happen. When there are no connections between neurons of the same layer, or no connections for information to flow backwards -from output to input- , such networks are called *feed-forward neural networks.*

The neurons take inputs and multiply it with a specific weight. The weighted inputs are summed by Summation Function. This summation includes bias in many cases, which is a constant that shows how far are the predicted values from actual values. The result of the summation function is then fed to Transformation(Activation) Function to produce an output, which can be transmitted to the next neuron (Buduma and Locascio, 2017).

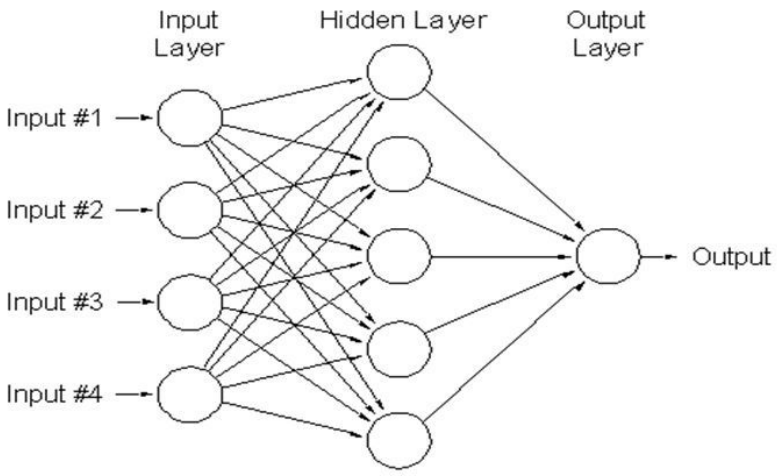

Figure 4. A Feedforward Neural Network [4]

Training of the Neural Network

We want our neural network to perform well, and do the tasks with high accuracy. However, this usually doesn't happen before the training process.

When training a neural network,the error is calculated as the difference between the desired output and the actual output, and this is achieved by *loss function* which helps the algorithm to figure out how much to adjust its parameters so that the actual output is close to the desired output. These adjustable parameters are often called *weights* and to fine-tune the weight vector, the *gradient vector* is computed (Courant and John, 1999).

A derivative is the rate of change or the slope of a function at a given point, i.e., it points to the direction of steepest ascent (Lecun et al., 2015). The gradient, which is a vector-valued function, captures partial derivatives of a multivariable function. If we have a function of two variables, f(x,y), the gradient would be the following;

$$\nabla f(x,y) = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}] \quad (3)$$

(Buck and Buck, 1965)

For each weight, the gradient vector calculates how much the error would change if that particular weight is increased by a small amount (Lecun et al., 2015).

If the loss function is seen as a sloping landscape, each of the components of gradient vector point to the direction of steepest ascent, however, for weight optimization, descent is needed instead of ascent, and that is why the negative of the gradient is used for training. The negative gradient vector points to the steepest descent, which helps find the lowest value of the loss function (Lecun et al., 2015).

However, this method might not always always find the best possible minima.

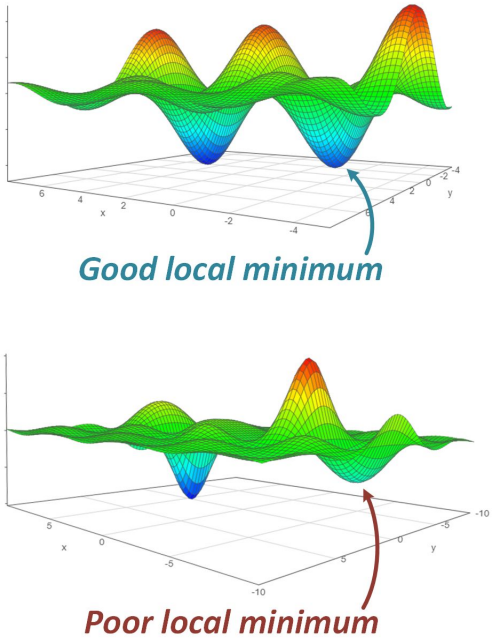

Figure 5. In the first diagram, the local minimum that the algorithm finds is close enough to the global minimum, thus it could be used as the low point. In the second diagram, the local minimum found is not nearly as low as the global minimum, and choosing this point would be a poor decision. [5]

Another aspect that could cause a problem in the process of gradient descent is the learning rate η.

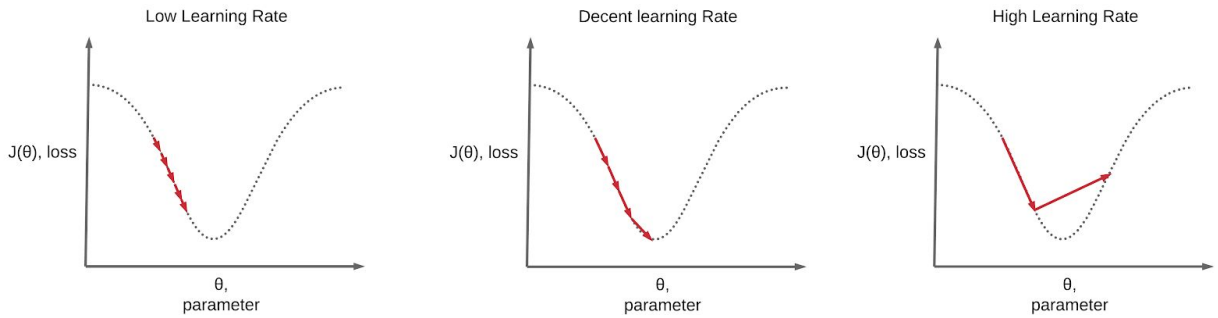

Figure 6. If is set to be too low, it would take many updates to finally reach the minimum point, which is time consuming. If is set to be too high, it could be too large that the algorithm misses a good local minima. [6]

Backpropagation

Learning algorithm of an ANN, which utilizes the gradient descent, uses gradient of the loss function to optimize weights, and the gradient with respect to weights is computed via *backpropagation* algorithm. i.e. backpropagation helps the algorithm to adjust its parameters to decrease the output of the loss function. After each iteration in the neural network where the information is passed forward, backpropagation sends information backwards to update the parameters.

In the training process of a feed-forward neural network, evaluating the gradient of error function is achieved by backpropagation, in which the information is sent forwards and backwards through a local message passing scheme (Mehrotra et al., 1999) . Repeating this process, weights between layers are modified and information is sent back and forth until the desired result, which is a low error value, is obtained.

Regularization

Overfitting in neural networks, as mentioned before, is a common problem. To combat this, there are a few methods that can be employed. One of these methods is *regularization*. Regularization penalizes large weights, hence modifies the objective function that we want to minimize, which is the loss function. It changes objective function into:

$$Error + \lambda f(\theta) \quad (4)$$

where the f(θ) is dependent on θ, and λ is the regularization strength. The λ that we choose determines how much to avoid overfitting. As much as it is a good way to get rid of overfitting, choosing regularization strength too large might cause the model to focus on keeping the θ small, instead of improving the performance (Buduma and Locascio, 2017).

1.2 Concepts

There are two major machine learning concepts to be used in the experiments: deep learning and transfer learning. Both of these approaches reduce the time and resource for the training of the models and help with complex tasks.

1.2.1 Deep Learning

Sometimes, conventional neural networks are not sufficient for very complex tasks. For these tasks, we can employ deep learning models, which consist of multiple processing layers. There has been an increased interest in deep learning, since it seems to outperform previous state-of-the-art machine learning techniques in several fields, as well as in the presence of large amounts of data from different sources (Voulodimos et al., 2018).

There are different methods of deep learning, however, Convolutional Neural Networks achieved breakthroughs for image processing and computer vision tasks (Courant and John, 1999).

Convolutional Neural Networks

Many data modalities are actually formed from multiple arrays, and Convolutional Neural Networks(CNNs) are designed to process this kind of data (Courant and John, 1999). For image processing, this could be images that are 2D, or video footages that are 3D. The name *convolutional* is the result of the fact that this type of networks use the mathematical operation "convolution" (Goodfellow et al., 2017).

The architecture of CNNs is structured with convolutional, pooling and fully connected layers. What makes CNN special for image processing tasks is the convolutional layers within, which

generates feature maps. To comprehend the process of how CNNs work, we need to break down these layers and understand their functions.

Convolutional Layer

The word convolution can be described as "an operation on two functions of a real valued argument" (Goodfellow et al., 2017). In convolutional neural network context, the first function is referred to as the *input*, and this can be the picture or the video that we are feeding the network with. The second function that the operation involves is referred to as *kernel.* Convolutional layer often consists of several kernels, which are utilized to convolve the whole image and extract various feature maps (Voulodimos et al., 2018).

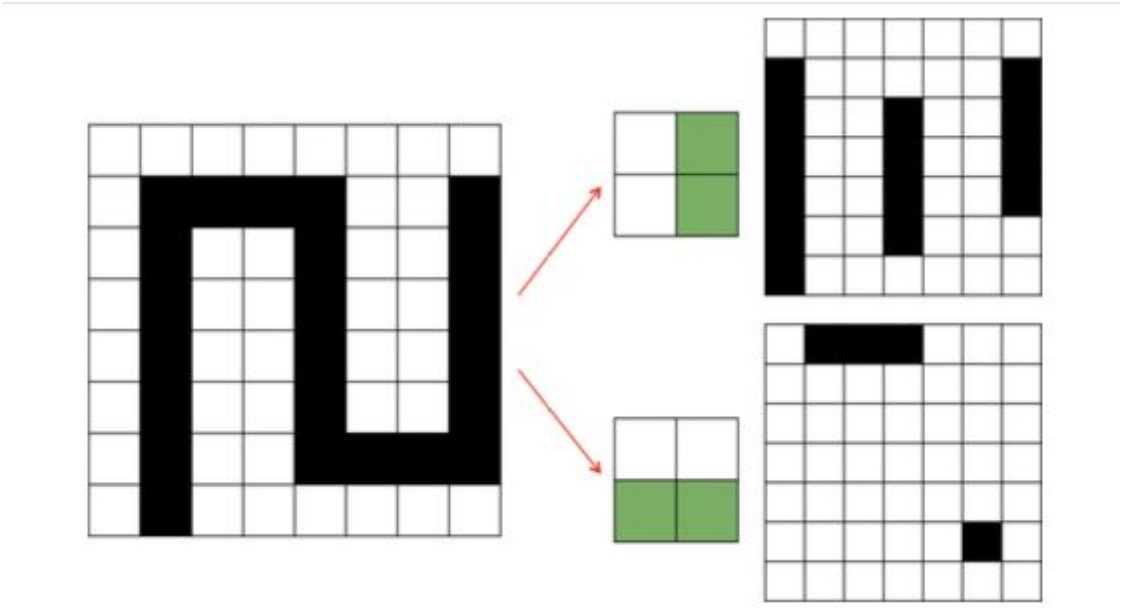

Figure 7. Line detection filters (marked with green) applied to an image, resulting in two different feature maps. [7]

The feature maps can be referred to as the output of the convolution operation (Goodfellow et al., 2017). If we refer to Figure 7, the vertical line detection filter(on the top) is slid over each pixel in the image, checking for a match. The regions where there is a match are shaded black, resulting in detection of vertical lines and the responding feature map (top right corner) (Buduma and Locascio, 2017).

Pooling Layer

Pooling layer helps to reduce spatial dimensions of the input, which are the height and the width. Due to the fact that this reduction in size can lead to the loss of information, this operation is often called *downsampling* (Voulodimos et al., 2018). Although this sounds like a bad operation to run, this loss is actually favorable for the network, as the reduction of the size makes the algorithm computationally less complex (Goodfellow et al., 2017).

One of the most important features of pooling is that it is invariant to translation. What this means is that even if the input is shifted or translated to some extent, the output of the pooling layer stays the same and this is helpful when we care about a feature's existence rather than its location (Buduma and Locascio, 2017).

For example, if we need to determine whether an image contains a face or not, we do not need to know the exact location of the face, we merely need to know that there is a face present in the image. For this task, the pooling layer is useful in the context that it will still preserve the output and extract a feature map with needed information, even if the location of the face changes in various inputs.

Fully Connected Layer

The last element of CNN is the fully-connected layer.The fully connected layer generates 1D feature vector from the 2D feature maps (Voulodimos et al., 2018). In this layer, each node is directly connected to all nodes in the adjacent layers. One important drawback of this layer is that all these connections between nodes result in an abundant amount of parameters, which in return makes the algorithm computationally very complex. Therefore, an elimination of some of these nodes and connections is necessary (Albawi et al., 2017). This is achieved by the *dropout* technique where some of the nodes are randomly ruled out, which ensures that the network is not too dependent on some of the neurons and is still accurate even when it lacks certain information (Gu et al., 2018).

1.2.2 Transfer Learning

Learning process in humans includes transferring knowledge between tasks. When a new task is encountered, relevant knowledge from previous learning experiences are applied to the new task (Torrey and Shavlik, 2009). For instance, if a person knows how to code in Python and is trying to learn MATLAB, the chances are that this person will learn MATLAB faster compared to a person who does not have prior experience in coding. The Python knowledge will make the learning process faster by helping the person utilizing the experience from before and relating to similar aspects of these programming languages.

However, transferring the knowledge has not been so instinctive in the machine learning field. In traditional machine learning algorithms, isolated tasks are addressed. As a result, they assume that both training data and the test data have the identical feature distribution. Therefore, if the feature distribution of the input data changes, the models need to be rebuilt from scratch and use a whole new set of training data (Pan and Yang, 2010). Since models require extensive amounts of training data to be accurate and perform well, recollecting the training data and rebuilding the model takes a considerable amount of time. The aim of transfer learning is to change this by transferring knowledge learned in other tasks and improve learning in a related task using previous knowledge.

3 Methodology

3.1 Tools

Python

Python is a general-purpose programming language that has been in use for a long time. It is very popular today due to several reasons; it is easy to learn, convenient in many areas, allows usage of numerous libraries and modules within, and it is available on all the major hardware platforms and operating systems (Brownlee, 2016).

The reason why I chose to work with Python is that it provides Keras and OpenCV interfaces, which are the two libraries that allow me to perform computer vision, image processing and machine learning tasks.

OpenCV

OpenCV is an open source library which was designed for computational effectiveness. It allows its users to benefit from its simple-to-use computer vision infrastructure and the OpenCV library has numerous functions that are being used in areas such as security, robotics, medical imaging, user interface, and many more (Bradski and Kaehler, 2013).

Why OpenCV is crucial for this thesis is the image processing functions it contains, which helped me prepare a dataset based on people's faces, which will be described in more detail later.

TensorFlow

TensorFlow is a Python library for fast numerical computing that was developed by Google. It is used for creation of deep learning models, however, it can be difficult to use it directly as it incorporates high level details and complexity. Despite this complication, it is possible to leverage TensorFlow by using a wrapper library like Keras (Brownlee, 2016).

Keras

Keras is an open source Python library that uses TensorFlow as a backend. It covers the complexity of TensorFlow and enables users to develop machine learning models easier and faster (Brownlee, 2016).

flow_from_directory

The data for machine learning tasks often needs to be labeled for algorithms to train on them. However, when there is an abundant number of images or any data from, it is quite time consuming to manually label every single data. flow_from_directory function helps to avoid this tedious process by automatically identifying classes from the folder name. Therefore, all it needs is pictures separated in their respective folders to "label" the data.

google_images_download

google_images_download is a library in Python which helps users retrieve images of interest. It is extremely useful in case of batch image retrieval, which can be quite time consuming when done manually. It takes arguments such as;

- Keywords: names of the items/people that needs to be searched for images in Google
- Limit: number of images to be retrieved for each keyword
- Format: allows user to choose the format of the images to be downloaded

It is very convenient to use as it can be customized for individual purposes via arguments.

Autocrop

Autocrop is a useful Python library for automatically cropping faces from a large number of pictures. It not only provides the face-cutting, but it also provides customizing options. It is possible to modify the output image size or the zoom factor, that is, how much of the image face makes up to.

Image Augmentation

Image augmentation is a technique to artificially increase the training dataset without having to gather new data. Image augmentation helps to achieve better accuracy by flipping, rotating, zooming, shearing or scaling the images that already exist in the dataset.

3.2 Methods

Data Preparation

The dataset for this research consists of face images of individuals who have achieved significant success in their profession areas. These 16 professions are given in the table which are included in this database.

Sports	Politics	Art	Science
Football Player	MP	Actor\Actress	Mathematician
Basketball Player	President	Director	Physician
Volleyball Player	CEO	Musician	Biologist
		Singer	Computer Scientist
		Dancer	Chemist

Table 1. Profession groups which the people belong to

There are steps to building this dataset, which will be explained in detail below.

1. **Find the people who are experts in their profession**

The algorithms needed to train on face images of people who are already professionals in their fields, so that the common features a profession group has can be extracted. Google searches such as "Top 100 female computer scientists" or "List of the most successful male CEOs" gave away such useful information, helping me extract names of successful professionals. In this process, one of the most important aspects to pay attention to was the person's age or when they have lived. For instance, despite the fact that Marie Curie was a revolutionary chemist, or Thales was a pioneer in mathematics, they cannot be used in this work as they do not have pictures that can be used in neural networks.

This process has resulted in around ninety names collected for each profession, separate for men and women, making up to 3267 people in total.

2. **Find and download the images of people**

After collecting names of people whose pictures to be used, data collection had to be performed. For this purpose, google_images_download library was utilized to download public domain images of the given list of people.

3. **Eliminate pictures without faces**

The pictures were batch downloaded via google_images_download, which means that I did not have control over the picture content. Since the algorithms needed to train on face images, elimination of images without faces was necessary. For this purpose, OpenCV library was once again very useful. It allowed me to write a script that was able to detect faces in images, and delete the images with no faces in them. As a result, a total number of 116223 images with faces were ready to be fed into neural networks, 62046 images belonging to men and 54177 to women.

4. **Group the pictures in accordance to class they belong to: training, test, or validation**

When training machine learning algorithms, it is important to divide the dataset appropriately into training, validation and test data. These are distinct datasets that are required for the different parts in the process.

Training data is given to the network with labels, for the algorithm to train on and learn from. Validation data, as the name suggests, is used to validate the model during training. This data is also given with labels and helps to adjust the hyper-parameters of the model to help improve generalization. However, the network does not learn from this dataset and hence the weights do not get updated according to validation results. The test dataset on the other hand, is not labeled and does not get involved in the training process. The aim of introducing a test dataset is to test how well our model predicts.

As a common practice, I first have divided the pictures roughly into 80% training and 20% test data. Then I further divided the training dataset into 80/20, 20 percent being validation data.

5. **Crop the faces from images to build a different dataset to train on**

After performing the first four steps, the result is a dataset that is properly divided into categories and this dataset not only contains faces, but also the backgrounds. In order to understand how the background affects the training and accuracy, I have cropped faces from the images using Python's *autocrop* library. This allowed me to have images of only faces, eliminating the background effect and having a separate dataset to train on.

Building Models

This section will present the architecture of the models that are used for experiments of this study. These networks are the initial versions, the modification and optimization of these models will be discussed later. The models are built using Keras library as a wrapper for TensorFlow.

4 types of models that have been tested:

- Network 1: Simple Neural Network
- Network 2: Simple Neural Network + image augmentation
- Network 3: Deep Learning + image augmentation
- Network 4: Transfer Learning + image augmentation

Network 1 & Network 2

The first and the second model share the same features: they are simple neural networks with the same three layers and same parameters. The only difference between these models is the image augmentation that is being employed in Network 2. This factor makes a difference in the sense that it artificially expands the training dataset by generating altered images from the original dataset. The results of data augmentation will be discussed later.

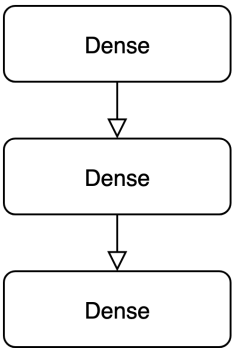

Figure 8. Architecture of Network 1 and Network 2

Parameter	Value
Input Image Shape	300,300
Activation Functions	ReLu
Output Layer Activation Function	Softmax
Optimizer	Adam
Loss Function	Categorical Cross Entropy
Learning Rate	0,001

Table 2. Network 1 and Network 2 Parameters

Network 3

Network 3 is a Convolutional Neural Network(CNN) which has significantly more layers than the former two. The architecture and parameters are depicted below.

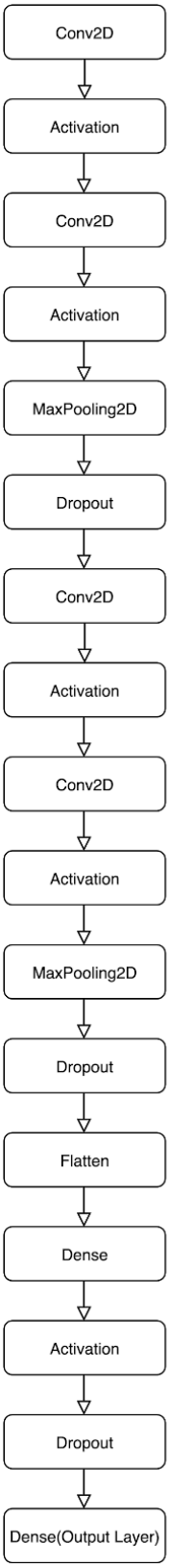

Figure 9. Network 3 Architecture

Parameter	Value
Input Image Shape	300,300
Activation Functions	ReLu
Output Layer Activation Function	Softmax
Optimizer	RMSProp
Loss Function	Categorical Cross Entropy
Learning Rate	0,0001

Table 3. Network 3 Parameters

Network 4

The last model is built to test how employing transfer learning would affect the accuracy. This network uses the Inception-v3 (Szegedy et al., 2016) pre-trained model and its weights to make predictions of professions on the new dataset. The Inception-v3 architecture is given below:

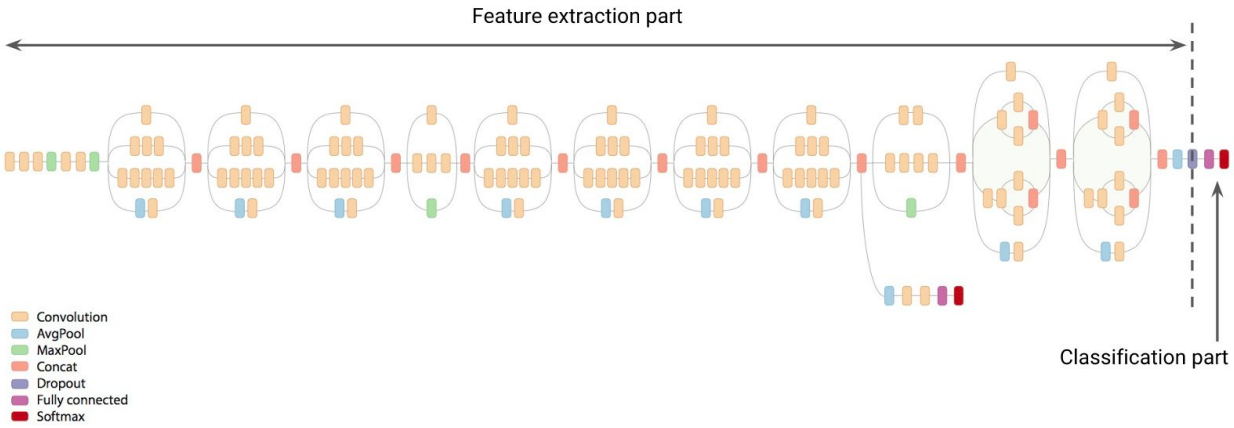

Figure 10. Inception-v3 Architecture [8]

Inception-v3 consists of two units: feature extraction part and classification part. Feature extraction is the part which is the most complex and requires an intense amount of computing power that takes a lot of time to train. To re-use this architecture in a new model, I have frozen the feature extraction part, meaning that this part will not be training further and its weights will

not be changed. After freezing the first layers, I have removed the top layers, which is the classification part, to modify it for the use of my new dataset. The resulting architecture is given in figure 12.

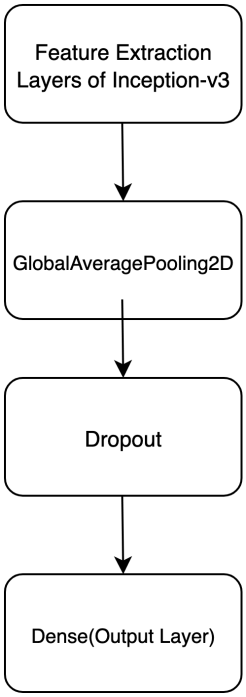

Figure 11. Architecture of Network 4

4 Experimental Results and Discussion

Results & Improving Models

For a neural network to perform well, it needs to be optimized in accordance to the task it is assigned to. As expected, the first batch of models in this research were not very accurate. In this chapter, the initial results are shown, followed by the explanation of how the model optimization is performed in order to achieve better results.

Results of Training on Images With Background

Network 1

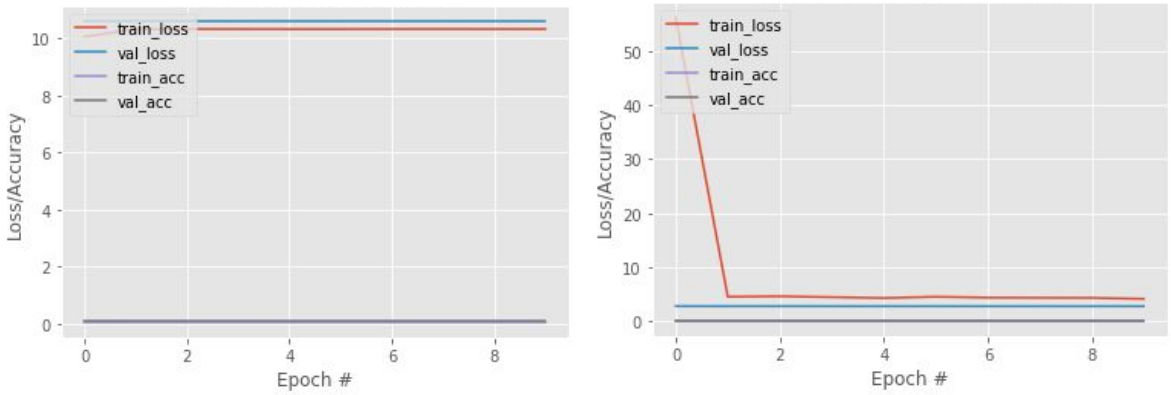

Figure 12. Loss and accuracy graph of Network 1, trained on men and women respectively

The first network, due to its architecture with very few layers, did not perform well. In fact, it did not learn anything, which was expected considering the complexity of the task and simplicity of the model with no convolutional layers.

Network 2

This model used image augmentation factor and has yielded slightly better results, however, it was hardly noticeable.

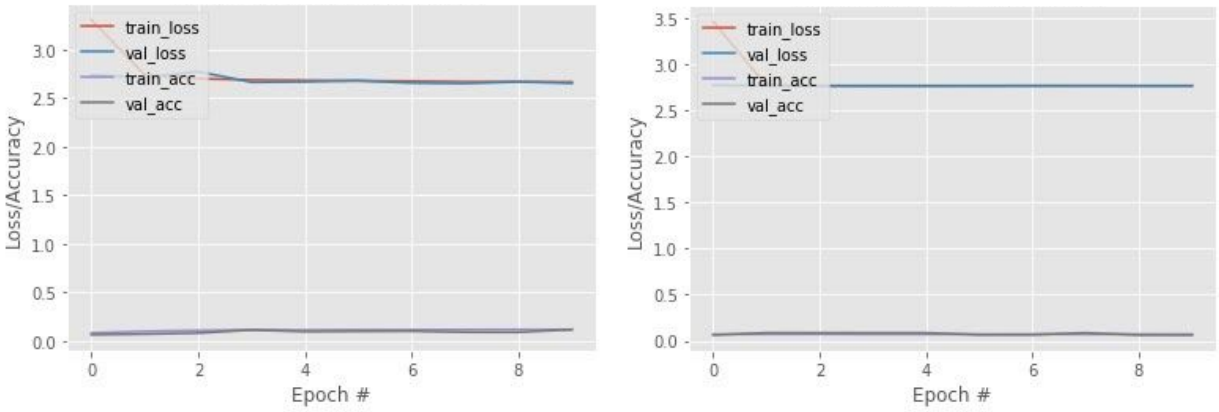

Figure 13. Loss and accuracy graph of Network 2, learning rate of 0,001, trained on men and women respectively

The validation accuracy rates have gotten stuck at a very low rate. After searching for what could have caused such strandedness, I have come to the conclusion that the learning rate was the problem, it was too large and the network could not get a good local minima of the cost function. After coming to this conclusion, I have changed the learning rate from 0,001 to 0,0001.

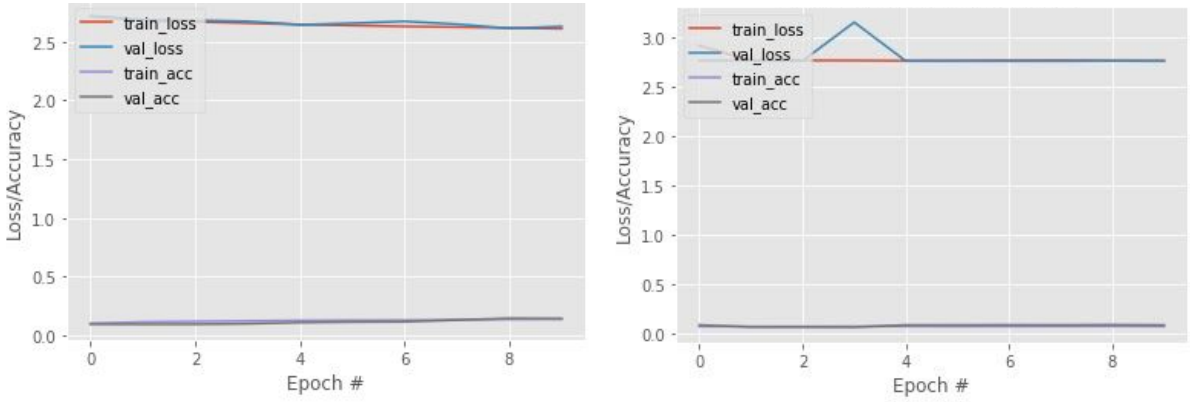

Figure 14. Loss and accuracy graph of Network 2, trained on men, learning rate of 0,001

Lower learning rate helped the network to find a good local minima, while image augmentation enhanced training of the model by providing a better variety of images. As a result, I have gotten a slightly better validation accuracy than the first attempt, however, it still needed to be improved greatly in order to achieve the aims of this thesis. This proves that for such complicated tasks with images, regular neural networks are not powerful enough. Hence, I have built a convolutional neural network for the next step.

Network 3

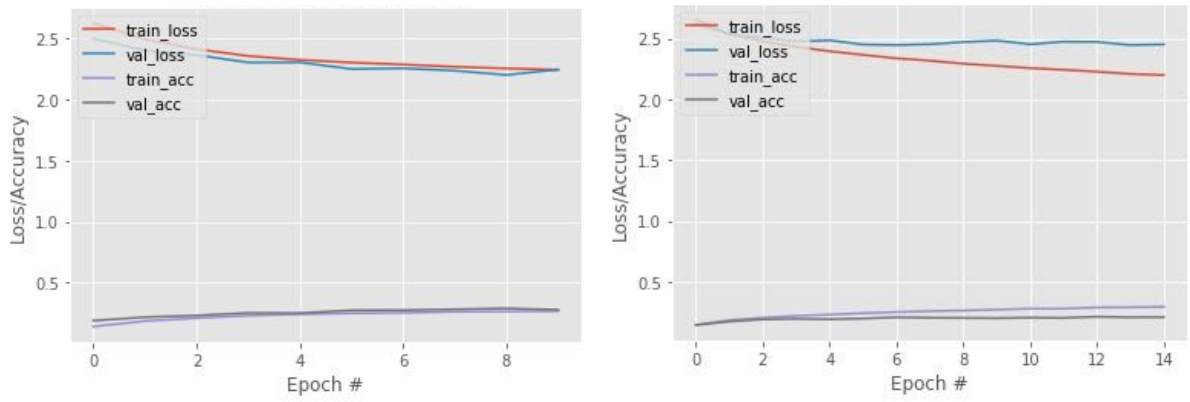

Figure 15. Loss and accuracy graph of Network 3, trained on men and women respectively

As can be seen from the graph, implementing convolutional neural network has increased the accuracy. In fact, it has performed almost %100 better than Network 2, with validation accuracy of 0,2899 in men dataset and 0,2011 in women dataset. However, there is still so much more to improve, considering the fact that these accuracy scores are still not very high.

Network 4

There is one more thing left to test: implementing a pre-trained model. The major open source pre-trained models such as InceptionV3 are results of intense research and very powerful machines. Using such models as a base for our own network can improve the performance.

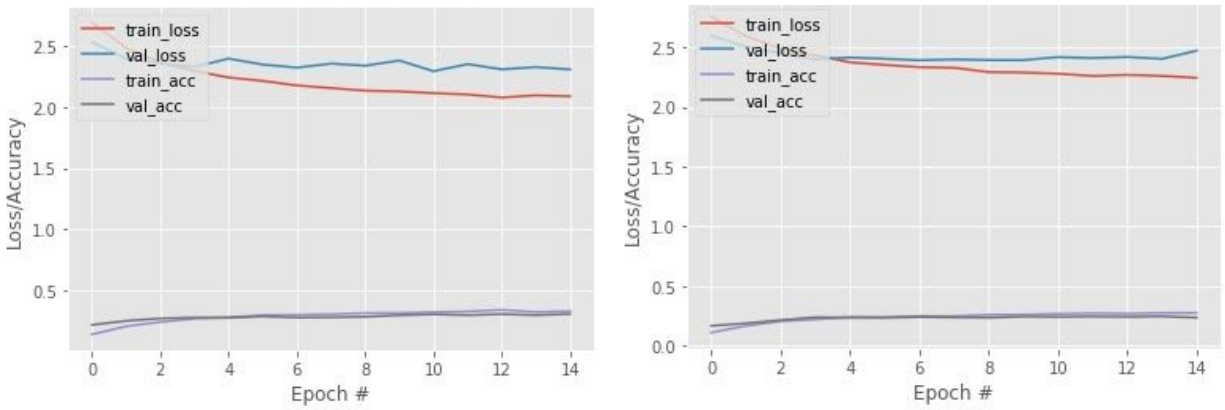

Figure 16. Loss and accuracy graph of Network 4, trained on men and women respectively

Freezing feature extraction layers of the Inception-V3 model and training it further on my own dataset did increase the accuracy, however, in a very insignificant manner. It only produced a validation accuracy of 0,3037 in men and 0,2451 in women, which is not a big jump compared to the results of Network 3.

In order to increase the performance, I have changed a few parameters in the network.

	Model	
Parameter	Network 4	Network 4 - Improved
Batch Size	32	16
Dropout Rate	40%	20%
Dense Layer	Not Used	Used
Inception-V3 Preprocessing Functions	Not Used	Used

Table 4. Comparison of the parameters that are changed in the improved version

Firstly, I have reduced batch size, which would help getting lower generalization error. Dropout layer excludes random inputs every cycle, which means that it prevents overfitting but it can also decrease accuracy, which is not favorable. Since I did not have an overfitting problem, a reduced dropout rate would help me increase the accuracy. Furthermore, in the improved version, there are two things that do not exist in the initial model: dense layer and preprocessing functions. When using the pre-trained model Inception-V3, it is necessary to preprocess images and have the images in the format that Inception-V3 requires. Hence I have added this function in the improved version, alongside with a dense layer with 128 hidden units.

The architecture and results of the improved version is shown below.

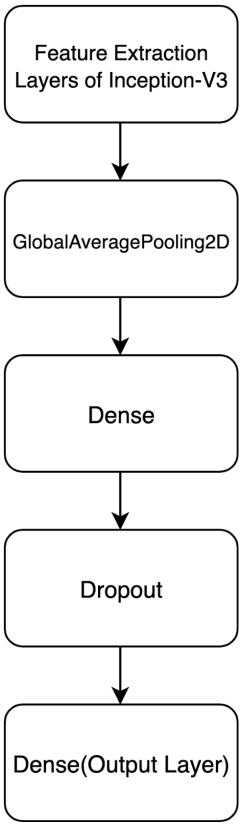

Figure 17. Architecture of Improved Version of Network 4

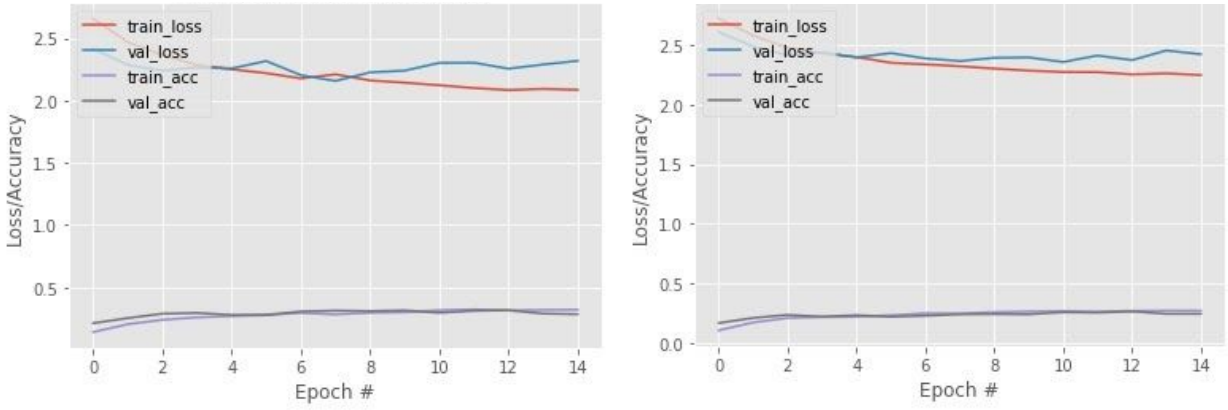

Figure 18. Loss and accuracy graph of Network 4 improved version, trained on men and women respectively

With changed parameters, validation accuracy went up to 0,3164 on the men dataset and 0,2637 on the women dataset. This shows somewhat improvement, but still not a very good performance considering the usage of a pre-trained model.

Results of Training on Images Without Background

Network 1

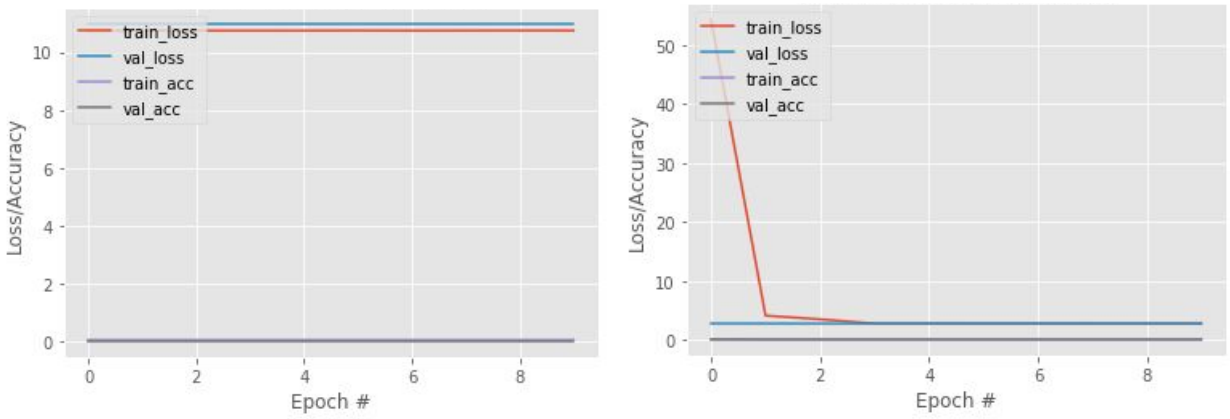

Figure 19. Loss and accuracy graph of Network 1, trained on men and women respectively

Network 1, did not seem to learn anything again when run on the dataset of faces only.

Network 2

Having learnt from experience, I set the learning rate of 0,0001 from the beginning, in order not to waste time. The results were as expected, slightly better than Network 1, but still no remarkable improvement. Network 2 resulted in validation accuracy of 0,1592 in the men dataset and 0,1097 in the women dataset.

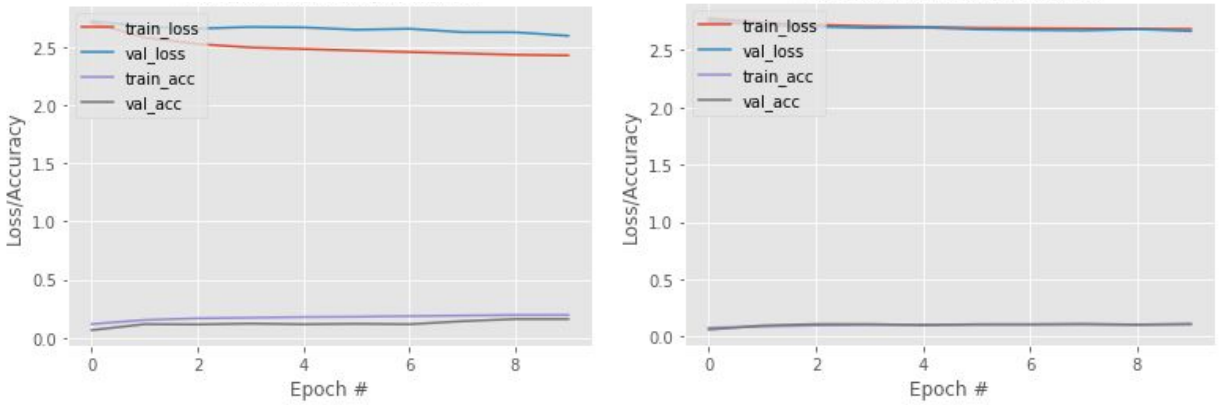

Figure 20. Loss and accuracy graph of Network 2, trained on men and women respectively

Network 3

For this set of experiments on deep learning, I have used two different input shapes: (100,100) and (300,300). This was for the purpose of understanding how the input shape would affect the algorithm in terms of detecting the details of the face.

Input Shape (100,100)

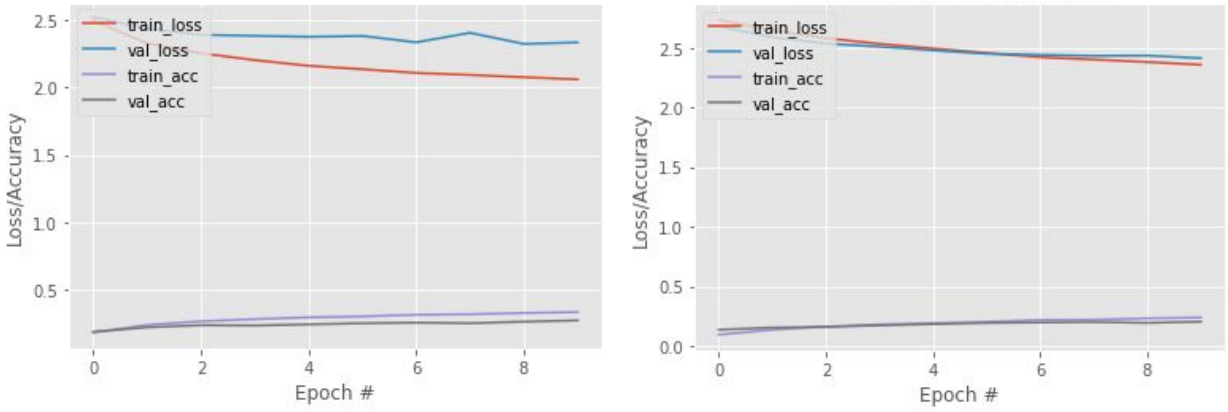

Figure 21. Loss and accuracy graph of Network 3, trained on men and women respectively

Input Shape (300,300)

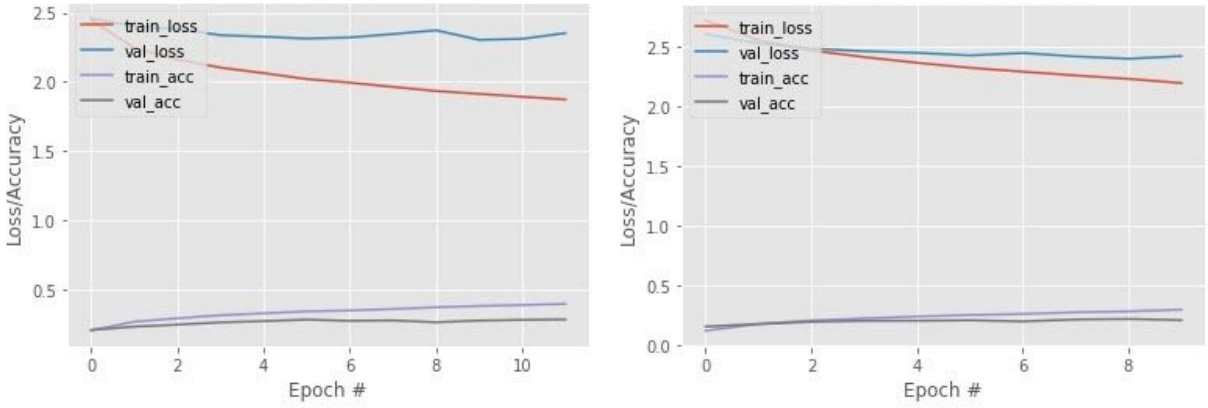

Figure 22. Loss and accuracy graph of Network 3, trained on men and women respectively

Training on images with shape of (100,100) has given validation accuracy of 0,2707 on the men dataset and 0,2046 on the women dataset. When I changed the shape from (100,100) to (300,300), the validation accuracy somewhat increased, with the value of 0,2853 on the men dataset and 0,2158 on the women dataset.

Network 4

Results of Initial Network 4

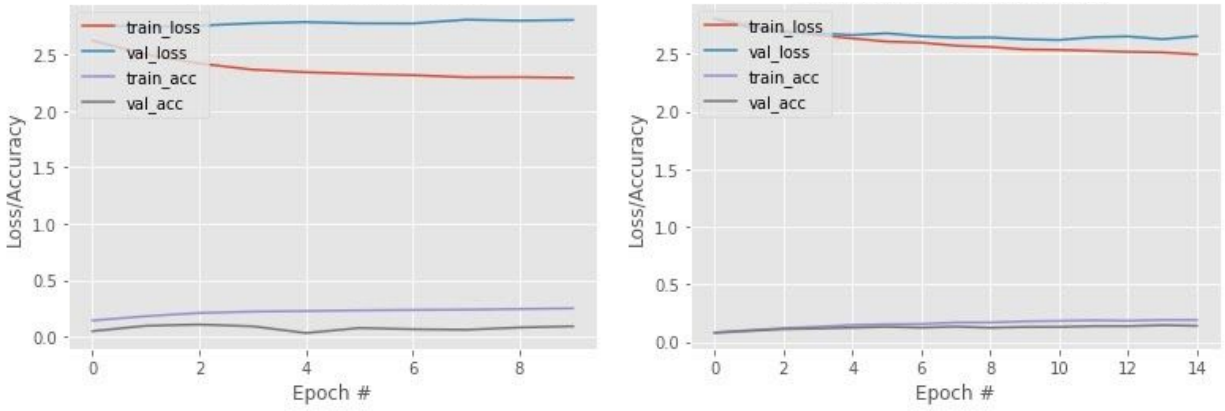

Figure 23. Loss and accuracy graph of Network 4, trained on men and women respectively

Considering the results that would be expected from transfer learning, Network 4 has performed poorly on face images, with validation accuracy of 0,1372 on men and 0,1436 on women.

I have then used the improved version to test if this would enhance learning.

Results of Network 4 - Improved

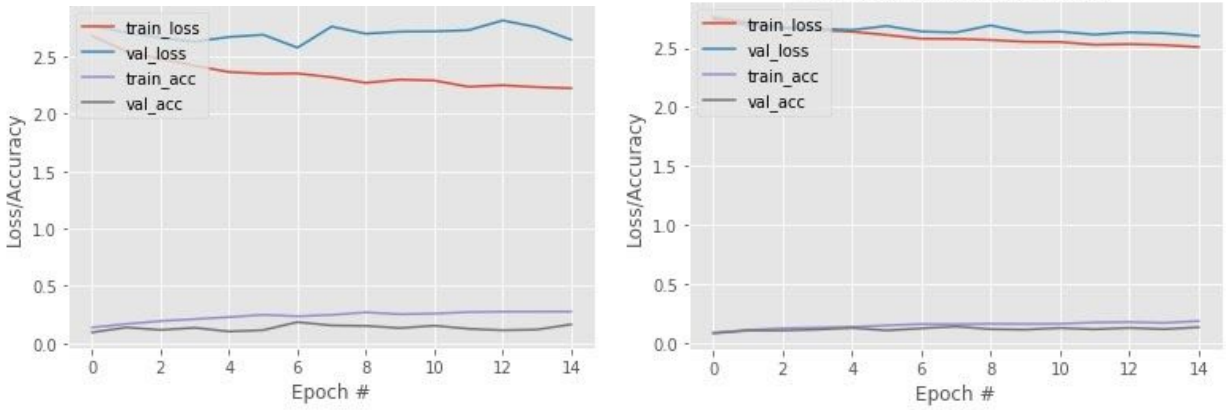

Figure 24. Loss and accuracy graph of Network 4 improved version, trained on men and women respectively

Unfortunately, this experiment did not yield very reliable results. The validation accuracy rate of the men dataset was highly unstable, constantly changing values. In women, the accuracy rate was only 0,1270, which is even lower than the first attempt.

Discussion

In this study, I have investigated the effect of different neural networks and their parameters on the accuracy of predicting professions from face images. For a more clear comparison, the results of all experiments are depicted in Table 4 and 5 altogether.

	Validation Accuracy	
Models	Images With Background	Images Without Background
Network 1	0,0388	0,0178
Network 2	0,1454	0,1592
Network 3 - Input Shape (100,100)	-	0,2707
Network 3 - Input Shape (300,300)	0,2899	0,2853
Network 4	0,3037	0,1372
Network 4 - Improved	0,3164	not reliable

Table 5. Results on Men Dataset

	Validation Accuracy	
Models	Images With Background	Images Without Background
Network 1	0,0868	0,0712
Network 2	0,0884	0,1097
Network 3 - Input Shape (100,100)	-	0,2046
Network 3 - Input Shape (300,300)	0,2011	0,2158
Network 4	0,2451	0,1436
Network 4 - Improved	0,2637	0,1270

Table 6. Results on Women Dataset

The data from the experiments shows that a simple neural network without convolutional and pooling layers is not powerful enough to perform complex classification tasks on face images. This argument still stands if the data variety is increased by adding image augmentation factor. In both cases, the validation accuracy is nowhere near a value which can indicate that our model is actually learning on this data and is associating facial features with professions. One way to improve these models can be to gather more data for each profession, however,it is not very likely that the model will be more accurate, because without convolutional and pooling layers, the model will fail to generate feature maps (Voulodimos et al., 2018) and it will still be very sensitive to any changes in the location of features (Buduma and Locascio, 2017) which are not desirable characteristics when dealing with image processing tasks.

When accuracy of two genders is compared, it can be seen that algorithms that are run on the men dataset gave better results. This is mostly because despite the fact that I have tried to keep the data uniformly distributed, there are more images of men than women (62046 images belonging to men and 54177 to women). Initially, there were roughly even numbers of pictures for each gender. However, when I performed the face detection on images and deleted those which did not contain faces, the number of images in the women dataset decreased.

Experiments on Network 3 shows that using deep learning considerably increases the accuracy. This network with multiple Dense, Pooling and Convolution layers has generated twofold increase in validation accuracy, which is a good step, however, there is still much more work to do considering the fact that the highest accuracy achieved by using this network is 0,3037. The experiment made on decreasing the input size while training on face images without background showed that smaller input size drives the accuracy towards low values, which indicates that the network is better at recognizing facial features when fed larger images, which was expected.

The accuracy rate of 0,3037 appears to be good, when compared to results of previous networks, nevertheless, it is not good enough for the purpose of this thesis.

The last experiment, which is using transfer learning, has not yielded results that was expected. Having trained on more than a million images with 1000 object classes, this pre-trained model did not give me better results compared to Network 3. In fact, Network 4 had significantly lower accuracy than the previous model when it is trained on images without background, indicating that using the Inception-V3 pre-trained model on face images does not hold much promise, since

it has not been heavily trained on faces and facial features. On the other hand, when it is used on images with background, it probably trains on background objects such as clothes and venues, thus providing a better accuracy.

5 Conclusion and Future Work Reference

This chapter will present the conclusions derived from this study and possible future works that can help solve the challenges and limitations of the experiments.

5.1 Conclusion

In this thesis, I have built and tested different neural networks for their accuracy on predicting professions based on face images. For neural networks to learn well, this task has required an extensive amount of data, hence, a considerable amount of time was put in gathering data. Before being able to feed this dataset into the networks, the data also had to be labeled. Using Python functions such as google_images_download and flow_from_directory have made the data collection and labeling process a lot more easier compared to if it has been done manually.

There were 4 neural networks that were tested and then improved. The first network did not learn, due to the fact that it was too simple to be dealing with such an image classification task. Using image augmentation as an addition in the second network has shown a slight improvement, however, still performed poorly because of the lack of layers that are needed for image processing tasks. The third network performed fairly well, compared to initial models. This is a result of multiple layers that employ pooling, convolution and other operations that are effective for image processing tasks. The fourth network, which benefits the implementation of a pre-trained model, Inception-V3, showed a minor improvement when trained on images with background, however, it was a disappointment on images without background. This shows that Inception-V3 was not the best option to use on face images for a face classification task, due to the fact that it does not have prior training purely on faces. The fact that it was somewhat good on images with background indicates that it might have trained on objects and not faces, such as clothes and places in the parts of the images that do not contain a face.

Overall results suggest that there is a need to experiment further and to a greater extent. Current validation accuracy results, which does not yield above %31,64 percent, are not satisfactory enough to achieve the goal, which is guiding young people towards a suitable career path.

5.2 Future Work

As stated before, in order to achieve the objectives of this thesis, the current accuracy scores are not enough. To improve this, further experiments and improvements are needed.

First and foremost, the dataset can be expanded. Including new professions can yield more influential results. In addition, more images for each profession can be added, and the data can be made to be more uniform, by balancing the gap in image number between two genders and perhaps collecting more names.

For Network 1 & 2, if the idea is the same, which is a very simple neural network with only a few Dense layers, it possibly can't be improved further, because it does not meet the requirements that are needed for an image processing task.

Experiments on Network 3 have shown that a deep neural network has the ability to learn from images, still, the results were not very accurate. To improve this network, more layers could be added and more parameters can be experimented with, such batch size, hidden units, dropout rates and loss/activation functions. These might have increased the performance of the models, however, there was not sufficient time for me to conduct these experiments.

Network 4 might have not generated very high accuracy, however, it can be improved so much more if right changes are made. Solutions to improve this network would be to train more layers of the pre-trained model if Inception-V3 is used, making the network recognize more of the facial features. This would take more time, which contradicts the main idea of transfer learning being fast, but would enhance the algorithm's learning ability on the faces. Another solution for using a pre-trained model is to use another pre-trained model that is heavily trained on faces.

All these improvements could lead to more accurate results, however, there was no sufficient time period for it to be done within this study. Using these references, models with better performances can be built, resulting in better prediction of the professions.

References

Morrell, Stephen L, et al. “Unemployment and Young People's Health.” *Medical Journal of Australia*, vol. 168, no. 5, 1998, pp. 236–240., doi:10.5694/j.1326-5377.1998.tb140139.x.

International Labour Organization, ILOSTAT Database

Winefeld, A. H., et al. “Psychological Concomitants of Satisfactory Employment and Unemployment in Young People.” *Social Psychiatry and Psychiatric Epidemiology*, vol. 23, no. 3, 1988, pp. 149–157., doi:10.1007/bf01794781.

Olivola, Christopher Y., et al. “The Many (Distinctive) Faces of Leadership: Inferring Leadership Domain from Facial Appearance.” *The Leadership Quarterly*, vol. 25, no. 5, 2014, pp. 817–834., doi:10.1016/j.leaqua.2014.06.002.

Bradski, Gary R., and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. OReilly, 2013.

Mitchell, Tom M. *Machine Learning*. McGraw Hill, 1997.

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Mehrotra, Kishan, et al. *Elements of Artificial Neural Networks*. NetLibrary, Inc., 1999.

Nilsson, N. J. (n.d.). Introduction to Machine Learning [An Early Draft of A Proposed Textbook]. Retrieved from https://ai.stanford.edu/~nilsson/MLBOOK.pdf

Voulodimos, Athanasios, et al. “Deep Learning for Computer Vision: A Brief Review.” *Computational Intelligence and Neuroscience*, vol. 2018, 2018, pp. 1–13., doi:10.1155/2018/7068349.

Rosenblatt, F. “The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.” *Psychological Review*, vol. 65, no. 6, 1958, pp. 386–408., doi:10.1037/h0042519.

Gallant, S.i. “Perceptron-Based Learning Algorithms.” *IEEE Transactions on Neural Networks*, vol. 1, no. 2, 1990, pp. 179–191., doi:10.1109/72.80230.

Buduma, Nikhil, and Nicholas Locascio. *Fundamentals of Deep Learning: Designing next-Generation Machine Intelligence Algorithms*. O´Reilly, 2017.

Courant, Richard, and Fritz John. *Introduction to Calculus and Analysis*. Springer, 2000.

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature,521(7553), 436-444. doi:10.1038/nature14539

Buck, R. Creighton., and Ellen F. Buck. *Advanced Calculus*. McGraw-Hill Book Company, 1965.

Goodfellow, Ian, et al. *Deep Learning*. The MIT Press, 2017. http://www.deeplearningbook.org

Gu, Jiuxiang, et al. “Recent Advances in Convolutional Neural Networks.” *Pattern Recognition*, vol. 77, 2018, pp. 354–377., doi:10.1016/j.patcog.2017.10.013.

Albawi, Saad, et al. “Understanding of a Convolutional Neural Network.” *2017 International Conference on Engineering and Technology (ICET)*, 2017, doi:10.1109/icengtechnol.2017.8308186.

Torrey, Lisa, and Jude Shavlik. “Transfer Learning.” *Handbook of Research on Machine Learning Applications and Trends*, pp. 242–264., doi:10.4018/978-1-60566-766-9.ch011.

Pan, Sinno Jialin, and Qiang Yang. “A Survey on Transfer Learning.” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, 2010, pp. 1345–1359., doi:10.1109/tkde.2009.191.

Brownlee, J. (2016). *Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras*. Place of publication not identified: Machine Learning Mastery.

Szegedy, Christian, et al. "Rethinking the Inception Architecture for Computer Vision." *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, doi:10.1109/cvpr.2016.308.

Image References

[1] https://www.sqlservercentral.com/articles/understanding-machine-learning

[2]https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76

[3] https://pythonmachinelearning.pro/perceptrons-the-first-neural-networks/

[4] http://computationalsciencewithsuman.blogspot.com/p/single-layer-and-multilayer-feed.html

[5] https://www.offconvex.org/2018/11/07/optimization-beyond-landscape/

[6] https://www.deeplearningwizard.com/deep_learning/boosting_models_pytorch/lr_scheduling/

[7] Buduma, N., & Locascio, N. (2017). Fundamentals of deep learning: designing next-generation machine intelligence algorithms. Canada: O´Reilly.

[8] https://codelabs.developers.google.com/codelabs/cpb102-txf-learning/index.html#1

NON-EXCLUSIVE LICENCE TO REPRODUCE THESIS AND MAKE THESIS PUBLIC

I, Gülce Naz Mert,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, Artificial Intelligence Based Profession Prediction Using Facial Analysis, supervised by PhD Gholamreza Anbarjafari, M.A. Doğuş Karabulut.

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Gülce Naz Mert

20/05/2020