GENDER DETECTION AND AGE PREDICTION USING OPENCV AND CAFFE

A PROJECT REPORT

Submitted in partial fulfillment of the requirements

for the award of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE &ENGINEERING

Submitted by

NAGA SATYA SAI PAVIRALA	**(18A21A0564)**
KUMMARAPURUGU RAMESH	**(18A21A05A1)**
POLISETTY PAVAN SITARAM	**(18A21A0580)**
TUMU RAM SAI JAGAN	**(18A21A05A5)**
SALADI RAMANJANEYULU	**(19A25A0511)**

Under the Esteemed Guidance of

Dr. B. RAMA KRISHNA, Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SWARNANDHRA COLLEGE OF ENGINEERING &TECHNOLOGY

(Approved by AICTE & Affiliated to JNTU-Kakinada, Accredited by NAAC)

(AUTONOMOUS)

Seetharampuram, Narsapur-534 280, W.G. Dt. (A.P)

2018-2022

SWARNANDHRA COLLEGE OF ENGINEERING &TECHNOLOGY

(Approved by AICTE & Affiliated to JNTU-Kakinada, Accredited by NAAC)

(AUTONOMOUS)

Seetharampuram, Narsapur-534 280, W.G. Dt. (A.P)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

Certified that this Project Report titled **"GENDER DETECTION AND AGE PREDICTION USING OPENCV AND CAFFE"** is a bonafide work of **NAGA SATYA SAI PAVIRALA (18A21A0564), KUMMARAPURUGU RAMESH (18A21A05A1), POLISETTY PAVAN SITARAM (18A21A0580), TUMU RAM SAI JAGAN (18A21A05A5), SALADI RAMANJANEYULU (19A25A0511) of B. Tech VIII SEMESTER** who carried out the work under my supervision, and submitted in partial fulfillment of the requirements for the award of the degree of "**Bachelor of Technology** in "**COMPUTER SCIENCE AND ENGINEERING"** during the academic year **2018-2022**.

Supervisor

Dr. B. RAMA KRISHNA

Professor.

Head of the Department

Dr. P. SRINIVASULU

Professor.

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of every task during my dissertation would be complete without the mention of the people who made it possible. I consider it my privilege to my gratitude and respect to all who guided, inspired and helped me incompletion of my final project.

I extend my heartfelt gratitude to the almighty for giving me strength in Proceeding with this final project report on "**GENDER DETECTION AND AGE PREDICTION USING OPENCV AND CAFFE**".

I would like to express my deep sense of gratitude to Principal **Dr. S. Suresh Kumar, BE, MS, M. Tech, Ph.D.** for his valuable help and guidance.

I would like to express my sincere thanks to HOD **Dr. P. Srinivasulu, M. Tech., Ph.D., Professor & Head,** Department of Computer Science and Engineering for valuable suggestions at the time of need.

I would like to express my profound sense of gratitude to Final Project Coordinator, **Mr. P. Govinda Raju, Assistant Professor,** Department of Computer Science and Engineering for his kind co-operation and help.

I would like to give my warmest thanks to Final Project Guide **, Dr. B. Rama Krishna, Professor,** Department of Computer Science and Engineering for his consistent encouragement and earnest support to complete it successfully.

NAGA SATYA SAI PAVIRALA (18A21A0564)

KUMMARAPURUGU RAMESH (18A21A05A1)

POLISETTY PAVAN SITARAM (18A21A0580)

TUMU RAM SAI JAGAN (18A21A05A5)

SALADI RAMANJANEYULU (19A25A0511)

DECLARATION

We certify that

a. The project work contained in the report is original and has been done by under the guidance of my supervisor.

b. The work has not been submitted to any other university for the award of any degree or diploma.

c. The guidelines of the university are followed in writing the report.

Date:

Place:

NAME OF THE STUDENT	REG. NO	SIGNATURE
NAGA SATYA SAI PAVIRALA	18A21A0564	
KUMMARAPURUGU RAMESH	18A21A05A1	
POLISETTY PAVAN SITARAM	18A21A0580	
TUMU RAM SAI JAGAN	18A21A05A5	
SALADI RAMANJANEYULU	19A25A0511	

ABSTRACT

ABSTRACT

To build a gender and age detector that can approximately guess the gender and age of the person (face)in a picture or through webcam. In this Python Project, we had used Deep Learning to accurately identify the gender and age of a person from a single image of a face. We used the models trained by using caffe framework. The predicted gender may be one of 'Male' and 'Female', and the predicted age may be one of the following ranges- (0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 –53), (60 – 100) (8 nodes in the final softmax layer). It is very difficult to accurately guess an exact age from a single image because of factors like makeup, lighting, obstructions, and facial expressions. And so, we made this a classification problem instead of making it one of regression. For this project we used the Adience dataset in Kaggle.The dataset is available in the public domain. This dataset serves as a benchmark for face photos and is inclusive of various real-world imaging conditions like noise, lighting, pose, and appearance. The images have been collected from Flickr albums and distributed under the Creative Commons (CC) license. It has a total of 26,580 photos of 2,284 subjects in eight age ranges (as mentioned above) and is about 1GB in size. This dataset serves as a benchmark for face photos and is inclusive of various real-world imaging conditions like noise, lighting, pose, and appearance. The images have been collected from Flickr albums and distributed under the Creative Commons (CC) license. It models we used are trained on this dataset.

CONTENTS

INTRODUCTION

CHAPTER-1

INTRODUCTION

1.1 Introduction:

Age and gender play fundamental roles in social interactions. Languages reserve different salutations and grammar rules for men or women, and very often different vocabularies are used when addressing elders compared to young people.

Therefore automated translation services and other forms of speech generation can factor in gender and age classification of subjects to improve their performance. Having an idea about the age and gender of a person makes the task of interaction with them easier.

Age and gender play fundamental roles in social interactions. Languages reserve different salutations and grammar rules for men or women, and very often different vocabularies are used when addressing elders compared to young people. Despite the basic roles these attributes play in our day-to-day lives, the ability to automatically estimate them accurately and reliably from face images is still far from meeting the needs of commercial applications. This is particularly perplexing when considering recent claims to super-human capabilities in the related task of face recognition. Past approaches to estimating or classifying these attributes from face images have relied on differences in facial feature dimensions or “tailored” face descriptors. Most have employed classification schemes designed particularly for age or gender estimation tasks, including and others. Few of these past methods were designed to handle the many challenges of unconstrained imaging conditions.

Faces from the Adience benchmark for age and gender classification. These images represent some of the challenges of age and gender estimation from real-world, unconstrained images. Most notably, extreme blur (low-resolution), occlusions, out-of-plane pose variations, expressions and more. Exploit the massive numbers of image examples and data available through the Internet in order to improve classification capabilities. In this paper we attempt to close the gap between automatic face recognition capabilities and those of age and gender estimation methods. To this end, we follow the successful example laid down by recent face recognition systems: Face recognition techniques described in the last few years have shown that tremendous progress can be made by the use of deep

convolutional neural networks (CNN). We demonstrate similar gains with a simple network architecture, designed by considering the rather limited availability of accurate age and gender labels in existing face data sets. We test our network on the newly released Adience benchmark for age and gender classification of unfiltered face images. We show that despite the very challenging nature of the images in the Adience set and the simplicity of our network design, our method outperforms existing state of the art by substantial margins.

1.2 Motivation:

With the approval of the General Data Protection Regulation (GDPR), the use of a client's personal information has become very strict. Since currently, it is not legal to store a person's sensitive information (name, email, phone number) without one's consent, there is a need to have an alternative way to gather clients information for marketing purposes. Such alternative ways include the use of artificial intelligence (AI) models that can make use of a client's information to extract useful knowledge. The incremental evolution of AI models and their successes on specific tasks, such as visual recognition and classification problems, have presented a new way to solve complex issues. Such issues that have in the last years received an increasing amount of attention are age and gender classification problems. There have been multiple approaches that focus their work on achieving a high accuracy rate for this kind of predictions. Latest approaches commonly use Convolutional Neural Networks (CNN), as those have been presenting state-of-the-art results, although a number of limitations to this kind of networks reduce the progress that such models can achieve. One of the limitations of such systems is the increased amount of data required when training the model. The model's accuracy rates depend not only on the implemented network but also on the data that is used to train it. The training data is what allows the system to learn to identify and predict outcomes correctly, therefore, the more data that is fed to the network, the better the results. Having large datasets to train a network is commonly a problem, as such datasets need to be labeled for the specific problem at hand and usually require an initial cleanup process before feeding it to the model. Another common problem, when it comes to face detection or age/gender classification, and usually one of the most important ones, are partial occlusions and low-quality images. Those influence directly the outcome results as the model has less information to work on, which makes it harder to predict.The same applies when it is a human making the prediction. If the image has low quality, it is harder for a human to be able to understand what is being seen, and therefore, to make a prediction.

1.3 Context:

In our specific study case, the focus is to allow a supermarket to gather client statistics to make informed decisions, increase the quality of their service, and improve customer service. This is done making use of face recognition and age/gender classification algorithms applied over client's images, which allows for such statistics to be generated.With this in mind, this research has the intent of creating a system which can efficiently detect and categorize people in images, identifying what their gender is and, given a set of pre-defined age classes, being able to predict in which one that person falls in.

One way to attain this is to use age and gender automatic, i.e., anonymized, classification methods. This can be achieved using neural networks, provided that these are able to produce high confidence predictions, i.e., relevant and trustworthy. Therefore, this project falls under the area of artificial intelligence (AI) since it uses Deep Learning models in order to extract information from images and make accurate predictions using those images.

The proposed system works with the client's images as input, using those to make the gender/age classifications and to, therefore, deliver that information to the users.

1.4 Need for the Project:

Automatic age and gender classification has become relevant to an increasing amount of applications, particularly since the rise of social platforms and social media.

Performance of existing methods on real-world images is still significantly lacking.

By learning representations through the use of deep-convolutional neural networks (CNN), a significant increase in performance can be obtained on these tasks. We used a simple convolutional net architecture that can be used even when the amount of learning data is limited.

1.5 Project Overview:

We used Adience dataset for age and gender classification.Trained CNN (Convolutional Neural Networks) using Adience dataset for age and gender prediction. Created a python application using OpenCV deep learning module to perform real time age and gender detection.

LITERATURE REVIEW

CHAPTER-2

LITERATURE REVIEW

2.1 Literature Survey:

We developed our project by taking the reference of PhD publication paper of Tal Hassner and GilLevi .Our main intention is to find the criminals by filtering them by using particular age group and gender. For us to be able to build a system and choose which models/techniques to use, we need to know what is being used by other approaches with results that could satisfy our goals. First, we need to identify which approach is being used the most with satisfactory results: this is the case of Deep Learning, which is used across all recent papers we investigated, where all of them use, more specifically, Convolutional Neural Networks. Deep learning has shown considerable improvements when compared to older algorithms, especially when processing imagesand videos, which covers problems like object detection, object recognition and speech recognition,which are areas where our theme falls into. This explains why latest approaches are currently all adopting Deep Learning to solve these issues.

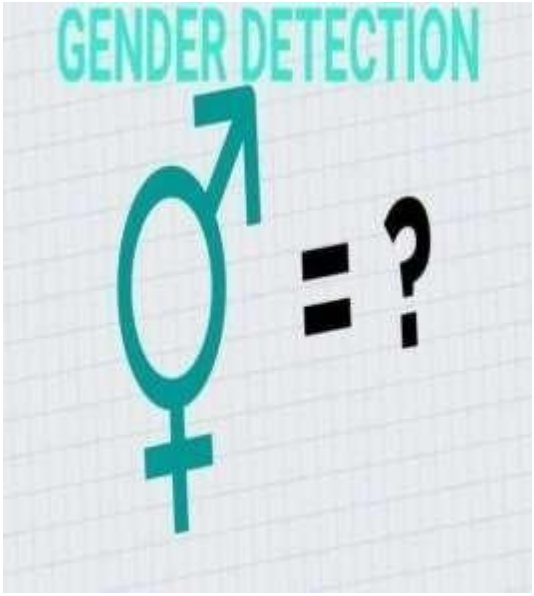
2.2 Existing System:

The existing systems till date are less accurate and they can't predict accurately in noise conditions and less lighting conditions developed using UTK face. Over the past several years, the studies are undergoing on the areas of facial feature extraction and predicting age and gender based on that. To address this concern scientists had come up with various approaches, and each of these approaches solved some critical problems that were raised in this field. For predicting age and gender accurately, even some of the minor differences in the images should be extracted carefully. Extracting facial features to that extent is challenging and only a few approaches focused on solving this problem. Those minor differences include size of the eyes, ears, mouth and the distances between them. most of early methods have focused on images which were maintained in lab conditions.

2.3 Proposed System:

Here we developed this system using adience data set which is a benchmark data set which gives better detection and prediction than existing systems.In this proposed system, we will develop a deep learning model using Convolutional neural networks and Caffe framework. The model receives the image, passes it through different layers by reducing the size in each layer. We then train our model using the Adience Benchmark Dataset from Kaggle which contains over 26,000 images of human beings which include different ethnicity, color, and many more factors. The dataset also provides a label for each image. Once the model is trained, we can use the model for testing. We first detect the presence of a human in each image. Then, we process the image using the Open CV framework to obtain all the human faces present in the image. Each face is then processed by a developed deep learning model to get the output label which essentially gives us the age and gender of each person.

2.4 Gender Detection and Age Prediction:



Age classification. The problem of automatically extracting age related attributes from facial images has received increasing attention in recent years and many methods have been put fourth. A detailed survey of such methods can be found in and, more recently, in. We note that despite our focus here on age group classification rather than precise age estimation (i.e., age regression), the survey below

includes methods designed for either task. Early methods for age estimation are based on calculating ratios between different measurements of facial features. Once facial features (e.g. eyes, nose, mouth, chin, etc.) are localized and their sizes and distances measured, ratios between them are calculated and used for classifying the face into different age categories according to hand-crafted rules. More recently, uses a similar approach to model age progression in subjects under 18 years old. As those methods require accurate localization of facial features, a challenging problem by itself, they are unsuitable for in-the-wild images which one may expect to find on social platforms. On a different line of work are methods that represent the aging process as a subspace or a manifold. A drawback of those methods is that they require input images to be near-frontal and well-aligned. These methods therefore present experimental results only on constrained data-sets of near-frontal images (e.g UIUC-IFP-Y, FG-NET and MORPH). Again, as a consequence, such methods are ill-suited for unconstrained images. Different from those described above are methods that use local features for representing face images. In Gaussian Mixture Models (GMM) were used to represent the distribution of facial patches. In [GMM were used again for representing the distribution of local facial measurements, but robust descriptors were used instead of pixel patches. Finally, instead of GMM, Hidden-Markov Model, super-vectors were used in for representing face patch distributions. An alternative to the local image intensity patches are robust image descriptors: Gabor image descriptors were used in along with a Fuzzy-LDA classifier which considers a face image as belonging to more than one age class. In a combination of Biologically-Inspired Features (BIF) and various manifold-learning methods were used for age estimation. Gabor and local binary patterns (LBP) features were used in along with a hierarchical age classifier composed of Support Vector Machines (SVM) to classify the input image to an age-class followed by a support vector regression to estimate a precise age. Finally, proposed improved versions of relevant component analysis and locally preserving projections . Those methods are used for distance learning and dimensionality reduction, respectively, with Active Appearance Models as an image feature. All of these methods have proven effective on small and/or constrained benchmarks for age estimation. To our knowledge, the best performing methods were demonstrated on the Group Photos benchmark. In state-of-the-art performance on this benchmark was presented by employing LBP descriptor variations and a dropout-SVM classifier. We show our proposed method to outperform the results they report on them are challenging Adience benchmark, designed for the same task.

Gender classification. A detailed survey of gender classification methods can be found in and more recently in. Here we quickly survey relevant methods. One of the early methods for gender classification used a neural network trained on a small set of near-frontal face images. In the combined 3D structure of the head (obtained using a laser scanner) and image intensities were used for classifying gender. SVM classifiers were used by, applied directly to image intensities. Rather than using SVM, used AdaBoost for the same purpose, here again, applied to image intensities. Finally, viewpoint-invariantage and gender classification was presented by. More recently, used the Webers Local texture Descriptor for gender recognition, demonstrating near perfect performance on the FERET benchmark.In intensity, shape and texture features were used with mutual information, againobtaining near- perfect results on the FERET benchmark.

2.5 Deep Learning

Machine Learning is mainly used for object identification, classification problems, and prediction problems, and is divided into Supervised Learning and Unsupervised Learning. Deep learning can be described as a set of techniques used as part of Machine Learning, more specifically used in neural networks, which work with a set of layers, where the layers allow the data that is the input to the system to be decomposed and analyzed; those layers are not specifically designed for each problem, but they are part of a generic procedure that is adaptable to multiple problem types.So the same network structure can be used in different problems. How the network is trained and which data is used is what will distinguish the behavior of such a model.

In the past years, Deep Learning has shown many improvements on various tasks that were for many years hard to solve by other algorithms, showing better results in multiple studies compared to previous works . Such tasks include problems where the input has a complex structure and cannot be easily learned by traditional artificial intelligence algorithms, such as image classification (where each image is represented as a pixel array) and speech recognition.

2.5.1 Convolutional Neural Network

Convolutional neural networks are a special type of feed-forward networks. These models are designed to emulate the behavior of a visual cortex. CNNs perform very well on visual recognition tasks. These networks have 3 types of layers: Input layer, hidden layer and output layer. In these networks, data moves from the input layer through the hidden nodes and to the output nodes.

The convolutional, pooling and ReLU layers act as learnable features extractors, while the fully connectedlayers acts as a machine learning classifier.

Since 2012, with the ImageNet competition, that these networks are achieving better results in image classification than other algorithms, which made them since then the most used approach for all recognition and detection tasks, approaching almost the same results as human performance. In this same contest, A. Krizhevsky et al. created a Convolutional Neural Network to classify images into a range of 1000 classes, attaining a test error rate of 15.3%, which was 11% better than the second-best entry in the competition .

Convolutional Neural Networks (CNN) are a type of neural network that generalizes better than previous neural networks (e.g., multi-layer perceptrons). CNN is designed to process data in the form of multiple arrays, passing the input between layers that extract the necessary features of the input and assign to those features calculated weights (filters) that will decide which are more important.

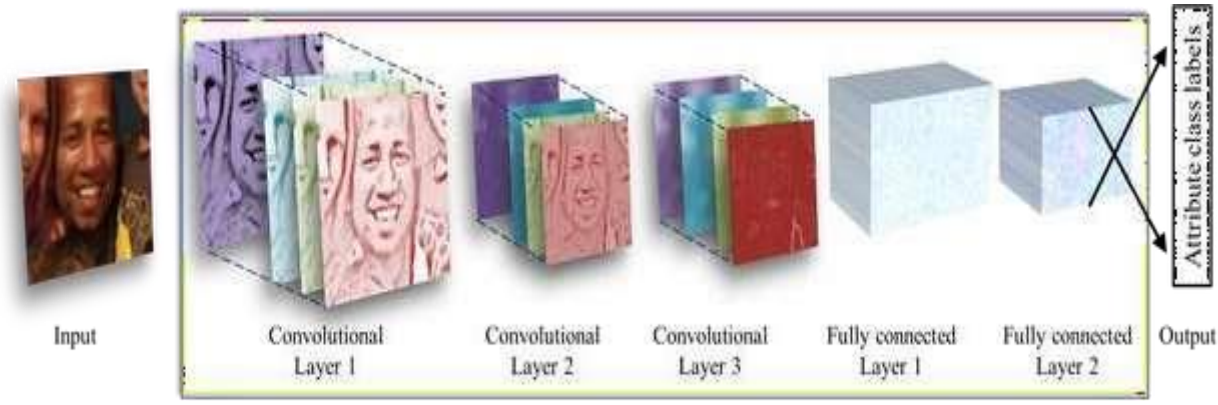


Figure 1. Illustration of our CNN architecture. The network contains three convolutional layers, each followed by a rectified linear operation and pooling layer. The first two layers also follow normalization using local response normalization [28]. The first Convolutional Layer contains 96 filters of 7×7 pixels, the second Convolutional Layer contains 256 filters of 5×5 pixels, The third and final Convolutional Layer contains 384 filters of 3 × 3 pixels. Finally, two fully-connected layers are added, each containing 512 neurons. See Figure 3 for a detailed schematic view and the text for more information.

Figure 1 depicts a typical architecture of a CNN, where the input image is fed into another layer and, then, the output of that layer is fed into the next ones. Each layer has a specific purpose. The Convolutional Layer is the one responsible for extracting features from the image (resulting in a feature map). It uses a set of filters to make computations over the initial vector and extract feature.

From it, feeding the result to the next layer; this can be used, for example, to detect edges in images.

The Pooling layer reduces the dimensionality of the feature map, retaining only the most important information depending on the type of pooling applied. This can be seen as downscaling an image, the image is not as clear as before, but we can still understand what it represents. This helps in improving the computation time and the results.

The ReLU layer is used after each convolution and its purpose is to replace every negative pixel in the feature map by zero. The main goal of such operation is to introduce non-linearity in the network (opposed to the linear convolutional layer) in order to simulate real data, which would be non-linear. Finally, the Fully Connected Layer is the one that classifies the results based on the high- level features extracted in previous layers, assigning a classification to the input image.

2.6 Gender and Age Estimation:

Following the same outcome as for the investigation regarding face detection systems, also for gender and age classification models, most authors use CNN. This is because deep neural networks have shown good performances in predicting the age and gender from images. This was validated by S. Lapuschkinetal where they achieved an accuracy of 92.6%on gender prediction and 62.8% on exact age class prediction. For this, they used a pre-trained model (VGG-16), using both ReLU and Pooling layers in their CNN. Other work also followed the same approach using a pre-trained model and achieve similar results on gender classification using the Caffe model and Face Recognition pre-trained models.

Gil Levi and Tal Hassner additionally also included data augmentation techniques (over-sampling and center-crop) to allow the testing dataset to be expanded and the model to be trained more efficiently . They also applied dropout learning during the testing phase in order to reduce over-fitting of their model, which consists of randomly ignoring the output of some layers. Their accuracy percentages were 86.8% for gender classification and 50.7% for age classification.

R.Ranjanetal justify their pre-trained model pick because using a network that was previously trained on a face recognition task has already learned information of a face that can be used by other face- related tasks. Additionally, and opposed to similar models that were analyzed, they also introduced the concept of having lower layer parameters shared by all the tasks done.

This translates into network that does all tasks since all of them need similar characteristics of the face to be learned. With this, they achieved a gender accuracy of 93%and a Mean Absolute Error (MAE)on age prediction of 2.00 (average age error of 2 years). Considering the gender results, this model outperformed all others so far.

Another case also achieved over 90% gender accuracy using Face Recognition pre- trained models. As opposed to the previous work, they created multiple models for each specific task because having separate networks allowed them to design faster and more portable models. Additionally, the running time of all models combined is less than the time that the all-in-one model from Rajeev Ranjanetal takes. In this work, they were also able to achieve the highest age classification accuracy reported so far of 70.5% on actual group estimation. To note that the 1-off group classification accuracy for this achieved 96.2% (1-off is when a person is classified in one group immediately above or below the correct one).

We should also note that it might be important to consider the encoding of our target labels depending on the problem at hands. This was one of the concerns from G. Antipovetal, where multiple age encodings were tested. There could be multiple approaches to implement this,as opposed to gender classification, where we only have two possible values (male/female). For example, in age estimation, we could be trying to predict exact age (person A has 29 years) or classify a person in an age group (Person A is in the group [20-30]). Their experiments showed that the encoding used could vary the results of age estimation in about 0.95 (MAE). The best encoding used was the Label Distribution Age Encoding, which treats the age as a set of classes representing all possible ages, having as the content of each vector cell a Gaussian Distribution centered at the target age, as opposed to what happens in pure per-year classification which contains a cell value as a binary encoding (0/1).

Table 1 shows a summary of the papers. To note that some of the papers approach age as a classification problem, and others as a regression problem, and therefore their measure indexes are different (accuracy and MAE) – those different indexes are not comparable to each other, so we are including both in the results below separately. The results also depend on the dataset used,so we only consider the best results for each paper.

	Results		
	Gender Prediction (%)	Age Prediction	
Paper	**Accuracy**	**Accuracy/ 1-off (%)**	**MAE**
[10] Eran Eidinger, Roee Enbar, Tal Hassner, 2014	88.6	66.6 / 94.8	
[9] Gil Levi, Tal Hassner, 2015	86.8	50.7 / 84.7	
[6] Sebastian Lapuschkin, Alexander Binder, Klaus-Robert Muller, Wojciech Samek, 2017	92.6	62.8 / 95.8	
[11] Afshin Dehghan, Enrique G. Ortiz, Guang Shu, Syed Zain Masood, 2017	91	70.5 / 96.2	
[12] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D. Castillo, Rama Chellappa, 2017	93.2		2.00
[13] Grigory Antipov, Moez Baccouche, Sid-Ahmed Berrani, Jean-Luc Dugelay, 2017	95		2.35

Table 1- Age and Gender classification related papers.

2.6.1 CNN for Gender and Age estimation:

Data-sets for age and gender estimation from real-world social images are therefore relatively limited in size and presently no match in size with the much larger image classification data-sets.

Over fitting is common problem when machine learning based methods are used on such small image collections. This problem is exacerbated when considering deep convolutional neural networks due to their huge numbers of model parameters. Care must therefore be taken in order to avoid over fitting under such circumstances.

SYSTEM DEVELOPMENT

CHAPTER-3

SYSTEM DEVELOPMENT

This section describes the architecture of the system. This system was developed using python, therefore the users only need to execute the main python executable with the relevant parameters described throughout this section in order to use it. A more detailed explanation over the files and directories that make up the system is described in the attachments section, under the User Guide section.

3.1 Requirements:

Software Requirements:

- Operating System : Windows
- IDE : Microsoft Visual Studio Code
- Programming language : Python
- Packages : Open CV
- Framework : Caffe

Minimum Hardware Requirements:

- Processor : Intel i3 or greater
- RAM : 4GB
- Hard disk : 30GB

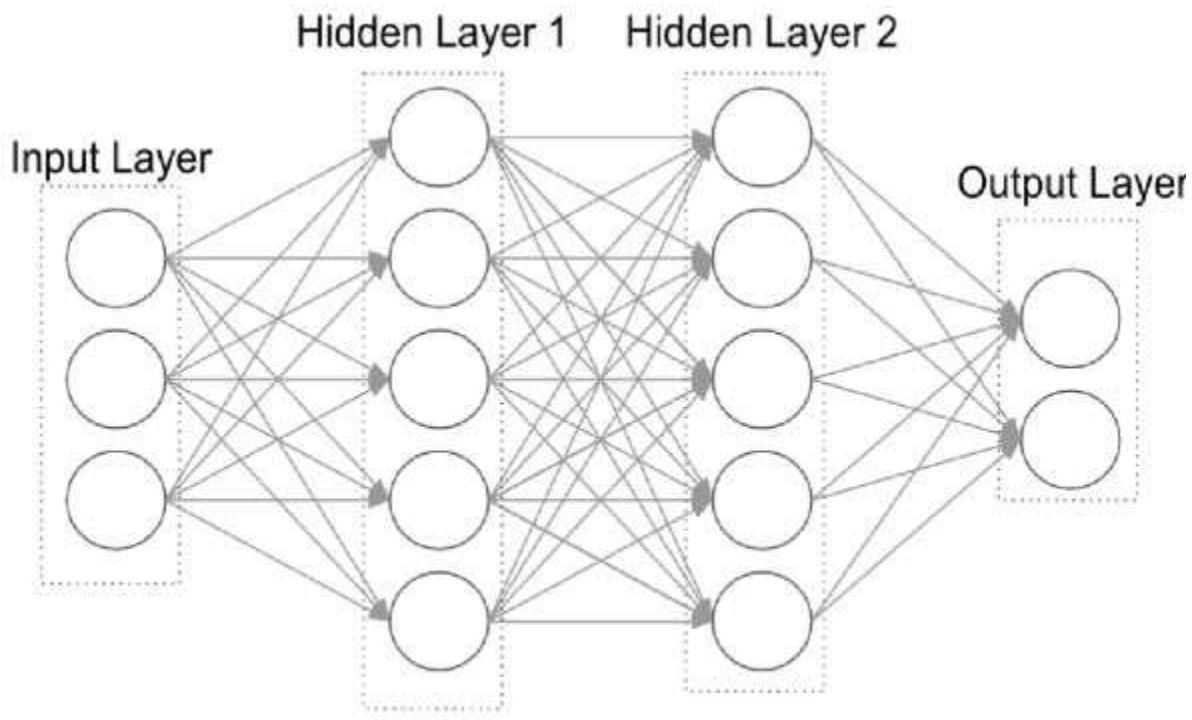
To proceed with the implementation of the system, we need to identify the requirements that are needed for us to be able to conclude it. We need a model that can extract a face from an image, and then feed it to another model that is able to predict the age and gender based on the facial characteristics of the person that is passed as input. From this, and taking the literature review into consideration, we can summaries the main system requirements that need to be taken into consideration throughout the system implementation, which is the following:

- The system should be capable of detecting faces in images with a high accuracy rate (greater than 90%), following other state-of-the-art results on such models.
- The system should be capable of predicting the age class of a person. The accuracy should be similar (or better) with other state-of-the-art models – it should have an accuracy rate greater than 60%.
- The system should be capable of predicting the gender class of a person. It should have a high accuracy rate (greater than 90%), following similar results from other models.
- The system should allow the user to customize the age classes.
- The system should allow users to validate the results. The results from the predictions should be saved into a specific folder where users can validate them manually if required. The results that are stored in such folder should be the ones that surpass a certain confidence threshold defined by the user.
- The system should allow the user to choose an additional module – the Caffe Network – for gender classification in case the initial classification surpasses a pre-set threshold of confidence.

Age classes, for validation purposes, used throughout this document are defined as:

- **Young Child** - Age 0-2.
- **Child** - Age 4-6.
- **Young Teen** - Age 8-12.
- **Teen** - Age 15-20.
- **Young Adult** – Age 25-32.
- **Adult** - Age 38-43.
- **Young Senior** –Age 48-53.
- **Senior** – Age 60-100.

3.2 Design:



Feedforward neural network with 2 hidden layers

Considering the fact that we need a model that can detect faces and make age/gender predictions, then we can break the system down into two modules in order to simplify our implementation: the first module which is the face detection, and the second module responsible for the age and gender classification. Additionally, dividing the global model into smaller modules ,allows us to have a faster and more portable model, which follows the approach from where their system had multiple models for each task, which was also the network which achieved better accuracy results from our initial investigation. Therefore, the first step is to find all the faces present in an image, and the next step is to do the classifications using the faces that were extracted.

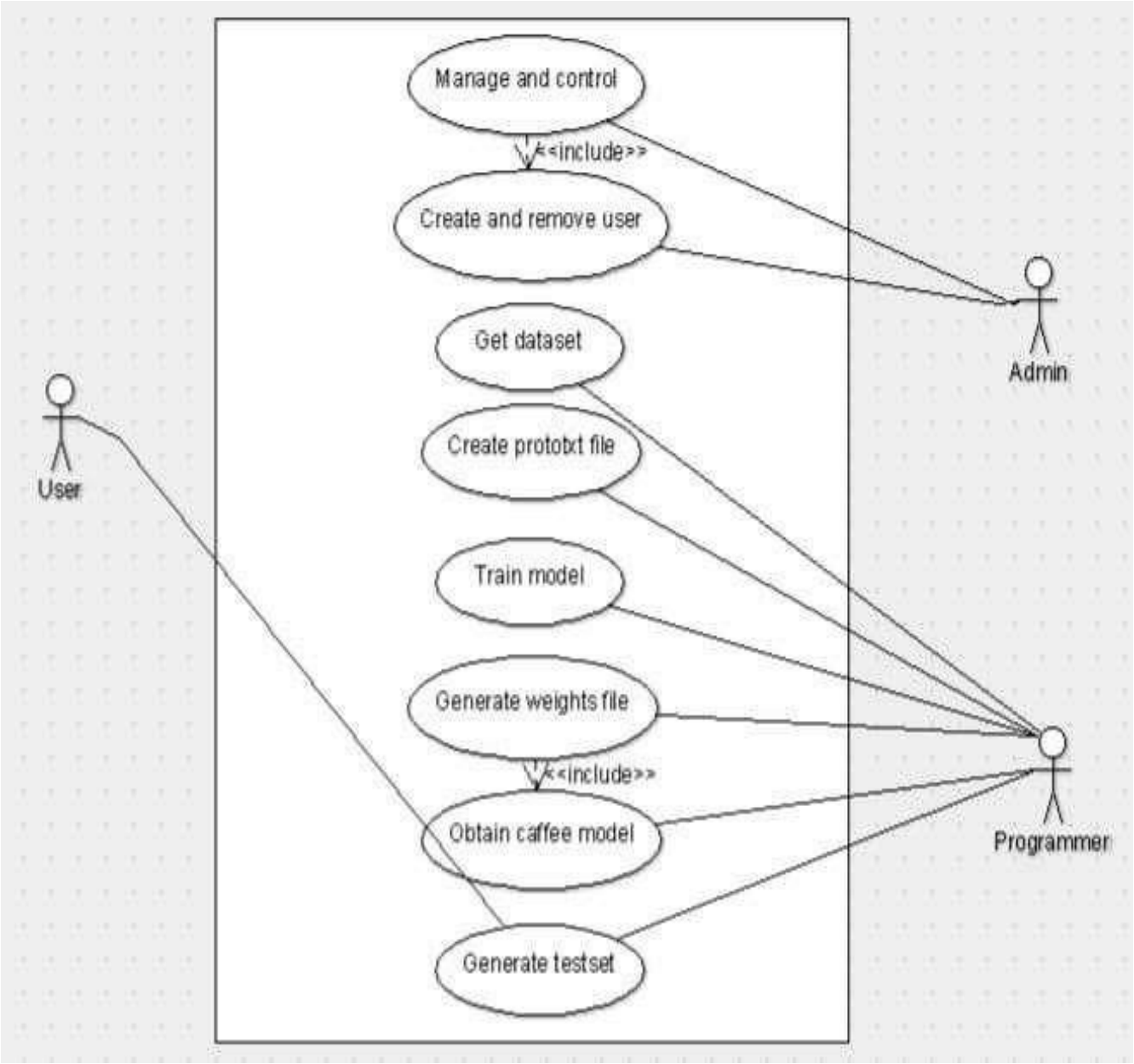
3.3 Algorithm Used:

J48 Classification Algorithm

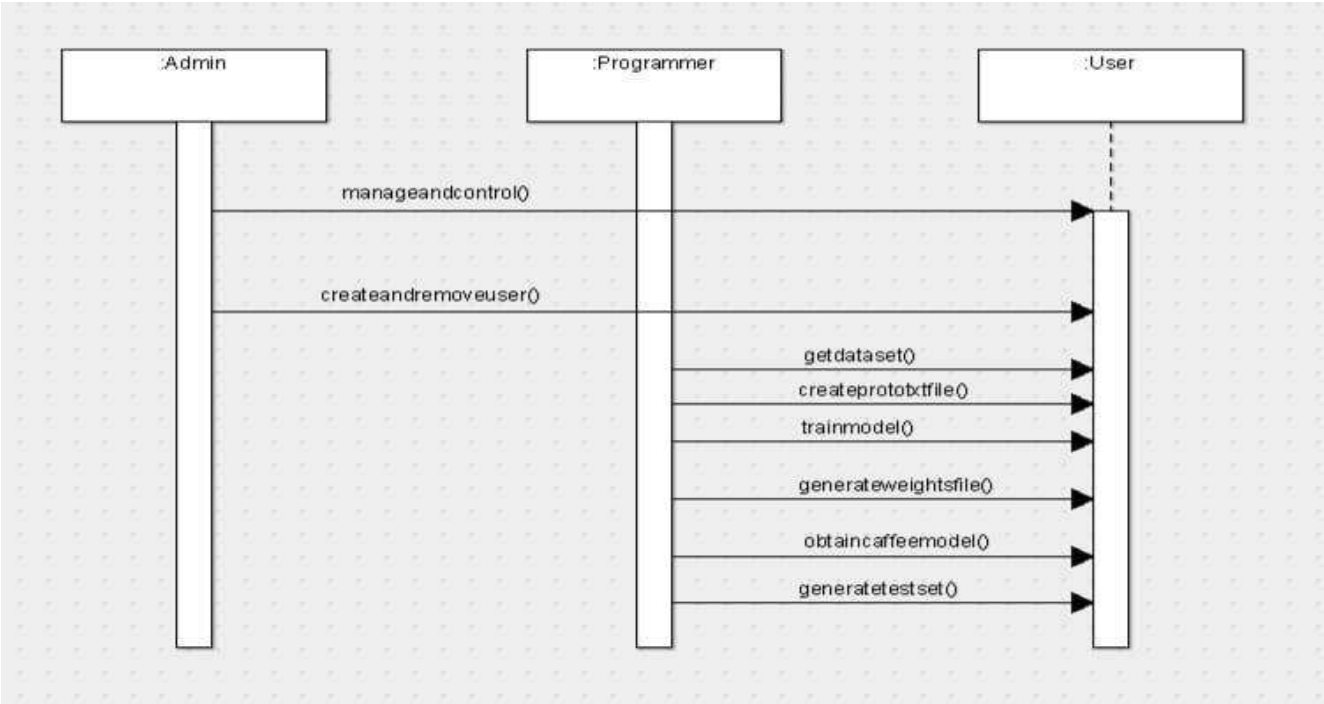
```
1: Create a root node N;
2: IF (T belongs to same category C)
       {leaf node = N;
         Mark N as class C;
         Return N;
       }
3: For i=1 to n
       {Calculate Information_gain (Ai);}
4: ta= testing attribute;
5: N.ta = attribute having highest information_gain;
6: if (N.ta == continuous )
         { find threshold;}
7: For (Each T in splitting of T)
8:       if (T is empty)
            {child of N is a leaf node;}
            else
            {child of N= dtree T)}
10: calculate classification error rate of node N;
11: return N;
```

3.4 UML Diagrams:

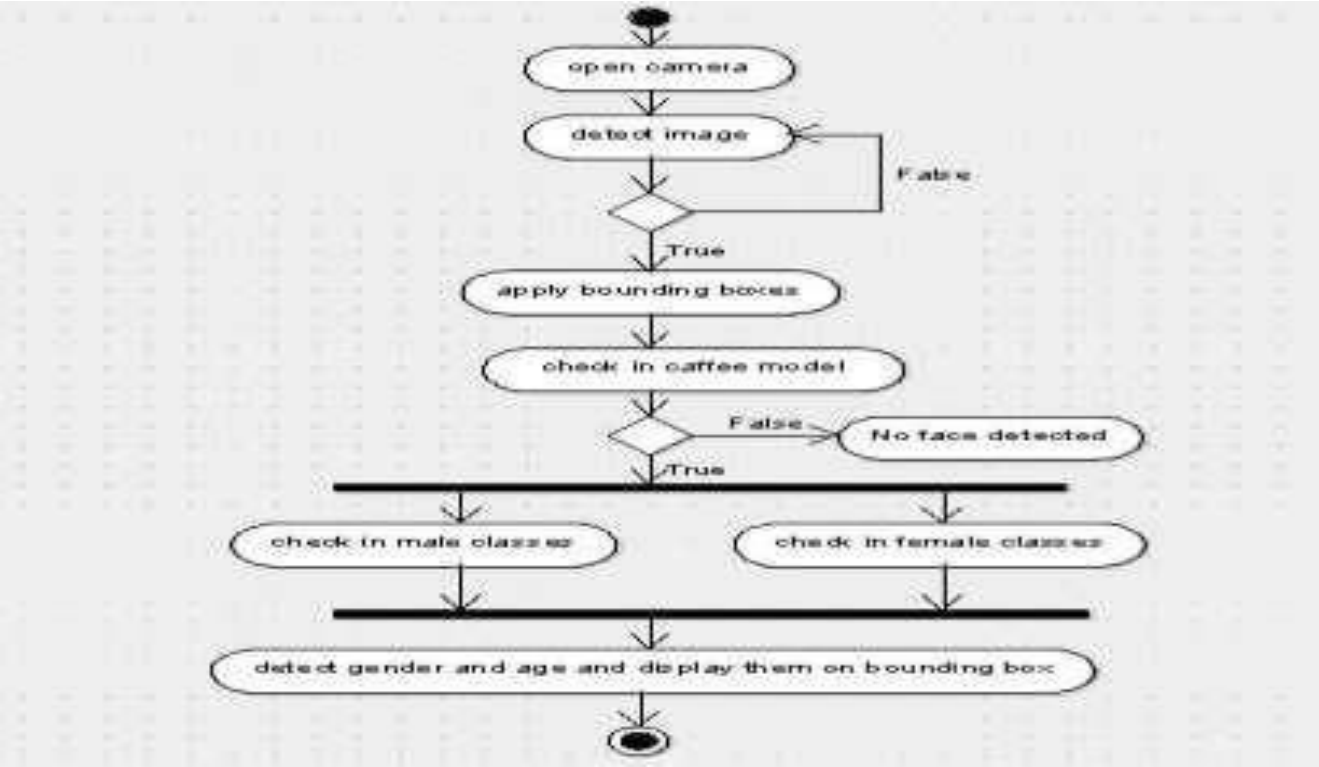
3.4.1 Use Case Diagram:



3.4.2 Sequence Diagram:



3.4.3 Activity Diagram:



3.5 Face Detection Diagram:

Taking the tests that were done into account, we have created the following model for the face detection module, which is shown in Figure 2.

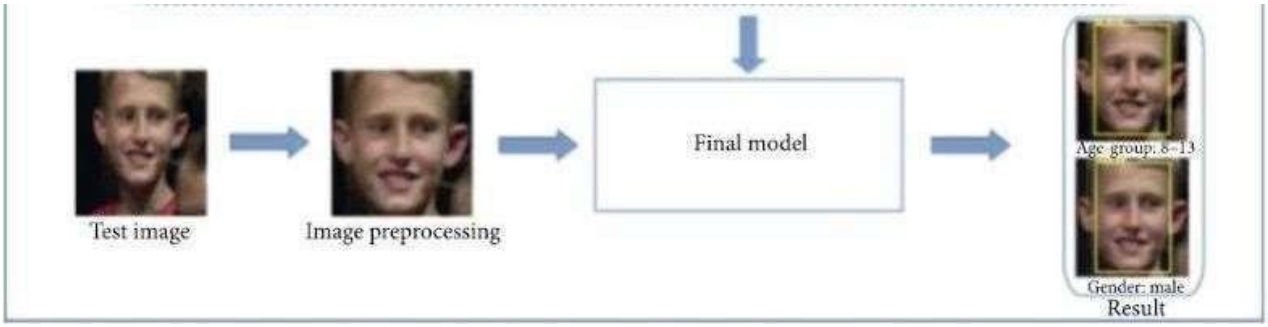


Figure 2- Diagram of the face detection model.

As a first step, the image is preprocessed and resized to specific image size and converted to a specific object (a large binary object (BLOB)) in order to pass it to the DNN. This image size is configurable by the user but has a default value of 256x256 – based on our tests, images with size 227x227 showed to have the best validation accuracy with the tested dataset, having the lowest combination of false positives and false negatives. The confidence with which the network detects the faces is also configurable but has a default threshold value of 0.5. This means that only detections with a probability higher than 0.5 are considered real faces. Finally, after the image is preprocessed, it is passed to the network which outputs all detectable faces with coordinates that indicate 4 points that are used to crop the face, and another output which is one value between 0 and 1 which shows the confidence that such detection is, in fact a face.

3.5.1 Global System Diagram:

The final model that we propose is represented. It was computed based on the system developed in chapter 3 and on the analysis that was done throughout chapters 3 and 4, taking into consideration the results obtained from each experience conducted. It includes the face detection part, which uses

Open CV's Caffe model, which showed high accuracy levels on the parameterization that was used. With this in mind, the confidence and image size are input parameters of the network, as well and the image to process. After preprocessing the images, the initial input is cropped so that only the face flows down to the subsequent module which does the age classification and gender classification. The age classes are configured in a file so that the users can manually edit them without requiring tore-train the network. As a final result, an array is returned with the classifications for each face detected in the input image. If multiple faces are detected, then the array size will be greater, having each array position occupied with the results for each face. The fact that we have a model that is divided into separate modules (the face detection module and the age/gender classification module) allows the global system to be more portable and easier to change. With the constant and rapid evolution on deeplearning networks, new models could emerge which could potentially replace one of the current modules in place. This way, the changes needed are minimal, as opposed to having one model to do everything.

In regards to our model, we need to consider that it will be potentially deployed in an environment where performance is important. From the previous analysis, we concluded that including a Network on uncertain prediction results could potentially increase the over all accuracy further when it comes to gender prediction. On the other hand, during our tests, the addition of this network also increased the execution of our model in an additional 1 second per image. In a real-time environment where thousands of images are fed into the system each time, an increase of 1 second per image is certainly a bottleneck, but for certain cases, if a validation wants to be done on a small number of examples as a one-off verification (opposed to be running constantly in a live environment), this could, in fact, be useful and the execution time is not problematic. Therefore, we included the Caffe Networks for gender classification, which overrides the result from the Network in case its confidence on the output is lower than a certain pre-set threshold. Such threshold is configurable by the user so that it can be adjusted at will. The whole Caffe Network can also be deactivated in the configuration file so that only the Network is used in case that is required. Finally, there is an additional threshold that can be set, where images are stored in a directory for users to manually validate them if required, so that they can validate low confidence results, doing a double check to guarantee that the results are actually correct.

3.6 Face Detection Model:

For face detection, there are a few models available that allow us to use an image to extract the coordinates of each face detected. In this section, we are going to validate the accuracy of three such models to see which one is better suited for our system. Those models are described below:

- The Local Binary Pattern (LBP) cascade implemented by Steven Puttemans et al, referenced as LBP cascade from here on, which is a common implementation used in face detection systems due to its simpler and faster implementation. LBP based systems have performed well in other problems, such as texture classification and image retrieval and speed can be important depending on the usability desired of the final system. This model is tested using OpenCV's Cascade Classifier module.
- The Multi-Task Cascaded Convolutional Networks (MTCNN) model, which is constructed using 3 cascaded CNN's, can detect the coordinates of the faces and also of the eyes, nose and mouth. This model presented state-of-the-art results (in 2016) compared to other models, and therefore, it is a good candidate for our testing. This model was trained by the authors on CASIA-Web Face dataset and VGGFace2 dataset.
- Open Source Computer Vision's (OpenCV) Deep Neural Network module, which was released as part of OpenCV version 3.3 in 2017 and which comes with a GoogLeNet model trained for face detection. This model was selected because OpenCV is a framework heavily used in image processing tasks for a long time. This uses the Caffe framework and is built by more than 200 layers, including convolutional layers, ReLU, normalization layers, and others. This model was trained by the authors on VOC2007 and VOC2012 datasets.

3.7 Age and Gender Prediction Model:

After a face is successfully detected from the previous step, the next goal is to identify the age class and gender class of that person. There are several models publicly available that can satisfy our initial requirements, therefore to avoid replicating something that already exists or that was already modeled, we first investigated for an existing model that could be applied in our context. During such investigation, multiple models were found that had their source code publicly available. Many of thosewere only test projects and tutorials for age/gender classification tasks.

To avoid such projects, we filtered down our research for models that were based on a conference presented paper. This excluded out those test projects and only gave us more robust models. Adding those specifications narrowed down the number of available models considerably. This investigation got us two possible models to test:

- A model created by Tal Hassner. This model was chosen because it was the baseline used for IBM's(International Business Machines) predicting model. It was presented in the Twenty-Seventh International Joint Conference on Artificial Intelligence;
- A model created by Gil Levi. The implementation of such model was inspired by some work done over Wide Residual Networks, which was done by Sergey Zagoruyko and Nikos Komodakis. This was presented in the British Machine Vision Conference. This implementation was trained on Adience Benchmark dataset. It had the training data that was used to train it available, with an example of validation data that could be used.

3.8 Validation:

Multiple frameworks were identified as capable of providing us with the functionalities that we need. Therefore, to be able to decide which one's are the best choices for our problem need to evaluate them and compare them against each other. This comparison is made by validating the accuracy thatis achieved by each framework. Such tests and analysis are described below.The dataset used was thesame across all tests, and the analysis described is based on such dataset only.The use ofother datasets could have different results from the ones stated here, so a real-life implementation can have differentresults as well.

3.9 Face Detection Model:

Several tests were conducted with the objective of validating, which can achieve better accuracy levels, for us to use it in our system. All those experiments were done with the exact same validation set in order to have a more reliable comparison, composed by 8593 samples from the dataset, which is a public dataset available online with a set of images from people from multiple ages and both genders. This dataset contains images that are both aligned and with a side pose. Images

were previously transformed to have a fixed size of 64x64 and used a confidence threshold of 0.5. The confidence threshold indicates the minimum confidence that the model needs to have on a detection for that to be considered a face. The used samples were randomly chosen, after performing ainitial clean-up on the dataset, removing low-quality images(e.g., with strong occlusions or blur).

3.9.1 OpenCV Caffe:

The last network that was tested for face detection was OpenCV's GoogLeNet, which takes around 0.08 seconds to process one image, which is similar to the MTCNN approach. OpenCV is known for having multiple libraries for visual recognition tasks, and this new model is their first using Deep Learning in order to detect faces. The tests conducted followed the same configurations as both previous tests, and results reached 82.4% detection rate, which outperforms the two previous ones. Table 2 shows a summary on the accuracy regarding all three models. This last network has two outputs. It gives us a confidence value (value from 0 to 1) representing the probability of the face being, in fact, a real face, which indicates how confident the network is with its prediction, and a set of coordinates for the extracted face, which can be used to crop the faces so that they can be used by the age and gender model.

Table 2- Comparison between face detection systems.

	LBP	MTCNN	OpenCV Caffe
Accuracy	77%	44.7%	82.4%
Number of undetected faces	1979	4755	1508
Prediction time	0.06s	0.09s	0.08s

3.9.2 Age and Gender Prediction Model:

After a face is successfully detected, the next step is to identify the age and gender classes of theperson. The first model that was tested for this purpose was created by Tal Hassner. Some initial tests were done using the classes that are described in the requirements section, which are the following:

Young Child - Age 0-2;
Child - Age 4-6;
Young Teen - Age 8-12;
Teen - Age 15-20;
Young Adult – Age 25-32;
Adult - Age 38-43.
Young Senior –Age 48-53;
Senior – Age 60-100;

Validation was done using the Adience dataset , the same as our previous validations over face detection. We chose randomly 26,000 images as our validation set and ran those against this model.Results achieved were not as expected, and we got an overall accuracy of 56%.

The second model tested was implemented by Gil Levi, and its architecture was based on Sergey Zagoruyko and Nikos Komodakis work . Same classes as in the previous test were used, which achieved 66.6% accuracy, which is an improvement to the previous model. This model predicts the age as an exact measure, using a softmax classifier that gives a probability for a total of 100 classes, representing ages from 0 to 100. For this, it does a dot product to calculate the exact age of a person.This product is done by multiplying each probability of each class by the age value of such class, summing up all the values obtained – this gives us an exact age estimate for the person, giving moreweight to ages that have a higher probability of being the correct one.

So the initial model gives us a set of probabilities regarding the possibility of the person being inanyof those classes and computes an estimate using those. Instead of numerical age, the use of age- group classes allows statistics to be gathered in a more direct way. In some cases, we do not really need to know the exact ages of our subjects. We just want to know how many of them did something and are in a certain age class. This is the case for examples of supermarket customers. Such age classes allow managers to make better marketing decisions. On top of the initial age calculation layer,we have added a procedure to decode the exact age into the range of classes that we are using, which we then use to do our validations. This additional procedure also allows us to customize the age classes without modifying the underlying network and without needing to retrain it, which gives us a more flexible application since the classes can be customized by the user whenever needed.

3.9.3 Softmax Classifier:

In order to increase the performance of the age classification further, we need to do additional investigations on the network we have. Can we create a mechanism that allows the users to do a second check on some of the predictions and validate those manually? How could we specify which predictions should be manually validated?

Taking into consideration that the formula used to calculate the age of a person uses a softmax classifier to calculate the probability of the person to have a specific age, then we could use that to know the confidence that the prediction has for a person to be in that specified class. Atypical heuristic to estimate the confidence on a softmax-based classifier's prediction is to simply take the probability of the class with the highest probability as the estimated confidence. This is useful to know the confidence that the prediction has for a person to be in the predicted age class. If we only consider valid predictions like the ones where the confidence reaches high values, can we guarantee that these correspond most often to correctly classified data samples? To answer this question, we are going to assume that high confidence values are values greater than 0.6.

Can data samples that produce lower confidence values be wrong most of the times? If this is, in fact, true, then we could use all the other examples where the probability is smaller than 60%and introduce a manual step to validate the examples and, therefore, increase the accuracy of the statistic that is generated by the application.

If we imagine that we have a supermarket that uses this kind of models to gathers statistics about who their customers are, having accurate statistics is relevant. If we know that when examples are tagged as had less than 60% confidence in the predicted result, then we can put those examples through an additional step which needs to be validated by the user. With this, we can guarantee that the statistics that the supermarket gathers on a daily basis are more accurate, which allows managers to do marketing campaigns based on this data with more confidence. Another test was conducted to investigate this.

The network probabilities are initially computed for each of the 100 possible ages, i.e., considering 100 age classes. To compute the probabilities for the five age-group classes, the age-wise probabilities were grouped into age-group classes. So, in order to obtain the probability of agent-group class 8-12, we sum up all single probabilities from age class 8 to age class 12. Our tests showed that for some cases, usually the ones that lie near the edge of another class, the softmax classifier has

a distribution similar between the two adjacent classes. As an example, we can analyze row 3 and row 4 from Table 3. The person represented in row 3 has an actual age of 27 and the probability around classes 15-20 and 25-32 present similar values (i.e., 47% against 49%,respectively). The same applies to row 4, which represents a person whose actual age is 62 years,resulting in a similar probability for classes 48-53 and 60-100 (i.e., 50% and 49.5%, respectively).In these specific cases, it is clearly difficult to predict to which of the classes the person actually belongs. Ideally, data samples would produce results similar to the ones obtained for the one represented in the first row, for which the prediction points to class 48-53 with a confidence value significantly higher than the ones obtained for the alternative age-group classes. This case has strong confidence in the predicted class and is, in fact, correct since the person has a real age of 59 years.

But before doing any conclusions, let us analyze the whole validation data and see how the accuracy performs on all the samples, and let us see if in fact cases where the probability is higher,are most of the times correct or not.

Table 3- Accuracy using different probability ranges (p stands for probability)

	Success	Fail
Age (p <60%)	1180 (39%)	1822 (61%)
Age (p >=60%)	2562 (71%)	1042 (29%)
Gender (0.25 < p < 0.75)	227 (63%)	133(37%)
Gender (p <= 0.25 OR p >= 0.75)	5987 (96%)	259 (4%)

From the results present in Table 3, we can see that there are a lot of examples that fail with a probability lower than 60% on the predicted class. But on the other hand, when probabilities are higher than 60%, we can also verify a lot of failed examples (1042). These numbers seem to be considerably high in order to proceed with a manual validation on them all - validating manually more than 1000 examples is very time-consuming. There are also several correct examples that are predicted with less than 60% probability. On top of that, even validating ages manually is not completely accurate – even for humans, some cases might be hard to predict due to diverse conditions, as light,

make up, and angles. Therefore, there is a need to find an additional automatic mechanism to increase the confidence with which these examples are classified.

When it comes to gender classification, we have a higher overall accuracy on the tested network. Also, manual validation is simpler for humans since there are only two classes available (i.e., male and female genders). For gender, the prediction is made differently from the one for age classification.The output of the model is a value that ranges from 0 to1,where values are greater than 0.5 are considered as a male classification, and values smaller than 0.5 are classified as a woman. So the network does provide a single probability as output, rather than multiple ones. If we consider probabilities near 0.5 as low confident, we may expect that most misclassified samples to be classified with probabilities near that reference value. Tests were done based on this assumption by checking how many samples are misclassified with a probability between 0.25 and 0.75, how many with a probability below 0.25 and how many with a probability above 0.75. This is expected to give us the amount of misclassified samples for which the network is not confident with its prediction. Results for this are shown in Table 3.

The results show that the probability of the network failing the predictions is higher when the probability of the prediction has confidence near 0.5. Although we still have more failed examples on high confidence inputs, proportionally to the total samples, they are a smaller proportion.

3.9.4 Increase the accuracy of low confidence outputs:

Followed by previous analysis, the network would benefit from having another mechanism to increase the correctness of a prediction when our base model has a confidence smaller than 60%,when it comes to age classification, or confidence near the reference value of 0.5, for gender classification. Instead of applying a manual validation, ideally, we would have an automated mechanism for it.

Caffe networks using training could be useful to compare a face against a database of pre-determined faces, and based on similarity levels, predict the person's class. If we use children's faces, it is more likely that it will be more similar to another child's face than if we compare it against an adult. This is because there are facial landmarks that clearly differentiate a child from an adult. If this hypothesis is true, then there would be no need for a manual validation system because this mechanism could increase the accuracy of the network further.

To test this hypothesis, we used OpenCV's framework , which has a comparison module based on Caffe Networks to compare people's faces. For this, both faces pass through the same model, and this model outputs a squared L2 distance (Euclidean distance) between both images. This means that lower output scores represent more similarity between the inputs.

The faces used were the ones with lower confidence levels extracted from the dataset that was tested previously with the Network. The training dataset was composed of pictures that were correctly predicted with greater confidence (95%) by the system because those showed to have a high classification rate and are therefore more trustworthy. In total, this was composed of a total of 3002 test examples and around 30 validation images for each of the used classes. Using a smaller training dataset allows the network to do computations faster since it is executed fewer times (once for each of the dataset images).

This can be viewed where it is shown how the training data was created and how it feeds the Caffe Network. The training set represents a folder where it has 2 distinct additional folders, one where the images for men are stored, and another folder for women. Those pre-added images can also be removed, and the user can include new samples manually.

On a first experiment, the faces were compared against the validation dataset, and the prediction considered as correct was the one that was the most similar considering all the classes. So a man with 25 years would be compared against all faces from all classes, and if the most similar face would be from a face present in class 25-32, then that class would be the output prediction.

Table 4- Accuracy over ranking similarity for under-confident examples

	Success	Fail
Age Minimum	1236 (41%)	1766 (59%)
Age Average minimum	362 (12%)	2640 (88%)
Gender	271 (75%)	89 (25%)

This gave us the results present in Table 4, under the "Age minimum" test. The accuracy was similar to the one obtained using only the age classification model, so there was no improvement in including this additional network. Samples from such an experiment are shown in Figure 3 and Figure 4, where the first one shows correctly predicted data, and the second shows the failed examples. The first face in each case was the one that was tested.

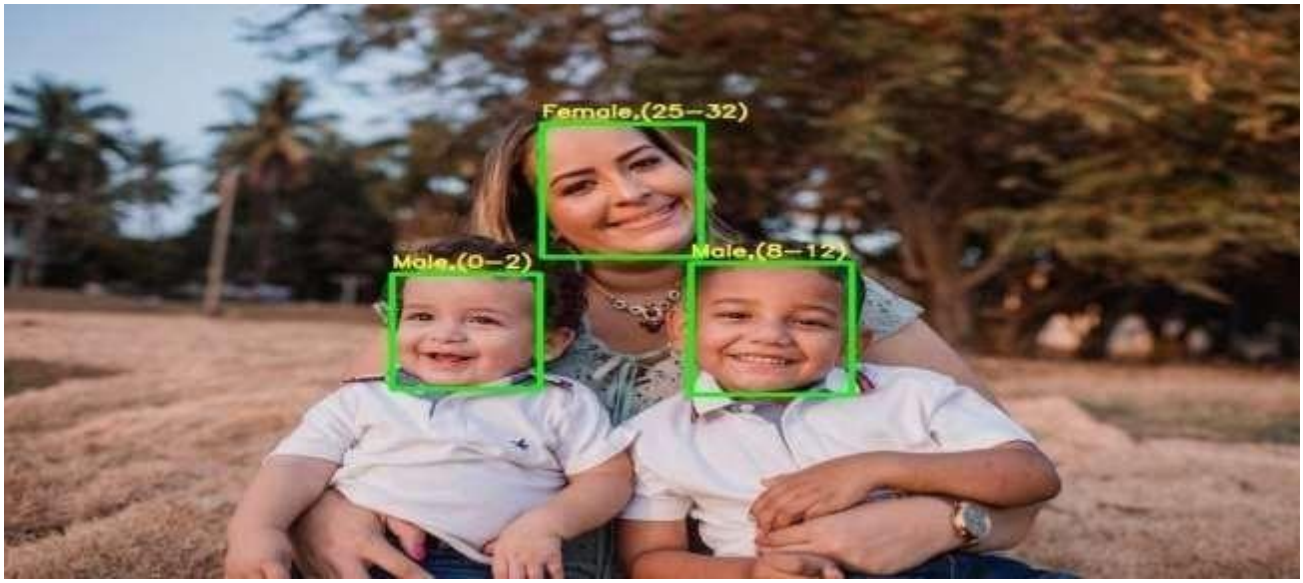


Figure 3- Classes (0-2),(8-12) and (25-32) are mentioned in the diagram

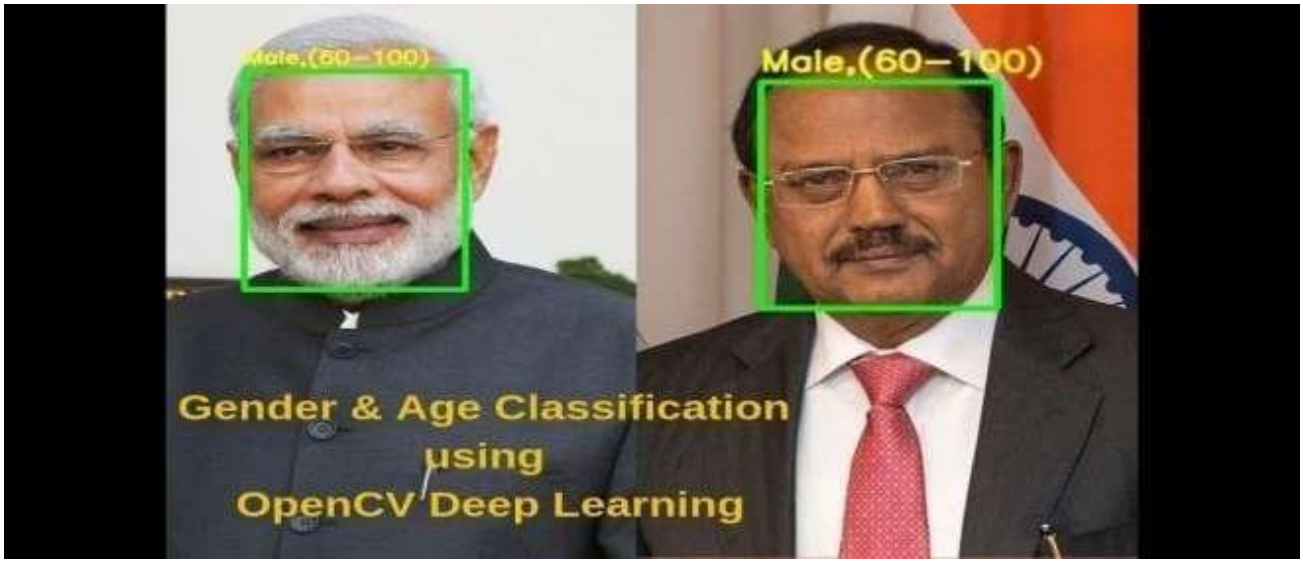


Figure 4- class (60-100)

Another validation that we considered relevant to do, was instead of using the most similar face to decide which class the input would fall in, was to do an average similarity by each class. So, we expected that a person with 60 years would be more similar to the people in such a class. This would make the average similarity from such class lower than compared to other age-groups.The results from this are present under the "Age Average minimum" test. As shown, those results are lower than the previous test, achieving only 12% accuracy levels. This low result is justified by the fact that, although people in the same age class should show similarities on some age-related characteristics, there are also people that are very different to one another, and these cases push the average similarity down. Also, when one person is compared against another class, sometimes there are people that present very similar characteristics in such class, and therefore, push the average similarity up, especially if the classes are next to each other and the age difference is not very high. This can be seen in Figure 4, where some samples were more similar to people from other classes instead of people from the same class. This concludes that it is not beneficial to trade the age classification model for the Caffe network in cases where the age classification model is not confident enough on its own prediction.

Before closing this experiment, we need to confirm also if it is worth including those networks for gender as well. For this, our testing samples are the examples where the confidence level for the gender was between 0.25 and 0.75. A similar test to the minimum age test was done where we tried to find the most similar person to be the driver for the classification. This attained an accuracy level of 75%, which is slightly better to what the gender classification model achieved for lower confidence examples (63%). It seems like gender can, in fact, be increased with the use of Siamese Networks, although it is only an increase of 12% and over a small part of the entire data, it still increases the overall reliability of the data that is gathered. In Figure 5 and Figure 6,we can see examples from the correct and incorrect predictions from such test. Similar to the previous test, faces from the left were used as test data and faces on the right were part of the validation dataset.

Figure 5- Gender correctly predicted data. Left side image is the input image, while the right side image is the validation image to which the input was matched.

Figure 6- Gender failed prediction data.
Left side image is the input image, while the right side image is the validation image to which the input was matched.

To note that from the samples that were incorrectly predicted, there are a few which are related to wrongly annotated data from the initial dataset. Those are shown in Figure 7, which contains two examples where two pictures of men were annotated as being women. Those cases might be influencing the prediction rate, as the model correctly matched them with a picture of men but since they are wrongly annotated, they were included as failed tests.

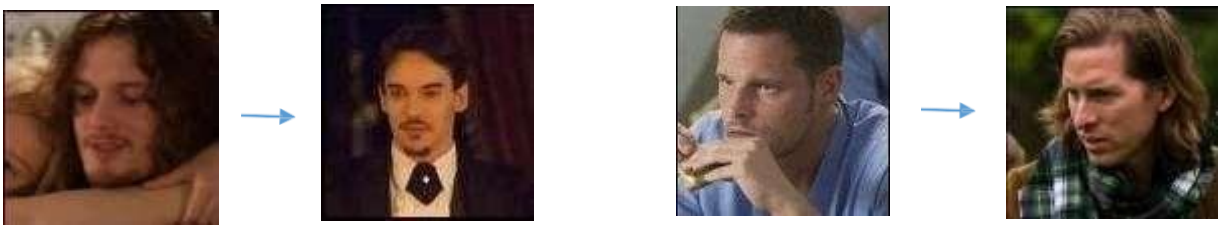


Figure 7- Gender failed prediction data
. The left side images were wrongly annotated in the initial dataset and are the inputs,while the right side image is the validation image to which the input matched.

TESTING AND EVALUATION

CHAPTER-4

TESTING AND EVALUATION

In the previous chapter, multiple frameworks were identified and tested, and the development of the system was described. Tests over such system are described in this chapter, with some evaluations that were done in order to find improvement points.

4.1 Training and detection:

Training a CNN using Caffe framework

Step 1 - Data preparation: store images in a format that can be used by Caffe. We will write a Python script that will handle both image pre-processing and storage.

Step 2 - Model definition: In this step, we choose a CNN architecture and we define its parameters in a configuration file with extension .prototxt.

Step 3 - Solver definition: The solver is responsible for model optimization. We define the solver parameters in a configuration file with extension .prototxt.

Step 4 - Model training: We train the model by executing one Caffe command from the terminal. After training the model, we will get the trained model in a file with extension .caffemodel.Testing for both age or gender classification is performed standard five-fold cross-validation protocol.

Split the images into 5 folds and then perform a subject-exclusive cross-validation protocol. The reason this type of protocol is necessary is because of the nature of the dataset being used, which contains multiple pictures of the same subjects. Therefore if the images were simply randomly shuffled and divided into fifths, the same subjects could potentially appear in both the training and test folds.

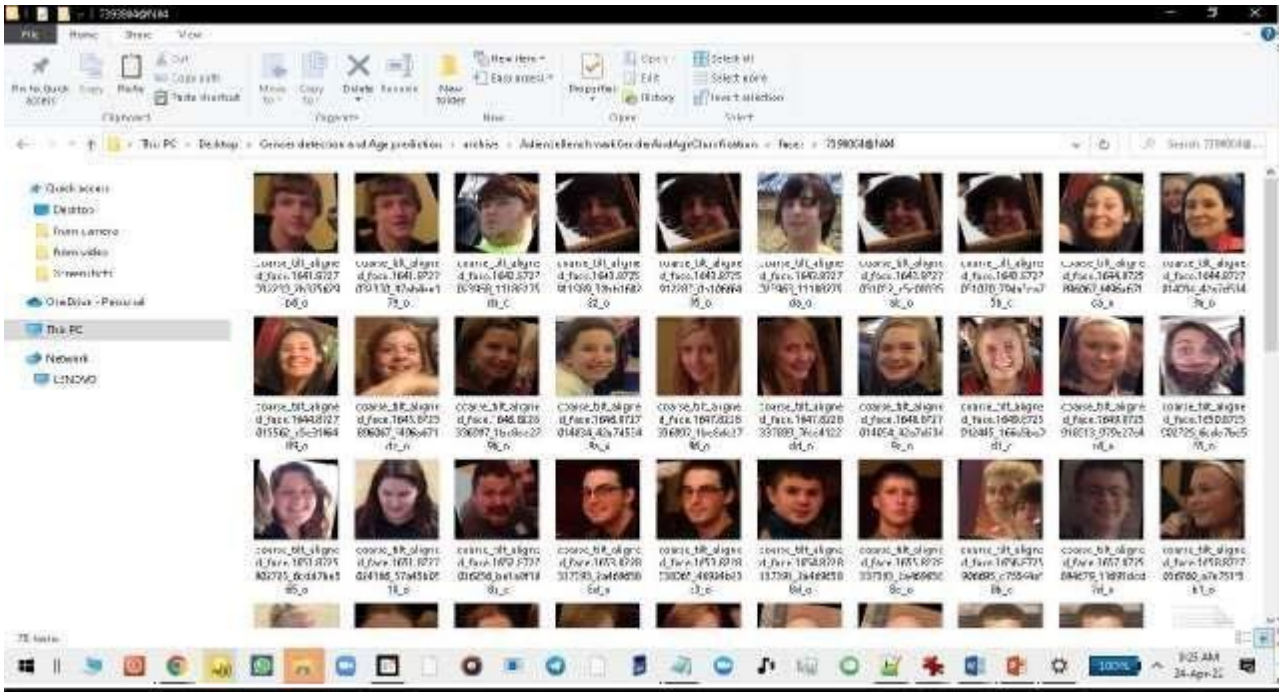
4.2 Benchmark Adience Dataset:

We test the accuracy of our CNN design using the recently released Adience benchmark, designed for age and gender classification. The Adience set consists of images automatically uploaded to Flickr from smart-phone devices. Because these images were uploaded without prior manual filtering, as is typically the case on media webpages (e.g., images from the LFW collection) or socialwebsites (the Group Photos set), viewing conditions in these images are highly unconstrained, reflecting many of the real-world challenges of faces appearing in Internet images. Adience images therefore capture extreme variations in head pose, lightning conditions quality, and more. The entire Adience collection includes roughly 26K images of 2,284 subjects. Table 1 lists the breakdown of the collection into the different age categories. Testing for both age or gender classification is performedusing a standard five-fold, subject-exclusive cross-validation protocol, defined in. We use the in-planealigned version of the faces, originally used in. These images are used rater than newer alignment techniques in order to highlight the performance gain attributed to the network architecture, rather than better preprocessing. We emphasize that the same network architecture is used for all test foldsof the benchmark and in fact, for both gender and age classification tasks. This is performed in order to ensure the validity of our results across folds, but also to demonstrate the generality of the networkdesign proposed here; the same architecture performs well across different related problems. We compare previously reported results to the results computed by our network. Our results include bothmethods for testing: center-crop and over-sampling.

	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-	Total
Male	745	928	934	734	2308	1294	392	442	8192
Female	682	1234	1360	919	2589	1056	433	427	9411
Both	1427	2162	2294	1653	4897	2350	825	869	19487

Table 1. **The AdienceFaces benchmark.** Breakdown of the AdienceFaces benchmark into the different Age and Gender classes.

G. Levi and T. Hassner 3



4.3 Network Architecture:

The network comprises of only three convolutional layers and two fully-connected layers with asmall number of neurons. Images are first rescaled to 256 X 256 and a crop of 227 X 227 is fed to thenetwork. Each convolutional layer, followed by a rectified linear operator (ReLU), a max pooling layer taking the maximal value of 3 X 3 regions with two-pixel strides and a local response normalization layer. Last fully connected layer maps to the final classes for age or gender. Our proposed network architecture is used throughout our experiments for both age and gender classification. It is illustrated in Figure. A more detailed, schematic diagram of the entire network design is additionally provided in Figure. The network comprises of only three convolutional layers and two fully-connected layers with a small number of neurons. This, by comparison to the much larger architectures applied, for example, in and. Our choice of a smaller network design is motivated both from our desire to reduce the risk of overfitting as well as the nature of the problems we are attempting to solve: age classification on the Adience set requires distinguishing between eight classes; gender only two. This, compared to, e.g., the ten thousand identity classes used to train the network used for face recognition in. All three color channels are processed directly by the network. Images are first rescaled to 256 × 256 and a crop of 227 × 227 is fed to the network. The three subsequent convolutional layers are then defined as follows.

1.96 filters of size 3×7×7 pixels are applied to the input in the first convolutional layer, followed by a rectified linear operator (ReLU), a max pooling layer taking the maximal value of 3 × 3 regions with two-pixel strides and a local response normalization layer.

2. The 96 × 28 × 28 output of the previous layer is then processed by the second convolutional layer, containing 256 filters of size 96 × 5 × 5 pixels. Again, this is followed by ReLU, a max pooling layer and a local response normalization layer with the same hyper parameters as before.

3. Finally, the third and last convolutional layer operates on the 256 × 14 × 14 blob by applying a set of 384 filters of size 256 × 3 × 3 pixels, followed by ReLU and a max pooling layer.

The following fully connected layers are then defined by:

4. A first fully connected layer that receives the output of the third convolutional layer and contains 512 neurons, followed by a ReLU and a dropout layer.

5. A second fully connected layer that receives the 512dimensional output of the first fully connected layer and again contains 512 neurons, followed by a ReLU and a dropout layer.

6. A third, fully connected layer which maps to the final classes for age or gender.

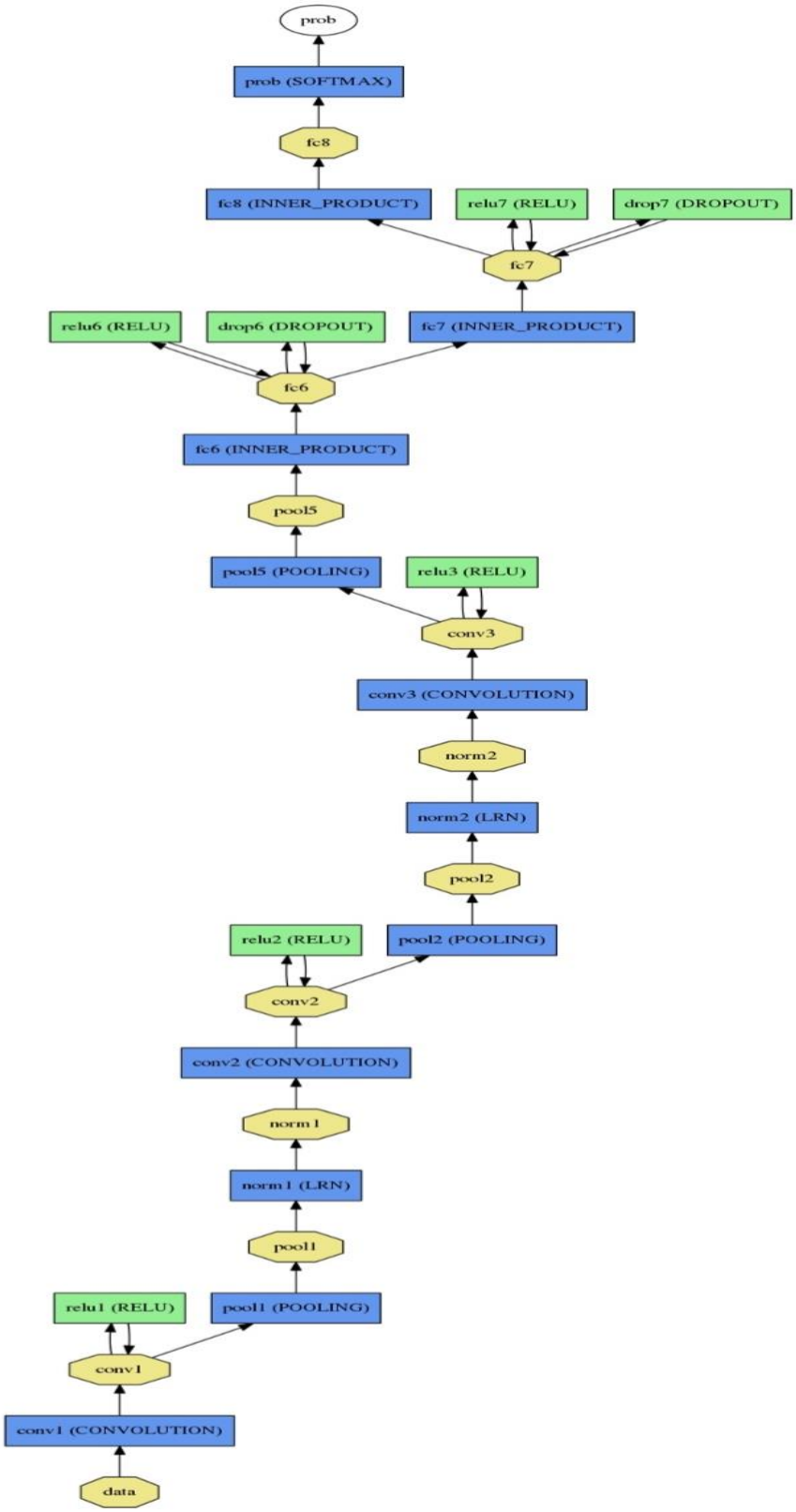
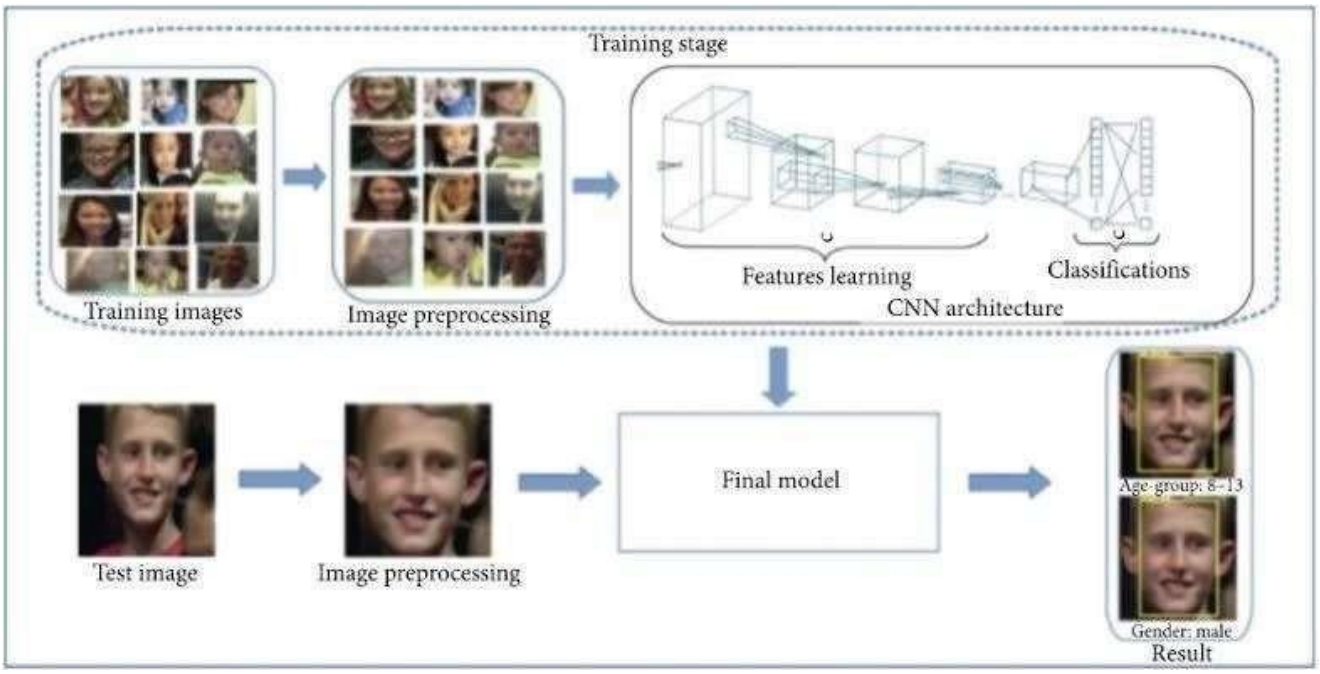


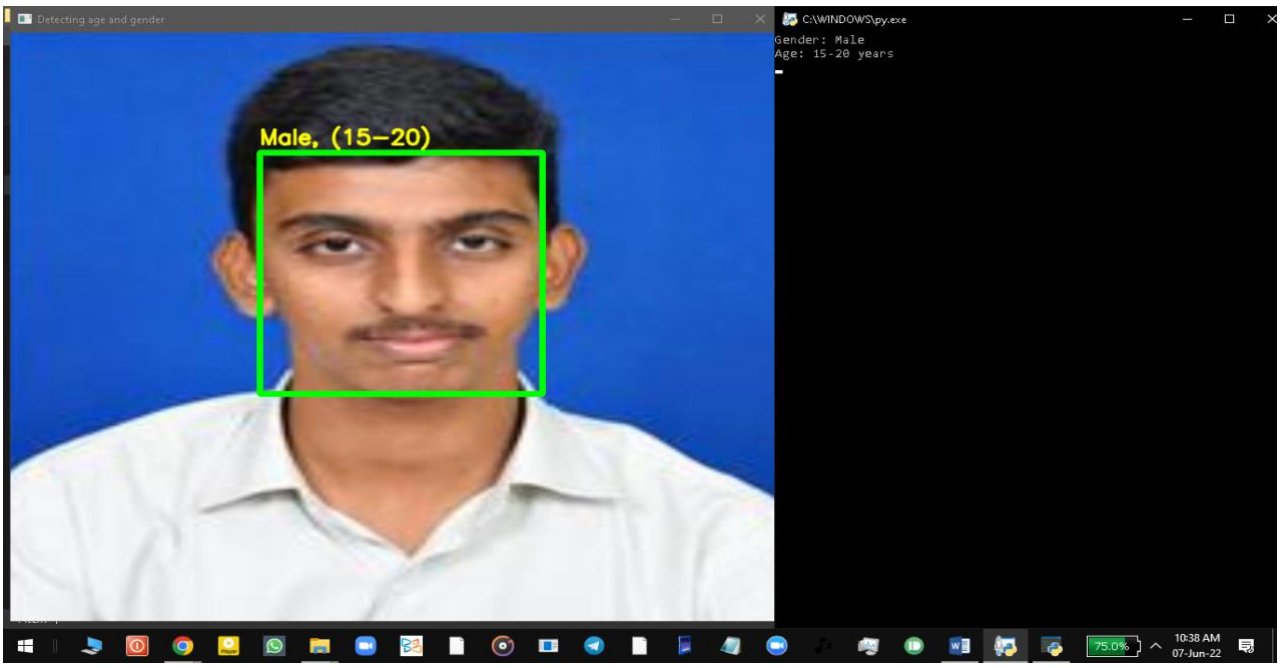
Figure- Full schematic diagram of our network architecture.

4.4 Working of the project:

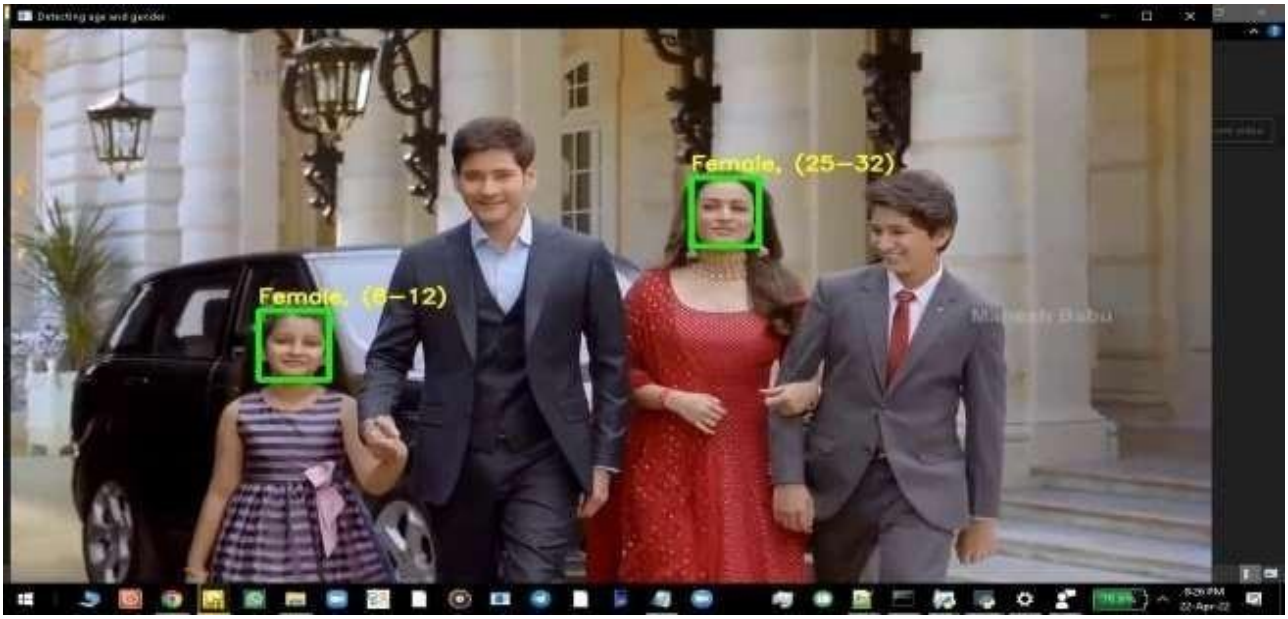


4.4.1 Input from camera:

4.4.2 Input from image:



4.4.3 Input from video:



4.5 Face Detection:

The results achieved with OpenCV and Caffe is the best option comparing the three networks that were validated, but is there any way to increase its accuracy even further? Taking into consideration that fixed values for both image size and confidence threshold were used, could they play an important role in the output accuracy? When it comes to image processing, when we reduce the size of an image, there is information that is lost with it, and therefore, fewer details are processed, which can impact the output result. This led us to do some experiments around the use of different image sizes and confidence thresholds in the following section.

4.5.1 Image Size Impact:

The previous tests that were conducted in Section 3.5.1 were done with 64x64 images, so additional tests were done with sizes 100x100, 200x200, 250x250, 300x300, and 320x320. Results in Figure 8 show that the image size has an important impact on accuracy. For this, measurements on both false negatives (undetected faces) and false positives (detections that were not real faces)were done, analyzing the relation between them and the image sizes. We concluded that both false positives and false negatives are inversely related to the image size. Thus, when the image size is increased, we can see an improvement on detecting faces that were undetected before, but at the same time, we also see an increase in detections that are not faces at all – the false positives. Hence, if a small image size is used – like 64x64 – a minimal false positive rate is obtained, but at the same time, a very high false-negative rate is achieved (17.5% of the data samples). With higher image sizes – like 320x320 – the false negative rate drops down to 0.58% and the false positive rate increases to 81.1%. If we compare results from the graph in Figure 8, we can see that the increase/decrease is slighter when values are low, and it starts to grow faster when false positives/negatives start getting higher. For example, comparing results from images with sizes of64x64 and 100x100, we can see that the number of false positives increased by 19 samples, which is not a significant increase. But the number of false negatives decreased by almost 50%, from 1508 to 725. After a certain point, we are able to state the opposite. Comparing results from images that are 250x250 and 300x300, the false negatives have a very slight variation of 2 samples (from0.6% to 0.62%), while false positives have an increase of more than 55%. This behavior can be explained due to the same principle that affects high-resolution images, which have the tendency to present more false positive detections because of the extra details in the image.

In order to have the best accuracy in our model, we need to find a balance in which the false positives and false negatives are at the lowest possible, which is in fact achieved using image sizes of 200x200. On the tests that were done using 200x200 sizes, we attained 78 false negatives and 96 false positives on a total sample of 8593 images which gave us a face detection accuracy of 98% using the Adience benchmark dataset.

Figure 8- False positives/negatives rate by image size.

4.5.2 Confidence Threshold Impact:

The face detection system has a confidence output result (a value that varies from 0 to 1), which indicates how confident the network is with its prediction about that specific location having a face. So the higher this prediction confidence, the higher the probability of the image having a person's face in the image.

Tests conducted in Sections 3.5.1 and 4.1 were done with the confidence parameter with a value of 0.5 (default value) – this indicates that the network only considers a prediction as a real face if the confidence level of that output reaches the pre-defined value of 0.5. If the confidence threshold is increased, maybe we could increase the image size as well, and at the same time reduce the false positive and false negative rates that are introduced with the change in image size. Looking at some of the examples of false positives, they are commonly on the lower right side of the image, and in a lot of cases, the confidence on this false positives is also very high as shown in Figure 9, where the false positive has a confidence value of 99.59%.

Of the examples of false positives, they are commonly on the lower right side of the image, and in a lot of cases, the confidence on this false positives is also very high as shown in Figure 9, where thefalse positive has a confidence value of 99.59%.

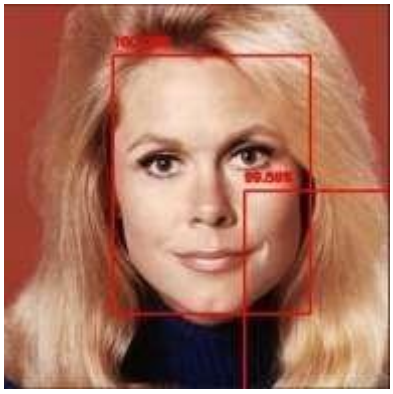


Figure 9- Sample of a false positive.

In fact, increasing the confidence threshold of the network to 0.9, with an image size of 300x300, reduces the false positive rate by 55%, reaching a total of approximately 2000 failing samples which is a clear improvement. But we still have a lot of failing examples, so increasing the image size further together with the confidence threshold parameter turns out not to be beneficial. A better overall accuracy can be obtained by choosing a different image size, in this case, 200x200. Table 5 shows a comparison of results obtained from changing the confidence threshold.

Table 5- Comparison using different confidences.

	Test 1	**Test 2**
Accuracy	40%	74%
False negatives	54	54
False positives	4956	2236
Image size	300x300	300x300
Confidence	0.5	0.9

4.6 Age and Gender Prediction Model:

4.6.1 Gender Validation:

When it comes to gender classification, there are only two possible outcomes. The confidence value that the model gives to the input image ranges from 0 to 1, where values near 1 mean that the image

depicts a male, and values near 0 that it depicts a female. Therefore, 0.5 is the gender classification threshold.

Using the same test dataset, an accuracy of approximately 95% was obtained for gender classification, which is very satisfactory for practical purposes. This is split by 94.7% accuracy for men and 92.7% for female. From this, we can conclude that age classification is more challenging than gender classification, which may require additional validation steps to improve its performance.

4.6.2 Age Validation:

Given the advantages of employing age-group classes, rather than the actual numerical age,we adapted the selected model to output an age class. Concretely, an output procedure was added to decode the exact age that the original model outputs into the aforementioned range of age-group classes. With this configuration, the model reached an accuracy of 66.6%. Table 5 shows the resulting confusion matrix. This matrix was obtained by distributing the predictions that the model made for each of the classes. As an example, if we take class 15-20, all the samples with this class were gathered from the validation dataset, and for each of the image predictions, we computed the percentage of examples that were predicted with each class. Thus, this tells us the amount of data that was predicted with class 15-20 (correctly) and the amount that was predicted with each of the other classes (incorrectly). So for samples in class 15-20, we can consider that 25.5% of the predictions were erroneously classified as class 25-32.

Due to the nature of the Adience Benchmark dataset, the distribution of the sample is mostly concentrated in the 15-20, 25-32, 48-53, and 60-100 classes, with the remaining two classes representing only 1% of the total samples. The lack of data for those age-group classes may impact the accuracy obtained for them, as the little amount of data that is available can be of examples that are not clear or hard to predict, impacting the accuracy of such classes considerably. If we look at class 8-12, we have only a total of 20 samples, since all of them failed, we ended up with 0% accuracy on this specific class. This is not too troublesome due to the fact that this class might be the least relevant of them all in our specific context, as small children are unlikely to frequent a supermarket by themselves and therefore might not be statistically relevant. For class 15-20, we only got an accuracy of 31.3%, having the highest accuracy around class 25-32. This could be explained due to the small class that this represents (only 5 ages) so it could make it harder for the algorithm to predict the class correctly for people with such ages.

4.6.3 Samples in Between Classes:

Nevertheless, it is also valuable to assess how much low accuracy is influenced by data samples laying on edge between two different classes and, therefore, harder to predict. Could the low accuracy of some classes (we can take the 60-100 class, for example) be explained with examples that are on the edge, between 2 different classes? For example, the assumption is that people that have an age of 25 could easily be mistaken for a 24-year-old and, therefore, fall in an adjacent class.To analyze this influence, we computed the average age, per age-group class, across the set of misclassified data samples, using for that the initial dataset (Adience Benchmark) annotations which had the year when the photo was taken, and the birth year of the person, which allowed us to compute the real age of the person at the time that the picture was taken.we can see the average ages of the failed samples from the previous test. Symbol (+) indicates that the average is from data samples that were misclassified as belonging to the adjacent older class (e.g., 8-12 data samples misclassified as 15-20). Conversely symbol (-) indicates that the average is from data samples that were misclassified as belonging to the adjacent younger class.

The results show that age classes 8-12 and 15-20 present a median age located in the middle of the class' age span. This could indicate that our examples are not near the edge between classes,which might seem like the assumption that near the edge ages would be misclassified could not be totally true in this case. But the lack of examples on those classes could be giving us an imprecise impression. A few examples that are wrongly classified could be pushing the average age considerably down, while several others are in fact in the edge of the class. Looking through the examples that failed, some are in fact wrongly annotated images from the initial dataset while others are on the edge of the class as initially proposed, also having a few examples that are not near the boundaries. Class 15-20 presents distances of 22.5% (lower boundary) and 37.5% (upper boundary) when comparing to the class age span. Those suggest that the examples are not near the edge cases, although we need to consider that there is wrongly annotated data (as seen for previous classes) that could be influencing the results. We should also note that although the distance seems high, the actual age distance is only two/three years apart, which is not much and could still justify the examples to be mistaken as one of the boundary classes.The same conclusion applies for class25-32, which has a distance of 29% and 33.3% to eachof the boundaries. Regarding class 48-53, this one has the most discrepant results, with a distance of 10% and 46.6%. Considering this, we can conclude that this class lower boundary results are in fact

near the edge, but the same conclusion cannot be obtained for the upper boundary, which has an average year of 49.4. The last class (60-100) presents an average age near the lower boundary, with a distance of 13.4%.

Older classes seem more likely than younger classes to be misclassified due to the fact that the examples have an age near one of the boundaries. But by analyzing the results, although we can see that some of the classes appear to match this assumption, this conclusion cannot be applied across all of them. Looking at the examples that failed, some of them are samples that are wrongly annotated on the initial dataset, which increases the estimated failure rate of the model (same across all classes). On the other hand, makeup also has an important role in the predictions. Any kind of makeup helps cover face lines that are used by the model to calculate the age – if those are covered the person seems younger, and therefore the algorithm can't make an accurate guess – the same applies when it comes to human predictions, where sometimes a person with makeup seems considerably younger than without it. The examples in demonstrate those 2 cases.The annotated first age is the predicted age by the algorithm and the second is the real age as annotated in the initial dataset. The first image is wrongly annotated, and the other two are near-the-edge examples. In the first two images are wrongly annotated, and the third one is to give an idea of the role that makeup plays on predictions.

CONCLUSION

CHAPTER-5

CONCLUSION

CNN can be used to provide improved age and gender classification results, even considering the much smaller size of contemporary unconstrained image sets labeled for age and gender.

The simplicity of the model implies that more elaborate systems using more training data may well be capable of substantially improving results beyond these results.

One can also try to use a regression model instead of classification for Age Prediction if enough data is available.

Artificial Intelligence systems have grown rapidly over the last years. This enabled us to create, using multiple models and frameworks, a system capable of detecting faces and classifying them by age and gender. The main objective of this research work was to create an efficient system that was able to detect faces in images and to classify such faces into age and gender, and to evaluate wrong outputs in order to find an underlying reason for such failures. In order to fulfill such objective, various frameworks capable of detecting faces in images and capable of classifying those into age and gender classes were tested and validated in order to understand which would fit better into our problem.

Due to the number of parameters that such models have, multiple parameterizations were tested in order to understand how the accuracy could be influenced, and an investigation was conducted on the examples that were failing in order to find common characteristics between them. Such parameters involve the image size from the images that were sourced to the face detection model, the confidence threshold, which indicates the confidence that the model has on a prediction,and the range and amount of classes used by the age and gender classification model. During this analysis, some improvement points were identified. One of such improvements included the use of two different models for gender classification, one using Networks and another using Caffe Networks. To the best of our knowledge, this implementation is the first one available that uses those two models together in the same system for gender classification.

:

A few goals were identified as crucial for us to be able to deliver the proposed system should detect faces and classify them into age and gender classes; the age classes should be configurable; an evaluation on existing models should have been done in order to identify which ones to include in the system; and an evaluation on the failed results should have been conducted in order to identify common failing causes, so that we could adapt the system to increase its accuracy. Those goals were tackled throughout Chapter 3 and 4.

In Chapter 3, a macro architecture of the model was described, evidencing how the components would interact and what their output would be. This chapter also includes the list of requirements and the development work of the system, which includes a comparison between several frameworks that have part of the functionality we desired. For the first part, the face detection model, an initial comparison between three available models was made in order to understand which one should be included in our global system. The CNN was the poorest one in terms of accuracy, achieving only 44.7% with the validation dataset that was used (from Adience Benchmark Dataset),followed by the simpler LBP cascade which attained 77% accuracy. Lastly, the chosen model was OpenCV's Caffe, which outperformed all others with 82.4% accuracy. The second part is the age/gender classification model. Similar to face detection, two different models were tested. The first model, created by Tal Hassner , only attained 56% on age accuracy, being immediately out performed by a model created based on Sergey Zagoruyko and Nikos Komodakis work with an accuracy of 66.6%. On the other hand, gender accuracy achieved 95%, which is very satisfactory for our purposes.

Validation of the model was described throughout Chapter 4, where more tests were conducted in order to identify possible improvement points. Experiments around the image size of the images that are fed to the system and around the confidence threshold used showed positive results in increasing the accuracy further when it comes to face detection. A final accuracy of 98% was obtained once images were pre-processed to be of size 227x227, which showed to be the point where false positives and false negatives were the lowest. This confirms that adjusting the image size for the problem at hands can increase its accuracy as well as improving the amount of data that is passed on to the age gender model.

For the age and gender classification, we have validated that such systems perform poorly when their confidence threshold to accept a prediction as valid is low. From our analysis, we can conclude that if we take examples with a low confidence level, they have a higher probability of

being incorrect. This can be used to filter out examples that need another kind of validation to guarantee that they are indeed correct. Adding a human-in-the-loop validation system on the filtered samples could increase the overall accuracy and provide more reliable statistics to be used. As an alternative validation system, we have tested out the use of Caffe Networks in order to find similarities in the low confidence examples, so that those could be compared against a pre-determined set of annotated samples, to find which class the input would be more similar to. This turned out to be imprecise for age classification, and therefore not a good approach. On the other hand, for gender classification, we have observed an increase of 12% on the accuracy rate from low confidence samplescompared to the one obtained by the Network. Finally, we have also tried to find a relation between the examples that failed and the fact that their age could lie in between two adjacent classes. The data that was analyzed seems to confirm this hypothesis, more significantly for older classes. However, the fact that makeup affects the classification of the model and the existence of wrongly annotated samples in the dataset made it harder to confirm the actual impact that wrongly predicting data near the boundaries of the classes have in the overall accuracy of the system.

Finally, the proposed system was created, and we achieved good results on age and gender classification when compared to existing state-of-the-art models, while for face detection, results exceeded the state-of-the-art models that were analyzed.

5.1 Future Scope:

The dataset used in our tests has some limitations when it comes to wrongly annotated images and lack of diverse angles. The angles are mostly aligned in the current dataset, as opposed to what would be obtained in a supermarket environment, where the camera would be viewing the customers from a top-down view. The final model should be further tested using a real environment scenario to see how it can adapt to using real data, which was not available at the time this work was conducted. Another experiment that could be conducted for future implementations is the use of age interpolation, using the central age of a class, to calculate the age of the inputs. The central age of a class would be multiplied by the probability of a person being in such class, instead of the current approach which does this multiplication for each age value in a class. So, if we had class A as 15-20 with a total probability of 30% and class 25-32 with a probability of 70%, the output age would be calculated using the central ages of each class (in this case, 20 and 33 respectively) against the

probability of the person being in fact in that class. So for this example, we would have 0.3*20+0.7*33 = 29.1, which gives the classification of 29 years old. Therefore, instead of multiplying each age's probability by its age value, as done in our tests, we would create a probability for the classes that we are using, calculated by adding the probabilities for each of the ages that are in the class age span. This probability would then be multiplied by the age value that lies in the center of the class. Additionally, probabilities that are lower than a certain threshold and that are expected to be the wrong predicted classes could also be ignored, in order to avoid pushing the age to another class that is not probable of being the right one. Such low probabilities could consequently be merged with their adjacent class that holds the highest probability. These validations could be helpful in predicting correctly near the boundary examples that, as seen before, have a higher chance of being wrongly classified.

An additional validation mechanism can be investigated further in order to analyze the examples where the current model has less confidence with its prediction. Our tests show that such examples are more likely to fail compared to others where they have higher confidence levels and that the introduction of such additional systems can be beneficial, as in the case of Siamese Networks for gender classification. In our implementation, this was implemented for gender classification only, as it demonstrated better results when the initial model is not confident on the gender of the person.

5.2 Advantages:

If we know the gender and age of the person we can use specific vocabulary to make them understand (for example if we know the particular age group of children then we can act accordingly).

It even helps in identifying criminals when we have the picture of the criminal and we then classify the age of the person so it will be helpful to search for the person only in that particular age group.

5.3 Applications:

Applications for this technology have a broad scope and the potential to make a large impact. This could be used to aid assisted vision devices for those with deteriorating, or lost, eyesight.

Social media websites like Facebook could use the information about the age and gender of the people to better infer the context of the image.

ATTACHMENTS

CHAPTER-6

ATTACHMENTS

6.1 User Guide of the Proposed System

The System that is proposed in this document is provided through a folder which contains several required files that include configuration files and executable files. The structure of such a folder is presented in Figure 10. Below are described each file's purpose in the system.

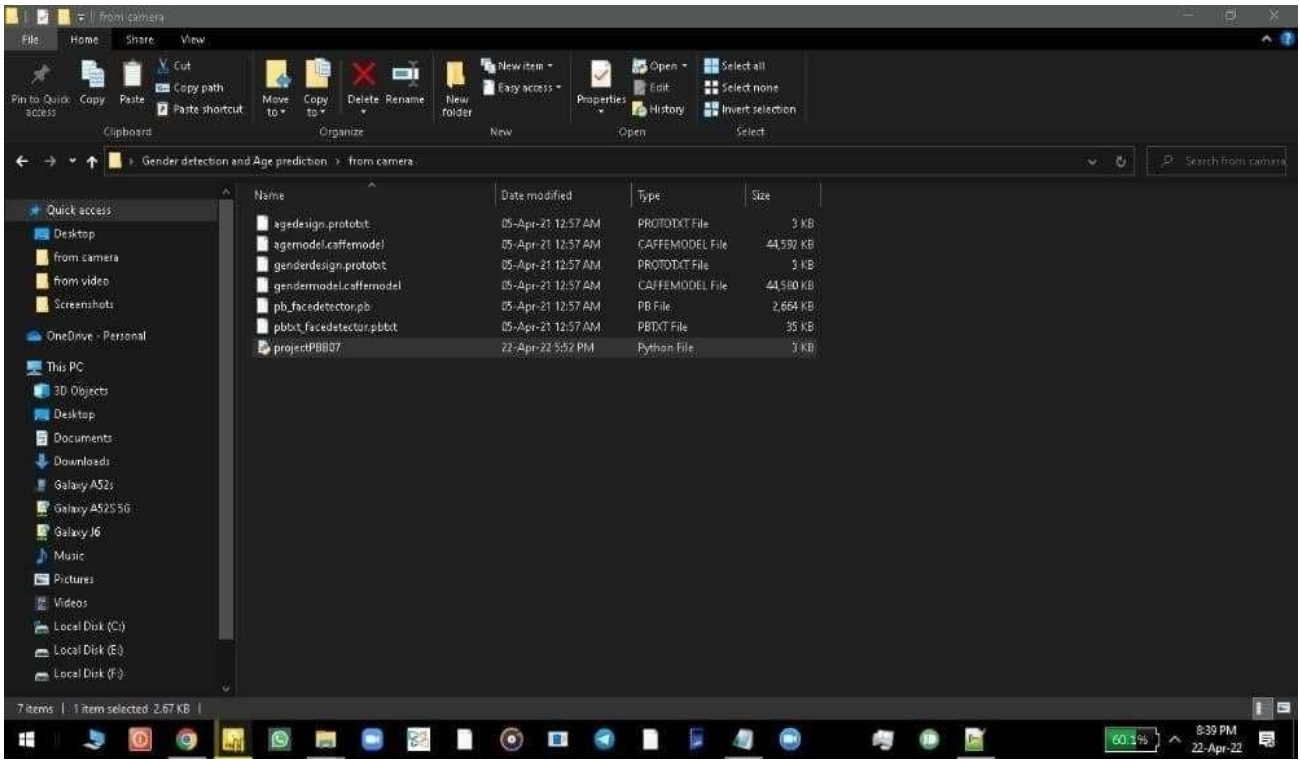


Figure 10- Folder structure

6.2 How to run the model ?

To run the model, use the projectPBB07 python file with the arguments needed. The only required argument is the *img* path for the image that is going to be processed. An example of such usage is below:

python projectPBB07.py – image passphoto.jpg

The output comes in 2 arrays. The first array is for the age predictions, returning the class to which the faces belong to. The second one returns the respective gender classes.

Additional Python Libraries Required :

- **OpenCV:** pip install opencv-python
- **Argparse:** pip install argparse

The contents of this Project :

- pbtxt_facedetector.pbtxt
- pb_facedetector.pb
- agedesign.prototxt
- agemodel.caffemodel
- genderdesign.prototxt
- gendermodel.caffemodel
- a few pictures to try the project on
- projectPBB07.py

For face detection, we have a .pb file- this is a protobuf file (protocol buffer); it holds the graph definition and the trained weights of the model. We can use this to run the trained model. And while a .pb file holds the protobuf in binary format, one with the .pbtxt extension holds it in text format. These are TensorFlow files. For age and gender, the .prototxt files describe the network configuration and the .caffemodel file defines the internal states of the parameters of the layers.

Usage :

- Open your Command Prompt or Terminal and change directory to the folder where all the files are present.

Detecting Gender and Age of face in Image

- Use Command :python detect.py --image <image_name>

Note: The Image should be present in same folder where all the files are present

Detecting Gender and Age of face through webcam

- Use Command :python detect.py

Note: Press **Ctrl + C** to stop the program execution.

APPENDIX

CHAPTER-7

APPENDIX

SOURCE CODE

7.1 projectPBB07.py:

```
#Gender detection and Age prediction Main Project
#by PBB07 batch members
#SWARNANDHRA COLLEGE OF ENGINEERING AND TECHNOLOGY
import cv2
import math
import argparse

def highlightFace(net, frame, conf_threshold=0.7):
    frameOpencvDnn=frame.copy()
    frameHeight=frameOpencvDnn.shape[0]
    frameWidth=frameOpencvDnn.shape[1]
    blob=cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)
#image,scalefactor,size224x224,227x227,299x299,subtractmeanvalues(r=103.93,g=116.77,b=123.68
),swapRB(openCV BGR to mean RGB)
    net.setInput(blob)
    detections=net.forward()
    faceBoxes=[]
    for i in range(detections.shape[2]):
        confidence=detections[0,0,i,2]
        if confidence>conf_threshold:
            x1=int(detections[0,0,i,3]*frameWidth)
            y1=int(detections[0,0,i,4]*frameHeight)
            x2=int(detections[0,0,i,5]*frameWidth)
            y2=int(detections[0,0,i,6]*frameHeight)
            faceBoxes.append([x1,y1,x2,y2])
            cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (0,255,0), int(round(frameHeight/150)), 8)
#image,startpoint,endpoint,color(BGR),thickness,return image value
    return frameOpencvDnn,faceBoxes

parser=argparse.ArgumentParser()
parser.add_argument('--image')
args=parser.parse_args()

faceProto="pbtxt_facedetector.pbtxt"
faceModel="pb_facedetector.pb"
ageProto="agedesign.prototxt"
ageModel="agemodel.caffemodel"
```

```
genderProto="genderdesign.prototxt"
genderModel="gendermodel.caffemodel"

MODEL_MEAN_VALUES=(78.4263377603, 87.7689143744, 114.895847746)
ageList=['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']
genderList=['Male','Female']

faceNet=cv2.dnn.readNet(faceModel,faceProto)
# to load network into memory
ageNet=cv2.dnn.readNet(ageModel,ageProto)
genderNet=cv2.dnn.readNet(genderModel,genderProto)

video=cv2.VideoCapture(args.image if args.image else 0)
padding=20
#wait for 1ms or key pressed here wait 1ms if 0 wait till key pressed to display image
while cv2.waitKey(1)<0 :
    hasFrame,frame=video.read()
    if not hasFrame:
        cv2.waitKey()
        break

    resultImg,faceBoxes=highlightFace(faceNet,frame)
    if not faceBoxes:
        print("No face detected")

    for faceBox in faceBoxes:
        face=frame[max(0,faceBox[1]-padding):
                   min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-padding)
                   :min(faceBox[2]+padding, frame.shape[1]-1)]
#img.shape(0=height,1=width,2=channels(alpha,r,g,b))
        blob=cv2.dnn.blobFromImage(face, 1.0, (227,227), MODEL_MEAN_VALUES,
swapRB=False)
        genderNet.setInput(blob)
        genderPreds=genderNet.forward()
        gender=genderList[genderPreds[0].argmax()]
        print(f'Gender: {gender}')

        ageNet.setInput(blob)
        agePreds=ageNet.forward()
        age=ageList[agePreds[0].argmax()]
        print(f'Age: {age[1:-1]} years')

#slicing index 1 to last but one
        cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 2, cv2.LINE_AA)
#image,text,org,font,fontScale,color in rgb(255,255,0)=yellow,thickness.linetype,bottomLeftOrigin
        cv2.imshow("Detecting age and gender", resultImg)
```

7.2 genderdesign.prototxt:

```
name: "CaffeNet"
input: "data"
input_dim: 10
input_dim: 3
input_dim: 227
input_dim: 227
layers {
  name: "conv1"
  type: CONVOLUTION
  bottom: "data"
  top: "conv1"
  convolution_param {
    num_output: 96
    kernel_size: 7
    stride: 4
  }
}
layers {
  name: "relu1"
  type: RELU
  bottom: "conv1"
  top: "conv1"
}
layers {
  name: "pool1"
  type: POOLING
  bottom: "conv1"
  top: "pool1"
  pooling_param {
  pool: MAX
  kernel_size: 3
    stride: 2
  }
}
layers {
  name: "norm1"
  type: LRN
  bottom: "pool1"
  top: "norm1"
  lrn_param {
  local_size: 5
    alpha: 0.0001
    beta: 0.75
  }
```

```
}
layers {
  name: "conv2"
  type: CONVOLUTION
  bottom:  "norm1"
  top: "conv2"
  convolution_param {
  num_output: 256
    pad: 2
    kernel_size: 5
  }
}
layers {
  name: "relu2"
  type: RELU
  bottom: "conv2"
  top: "conv2"
}
layers {
  name: "pool2"
  type: POOLING
  bottom: "conv2"
  top: "pool2"
  pooling_param {
  pool: MAX
  kernel_size: 3
    stride: 2
  }
}
layers {
  name: "norm2"
  type: LRN
  bottom: "pool2"
  top: "norm2"
  lrn_param {
  local_size: 5
    alpha: 0.0001
    beta: 0.75
  }
}
layers {
  name: "conv3"
  type: CONVOLUTION
  bottom: "norm2"
  top: "conv3"
  convolution_param {
```

```
    num_output: 384
    pad: 1
    kernel_size: 3
  }
}
layers{
  name: "relu3"
  type: RELU
  bottom: "conv3"
  top: "conv3"
}
layers {
  name: "pool5"
  type: POOLING
  bottom: "conv3"
  top: "pool5"
  pooling_param {
  pool: MAX
  kernel_size: 3
    stride: 2
  }
}
layers {
  name: "fc6"
  type: INNER_PRODUCT
  bottom: "pool5"
  top: "fc6"
  inner_product_param {
    num_output: 512
  }
}
layers {
  name: "relu6"
  type: RELU
  bottom: "fc6"
  top: "fc6"
}
layers {
  name: "drop6"
  type: DROPOUT
  bottom: "fc6"
  top: "fc6"
  dropout_param {
    dropout_ratio: 0.5
  }
}
```

```
layers {
  name: "fc7"
  type: INNER_PRODUCT
  bottom: "fc6"
  top: "fc7"
  inner_product_param {
    num_output: 512
  }
}
layers {
  name: "relu7"
  type: RELU
  bottom: "fc7"
  top: "fc7"
}
layers {
  name: "drop7"
  type: DROPOUT
  bottom: "fc7"
  top: "fc7"
  dropout_param {
    dropout_ratio: 0.5
  }
}
layers {
  name: "fc8"
  type: INNER_PRODUCT
  bottom: "fc7"
  top: "fc8"
  inner_product_param {
    num_output: 2
  }
}
layers {
  name: "prob"
  type: SOFTMAX
  bottom: "fc8"
  top: "prob"
}
```

7.3 agedesign.prototxt:

```
name: "CaffeNet"
input: "data"
input_dim: 1
input_dim: 3
input_dim: 227
input_dim: 227
layers {
  name: "conv1"
  type: CONVOLUTION
  bottom: "data"
  top: "conv1"
  convolution_param {
    num_output: 96
    kernel_size: 7
    stride: 4
  }
}
layers {
  name: "relu1"
  type: RELU
  bottom: "conv1"
  top: "conv1"
}
layers {
  name: "pool1"
  type: POOLING
  bottom: "conv1"
  top: "pool1"
  pooling_param {
  pool: MAX
  kernel_size: 3
    stride: 2
  }
}
layers {
  name: "norm1"
  type: LRN
  bottom: "pool1"
  top: "norm1"
  lrn_param {
  local_size: 5
    alpha: 0.0001
    beta: 0.75
  }
}
```

```
layers {
  name: "conv2"
  type: CONVOLUTION
  bottom:  "norm1"
  top: "conv2"
  convolution_param {
  num_output: 256
    pad: 2
    kernel_size: 5
  }
}
layers {
  name: "relu2"
  type: RELU
  bottom: "conv2"
  top: "conv2"
}
layers {
  name: "pool2"
  type: POOLING
  bottom: "conv2"
  top: "pool2"
  pooling_param {
  pool: MAX
  kernel_size: 3
    stride: 2
  }
}
layers {
  name: "norm2"
  type: LRN
  bottom: "pool2"
  top: "norm2"
  lrn_param {
  local_size: 5
    alpha: 0.0001
    beta: 0.75
  }
}
layers {
  name: "conv3"
  type: CONVOLUTION
  bottom:  "norm2"
  top: "conv3"
  convolution_param {
  num_output: 384
    pad: 1
```

```
    kernel_size: 3
  }
}
layers{
  name: "relu3"
  type: RELU
  bottom: "conv3"
  top: "conv3"
}
layers {
  name: "pool5"
  type: POOLING
  bottom: "conv3"
  top: "pool5"
  pooling_param {
  pool: MAX
  kernel_size: 3
    stride: 2
  }
}
layers {
  name: "fc6"
  type: INNER_PRODUCT
  bottom: "pool5"
  top: "fc6"
  inner_product_param {
    num_output: 512
  }
}
layers {
  name: "relu6"
  type: RELU
  bottom: "fc6"
  top: "fc6"
}
layers {
  name: "drop6"
  type: DROPOUT
  bottom: "fc6"
  top: "fc6"
  dropout_param {
    dropout_ratio: 0.5
  }
}
layers {
  name: "fc7"
  type: INNER_PRODUCT
```

```
bottom: "fc6"
  top: "fc7"
  inner_product_param {
    num_output: 512
  }
}
layers {
  name: "relu7"
  type: RELU
  bottom: "fc7"
  top: "fc7"
}
layers {
  name: "drop7"
  type: DROPOUT
  bottom: "fc7"
  top: "fc7"
  dropout_param {
    dropout_ratio: 0.5
  }
}
layers {
  name: "fc8"
  type: INNER_PRODUCT
  bottom: "fc7"
  top: "fc8"
  inner_product_param {
    num_output: 8
  }
}
layers {
  name: "prob"
  type: SOFTMAX
  bottom: "fc8"
  top: "prob"

}
```

REFERENCES

CHAPTER-8

REFERENCES

- Jo Chang-yeon, "*Face Detection using LBP features",* 2022, Retrieved February 19, 2022 from: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.217.6206 . doi: 10.1.1.217.6206
- Nagesh Singh Chauhan (2021), Predict Age and Gender using Convolutional Neural Network and openCV. Retrieved from https://towardsdatascience.com/predict-age-and-gender-usingconvolutional-neural-network-and-opencv-fd90390e3ce6
- Xudong Suna, Pengcheng Wua & Steven C.H. Hoi, "*Face detection using deep learning: An improved faster RCNN approach"*, 2020. doi: 10.1016/j.neucom.2020.03.030
- Sergey Ioffe & Christian Szegedy, "*Batch Normalization: Accelerating Deep Network Training byReducing Internal Covariate Shift"*, 2020, Retrieved November 26, 2020 from: https://arxiv.org/abs/1502.03167
- Tsun-Yi Yang, Yi-Hsuan Huang, Yen-Yu Lin, Pi-Cheng Hsiu & Yung-Yu Chuang, "*SSR- Net: A Compact Soft Stagewise Regression Network for Age Estimation"*, 2019, In Proceedings ofthe Twenty-Seventh International Joint Conference on Artificial Intelligence. doi: 10.24963/ijcai.2018/150
- Sebastian Lapuschkin, Alexander Binder, Klaus-Robert Muller & Wojciech Samek, "*Understanding and Comparing Deep Neural Networks for Age and Gender Classification"*, 2019, In Proceedings of the IEEE International Conference on Computer Vision p. 1629-1638. doi: 10.1109/ICCVW.2017.191
- Rajeev Ranjan, Swami Sankaranarayanan, Carlos D. Castillo & Rama Chellappa, "*An All-In-One Convolutional Neural Network for Face Analysis"*, 2018, In Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition. doi: 10.1109/FG.2018.137
- Shuo Yang, Yuanjun Xiong, Chen Change Loy & Xiaoou Tang, "*Face Detection through Scale-Friendly Deep Convolutional Networks"*, 2018, Neurocomputing Volume 299 p.42-50
- Afshin Dehghan, Enrique G. Ortiz, Guang Shu & Syed Zain Masood, "*DAGER: Deep Age, Gender and Emotion Recognition Using Convolutional Neural Networks"*, 2017, Retrieved December 3, 2017 from: https://arxiv.org/abs/1702.04280

- Pattern Recognition Volume 72 p.15-26. doi: 10.1016/j.patcog.2017.06.031
- Sergey Zagoruyko & Nokis Komodakis, "*Wide Residual Networks*", 2017, Retrieved February 17, 2017 from: https://arxiv.org/abs/1605.07146
- Ujjwal Karn, "*An Intuitive Explanation of Convolutional Neural Networks"*, 2016. Retrieved January 5, 2016 from: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
- Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li & Yu Qiao, "*Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks*", 2016, IEEE Signal Processing Letters Volume *23* p.1499-1503. doi: 10.1109/LSP.2016.2603342
- Gil Levi & Tal Hassner, "*Age and Gender Classification using Convolutional Neural Networks"*, 2015, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition p.34-42. doi: 10.1109/CVPRW.2015.7301352
- Rasmus Rothe, Radu Timofte & Luc Van Gool, "*Deep EXpectation of apparent age from a single image*", 2015, Looking at People Workshop at International Conference on Computer Vision. doi: 10.1109/ICCVW.2015.41
- *Detection: A Deep Learning Approach"*, 2015, In Proceedings of the IEEE International Conference on Computer Vision p. 3676-3684. doi: 10.1109/ICCV.2015.419
- Yann LeCun, Yoshua Bengio & Geoffrey Hinton, "*Deep Learning"*, 2015, Nature Volume 521 p. 436–444. doi: 10.1038/nature14539
- Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt & Gang Hua, "*A Convolutional Neural Network Cascade for Face Detection"*, 2015, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition p. 5325-5334. doi: 10.1109/CVPR.2015.7299170
- IEEE Workshop on Analysis and Modeling of Faces and Gestures (AMFG), at the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, 2015.
- Eran Eidinger, Roee Enbar & Tal Hassner , "*Age and Gender Estimation of Unfiltered Faces"*, 2014, IEEE Transactions on Information Forensics and Security Volume 9 p. 2127-2179. doi: 10.1109/TIFS.2014.2359646