AGE AND GENDER CLASSIFICATION USING CNN

A PROJECT REPORT

Submitted by

ANWEASHA SAHA [Reg No: RA1911003010235]

NITHISH KUMAR S [Reg No: RA1911003010217]

Under the Guidance of

Mrs. V. S. SARANYA

(Assistant Professor, Department of Computing Technologies)

In partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTING TECHNOLOGIES

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603203

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 18CSP109L project report titled "**AGE AND GENDER CLASSIFICATION USING CNN**" is the bonafide work of **ANWEASHA SAHA [Reg. No.: RA1911003010235]** and **NITHISH KUMAR S [Reg. No.: RA1911003010217]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Mrs. V. S. SARANYA
SUPERVISOR
Assistant Professor
Department of Computing Technologies

DR. B. BARANIDHARAN
PANEL HEAD
Associate Professor
Department of Computing Technologies

DR. M. PUSHPALATHA
HEAD OF THE DEPARTMENT
Department of Computing Technologies

INTERNAL EXAMINER

EXTERNAL EXAMINER

Department of Computing Technologies

SRM Institute of Science and Technology

Own Work Declaration Form

Degree/ Course **:** B.Tech in Computer Science and Engineering

Student Names **:** ANWEASHA SAHA, NITHISH KUMAR S

Registration Number: RA1911003010235, RA1911003010217

Title of Work **:** Age and Gender Classification Using CNN

We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that we have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own

- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g., fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:
I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except were indicated by referring, and that I have followed the good academic practices noted above. **Student 1 Signature:** **Student 2 Signature:** **Date:**
If you are working in a group, please write your registration numbers and sign with the date for every student in your group.

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T.V. Gopal**, Dean, SRM Institute of Science and Technology, for his invaluable support.

We extend our sincere thanks to **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. M. Pushpalatha**, Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for her valuable suggestions and encouragement throughout the period of the project work.

We are extremely grateful to our Project Coordinators, **Dr. M. Kanchana**, **Dr. G. Usha**, **Dr. R. Yamini,** and **Dr. K. Geetha**, Panel Head, **Dr. B. Baranidharan**, Associate Professor and Panel Members, **Dr. P. Saranya**, Assistant Professor, **Mrs. M. Rajalakshmi**, Assistant Professor, **Mrs. P. Nithyakani**, Assistant Professor, **Mr. V. Arulalan,** Assistant Professor, **Mrs. V. S. Saranya**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for their inputs during the project reviews.

We register our immeasurable thanks to our Faculty Advisor, **Dr. P. Madhavan**, Associate Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Mrs. V. S. Saranya**, Assistant Professor, Department of Computing Technologies, for providing us an opportunity to pursue ourproject under her mentorship. He provided us the support to explore the research topics of our interest.

We sincerely thank staff and students of the Computing Technologies Department, SRM Institute of Science and Technology, for their help during our project work. Finally, we would like to thank our parents, our family members and our friends for their unconditional love, constant support and encouragement.

ANWEASHA SAHA [RA1911003010235]

NITHISH KUMAR S [RA1911003010217]

ABSTRACT

The classification of age and gender has drawn increased attention recently because of its significance in creating user-friendly intelligent systems. Age estimation from a single facial image has been a key task in the fields of image processing and computer vision. Convolutional Neural Network (CNN) based techniques have been frequently adopted for the classification problem in the recent past because of their precise results in facial analysis. This study presents an end-to-end CNN approach for obtaining accurate gender and age group classification of real-world faces. The complete feature extraction and classification processes are included in the two-level CNN architecture. The feature extraction task pulls features that are related to gender and age while the classification assigns the facial photographs to the proper gender and age group. The experiment results appear to support the claim that our model may perform better in gender and age group categorization when analysed for classification accuracy using the equivalent Adience benchmark. Technically speaking, our network will be trained and tested on both Adience (original) and IMDB-WIKI datasets.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DNN	Deep Neural Network
ANN	Artificial Neural Network
ReLU	Rectified Linear Unit
NN	Neural Network
FC-CNN	Fully Connected Convolutional Neural Network
CNN	Convolutional Neural Network
SGD	Stochastic Gradient Descent
LMTCNN	Light Weight Multi Task CNN
MAE	Mean Absolute Error
LSTM	Long Short Term Memory
SVM	Support Vector Machine
SGD	Stochastic Gradient Descent

CHAPTER 1

INTRODUCTION

1.1 General

The area of computer vision was created as a result of increased knowledge of how biological neurons in the brain's visual cortex function. The key lesson from biology is that processing visual data begins with the detection of basic forms like edges before moving on to more complex structures. Extraction of three-dimensional forms from two-dimensional pictures was the primary goal of computer vision in its early years.

Neocognitron, a multi-layered artificial neural network with convolutional layers created by Japanese researcher Kunihiko Fukushima, marked a significant advancement in the area. The first convolutional neural network, LeNet-51, was then launched in 1989 after scientist Yann LeCun used the backpropagation method to the Neocognitron.The next breakthrough came with the introduction of AlexNet by Krizhevsky. AlexNet is an 8 layer convolutional neural network which contains neurons with non-saturating ReLU activation. The network won the 2012 ImageNet Large Scale Visual Recognition Challenge achieving top-5 error of 15.3%. AlexNet is considered the most influential neural network architecture.

A picture of a face may be used to infer a variety of characteristics, including age, gender, expression, ethnicity, and mood. The most important characteristics are age and gender. The potential uses are simple to envision, ranging from marketing to security systems to human- computer interaction.

The issue of properly measuring age and gender can be resolved using a variety of techniques. The first method focuses on manually extracting parameters including head size, eye placement, and nose length. End-to-end deep learning models are the foundation of a different strategy. A new hybrid strategy can be created by combining the two approaches. Most modern methods use deep learning approach.

Labelled datasets are important to train age and gender models. There are many of datasets that are publicly accessible that have age or gender annotations. Another choice is to manually categorise photographs of faces to produce a new dataset.

The goal of this thesis was to develop an application for identifying people's ages and genders using a model that might be used to make predictions about the future. Many models only consider limited face datasets, making them unsuitable for in-the-wild estimation. We will emphasise deep learning end-to-end methodologies in this thesis. Three distinct deep learning models will be assessed.

1.2 Purpose

Through this project, we aim to deliver a feasible, and viable method to detect and predict the age and gender of human facial images. It should be taken into consideration that the tasks of age prediction and gender classification are not easy for human beings, let alone machine learning models. There are many difficulties which hinder this task such as changes in alignment of the face, cultural and regional differences among people, data ambiguity, etc. For example, these models may struggle to accurately predict the gender and age for people belonging to different cultures or regions, owing to their differences in physical characteristics and aging patterns. Another instance is that long hair is an appearance trait usually associated with women, but this is not typically the case. Moreover, training predictor models for these tasks require large datasets to train and improve the performance. However, if the dataset is biased towards a certain age range or gender, then the model might not be able to determine the other age ranges or gender equally well. As such, the accuracy will be compromised. Thus, it is a complex process for AI models to predict correctly.

The most common deep learning approach to predict age and gender from facial images is using a CNN model with two phases, that is, feature extraction phase and classification phase. Conducting the literature survey, we have seen that various types of convolutional neural networks have been used in the previous works. Some of the noteworthy ones are Multi-Task CNN (MTCNN), Light Weight Multi-Task CNN (LWMTCNN), CNNs with two or three convolutional layers, etc. Other popular methodologies that have been used are transfer learning and LSTM. In this study, we have used a six layer CNN model consisting of two fully connected layers and four convolutional layers. Here, gender detection has been considered as binary classification and age prediction to be a multi class classification problem.

1.3 Scope

For the project on hand, we have collected unfiltered real world facial images from two datasets, namely, Adience dataset and IMDB WIKI dataset. The former contains about more than twenty six thousand face images and the latter half a million images with gender and age labels for training. Our solution requires the facial images to be consistent in a way so that the original image is denoised and the face and its parts are detected and aligned properly. This allows the images to be suitable for training to feed to our CNN model in order to ensure better accuracy.

Age and gender recognition has various use cases in different aspects of life. This AI technique can be used in a wide range of industries and applications like authentication, security, retail businesses, human-computer interaction, surveillance, law enforcement, marketing, access control, interactive systems, and so on. In addition it helps with customer service for personalising interactions with customers, and on social media to deliver targeted content to specific users. In healthcare, age and gender prediction can prove to be useful to comprehend certain trends or patterns in particular diseases.

1.4 AI, Machine Learning and Deep Learning

Artificial Intelligence is a subset of computational theory and logic which focuses on the development of computer systems and algorithms designed to perform tasks that usually require human intelligence, logical thinking and expertise. These programs and applications imitate the behaviour of the human mind, hence require superabundant knowledge related to all the variables of the problem. These include the various objects, their properties, differentiating categories and the relationships between them. Artificial Intelligence can be broadly classified into two types depending on the kind of problems they are developed to solve.

- **Vertical AI**: These AI algorithms focus on learning to solve a single problem. They are usually restrictively programmed to follow the instructions for a single, automated and repetitive task. For example, scheduling meetings and calls on a daily basis, 5 automatically updating the database with information obtained from a single external source.

- **Horizontal AI**: These AI algorithms focus on broader problem statements. Applications classified under this type of AI have the ability to handle multiple tasks and cater to all multiple needs of their user with a single logic and setting. For example, virtual assistants like Siri, Cortana and Alexa that tackle multiple tasks for their users.

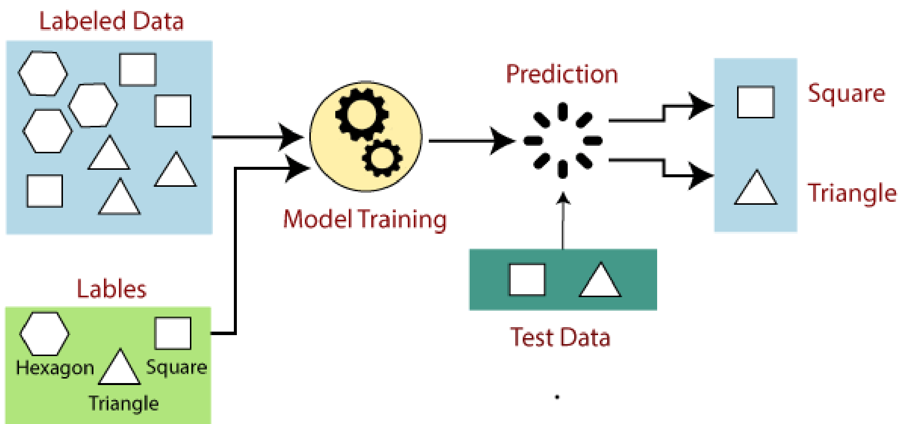
Machine Learning (ML), though often confused and interchangeably used with Artificial Intelligence, is a subset of artificial intelligence and is the science of developing machine intelligent algorithms that help the machine learn from existing and past data. The algorithms achieve this by identifying and learning the various relationships between individual features of the data, looking for common patterns, responding to different situations outside of their programming restrictions and makes predictions accordingly.

Machine Learning can be broadly classified into three categories (Figure: 1.1.1) based on the kind of data they use to learn and the type of outputs produced by them. They are:

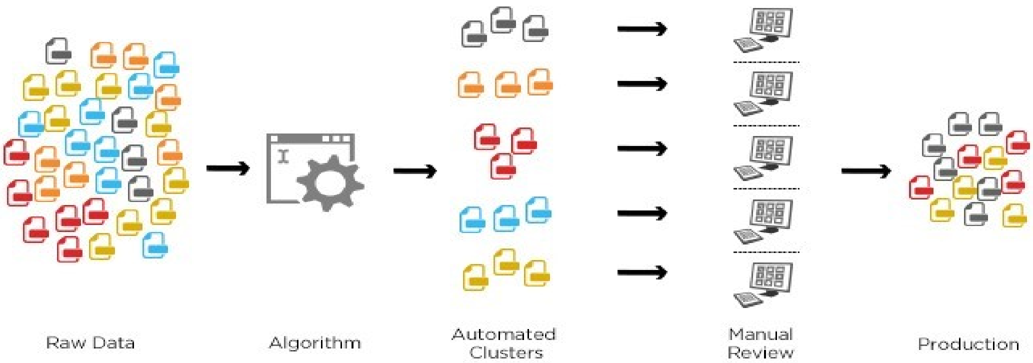
- **Supervised Learning**: In this type of learning, the datasets that the machine utilises to learn are entirely structured and labelled. The datasets have a set of input features and their corresponding outputs and are given to the ML algorithm as inputs for training. While training, the model learns by mapping its predictions to the true predictions present in the dataset, evaluates its performance, and automatically updates its learning curve to achieve better scores of the performance metrics. For example, classification and regression tasks such as image classification and commodity price prediction respectively.

- **Unsupervised Learning**: In this type of learning, the datasets are not mapped with their outputs. The models are trained on a dataset consisting of several features and are expected to produce outputs by learning and identifying common patterns without conforming to a certain "correct answer". For example, clustering algorithms used for customer segmentation, DNA pattern recognition and grouping in evolutionary biology.

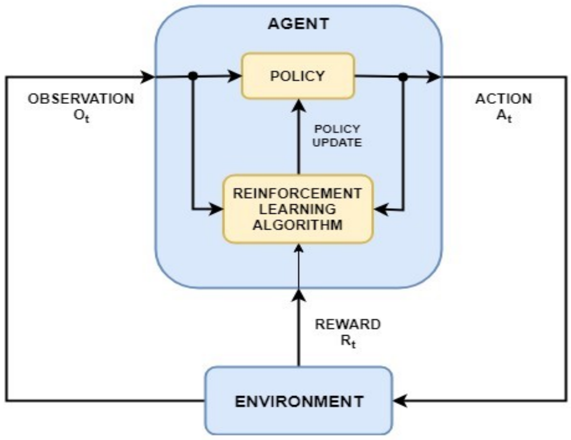
- **Reinforcement Learning**: In this type of learning, the algorithm updates its learning curve by studying the features of the learning problem, environment and the behaviour of the agent in order to maximise its performance. It works on a action-reward based system by learning any method that is well suited to solve the problem, while simultaneously correcting any method that does not. The ultimate goal is reached efficiently and rapidly and the machine makes sure to learn only those methods that facilitate this. For example, autonomous driving and gaming technologies.



(a)



(b)



(c)

Figure 1.1 Types of Machine Learning: (a) Supervised Learning (b) Unsupervised Learning and (c) Reinforcement Learning

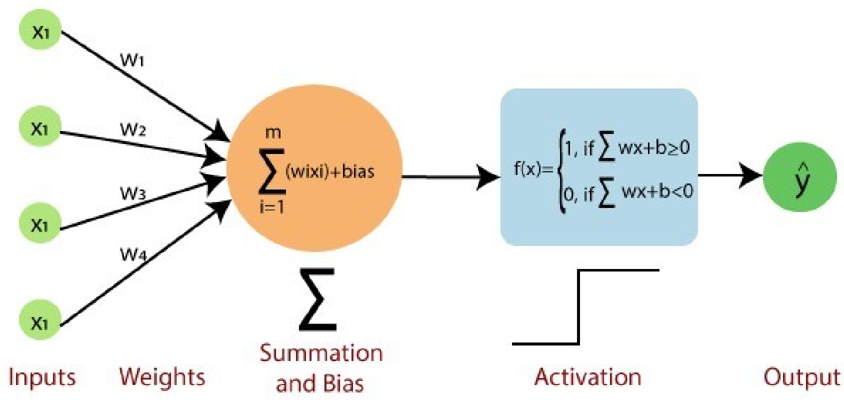
Deep Learning is a subset of machine learning, where the machine intelligent models are essentially imitations of the human brain. The model, called an Artificial Neural Network, is similar to the human brain where connections exist between the individual units (neurons). However, unlike the human brain, the individual perceptrons are restricted to forming a set number of connections with certain layers and are allowed to propagate data only in restricted directions defined according to the type of the neural network.

Neural networks can be designed and trained to perform any kind of task that machine learning algorithms can and are more robust than ML models. These networks have the ability to learn from raw data, grasp the various features as the data is passed on through the different layers while simultaneously compressing the data and produce the output from these set of features extracted during propagation.

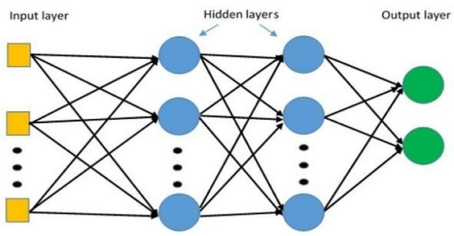
A perceptron, which is an individual unit of a layer of a neural network, is a complex mathematical function which takes a set of inputs, adds weights and biases separately to each input and passes it through a non-linear function to produce the output. This output is carried to the perceptron of the next layer through a connection link along with certain other characteristics such as the activation signal, previous internal state, etc. depending on the type of neural network.

There are two types of perceptrons:

- **Single-layer perceptron**: They can only learn linearly-separable tasks. For example, linear binary classification problem.
- **Multi-layer perceptron**: Also called a fully connected neural network. It contains two or more layers (input layer, hidden layer(s) and output layer), which helps bring out the non-linearity in the input data, thereby increasing processing power and helping the model learn better. For example, image recognition, stock analysis.



(a)



(b)

Figure 1.2 (a) Single-Layer Perceptron and (b) Multi-Layer Perceptron

1.5 Computer Vision

Computer vision is a subset of artificial intelligence which includes the methods designed to enable machines to process visual data such as images, videos and data from other visual inputs in order to derive meaningful information from them. These methods help machines gather the knowledge and learn the context to tell objects apart, understand the various components and features present in an image, and process sequences of slices of 3D images and videos.

Computer vision uses machine learning and deep learning models such as convolution neural networks (CNNs) to learn, analyse and interpret visual data like humans do. At a basic level, computer vision is mostly pattern recognition. Hence, the most common way to increase the efficiency of a computer vision algorithm is to feed it large amounts of labelled visual data and further use different methods and techniques to help the machine learn these patterns. Computer vision can further be classified into various domains based on its applications (figure 1.3.). For example, scene reconstruction, object detection, object recognition, event detection, motion estimation, 3D pose estimation, 3D scene modelling and image restoration to name a few.

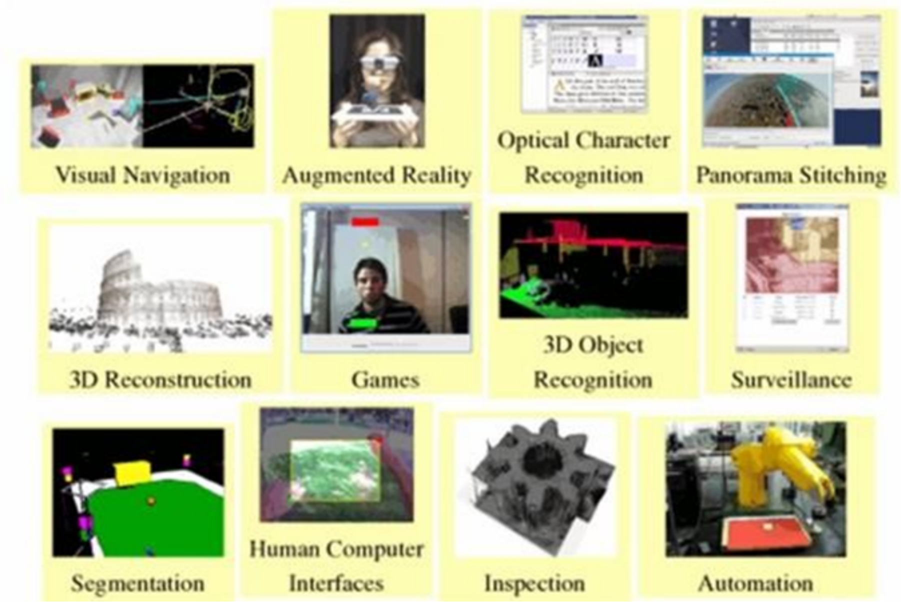


Figure 1.3 Applications of Computer Vision

1.6 Software Requirements Specification

1.6.1 Introduction to Python

Python is a high-level programming language that is interpreted and object-oriented, and it has dynamic semantics. As a scripting or glue language for integrating existing components, it is particularly well suited for Rapid Application Development due to its high-level constructed data structures, dynamic typing, and dynamic binding. Readability of Python's clear, uncomplicated syntax is prioritised above other considerations, resulting in lower software maintenance costs. Python encourages programme customization and code reuse thanks to its availability for modules and packages. Python's processor and basic library are free to use, share, and change, and are available in original or object code for all popular platforms. Python is open source software. Because no compilation is required, the edit-test-debug cycle is very quick. When debugging Python programmes, it is important to remember that a flaw or erroneous input will never result in a segmentation failure. Whenever an error is detected by the interpreter, an exception is thrown. Python is classified as an intermediate language due to the fact that its instructions are executed by a programme known as an interpreter.

1.6.2 Keras

Keras is equipped with a plethora of extra tools for working with picture and text data, including layers, objectives, nine activation functions, optimizers, and a range of other applications. Convolutional and recurrent neural networks are both available to use with Keras, in addition to the regular neural networks. Users of Keras are granted the ability to 23 productize deep models on mobile platforms (iOS and Android), the web, and the Java Virtual Machine.

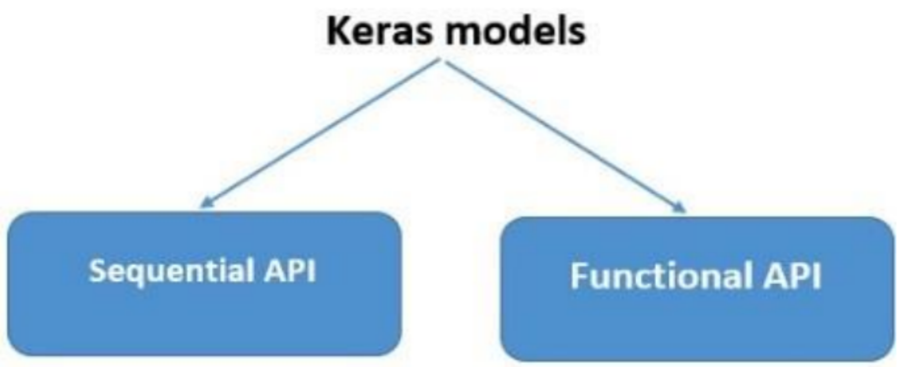


Figure 1.4 Models of Keras

1.6.3 TensorFlow

TensorFlow is a framework for dataflow and differentiable programming that may be utilised for a wide range of different tasks. It is a package for symbolic mathematics that is used in neural networks and other applications that deal with machine learning. Object identification algorithms that are built on TensorFlow are able to classify and recognise arbitrary items that are present in larger photos. In engineering applications, this is frequently used to recognise shapes for modelling (3D space creation from 2D photographs), and it is also utilised by social networking platforms to tag photos (Facebook's Deep Face).

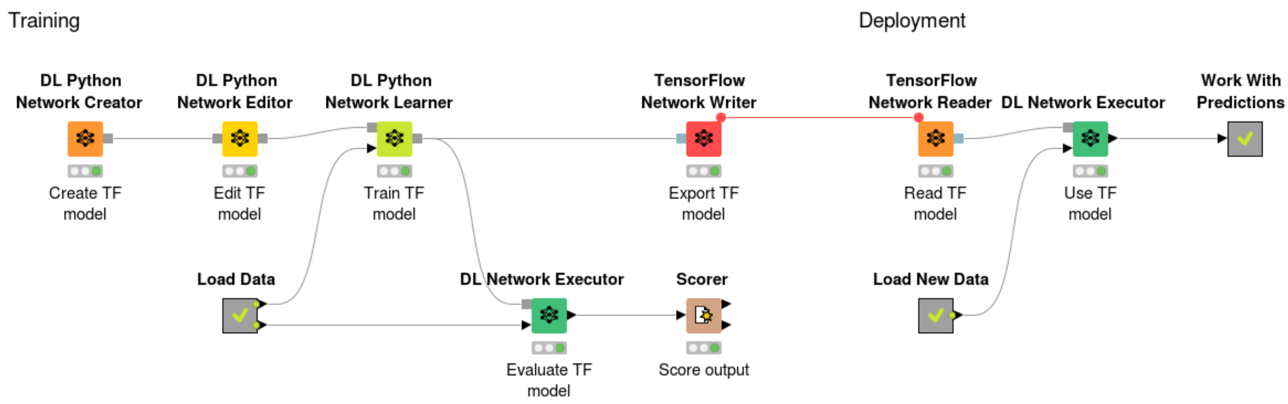


Figure 1.5 TensorFlow Block Diagram

1.6.4 Fast.ai

Fast.ai is a popular python framework used for deep learning that provides high-level as well as low-level aspects of a deep learning framework. This is built with contemporary aspects and it offers programmers high-level attributes that can rapidly and easily deliver state-of-the-art results in common deep learning domains. The goal is to do both of these objectives without making significant sacrifices in terms of ease of use, versatility, or productivity. This is made feasible by a carefully layered design, which describes the basic underlying patterns of a wide variety of deep learning and data analysis methods in the form of decoupled abstractions. This makes it possible for the system to achieve the desired results. By using the dynamic of the core Python language and the adaptability of the PyTorch library, these abstractions may be stated in a manner that is both succinct and crystal obvious. Fast.ai is built with two primary design objectives in mind: first, it should be user-friendly and swiftly productive; second, it should be highly hackable and flexible. It is constructed on the foundation of a hierarchy of relatively low APIs, each of which provides composable building pieces, and this structure serves as its foundation. Because of this, a user who wants to modify a section of the high-level API and perhaps even add specific behavior to fit their requirements does not need to learn how to utilize the lowest level in order to accomplish either of these goals.

1.6.5 Dlib

Generally, face and landmark detectors finds the frontal human faces in an image and estimate their pose. They extracts important points from the faces such as along the eyebrows, corners of the mouth, on the eyes and so on. The Dlib landmark detector is one of the most widely used library which has a very good accuracy. In fact, it outputs 68 landmarks out of a single facial image. This package is made using a linear classifier with HOG (Histogram of Oriented Gradients) feature. This package was created with an ensemble of regression trees by Vahid Kazemi and Josephine Sullivan in the year 2016. It was trained on iBUG-w Face landmark dataset. To use it, a dependent package name Cmake should be installed.

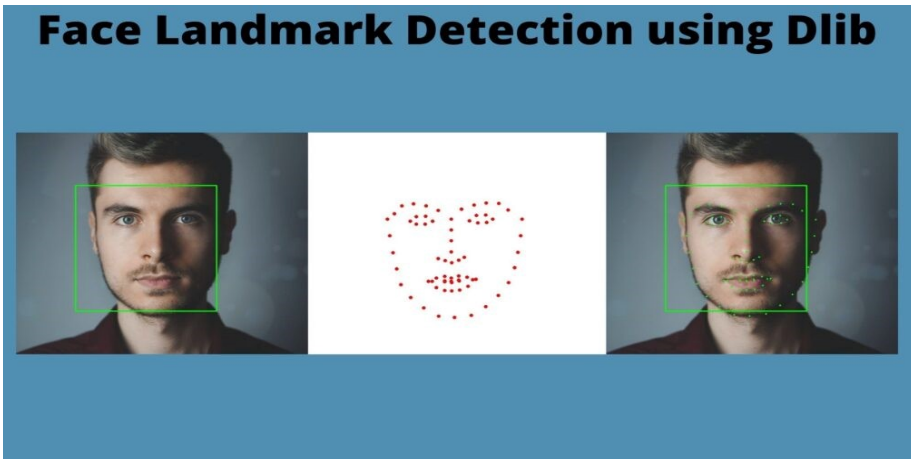


Figure 1.6 Face Landmark Detection Using Dlib Library

1.6.6 PyCharm

PyCharm is an integrated development environment (IDE) for Python that was built by JetBrains on a hybrid platform. It is a tool that is frequently used for the development of Python applications. PyCharm is an integrated development environment (IDE) for the Python programming language that is used by some of the most successful companies in the world, including Twitter, Facebook, Amazon, and Pinterest.

PyCharm can be used on computers running Windows, Linux, or the Mac operating system, and it includes modules and packages that assist programmers in developing applications using Python in a shorter amount of time and with less work. In addition to that, it can be altered to meet the specific needs of the developers that use it.

1.6.7 IDLE

IDLE is a Python 2.x or Python 3 integrated development and learning environment for editing and executing applications. Incorporated development environment (IDE) for Python that comes with pre with the language's default runtime. With the Python interpreter, the IDLE GUI is installed automatically. IDLE was created exclusively for Python users. IDLE has a variety of tools that aid Python programme development, including excellent syntax highlighting.

CHAPTER 2

LITERATURE SURVEY

If you're looking for a comprehensive look at what's out there on a subject, a literature review is the best place to start. The material presented here is derived from independent sources and includes released knowledge on a certain topic and, on occasion, information from a particular time period. Preparation for a proposed study or a synopsis of sources is the ultimate purpose of this kind of paper, which serves as a foundation for further study in the field. Typically, it has a logical structure and incorporates both summation and synthesis. Unlike a summary, a synthesis involves rearranging and rearranging information from a source. As a result, it may provide a fresh take on old material, or it could look back at how the profession has evolved through time, including significant arguments. There are a variety of ways in which a literature review may be used, depending on the circumstances. Up until now, much researches have been done to predict electric power consumption utilizing different machine learning techniques and algorithms.

2.1 Review of Literature

In the realm of facial analysis, predicting an individual's age has proven to be one of the most difficult tasks. In the past, age estimation was utilized to manually extract facial features; but, more recently, CNN approaches have been favored because to the effective training of CNN directly on age datasets. This is the case since CNN methods are being used to extract facial characteristics.

Transfer learning is implemented with the use of two pre-trained models known as VGG19 and VGGFace, in addition to a number of other training strategies (referenced in [1]). A hierarchy of deep CNNs has been studied that first groups subjects by gender before estimating their age. This is accomplished by using male and female age models that are separate from one another. The suggested method has obtained an accuracy of 98.7% for gender classification on the MORPH-II dataset and a mean absolute error (MAE) of 4.1 years for age prediction. To predict the age and gender recognition simultaneously, a lightweight multi-task convolutional neural network (LMTCNN) has been created by Reference [2] that is more effective than baseline multi-task CNN approaches. The use of depth-wise separable convolution is one of the strategies that LMTCNN puts to work so that it can cut down on the overall size of the model and speed up the inference process. On the Adience dataset, the accuracies achieved for the age classification are 44.26%, and the accuracies obtained for the gender classification are 85.16%.

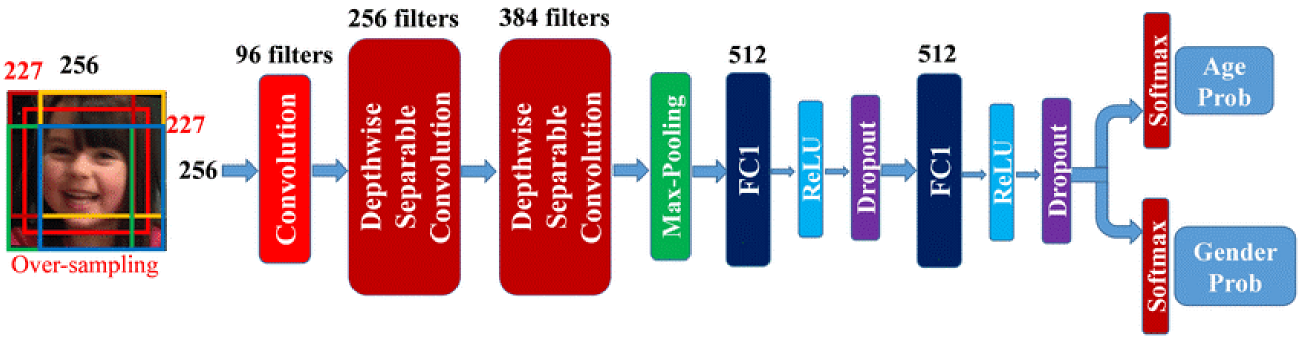


Figure 2.1 The Architecture of the LMTCNN

A Multi-Task Convolution Neural Network (MTCNN) was proposed in the same year by Reference [3]. This network makes use of joint dynamic loss weight modification in order to distinguish age and gender. The BEFA challenge dataset has an average accuracy of 93.72% for gender classification, but the UTKFace dataset has an average accuracy of 98.23% for gender classification. When compared to the BEFA challenge dataset, the UTKFace dataset has an accuracy of age estimation of 70.1%, while the BEFA challenge dataset has an accuracy of 71.83%.

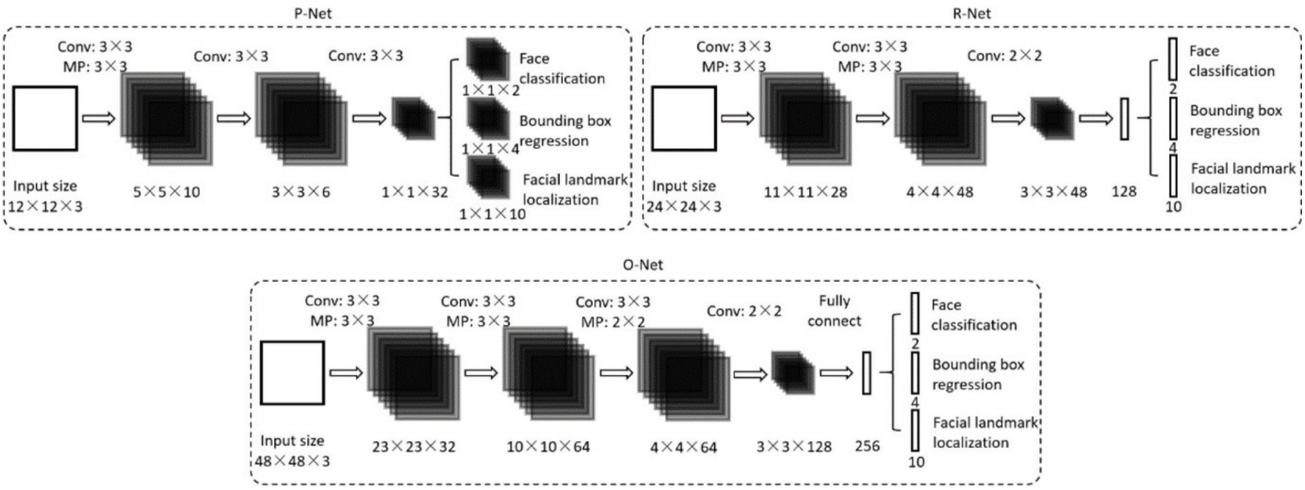


Figure 2.2 The Architecture of the MTCNN

Back-propagation is used inside a CNN-based architecture to use Gabor filter responses as input for integrated identification of gender as well as age by the following method [4]. Using the Adience dataset, the network performed with 61.3% accuracy when it came to age recognition and 88.9% when it came to gender categorization.

Reference [5] has introduced a novel method for estimating a person's age at a more granular level using the attention long-term memory (AL) network. The method generates AL-RoR or AL-ResNets networks by integrating residual network (ResNet) models with LSTM units. These models represent either the residual network of residual networks (RoR) or the residual network of residual networks (ResNets). These networks significantly improve the accuracy by isolating the regional characteristics of age-sensitive areas, which results in an increase. When it comes to the classification of age groups, the Adience dataset and the MORPH Album 2 dataset, respectively, yield an accuracy of 67.83% and 2.63 MAE correspondingly.

The CNN model described in reference [6] is a straightforward implementation with three convolutional layers. The training of the model has been done with the assistance of HAAR Feature-based Cascade Classifiers in order to accomplish a discernible level of progress. The approach produced age identification accuracy rates of 79.3% and gender identification accuracy rates of 50.7%, respectively.

The same year, it was revealed that in camera-based testing, a CNN dubbed Googlenet that was trained with grayscale photos displayed considerably superior age group and gender prediction accuracy. [7] published this finding. It would appear that the use of grayscale image files makes CNN more robust to shifts in lighting than the use of RGB images. The performance of this network was 78.10% and 38.50% in terms of accuracy and correctness in classification of gender and age respectively.

A two-stage solution utilizing a modified MobileNet was suggested in Reference [8, and it was subsequently incorporated into an Android application]. In the beginning, the CNN network determines the subject's age and gender while simultaneously extracting aspects of the face that are appropriate for face recognition. After the faces have been retrieved, the second stage, which utilizes hierarchical agglomerative clustering algorithms, subsequently groups the extracted faces. On the UTKFace dataset, this technique achieves a gender recognition accuracy of 94.1% and a mean absolute error (MAE) of 5.44 for age prediction.

Reference [9] has adopted a transfer learning strategy in order to cut down on the amount of time spent training while simultaneously improving the overall accuracy. They have utilized ImageNet pre-trained models by segmenting those models into various stages with a schedule that gradually decreases the learning rate. They have achieved an accuracy of 91.09 percent through the use of this strategy.

CNN 1, CNN 2, and CNN 3 are the names of the three different CNN models that have been published in Reference [10]. These models each have a different number of filters, pooling, and convolutional layers in their respective architectures. In terms of gender classification, CNN 3 has provided the best accuracy rates, which are 94.49% for the IMDb dataset and 93.56% for the WIKI dataset respectively. In order to validate the findings of this model, it has been applied to the process of age estimation on a class-by-class basis, specifically for adults between the age of 29-39 years, 40-59 years, and 60+ years. On the IMDb dataset, a mean accuracy of 86.20% was found, while on the WIKI dataset, a mean accuracy of 83.97% was found.

The efficiency of Wide Residual Networks was investigated in the same year in the reference [11], which was applied to a dataset of surveillance video gathered from two CCTV cameras in a clothing store. This study made use of a model known as the Wide ResNet 16-8, which is a variant of the WRN network that contains 16 conv2D layers and a widening by a factor of 8. The accuracy of gender categorization task is 82.926 percent, whereas the performance of age prediction is 78.81% accuracy.

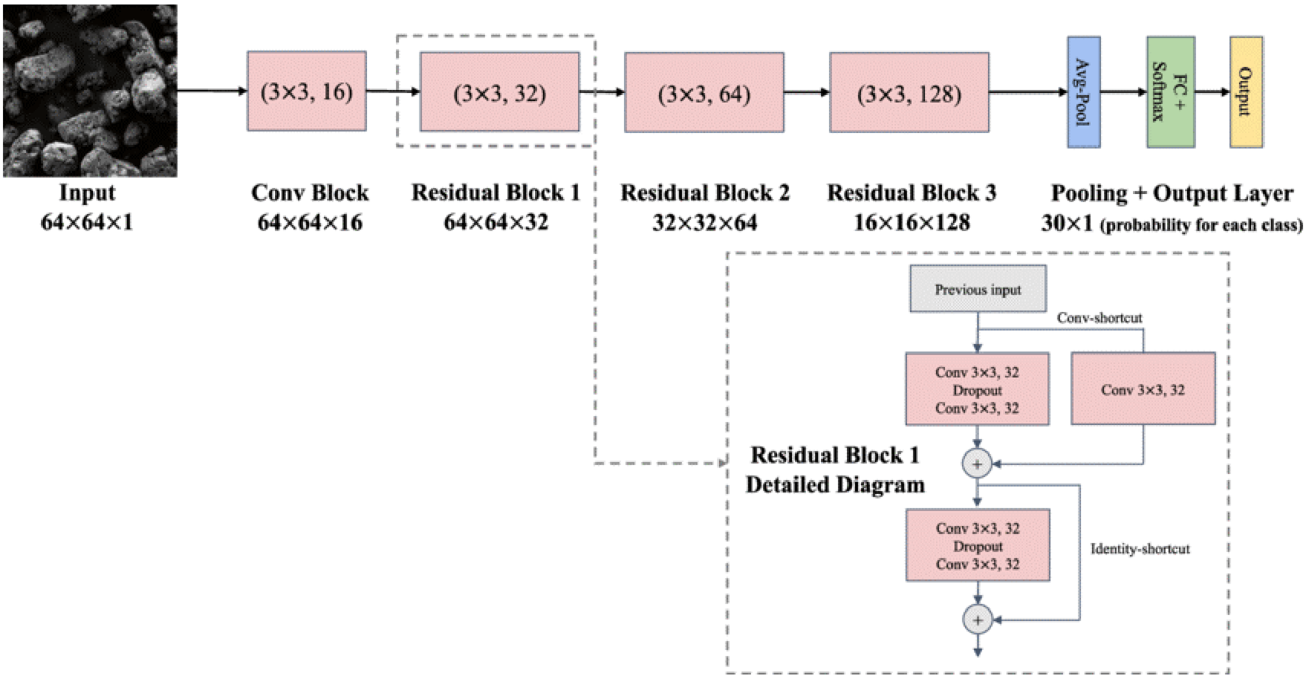


Figure 2.3 Wide ResNet architecture with a depth of 16 and a width of 2

A new end-to-end CNN technique has been reported in reference [12] for determining age and gender from unfiltered real-world faces. This technique can do this by analyzing facial features. In order to evaluate the effectiveness of a two-level CNN architecture, the OIU-Adience dataset is utilized. The method that was developed was successful in achieving an accurate accuracy of 83.1% for age prediction and 96.2% for gender recognition.

In the method proposed by [13], a CNN network is utilized. This network is comprised of three convolutional layers, two layers that are entirely interconnected, and a layer that is used for the output. It is accurate to within 0.6170588 of a millimeter, which is rather impressive.

Both the back propagation neural network and the support vector machine were evaluated and contrasted in the research that is presented in reference [14]. On the Adience dataset, the more straightforward SVM model achieved an accuracy of 90%, outperforming the more complex BPNN model, which achieved just 78% accuracy.

A CNN with 3 dense layers and 2 associated layers each with a least number of neurons has been presented in reference [15]. The goal of this network is to limit the likelihood of overfitting by minimizing the number of neurons in each layer. The model attained an accuracy of 96.2% for the prediction of the gender of the subject, and it achieved an accuracy of 57.51% for the assessment of the subject's age.

Methodology	Database	Drawbacks	Accuracy
CNN model with three convolution layers	Adience	Simple model for higher complexity	Gender = 96.2% Age = 57.51%
Back propagation NN (BPNN) and SVM	Adience	Accuracy is still lagging for BPNN model	BPNN = 78% SVM= 90%
First trained with pre-trained model and then with 4 different stages with learning rates in a schedule.	IMDB-WIKI	Abnormal validation losses for resnet and densenet model, VGG19 classification model losses are in downward trend	Gender = 91.09% MAE (Age) = 15.23
CNN with 3 different architecture	IMDB-WIKI	Only accuracy as metrics and broad classification for age	Gender = 94.49% Age = 86.20%
Wide ResNet 16-8	CCTV video	Accuracy is too low and only accuracy metrics, less data	Gender = 82.92% Age = 70.80%
Two-level CNN architecture	OIU-Adience	Low accuracy for the task of age classification	Age = 83.1% Gender = 96.2%
CNN having 3 convolution layers and 2 dense layers	UTKFace	Face recognition was not accurate enough due to variation in facial orientation	Age = 61.7%
Combined attention LSTM units with RoR models to build AL-ResNet	Adience, MORPH FGNET	Low age group classification accuracy	Age: Adience = 87.83% MORPH (MAE) = 2.36
3 convolution layer CNN model	Online dataset	Less precise architecture and inefficient algorithm for face detection (HAAR cascading)	Gender = 50.7% Age = 79.3%
GoogleNet pretrained model	Adience	Correlation not defined between accuracy and improvement	Gender = 78.10% Age = 38.50%

Methodology	Database	Drawbacks	Accuracy
A modified MobileNet	UTKFace	Age recognition task produced more errors (higher MAE)	Gender = 94.1% MAE (Age) = 5.44
Transfer learning is used with VGG19 and VGGFACE models	MORPH- II	Minimal changes in the face reduces the orientation of face which increases the MAE score.	Gender = 98.7% MAE (Age) = 4.1
Light Weight Multi-Task Convolutional Neural Network (LMTCNN)	Adience	Accuracy of age detection is low and does not work well with huge datasets	Gender = 85.16% Age = 44.26%
Multi-Task Convolutional Neural Network (MTCNN) with joint dynamic loss weight adjustment	UTKFace, BEFA challenge	Lower accuracy for age classification. Used limited amount of facial attributes.	Gender (UTKFace) = 98.23% Age (BEFA) = 71.83%
Gabor filter responses are used as the input in a wide CNN based architecture	Adience	Achieved lower accuracy for the age detection task	Gender = 88.9% Age = 61.3%

Table 2.1 Summary of Literature Survey

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

3.1 Architecture Diagram

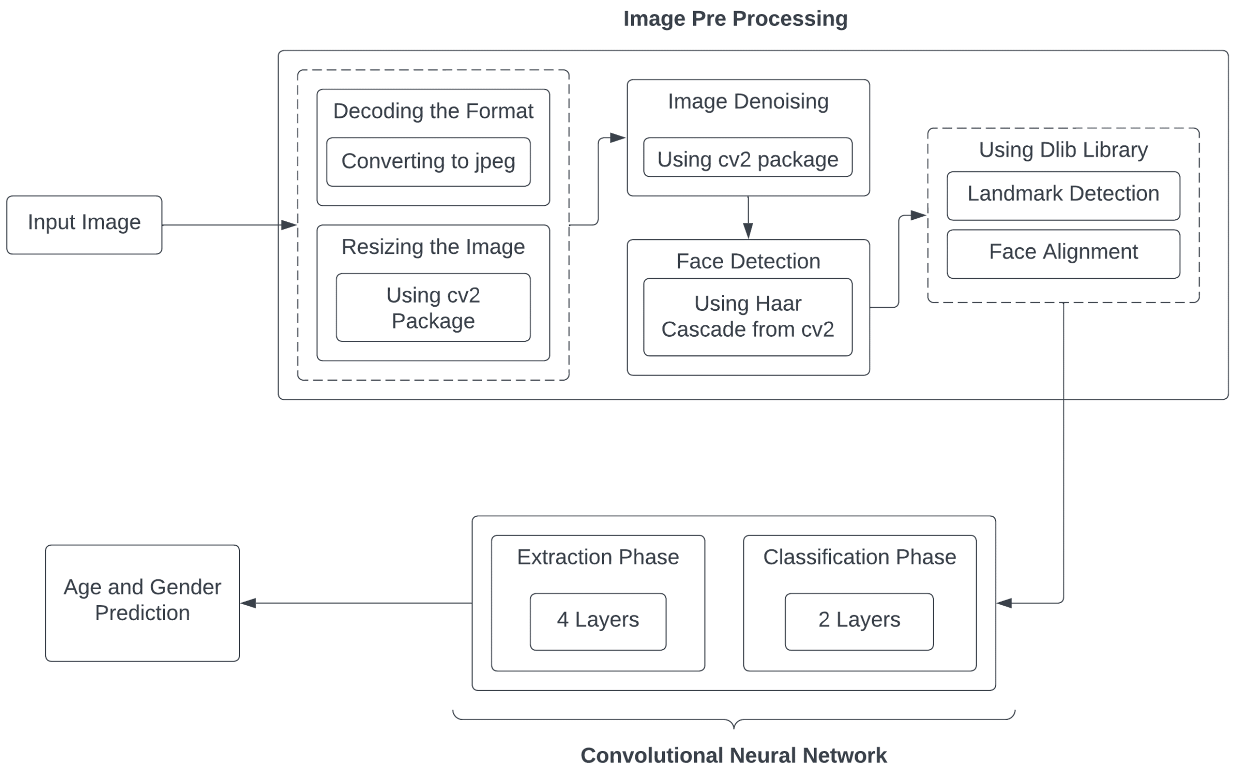


Figure 3.1 Architecture Diagram

- A facial image is first provided as input by either directly uploading an image or detecting the face through live camera.
- The image was subjected to some pre-processing procedures, including resizing, formatting, segmenting, and denoising.

- Every image is resized to a standard size of 227 X 227 (Length x Width).
- Eliminating noise in every image is done using cv2. It uses a special algorithm to accomplish the task (non local means algorithm).
- Using CV2's Haar cascades function, the face is identified and the content is cropped accordingly. The local storage overwrites this image.
- Next, landmark detection is done with a C++ based library called Dlib. The ultimate purpose of this step is to obtain the necessary facial landmarks to align the face accordingly in the next step.
- Now subjecting these landmarks to a few mathematical equations, we get a face aligned image in the end. This process is done using the library called Imutils from cv2.
- In between all the above preprocessing steps, every image is alternatively subjected to gray scale change according the function requirement.
- The stage of CNN called feature extraction begins with the processed image.
- Next, move on to CNN's classification phase to make generalizations and find a solution.
- Now, the output is determined by the task that was completed.

3.2 Class Diagram

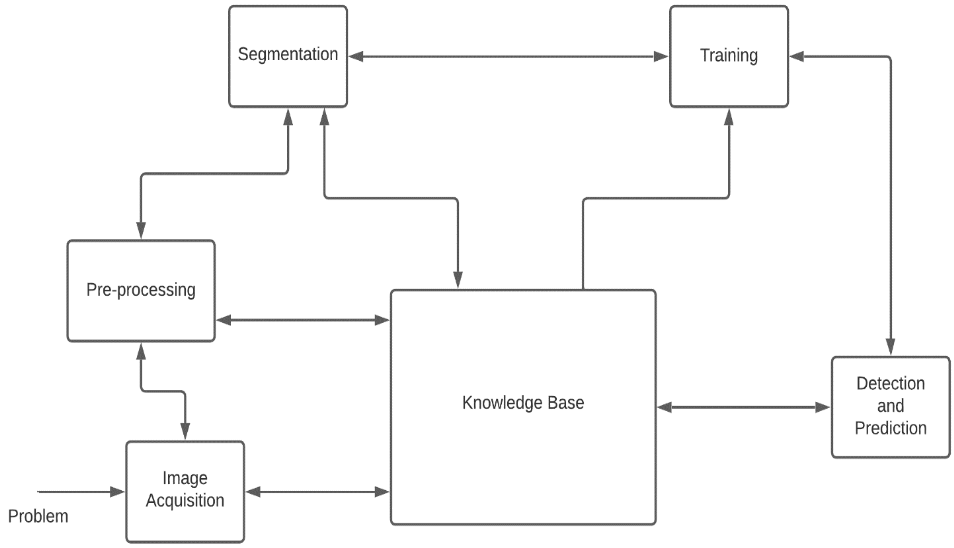


Figure 3.2 Class Diagram

A class diagram is basically a structural diagram which models the static view of an application. Class diagrams play various roles for a software application, starting from visualizing, illustrating and reporting different facets of a system, to constructing executable code for the same. They describe the attributes and functions of a class, and serve as a base for component and deployment diagrams. One of the most important purposes of a class diagram is to portray the responsibilities of a system. As such, it is essentially a collection of classes, interfaces, associations, collaborations, and constraints.

The above diagram clearly explains the association of a central knowledge base with the entire process, starting from image acquisition to training the model and predicting the result. These modules are primarily the classes and interfaces of our architecture system, and the arrows denote the associations between them.

3.3 Use Case Diagram

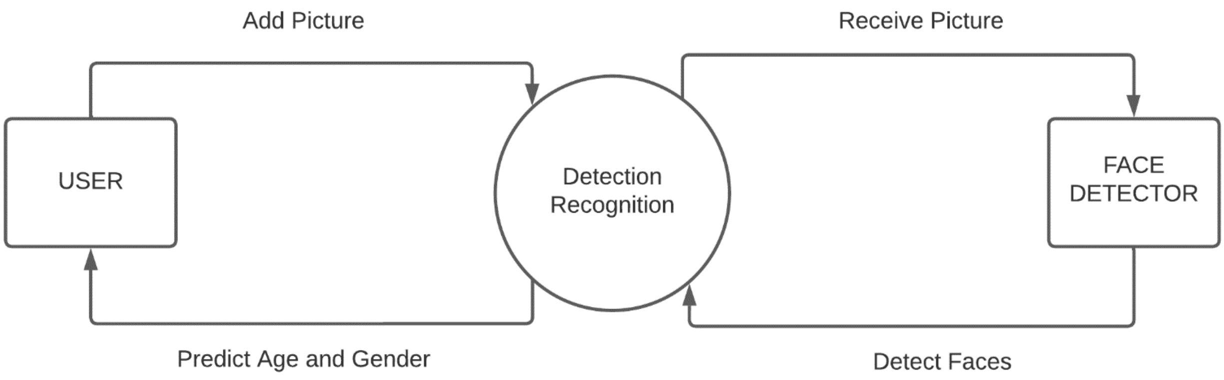


Figure 3.3 Use Case Diagram

Contrary to the class diagram, a use case diagram depicts the dynamic aspect of a system. Its common components include actors, system and goals. These building blocks come together to narrate the system's functionality and the goals of system-user interactions. A use case diagram models the tasks, operations and services needed by an application system. It is ideal for defining the requirements of the system and modelling the basic event flow in a use case.

In the above use case diagram, the user is an actor, and the face detector is the application. The processes in the central component are the end actions or goals of the entire system. The arrows denote the steps needed to reach the objective. The diagram principally illustrates the interaction between the user and the face detector application to arrive at the required goal.

3.4 Model Flowchart

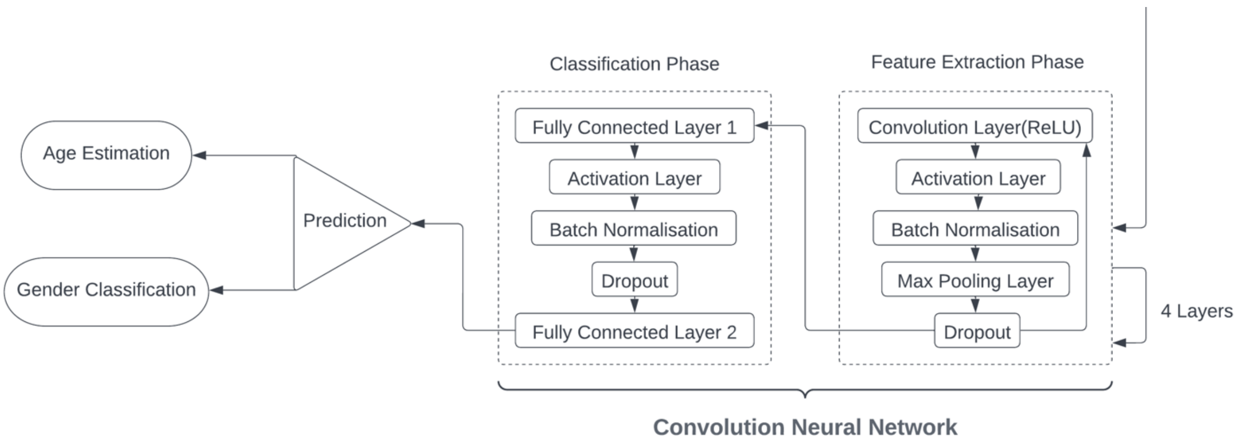


Figure 3.4 Flowchart of the Model

- The extraction phase and the classification phase are two distinct processes that are clearly shown in the model architecture diagram above.
- The extraction Phase extracts as many features as it can from the images in order to create a sufficient pattern. This is done by repeating a sequential set of layers four times with different values for the filter size, kernel size, and stride.
- Two fully connected layers are set up in the classification phase to generalize the result with the aid of all regularization and activation functions.
- Finally, the outcome is produced based on the task it is put through.

CHAPTER 4

METHODOLOGY

The recommended deeply learned network for the task of age as well as gender prediction of real time images. Before the face images are entered into the recommended network, our technique includes an image preparation step (facial identification, landmark detection, and face alignment). Thus, picture pre-processing, feature learning, and classification itself are the three key parts of our method.

4.1 Environment Setup

In this module, we installed and imported a number of different libraries, including Tensorflow, Numpy, PIL, and OpenCV, all of which will be of use to us during various stages of the development of the project. Tensorflow was utilized in particular for the building of deep learning models, while the remaining frameworks were applied at various phases of the development cycle.

4.2 Datasets Used in the Proposed Work

4.2.1 Adience Dataset

The Adience dataset, containing 26,580 photos across 2,284 subjects, is as close as it can get to the real-world face imaging conditions. It comes with a binary gender label and eight distinct age groups, sectioned into five splits. The images were crawled from Flickr albums which were uploaded from smartphones without any filtering. As a result, the dataset has a diverse collection of faces having variations in all aspects such as pose, appearance, lighting, image quality and so on. For the age group and gender classification, Adience dataset was employed to train our network.

Figure 4.1 Adience Dataset

Pre-processing of Adience Dataset

There is a huge necessity to pre-process the unfiltered real-world images so that our proposed network can tackle the classification task successfully. The majority of those facial images are non-frontal, nonaligned, and have different degrees of lighting, stance and surrounding conditions. As such, in order to get a better performance, the raw images need to be detected and aligned, and then fed into the model.

4.2.2 IMDB WIKI Dataset

The IMDB-WIKI dataset has a huge collection of images for training a network for age and gender prediction. To make up this dataset, an IMDB list of 100,000 most popular actors was used and their demographic data. Along with that, profile pictures of various personalities were also taken. To refine it, the pictures lacking timestamps were removed from the dataset. In total 523,051 facial images were obtained, that is, 4.6 lakh pictures approx. from 20 thousand and 60 thousands personalities from IMDb and Wikipedia respectively.

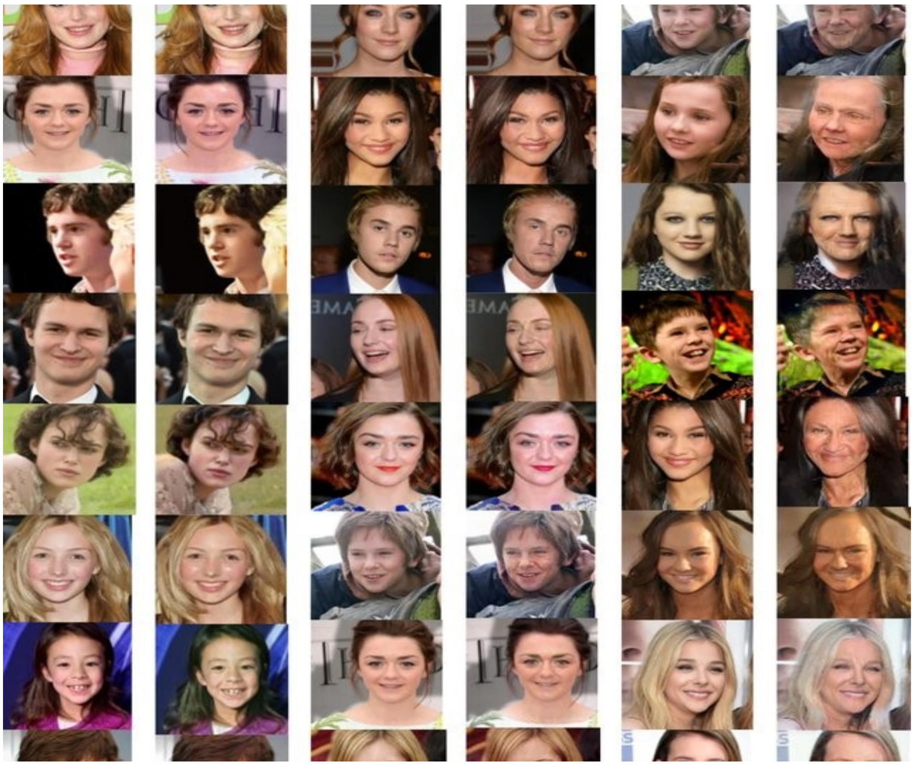


Figure 4.2 IMDB WIKI Dataset

Pre-processing of IMDB WIKI Dataset

Being a huge open source dataset of facial images, it has several issues that are to be handled before passing in to the model. The original dataset has 500 thousand+ images in jpg format with all the required meta data. There are issues like all the images are of different sizes, many of them are corrupted due to improper extraction, some of them are not facial images, some of their meta data has typographical errors such as age mismatch and so on, there are images with multiple faces too and the dataset is imbalance with respect to gender (male count is higher than the female count). Out of all problems, the major one is that it has several grey scale images which are not suitable for the model input. After resolving all of the above issues, the dataset size has reduced till 200 thousand plus images.

4.3 Image Denoising

There is a huge necessity to pre-process the unfiltered real- world images so that our proposed network can tackle the classification task successfully. While working on image processing and computer vision, denoising images is one of the fundamental problems that is to be resolved. As such, in order to estimate the original image, noise has to be suppressed from a noise-contaminated version of the image, the process of which is known as image denoising.

Cv2 library uses non local means algorithm to accomplish this task. The non local means algorithm's core idea is to replace a pixel's color by averaging the colours of several image sub- windows that are comparable to the one that makes up the pixel neighbourhood.

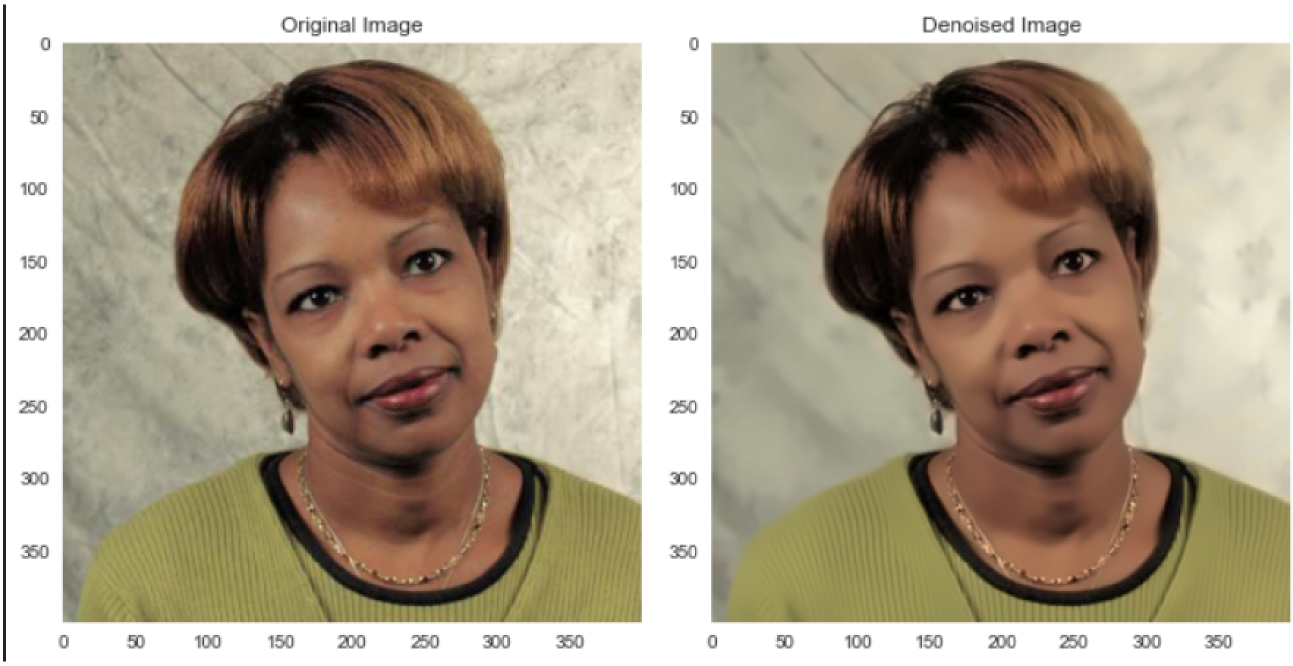


Figure 4.3 A denoised image from an original image

4.4 Face Detection

Face detection is the initial step in image pre-processing. We have made use of Haar Cascades, an open-source face detector. All input photos are rotated between -90° and 90° angles with a 5° step in order to detect faces. Next, the processed image which is the greatest is selected by the detector while if a case arise where no face is segmented even after the modifications, the given image is upscaled and passed on to the next phase till a face is found. This processing step helps in detecting the face in all the photos.

Figure 4.4 An image of a face detected from a denoised image

4.5 Landmark Detection

During this phase of the process, the landmarks (important spots) that are present on the face are identified and monitored. The system makes use of five different landmark detection models, each of which has been trained to recognise the many facial positions that might be connected with a person's face. Several models exist like two full-profile models, profile models etc. Here are some advantages and disadvantages of landmark detection.

Advantages:

- Improved accuracy: Landmark detection has the tendency of outputting precise facial references, which will improve the accuracy of major computer vision tasks such as image segmentation and face detection.
- Robustness to image variations: The dataset that we use in this project is huge and has a variety of images captured under different conditions. Detecting landmarks in a facial image can be robust to variations in orientation and lighting conditions.

Disadvantages:

- Requires large amount of data with labels: Huge amount data with labels are necessary for landmark detection.
- Sensitive to noise: Landmark detection can be sensitive to noise and occlusions in the image, which can affect the accuracy of the detected landmarks.

The above drawbacks of landmark detection are tackled. Denoising is done already and we are using a huge dataset (IMDB-WIKI).

4.6 Face Alignment

Furthermore, the angle of face phase consists of applying all the above models to the segmented faces. The model having the greatest confidence score is then transformed using an affine function to the predetermined ideal settings for those landmarks.

Figure 4.5 An image of an aligned face from an input image

4.7 CNN Framework

Our CNN architecture is a six-layer network that consists of two fully connected layers and four convolutional layers. A convolutional neural network is being employed for this task which includes extraction of features as well as predicting labels. The network contains various convolutional layers (4) with different number of filters, kernels and strides. The other aspects of this network includes batch normalisation, activation functions and pooling layers. The final stage of this network includes two layers which are fully connected to predict the age and gender of the input image.

The architecture we finally used for this problem statement consists of six-layers. The 4 convolutional layers and final dense layers not only extract features but also perform classification. Various others entities like filters, stride length, activation layers and pooling layers.

To be more specific the first dense layer of the age classification task, we have used an L2 regularisation factor of 0.01 to decay the weights and avoid the process from falling into a local minima.

Both tasks are executed with different learning rates to balance the compatibility between the model complexity and the size of the dataset used. The values for the same are 0.01 and 0.001 for gender and age, respectively. Since the gender task is a binary classification, starting with a momentum value of 0.9 was required to avoid learning too much from the training data.

Finally, for both the tasks, the stochastic gradient descent (SGD) algorithm is used as an optimizer, which has more popularity and is faster than any other optimizers.

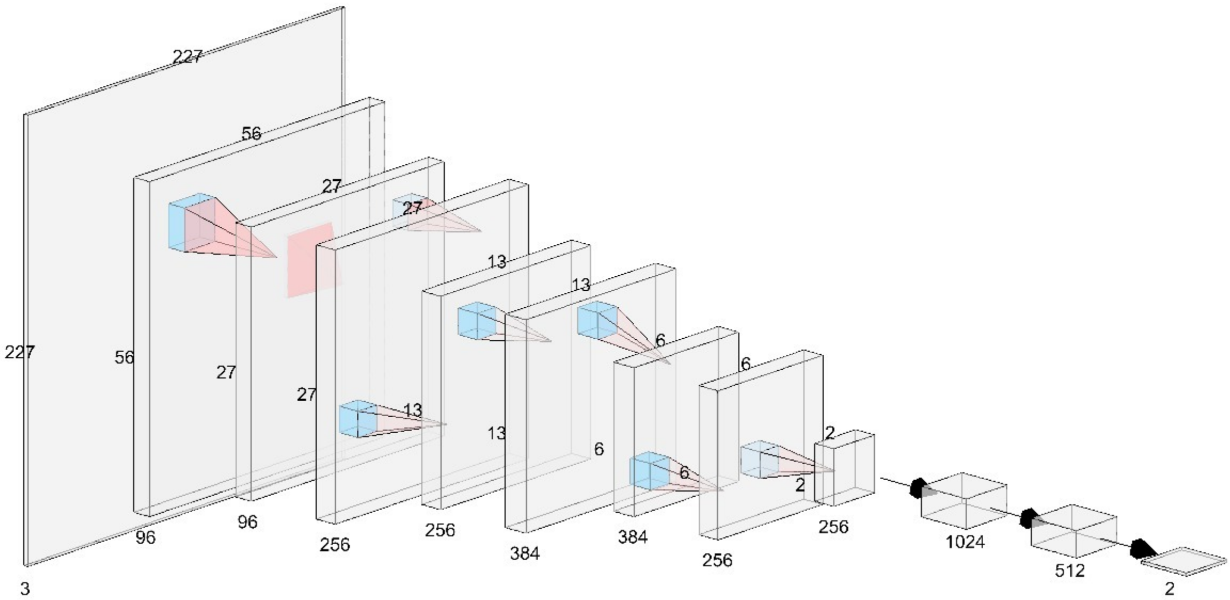


Figure 4.6 A Detailed Figure of the CNN Framework

CHAPTER 5

CODING AND TESTING

5.1 Classification Using Convolutional Neural Network

Convolutional Neural Networks, more commonly referred to as CNNs, are a type of Deep Learning network that have the ability to learn from their experiences. This is a specialized neural network that has the capability of learning spatial information, which enables them to handle photos, videos, and other issues related to computer vision very well. It is possible for convolutional neural networks to extract features from an image or video and integrate them in such a way that essential information is preserved when doing so. CNNs come in a variety of forms, and they each serve a specific purpose in the realm of vision-related work. These networks may be joined together in a number of different ways and then utilised in information- gathering pooling layers.

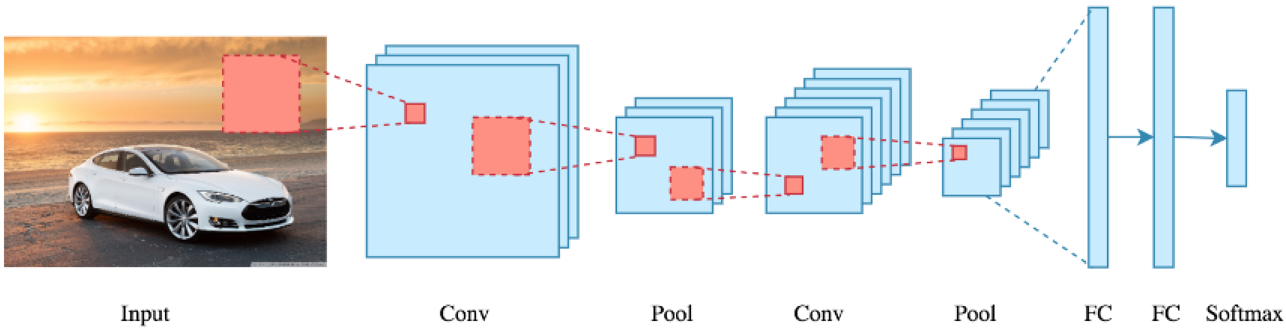


Figure 5.1 Illustration of neural network

These days, there is a wealth of digital material available, particularly in the form of images. A number of CNNs have been trained to classify a wide variety of real-world things by making use of several million training instances. These trained networks are potentially useful for a variety of additional applications.

CNNs have the potential to learn the relevant features of the task given enough data, which eliminates the need for hand-engineered filters for specific tasks. This ability is only possible if the data is large enough. During vision tasks, these are preferred to feedforward neural networks due to the reduction in the number of parameters required.

The Pooling layer, like the Convolutional Layer, is in charge of shrinking the Convolved Feature's spatial size. By lowering the dimensions, this will lower the amount of CPU power needed to process the data. Average pooling and maximum pooling are the two types of pooling. I've just used Max Pooling once, but so far I haven't encountered any issues.

A convolution neural network (CNN) is a type of deep feed-forward neural network that is used to process data with spatial and temporal dependencies. They are most commonly used for image recognition, classification and detection tasks. Convolution is the mathematical function, a special form of a linear operation, which produces a function as a complex product of two different functions, that in-turn determines how the shape of one function is modified by the other. In terms of image processing, two images (in the form of tensors) are multiplied, resulting in an output that is used to extract the features of the image. The various layers of a CNN are:

- **Convolution Layer**: This layer performs the convolution operation between the input images and a kernel of size defined in the layer, resulting in the extracted features. It usually forms the first layer of any CNN, following the input layer that defines the resolution of the input images. The kernel sizes usually range from 1x1, 3x3, 5x5, 7x7, etc.

- **Pooling Layer**: This layer decreases the number of features in the output of the convolution layer by decreasing the number of connections between the layers and independently computing each feature map. The two most commonly used pooling techniques are discussed later in this section.

- **Fully Connected Layer**: It consists of the flatten and the dense layers, where the output of the convolution and pooling layers are flattened and passed through several dense layers where the linear/non-linear operations take place. They contain weights and biases which are tuned during training. This usually forms the final classification part of a CNN.

- **Dropout layers**: These layers are used to induce dropouts by randomly silencing a selected number of neurons at a time during training. This method helps in reducing overfitting by decreasing the size and complexity of the model. For eg. A dropout value of 0.25 results in 25% of the neurons to be randomly "dropped" during training.

Pooling is used to reduce the dimensions of the features produced by the network. By reducing the dimensions and the features, it reduces the number of parameters to be learnt by the model during training, thereby reducing the computation cost of training the network, and increases its efficiency.

There are two types of pooling that can be performed on the output of features:

- **Max Pooling**: in this type of pooling, it returns the maximum value of the pixels that are covered by the kernel. Since the maximum value among a set of pixels is towards the lighter end of the colour spectrum, Max Pooling selects the brighter and lighter- coloured pixels of the image. This can be used for tasks where the background of the object to be segmented is dark or if this object needs to be distinctively identified.

- **Average Pooling**: in this type of pooling, it computes and returns the average of the pixel values that are covered by the kernel. In general, Average Pooling smoothens out the image, and prevents the sharp features of the images such as edges, lines, and grains from being retained.

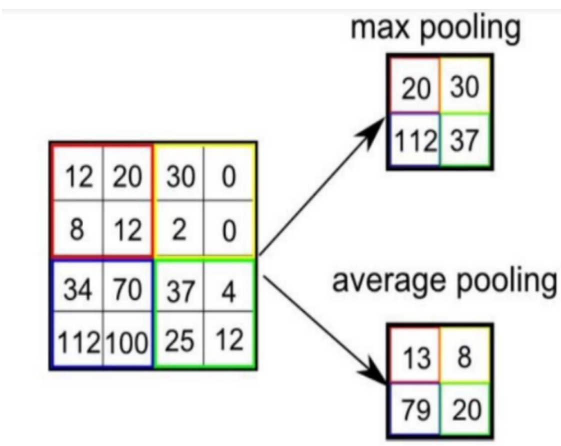


Figure 5.2 Max Pooling and Average Pooling

An activation function is a non-linear function performed on the weighed and biased inputs of a perceptron, to obtain the output to be sent to the next layer. Non-linearity helps the model learn better by learning complex correlations and hidden features of the data and deviating from the ideal conditions and attributes. The two types of activation functions recommended for use in a CNN are:

- **Sigmoid**: Also known as the logistic activation function, it returns values between 0 and 1. The function, being smoother, allows for ease of classifying elements of a binary classification problem. However, a high negative input produces a value close to 0, thereby reducing the speed with which weights are updated during back-propagation.

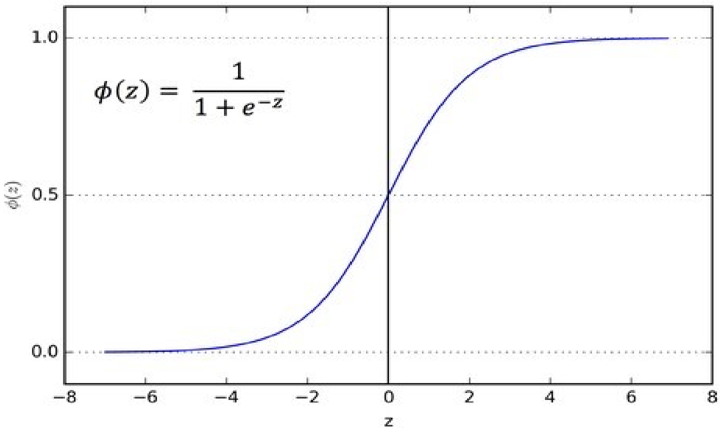
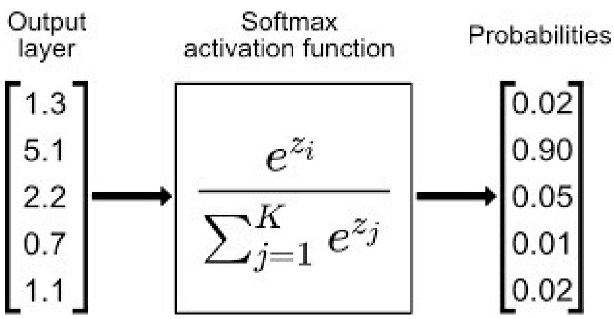


Figure 5.3 Sigmoid graph

- **Softmax**: This activation function, usually used in the final layer of neural networks designed for classification, returns values between 0 and 1 (the probabilities of their occurrence in different classes), where all the inputs sum up to 1.



$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (5.1)$$

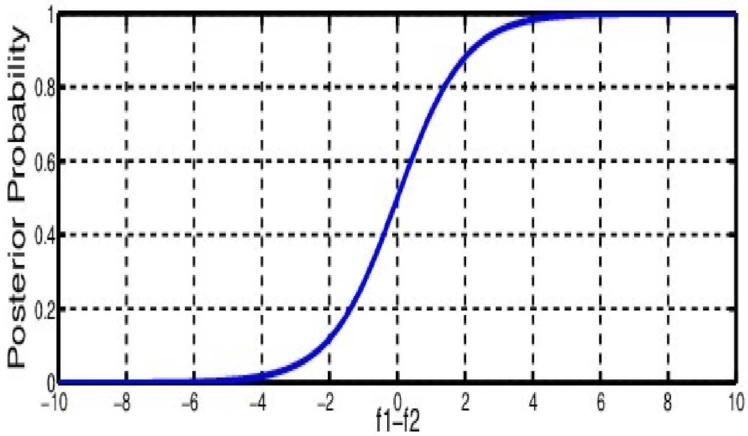


Figure 5.4 Softmax graph

Bias and variance are prediction errors that occur while training a machine learning model. Bias can be defined as the difference between the average prediction of the model and the ground truth values. Whereas variance is the model's sensitivity to and variation in predictions for different portions of the training dataset. There is a trade-off between these two parameters when the model is required to simultaneously minimize both.

A model is said to be a good fit when it learns most of the patterns present in the dataset but not to the extent of including outliers and random data points, and hence is able to generalize efficiently by producing the right predictions for new unseen data (Figure 5.5).

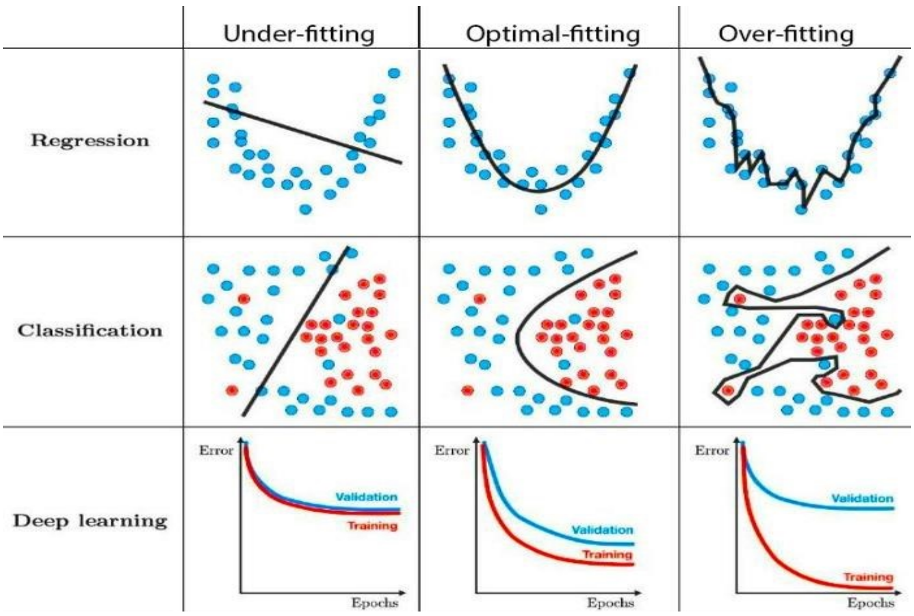


Figure 5.5 Curves of predictions and true values for different classes of model performance

When a model is unable to learn the inherent patterns and correlations between the variables of a dataset, it is said to underfit. In this case, the model is not efficient and has to be enhanced or replaced with a more complex model that is appropriate for the task. On the other hand, allowing a model to learn the data to an extent where it learns a large number of patterns pertaining only to the training data, including erroneous (noise) and rare occurrences of data points, will cause the model to overfit. Such models perform exceptionally on the training set but fail to generalize and hence under-perform on the test and validation datasets. An underfitting model is said to have high bias and low variance while an overfitting model is said to have high variance and low bias.

Regularization is a technique used to add a penalty to the model as its complexity increases, in order to penalize the parameters, thereby helping it generalize and prevent it from overfitting. The two types of regularization methods used in the dense layers of a CNN are:

- **L1 regularization**: Also called as Lasso (Least Absolute Shrinkage and Selection Operator) regression, adds the absolute value of the magnitude of the coefficient as the penalty term to the loss function. This type of regularization shrinks the less important feature's coefficients to zero, thus acting as an implicit feature selection algorithm. The modified loss function is given as:

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \quad \ldots\ldots\ldots\ldots\ldots \quad (5.2)$$

- **L2 regularization**: Also called as Ridge regression, which adds the squared magnitude of the coefficient as the penalty term to the loss function. This type of regularization helps prevent overfitting of the model. The higher the value of the penalty, the greater is the decrease in magnitude of the coefficients. It thus reduces the complexity of the model by shrinking the coefficients. The loss function is given as:

$$\sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \quad \ldots\ldots\ldots\ldots\ldots \quad (5.3)$$

where in both the equations, the penalty term is denoted by λ and the first component depicts the loss (sum of squared residuals).

5.1.1 Advantages of CNN

The CNNs have several advantages compared to other types of neural networks. Efficient computation and the ability to learn important features without human aid are prominent advantages of neural networks. They reduce the number of learnable parameters drastically compared to feedforward neural networks and capture vital information on their own using large amounts of data. Furthermore, the most important of all is the ability to use transfer learning. Popular networks which were trained using several million images for some other tasks can also perform reasonably well in other problems from completely different domains. These networks are also extensively available to the public. Few of the advantages of transfer learning are:

- Improved Performance: Significant performance than training from scratch.
- Reduced Training Time: The model starts with better weights compared to random weights and therefore, reduced the time to train.
- Require Fewer Data: Requires less amount of data samples to generalize.
- Generalization: The pre-trained model has learned a set of generic features that can be useful across multiple tasks and domains.
- Better Optimization: Deals with the problem of overfitting.

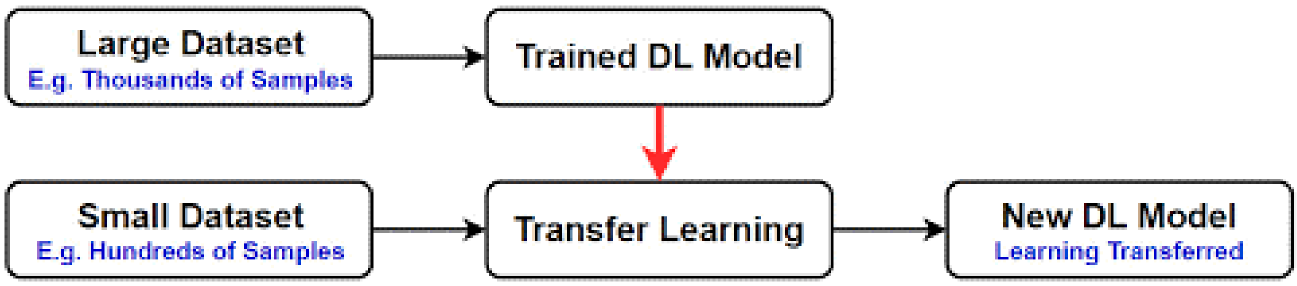


Figure 5.6 Transfer Learning

While transfer learning has a wider variety of advantages, they are always not the best option. Therefore, it is important to understand where they fall short and choose according to their pros and cons. The disadvantages of transfer learning are:

- Limited Applicability: Significant performance than training from scratch.
- Model Bias: The model starts with better weights compared to random weights and therefore, reduced the time to train.
- Domain Mismatch: Requires less amount of data samples to generalize.
- May not learn full complexity of the problem.

5.2 Pre-trained Convolutional Neural Networks

VGG: The VGG network was named after a group of researchers who worked at the University of Oxford. The network consists of 16 layers of convolutional and fully connected layers, with a total of 138 million parameters. The network was trained on the ImageNet dataset and achieved state-of-the-art performance at the time, with a top-5 error rate of 7.3%. VGG-16's simplicity and uniformity are the key features of the network, with all convolutional layers having a 3x3 kernel size and all max pooling layers having a 2x2 kernel size. Its architecture has since been widely used as a starting point for many computer vision tasks, including object detection, segmentation, and transfer learning.th the datasets were extracted for usage in various field, from which using them for machine learning is also a purpose. Hence, it is evident that these datasets are not readily accessible for immediate usage in machine learning pipelines. The datasets has to undergo a thorough process of pre-processing to prevent the model from driving towards any unintended directions.

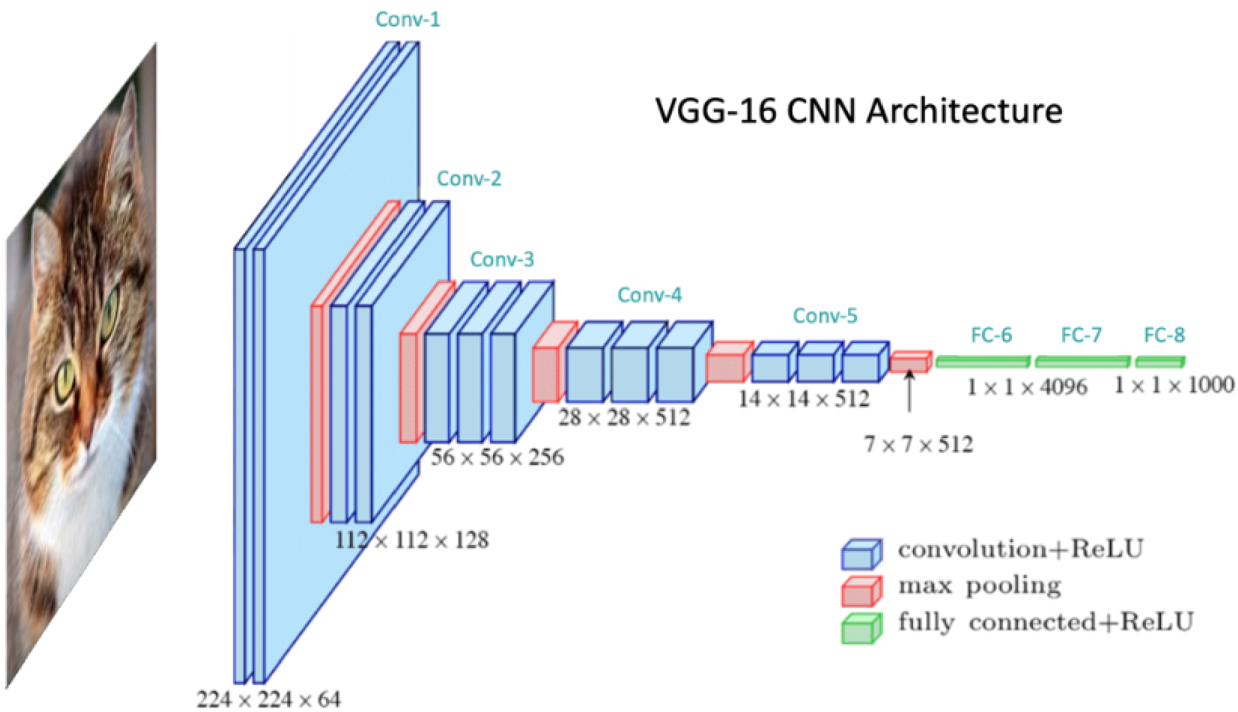


Figure 5.7 VGG Architecture

ResNet: The ResNet architecture is widely regarded as one of the most popular Convolutional Neural Network architectures. It was one of the networks which changed the face of deep learning, especially computer vision. It totally consists of 5 parts, where a convolutional and identity block represents each part. Each convolutional part has 3 convolutional layers, as do the identity blocks. Over 23 million trainable parameters are available in the ResNet-50. ResNet50's main features are the previously explained blocks. The standard block in ResNet is the Identity block which has the same input and output dimensions. A convolutional block is used when the dimension does not match. The identity block differs from the shortcut path. The "skip connection" used in ResNet allows the gradient to back propagate directly to the initial layers, allowing information to transfer between various layers rather than consecutive layers, allowing it to use both low-level and high-level features while making a prediction.

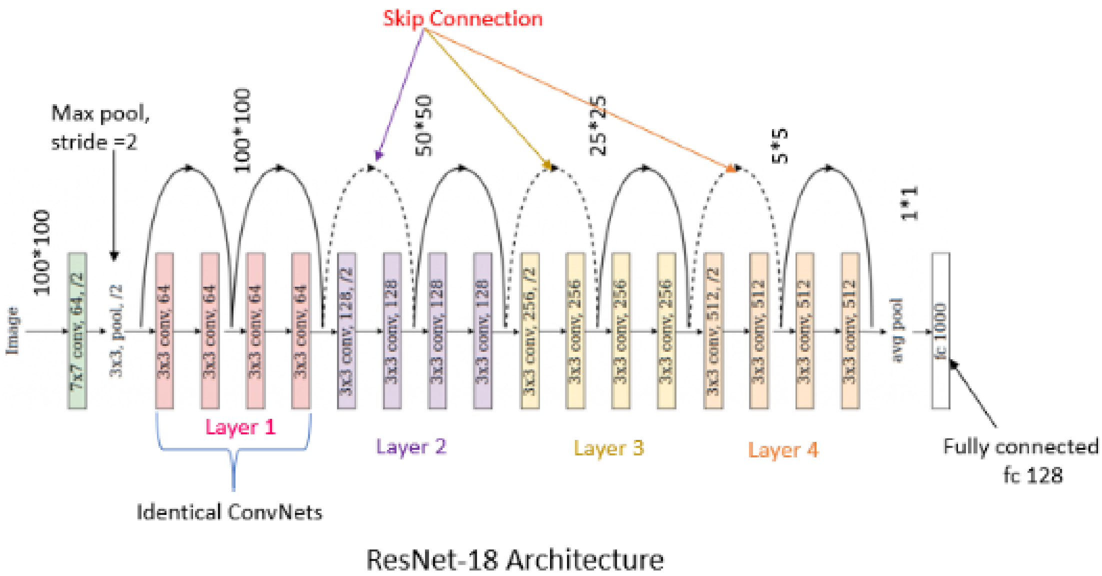


Figure 5.8 ResNet Architecture

5.3 Pre-processing the Datasets

Both the datasets were extracted for usage in various field, from which using them for machine learning is also a purpose. Hence, it is evident that these datasets are not readily accessible for immediate usage in machine learning pipelines. The datasets has to undergo a thorough process of pre-processing to prevent the model from driving towards any unintended directions.

```
final_imdb = np.vstack((imdb_age, imdb_genders, imdb_path, imdb_face_score1, imdb_face_score2)).T
final_wiki = np.vstack((wiki_age, wiki_genders, wiki_path, wiki_face_score1, wiki_face_score2)).T

final_imdb_df = pd.DataFrame(final_imdb)
final_wiki_df = pd.DataFrame(final_wiki)

final_imdb_df.columns = cols
final_wiki_df.columns = cols

meta = pd.concat((final_imdb_df, final_wiki_df))

meta = meta[meta['face_score1'] != '-inf']
meta = meta[meta['face_score2'] == 'nan']

meta = meta.drop(['face_score1', 'face_score2'], axis=1)

meta = meta.sample(frac=1)

meta.to_csv('meta.csv', index=False)

path = r'E:\age and gender prediction\project\meta.csv'
data = pd.read_csv(path)
```

Figure 5.9 Eliminating Corrupted Images

```python
def isgray(imgpath):
    img = cv2.imread(imgpath)
    if len(img.shape) < 3: return True
    if img.shape[2] == 1: return True
    b,g,r = img[:,:,0], img[:,:,1], img[:,:,2]
    if (b==g).all() and (b==r).all(): return True
    return False

gray_index_lst = []
for index,i in zip(data.index,data['path']):
    if isgray('/age and gender prediction/project/data/'+ i) == True:
        gray_index_lst.append(index)

# write list to binary file
def write_list(a_list):
    # store list in binary file so 'wb' mode
    with open('IMDB_WIKI_gray_img_list', 'wb') as fp:
        pickle.dump(a_list, fp)
        print('Done writing list into a binary file')

# Read list to memory
def read_list():
    # for reading also binary mode is important
    with open('IMDB_WIKI_gray_img_list', 'rb') as fp:
        n_list = pickle.load(fp)
        return n_list

write_list(gray_index_lst)
gray_img_index = read_list()

dataset = data.copy()
dataset.drop(gray_index_lst, axis = 0, inplace =True)
dataset.reset_index(drop=True, inplace =True)
```

Figure 5.10 Eliminating Grayscale Images

5.4 Gender Prediction

The learning rate started with 0.01 while the L2 weight decay was 0.005 and momentum term to 0.9 while training the neural network. We also went the old fashioned Stochastic Gradient Descent optimizer.

Face Detection, Face Alignment, Landmark Detection, and Image Denoising

```
def yield_training_values(X_train,y_train):
    detector = dlib.get_frontal_face_detector()
    predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
    fa = FaceAligner(predictor, desiredFaceWidth=227)
    for image_path, value in zip(X_train, y_train):
        imageP = image_path[0].decode("utf-8")
        img= cv2.imread(imageP, 1)
        denoised_image = cv2.fastNlMeansDenoisingColored(img, None, 5, 6, 7, 21)

        gray = cv2.cvtColor(denoised_image, cv2.COLOR_BGR2GRAY)
        # Detect the face
        rects = detector(gray, 1)
        # Detect landmarks for each face
        try:
            for rect in rects:
                faceAligned = fa.align(img, gray, rect)

            gray1 = cv2.cvtColor(faceAligned, cv2.COLOR_BGR2GRAY)
            face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
            faces = face_cascade.detectMultiScale(gray1, 1.3, 5)

            try:
                for (x,y,w,h) in faces:
                    # to put rectangle on face
                    #cv2.rectangle(faceAligned, (x,y), (x+w, y+h), (0, 255, 0),3)
                    roi_color = faceAligned[y:y+h, x:x+w]
                    cv2.imwrite(imageP , roi_color)
            except:
                continue
        except:
            continue

        image = preprocess_image([bytes(imageP, 'utf-8')])
        yield image, value
```

Figure 5.11 Gender Image Pre Processing

Model Building

```
model = keras.models.Sequential([
    data_augmentation,
    keras.layers.Conv2D(filters=96, kernel_size=(7,7), strides=(4,4), activation='relu', input_shape=(227,227,3)),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    keras.layers.Conv2D(filters=256, kernel_size=(5,5), strides=(1,1), activation='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), activation='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=((2,2))),
    keras.layers.Conv2D(filters=256, kernel_size=(3,3), strides=(1,1), activation='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=((2,2))),
    keras.layers.Flatten(),
    #dense change from 4096
    keras.layers.Dense(512, activation='relu'),
    keras.layers.Dropout(0.5),
    #dense change from 4096
    keras.layers.Dense(512, activation='relu'),
    keras.layers.Dropout(0.5),
    #activation change from softmax
    #dense change from 2
    keras.layers.Dense(1, activation='sigmoid')
])
```

Figure 5.12 Gender Model Architecture

```
1  model.summary()

Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (None, 227, 227, 3)       0

 conv2d (Conv2D)             (None, 56, 56, 96)        14208

 batch_normalization (BatchN  (None, 56, 56, 96)       384
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 27, 27, 96)       0
 )

 conv2d_1 (Conv2D)           (None, 27, 27, 256)       614656

 batch_normalization_1 (Batc  (None, 27, 27, 256)      1024
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 13, 13, 256)      0
 2D)

 conv2d_2 (Conv2D)           (None, 13, 13, 384)       885120

 batch_normalization_2 (Batc  (None, 13, 13, 384)      1536
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 6, 6, 384)        0
 2D)

 conv2d_3 (Conv2D)           (None, 6, 6, 256)         884992

 batch_normalization_3 (Batc  (None, 6, 6, 256)        1024
 hNormalization)

 max_pooling2d_3 (MaxPooling  (None, 2, 2, 256)        0
 2D)

 flatten (Flatten)           (None, 1024)              0

 dense (Dense)               (None, 512)               524800

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 512)               262656

 dropout_1 (Dropout)         (None, 512)               0

 dense_2 (Dense)             (None, 4)                 2052

=================================================================
Total params: 3,192,452
Trainable params: 3,190,468
Non-trainable params: 1,984
_________________________________________________________________
```

Figure 5.13 Gender Model Summary

```
Epoch 48/50
438/438 [==============================] - 616s 1s/step - loss: 0.2363 - accuracy: 0.8981 - val_loss: 0.3428 - val_accuracy: 0.8405
Epoch 49/50
438/438 [==============================] - 712s 2s/step - loss: 0.2337 - accuracy: 0.8975 - val_loss: 0.3297 - val_accuracy: 0.8468
Epoch 50/50
438/438 [==============================] - 20081s 46s/step - loss: 0.2284 - accuracy: 0.8990 - val_loss: 0.3698 - val_accuracy: 0.8308
```

Figure 5.14 Accuracy Result of Gender Model

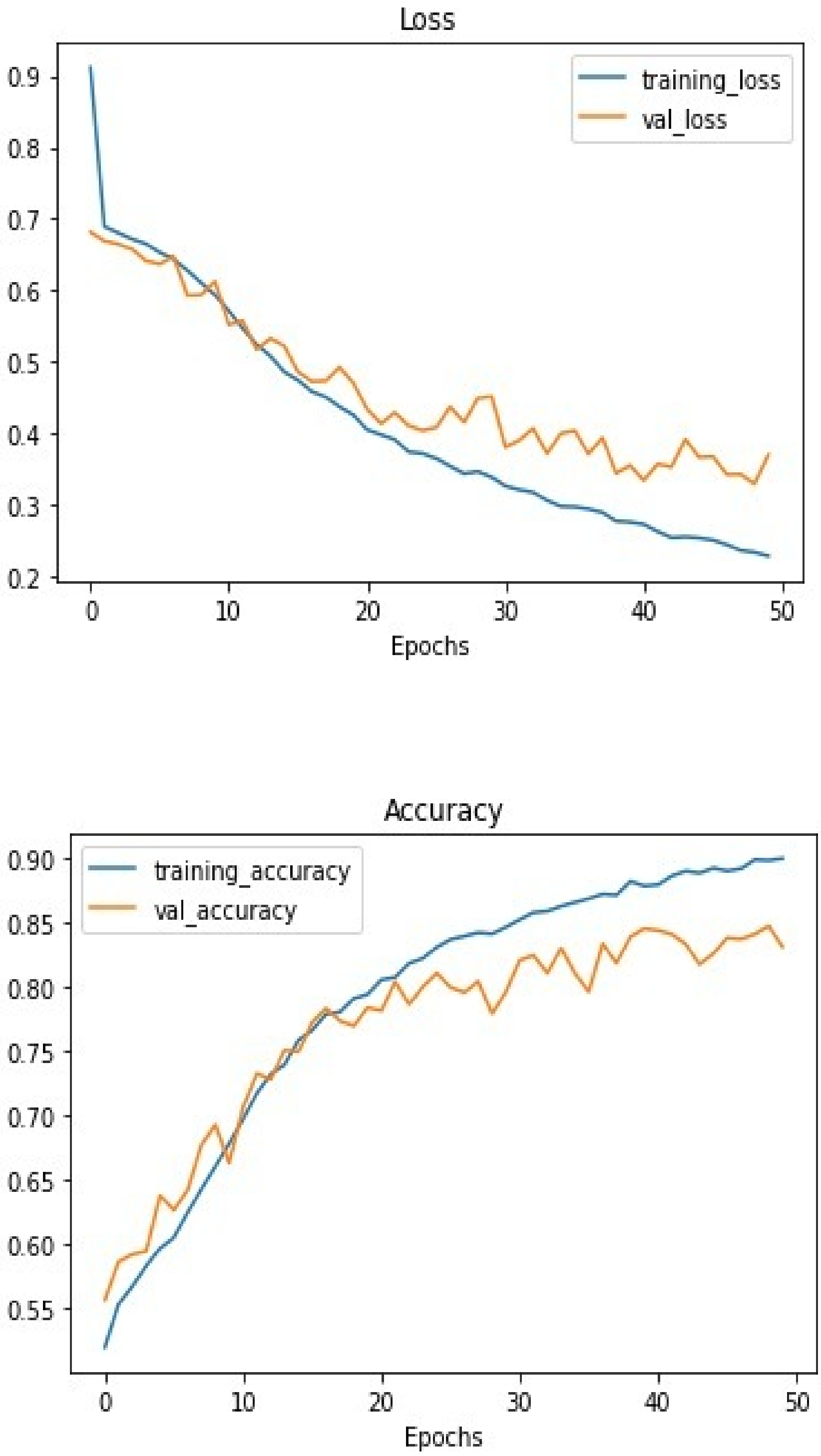


Figure 5.15 Gender: Accuracy and loss graph

5.5 Age Prediction

Age group to prepare using a CNN-based classifier to determine the appropriate age group for unfiltered face photos, we set the initial learning rate to 0.0001 to give the model more time to develop before using an L2 weight decay of 0.0005. We use the Adam optimizer to change the network weights during training in order to make our model capable of generalisation and accurate prediction.

Image Preprocessing ¶

Preparing to split for train and test set

```
X = new_df[['image_path']].values
y = new_df[['age']].values
```

```
set(y.flatten().tolist())
```

```
{0, 1, 2, 3}
```

Train Test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Assigning uniform image extensions and resizing

```
def preprocess_image(individual_path):
    img = tf.io.read_file(np.array(individual_path).ravel()[0])
    img = tf.image.decode_jpeg(img)
    img = tf.image.resize(img, [227,227])
    return img
```

Figure 5.16 Age Image Pre Processing

Model Building

```
model = keras.models.Sequential([
    data_augmentation,
    keras.layers.Conv2D(filters=96, kernel_size=(7,7), strides=(4,4), activation='relu', input_shape=(227,227,3)),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    keras.layers.Conv2D(filters=256, kernel_size=(5,5), strides=(1,1), activation='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=(2,2)),
    keras.layers.Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), activation='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=((2,2))),
    keras.layers.Conv2D(filters=256, kernel_size=(3,3), strides=(1,1), activation='relu', padding="same"),
    keras.layers.BatchNormalization(),
    keras.layers.MaxPool2D(pool_size=(3,3), strides=((2,2))),
    keras.layers.Flatten(),
    keras.layers.Dense(512, activation='relu'),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(512, activation='relu', kernel_regularizer=keras.regularizers.l2(l=0.01)),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(4, activation='softmax')
])
```

```
callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=10)
#adam = tf.keras.optimizers.Adam(learning_rate=0.001)
sgd = SGD(learning_rate=0.001)
model.compile(optimizer=sgd, loss = tf.keras.losses.SparseCategoricalCrossentropy(), metrics='accuracy')
```

Figure 5.17 Age Model Architecture

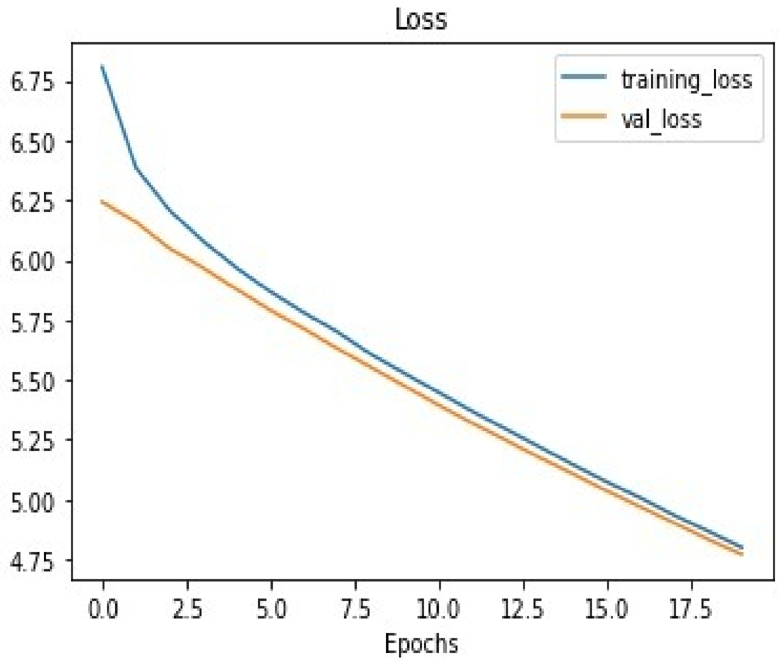
```
1  model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (None, 227, 227, 3)       0

 conv2d (Conv2D)             (None, 56, 56, 96)        14208

 batch_normalization (BatchN  (None, 56, 56, 96)       384
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 27, 27, 96)       0
 )

 conv2d_1 (Conv2D)           (None, 27, 27, 256)       614656

 batch_normalization_1 (Batc  (None, 27, 27, 256)      1024
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 13, 13, 256)      0
 2D)

 conv2d_2 (Conv2D)           (None, 13, 13, 384)       885120

 batch_normalization_2 (Batc  (None, 13, 13, 384)      1536
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 6, 6, 384)        0
 2D)

 conv2d_3 (Conv2D)           (None, 6, 6, 256)         884992

 batch_normalization_3 (Batc  (None, 6, 6, 256)        1024
 hNormalization)

 max_pooling2d_3 (MaxPooling  (None, 2, 2, 256)        0
 2D)

 flatten (Flatten)           (None, 1024)              0

 dense (Dense)               (None, 512)               524800

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 512)               262656

 dropout_1 (Dropout)         (None, 512)               0

 dense_2 (Dense)             (None, 4)                 2052

=================================================================
Total params: 3,192,452
Trainable params: 3,190,468
Non-trainable params: 1,984
_________________________________________________________________
```

Figure 5.18 Age Model Summary

```
447/447 [==============================] - 590s 1s/step - loss: 5.0059 - accuracy: 0.3830 - val_loss: 4.9688 - val_accuracy: 0.3989
Epoch 18/20
447/447 [==============================] - 608s 1s/step - loss: 4.9343 - accuracy: 0.3871 - val_loss: 4.9022 - val_accuracy: 0.3903
Epoch 19/20
447/447 [==============================] - 619s 1s/step - loss: 4.8681 - accuracy: 0.3955 - val_loss: 4.8344 - val_accuracy: 0.4029
Epoch 20/20
447/447 [==============================] - 611s 1s/step - loss: 4.8005 - accuracy: 0.3948 - val_loss: 4.7720 - val_accuracy: 0.3992
```

Figure 5.19 Accuracy Result of Age Model



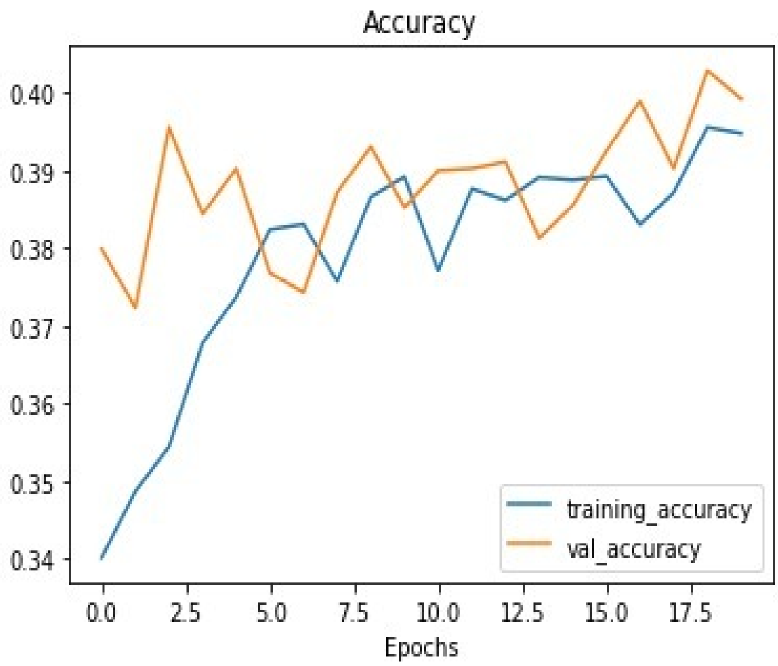


Figure 5.20 Age: Accuracy and loss graph

5.6 User Interface

For any machine learning system, user interface plays a vital role. It enables the user to interact with the system which consists of ML models, data and results. A well-designed UI can make ML more accessible and usable for non-experts, enabling them to easily train, evaluate, and deploy models without requiring deep knowledge of ML algorithms or programming. The underlying data, model architecture, training process, prediction process which is critical for ensuring the working of the model. The web design of ML systems are simple as well as easy to use for a wide variety of people. Few examples of ML systems that have simple UI design are ChatGPT by OpenAI, Segment Anything by Meta AI.

Moreover, a UI aids in collaboration and communication among stakeholder and team members. Tracking and Experimentation will also be easy when a good UI is designed. The usability, accessibility and efficiency of ML systems are enhanced using the UI and it helps users to use them.

The advantages of having a user interface with a machine learning system, including:

- User-Friendliness
- Accessibility
- Visualization
- Control
- Feedback

Finally, a UI will greatly improve the user experience as well as improve the amount of users using the ML Systems.

The below website was implemented to enable users to use the gender age prediction machine learning. The website has a landing page with several options to navigate through the website. The navigation bar has links to home, services, about, contact us and login.

Features of our website:

- **Home**: This is the landing page of our web application. It comprises of the banner which has the following sections- Services, About, Contact Us and Login.

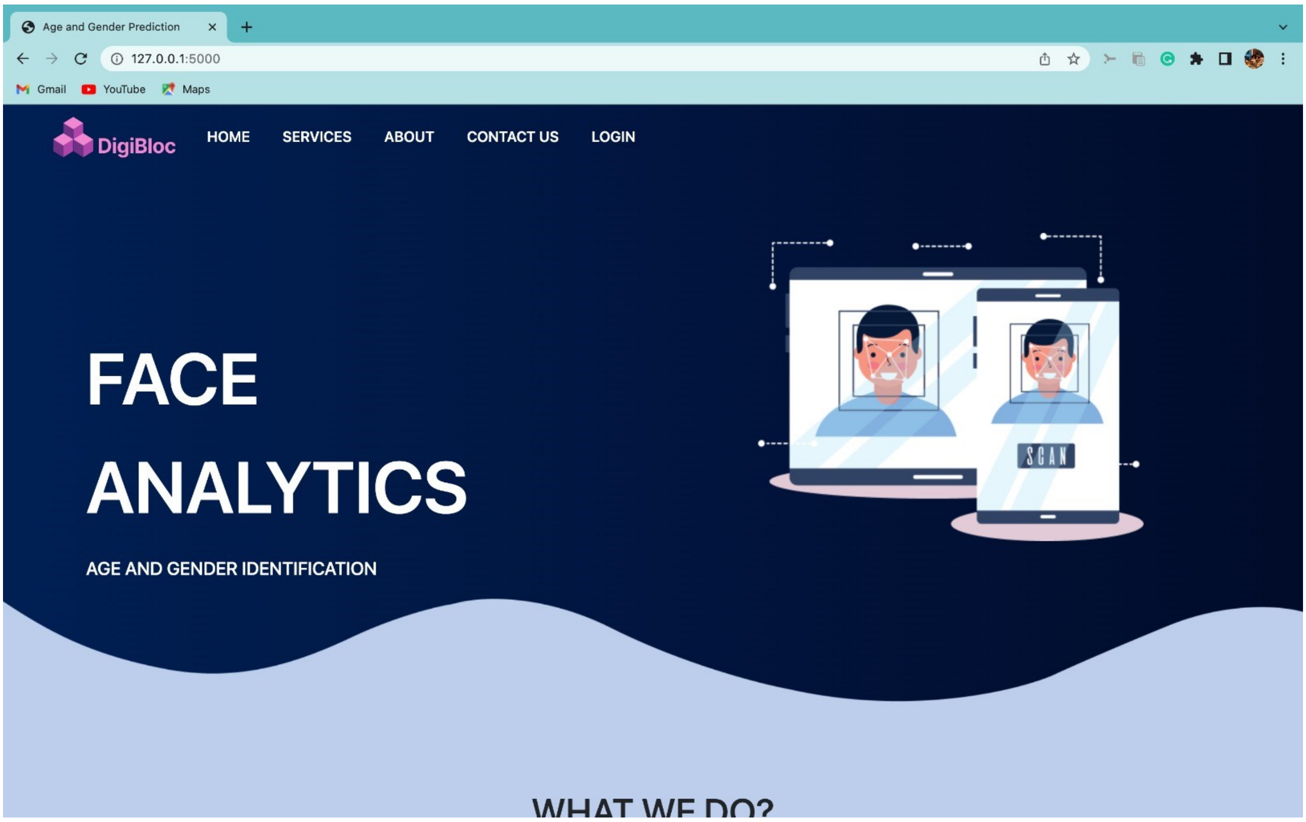


Figure 5.21 Landing Page

- **Services**: The Services section portrays the main applications the website provides which are namely, Age Estimation and Gender Classification. Clicking on the button brings us to the demo page where the user can try out the services, i.e., the Live and Photo Upload features. The Live feature allows one to detect age and gender in real time, whereas in the Photo Upload feature the user can upload any facial image to predict the age and gender of the person.

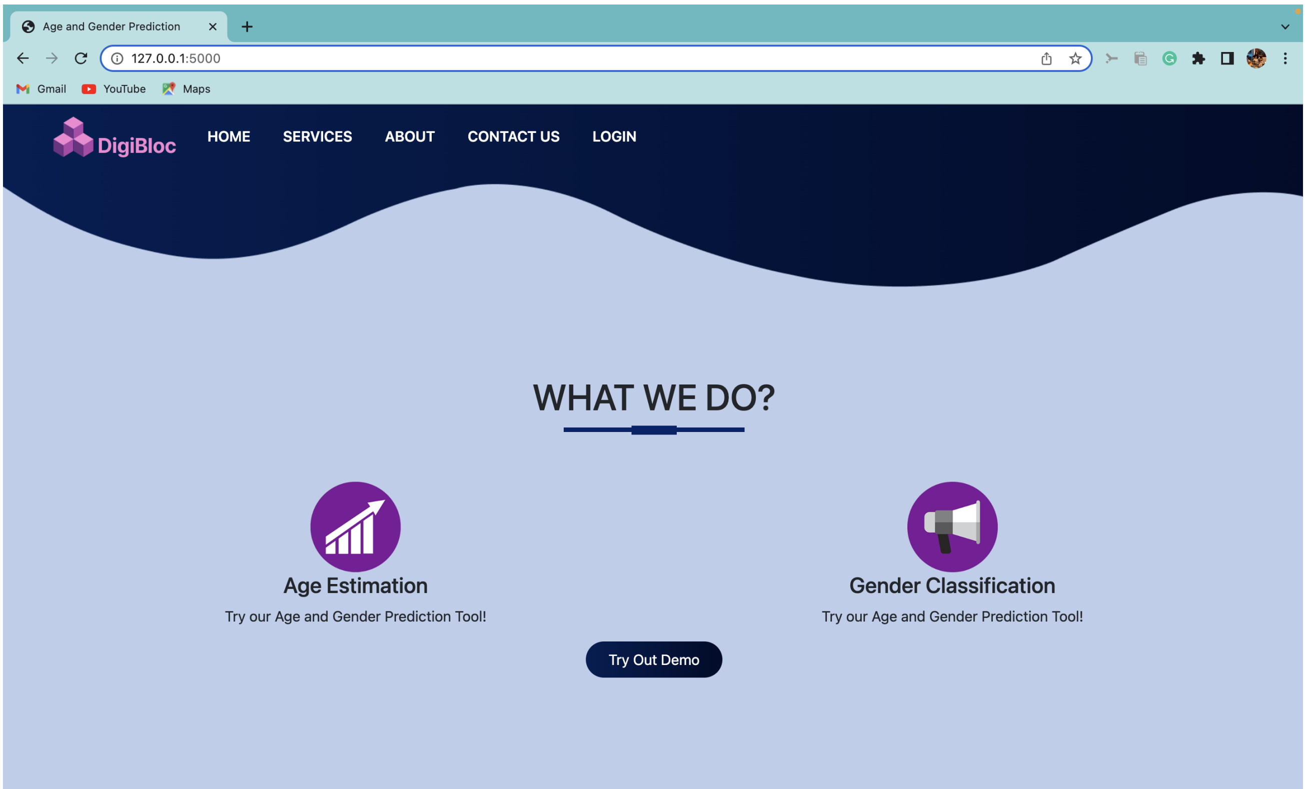


Figure 5.22 Services Section

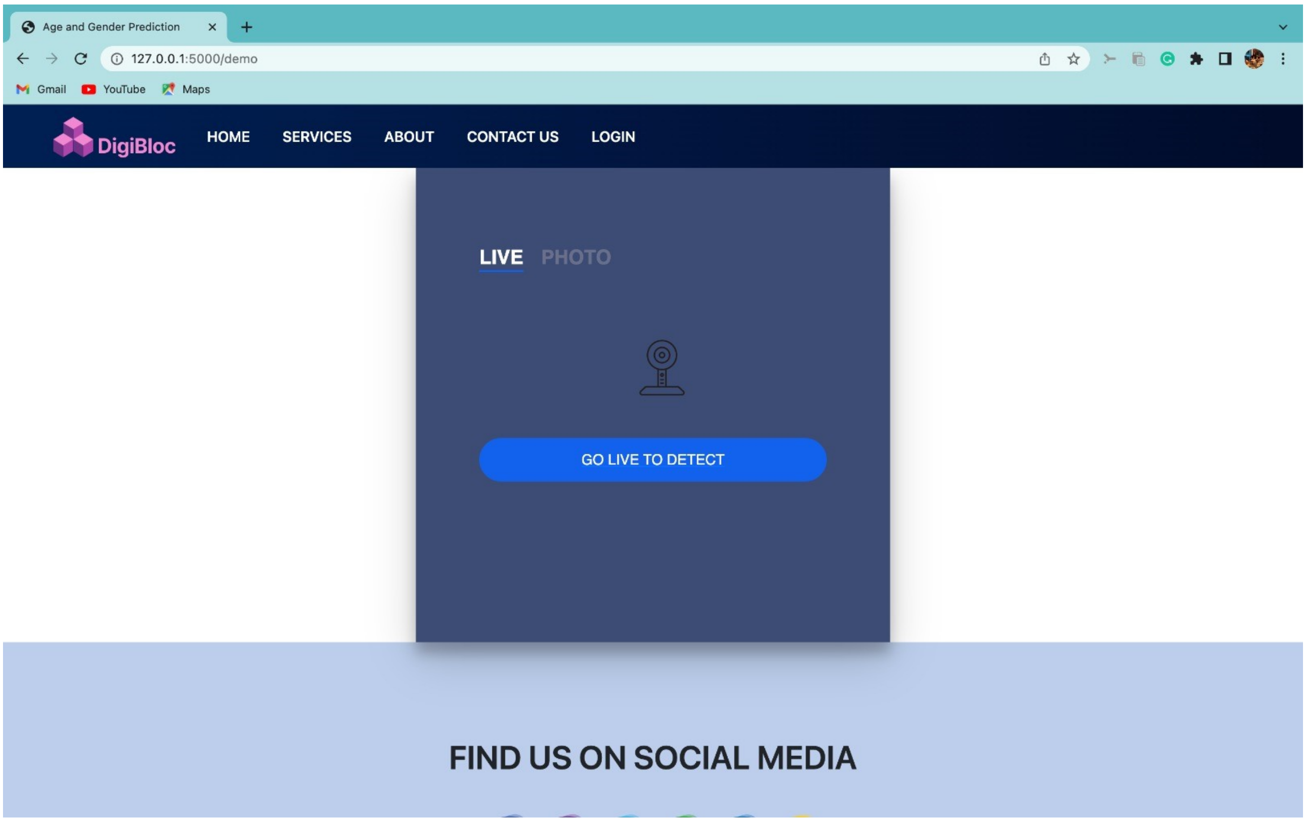


Figure 5.23 Demo Page of Live Feature

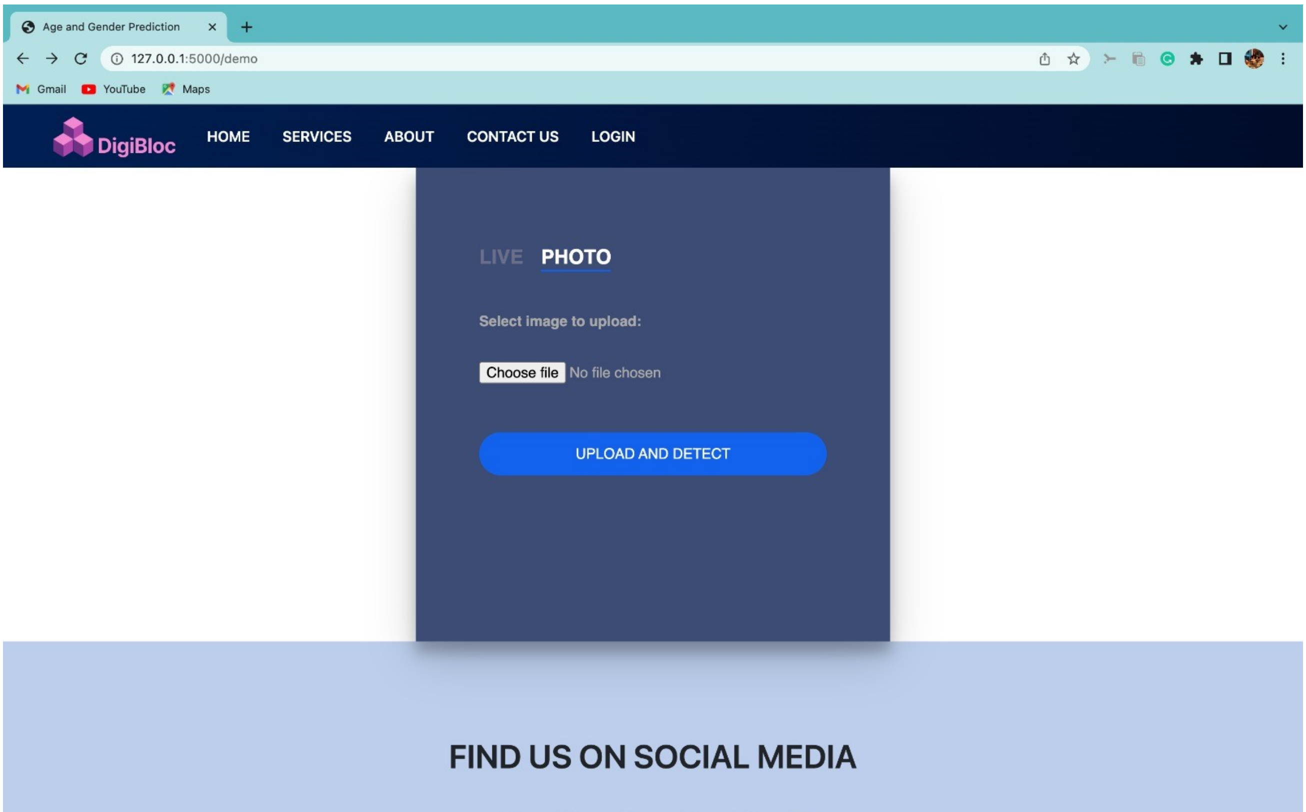


Figure 5.24 Demo Page of Photo Upload Feature

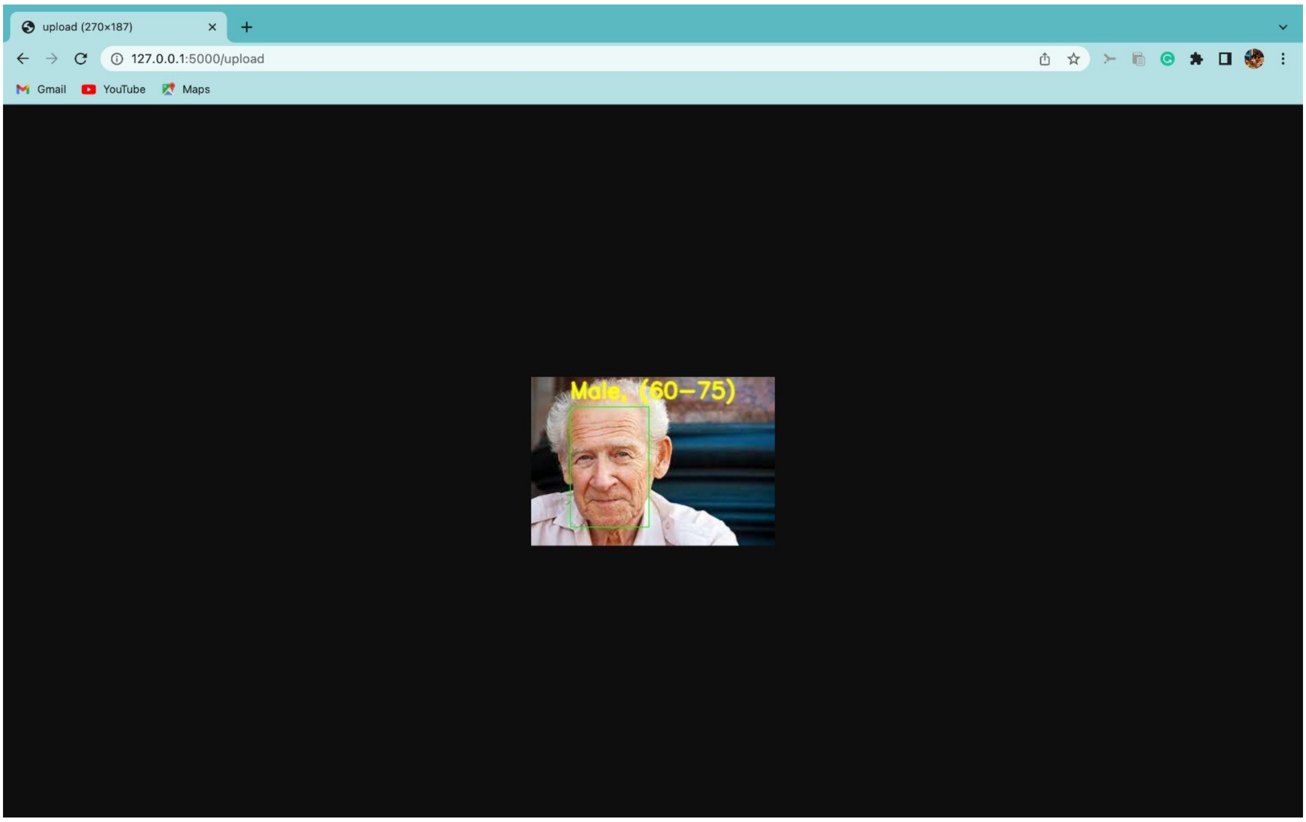


Figure 5.25 Output of Photo Upload Feature

- **About Us**: This segment defines the qualities that set us apart from the other similar services available online.

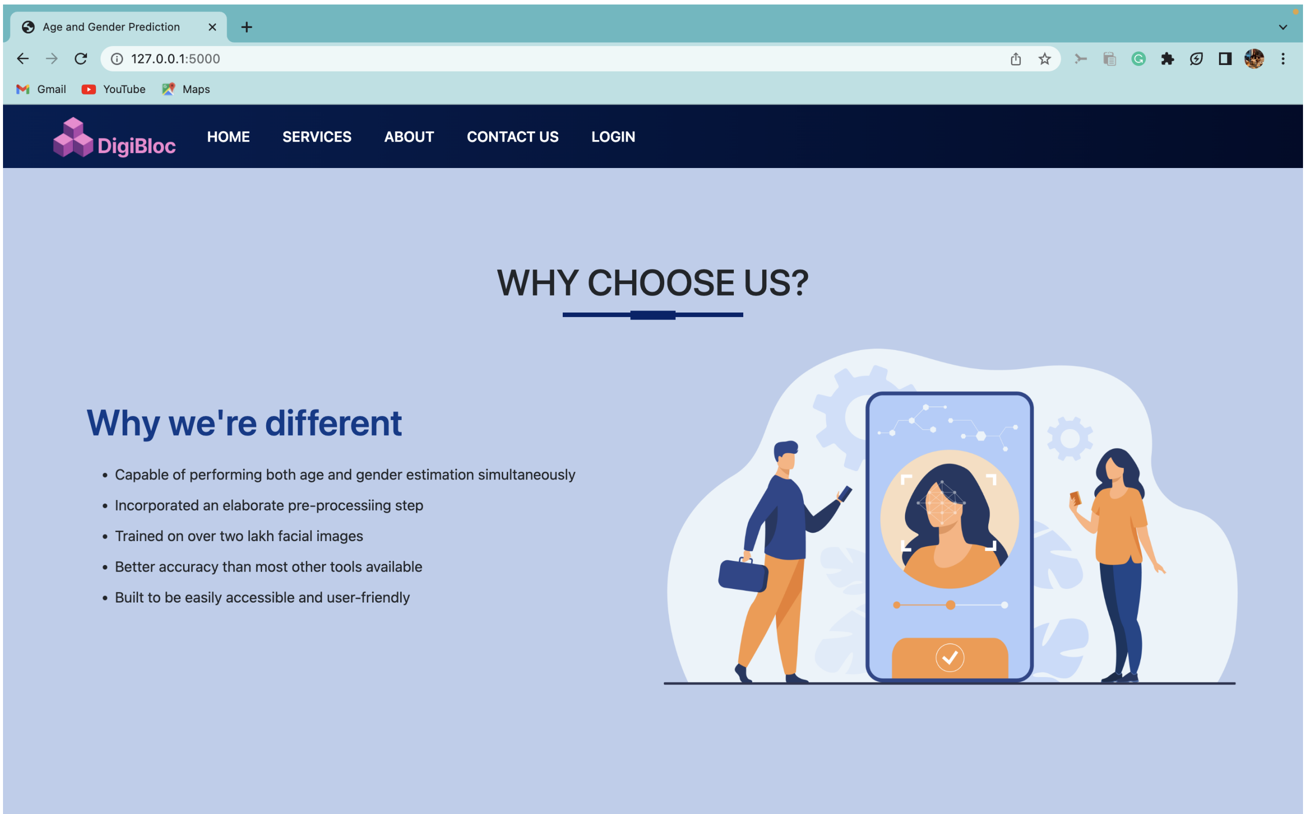


Figure 5.26 About Section

- **Contact Us**: This module contains contact options like email id and phone number for users to directly contact the website owner or team members in case of any doubts or queries. There are also other options to reach out to, for example on different social media platforms.

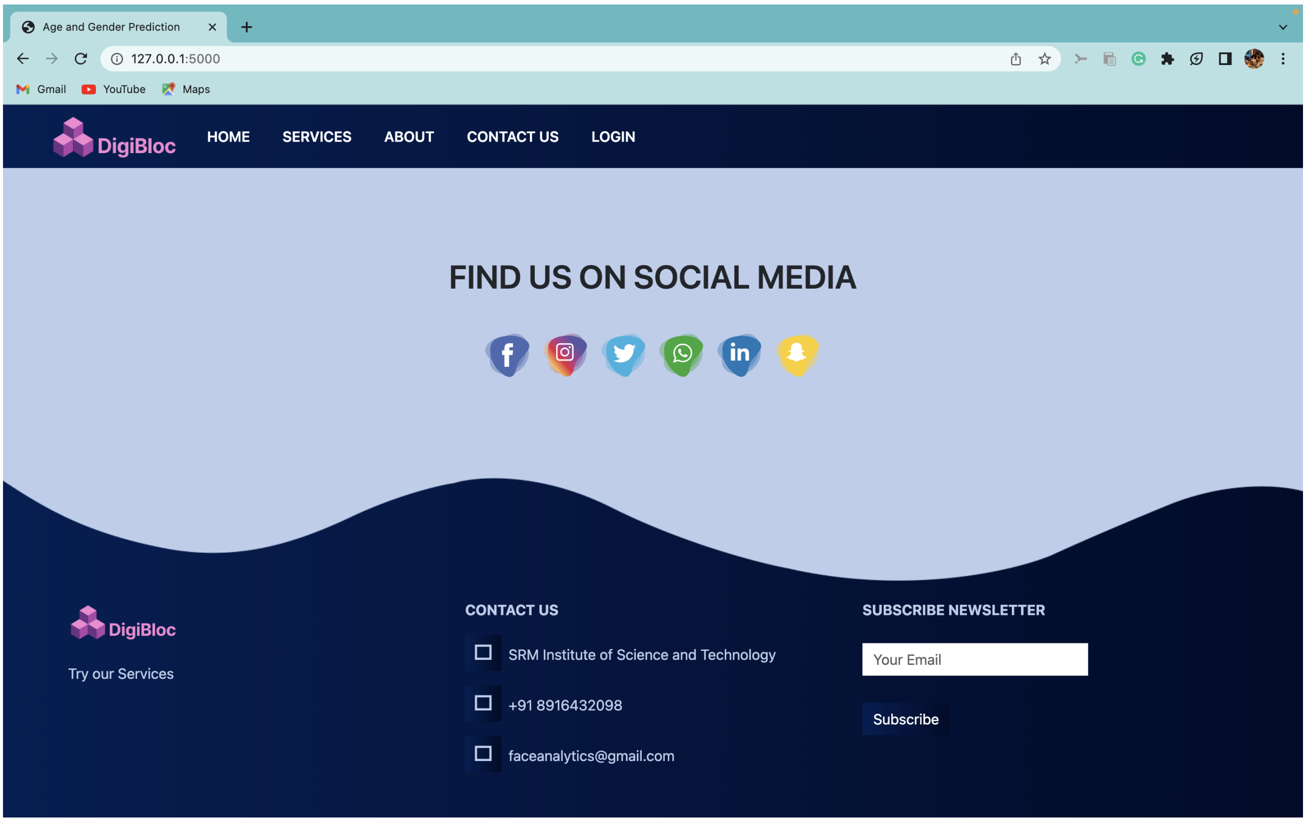


Figure 5.27 Contact Section

- **Login**: The Login section comprises of the Login and the Registration feature for returning and new users of the website respectively. Returning users can easily log into the website using their email id and password. The option for forgot password has also been introduced to make it user-friendly. For new users, it is necessary to create an username and password and sign up using their email id.

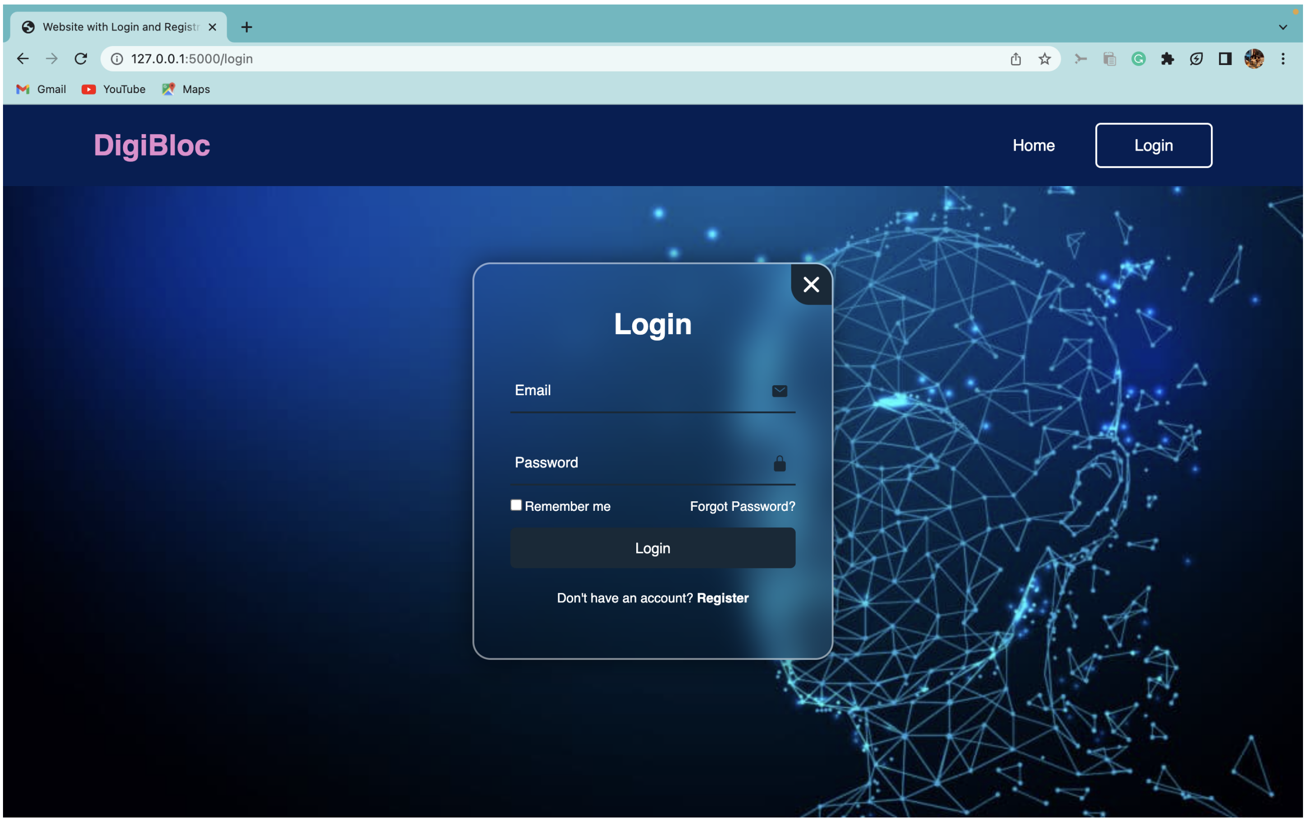


Figure 5.28 Login Page

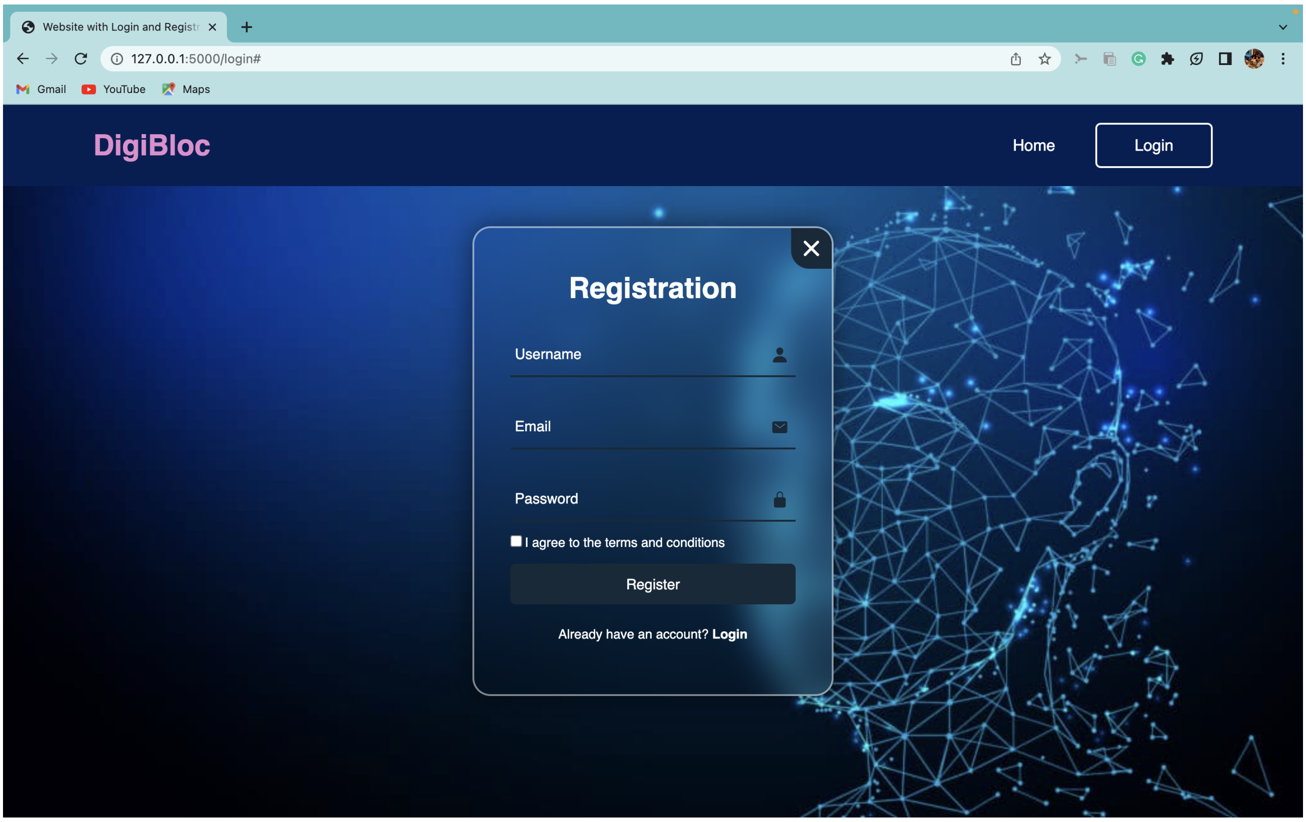


Figure 5.29 Registration Page

5.7 Challenges

- One of the initial shortcoming was the ability of the network to reduce the loss and thereby increase the validation accuracy of the age estimation task.
- To achieve the state of the art accuracy for gender classification.
- The ability of the network to learn and predict image which consist of faces at abnormal angle.
- To generalize well for all the various classes of age as well as learn various features of the face when it's not aligned.
- To relearn the features of production data and handle the shift in distribution of the images i.e. data shift.

CHAPTER 6

RESULTS AND DISCUSSION

In this subsection, the outcomes of conducting an analysis of the suggested procedure using the Adience and the IMDB-WIKI datasets are shown. In order to estimate gender and age, we have used a binary classification for the former and a multi-class classification for the latter. In terms of correctly identifying a person's gender, our CNN model obtained an accuracy of 84.68% and 86.25% on the Adience and the IMDB-WIKI datasets respectively. The age prediction is reviewed in order to place the subject in the appropriate age range among the eight groups, which are of the following ages: 0 to 2, 4 to 6, 8 to 12, 15 to 20, 25 to 32, 38 to 43, 48 to 53 and 60 to 75. On the Adience dataset, the accuracy of the model was measured at 40.29 percent for this particular metric. Whereas on the IMDB-WIKI dataset, it was 70.5%.

When the face alignment was correct, which is to say when the subject was looking directly into the camera, the model had better results. In certain cases, the performance of the model was negatively impacted by quite insignificant shifts in the alignment. The CNN architecture had trouble recognizing the side facial profile in those particular circumstances, which had an impact on the numbers.

It's possible that our model requires either more data or larger datasets in order to be properly trained, which would explain the decreased accuracy of the age estimate. The results would be improved if additional data were input into the model.

CHAPTER 7

CONCLUSION AND FUTURE ENCHANCEMENT

In this work, we tackled the problem of age group and gender classification of unfiltered real- world facial images. The gender identification and age estimation tasks were posed as a binary and a multi-class classification problem respectively. A six layer CNN architecture has been proposed and implemented for the same. Our proposed model is originally trained on the Adience dataset and then on the IMDB-WIKI dataset. It has achieved a gender accuracy of 86.25% and an accuracy of 70.5% for the age metrics on the IMDB-WIKI dataset. Haar Cascades has been employed to improve the results in the aspect of face detection.

For future works, we aim to increase the accuracy by making use of a larger dataset for fine- tuning our model. Moreover, the model can be trained further as their loss functions are still comparatively on a downward trend.

REFERENCES

[1] P. Smith and C. Chen, "*Transfer learning with deep CNNs for gender recognition and age estimation*", *Proc. IEEE Int. Conf. Big Data (Big Data)*, pp. 2564-2571, Dec. 2018.

[2] J. H. Lee, Y. M. Chan, T. Y. Chen and C. S. Chen, "*Joint estimation of age and gender from unconstrained face images using lightweight multi-task CNN for mobile applications*", *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Apr. 2018.

[3] A. Das, A. Dantcheva and F. Bremond, "*Mitigating bias in gender age and ethnicity classification: A multi-task convolution neural network approach*", *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, pp. 1-13, 2018.

[4] S. Hosseini, S. H. Lee, H. J. Kwon, H. I. Koo and N. I. Cho, "*Age and gender classification using wide convolutional neural network and Gabor filter*", *Proc. Int. Workshop Adv. Image Technology (IWAIT)*, pp. 1-3, 2018.

[5] K. Zhang, N. Liu, X. Yuan, X. Guo, C. Gao, Z. Zhao, et al., "*Fine-grained age estimation in the wild with attention LSTM networks*", *IEEE Trans. Circuits Syst. Video Technology*, vol. 30, no. 9, pp. 3140-3152, Sep. 2020.

[6] Insha Rafique, Awais Hamid, Sheraz Naseer, Muhammad Asad, Muhammad Awais, Talha Yasir, "*Age and Gender Prediction using Deep Convolutional Neural Networks*", *2019 International Conference on Innovative Computing (ICIC)*, 2019.

[7] Kyoungson Jhang, Junsoo Cho, "*CNN Training for Face Photo based Gender and Age Group Prediction with Camera*", *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2019.

[8] A. V. Savchenko, "*Efficient facial representations for age gender and identity recognition in organizing photo albums using multi-output ConvNet*", *PeerJ Comput. Sci.*, vol. 5, pp. e197, Jun. 2019.

[9] Cao Hong Nga, Khai-Thinh Nguyen, Nghi C. Tran, Jia-Ching Wang, "*Transfer Learning for Gender and Age Prediction*", *2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, 2020.

[10] Mohammed Kamel Benkaddour, Sara Lahlali, Maroua Trabelsi, "*Human Age And Gender Classification using Convolutional Neural Network*", *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*, 2020.

[11] Earnest Paul Ijjina, Goutham Kanahasabai, Aniruddha Srinivas Joshi, "*Deep Learning based approach to detect Customer Age, Gender and Expression in Surveillance Video*", *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020.

[12] O. Agbo-Ajala and S. Viriri, "*Deeply learned classifiers for age and gender predictions of unfiltered faces*", *Sci. World J.*, vol. 2020, pp. 1-12, Apr. 2020.

[13] Anirudh Ghildiyal, Sachin Sharma, Ishita Verma, Urvi Marhatta, “*Age and Gender Predictions using Artificial Intelligence Algorithm*”, *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, 2020.

[14] Jayaprada S. Hiremath, Shantakumar B. Patil, Premjyoti S. Patil, “*Human Age and Gender Prediction using Machine Learning Algorithm*”, *2021 IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, 2021.

[15] N. Shanthi;P Yuvasri, S Vaishnavi, P Vidhya, “*Gender and Age Detection using Deep Convolutional Neural Networks”, 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2022.

[16] A. Garain, B. Ray, P. K. Singh, A. Ahmadian, N. Senu and R. Sarkar, "*GRA_Net: A Deep Learning Model for Classification of Age and Gender From Facial Images*", in *IEEE Access*, vol. 9, pp. 85672-85689, 2021.

APPENDIX A

APPENDIX

The below section contains the code, snippets and other miscellaneous screenshots from training and developing the model and application:

File Edit View Window Help cleaning and preprocessing imdbwiki.py - ...\age and gender prediction\project

cleaning and preprocessing imdbwiki.py

```
import numpy as np
from scipy.io import loadmat
import pandas as pd
import datetime as date
from dateutil.relativedelta import relativedelta
import cv2

# Python program to store list to file using pickle module
import pickle

cols = ['age', 'gender', 'path', 'face_score1', 'face_score2']

imdb_path = r'E:\age and gender prediction\project\data\imdb_crop\imdb.mat'
wiki_path = r'E:\age and gender prediction\project\data\wiki_crop\wiki.mat'

imdb_mat = imdb_path
wiki_mat = wiki_path

imdb_data = loadmat(imdb_mat)
wiki_data = loadmat(wiki_mat)

del imdb_mat, wiki_mat

imdb = imdb_data['imdb']
wiki = wiki_data['wiki']
```

File Edit View Window Help cleaning and preprocessing imdbwiki.py - ...\a

cleaning and preprocessing imdbwiki.py

```
    wiki_path.append('wiki_crop/' + path[0])

imdb_genders = []
wiki_genders = []

for n in range(len(imdb_gender)):
    if imdb_gender[n] == 1:
        imdb_genders.append('male')
    else:
        imdb_genders.append('female')

for n in range(len(wiki_gender)):
    if wiki_gender[n] == 1:
        wiki_genders.append('male')
    else:
        wiki_genders.append('female')

```

File Edit View Window Help cleaning and preprocessing imdbwiki.py - ...\age and gender prediction\project

cleaning and preprocessing imdbwiki.py

```python
imdb_age = []
wiki_age = []

for i in range(len(imdb_dob)):
    try:
        d1 = date.datetime.strptime(imdb_dob[i][0:10], '%Y-%m-%d')
        d2 = date.datetime.strptime(str(imdb_photo_taken[i]), '%Y')
        rdelta = relativedelta(d2, d1)
        diff = rdelta.years
    except Exception as ex:
        print(ex)
        diff = -1
    imdb_age.append(diff)

for i in range(len(wiki_dob)):
    try:
        d1 = date.datetime.strptime(wiki_dob[i][0:10], '%Y-%m-%d')
        d2 = date.datetime.strptime(str(wiki_photo_taken[i]), '%Y')
        rdelta = relativedelta(d2, d1)
        diff = rdelta.years
    except Exception as ex:
        print(ex)
        diff = -1
    wiki_age.append(diff)
```

Gender Classification using Facial Images

Import Statements

```
import numpy as np
import plotly.express as px
import pandas as pd
import cv2
import os
from glob import glob
from PIL import Image
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPool2D, Activation, Dropout, Flatten, Dense, Dropout, LayerNormalization
from tensorflow.keras.preprocessing.image import ImageDataGenerator, img_to_array, load_img
import matplotlib.pyplot as plt
import pickle

import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt

#added import for sgd
from tensorflow.keras.optimizers import SGD
```

Date Retrival and Cleaning

Converting the txt data to a dataframe

```
df_list = []
for file_name in glob("/age and gender prediction/project/data/AdienceGender/text reference to images/*.txt"):
    df_temp = pd.read_csv(file_name, sep="\t")
    df_list.append(df_temp)
df = pd.concat(df_list, axis=0, ignore_index=True)
del df_list
```

```
df.shape
```

```
(19370, 12)
```

Using the details of dataframe to get the image path

```
df['image_path'] = df[['user_id', 'face_id', 'original_image']].apply(
    lambda x: os.path.join('/age and gender prediction/project/data/AdienceGender/aligned', f"{x[0]}", f"landmark_aligned_face.{x[1]}.{x[2]}"), axis=1)
```

Eliminating images with gender as *unidentified*

```
new_df = df[df['gender'] != 'u'][['age', 'gender', 'x', 'y', 'dx', 'dy','image_path']]
```

```
del df
```

Mapping target to float value

```
new_df['gender'] = new_df['gender'].apply(lambda x : 1 if x == 'm' else 0).astype(np.float32)
```

Image Preprocessing

Preparing to split for train and test set

```
X = new_df[['image_path']].values
y = new_df[['gender']].values
```

Train Test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Assigning uniform image extensions and resizing

```
def preprocess_image(individual_path):
    img = tf.io.read_file(np.array(individual_path).ravel()[0])
    img = tf.image.decode_jpeg(img)
    img = tf.image.resize(img, [227,227])
    return img
```

Shuffling the data

```
AUTOTUNE = tf.data.AUTOTUNE
ds_train = ds_train.cache().shuffle(buffer_size=1000).batch(32).prefetch(buffer_size=AUTOTUNE)
ds_test = ds_test.cache().shuffle(buffer_size=1000).batch(32).prefetch(buffer_size=AUTOTUNE)
```

Model Implementation

Data Augmentation

```
data_augmentation = tf.keras.Sequential([
  tf.keras.layers.RandomFlip("horizontal_and_vertical"),
  tf.keras.layers.RandomRotation(0.2),
  tf.keras.layers.RandomZoom(0.2,0.2),
])
```

```
callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3)
#optimiser variables added
learning_rate = 0.01
epochs = 50
decay_rate = learning_rate / epochs
momentum = 0.9
sgd = SGD(learning_rate=learning_rate, momentum=momentum, decay=decay_rate, nesterov=False)
#loss and optimiser change from sparse and adam
model.compile(optimizer=sgd, loss = tf.keras.losses.BinaryCrossentropy(), metrics='accuracy')
```

Age Classification Using Facial Images ¶

Import Statements

```
import numpy as np
import plotly.express as px
import pandas as pd
import cv2
import os
from glob import glob
from PIL import Image
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPool2D, Activation, Dropout, Flatten, Dense, Dropout, LayerNormalization
from tensorflow.keras.preprocessing.image import ImageDataGenerator, img_to_array, load_img
import matplotlib.pyplot as plt
import pickle

import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt

from tensorflow.keras.optimizers import Adam
#added import for sgd
from tensorflow.keras.optimizers import SGD
```

Data Retrival and Cleaning

Converting the txt data to a dataframe

```
df_list = []
for file_name in glob("/age and gender prediction/project/data/AdienceAge/text reference to images/*.txt"):
    df_temp = pd.read_csv(file_name, sep="\t")
    df_list.append(df_temp)
df = pd.concat(df_list, axis=0, ignore_index=True)
del df_list
```

```
df.shape
```

```
(19370, 12)
```

Shuffling the data

```
AUTOTUNE = tf.data.AUTOTUNE
ds_train = ds_train.cache().shuffle(buffer_size=1000).batch(32).prefetch(buffer_size=AUTOTUNE)
ds_test = ds_test.cache().shuffle(buffer_size=1000).batch(32).prefetch(buffer_size=AUTOTUNE)
```

Model Implementation

Data Augmentation

```
data_augmentation = tf.keras.Sequential([
  tf.keras.layers.RandomFlip("horizontal_and_vertical"),
  tf.keras.layers.RandomRotation(0.2),
  tf.keras.layers.RandomZoom(0.2,0.2),
])
```

```
callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3)
#optimiser variables added
learning_rate = 0.01
epochs = 50
decay_rate = learning_rate / epochs
momentum = 0.9
sgd = SGD(learning_rate=learning_rate, momentum=momentum, decay=decay_rate, nesterov=False)
#loss and optimiser change from sparse and adam
model.compile(optimizer=sgd, loss = tf.keras.losses.BinaryCrossentropy(), metrics='accuracy')
```

APPENDIX B

PUBLICATION STATUS

Our paper "Age and Gender Prediction using Adaptive Gamma Correction and Convolutional Neural Network" was presented at the *2023 International Conference on Computer Communication and Informatics* held at Coimbatore, India on Jan, 23-25, 2023. We presented this paper at the conference and received positive feedback.


2023 INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATION AND INFORMATICS
TECHNICAL SPONSORS
IEEE
IEEE MADRAS SECTION
13 TH EDITION
CERTIFICATE
OF APPRECIATION
THIS CERTIFICATE IS PROUDLY PRESENTED TO
Nithish Kumar S
HAS PRESENTED THE 'TECHNICAL PAPER' ENTITLED
Age and Gender Prediction using Adaptive Gamma Correction and Convolutional Neural Network
in the International Conference on Computer Communication and Informatics held from 23-25 January 2023
23-25 Jan 2023
DATE
CONFERENCE CHAIR
SRI SHAKTHI
INSTITUTE OF ENGINEERING AND TECHNOLOGY
AN AUTONOMOUS INSTITUTION
Chinniyampalayam, Coimbatore- 62
2727
COUNSELLING CODE
NAAC

IEEE ICCCI 2023 - Paper Accepted - Paper ID: 339 External Inbox ×

ICCCI <info@iccci.in>
to me

Fri, Dec 30, 2022, 6:06 PM

Dear Anweasha Saha

Paper ID:	CS339
Name	Anweasha Saha
Paper Title	Age and Gender Prediction using Adaptive Gamma Correction for Face Image Enhancement and Convolutional Neural Network
Initial Status	Accepted
Plagiarism Status	Accepted
Technical Status	Accepted
Reviewer Comments	
Registration Status	

Dear Author,
With reference to the Review report from the Technical Review Committee, your paper has been accepted for ICCCI 2023 Conference.
Comments:
Accepts the paper as it is.
Strictly adhere to IEEE template
Images shall be in good quality
Strengthen the title of the paper
The registration and payment process will start only from 1st of January 2023.
During registration process, submit the following:
1.CRC paper
2.Payment receipt
3.copyright form
4.PDF copy generated from PDF Express login
Other Guidelines:
Procedures for registration:

APPENDIX C

PLAGIARISM REPORT

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)		
Office of Controller of Examinations		
REPORT FOR PLAGIARISM CHECK ON THE SYNOPSIS/THESIS/DISSERTATION/PROJECT REPORTS		
1	Name of the Candidate **(IN BLOCK LETTERS)**	NITHISH KUMAR S ANWEASHA SAHA
2	Address of the Candidate	60, LKA Nagar, Dharapuram Road, Vellakovil, Tirppur (dt) – 638111. Mobile Number: +91-9360637610 Mail Id: sk7649@srmist.edu.in Flat no 3 C&D Shankalan Apartment Nutanpally, near durga bari garage Purba bardhaman, west bengal PIN: 713101 Mobile Number: +91-7890102070 Mail Id: as6009@srmist.edu.in
3	Registration Number	RA1911003010217 RA1911003010235
4	Date of Birth	01/04/2002 03/03/2001
5	Department	Department of Computing Technologies
6	Faculty	Mrs. V. S. SARANYA
7	Title of the Synopsis/ Thesis/ Dissertation/Project	Age and Gender Classification Using CNN
8	Whether the above project /dissertationis done by	~~Individual or~~ group: (Strike whichever is not applicable) If the project/ dissertation is done in group, then how many students together completed the project : 2 (Two) Mention the Name & Register number of other candidates: Nithish Kumar S, RA1911003010217 Anweasha Saha, RA1911003010235
9	Name and address of the Supervisor / Guide	Mrs. V. S. Saranya Assistant Professor Department of Computing Technologies SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamil Nadu, 603203 Mail ID: saranyav3@srmist.edu.in Mobile Number: +91-9080584021

10	Name and address of the Co-Supervisor / Co- Guide (if any)	NA		
11	Software Used	Turnitin		
12	Date of Verification	May 5, 2023		
13	**Plagiarism Details: (to attach the final report)**			
Chapter	**Title of the Chapter**	**Percentage of similarity index (Including self citation)**	**Percentage of similarity index (Excluding self citation)**	**% of plagiarism after excluding Quotes, Bibliography, etc.,**
1	Introduction	<1	< 1	< 1
2	Literature Survey	<1	< 1	< 1
3	System Architecture and Design	1	1	1
4	Challenges and Methodology	1	1	1
5	Coding and Testing	<1	<1	< 1
6	Results and Discussions	0	0	0
7	Conclusions and Future Enhancements	0	0	0
Thesis abstract		<1	<1	<1
Appendices		<1	<1	<1
We declare that the above information has been verified and found true to the best of our knowledge.				
Signature of the Candidate		**Name & Signature of the Staff (Who uses the plagiarism check software)**		
Name & Signature of the Supervisor/Guide		**Name & Signature of the Co-Supervisor/Co-Guide**		
Name & Signature of the HOD				

AGE AND GENDER CLASSIFICATION USING CNN

ORIGINALITY REPORT

6%	2%	3%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Westminster Student Paper	1%
2	Submitted to University of Greenwich Student Paper	<1%
3	dokumen.pub Internet Source	<1%
4	Submitted to University of Nottingham Student Paper	<1%
5	"MultiMedia Modeling", Springer Nature, 2017 Publication	<1%
6	Philip Smith, Cuixian Chen. "Transfer Learning with Deep CNNs for Gender Recognition and Age Estimation", 2018 IEEE International Conference on Big Data (Big Data), 2018 Publication	<1%
7	Paul L. Myers, Dothang Truong. "A New Research Model for Higher Risk ACTIVITES Applied to the Use of Small Unmanned Aircraft for Data Gathering Operations", Journal of Intelligent & Robotic Systems, 2020	<1%