**A PROJECT REPORT**

**of**

**REAL TIME GENDER, AGE AND**

**EMOTION DETECTION**

**USING DEEP LEARNING**

by

Ritesh Kumar Dubey (1803310166)

Shivam Kripashankar Singh (1803310215)

Prakhar Kumar (1803310139)

Tanmay Gupta (1803310220)

Under the Supervision of

**Dr. PAWAN**

**Department of Computer Science and Engineering**

**Raj Kumar Goel Institute of Technology, Ghaziabad**

**Dr. A.P.J Abdul Kalam Technical University Lucknow**

**Session: 2021-22**

**May, 2022**

**REAL TIME GENDER, AGE AND**

**EMOTION DETECTION**

**USING DEEP LEARNING**

By

Ritesh Kumar Dubey (1803310166)

Shivam Kripashankar Singh (1803310215)

Prakhar Kumar (1803310139)

Tanmay Gupta (1803310220)

Under the Supervision of

**Dr. PAWAN**

Submitted to the Department of Computer Science & Engineering

In partial fulfillment of the requirements

For the degree of

**Bachelor of Technology**

In

**Computer Science & Engineering**

Raj Kumar Goel Institute of Technology, Ghaziabad

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

Session: 2021-22

May, 2022

# **CERTIFICATE**

This is to certify that Synopsis Report Entitled “Real Time Gender, Age and Emotion Detection Using Deep Learning” which is submitted in partial fulfillment of the requirement for the award of degree B.Tech. in Computer Science and Engineering to R.K.G.I.T, Ghaziabad, Dr. A.P.J. Abdul Kalam Technical University, Lucknow comprises only original work and studies carried out by students himself. The matter embodied in this synopsis has not been submitted for the award of any other degree.

**Date:**

**Project Guide Name and Signature:**

Dr. Pawan

# **DECLARATION**

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature  
Name: Ritesh Kumar Dubey Roll No. -1803310166

Signature  
Name: Shivam Kripashankar Singh

Roll No.- 1803310215

Signature  
Name: Prakhar Kumar

Roll No. -1803310139

Signature  
Name: Tanmay Gupta

Roll No. -1803310220

**Approved By:**

**Signature**

**Dr. Pawan**

**Project Coordinator**

**CSE Department**

**RKGIT, Ghaziabad**

**Signature**

**Dr. Lalit Saraswat**

**HOD**

**CSE Department**

**RKGIT, Ghaziabad**

# **ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year, we owe special debt of gratitude to Prof. (Dr.) D.R. Somashekhar, Director, Raj Kumar Goel Institute of Technology, Ghaziabad for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Project guide Dr. Pawan, Department of Computer Science & Engineering, Raj Kumar Goel Institute of Technology, Ghaziabad for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature  
Name: Ritesh Kumar Dubey

Roll No. - 1803310166

Signature  
Name: Shivam Kripashankar Singh

Roll No.-1803310215

Signature  
Name: Prakhar Kumar

Roll No. -1803310139

Signature  
Name: Tanmay Gupta

Roll No. -1803310220

**Table of Contents**

[CERTIFICATE iii](#_Toc104826457)

[DECLARATION iv](#_Toc104826458)

[ACKNOWLEDGEMENT v](#_Toc104826459)

[ABSTRACT viii](#_Toc104826460)

[LIST OF FIGURES ix](#_Toc104826461)

[CHAPTER 1 1](#_Toc104826462)

[INTRODUCTION 1](#_Toc104826463)

[**1.1** FUNDAMENTALS OF EMOTION, AGE & GENDER DETECTION 1](#_Toc104826464)

[1.1.1 CONVOLUTIONAL NEURAL NETWORKS 3](#_Toc104826465)

[**1.2** TOOLS AND LIBRARY USED 4](#_Toc104826466)

[**1.2.2** NUMPY 8](#_Toc104826467)

[**1.2.3** PANDAS 9](#_Toc104826468)

[**1.2.4** TENSORFLOW 11](#_Toc104826469)

[**1.2.5** TENSORFLOW LITE 12](#_Toc104826470)

[**1.2.6** KERAS 13](#_Toc104826471)

[**1.2.7** ANACONDA 14](#_Toc104826472)

[1.2.7.1 JUPYTER NOTEBOOK 15](#_Toc104826473)

[**1.2.8** ANDROID STUDIO 17](#_Toc104826474)

[CHAPTER 2 18](#_Toc104826475)

[LITERATURE REVIEW 18](#_Toc104826476)

[**2.1** LITERATURE INTRODUCTION 18](#_Toc104826477)

[**2.2** LITERATURE REVIEW 19](#_Toc104826478)

[CHAPTER 3 22](#_Toc104826479)

[INFERENCES DRAWN FROM LITERATURE 22](#_Toc104826480)

[3.1 INFERENCES DRAWN OUT OF LITERATURE SURVEY 22](#_Toc104826481)

[CHAPTER 4 26](#_Toc104826482)

[PROBLEM STATEMENT AND SOLUTION APPROACH 26](#_Toc104826483)

[4.1 PROBLEMS WITH EXISTING EMOTION, AGE & GENDER DETECTION 26](#_Toc104826484)

[**4.1.1** CNN ARCHITECTURE FOR FACIAL RECOGNITION 32](#_Toc104826485)

[CHAPTER 5 44](#_Toc104826486)

[CONCLUSION 44](#_Toc104826487)

[5.1 SUMMARY 44](#_Toc104826488)

[5.2 POTENTIAL ACTIONS FOR IMPROVEMENTS 44](#_Toc104826489)

[5.3 CONCLUSION 47](#_Toc104826490)

[5.4 ADVANTAGES 47](#_Toc104826491)

[5.5 DISADVANTAGES 47](#_Toc104826492)

[REFERENCES 48](#_Toc104826493)

# **ABSTRACT**

The main aim of this article is to detect age and gender through the given data set. We will use simple python and Keras methods for detecting age and gender.

There are different sorts of procedures required for, just as the expulsion of the issue. In a Facial identification strategy: The articulations that the faces contain hold a great deal of data. At whatever point the individual associates with the other individual, there is an association of a ton of ideas.

The evolving of ideas helps in figuring certain boundaries. Age assessment is a multi- class issue in which the years; are categorized into classes. Individuals of various ages have various facials, so it is hard to assemble the pictures.

To identify the age and gender of several faces’ procedures, are followed by several methods. From the neural network, features are taken by the convolution network. In light of the prepared models, the image is processed into one of the age classes. The highlights are handled further and shipped off the preparation frameworks.

# **LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NUMBER** | **FIGURE DESCRIPTION** | **PAGE NUMBER** |
| Figure 1.1 | Flow chart of the age, emotion  and gender classifier model | 8 |
| Figure 1.2 | CNN architecture | 9 |
| Figure 1.3 | OpenCv | 11 |
| Figure 1.4 | Output of Personal webcam for  face and eye detection using OpenCv | 14 |
| Figure 1.5 | NumPy | 15 |
| Figure 1.6 | Pandas | 17 |
| Figure 1.7 | Tensorflow | 19 |
| Figure 1.8 | Keras | 21 |
| Figure 1.9 | Anaconda | 22 |
| Figure 1.10 | Jupyter | 23 |
| Figure 1.11 | Android Studio | 25 |
| Figure 2.1 | Cropping detected faces | 28 |
| Figure 2.2 | Putting faces in the face boxes | 30 |
| Figure 3.1 | Feature extraction and  classification flowchart | 31 |
| Figure 4.1 | The first curve shows the loss function decrease and the second shows the accuracy  growth with epochs | 36 |
| Figure 4.2 | The first bar graph shows the distribution of gender  The second line graph shows  the variation of samples of different ages | 37 |
| Figure 4.3 | Schematic diagram of the  model | 38 |
| Figure 4.4 | Model’s loss Curve | 39 |
| Figure 4.5 | Scatter plot for age prediction | 39 |
| Figure 4.6 | Gender accuracy with epochs | 40 |
| Figure 4.7 | CNN architecture for facial  recognition | 42 |
| Figure 4.8 | An overview of system  approaches | 49 |
| Figure 4.9 | Sample images from all  datasets | 52 |
| Figure 5.1 | Output 1 | 54 |
| Figure 5.2 | Output 2 | 55 |

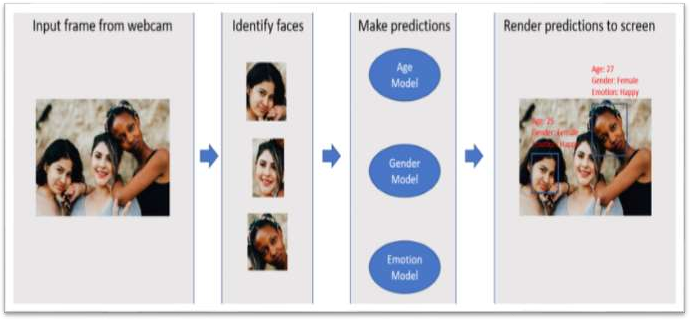
# **CHAPTER 1**

## INTRODUCTION

## FUNDAMENTALS OF EMOTION, AGE & GENDER DETECTION

One of the most active areas in facial technology is facial features such as age, gender and emotion recognition. A lot of research has been done using deep learning methods such as ANN, CNN to determine age, gender estimation and emotion detection. Human Facial expression depicts human feelings. Human tends to change feelings, perhaps due to their mental or health conditions, in the course of lifespan. Although people are brimming with various feelings, modern psychology characterizes six basic facial expressions: happiness, sadness, surprise, fear, disgust and anger as widespread feelings. Facial muscles exercise to assist with perceiving individual’s feelings. Fundamental facial consideration features are eyebrows, mouth, nose and eyes.

An architecture based on the convolution Neural network (CNN) proposed here for age, gender and emotion classification. This is one of the well-known deep artificial neural networks. Convolutional Neural Network based design models are broadly utilized in classification task because of their remarkable execution in facial investigation and emotion detection. The Convolutional Neural Network includes Feature extraction which extracts Features corresponding to age, gender, emotion. Furthermore, CNN includes Feature classification which classifies facial images into the correct age group, gender and emotion such as happy, sad, angry, neutral.



**Figure 1.1**

As shown in above figure, the implementation contains 4 main steps:

* + 1. Receive input frame from the webcam
    2. Identify faces in the webcam and prepare these images for the 3 deep learning models, i.e. age, gender and emotion models
    3. Send processed faces to the models and receive prediction outcomes
    4. Render prediction outcomes with bounding boxes to screen

In this implementation, we will be using one of the state-of-the-art facial recognition models, MTCNN for step 2. It has a stable Python release based on Keras available at [here](https://github.com/ipazc/mtcnn).

For step 3, we will be training our own customized models. However, for less efforts and better accuracy, you might want to consider transfer learning techniques. Many pre- trained models including, [VGG-face](https://github.com/rcmalli/keras-vggface), [FaceNet](https://github.com/davidsandberg/facenet), [GoogLeNet](https://github.com/Lornatang/GoogLeNet-PyTorch) are available. Note that these pre-trained models might have different input size requirements. Thus faces identified from step 2 need to be processed accordingly.

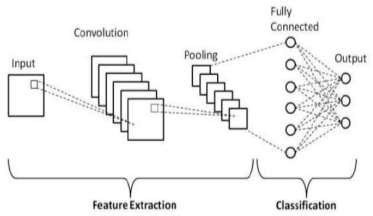
Convolutional networks were [inspired](https://en.wikipedia.org/wiki/Mathematical_biology) by [biological](https://en.wikipedia.org/wiki/Biological) processes[[9][10](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-fukuneoscholar-9)[][11][12]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-intro-11) in that the connectivity pattern between [neurons](https://en.wikipedia.org/wiki/Artificial_neuron) resembles the organization of the animal [visual](https://en.wikipedia.org/wiki/Visual_cortex) [cortex](https://en.wikipedia.org/wiki/Visual_cortex). Individual [cortical neurons](https://en.wikipedia.org/wiki/Cortical_neuron) respond to stimuli only in a restricted region of the [visual field](https://en.wikipedia.org/wiki/Visual_field) known as the [receptive field](https://en.wikipedia.org/wiki/Receptive_field). The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other [image classification](https://en.wikipedia.org/wiki/Image_classification) [algorithms](https://en.wikipedia.org/wiki/Image_classification). This means that the network learns to optimize the [filters](https://en.wikipedia.org/wiki/Filter_(signal_processing)) (or kernels) through automated learning, whereas in traditional algorithms these filters are [hand-](https://en.wikipedia.org/wiki/Feature_engineering) [engineered](https://en.wikipedia.org/wiki/Feature_engineering). This independence from prior knowledge and human intervention in feature extraction is a major advantage.

Although people are brimming with various feelings, modern psychology characterizes six basic facial expressions: happiness, sadness, surprise, fear, disgust and anger as widespread feelings. Facial muscles exercise to assist with perceiving individual’s feelings. Fundamental facial consideration features are eyebrows, mouth, nose and eyes.

## 1.1.1 CONVOLUTIONAL NEURAL NETWORKS

In [deep learning](https://en.wikipedia.org/wiki/Deep_learning), a **convolutional neural network** (**CNN**, or **ConvNet**) is a class of [artificial neural network](https://en.wikipedia.org/wiki/Artificial_neural_network), most commonly applied to analyze visual imagery.[[1]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-Valueva_Nagornov_Lyakhov_Valuev_2020_pp._232–243-1) They are also known as **shift invariant** or **space invariant artificial neural networks** (**SIANN**), based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation [equivariant](https://en.wikipedia.org/wiki/Equivariant_map) responses known as feature maps.[[2][3]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-%3A0-2) Counter-intuitively, most convolutional neural networks are only [equivariant](https://en.wikipedia.org/wiki/Equivariant_map), as opposed to [invariant](https://en.wikipedia.org/wiki/Translation_invariant), to translation.[[4]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-%3A6-4) They have applications in [image and video](https://en.wikipedia.org/wiki/Computer_vision) [recognition](https://en.wikipedia.org/wiki/Computer_vision), [recommender systems](https://en.wikipedia.org/wiki/Recommender_system),[[5]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-5) [image classification](https://en.wikipedia.org/wiki/Image_classification), [image segmentation](https://en.wikipedia.org/wiki/Image_segmentation), [medical image analysis](https://en.wikipedia.org/wiki/Medical_image_computing), [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing),[[6]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-6) [brain-computer interfaces](https://en.wikipedia.org/wiki/Brain%E2%80%93computer_interface),[[7]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-7) and financial [time series](https://en.wikipedia.org/wiki/Time_series).[[8]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-Tsantekidis_7–12-8)



**Figure 1.2**

CNNs are [regularized](https://en.wikipedia.org/wiki/Regularization_(mathematics)) versions of [multilayer perceptrons](https://en.wikipedia.org/wiki/Multilayer_perceptron). Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one [layer](https://en.wikipedia.org/wiki/Layer_(deep_learning)) is connected to all neurons in the next [layer](https://en.wikipedia.org/wiki/Layer_(deep_learning)). The "full connectivity" of these networks make them prone to [overfitting](https://en.wikipedia.org/wiki/Overfitting) data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) CNNs take a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble patterns of increasing complexity using smaller and simpler patterns embossed in their filters. Therefore, on a scale of connectivity and complexity, CNNs are on the lower extreme.

Convolutional networks were [inspired](https://en.wikipedia.org/wiki/Mathematical_biology) by [biological](https://en.wikipedia.org/wiki/Biological) processes[[9][10](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-fukuneoscholar-9)[][11][12]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-intro-11) in that the connectivity pattern between [neurons](https://en.wikipedia.org/wiki/Artificial_neuron) resembles the organization of the animal [visual](https://en.wikipedia.org/wiki/Visual_cortex) [cortex](https://en.wikipedia.org/wiki/Visual_cortex). Individual [cortical neurons](https://en.wikipedia.org/wiki/Cortical_neuron) respond to stimuli only in a restricted region of the [visual field](https://en.wikipedia.org/wiki/Visual_field) known as the [receptive field](https://en.wikipedia.org/wiki/Receptive_field). The receptive fields of different neurons partially overlap such that they cover the entire visual field.

CNNs use relatively little pre-processing compared to other [image classification](https://en.wikipedia.org/wiki/Image_classification) [algorithms](https://en.wikipedia.org/wiki/Image_classification). This means that the network learns to optimize the [filters](https://en.wikipedia.org/wiki/Filter_(signal_processing)) (or kernels) through automated learning, whereas in traditional algorithms these filters are [hand-](https://en.wikipedia.org/wiki/Feature_engineering) [engineered](https://en.wikipedia.org/wiki/Feature_engineering). This independence from prior knowledge and human intervention in feature extraction is a major advantage.

## TOOLS AND LIBRARY USED

* + 1. **OPENCV**



**Figure 1.3**

OpenCV was started at Intel in 1999 by **Gary Bradsky**, and the first release came out in 2000. **Vadim Pisarevsky** joined Gary Bradsky to manage Intel's Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle that won the 2005 DARPA Grand Challenge. Later, its active development continued under the support of Willow Garage with Gary Bradsky and Vadim Pisarevsky leading the project. OpenCV now supports a multitude of algorithms related to Computer Vision and

Machine Learning and is expanding day by day.

OpenCV supports a wide variety of programming languages such as C++, Python, Java, etc., and is available on different platforms including Windows, Linux, OS X, Android, and iOS. Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development.

OpenCV-Python is the Python API for OpenCV, combining the best qualities of the OpenCV C++ API and the Python language.

OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

Using OpenCV library, we can −

* Read and write images
* Capture and save videos
* Process images (filter, transform)
* Perform feature detection
* Detect specific objects such as faces, eyes, cars, in the videos or images.
* Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

OpenCV was originally developed in C++. In addition to it, Python and Java bindings were provided. OpenCV runs on various Operating Systems such as windows, Linux, OSx, FreeBSD,Net BSD, Open BSD, etc.

This tutorial explains the concepts of OpenCV with examples using Java bindings. Following are the main library modules of the OpenCV library:

#### Core Functionality

This module covers the basic data structures such as Scalar, Point, Range, etc., that are used to build OpenCV applications. In addition to these, it also includes the multidimensional array Mat, which is used to store the images. In the Java library of OpenCV, this module is included as a package with the name org.opencv.core.

#### Image Processing

This module covers various image processing operations such as image filtering, geometrical image transformations, color space conversion, histograms, etc. In the Java library of OpenCV, this module is included as a package with the name org.opencv.imgproc.

#### Video

This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the Java library of OpenCV, this module is included as a package with the name org.opencv.video.

#### Video I/O

This module explains the video capturing and video codecs using OpenCV library. In the Javalibrary of OpenCV, this module is included as a package with the name org.opencv.videoio.

#### calib3d

This module includes algorithms regarding basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence and elements of3D reconstruction. In the Java library of OpenCV, this module is included as a package withthe name org.opencv.calib3d.

#### features2d

This module includes the concepts of feature detection and description. In the Java library of OpenCV, this module is included as a package with the name org.opencv.features2d.

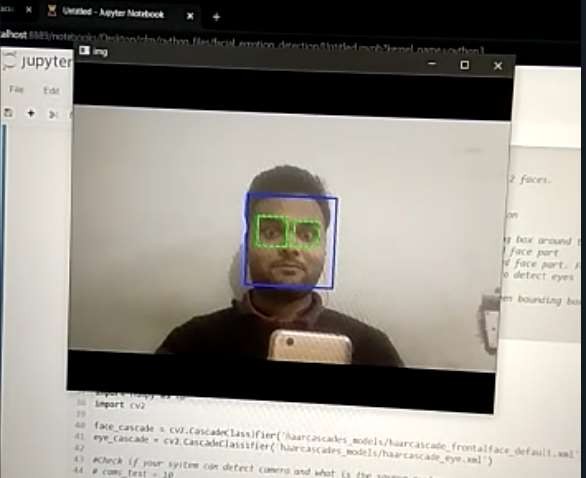
#### Objdetect

This module includes the detection of objects and instances of the predefined classes such as faces, eyes, mugs, people, cars, etc. In the Java library of OpenCV, this module is included asa package with the name org.opencv.objdetect.

#### Highgui

This is an easy-to-use interface with simple UI capabilities. In the Java library of OpenCV, the features of this module is included in two different packages namely, org.opencv.imgcodecs and org.opencv.videoio.

OpenCV (Open Source Computer Vision Library: http://opencv.org) is an open-source library that includes several hundreds of computer vision algorithms. The document describes the so- called OpenCV 2.x API, which is essentially a C++ API, as opposed to the C-based OpenCV

1.x API (C API is deprecated and not tested with "C" compiler since OpenCV 2.4 releases)

**Figure 1.4**

## NUMPY

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by TravisOliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important. NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematicaland logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.



**Figure 1.5**

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

**Numeric**, the ancestor of NumPy, was developed by Jim Hugunin. Another package

Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

Using NumPy, a developer can perform the following operations − Mathematical and logical operations on arrays.

Fourier transforms and routines for shape manipulation. Operations related to linear algebra.

NumPy has in-built functions for linear algebra and random number generation.

## PANDAS

**pandas** is a [software library](https://en.wikipedia.org/wiki/Software_library) written for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)) for data manipulation and [analysis](https://en.wikipedia.org/wiki/Data_analysis). In particular, it offers [data structures](https://en.wikipedia.org/wiki/Data_structure) and operations for manipulating numerical tables and [time series](https://en.wikipedia.org/wiki/Time_series). It is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [three-clause BSD license](https://en.wikipedia.org/wiki/3-clause_BSD_license).[[1]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-2) The name is derived from the term "[panel data](https://en.wikipedia.org/wiki/Panel_data)", an [econometrics](https://en.wikipedia.org/wiki/Econometrics) term for [data sets](https://en.wikipedia.org/wiki/Data_set) that include observations over multiple time periods for the same individuals.[[2]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-3) Its name is a play on the phrase "Python data analysis" itself.[[3]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-4) [Wes McKinney](https://en.wikipedia.org/wiki/Wes_McKinney) started building what would become pandas at [AQR Capital](https://en.wikipedia.org/wiki/AQR_Capital) while he was a researcher there from 2007 to 2010.[[4]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-5)



**Figure 1.6**

LIBRARY FEATURES

* + - * Data Frame [object](https://en.wikipedia.org/wiki/Object-oriented_programming) for data manipulation with integrated indexing.
      * Tools for reading and writing data between in-memory [data structures](https://en.wikipedia.org/wiki/Data_structure) and different [file formats](https://en.wikipedia.org/wiki/File_format).
      * Data alignment and integrated handling of missing data.
      * Reshaping and pivoting of data sets.
      * Label-based slicing, fancy indexing, and subsetting of large data sets.
      * Data structure column insertion and deletion.
      * Group by engine allowing split-apply-combine operations on data sets.
      * Data set merging and joining.
      * Hierarchical axis indexing to work with high-dimensional data in a lower- dimensional data structure.
      * Time series-functionality: Date range generation[[6]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-6) and frequency conversions, moving window [statistics](https://en.wikipedia.org/wiki/Statistics), moving window [linear regressions](https://en.wikipedia.org/wiki/Linear_regression), date shifting and lagging.
      * Provides data filtration.

The library is highly optimized for performance, with critical code paths written in [Cython](https://en.wikipedia.org/wiki/Cython) or [C](https://en.wikipedia.org/wiki/C_(programming_language)).[[7]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-7)

DATA FRAMES

Pandas is mainly used for [data analysis](https://en.wikipedia.org/wiki/Data_analysis). Pandas allows importing data from various file formats such as [comma-separated values](https://en.wikipedia.org/wiki/Comma-separated_values), [JSON](https://en.wikipedia.org/wiki/JSON), [SQL](https://en.wikipedia.org/wiki/SQL) database tables or queries, and [Microsoft Excel](https://en.wikipedia.org/wiki/Microsoft_Excel).[[8]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-8) Pandas allows various data manipulation operations such as merging,[[9]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-9) reshaping,[[10]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-10) selecting,[[11]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-11) as well as [data cleaning](https://en.wikipedia.org/wiki/Data_cleaning), and [data wrangling](https://en.wikipedia.org/wiki/Data_wrangling) features.

## TENSORFLOW

**TensorFlow** is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source_software) [software library](https://en.wikipedia.org/wiki/Library_(computing)) for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) and [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence). It can be used across a range of tasks but has a particular focus on [training](https://en.wikipedia.org/wiki/Types_of_artificial_neural_networks#Training) and [inference](https://en.wikipedia.org/wiki/Statistical_inference) of [deep neural networks](https://en.wikipedia.org/wiki/Deep_neural_networks)[.[4][5]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-4)

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal [Google](https://en.wikipedia.org/wiki/Google) use in research and production.[[6][7](https://en.wikipedia.org/wiki/TensorFlow#cite_note-6)[][8]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-8) The initial version was released under the [Apache](https://en.wikipedia.org/wiki/Apache_License_2.0) [License 2.0](https://en.wikipedia.org/wiki/Apache_License_2.0) in 2015.[[1][9]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-Credits-1) Google released the updated version of TensorFlow, named TensorFlow 2.0, in September 2019.[[10]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-%3A12-10)

TensorFlow can be used in a wide variety of programming languages, most notably Python, as well as JavaScript, C++, and Java.[[11]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-%3A13-11) This flexibility lends itself to a range of applications in many different sectors.



**Figure 1.7**

TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017.[[16]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-16) While the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) runs on single devices, TensorFlow can run on multiple [CPUs](https://en.wikipedia.org/wiki/Central_processing_unit) and [GPUs](https://en.wikipedia.org/wiki/GPU) (with optional [CUDA](https://en.wikipedia.org/wiki/CUDA) and [SYCL](https://en.wikipedia.org/wiki/SYCL) extensions for [general-purpose computing on graphics processing units](https://en.wikipedia.org/wiki/General-purpose_computing_on_graphics_processing_units)).[[17]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-Metz-Nov10-17) TensorFlow is available on 64-bit [Linux](https://en.wikipedia.org/wiki/Linux), [macOS](https://en.wikipedia.org/wiki/MacOS), [Windows](https://en.wikipedia.org/wiki/Windows), and mobile computing platforms including [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) and [iOS](https://en.wikipedia.org/wiki/IOS).

Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, [TPUs](https://en.wikipedia.org/wiki/Tensor_processing_unit)), and from desktops to clusters of servers to mobile and edge devices.

TensorFlow computations are expressed as [stateful](https://en.wikipedia.org/wiki/State_(computer_science)) [dataflow](https://en.wikipedia.org/wiki/Dataflow_programming) [graphs](https://en.wikipedia.org/wiki/Directed_graph). The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as [*tensors*](https://en.wikipedia.org/wiki/Tensor). During the [Google I/O](https://en.wikipedia.org/wiki/Google_I/O) [Conference](https://en.wikipedia.org/wiki/Google_I/O) in June 2016, Jeff Dean stated that 1,500 repositories on [GitHub](https://en.wikipedia.org/wiki/GitHub) mentioned TensorFlow, of which only 5 were from Google.[[18]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-1500repo%27s-18)

In December 2017, developers from Google, Cisco, RedHat, CoreOS, and CaiCloud introduced [Kubeflow](https://en.wikipedia.org/wiki/Kubeflow) at a conference. Kubeflow allows operation and deployment of TensorFlow on [Kubernetes](https://en.wikipedia.org/wiki/Kubernetes).

In March 2018, Google announced TensorFlow.js version 1.0 for machine learning in [JavaScript](https://en.wikipedia.org/wiki/JavaScript).[[19]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-19)

In Jan 2019, Google announced TensorFlow 2.0.[[20]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-20) It became officially available in Sep 2019.[[10]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-%3A12-10)

In May 2019, Google announced TensorFlow Graphics for deep learning in computer graphics.[[21]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-21)

## TENSORFLOW LITE

In May 2017, Google announced a software stack specifically for mobile development, TensorFlow Lite.[[27]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-27) In January 2019, TensorFlow team released a developer preview of the mobile GPU inference engine with OpenGL ES 3.1 Compute Shaders on Android devices and Metal Compute Shaders on iOS devices.[[28]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-28) In May 2019, Google announced that their TensorFlow Lite Micro (also known as TensorFlow Lite for Microcontrollers) and [ARM's](https://en.wikipedia.org/wiki/Arm_Holdings) uTensor would be merging.[[29]](https://en.wikipedia.org/wiki/TensorFlow#cite_note-29)

## KERAS

**Keras** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software](https://en.wikipedia.org/wiki/AI_software) library that provides a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) interface for [artificial](https://en.wikipedia.org/wiki/Artificial_neural_network) [neural networks](https://en.wikipedia.org/wiki/Artificial_neural_network). Keras acts as an interface for the [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow) library.

Up until version 2.3, Keras supported multiple backends, including [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow), [Microsoft Cognitive Toolkit](https://en.wikipedia.org/wiki/Microsoft_Cognitive_Toolkit), [Theano](https://en.wikipedia.org/wiki/Theano_(software)), and [PlaidML](https://en.wikipedia.org/wiki/PlaidML).[[2][3](https://en.wikipedia.org/wiki/Keras#cite_note-2)[][4]](https://en.wikipedia.org/wiki/Keras#cite_note-4) As of version 2.4, only [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow) is supported. Designed to enable fast experimentation with [deep neural](https://en.wikipedia.org/wiki/Deep_learning) [networks](https://en.wikipedia.org/wiki/Deep_learning), it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System),[[5]](https://en.wikipedia.org/wiki/Keras#cite_note-5) and its primary author and maintainer is [François Chollet](https://en.wikipedia.org/wiki/Fran%C3%A7ois_Chollet), a [Google](https://en.wikipedia.org/wiki/Google) engineer. Chollet is also the author of the XCeption deep neural network model.[[6]](https://en.wikipedia.org/wiki/Keras#cite_note-6)



**Figure 1.8**

FEATURES

Keras contains numerous implementations of commonly used neural-network building blocks such as layers, [objectives](https://en.wikipedia.org/wiki/Objective_function), [activation functions](https://en.wikipedia.org/wiki/Activation_function), [optimizers](https://en.wikipedia.org/wiki/Mathematical_optimization), and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code. The code is hosted on [GitHub](https://en.wikipedia.org/wiki/GitHub), and community support forums include the GitHub issues page, and a [Slack](https://en.wikipedia.org/wiki/Slack_(software)) channel.

In addition to standard neural networks, Keras has support for [convolutional](https://en.wikipedia.org/wiki/Convolutional_neural_networks) and [recurrent neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks). It supports other common utility layers like [dropout](https://en.wikipedia.org/wiki/Dropout_(neural_networks)), [batch](https://en.wikipedia.org/wiki/Batch_normalization) [normalization](https://en.wikipedia.org/wiki/Batch_normalization), and [pooling](https://en.wikipedia.org/wiki/Pooling_(neural_networks)).[[7]](https://en.wikipedia.org/wiki/Keras#cite_note-7)

Keras allows users to productize deep models on smartphones ([iOS](https://en.wikipedia.org/wiki/IOS) and [Android](https://en.wikipedia.org/wiki/Android_(operating_system))), on the web, or on the [Java Virtual Machine](https://en.wikipedia.org/wiki/Java_Virtual_Machine).[[3]](https://en.wikipedia.org/wiki/Keras#cite_note-why-keras-3) It also allows use of distributed training of deep-learning models on clusters of [Graphics processing units (GPU)](https://en.wikipedia.org/wiki/Graphics_processing_unit) and [tensor](https://en.wikipedia.org/wiki/Tensor_processing_unit) [processing units (TPU)](https://en.wikipedia.org/wiki/Tensor_processing_unit).[[8]](https://en.wikipedia.org/wiki/Keras#cite_note-8)

## ANACONDA

Anaconda is a [distribution](https://en.wikipedia.org/wiki/Software_distribution) of the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [R](https://en.wikipedia.org/wiki/R_(programming_language)) [programming languages](https://en.wikipedia.org/wiki/Programming_language) for [scientific](https://en.wikipedia.org/wiki/Scientific_computing) [computing](https://en.wikipedia.org/wiki/Scientific_computing) ([data science](https://en.wikipedia.org/wiki/Data_science), [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications, large-scale [data processing](https://en.wikipedia.org/wiki/Data_processing), [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics), etc.), that aims to simplify [package management](https://en.wikipedia.org/wiki/Package_management) and [deployment](https://en.wikipedia.org/wiki/Deployment_environment). The distribution includes data-science packages suitable for [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://en.wikipedia.org/wiki/Linux), and [macOS](https://en.wikipedia.org/wiki/MacOS). It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) in 2012.[[8]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-8) As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.[[6][7]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-l1-6)

Package versions in Anaconda are managed by the package management system [*conda*](https://en.wikipedia.org/wiki/Conda_(package_manager)).[[9]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-9) This package manager was spun out as a separate [open-source](https://en.wikipedia.org/wiki/Open_source) package as it ended up being useful on its own and for things other than Python.[[10]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-10) There is also a small, [bootstrap](https://en.wikipedia.org/wiki/Bootstrapping) version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.[[11]](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)#cite_note-11)



**Figure 1.9**

## 1.2.7.1 JUPYTER NOTEBOOK

Jupyter Notebook (formerly IPython Notebooks) is a [web-based interactive](https://en.wikipedia.org/wiki/Web_application) computational environment for creating [notebook](https://en.wikipedia.org/wiki/Notebook_interface) documents.

A Jupyter Notebook document is a browser-based [REPL](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop) containing an ordered list of input/output cells which can contain code, text (using [Markdown](https://en.wikipedia.org/wiki/Markdown)), mathematics, [plots](https://en.wikipedia.org/wiki/Plot_(graphics)) and [rich media](https://en.wikipedia.org/wiki/Interactive_media). Underneath the interface, a notebook is a [JSON](https://en.wikipedia.org/wiki/JSON) document, following a versioned schema, usually ending with the ".ipynb" extension.



**Figure 1.10**

Jupyter notebooks are built upon a number of popular [open-source](https://en.wikipedia.org/wiki/Open-source_software) libraries:

* + - * [IPython](https://en.wikipedia.org/wiki/IPython)
      * [ZeroMQ](https://en.wikipedia.org/wiki/ZeroMQ)
      * [Tornado](https://en.wikipedia.org/wiki/Tornado_(web_server))
      * [jQuery](https://en.wikipedia.org/wiki/JQuery)
      * [Bootstrap (front-end framework)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))
      * [MathJax](https://en.wikipedia.org/wiki/MathJax)

Jupyter Notebook can connect to many *kernels* to allow programming in different languages. A Jupyter kernel is a program responsible for handling various types of requests ([code execution](https://en.wikipedia.org/wiki/Execution_(computing)), [code completions](https://en.wikipedia.org/wiki/Autocomplete), inspection), and providing a reply. Kernels talk to the other components of Jupyter using [ZeroMQ](https://en.wikipedia.org/wiki/ZeroMQ), and thus can be on the same or [remote machines](https://en.wikipedia.org/wiki/Remote_computer). Unlike many other Notebook-like interfaces, in Jupyter, kernels are not aware that they are attached to a specific document, and can be connected to many clients at once. Usually kernels allow execution of only a single language, but there are a couple of exceptions.[*citation needed*] By default Jupyter Notebook ships with the IPython kernel. As of the 2.3 release[[12][13]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-releasenote23-12) (October 2014), there are 49 Jupyter-compatible kernels for many programming languages, including Python, R, Julia and Haskell.[[14]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-14)

A Jupyter Notebook can be converted to a number of [open standard](https://en.wikipedia.org/wiki/Open_standard) output formats ([HTML](https://en.wikipedia.org/wiki/HTML), [presentation slides](https://en.wikipedia.org/wiki/Presentation_slide), [LaTeX](https://en.wikipedia.org/wiki/LaTeX), [PDF](https://en.wikipedia.org/wiki/PDF), [ReStructuredText](https://en.wikipedia.org/wiki/ReStructuredText), [Markdown](https://en.wikipedia.org/wiki/Markdown), Python) through "Download As" in the web interface, via the nbconvert library[[15]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-15) or "jupyter nbconvert" command line interface in a shell. To simplify visualisation of Jupyter notebook documents on the web, the nbconvert library[[16]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-16) is provided as a service through NbViewer[[17]](https://en.wikipedia.org/wiki/Project_Jupyter" \l "cite_note-17) which can take a URL to any publicly available notebook document, convert it to HTML on the fly and display it to the user.

The notebook interface was added to IPython in the 0.12 release[[18]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-releasenote012-18) (December 2011), renamed to Jupyter notebook in 2015 (IPython 4.0 is Jupyter 1.0). Jupyter Notebook is similar to the notebook interface of other programs such as [Maple](https://en.wikipedia.org/wiki/Maple_(software)), [Mathematica](https://en.wikipedia.org/wiki/Mathematica), and [SageMath](https://en.wikipedia.org/wiki/SageMath), a computational interface style that originated with Mathematica in the 1980s.[[19]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-Somers2018-19) Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.[[19]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-Somers2018-19)

JupyterLab is a newer user interface for Project Jupyter. It offers the building blocks of the classic Jupyter Notebook (notebook, terminal, text editor, file browser, rich outputs, etc.) in a flexible user interface. The first stable release was announced on February 20, 2018.[[20]](https://en.wikipedia.org/wiki/Project_Jupyter#cite_note-20)

## ANDROID STUDIO

Android Studio is the official[[7]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-%3A0-7) [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [Google'](https://en.wikipedia.org/wiki/Google)s [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) [operating system](https://en.wikipedia.org/wiki/Operating_system), built on [JetBrains'](https://en.wikipedia.org/wiki/JetBrains) [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA) software and designed specifically for [Android development](https://en.wikipedia.org/wiki/Android_software_development).[[8]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-8) It is available for download on [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) based operating systems or as a subscription-based service in 2020.[[9][10]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-9) It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (E-ADT) as the primary IDE for native Android application development.



**Figure 1.11**

Android Studio was announced on May 16, 2013 at the [Google I/O](https://en.wikipedia.org/wiki/Google_I/O) conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014.[[11]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-11) The first stable build was released in December 2014, starting from version 1.0.[[12]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-12)

On May 7, 2019, [Kotlin](https://en.wikipedia.org/wiki/Kotlin_(programming_language)) replaced [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) as Google's preferred language for Android app development.[[13]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-13) Java is still supported, as is [C++](https://en.wikipedia.org/wiki/C%2B%2B).[[14]](https://en.wikipedia.org/wiki/Android_Studio#cite_note-14)

# CHAPTER 2

## LITERATURE REVIEW

## LITERATURE INTRODUCTION

Though convolutional neural networks (often shortened to ConvNets, CNNs, dCNNs) was not entirely new technology, it had taken the world by storm since [12]. CNNs now developed which are utilized in different digital applications were actually first observed in living organisms. In the 1950s & 1960s, Hubel & Wiesel have worked on cat and monkey which showed that their visual cortexes contained neurons that separately reciprocate to small areas of the visual field. The visual stimuli in the area of visual space affects the firing of a single neuron provided their eyes are not moving. This is its receptive field.

Likely and intercepting receptive regions have been observed in the neighbouring cells. Each hemisphere in the cortex represents the contralateral visual field. This lead to the introductionof neocognitron, delay in the time of neural networks and trainable weights. All these formed basics to the first ever documented commercial use of CNNs which dates back to 1998, with LeNet-5. LeNet-5 was designed by LeCun et al [7]. Years of researching CNNs, made it possible to recognise text character based on 32x32 pixel images.

Model consists of four main components: a low-level features network, a mid-level features network, a global features network, and a colorization network. Conceptually, these networks function as follows: First, a common set of shared low-level features are extracted from the image. Using these features, a set of global image features and mid-level image features are computed. Then, the midlevel and the global features are both fused by our proposed “fusion layer” and used as the input to a colorization network that outputs

the final chromimance map. Needless to say, this is not explicitly implemented as a sequential procedure; rather, it is realized as a single network. Note that no pre- processing nor post- processing is done: it is all computed in a single step. Additionally, as a side product of our approach, we can also perform classification of the scene. While the global features are computed using fixed-sized images, our novel approach for fusing the global and local features allows our model to be run on input images of arbitrary resolutions, unlike most Convolutional Neural Networks [13]. Due to the separation between the global and local features, it is possible to use global features computed on one image in combination with local features computed on another image, to change the style of the resulting colorization.

In the past few years, several papers have successfully solved the problem of facial expression recognition So far, several tasks have been completed for real- time emotion, gender, age classification. So, Some state of artwork relevant to the proposed work is discussed in this section. Md jashim uddin Dr. paresh Chandra Barman, khandaker Takdir Ahmed et al.[1] proposed a detection system using

CNN model which can achieved 95% accuracy rate in age, gender detection with IMDB-WIKI dataset and 66% accuracy achieved in emotion detection with FER dataset.

## LITERATURE REVIEW

Since then, CNNs have dominated this area of computer vision and surpassed results obtainable by other machine learning methods. But the large scale application of CNNs were not possible until a decade later. In 2012, when the CNN based model entry of Alex Krizhevsky et al and their AlexNet in a ImageNet Large Scale Visual Recognition Challenge won that year’s image classification challenge by a significant margin, it took most of the computer vision research community by surprise and lit a huge amount of interest.

This event made CNN a staple in the computer vision field and many others. Countless problems, such as image recognition, facial recognition, video sequence tracking, automatic image segmentation, handwriting to text conversion, natural language processing have been made possible to solve easily with the help of CNN. CNNs have proven to be functioning as automatic data encoders i.e., they can learn very complex mappings of inputs to outputs from huge amounts of data.

The convolutional neural network for this python project has 3 convolutional layers:

* + - Convolutional layer; 96 nodes, kernel size 7
    - Convolutional layer; 256 nodes, kernel size 5
    - Convolutional layer; 384 nodes, kernel size 3

It has 2 fully connected layers, each with 512 nodes, and a final output layer of softmax type.

To go about the python project, we’ll:

* + - Detect faces
    - Classify into Male/Female
    - Classify into one of the 8 age ranges
    - Put the results on the image and display it



**Figure 2.1**

For face detection, we have a .pb file- this is a protobuf file (protocol buffer); it holds the graph definition and the trained weights of the model. We can use this to run the trained model. And while a .pb file holds the protobuf in binary format, one with the

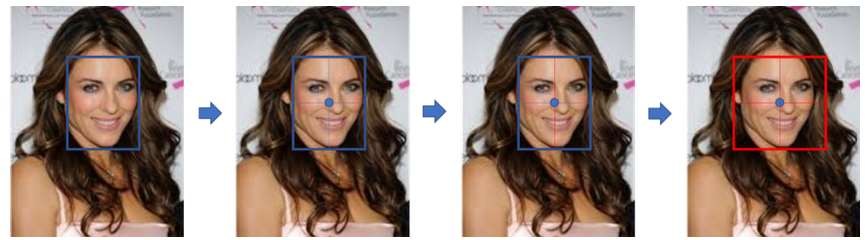
.pbtxt extension holds it in text format. These are TensorFlow files. For age and gender, the .prototxt files describe the network configuration and the .caffemodel file defines the internal states of the parameters of the layers.

1. We use the argparse library to create an argument parser so we can get the image argument from the command prompt. We make it parse the argument holding the path to the image to classify gender and age for.
2. For face, age, and gender, initialize protocol buffer and model.
3. Initialize the mean values for the model and the lists of age ranges and genders to classify from.
4. Now, use the readNet() method to load the networks. The first parameter holds trained weights and the second carries network configuration.
5. Let’s capture video stream in case you’d like to classify on a webcam’s stream. Set padding to 20.
6. Now until any key is pressed, we read the stream and store the content into the names hasFrame and frame. If it isn’t a video, it must wait, and so we call up waitKey() from cv2, then break.
7. Let’s make a call to the highlightFace() function with the faceNet and frame parameters, and what this returns, we will store in the names resultImg and faceBoxes. And if we got 0 faceBoxes, it means there was no face to detect. Here, net is faceNet- this model is the DNN Face Detector and holds only about 2.7MB on disk.

Create a shallow copy of frame and get its height and width. Create a blob from the shallow copy.

Set the input and make a forward pass to the network.

faceBoxes is an empty list now. for each value in 0 to 127, define the confidence (between 0 and 1). Wherever we find the confidence greater than the confidence threshold, which is 0.7, we get the x1, y1, x2, and y2 coordinates and append a list of those to faceBoxes.

Then, we put up rectangles on the image for each such list of coordinates and return two things: the shallow copy and the list of faceBoxes.

**Figure 2.2**

1. But if there are indeed faceBoxes, for each of those, we define the face, create a 4- dimensional blob from the image. In doing this, we scale it, resize it, and pass in the mean values.
2. We feed the input and give the network a forward pass to get the confidence of the two class. Whichever is higher, that is the gender of the person in the picture.
3. Then, we do the same thing for age.
4. We’ll add the gender and age texts to the resulting image and display it with imshow().

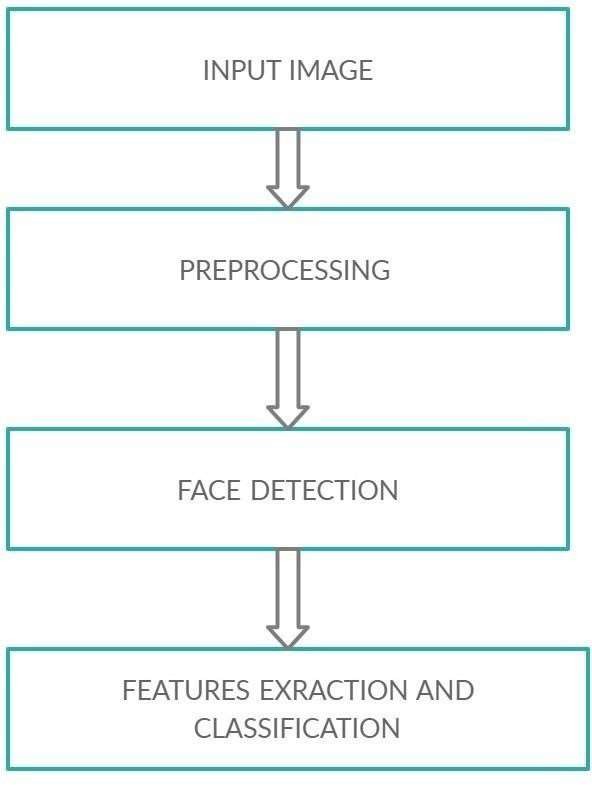
# **CHAPTER 3**

## INFERENCES DRAWN FROM LITERATURE

### 3.1 INFERENCES DRAWN OUT OF LITERATURE SURVEY

The primary objective of the proposed methodology is to recognize the gender and age range with emotion from the human face images utilizing the set of facial features in real-time

application. Feature extraction from face images is an important part of this method. In here, Fig-13 represents to show the flow of step to this methodology.



**Figure 3.1**

1. **PRE-PROSSESSING**

The pre-processing stage can improve the quality of the input image and find the data of interest by laminating noise and smoothing the image. It eliminates image redundancy without image details. Pre-processing also includes filtering and normalizing the image to produce a uniform size and rotated image.

1. **FACE DETECTION**

The face detection process is to extract the face area from the background of Input images with various lighting conditions and complex backgrounds can be confusing and fail to recognize these expressions. It involves segmentation and extraction of facial features from the uncontrolled background.

**(III)FEATURE EXTRACTION**

In object detection feature extraction plays important role. Here, it includes shapes, movement, color, the texture of the facial image. It extracts meaningful information of an image compared to the original image. Feature extraction greatly reduces information of image, which has advantages in storage.

**(IV)FEATURE CLASSIFICATION**

The classification stage recognizes facial images and group them according to certain classes and helps them skilled recognition. Classification is a complicated process because it can be playing role in many areas. It classification stage also can called feature selection stage, which is deals with exchanges the retain essential information and connect them in certain parameters.

**(V) POPULAR FEATURE EXTRACTION AND CLASSIFICATION METHODS**

##### SVM

It is a machine learning algorithm which is used for classification and regression. But usually, they are used for classification problems. SVM was first introduced in the

1960s, but it was perfected in 1990. Compared with other machine learning algorithms, SVM has its unique implementation. The model is basically a representation of different classes in a hyper plane of a multidimensional space . SVM will generate hyper planes in an iterative manner, which can minimize errors. The goal of SVM is to divide the data set into several categories to find the maximum edge hyper plane (MMH).

##### LBP

The Local Binary Pattern was used for texture classification, later it was applied in other application. LBP assigns image pixel by the neighborhood (p) each pixel values

within the radius (R), represented by (gp) taking the central value (gc) as the threshold and converting its threshold into a given decimal number label by eq. The main property of LBP is its computational simplicity, which make it analysing possible a image in real-time.

##### HOG

Histogram of oriented gradient is feature descriptor. Feature descriptor is representations of images or Image patches, which simplify the process of extracting useful information from images and discarding redundant information. Which was proposed for object detection in various machine learning application also computer vision. HOG descriptor mainly focused on the structure of an image or object by counting the localized portion. It is a feature extraction dense method for images. Dense means that it extracts features from all positions in the image (or regions of interest in the image),instead of only key points like SIFT in the local neighborhood.

##### PCA

PCA algorithm is an eigenvector method to model linear variation of high dimensional data. Principal component analysis (PCA) is the most popular appearance-based statistical method used mainly for dimensionality reduction in

compression and recognition. The PCA technique is introduced by Kirby and Sirovich in 1988.This technique are used for eliminate the dimension of a face space data. The

reduction of dimension helps to remove the non- essential information from input images to recognize the face. The main face images can be introduced as a feature vector or weighted sum of the Eigen’s faces and stored in one – dimensional array. The PCA requires full frontal face image to be presented each time to give the good performance. The advantage of PCA is it can reduce the required data to identify individuals to 1/1000 of the displayed data.

##### Viola-Jones Algorithm

The viola-Jones algorithm mainly used for object detection work that can supply a competitive rate of object detection in real-time application. Also it can be trained to detect the object classes also face detection. Three major contributions/phase of this algorithm are feature extraction, classification using boosting , Multi- scale detection algorithm . The main advantage of viola-jones algorithm, it is extremely fast feature computation & effective feature selection.

# **CHAPTER 4**

## PROBLEM STATEMENT AND SOLUTION APPROACH

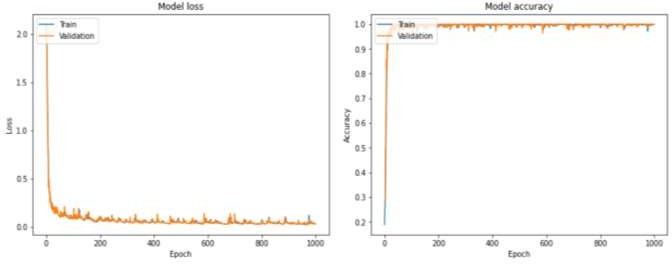
### 4.1 PROBLEMS WITH EXISTING EMOTION, AGE & GENDER DETECTION

Deep Learning has found huge applications in the fields of Computer vision. Some of the most important applications of computer vision are in the fields that deal with facial data. Face Detection and recognition are being widely used in security-based applications. If you want to explore these two areas, feel free to go through:

1. [**Face Detection**](https://levelup.gitconnected.com/a-small-gui-application-based-on-computer-vision-and-python-imaging-libraries-using-opencv-e07b3f8c06c1): In this article, I have talked about an application based on face detection, and
2. **Face Recognition:** This article talks about how we can implement a security mechanism using face recognition.

Whenever a model gives a 100% accuracy on test data, we need to check the training accuracy, if that is also 100%. It means the model is actually overfitting and the test set is having a very close distribution to the train set. So, it is showing great results. I think in these circumstances, it’s better to use cross-validation to get the correct intuition of how the model actually works.

Let’s continue with the evaluation.



**Figure 4.1**

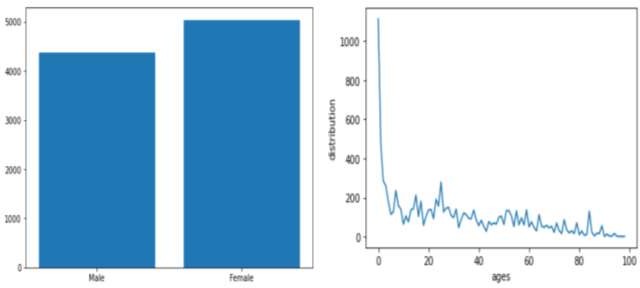
The two curves show the learning of the model. The first curve shows the loss function decrease and the second shows the accuracy growth with epochs.

#### Age and Gender Prediction

We will use Kaggle’s [UTKFace Dataset](https://www.kaggle.com/jangedoo/utkface-new) for predicting age and Gender.

#### Data Pre-processing

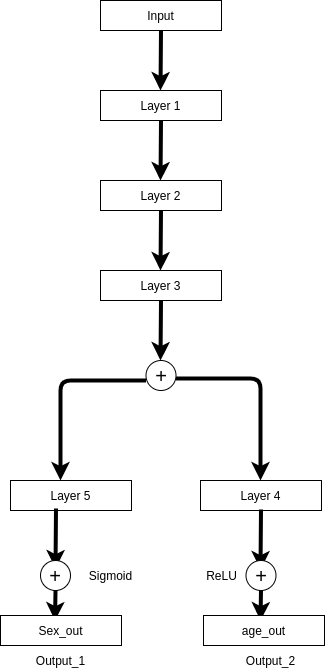
Here I have used the dataset having 9780 files. It has 9780 images of faces belonging to both males and females with ages ranging from 0 to 116. Each image has labels that show the corresponding age and gender. Male is given by 0 and Female is given by 1.



**Figure 4.2**

The first bar graph shows the distribution of gender. It seems well balanced. The second line graph shows the variation of samples of different ages. We can see that the samples with age less than 40 is much more than the number of samples with age more than 40. This creates a skewness in the train set distribution.

We have seen in this case, we will actually need to predict both age and gender using the same model.

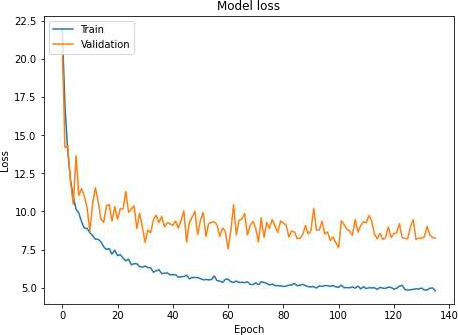


**Figure 4.3**

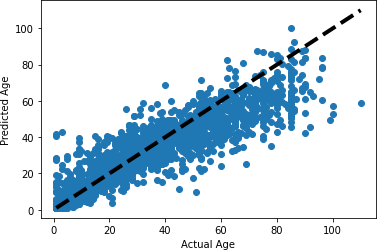
The above is a schematic diagram of our model. After the ‘flatten’ layer we are going to use two different dense layers and dropouts corresponding to the corresponding outputs. Now, gender prediction is a classification problem, while age prediction is a regression problem, so, we will use sigmoid as output activation for gender prediction and ReLU as the activation function for Age prediction. Similarly, we will use ‘binary cross-entropy’ as the loss function for gender and ‘mean absolute error’ as the loss function for the age.

The model gives an accuracy of 82% for the gender classification.

#### Evaluation

Let us look at the model’s loss curve.

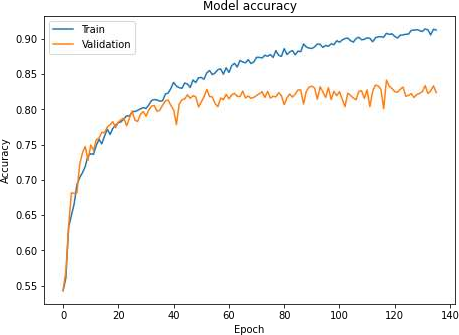
**Figure 4.4**

This is the generated loss curve for our model. Let’s look at the evaluation for age prediction:

**Figure 4.5**

The above curve shows the model traced linear regression line in black and the blue dots show the distribution of test samples. So, we can see our predicted line almost passes through the middle of the distribution. Above the age of 80, there were very few samples, so, maybe owing to that there our model didn’t perform so well.

Evaluation for gender prediction:



**Figure 4.6**

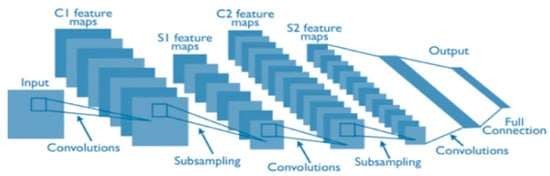
The above curve shows the increase in gender accuracy with epochs.

## CNN ARCHITECTURE FOR FACIAL RECOGNITION

Face recognition (FR) is defined as the process through which people are identified using facial images. This technology is applied broadly in biometrics, security information, accessing controlled areas, keeping of the law by different enforcement bodies, smart cards, and surveillance technology. The facial recognition system is built using two steps. The first step is a process through which the facial features are picked up or extracted, and the second step is pattern classification. Deep learning, specifically the convolutional neural network (CNN), has recently made commendable progress in FR technology. This paper investigates the performance of the pre-trained CNN with multi-class support vector machine (SVM) classifier and the performance of transfer learning using the AlexNet model to perform classification. The study considers CNN architecture, which has so far recorded the best outcome in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in the past years, more specifically, AlexNet and ResNet-50. In order to determine performance optimization of the CNN algorithm, recognition accuracy was used as a determinant. Improved classification rates were seen in the comprehensive experiments that were completed on the various datasets of ORL, GTAV face, Georgia Tech face, labelled faces in the wild (LFW), frontalized labeled faces in the wild (F\_LFW), YouTube face, and FEI faces. The result showed that our model achieved a higher accuracy compared to most of the state-of-the-art models. An accuracy range of 94% to 100% for models with all databases was obtained. Also, this was obtained with an improvement in recognition accuracy up to 39%.

#### CNNs Preliminaries:

Convolutional neural networks were initially proposed by LeCun in [[23](https://www.mdpi.com/2076-3417/9/20/4397/htm#B23-applsci-09-04397)]. They have been successfully applied to computer vision problems, such as hand-written digit recognition [[24](https://www.mdpi.com/2076-3417/9/20/4397/htm#B24-applsci-09-04397)]. CNNs have recently grown in popularity in the field of pattern classification. CNNs have outperformed traditional computer vision methods in image classification. A convolutional neural network is a sort of artificial neural network (ANN) inspired by the performance of visual recognition of objects by animals and human beings’ cortex, which is used for applications including systems recommender [[25](https://www.mdpi.com/2076-3417/9/20/4397/htm#B25-applsci-09-04397)], video and image recognition [[26](https://www.mdpi.com/2076-3417/9/20/4397/htm#B26-applsci-09-04397)], and natural processing of languages [[27](https://www.mdpi.com/2076-3417/9/20/4397/htm#B27-applsci-09-04397),[28](https://www.mdpi.com/2076-3417/9/20/4397/htm#B28-applsci-09-04397)]. CNN architectures makes the explicit assumption that the inputs are images, which allows encoding of certain properties into the architecture. Neurons in CNN are 3D filters that activate depending on their inputs. They are connected only to a small region, called the receptive field [[29](https://www.mdpi.com/2076-3417/9/20/4397/htm#B29-applsci-09-04397)], of a previous neuron’s activations. They compute a convolution operation between the connected inputs and their internal parameters, and they get activated depending on their output and a non-linearity function [[30](https://www.mdpi.com/2076-3417/9/20/4397/htm#B30-applsci-09-04397)].



**Figure 4.7**

Convolutional neural network layers are divided into three types: the convolutional, pooling and, fully connected layers. Each layer plays a different role. The CNN architecture is shown in the [Figure.](https://www.mdpi.com/2076-3417/9/20/4397/htm#fig_body_display_applsci-09-04397-f001)

Convolutional layer: Convolutional layer is known as the elemental development block for CNN. In CNN technology, it is crucial to understand that the layers’ parameters are made up of a set of learnable filters or neurons. These filters have a small receptive field, but they go all the way through the input volume. In the forward pass process, each individual filter goes across the width and height of the input volume, calculating the dot product from the filter entries and the input. The product of this computation is a two-dimensional activation map of that filter. Through this, the network learns filters created when it senses some particular type of feature at a spatial location within the feature-map input **X**, generating a feature map of weighted summations **Y**. Each of the neurons computes convolutions with small regions in **X**, shown in Equation (1) [[23](https://www.mdpi.com/2076-3417/9/20/4397/htm#B23-applsci-09-04397)].

yi=bi+ ∑xi∈Xwij∗xi

where yj ∈ Y, j = 1, 2, ..., D. D is the depth of the convolutional layer, and each filter **wij** is a 3D matrix of size [F × F × Cx]. Its size is determined by a chosen receptive field (**F**), and its feature-map input’s depth (**Cx**); for example, if the receptive field is fivepixels and the feature-map input **X** is a [32 × 32 × 3] RGB image, then the filter’s size will be [5,5,3]. The filter’s size represents the number of weights that a neuron has connecting to a region in the input. The convolutional layer has the advantage of using the same neurons for each pixel in the layer to improve the system’s performance. In addition, this results in the reduction of the footprint’s memory making it efficient.

#### Pooling layers:

Pooling layers are responsible for regulating the width by height dimensions by reducing the input volume spatial dimensions for the next convolutional layer without affecting the dimensional depth of the volume. The process performed by the pooling layer is also known as down-sampling or sub-sampling because the decrease of size results in simultaneous information loss that benefits the network. The reduction becomes less computational as the information progresses to the next pooling layers, and it also works against over-fitting. The most common strategies used in the pooling layer networks are max-pooling and average-pooling. In [[31](https://www.mdpi.com/2076-3417/9/20/4397/htm#B31-applsci-09-04397)], a comprehensive theoretical analysis of the max pooling and average pooling is generated, whereas in [[32](https://www.mdpi.com/2076-3417/9/20/4397/htm#B32-applsci-09-04397)] it has shown that max pooling can result in faster convergence of information, and the network picks the high-ranking features in the image thus enhancing generalization. Also, pooling layer possesses other variations such as stochastic pooling [[33](https://www.mdpi.com/2076-3417/9/20/4397/htm#B33-applsci-09-04397)], spatial pyramid pooling [[34](https://www.mdpi.com/2076-3417/9/20/4397/htm#B34-applsci-09-04397)], and def-pooling [[35](https://www.mdpi.com/2076-3417/9/20/4397/htm#B35-applsci-09-04397)] that serves marked purposes.

Fully connected layers: Fully connected layers (FC) are where the levels of high reasoning are carried out. The filters and neurons in this layer are connected to all the activation in the previous layers, resulting in full connections as their name implies. The calculations in this level are done through the multiplication of matrix followed by the bias offset. FC layer goes through a process that converts the 2D feature map to the 1D feature vector. In addition, the vector formed in this process is either classified as classes for classification [[36](https://www.mdpi.com/2076-3417/9/20/4397/htm#B36-applsci-09-04397)] or the feature vector undergoes further processing[[37](https://www.mdpi.com/2076-3417/9/20/4397/htm#B37-applsci-09-04397)].

#### CNN Pre-Trained Models

In the convolution neural network, face representation extensively affects the performance of the FR system and has also become a focus of attention in the current FR research. In this study, we employed two pre-trained convolution neural networks. These networks were AlexNet and ResNet-50. These pre-trained CNN networks have been used to extract suitable image features and utilize them in the classification stage.

#### AlexNet

AlexNet, introduced by Krizhevsky et al. [[36](https://www.mdpi.com/2076-3417/9/20/4397/htm#B36-applsci-09-04397)], was the first CNN to win the ImageNet challenge in 2012, with a top 5 error of 16.4%. The use of rectified linear units (ReLUs) was also introduced in AlexNet. It includes five convolutional layers, three max pool layers, and three fully connected layers. This architecture uses a [227 × 227 × 3] image as an input. In AlexNet, a 4096-dimensional feature vector represents the 227 × 227 image.

#### ResNet-50

ResNet or deep residual networks [[38](https://www.mdpi.com/2076-3417/9/20/4397/htm#B38-applsci-09-04397)], developed by Kaiming He et al., is one of the networks that are considered the latest and greatest in terms of using convolutional neural networks for image recognition. ResNet won the ImageNet Large-Scale Visual Recognition Challenge in 2015 (ILSVRC-15) with a top 5 error of 3.57%. In our study, we used ResNet-50. It includes five convolutional layers. ResNet-50 architecture uses a [224 × 224 × 3] image as an input.

#### Related Work

Recently, convolutional neural networks have made great achievements in resolving different image processing problems for FR applications. Yu et al. [[39](https://www.mdpi.com/2076-3417/9/20/4397/htm#B39-applsci-09-04397)] proposed a novel method called biometric quality assessment (BQA) for face images, investigating its applicability in FR applications. They used a light CNN with the max-feature-map units to make the BQA method more robust to noisy labels. Their studies have been explored further through experiments on the YouTube, FLW, and CASIA databases.

The results of their experiments show a high degree of effectiveness of their proposed BQA method.

Sun et al. [[40](https://www.mdpi.com/2076-3417/9/20/4397/htm#B40-applsci-09-04397)] conducted research on the potential use of hybrid deep learning for face verification. The researchers used, in particular, an experimental design involving a hybrid convolutional network (ConvNet) based on the restricted Boltzmann machine (RBM) model for purposes of face verification. The results showed that the hybrid deep learning achieved an excellent performance when it came to face verification as compared to the other commonly used methods. Singh and Om [[41](https://www.mdpi.com/2076-3417/9/20/4397/htm#B41-applsci-09-04397)] used the deep convolutional neural network to identify the specific individuals from newborn infant datasets. The datasets used for their research contained 210 infants. Each infant consisted of 10 images with different facial expressions. They inferred that increasing the number of hidden layers does not increase the identification accuracy, and they also found that using a greater number of convolution layers tended to fit over the model, and that may also decrease the performance. Guo et al. [[42](https://www.mdpi.com/2076-3417/9/20/4397/htm#B42-applsci-09-04397)] put forward a CNN-based model that meant to use both visible light image and near-infrared image to obtain facial recognition. Additionally, they created an adaptive score fusion strategy whose purpose was to significantly improve the performance. In comparison to the traditional deep learning procedure, this scheme can develop a robust face feature extraction model. When in use practically, it is robust to illumination variation. The researchers conducted a validity testing through various datasets. The results of the experiments indicated that the new model achieved enhanced performance. Hu et al. [[43](https://www.mdpi.com/2076-3417/9/20/4397/htm#B43-applsci-09-04397)] investigated the performance of CNN on 2D and 3D FR systems. In their research, two CNN models were constructed—CNN-1 and CNN-2. The experiments of the study found a better accuracy on the CNN-2 model on both 2D and 3D face recognition. Also, the experimental results for CNN-2 showed an accuracy of 85.15% with the FRGCv2.0 dataset and 95% with the AT&T dataset. The results of their research showed that the CNN model is effective for facial images in 2D and 3D.

G. P. Nam et al. [[44](https://www.mdpi.com/2076-3417/9/20/4397/htm#B44-applsci-09-04397)] proposed a CNN model named PSI-CNN for face recognition.

The PSI-CNN model extracts untrained features from the image, then fuses these features with original feature maps. The results of the experiments are shown in terms of matching accuracy, with the model outperforming the model derived from the VGG- Face model. Also, PSI-CNN was able to maintain stable performance when tested on low-resolution images acquired from CCTV cameras. In case of change in image resolution and quality, PSI-CNN is robust. P. S. Prasad et al. [[7](https://www.mdpi.com/2076-3417/9/20/4397/htm#B7-applsci-09-04397)] studied deep learning- based face representation for different face recognition challenges, such as misalignment, lower and upper face occlusions, illuminations, and different angles of head poses. They used two approaches—VGG-Face and lightened CNN. The AR face database used to evaluate the approaches’ results of the study showed that deep learning approaches provide a good result in terms of recognizing faces and pre-processing. Suleman Khan et al. [[45](https://www.mdpi.com/2076-3417/9/20/4397/htm#B45-applsci-09-04397)] proposed a system for face recognition using portable smart glasses based on CNN. The detection process was performed using Haar-like features. The method archived detection rate at 98% using 3099 features. They used transfer learning from AlexNet for trained CNN model. The experiments of the study were conducted using 2500 images in a class. The results of the study showed that the accuracy of the system proposed was 98.5%. Chen Qin et al. [[46](https://www.mdpi.com/2076-3417/9/20/4397/htm#B46-applsci-09-04397)] proposed a recognition algorithm based on deep CNNs. The algorithm contained face detection, face alignment, and feature extraction. The deep CNNs VGG16 was used to extract facial features. The experiments used the images of five angles (left, right, front, overlook, and look up). The experiment results showed that the algorithm achieved well on recognizing faces for cases of various poses in an indoor environment.

Menotti et al. [[47](https://www.mdpi.com/2076-3417/9/20/4397/htm#B47-applsci-09-04397)] investigated two deep presentation processes composed of learning from CNN and weight adjustment, and iris spoof detection and fingerprints, the latter of which was the best approach for face detection and imaging. They admitted that indeed there was very limited experimental knowledge on the biometric spoofing at the sensors for deriving an outstandingly comprehensive spoofing detection framework for the face, iris, and fingerprint variations based on two major deep learning approaches. These approaches included a focus on learning of the weights of the networks through back propagation and learning of suitable convolutional network architectures for each of the CNN’s domains. Simón et al. [[48](https://www.mdpi.com/2076-3417/9/20/4397/htm#B48-applsci-09-04397)] proposed a method on how to improve facial recognition. A multimodal facial recognition using the CNN’s systems is a good approach to facial recognition. They fused the modality-specific CNNs with histograms of Gabor ordinal measures (HOGOMs), local binary patterns (LBP), histograms of oriented gradients, and Haar-like features. The result of the approach significantly reduced the recognition error rate. Using more sophisticated computer systems will improve the process of deep learning. Similarly, there has been research-applied CNN, but this has been used on newborn FR [[41](https://www.mdpi.com/2076-3417/9/20/4397/htm#B41-applsci-09-04397)].

Another study by Parkhi et al. [[49](https://www.mdpi.com/2076-3417/9/20/4397/htm#B49-applsci-09-04397)] proposed VGG-Face system, which applied a 16- layer CNN trained on 2.6 million images and was shown to achieve even better results. Zhenyao et al. [[50](https://www.mdpi.com/2076-3417/9/20/4397/htm#B50-applsci-09-04397)] used a deep network to “warp” faces into a canonical frontal view, after which the system learned CNN, which in turn classified the particular faces as those that matched a particular identity. For face verification, PCA on the network output in conjunction with an ensemble of SVMs was used. Also, Guo et al. [[51](https://www.mdpi.com/2076-3417/9/20/4397/htm#B51-applsci-09-04397)] proposed an FR system based on CNN for feature extraction and SVM as a classifier. In order to enhance the performance of CNN, they used techniques for optimization to be training CNN. The model spends less time for training and gains a high recognition rate. The experiments in the study were conducted on the basis of the FERET and ORL dataset. The results of the experiments showed the system obtained and demonstrated a high recognition rate and less training time.

Even though CNNs have been used in FR technology dating back to 1997 [[52](https://www.mdpi.com/2076-3417/9/20/4397/htm#B52-applsci-09-04397)], there are major improvements in that there are massive image datasets that are available and have revealed their power. A work used representatively for this approach is Deep-Face [[13](https://www.mdpi.com/2076-3417/9/20/4397/htm#B13-applsci-09-04397)], whereby the researchers trained an eight-layer CNN architecture. These layers were distributed—the first three were conventional convolution-pooling-convolution layers, followed three layers that were locally connected and then two fully connected layers. It is crucial to note that the pooling layers had an effect of making learned features robust to local transformations but caused a miss in local texture details. The pooling layers were critical for object recognition because these objects were not properly aligned. It is, however, important to note that face images should be well- trained before CNN training. Deep-Face is trained on a large database of faces, which consists of 4 million facial images of 4000 subjects. The same study also proposed a 3D alignment approach that uses an affine camera model. This has realized an exemplary performance in both LFW and YouTube face benchmarks. Y. Sun et al. [[53](https://www.mdpi.com/2076-3417/9/20/4397/htm#B53-applsci-09-04397)] proposed a CNN-based approach called DeepID. It is unlike DeepFace, which used one big CNN; instead, DeepID learns by training an ensemble of small CNNs and through building network fusion. In DeepID, each network includes four convolutional layers, three max-pooling layers, and two fully connected layers. DeepID achieved 97.45% accuracy on the LFW dataset. An extension work of DeepID is DeepID2 [[54](https://www.mdpi.com/2076-3417/9/20/4397/htm#B54-applsci-09-04397)]. It trains CNN for verification and identification. DeepID2+ [[55](https://www.mdpi.com/2076-3417/9/20/4397/htm#B55-applsci-09-04397)] has been proposed to improve the performance of DeepID and DeepID2. DeepID2+ net uses a larger training set than DeepID and DeepID2, and also improves the number of filters of all layers. DeepID2+ found that the face representations learned are sparse, selective, and robust. Recently, the success of deep convolutional neural networks has enhanced the performance of the FR model. Lu et al. [[56](https://www.mdpi.com/2076-3417/9/20/4397/htm#B56-applsci-09-04397)] proposed a novel CNN-based approach called the Deep Coupled ResNet (DCR) model, which consists of one trunk network and two branch networks. The trunk network is used to extract discriminative features for face images of different resolutions. Then, the two branch networks are used to transform high- resolution (HR) images and corresponding images of the targeted low resolution (LR). The DCR model achieves better performance than the state-of-the-art models on the LFW and SCface datasets.

The reviewed related work shows that convolutional neural networks have been applied in different applications for feature extraction and classification, and many databases have been created to be used for this purpose. [Table 3](https://www.mdpi.com/2076-3417/9/20/4397/htm#table_body_display_applsci-09-04397-t003) summarizes the convolutional neural network application for face modality and face databases used in the related works presented in this section. Some research focuses on studying FR using convolutional neural networks, and they train the networks from scratch. In addition, some studies have conducted an experiment on one or two datasets. In our study, we used pre-trained convolutional neural networks and conducted all our experiment on seven datasets.

G. P. Nam et al. [[44](https://www.mdpi.com/2076-3417/9/20/4397/htm#B44-applsci-09-04397)] proposed a CNN model named PSI-CNN for face recognition.

The PSI-CNN model extracts untrained features from the image, then fuses these features with original feature maps. The results of the experiments are shown in terms of matching accuracy, with the model outperforming the model derived from the VGG- Face model. Also, PSI-CNN was able to maintain stable performance when tested on low-resolution images acquired from CCTV cameras. In case of change in image resolution and quality, PSI-CNN is robust. P. S. Prasad et al. [[7](https://www.mdpi.com/2076-3417/9/20/4397/htm#B7-applsci-09-04397)] studied deep learning- based face representation for different face recognition challenges, such as misalignment, lower and upper face occlusions, illuminations, and different angles of head poses. They used two approaches—VGG-Face and lightened CNN. The AR face database used to evaluate the approaches’ results of the study showed that deep learning approaches provide a good result in terms of recognizing faces and pre-processing. Suleman Khan et al. [[45](https://www.mdpi.com/2076-3417/9/20/4397/htm#B45-applsci-09-04397)] proposed a system for face recognition using portable smart glasses based on CNN. The detection process was performed using Haar-like features. The method archived detection rate at 98% using 3099 features. They used transfer learning from AlexNet for trained CNN model. The experiments of the study were conducted using 2500 images in a class. The results of the study showed that the accuracy of the system proposed was 98.5%. Chen Qin et al. [[46](https://www.mdpi.com/2076-3417/9/20/4397/htm#B46-applsci-09-04397)] proposed a recognition algorithm based on deep CNNs. The algorithm contained face detection, face alignment, and feature extraction. The deep CNNs VGG16 was used to extract facial features. The experiments used the images of five angles (left, right, front, overlook, and look up). The experiment results showed that the algorithm achieved well on recognizing faces for cases of various poses in an indoor environment.

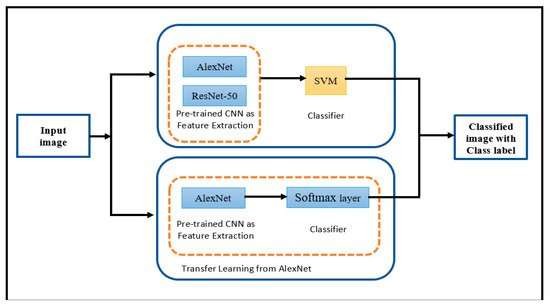
Menotti et al. [[47](https://www.mdpi.com/2076-3417/9/20/4397/htm#B47-applsci-09-04397)] investigated two deep presentation processes composed of learning from CNN and weight adjustment, and iris spoof detection and fingerprints, the latter of which was the best approach for face detection and imaging. They admitted that indeed there was very limited experimental knowledge on the biometric spoofing at the sensors for deriving an outstandingly comprehensive spoofing detection framework for the face, iris, and fingerprint variations based on two major deep learning approaches. These approaches included a focus on learning of the weights of the networks through back propagation and learning of suitable convolutional network architectures for each of the CNN’s domains. Simón et al. [[48](https://www.mdpi.com/2076-3417/9/20/4397/htm#B48-applsci-09-04397)] proposed a method on how to improve facial recognition. A multimodal facial recognition using the CNN’s systems is a good approach to facial recognition. They fused the modality-specific CNNs with histograms of Gabor ordinal measures (HOGOMs), local binary patterns (LBP), histograms of oriented gradients, and Haar-like features. The result of the approach significantly reduced the recognition error rate. Using more sophisticated computer systems will improve the process of deep learning. Similarly, there has been research-applied CNN, but this has been used on newborn FR [[41](https://www.mdpi.com/2076-3417/9/20/4397/htm#B41-applsci-09-04397)].

G. P. Nam et al. [[44](https://www.mdpi.com/2076-3417/9/20/4397/htm#B44-applsci-09-04397)] proposed a CNN model named PSI-CNN for face recognition.

The PSI-CNN model extracts untrained features from the image, then fuses these features with original feature maps. The results of the experiments are shown in terms of matching accuracy, with the model outperforming the model derived from the VGG- Face model. Also, PSI-CNN was able to maintain stable performance when tested on low-resolution images acquired from CCTV cameras. In case of change in image resolution and quality, PSI-CNN is robust. P. S. Prasad et al. [[7](https://www.mdpi.com/2076-3417/9/20/4397/htm#B7-applsci-09-04397)] studied deep learning- based face representation for different face recognition challenges, such as misalignment, lower and upper face occlusions, illuminations, and different angles of head poses. They used two approaches—VGG-Face and lightened CNN. The AR face database used to evaluate the approaches’ results of the study showed that deep learning approaches provide a good result in terms of recognizing faces and pre-processing. Suleman Khan et al. [[45](https://www.mdpi.com/2076-3417/9/20/4397/htm#B45-applsci-09-04397)] proposed a system for face recognition using portable smart glasses based on CNN. The detection process was performed using Haar-like features. The method archived detection rate at 98% using 3099 features. They used transfer learning from AlexNet for trained CNN model. The experiments of the study were conducted using 2500 images in a class. The results of the study showed that the accuracy of the system proposed was 98.5%. Chen Qin et al. [[46](https://www.mdpi.com/2076-3417/9/20/4397/htm#B46-applsci-09-04397)] proposed a recognition algorithm based on deep CNNs. The algorithm contained face detection, face alignment, and feature extraction. The deep CNNs VGG16 was used to extract facial features. The experiments used the images of five angles (left, right, front, overlook, and look up). The experiment results showed that the algorithm achieved well on recognizing faces for cases of various poses in an indoor environment.

Menotti et al. [[47](https://www.mdpi.com/2076-3417/9/20/4397/htm#B47-applsci-09-04397)] investigated two deep presentation processes composed of learning from CNN and weight adjustment, and iris spoof detection and fingerprints, the latter of which was the best approach for face detection and imaging. They admitted that indeed there was very limited experimental knowledge on the biometric spoofing at the sensors for deriving an outstandingly comprehensive spoofing detection framework for the face, iris, and fingerprint variations based on two major deep learning approaches. These approaches included a focus on learning of the weights of the networks through back propagation and learning of suitable convolutional network architectures for each of the CNN’s domains. Simón et al. [[48](https://www.mdpi.com/2076-3417/9/20/4397/htm#B48-applsci-09-04397)] proposed a method on how to improve facial recognition. A multimodal facial recognition using the CNN’s systems is a good approach to facial recognition. They fused the modality-specific CNNs with histograms of Gabor ordinal measures (HOGOMs), local binary patterns (LBP), histograms of oriented gradients, and Haar-like features. The result of the approach significantly reduced the recognition error rate. Using more sophisticated computer systems will improve the process of deep learning. Similarly, there has been research-applied CNN, but this has been used on newborn FR [[41](https://www.mdpi.com/2076-3417/9/20/4397/htm#B41-applsci-09-04397)].

#### Methodology and Experiments

The main goal of this study was to investigate the FR performance through convolutional neural networks. For our system, we followed two approaches, as shown in the figure.

**Figure 4.8** An overview of system approaches.

First approach: Applying the pre-trained CNN for extracting features and support vector machine (SVM) for classification.

Method 1: Pre-trained CNN AlexNet with SVM. Method 2: Pre-trained CNN ResNet-50 with SVM.

Second approach: Applying transfer learning from AlexNet model for extracting features and classification.

In our study, we followed the following stages. First, the pre-processing stage, in which we resized each image to a suitable size for each CNN model and converted any grey images to RGB images. In the second stage, face representation, we employed two pre- trained convolution neural networks. These networks were AlexNet and ResNet-50. CNN networks have been used to extract suitable image features and utilize them in the following classification stage. Finally, the process of classifying faces occurred with different convolutional neural networks. First, we used two pre-trained convolution neural networks, AlexNet and ResNet-50, for extracting features, followed by an SVM as a classifier. Second, we applied transfer learning from the pre-trained AlexNet CNN for the classification task. Tests were conducted with different datasets. We then looked at the different results and analyzed the effectiveness of each approach and compared the results when using support vector machines (SVM) and transfer learning from pre- trained AlexNet. In our study, we used SVM as a classifier to recognize faces because of its observable classification result on nonlinear data. SVM has many advantages in solving pattern recognition problems and machine learning problems such as FR and function overfitting.

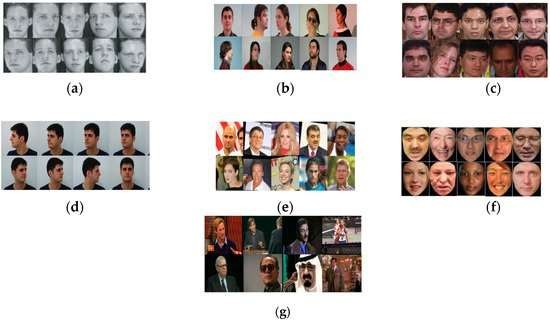
The SVM [[15](https://www.mdpi.com/2076-3417/9/20/4397/htm#B15-applsci-09-04397)] classification is referred to as a process whereby the supervised binary classification method is used and when a training set is introduced, wherein the algorithm develops a hyperplane that maximizes the margin that exists between two input classes. For instance, considering linearly separate data with two distinct classes, the system can have numerous hyperplanes which separate two classes. SMVs identify the most ideal hyperplane that has a maximum margin between all available hyperplanes, whereby the margin is the distance difference between the hyperplane and the support vectors. In SVM, assuming that we represent the input/output sets as X and Y, the goal is to learn the function y = f (x, α), where α is the parameters of the function, and f can be defined as f (x, {w, b}) = sign (w × x + b). Thus, the goal is to find the best set of parameters **w** and b so that the margin is maximized. However, in the real world, the data are not always linear, and it is not possible to classify by a linear classifier, and thus the non-linear SVM classifier is proposed. The non-linear SVM comes with the kernel trick. The kernel trick is a very interesting and powerful tool. The selection of a suitable kernel for a given application or for a set of features is still an open problem. In this paper, the selected kernel function is a linear kernel function without any optimization, which means that linear kernel function does not have any parameters to optimize. In this study, we will not focus on investigating the strategies for SVM optimization.

#### Setting

All experiments were conducted using the platform of Windows with the configuration of Intel Core i7- CPU @ 2.7 GHz with 16 GB on NVIDIA GEFORCE GTX 1050TI. MATLAB 2018a tool was used to evaluate the method and perform the feature selection and classification task. As previously mentioned, before beginning the training process for the convolutional neural network architectures, a previous pre-processing is required. For all datasets, a rescale is applied to resize the images to a 227 × 227 as input for AlexNet and 224 × 224 as input for ResNet-50. The performance of the pre- trained convolutional neural network system is evaluated on the basis of the quality metric known as recognition accuracy. The accuracy is the fraction of the predicted labels that are correct.

Dataset Description

This section describes all datasets used in this study. [Table 4](https://www.mdpi.com/2076-3417/9/20/4397/htm#table_body_display_applsci-09-04397-t004) summarizes the data in each database used in the study. Some samples from all datasets are shown in the following Figure.



**Figure 4.9** Sample images from all datasets: (**a**) sample images from ORL dataset; (**b**) sample images from GTAV face dataset; (**c**) sample images from Georgia Tech face dataset; (**d**) sample images from FEI dataset; (**e**) sample images from LFW dataset; (**f**) sample images from F\_LFW dataset; (**g**) sample images from YTF dataset.

# **CHAPTER 5**

## CONCLUSION

### 5.1 SUMMARY

Based on this study a complete survey of the state of the art technique for age, gender and emotion classification has been reviewed and discussed via face images. Face images have become important in recent decades primarily due to their promising real- world application in several emerging fields. In this paper, various algorithms and various data set have been proposed by their researcher. Also, a summary of the published paper in this field of study was done, including the method used, their performance, and limitation. The result of this study indicates that for Real- time age, gender, and emotion HOG+viola-jones algorithm (70%) has good accuracy rates in FER13 datasets. For only age, gender EDA (72.53% and 98.90%) has good accuracy rates. Only for emotion using the CNN model (97%) has the highest accuracy rates. An overall study of the contribution made on gender, emotion classification, and age estimation used to solve the real-time application problems and its application areas are forensic, security, face detection synthesis, lie detection music for mood, automated tutoring systems. Most of the research work is done is in Convolutional Neural Network and algorithm like - AdaBoost, HOG, HAAR, LBP, PCA, FPLBP, and LDA which are used for age, gender and emotion recognition.

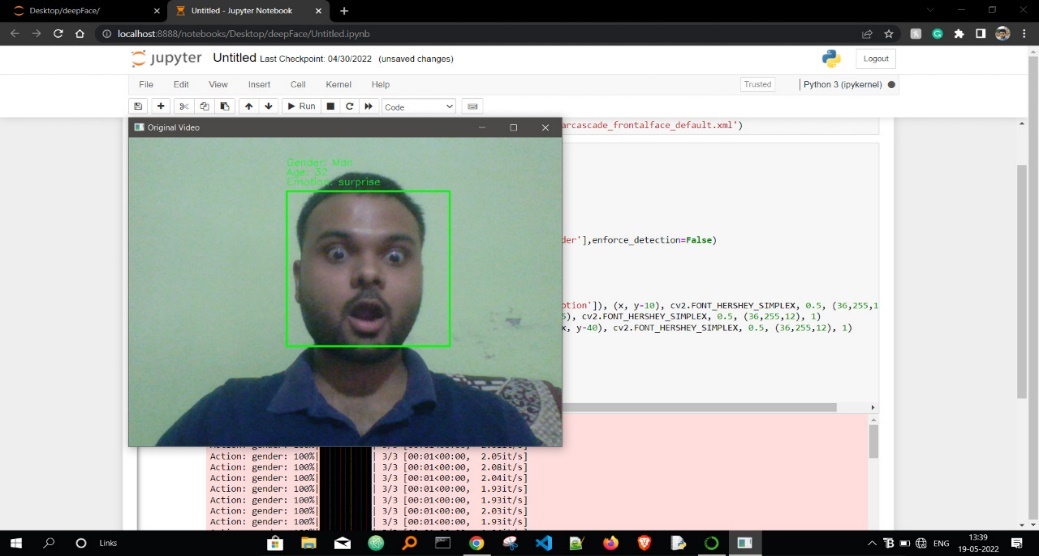
## POTENTIAL ACTIONS FOR IMPROVEMENTS

* Better dataset for emotion prediction
* Due to computational resource constraint, only 5k images from FER2013 dataset were used for age/gender model training.
* Model performance could be improved by using more images
* Image augmentation
* Transfer learning / tuning model architecture

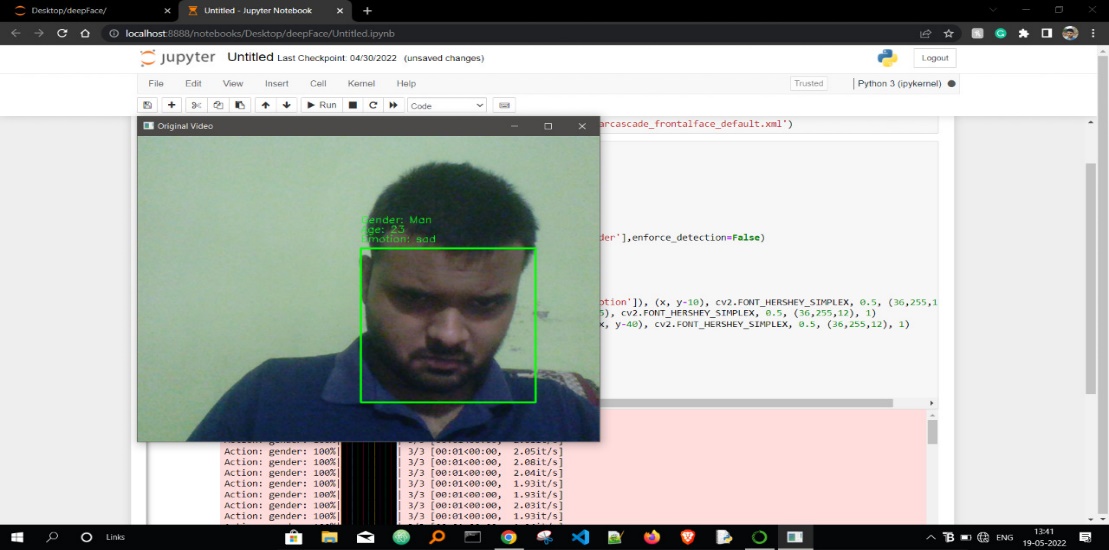
Basically, openCV captures video from your webcam. For every frame, it will convert it to RGB format. This RGB frame will be sent to detect\_face function, which firstly detects all the faces in the frame using MTCNN and for every face, predicts using the 3 trained models to generate outcomes. These outcomes are returned together with the face bounding box locations (top, right, bottom, left).

OpenCV then makes use of the bounding box locations to draw rectangle on the frame and display prediction outcomes in text.

Implementation of the detect\_face function can be found in the source code. Note that since emotion model is trained from grey-scale images, RGB image needs to be grey- scaled before being predicted by emotion model.



**Figure 5.1 *Emotion: Surprise, age: 32, sex: man***



**Figure 5.2 *Emotion: sad, age: 23, sex: man***

Thus we can predict gender and age using face data.

## 5.3 CONCLUSION

We have seen how to predict emotion, gender, and sex from a facial image.

## 5.4 ADVANTAGES

* We proposed a fully automatic approach that produces accurate predictions about gender, emotion and age.
* We embrace the underlying uncertainty of the problem by posing it as a classification taskand use class-rebalancing at training time to increase the accuracy of the result.
* The system is implemented as a feed-forward pass in a CNN at testing time.
* This approach is used at airport security surveillance for the threat perception.
* This problem is clearly under constrained, so previous approaches have either relied.

## 5.5 DISADVANTAGES

* Like many systems built on Convolutional Neural Networks, emotion, gender and age produces some remarkable results, but it struggles with edge cases.
* The accuracy may be compromised if we use some images which are less in quantity. For example, the images of people who are above 100 years are less in FER2013. Hence our model may struggle while predicting ages of such people.
* The accuracy of the model is not up to the mark.
* High computational resources are required.

# **REFERENCES**

1. Md. Jashim Uddin, Dr. Paresh Chandra Barman, Khandaker Takdir Ahmed S.M. Abdur Rahim , Abu Rumman Refat , Md Abdullah-Al-Imran6 "A Convolutional Neural Network for Real-time Face Detection and Emotion & Gender Classification" IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
2. Thakshila R. Kalansuriya and Anuja T. Dharmaratne,"Neural Network based Age and Gender Classification for Facial Images" International Journal on Advances in ICT for Emerging Regions
3. M. R. Dileepa and Ajit Dantib "Human Age and Gender Prediction Based on Neural Networks and Three Sigma Control Limits" ISSN: 0883-9514 (Print) 1087- 6545 (Online) Journal homepage: <http://www.tandfonline.com/loi/uaai20DIBYA> JYOTI SHARMA ET AL.: REAL-TIME EMOTION, GENDER, AGE DETECTION USING CNN 117
4. 2018-Sepidehsadat Hosseini, Seok Hee Lee, Hyuk Jin Kwon, Hyung Il Koo Nam Ik Cho, “Age and Gender Classification Using Wide Convolutional Neural Network and Gabor Filter”, IEEE2018.
5. Imane Lasri, Anouar Riad Solh Mourad E Belkacemi, “Facial Emotion Recognition of Students using Convolutional Neural Network”, IEEE-2019.
6. Rajesh Kumar G A, Ravi Kant Kumar Goutam Sanyal, “Facial Emotion Analysis using Deep Convolutional Neural Network”,2017 International Conference on Signal Processing and Communication (ICSPC). doi: 10.1109/ cspc.2017.8305872, Pg.No.- 369 to374.
7. Md Abdullah-Al-Imran “A Convolutional Neural Network for Realtime Face Detection and Emotion & Gender Classification’’ e-ISSN: 2278-2834, p- ISSN: 2278- 8735. Volume15, Issue 3, Ser. I (May - June2020), PP 37-46.
8. S L Happy and Aurobinda Routray“Automatic Facial Expression Recognition Using Features of Salient Facial Patches’’ DOI: 10.1109/TAFFC. 2014. 2386334 <https://rb.gy/9m5dt2>.
9. Ramin Azarmehr, Robert Laganiere, Won-Sook Lee Real-time Embedded Age and Gender Classification in Unconstrained Video tps://rb.gy/pnvd2n.
10. Jang-Hee Yoo, So-Hee Park, and Yongjin Lee “Real-Time Ageand Gender Estimation from Face Image” ISBN: 978-0-6480147-3-7.
11. Octavio Arriaga1 and Matias Valdenegro-Toro and Paul G. Pl¨oger Real-time Convolutional Neural Networks for emotion and gender classification.”
12. Eran Eidinger, Roee Enbar, Tal Hassner “Age and Gender Estimation of Unfiltered Faces’’
13. Ajit P. Gosavi, S. R. Khot “Facial Expression Recognition Using Principal Component Analysis” ISSN: 2231-2307, Volume-3, Issue-4, September 2013
14. Sidharth Nair, Dipesh Nair, “Detection of Gender, Age and Emotion of a Human Image using Facial Features” e-ISSN: 2395-0056 [www.irjet.net](http://www.irjet.net/) p-ISSN: 2395-0072
15. Rekha N, Dr.M.Z.Kurian "Face Detection in Real Time Based on HOG" international journal of advanced Research in computer engineering & tchnology volume 3 issue 4,april2014 ISSN:2278-1323
16. Tanner Gilligan, Baris Akis Emotion AI, Real-Time Emotion Detection using CNN " <http://web.stanford.edu/class/cs231a/>prev\_projects\_2016/emotion-ai-real.pdf