

One DNN for Joint Face Detection and Affect Recognition

Name: Arun Narayanaswamy

ID: 001122220

Project Supervisor: Dr. Dimitrios Kollias

School of Computing and Mathematical Sciences



Table of contents

Chapter 1 -Introduction

- 1.1 Problem Statement
- 1.2 Aim
- 1.3 Objectives
- 1.4 Approach
- 1.5 Resource Overview
- 1.6 Significance of the study
- 1.7 Work Plan management with the Gantt Chart

Chapter 2 – Literature Survey

- 2.1 Related Works
- 2.2 Research Interpretations from other studies
- 2.3 Comparison Charts of other research works

Chapter 3: Methodology and Procedure

- 3.1 Algorithm Model and Insights
 - 3.1.1 Face Detection
 - 3.1.2 Deep Learning – DNN
 - 3.1.3 Convolutional Neural Networks
- 3.2 Software Requirements
- 3.3 Pre-essential Understanding

Chapter 4 – Experimentation and Executional Setup

- 4.1 Data Set Description
- 4.2 Data Packages
- 4.3 Data Preparation
- 4.4 Displaying the trained Dataset
- 4.5 Building Face Detection Model
- 4.6 Accuracy Check
- 4.7 Testing faces in utilized Database

Chapter- 5 Conclusion

- 5.1 Challenges we face in Face detection-
- 5.2 Our Findings-
- 5.3 Future Scope of our study
- 5.4 Advantages and Disadvantages of our study

Chapter 6 – References

List of Figures

Figure 1: Three VFM areas

Figure 2: Positions of the eye for comparison.

Figure 3: Positions of the mouth for comparison.

Figure 4: 4 Stages for Face Detection

Figure 5: CNN framework [Source: Towards Data Science]

List of Tables

Table 1 Gantt Chart

Table 2 Features in the eye figure.

Table 3 Features in the mouth figure.

Table 4 Feature values of eye template.

Table 5 Feature values of mouth template.

Table 6 Detection accuracy

Table 7: Number Code and Corresponding emotion

List of Outputs

Output 1: Emotion disgust

Output 2: Emotion Surprise

Output 3: Emotion Sad

Output 4: Emotion Happy

Output 5: Emotion Fear

Output 6: Emotion Angry

Output 7: Module Summary

Output 8: Display Epochs 1 to 17

Output 9: Display Epochs 18 to 34

Output 10: Display Epochs 35 to 50

Output 11: Epochs Vs Loss in Training Vs Validation

Output 12: Epochs Vs Accuracy in Training Vs Validation

Output 13: Test Loss and Accuracy Results

Output 14: Testing for Surprise

Output 15: Testing for Angry

Output 16: Testing for Happy

Output 17: Testing for Sad

Output 18: Testing for Fear

Output 19: Testing for Disgust

Abstract

Topic: One DNN for Joint Face Detection and Affect Recognition

Facial expression recognition software is a system that detects emotions in human faces by using biometric indicators. This technology, more precisely, is a sentiment analysis tool that can recognize the six essential or universal expressions: happy, sorrow, anger, surprise, fear, and disgust, automatically. Because a facial expression recognition system is a computer-based technology, it employs algorithms to detect faces, code facial expressions, and recognize emotional states in real-time. It accomplishes this by evaluating faces in photos or video via computer-powered cameras embedded in laptops, smartphones, and digital signage systems, as well as cameras mounted on computer screens. The goal of this research titled "ONE DNN FOR JOINT FACE DETECTION & AFFECT RECOGNITION" is to construct a single DNN that can do both the face detection and alignment stage as the face analysis step, which affects recognition. Deep neural networks (DNNs) have been widely used to solve many real-world applications in the past few years. The primary issue now is to analyse all of the facial ratios. A face-SSD program may contain several weaknesses to study, as we will handle the entire software online. This project aims to create one DNN that performs both the face detection and alignment step and the face analysis step, which affects recognition in this project.

Keywords: One DNN, Training, Testing, Annotations, Emotions, Sentiments, Joint Face Detection, Affect Recognition, Machine Learning, Deep Neural Networks, Python, Collab Notebooks, etc.

Chapter 1 -Introduction

The importance of emotions in rational decision-making, perception, and human interaction has sparked interest in programming computers to perceive and replicate emotions. DNN has become quite popular due to its accuracy when taught with essential data. ML is a set of algorithms that learns them, parses data, and applies what they've learned. This project aims to do research on joint face detection and affect recognition using a single DNN. Four standard pipelines are employed for joint face detection and affect credit in the face analysis area.

I will discuss the primary objectives of this study proposal and methods for achieving them. Then, to conduct this research, we will provide a project plan. We will examine various concerns involving legal, social, and ethical considerations. People, resources, and success factors are all critical.

Many applications that could profit from this skill are discussed in [R.W. Picard, "Affective Computing", MIT Press, Cambridge 1997]. [F. Cassell et al. 1999] and [F. Cassell et al. 1999] emphasize the relevance of including nonverbal communication into autonomous dialogue systems. However, because the study on recognizing human facial expressions has been focused on static photos or image sequences in which the individuals display a specific emotion but no voice, it cannot be immediately adapted to a dialogue system. According to psychological studies, happiness, sadness, surprise, fear, and anger are all universally connected with various facial expressions.

Some studies have been done on photographs that catch the subject expression at its most intense. The existence of static indicators such as wrinkles and the position and form of facial features can be detected using these images [A Young & H Ellis et al. 1989]. Classifiers for facial expressions, for example, were based on neural networks in [C. Padgett, G. Cottrell et al] and [G.W. Cottrell and J. Metcalfe et al 1991]. The network was shown static faces as projections of feature region blocks onto the principal component space derived from the image data set. They recognize the six basic emotions with an accuracy of roughly 86 percent.

The extraction of facial expressions from video sequences has also been studied. Most studies in this field create a video database from people who make demand expressions. They attempted to categorize the six primary faces indicated above, but They did not evaluate them in a discussion setting—the Ekman and Friesen Facial Action Coding System [I. Essa and A. Pentland et al. 1995] is used in most efforts in this field. The FACS counts all of a face's Action Units that generate facial motions. When these active components are combined, many different facial expressions are available.

[Y. Yacoob, L.S. Davis et al.,] computes the optical flow at the spots with a large gradient at each frame to identify the orientations of rigid and non-rigid motions induced by human face emotions. The authors then propose a vocabulary to characterize the FACS'FACS' face activities using feature motion and a rule-based system to distinguish facial expressions from these actions.

[M. Black, Y. Yacoob., et al.] takes a similar technique, using a local parameterized model of picture motion in specific facial parts to recover and recognize the non-rigid and articulated movement of the faces. The characteristics of this identified motion are linked to facial feature motion during facial expressions.

1.1 Problem Statement

The dataset was provided, and it comprised all of the difficulties related to face ratio and expression analysis. As shown, FCNN is capable of assessing all of the software changes needed to align all of the faces. The primary issue now is to analyse all of the facial ratios. A face-SSD program may contain several weaknesses to study, as we will handle the entire software online.

Deep neural networks (DNNs) have been widely used to solve many real-world applications in the past few years. In the face analysis domain, the standard pipeline has been:

- Train a face detection software,
- Using this face detection software, detect all faces in the utilized database,
- Optional face registration step & align all detected faces
- Train a DNN using the aligned-detected faces for the studied face analysis problem.

The above four steps were done consecutively and in most of the cases involved DNN model: the face detection one and the one trained with the aligned-detected faces that solved the face analysis problem. This project aims to create one DNN that performs both the face detection and alignment step and the face analysis step, which affects recognition in this project.

1.2 Aim

The goal of this research, titled "ONE DNN FOR JOINT FACE DETECTION & AFFECT RECOGNITION", is to construct a single DNN that can do both the face detection and alignment stage the face analysis step, which affects recognition.

1.3 Objectives

In this early stage of this research, it is imperative to set some objectives. This will help us to have a direction where the study is going. Here is a list of some goals that will be the main focus of this research.

- To develop and train face detection software for the development of DNN. To detect all the faces in a utilized database that will help to improve DNN by using that face detection software.
- To implement a face detection process that will align all the faces that would be detected.
- To perform training of the face detection software for solving face analysis problems by using aligned detected faces.

1.4 Approach

I implement face detection software at the outset to analyze all of the faces we will save in the database. According to Bargshady et al. (2019), human facial expression is the most crucial factor in software development. There are some tools available to aid in the creation of DNN. Face-SSD is software that uses a "Fully Convolutional Neural Network (FCNN)" to detect all the sizes of many faces, according to Jang et al. (2019). Face-SSD can thus be stated to catch all of the different looks in the database. Several fronts will be evaluated and aligned with this software to train the face detection of Images prepared.

FCNN will be a suitable method for implementing the Face-SSD since it will help identify changes, including any facial expression restructuring. According to Yang (2019), all tools have been curated, and preparation for the design portion of the application has begun. It can be shown that there are many single-stage face detectors available, with Face Retina being chosen for this study. According to Deng et al. (2019), five facial landmarks on the larger face dataset will be manually annotated to monitor the use of all datasets. For a VGA-resolution image, Retina Face will execute in real-time on a single CPU core. To achieve this, we will follow backbone networks of a lighter weight.

Two branches will assist Face-SSD to align all of the recognized collected databases. As can be seen, both filters will execute different jobs to improve face detection performance. According to Jang et al. (2019), one branch discusses how to recognize diverse faces, while the other branch deals with analyzing such faces. When Face-SSD is implemented, several ratios will be examined, including smile detection, attribute prediction, and Root Mean Square (RMS) error for valence and arousal estimate. Jyoti (2018) believes that design creation is the most critical aspect.

On HCI, most of the brain, knowledge, and computing language understanding is required. There are various setups for each face shape, like wider faces and many more. According to Jang et al. (2019), the face identification and analysis will be two separate branches accomplished by adopting FCNN. All of the face ratios that would have been obtained will be compared in this software. According to Deng et al. (2019), Retina Face employs pixel-wise face localization. Multi-task learning, both supervised and self-supervised, can be used to implement it. This stage involves creating software for a specific application utilizing the chosen software to detect faces and recognize the effects of people's faces. According to Yang (2017), We will test the software-based application on individuals after it is built. The software is created using a software algorithm. Face-SSD has the power to catch real-time face detection, and it has generalized architecture.

1.5 Resource Overview

I will need to use numerous tools like Tensor Flow and PyTorch to finish this research. Python is the primary programming language used in this study to build the model. This research will benefit significantly from previously published journals, publications, and web pages. The research and development of the application will necessitate several tools. While conducting this research, all people and resources are equally important. As a result, it is critical to pay attention to these details.

- Basic understanding of deep learning and neural networks;
- Familiarity with existing packages
- TensorFlow and PyTorch are a plus, but not required;
- Excellent programming skills, primarily in Python.

1.6 Significance of the study

In this research proposal, several aspects of face detection and affect recognition using one DNN are adequately discussed to understand its facilities and drawbacks. We set some adequately explained objectives, and how we can achieve them is also included.

[M. Rosenblum, Y. Yacoob, et al.] propose a radial basis function network architecture for learning the link between facial feature motion patterns and human emotions. The optical flow at the points with a strong gradient is also used to compute the motion of the features. All of these techniques have an accuracy of roughly 85%. Other approaches [I. Essa and A. Pentland et al.] use physically-based models of heads, such as skin and muscles.

Because facial expression recognition model provides raw emotional responses, it can provide important information about a target audience's mood toward a marketing message, product, or brand for organizations.

- Market research is typically conducted using surveys to determine what consumers want and need. On the other hand, this strategy assumes that the expressed preferences are accurate and reflect future activities.
- This isn't always the case, though. Behavioural approaches, in which users' reactions are observed while interacting with a brand or a product, are another prominent strategy in market research. Although successful, as the sample size grows, such procedures can soon become labour-intensive.
- Facial expression recognition technology can save the day in such situations by allowing organizations to perform market research and automatically measure moment-by-moment facial expressions of emotions, making it simple to aggregate the results.

It's the best technique to figure out how effective any business material is. To quantify muscle actuations, they combine this physical model with registered optical flow data from human faces. They offer two techniques, which generate typical muscle activation patterns for various facial emotions.

- We can also use facial expression recognition in the testing phase of video games. During this stage, a focus group of users is customarily instructed to play a game for a set amount of time while their behaviour and emotions are observed.
- Game makers can obtain insights and form inferences about the emotions experienced during gameplay utilizing facial expression recognition and include that feedback into the final product.

- Facial expression analysis is a valuable tool for going beyond traditional survey methods. It's a way of appreciating what the user is going through while also receiving feedback.
- When given this way, feedback becomes truly non-intrusive in the user experience.

The resemblance of a novel picture sequence to conventional muscle activation patterns is used to classify it. The second uses the same process to generate the typical moving energy associated with each face expression from muscle actuation.

In [J. Lien, T. Kanade, J. Cohn, C. Li et al. 1998], a method is provided that extracts the FACS Action Units from which expressions can be inferred using facial feature point tracking, dense flow tracking, and high gradient component analysis in the spatiotemporal domain. [M. Pantic, L. Rothkrantz, et al. 2000] provide a comprehensive overview of various strategies.

1.7 Work Plan management with the Gantt Chart

Here, I list the rules I shall abide by as I finish my project-

1. Focus on the following components to begin building a project plan-Determine the scope of our project.
2. To get started, I'll need to break down the project and figure out what it includes.
3. Define my aims and goals. After I've whittled down our project, I'll need to decide on the aims and objectives I want to achieve.
4. My objectives are the 'what.' The 'how' is determined by my goals.
5. The time allotted for the project's approach is depicted in the Gantt chart below. The periods are set up in such a way that the majority of it pays off.
6. Tasks should be defined. To finish the project, I'll need to precisely outline the tasks required.
7. Keep an eye out for any potential project killers. I must identify and resolve these project killers for the project to function successfully. Make a timetable.
8. Break down work into phases and set deadlines for when I want to achieve my goals and objectives. Knowing how long each stage takes can help me stay on track with the project.
9. Obtain feedback. Before moving forward with my strategy, I gathered input from my staff, co-students, and Professors.
10. Make the necessary changes to my strategy. Even the most well-thought-out project plans cannot anticipate all potential issues

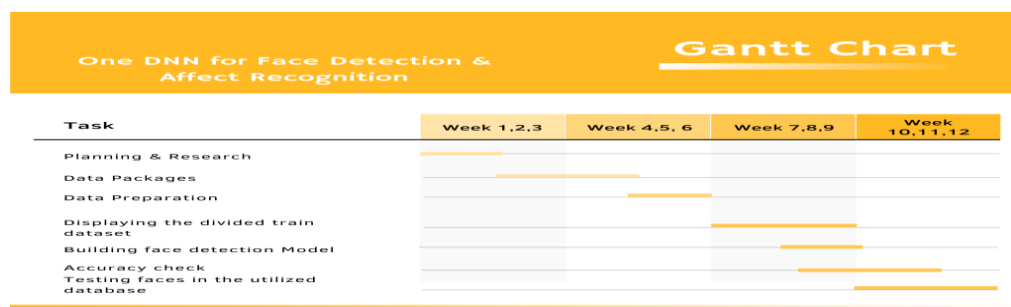


Table 1: Gantt Chart

In the 12-week schedule as mentioned above, we are supposed to owe the need to possess and follow the below traits to finish the project as planned. As the chart displays the first 3 weeks are dedicated to Planning and research. The next 3 weeks are indulged in exploring and importing data packages and data preparation. The following 3 weeks are dedicated to Building the face detection model, Accuracy check, and Testing. In the next 3 weeks schedule, the build of the model will coincide along with metrics evaluation and compilation of the model. This shall later be followed by testing of all the faces in the utilized database. We shall learn more about this in the experimentation chapter. I have kept the following in mind to keep the project on track.

- Check-in on the project at frequent times. I should arrange a time to assess the project's progress every other day or once a week, whether it's every other day or once a week.
- Be adaptable. When working on a project. I am the professionals who have the expertise and experience to make our vision and plan a reality, even if I have a vision and a plan in place. Keep an open mind and listen to what people have to say if they have any recommendations.
- Keep everyone informed and offer status updates regularly. Better, hold weekly meetings so that everyone is up to speed and can discuss any concerns that arise, in my case I shall update my professor.
- Address any issues before they become a problem. Prepare for any potential hurdles ahead of time so I can deal with them before they become an issue. Give instructions. It is our job as the project manager to keep our professors and guides informed and ensure that everything goes well. To accomplish so, I must equip myself with the necessary resources and knowledge to ensure that the project is completed on time.
- While there is a multitude of strategies, I may employ to manage your time, the following are some of the most effective: Delegate responsibilities, Make a list of your priorities. Keep your meetings to a minimum.

Chapter 2 – Literature Survey

2.1 Related Works

One of the articles [Widanagamaachchi, Wathsala, 2009 et.al] that found very fascinating used an effective methodology to recognise the moods of people by their facial expressions, and we thought that was particularly interesting. The researchers claim that they have been able to classify and categorise the six universal emotions of joy, sorrow, anger, contempt, surprise, and fear using neural networks and visual processing techniques. Furthermore, they assert that this approach allows them to discern between various feelings. In the first stage, the input photo is subjected to a facial recognition stage, which is responsible for detecting the characteristics of the subject's face. A technique for feature point extraction that is based on image processing techniques is then used to the image processing data in order to extract the data points from the image processing data. In the last stage, a neural network for emotional expression recognition is established by feeding a collection of values obtained by the processing and retrieval of feature points into a neural network for emotional expression identification in order to recognise emotional expressions.

Many studies have been undertaken on the subject of recognising a person's facial emotions, and it has been widely documented. The use of VFM is one of the most powerful ways that we have identified in a study paper [Kim, & Joo, Young Hoon & Park, Jin. 2015 et.al] and it is one of the most powerful approaches that we have discovered. The face region and facial components are obtained during the image processing stage, and the research team goes through three stages during this stage. When attempting to extract the face region from a facial image, procedures such as the fuzzy colour filter and histogram analysis are used. After that, the face components are extracted from the video using the VFM and histogram analysis methods, respectively. Novel feature extraction algorithms are introduced in the feature extraction step, with the goal of extracting features for the purpose of emotion detection.

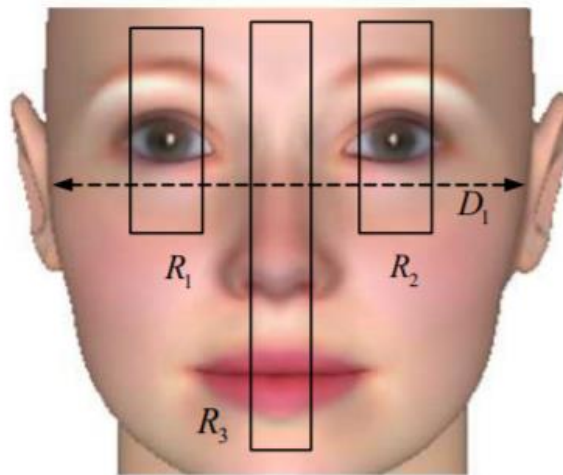


Figure 1: Three VFM areas

The emotion of a facial image can be assessed by analysing the collected features and using a fuzzy classifier to classify the image. In their work, the researchers show experimental results that demonstrate that the proposed algorithm is successful at distinguishing emotional states. Numerous studies on real-time emotion identification have been conducted, but this research [R. S. Deshmukh, V. Jagtap, S. Paygude et.al] indicates that facial expressions are taken into account when distinguishing emotions in human facial photographs acquired with a webcam.

Their proposed system is independent of gender, age, ethnic group, beard, background, and birthmarks. Their proposed strategy is intended to be incredibly effective and designed for individuals who are stressed out throughout their work hours through the use of music therapy. This can be a challenging undertaking due to the uncertainty involved in inferring hidden mental states from behavioural signs and the difficulties inherent in automating the study of facial expressions in images.

2.2 Research interpretations from other studies

This paper[André, E., Rehm, M., Minker, W., Bühler, D. 2004 et.al] describes studies on automatic emotion detection systems, which are being conducted to develop efficient, real-time methods of identifying the emotions of mobile phone users, call centre operators and consumers, car drivers, pilots, and a variety of other users of human-machine communication systems, including but not limited to In order to seem and behave in a human-like manner when communicating with people, it has been recognised since the beginning of time that robots must be endowed with the ability to communicate their emotions through programming.

Despite the fact that the application of convolutional neural networks has been extensively investigated, our findings indicate that deep CNNs are capable of providing good results on a difficult dataset when trained with supervised learning. One piece of research [Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. 2017 et.al] showed that removing a single convolutional layer from a network causes the network's performance to degrade. In their research, they discovered that removing any of the middle layers results in a loss of around 2 percent in the top-1 performance of the network when the middle layer is removed from the network. So, when it comes to getting good results from CNN, depth is essential.

They chose not to conduct any unsupervised pre-training for the sake of simplicity, despite the fact that they anticipated that it would be beneficial, particularly if they had been able to obtain sufficient computational power to significantly increase the size of the network without obtaining an equal increase in the amount of labelled data, as they did in this study. As the researchers' network has grown and been taught over longer periods of time, their results have improved, but they still have a long way to go before their network can match the infero temporal pathway of the human visual system, which is many orders of magnitude away. The researchers have come to the conclusion that very large and deep convolutional nets should be used on video sequences where the temporal structure provides extremely useful information that is either absent or significantly less visible in static images. They have done extensive research to reach this conclusion. To which we totally concur, while also acknowledging the difficulties that can occur when using statistical imagery in our work.

There have been various arguments in recent years in favour of employing SER (speech emotion recognition) to detect emotions in speech rather than images or video rather than images or video. One of the research study [Mao, Qirong; Dong, Ming; Huang, Zhengwei; Zhan, Yongzhao 2014 et.al] is concerned with the recognition of prototypic expressions of various basic emotions from exhibited emotional utterances in laboratory settings, with the goal of improving recognition accuracy.

In comparison to overt or continuous interpretations of the same utterances in diverse situations, emotional utterances recorded in naturalistic and real-world settings are unquestionably more important and more challenging study subjects to answer. It is particularly well-suited for overcoming these difficulties is feature learning, an advanced technique for learning a transformation of raw inputs into a representation that can be efficiently exploited by a classifier.

2.3 Comparison Tables and more about the methodology of various papers

Among the findings of the research paper [Kim, & Joo, Young Hoon & Park, Jin. 2015 et.al], the main emphasis was on VFM, and there are specific formulae for calculating the emotion on a person's face based on a few characteristics, as illustrated by the image and the tabular form below.

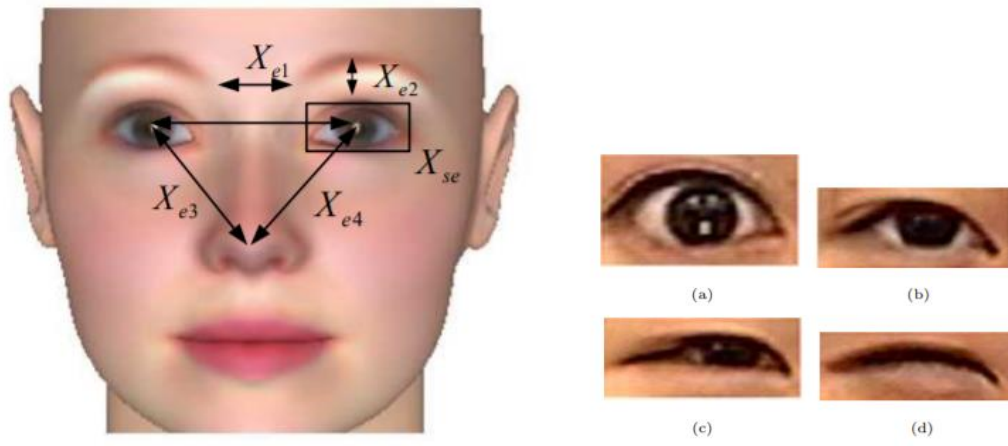


Figure 2: Positions of the eye for comparison.

Features	Description	Size
X_{e1}	Distance between two eye brow	1×1
X_{e2}	Distance between eye and eye brow	1×1
X_{e3}	Distance between nose and eye(left side)	1×1
X_{e4}	Distance between nose and eye(right side)	1×1
X_{se}	Error between eye and template	4×1

Table 2: Features in the eye figure.

On the following page, you will find a list of the distinctive properties of the ocular area. There are eight features that can be extracted from the ocular area. The arrangement of characteristics in the eye region is represented in the diagrams and illustrations below. These four properties of the eye and eyebrow represent geometric information in the form of geometric information. The remaining four features represent the information about the eye's shape that it contains. It is possible to gather this information about the shape by comparing it to a template.

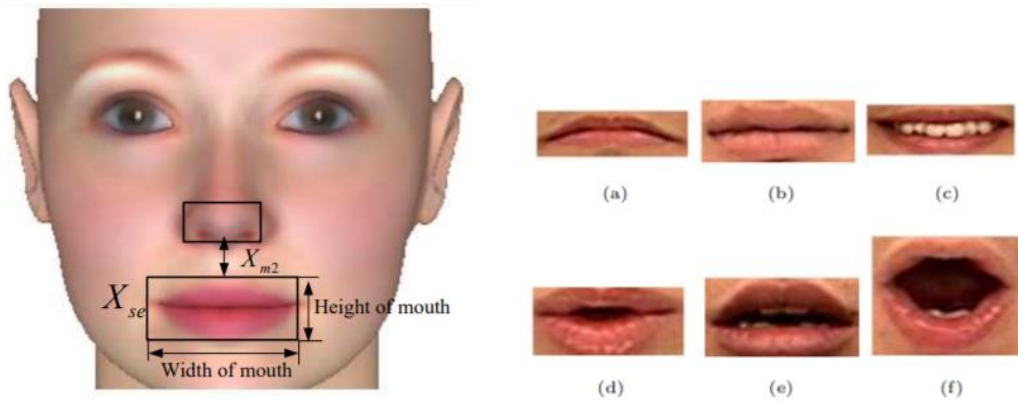


Figure 3: Positions of the mouth for comparison.

Features	Description	Size
X_{m1}	$\frac{\text{Width of mouth}}{\text{Height of mouth}}$	1×1
X_{m2}	Distance between nose and mouth	1×1
X_{se}	Error between mouth and template	6×1

Table 3 Features in the mouth figure.

According to the following table, the mouth region and its surrounding territories have a number of traits that can be identified. In the figures, a range of features in the area around the mouth are illustrated in relation to one another and to the rest of the body. It takes only two characteristics to accurately express geometric information in geometry, however it takes six features to accurately represent shapes in geometry.

Using a comparison template to its full capacity is depicted in the graphic above the table. The below values are the feature values according to the research paper [Kim, & Joo, Young Hoon & Park, Jin. 2015 et.al]

Template	X_w	X_h	$\frac{X_w}{X_h}$	X_p
Fig. 4(a)	0.22	0.31	0.66	13.1
Fig. 4(b)	0.23	0.24	0.46	9.6
Fig. 4(c)	0.23	0.25	0.49	10.5
Fig. 4(d)	0.23	0.25	0.48	11.4

Table 4 Feature values of eye template.

Template	X_w	X_h	$\frac{X_w}{X_h}$	X_p
Fig. 6(a)	1.79	0.37	0.40	20.12
Fig. 6(b)	1.83	0.34	0.35	20.10
Fig. 6(c)	1.55	0.49	0.61	18.89
Fig. 6(d)	1.81	0.54	0.63	20.51
Fig. 6(e)	1.76	0.34	0.39	15.98
Fig. 6(f)	2.14	0.41	0.37	28.0

Table 5 Feature values of mouth template.

According to the researchers [Kim, & Joo, Young Hoon & Park, Jin. 2015 et.al] , image comparison is challenging and requires a large amount of computational time. For this challenge, a new computed algorithm is used to compare the facial component image with the template in order to find a solution. On the other hand, just a few unclear patterns are presented in the emotion recognition problem on the other hand. According to the researchers, they are unable to make a firm guarantee that the recovered feature vector represents only a single specific emotion because they do not have access to the data. As a result, the creation of an emotion identification system is not an easy undertaking. During the emotion detection stage, a fuzzy classifier is employed in order to overcome this difficulty. When it comes to dealing with classification problems that have unclear input, the fuzzy classifier is one of the most powerful classifiers that can be used. The fuzzy classifier can be thought of as a set of fuzzy rules that are applied to data in order to make classification decisions. The structure of the fuzzy rule is taken into consideration in each and every situation.

Facial component extraction sucess	82.7%
Fuzzy classification accuracy	89.5%
Final emotion detection accuracy	74.0%

Table 6 Detection accuracy

The final emotion recognition rate is calculated by multiplying the accuracy of facial image analysis and the accuracy of fuzzy classification combined in a single equation to obtain the final emotion recognition rate. Table 5 summarises the accuracy of the emotion detection system as determined by a fuzzy classifier, as well as the results of an experiment with over a hundred and twenty-four patterns. It has been determined by the researchers [Kim, & Joo, Young Hoon & Park, Jin. 2015 et.al] that there are a total of thirteen facial pictures that cannot be categorised. When it comes to each emotion, the accuracy of emotion recognition is examined in Table 5, where they have placed the most emphasis on recognition accuracy and where the results are similar to each other.

Chapter 3: Methodology and Procedure

In this chapter, we shall see the methodology, software, hardware resources, and domain on which our project One DNN for Face detection and Affect Recognition shall be built.

3.1 Algorithm Model and Insights

In this we have mainly utilized the domains of deep neural networks of CNN and NLP, why and how are listed in the latter chapters.

3.1.1 Face Detection

The human brain takes less than a minute to locate and distinguish an item inside a picture; however, machines require time and a significant quantity of data to do the same operation. In Face identification and classification, a deep neural network based on a convolutional neural network provides high accuracy and excellent results. Deep neural networks demand a significant quantity of data images and videos as well as time to train. Due to the high computational cost of computer vision, the transfer-learning approach, in which a model educated on one job is reused on a related task, produces superior results. Various deep learning-based object identification methods have been proposed by the authors.

Face identification from pictures and recordings has consistently been an intriguing issue in computer vision and man-made reasoning applications like mechanical technology, self-driving vehicles, robotized video observation, swarm the executives, home robotization and assembling, movement acknowledgment frameworks, clinical imaging, and biometrics.

3.1.2 Deep Learning – DNN

The best proven on the market for prediction, identification, surveillance, authentication, recognition, diagnosis, and analysis in the defense ministry and many other industries, such as crime, banking, trade, forensics, and medicine, have been deep learning and neural networks. Computer vision contains many fascinating problems, from simple image classification to an estimate of 3D positions. The detection of objects is one of the most fascinating and hardworking challenges for engineers. A neural network technique is followed by 4 stages to detect an item as shown in this figure-



Image Classification: It entails categorizing a picture into one of several possible groups.

Localization: Localization, like classification, determines the position of a single item inside an image.

Instance Segmentation: Taking object detection a step further, we'd like to not only discover things inside a picture, but also a pixel-by-pixel mask of each of the identified objects. This is referred to as instance or object segmentation.

Object Detection: The final step. We need to identify and categorize numerous items at the same time while we iterate through the issue of localization and classification. The issue of identifying and categorizing a variable number of items on a picture is known as object detection.

Figure 5: 4 Stages for Face Detection

3.1.3 Convolutional Neural Networks

CNN is a difficult model to build in comparison with a basic and even state-of-the-art, profound convolutionary neural networking model. A computer vision task requires an effective location of objects as well as categorization of items.

A development of article recognition includes deciding the specific pixels in the picture that compares to each recognizably perceived article instead of utilizing erroneous bouncing boxes during thing confinement. Item division is utilized to portray this more unpredictable variation of the issue. To accomplish identification, CNN utilizes shadings or channels.

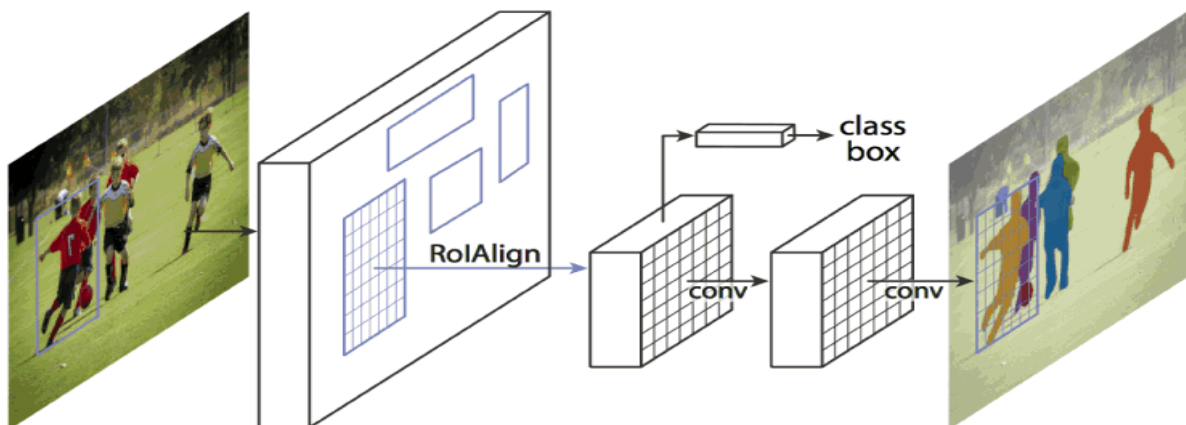


Figure 6: CNN framework [Source: Towards Data Science]

CNN is an intricate model, particularly in examination with an essential or in any event, state-of-the-art neural organization model that is tangled. For every emphasis of the CNN model, source code is available, given in discrete GitHub archives with Caffe Deep learning system-based model models. We might use a dependable outsider execution dependent on the Keras deep learning system, rather than carrying out the model CNN without any preparation.

The CNN project created by Matterport is the best outsider execution. It is an open-source permit and the code has broadly been used in many ventures and Kaggle challenges. Today, in Face Detection, deep learning networks are better than people, showing how potent this method is. But as people, while looking at the environment and engaging with it we do considerably better than merely categorizing images. Also, inside our area of view, we locate and classify each element. These are far more difficult jobs that machines, as well as people, still struggle to do. Indeed, if properly executed, I would argue that Face detection gets robots closer to true scene understanding.

3.4 Software Requirements

The subject of computer vision is fascinating and expanding, with applications ranging from the food sector to unmanned aircraft systems. Through the software and processors built in the devices, the industry has progressed beyond simple visual transmission to pattern recognition and rule-based decision making.

The objective of computer vision is to mimic, reproduce, and surpass human visual capabilities utilizing various levels of computer software and hardware. It necessitates the use of computer science, electrical engineering, mathematics, physiology, biology, cognitive science, and human factors engineering to manage knowledge.

One of the major challenges in computer vision research appears to be noisy picture data, confusing cues between vision and logic, and methods for conflict resolution. Picture processing and pattern recognition are promising areas for future research, particularly in the areas of image reduction, restoration, and improvement.

J Notebook-It is an open-source online application for making and sharing files that fuse account articles, conditions, and live code. Information cleansing and change, quantifiable showing, numerical entertainment, data discernment, ML, and an enormous number of additional possibilities are in general conceivable.

Adobe Reader: To scrutinize, mull over and understand the fundamentals of tantamount examines done by famous researchers and remarkable worldwide journals. It is a gadget that engages any customer to see a report in a PDF plan.

Microsoft Office: I have made use Excel sheets for acquiring and exchanging annotations assortments CSV records.

Navigator: Anaconda Navigator is a workspace based graphical UI Spyder and Jupyter Notebook that helps in dispatching diverse particular programming, supervising and presenting the important packs, etc

Spyder- It is a Python-based open-source stage that offers customers to code and run projects and is significant for researchers who need to proceed in Python programming-related fields.

Programming Language: Python assists originators with being more critical and sure about what they're making, from progress to connection and backing.

- It is inconspicuous
- It is dependable
- It has a wide assortment of libraries and foundations.

3.4 Pre-essential Understanding

One necessities energy to consider something that interests them under referred to region data is principal for the endeavor.

- It's about bits of knowledge and probability in AI. In this manner, having prior capacity in that field would be exceptionally profitable. Be entirely equipped for seeing and overseeing allocations.
- It would be exceptionally valuable to have a strong appreciation of direct factor-based math. Knowing the foundations of Linear Algebra can be satisfactory for the most major subjects in AI, similar to backslide assessment. In any case, when one advances further into ML, a more noticeable understanding of LA will be indispensable.
- Language-Preferably python would be significant if you have prior programming dominance. Python, Java, and R are the most notable programming tongues. Regardless, Python and R have emerged as the precursors lately. Python is a clear language to learn.
- Any fledgling in AI or man-made intellectual competence should have a strong mathematical foundation.

Chapter 4 – Experimentation and Executional Setup

In this section, I set up our experiment and go over each stage in detail, as outlined in the project's approach plan for One DNN for Face Detection & Affect Recognition.

4.1 Data Set Description

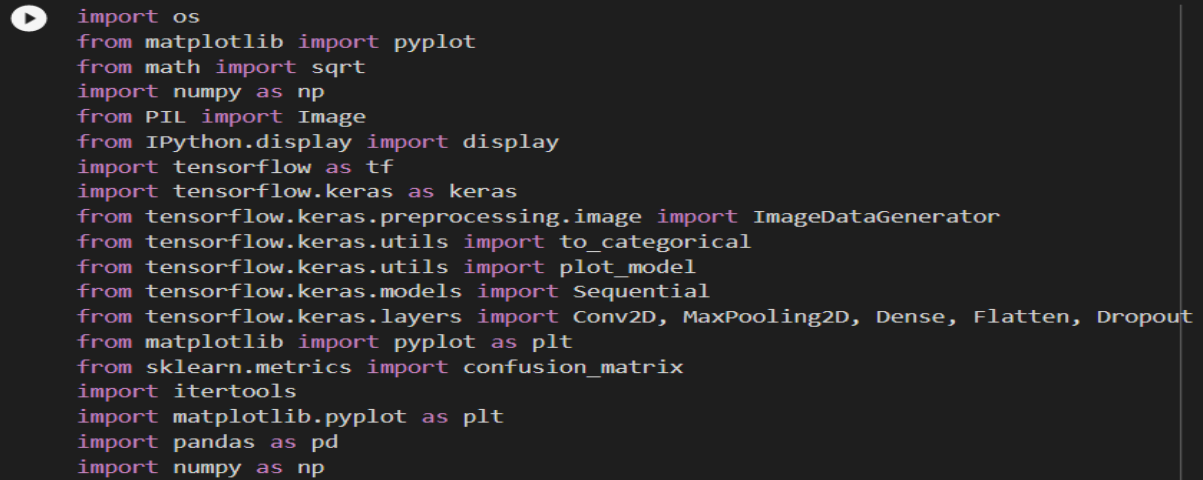
In our project, all our data has been allotted by our professor. We handpicked data obtained from all the google images expanding over the size of 12 thousand images, among which 9 thousand images were allotted for Training, 3 thousand were allotted for testing. All the images we obtained contained annotations that had a set value for their corresponding emotion. All the images we obtained are in .jpg format only. The emotion scale was in the range of [0,1,2,3,4,5] where each number indicates a specific most common emotion for detection as tabulated below-

<i>Sl. No</i>	<i>Number Code</i>	<i>Emotion</i>
1	Angry	5
2	Sad	4
3	Happy	3
4	Disgust	2
5	Fear	1
6	Surprise	0

Table 7: Number Code and Corresponding emotion

4.2 Data Packages

In the image presented below, there are the packages that we have utilized to accomplish the task. As described in the below image we have imported the packages of pyplot, sqrt, NumPy, PIL, display, tensorflow, Keras, Image data generator, Models, layers, confusion matrix, itertools, pandas, etc.



```
import os
from matplotlib import pyplot
from math import sqrt
import numpy as np
from PIL import Image
from IPython.display import display
import tensorflow as tf
import tensorflow.keras as keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.utils import plot_model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, Dropout
from matplotlib import pyplot as plt
from sklearn.metrics import confusion_matrix
import itertools
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

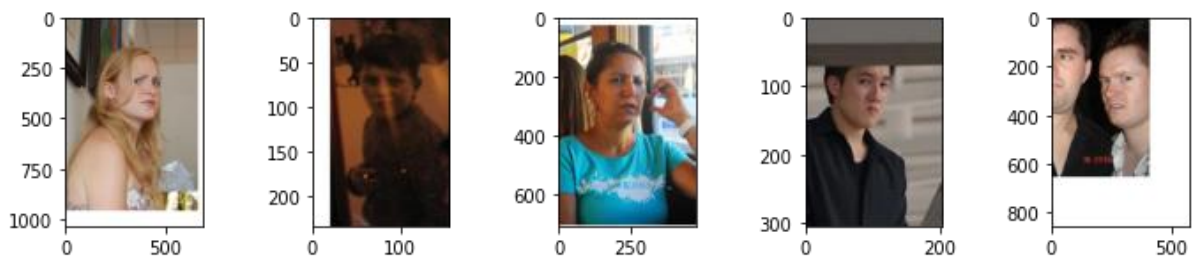
We are using a pyplot package for 2D graphics visualization, math package, and by sqrt, we mean “Square root”, for fundamental math operations. We are using NumPy for Numerical math ie., for the utilization of arrays. We are using Display for viewing and opening our visualization needed files. Here, we have also imported PIL which is an abbreviation for Python Image Library which is an exclusive package designed to view, open, manipulate, save all forms of image data contained in our datasets, programs. We have also imported tensorflow, as it is an intelligent open-source library, we can utilize it for data flow graphs and model building. Us developers can also utilize it to make giant layered neural network models. In general, we use them for predictions, perception, discovery, classifications, and creation. We have utilized the Keras package as it allows the distributed use of learning models in GPUs. As we are processing thousands of images files, we have imported an image data generator as it enables free rotation of the images to our desired degrees. Apart from this we have imported models and layers of the CNN model for the deep neural network build. For metrics purpositive for measuring the performance of our model we will utilize accuracy and confusion matrix. We used the pandas' package for data analysis.

4.3 Data Preparation

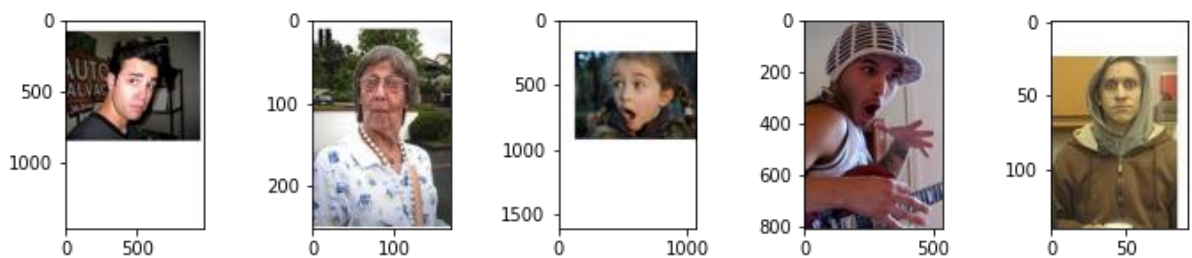
In the data preparation stage, we process the data for effective usage i.e., by adding, subtracting, few columns, or by manipulating or cleansing the data, like filling the outliers, removing the null values, etc. As in our project, we only have image data, the annotations are already given for this. We have classified by categorization into the range of 6 to 1 as mentioned above. For the preparation stage, we have transformed the text data into excel. We have introduced a loop function after mounting the drive to save in the specified folder. Here, the original data set will be gathered and we further split it and fit it into the range of emotions as categorized into 6 different folders in the drive 6,5,4,3,2 and 1. Here, we import the shuttle package library, read the annotations from the CSV file, and make a directory. In the latter stage, we test the annotations and train them by splitting the dataset package.

4.4 Displaying the trained Dataset

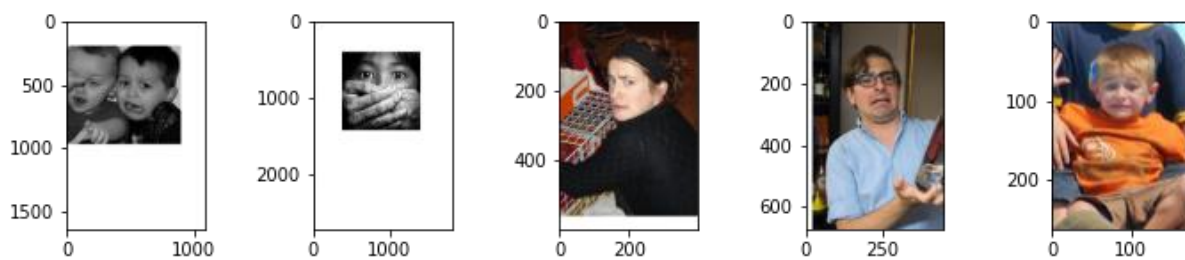
In this stage, we import the plot images, functions and define the image paths for the range of emotions, and define it in the list directory. We plot the images in 6 rows according to the corresponding emotion. As we split the data, we shall be using nine thousand images for training and the remaining 3 thousand for testing. Here, we display the sorted trained, tested dataset arranged according to its corresponding emotion of 1,2,3,4,5 and 6 as angry, sad, happy, disgust, fear, and surprise.



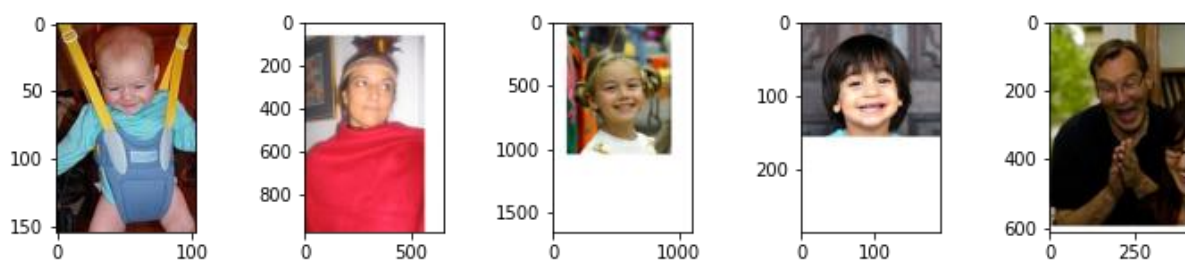
Output 1: Emotion disgust



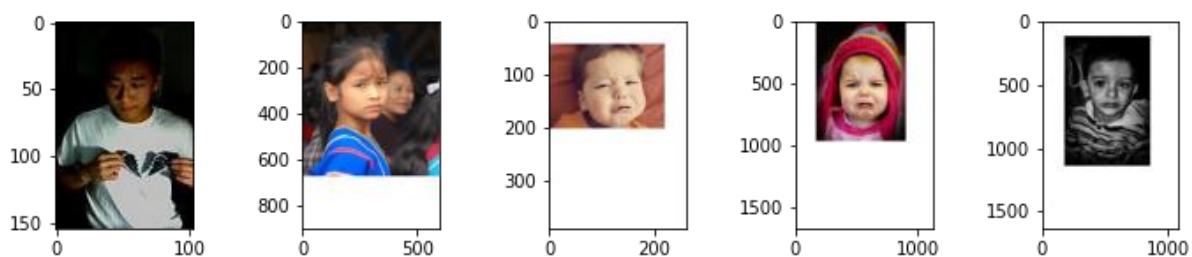
Output 2: Emotion Surprise



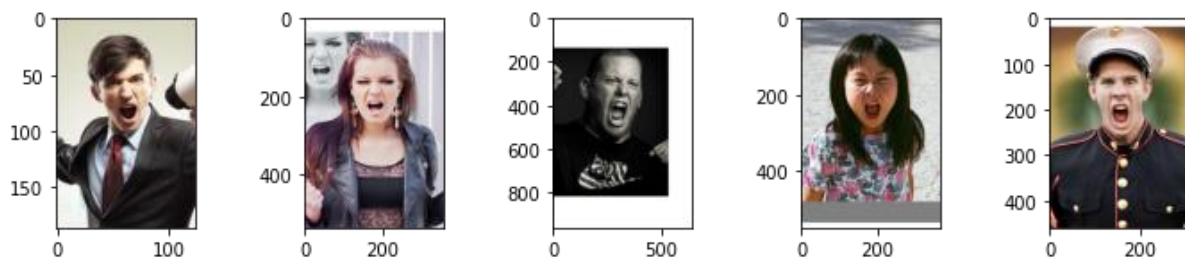
Output 3: Emotion Sad



Output 4: Emotion Happy



Output 5: Emotion Fear



Output 6: Emotion Angry

4.5 Building Face Detection Model

We are here implementing a face detection model imbibed from Convolutional Neural Networks, which works similar to Object Detection using CNN. For the same, we have divided into 6 number classes as aspired ranging from 1 to 6, corresponding to the emotions of Angry to Surprise. We have dragged the emotions to labeled allocations. We put the target size as 48 by 48, batch size as 64, rescaled the training, testing data with an Image data generator with the value of $1./255$. Here, we have introduced 2 directories specifically for training and testing and set the color mode and class mode as grayscale and categorical respectively for both train and test generator.

As we move to the literal model build, we have chosen the model of sequential and added 3 convolutional layers and fully connected layers. In the 1st Conv layer, parameter factors are 32, kernel size is 3 by 3, padding value is same, activation layer we used here is Relu and the input shape is 48 by 48. Here, for this conv2D layer, we added another max pooling layer with a pool size of 2 by 2. In the 2nd Conv layer, parameter factors are 64, kernel size is 3 by 3, padding value is same, activation layer we used here is Relu and the input shape is same. Here, for this conv2D layer, we added another max pooling layer with a pool size of 2 by 2.

In the 3rd Conv layer, parameter factors are 128, kernel size is 3 by 3, padding value is same, activation layer we used here is Relu and the input shape is same. Here, for this conv2D layer, we added another max pooling layer with a pool size of 2 by 2.

As we move to the FC layers, we have added a flatten layer, a dense layer of the parameters 128 with activation of Relu, a dropout layer with 0.5 param value, another dense layer with parameters of number classes, and a softmax activation function. We can view the summary of the build to check if it is built as expected or not.

Now, for the compilation of the model, we have calculated the loss according to the categorical cross-entropy with the Adam Optimizer. For the performance measurement, we have utilized the “Accuracy” function. The number of epochs we chose here is 50 to ensure we get better accuracy and correct predictions.


```

Found 9747 images belonging to 6 classes.
Found 3067 images belonging to 6 classes.
Model: "sequential_1"

Layer (type)                 Output Shape                 Param #
=====
conv2d_3 (Conv2D)            (None, 48, 48, 32)          320

max_pooling2d_3 (MaxPooling  (None, 24, 24, 32)          0
2D)

conv2d_4 (Conv2D)            (None, 24, 24, 64)          18496

max_pooling2d_4 (MaxPooling  (None, 12, 12, 64)          0
2D)

conv2d_5 (Conv2D)            (None, 12, 12, 128)         73856

max_pooling2d_5 (MaxPooling  (None, 6, 6, 128)           0
2D)

flatten_1 (Flatten)          (None, 4608)                 0

dense_2 (Dense)              (None, 128)                  589952

dropout_1 (Dropout)          (None, 128)                  0

dense_3 (Dense)              (None, 6)                   774

=====
Total params: 683,398
Trainable params: 683,398
Non-trainable params: 0

```

Output 7: Module Summary

```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:70: UserWarning: 'Model.fit_generator' is deprecated and will be removed in a future version.
Epoch 1/50
152/152 [=====] - 276s 2s/step - loss: 1.4527 - accuracy: 0.4866 - val_loss: 2.2926 - val_accuracy: 0.1416
Epoch 2/50
152/152 [=====] - 131s 861ms/step - loss: 1.3995 - accuracy: 0.4953 - val_loss: 2.4287 - val_accuracy: 0.1489
Epoch 3/50
152/152 [=====] - 131s 864ms/step - loss: 1.3846 - accuracy: 0.4989 - val_loss: 2.3383 - val_accuracy: 0.1483
Epoch 4/50
152/152 [=====] - 131s 865ms/step - loss: 1.3602 - accuracy: 0.5051 - val_loss: 2.2081 - val_accuracy: 0.1612
Epoch 5/50
152/152 [=====] - 132s 869ms/step - loss: 1.3393 - accuracy: 0.5070 - val_loss: 2.4020 - val_accuracy: 0.1546
Epoch 6/50
152/152 [=====] - 130s 856ms/step - loss: 1.3159 - accuracy: 0.5186 - val_loss: 2.4360 - val_accuracy: 0.1666
Epoch 7/50
152/152 [=====] - 131s 864ms/step - loss: 1.2863 - accuracy: 0.5283 - val_loss: 2.4873 - val_accuracy: 0.1795
Epoch 8/50
152/152 [=====] - 131s 866ms/step - loss: 1.2379 - accuracy: 0.5382 - val_loss: 2.4574 - val_accuracy: 0.1719
Epoch 9/50
152/152 [=====] - 131s 865ms/step - loss: 1.1632 - accuracy: 0.5709 - val_loss: 2.8709 - val_accuracy: 0.1712
Epoch 10/50
152/152 [=====] - 131s 863ms/step - loss: 1.1067 - accuracy: 0.5893 - val_loss: 2.5069 - val_accuracy: 0.1652
Epoch 11/50
152/152 [=====] - 131s 861ms/step - loss: 1.0355 - accuracy: 0.6214 - val_loss: 3.0836 - val_accuracy: 0.1789
Epoch 12/50
152/152 [=====] - 131s 866ms/step - loss: 0.9644 - accuracy: 0.6399 - val_loss: 3.1510 - val_accuracy: 0.1858
Epoch 13/50
152/152 [=====] - 132s 867ms/step - loss: 0.9005 - accuracy: 0.6677 - val_loss: 3.0041 - val_accuracy: 0.1695
Epoch 14/50
152/152 [=====] - 132s 866ms/step - loss: 0.8088 - accuracy: 0.6990 - val_loss: 3.6303 - val_accuracy: 0.1825
Epoch 15/50
152/152 [=====] - 130s 858ms/step - loss: 0.7489 - accuracy: 0.7189 - val_loss: 4.1064 - val_accuracy: 0.1795
Epoch 16/50
152/152 [=====] - 130s 856ms/step - loss: 0.6881 - accuracy: 0.7427 - val_loss: 4.1227 - val_accuracy: 0.1679
Epoch 17/50

```

Output 8: Display Epochs 1 to 17

```

Epoch 18/50
152/152 [=====] - 130s 860ms/step - loss: 0.5473 - accuracy: 0.7974 - val_loss: 5.3233 - val_accuracy: 0.1775
Epoch 19/50
152/152 [=====] - 131s 863ms/step - loss: 0.4819 - accuracy: 0.8227 - val_loss: 5.5606 - val_accuracy: 0.1695
Epoch 20/50
152/152 [=====] - 132s 867ms/step - loss: 0.4273 - accuracy: 0.8439 - val_loss: 5.2331 - val_accuracy: 0.1779
Epoch 21/50
152/152 [=====] - 131s 866ms/step - loss: 0.3807 - accuracy: 0.8617 - val_loss: 5.4422 - val_accuracy: 0.1785
Epoch 22/50
152/152 [=====] - 130s 857ms/step - loss: 0.3371 - accuracy: 0.8763 - val_loss: 7.2129 - val_accuracy: 0.1772
Epoch 23/50
152/152 [=====] - 129s 850ms/step - loss: 0.2962 - accuracy: 0.8954 - val_loss: 7.1562 - val_accuracy: 0.1702
Epoch 24/50
152/152 [=====] - 130s 855ms/step - loss: 0.2742 - accuracy: 0.9032 - val_loss: 7.4118 - val_accuracy: 0.1812
Epoch 25/50
152/152 [=====] - 130s 854ms/step - loss: 0.2475 - accuracy: 0.9139 - val_loss: 8.2572 - val_accuracy: 0.1725
Epoch 26/50
152/152 [=====] - 129s 852ms/step - loss: 0.2326 - accuracy: 0.9166 - val_loss: 7.4141 - val_accuracy: 0.1749
Epoch 27/50
152/152 [=====] - 130s 856ms/step - loss: 0.2043 - accuracy: 0.9279 - val_loss: 8.4503 - val_accuracy: 0.1692
Epoch 28/50
152/152 [=====] - 130s 856ms/step - loss: 0.1842 - accuracy: 0.9360 - val_loss: 9.6861 - val_accuracy: 0.1775
Epoch 29/50
152/152 [=====] - 130s 857ms/step - loss: 0.1787 - accuracy: 0.9356 - val_loss: 9.3747 - val_accuracy: 0.1765
Epoch 30/50
152/152 [=====] - 130s 855ms/step - loss: 0.1667 - accuracy: 0.9436 - val_loss: 9.4029 - val_accuracy: 0.1732
Epoch 31/50
152/152 [=====] - 130s 857ms/step - loss: 0.1449 - accuracy: 0.9499 - val_loss: 9.2152 - val_accuracy: 0.1735
Epoch 32/50
152/152 [=====] - 132s 867ms/step - loss: 0.1421 - accuracy: 0.9507 - val_loss: 10.5775 - val_accuracy: 0.1692
Epoch 33/50
152/152 [=====] - 132s 870ms/step - loss: 0.1403 - accuracy: 0.9507 - val_loss: 9.7530 - val_accuracy: 0.1888
Epoch 34/50
152/152 [=====] - 132s 870ms/step - loss: 0.1385 - accuracy: 0.9521 - val_loss: 11.1687 - val_accuracy: 0.1695

```

Output 9: Display Epochs 18 to 34

```

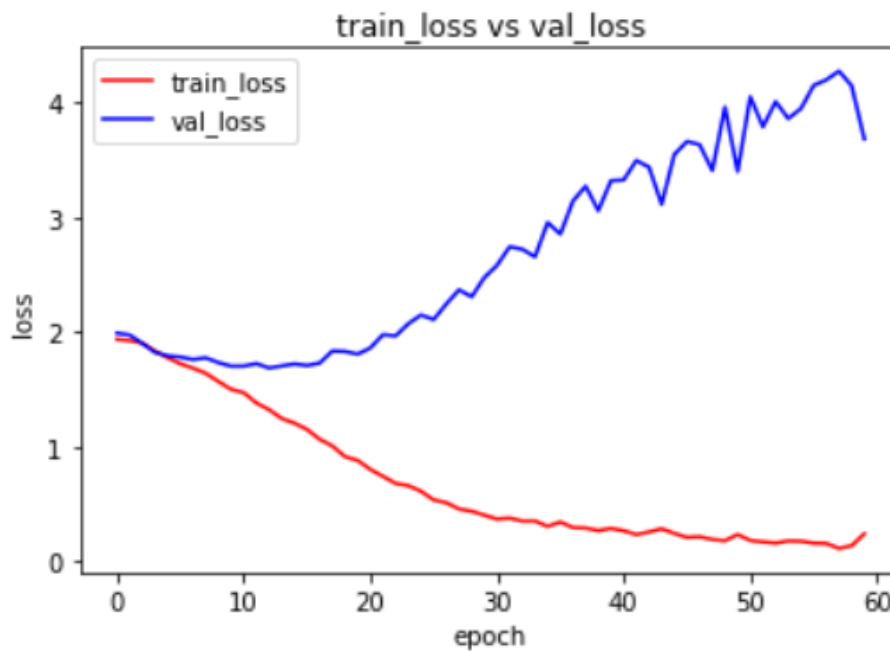
+ Code + Text
Epoch 34/50
152/152 [=====] - 132s 870ms/step - loss: 0.1385 - accuracy: 0.9521 - val_loss: 11.1687 - val_accuracy: 0.1695
Epoch 35/50
152/152 [=====] - 131s 863ms/step - loss: 0.1297 - accuracy: 0.9556 - val_loss: 12.0590 - val_accuracy: 0.1722
Epoch 36/50
152/152 [=====] - 132s 870ms/step - loss: 0.1166 - accuracy: 0.9594 - val_loss: 12.0587 - val_accuracy: 0.1662
Epoch 37/50
152/152 [=====] - 132s 870ms/step - loss: 0.1098 - accuracy: 0.9613 - val_loss: 11.8823 - val_accuracy: 0.1709
Epoch 38/50
152/152 [=====] - 131s 863ms/step - loss: 0.1119 - accuracy: 0.9622 - val_loss: 12.1548 - val_accuracy: 0.1722
Epoch 39/50
152/152 [=====] - 132s 867ms/step - loss: 0.0993 - accuracy: 0.9649 - val_loss: 14.0629 - val_accuracy: 0.1695
Epoch 40/50
152/152 [=====] - 132s 870ms/step - loss: 0.1165 - accuracy: 0.9592 - val_loss: 12.3918 - val_accuracy: 0.1755
Epoch 41/50
152/152 [=====] - 132s 869ms/step - loss: 0.1099 - accuracy: 0.9626 - val_loss: 12.6489 - val_accuracy: 0.1626
Epoch 42/50
152/152 [=====] - 131s 865ms/step - loss: 0.1012 - accuracy: 0.9640 - val_loss: 11.9835 - val_accuracy: 0.1749
Epoch 43/50
152/152 [=====] - 131s 865ms/step - loss: 0.0834 - accuracy: 0.9730 - val_loss: 14.9114 - val_accuracy: 0.1735
Epoch 44/50
152/152 [=====] - 131s 860ms/step - loss: 0.1065 - accuracy: 0.9627 - val_loss: 12.0697 - val_accuracy: 0.1709
Epoch 45/50
152/152 [=====] - 131s 863ms/step - loss: 0.0965 - accuracy: 0.9661 - val_loss: 12.1005 - val_accuracy: 0.1775
Epoch 46/50
152/152 [=====] - 131s 863ms/step - loss: 0.0920 - accuracy: 0.9695 - val_loss: 14.3290 - val_accuracy: 0.1626
Epoch 47/50
152/152 [=====] - 131s 861ms/step - loss: 0.0996 - accuracy: 0.9661 - val_loss: 14.8219 - val_accuracy: 0.1702
Epoch 48/50
152/152 [=====] - 130s 859ms/step - loss: 0.0996 - accuracy: 0.9634 - val_loss: 14.6375 - val_accuracy: 0.1666
Epoch 49/50
152/152 [=====] - 129s 849ms/step - loss: 0.0934 - accuracy: 0.9677 - val_loss: 13.9466 - val_accuracy: 0.1715
Epoch 50/50
152/152 [=====] - 130s 857ms/step - loss: 0.0918 - accuracy: 0.9706 - val_loss: 14.8670 - val_accuracy: 0.1676

```

Output 10: Display Epochs 35 to 50

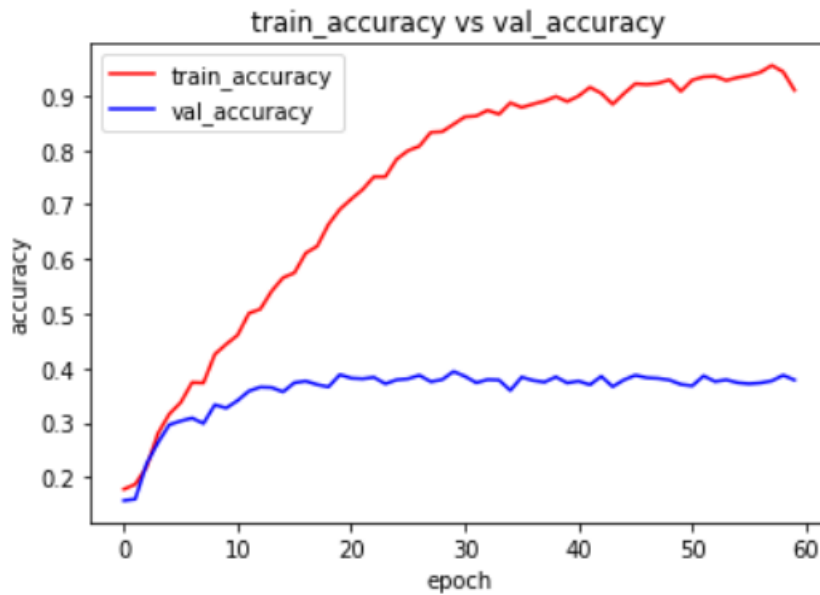
4.6 Accuracy Check

For checking the performance and measuring the score we are using the metrics of Accuracy as mentioned already. As accuracy calculates the number of times forecasts match labels. This measure generates two local variables, total and count, which are used to calculate how often y_{pred} matches y_{true} . This frequency is eventually expressed as binary accuracy, which is an idempotent operation that divides the total by count.



Output 11: Epochs Vs Loss in Training Vs Validation

From the above chart, we can say that the Loss graph is falling for training which is a good sign for a Neural network model. It has fallen from the 0th epoch to the 1st epoch and the value range has fallen from 1.94 to 1.94. Now, for the validation set, also the loss has fallen from 2 to 1.95 from the 0th epoch to the 1st epoch.



Output 12: Epochs Vs Accuracy in Training Vs Validation

From the above chart, we can say that the Accuracy graph is rising for training which is a good sign for a Neural network model. It has risen from the 0th epoch to the 1st epoch and the value range has increased from 0.185 to 0.195. Now, for the validation set, also the loss has risen from 0.155 to 0.160 from the 0th epoch to the 1st epoch. In the last stage of this step, we evaluate the model and we get the test results as thus- Loss = 14 and Accuracy = 0.1675.

```
# Evaluate Model
result = model.evaluate_generator(test_generator, steps=step_size_t
print("Test Loss: " + str(result[0]))
print("Test Accuracy: " + str(result[1]))
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: Use

Test Loss: 3.672173023223877

Test Accuracy: 0.3805803656578064

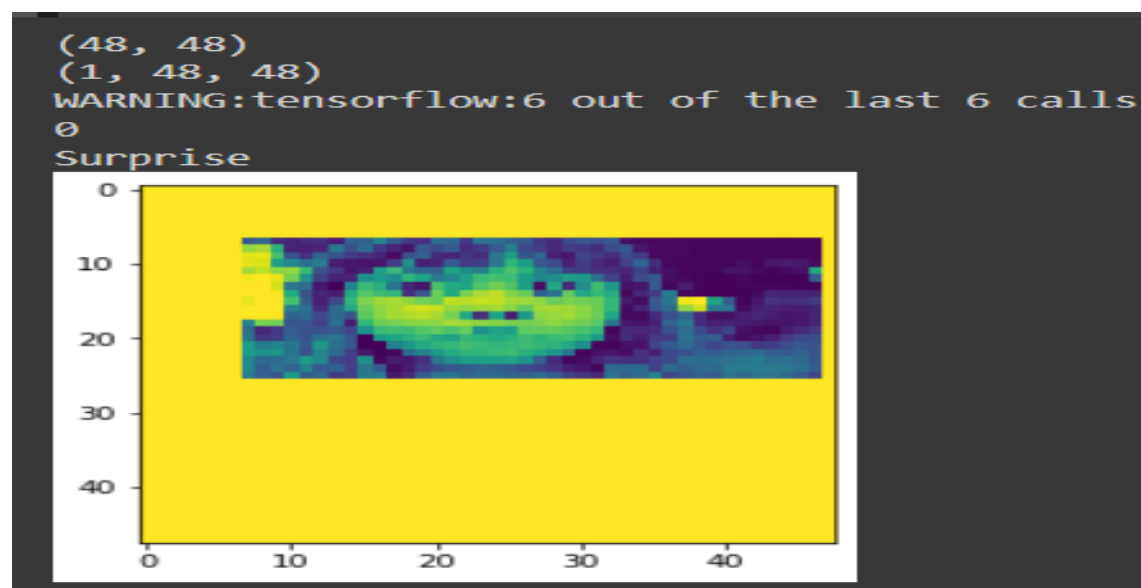
Output 13: Test Loss and Accuracy Results

4.7 Testing faces in utilized Database

Here, in this last step, we have divided this phase into 6 different stages each dedicated specifically to an emotion.

A. Surprise as Zero.

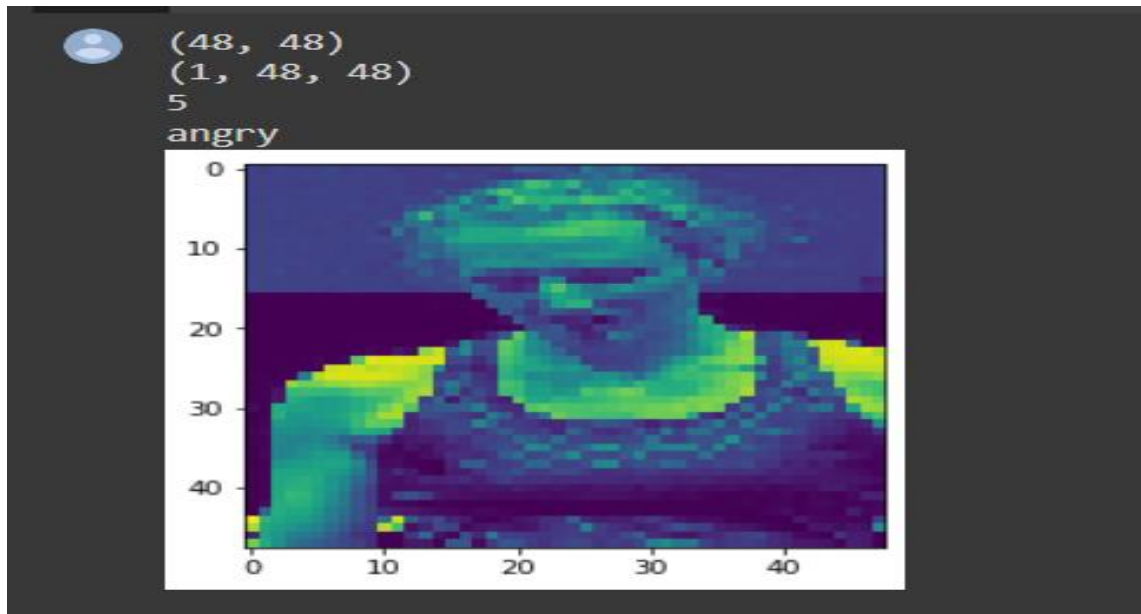
For the same, using an Op Dictionary, we have imported Keras pre-processing image, NumPy, pyplot. Now we have loaded any image and specified the target size as 48 by 48 with grayscale color mode. Now we have loaded the build model and reshaped the image to 1 by 48 by 48 by 1. From the saved model we are now predicting for Surprise. The result obtained is as below-



Output 14: Testing for Surprise

B. Angry as Five

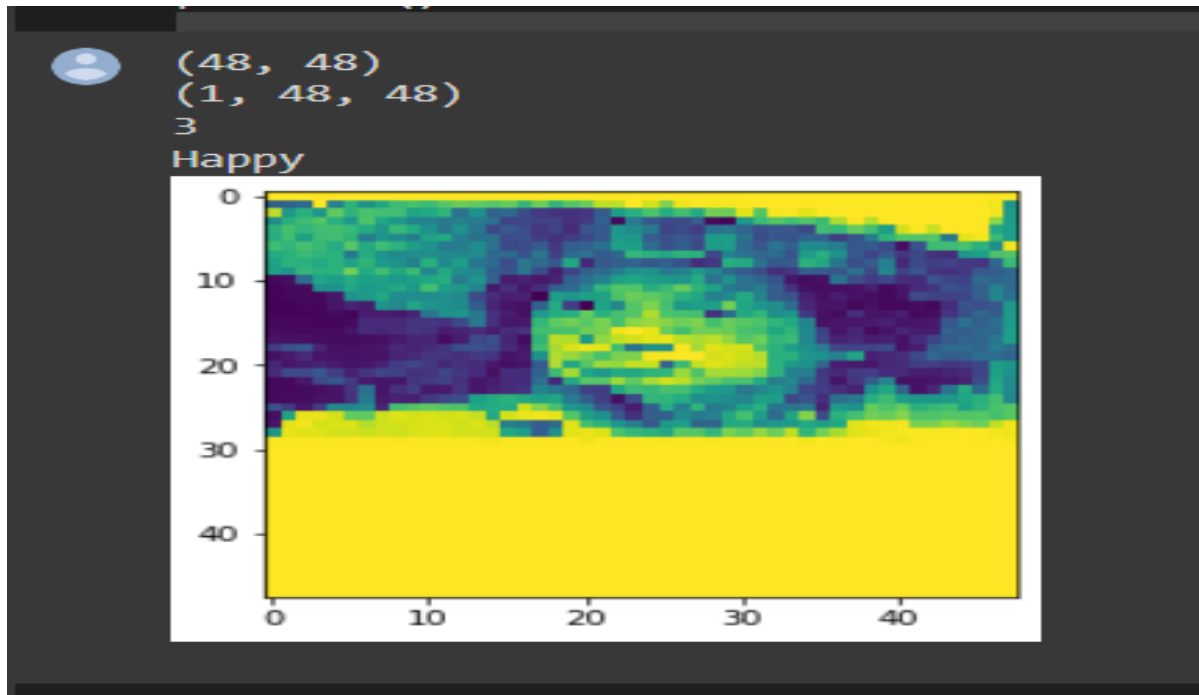
For the same, using an Op Dictionary, we have imported Keras pre-processing image, NumPy, pyplot. Now we have loaded any image and specified the target size as 48 by 48 with grayscale color mode. Now we have loaded the build model and reshaped the image to 1 by 48 by 48 by 1. From the saved model we are now predicting Angry emotion. The result obtained is as below-



Output 15: Testing for Angry

C. Happy as three

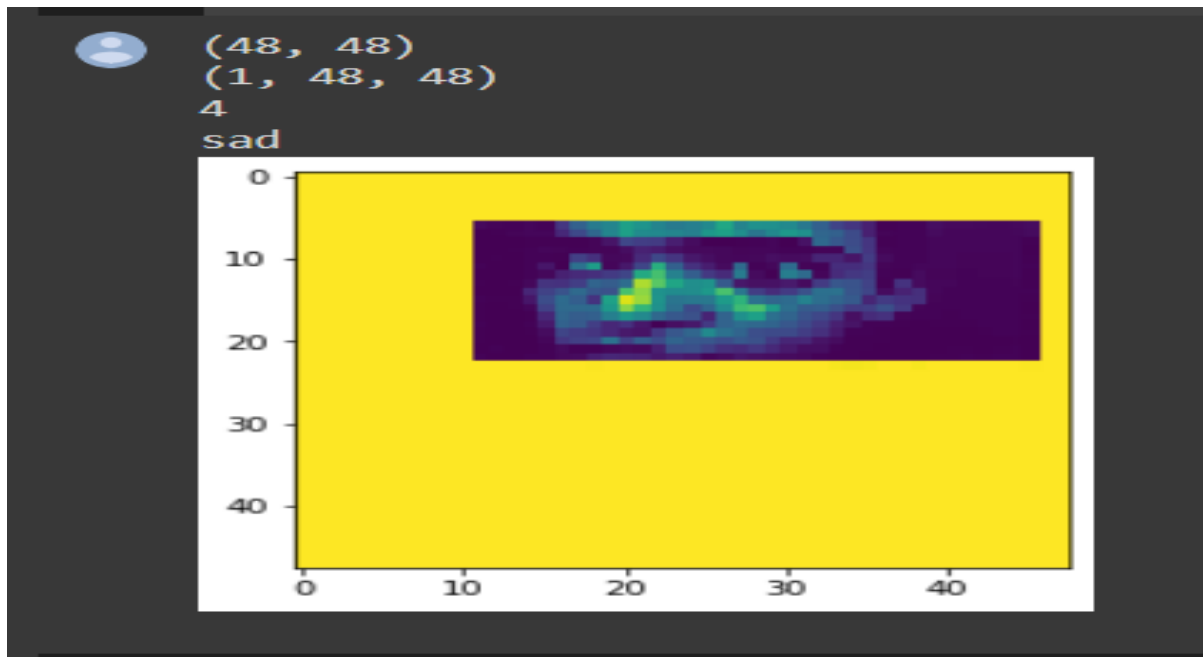
For the same, using an Op Dictionary, we have imported Keras pre-processing image, NumPy, pyplot. Now we have loaded any image and specified the target size as 48 by 48 with grayscale color mode. Now we have loaded the build model and reshaped the image to 1 by 48 by 48 by 1. From the saved model we are now predicting Happy emotions. The result obtained is as below-



Output 16: Testing for Happy

D. Sad for four

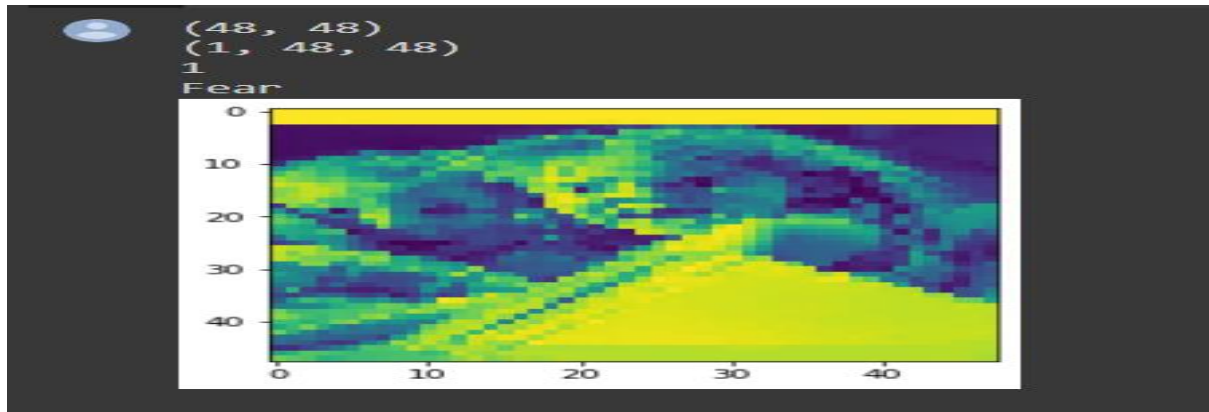
For the same, using an Op Dictionary, we have imported Keras pre-processing image, NumPy, pyplot. Now we have loaded any image and specified the target size as 48 by 48 with grayscale color mode. Now we have loaded the build model and reshaped the image to 1 by 48 by 48 by 1. From the saved model we are now predicting for Sad emotion. The result obtained is as below-



Output 17: Testing for Sad

E. Fear for One

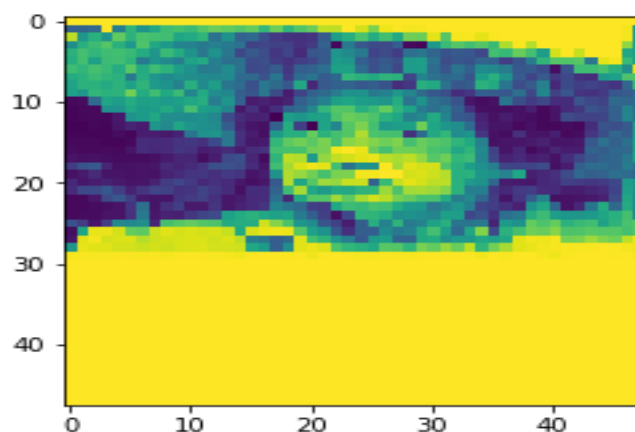
For the same, using an Op Dictionary, we have imported Keras pre-processing image, NumPy, pyplot. Now we have loaded any image and specified the target size as 48 by 48 with grayscale color mode. Now we have loaded the build model and reshaped the image to 1 by 48 by 48 by 1. From the saved model we are now predicting for Fear emotion. The result obtained is as below-



Output 18: Testing for Fear

F. Disgust for two

For the same, using an Op Dictionary, we have imported Keras pre-processing image, NumPy, pyplot. Now we have loaded any image and specified the target size as 48 by 48 with grayscale color mode. Now we have loaded the build model and reshaped the image to 1 by 48 by 48 by 1. From the saved model we are now predicting for Disgust emotion. The result obtained is as below-



Output 19: Testing for Disgust

Chapter- 5 Conclusion

Face detection can be used in a variety of fields, including defence surveillance, human-computer interface, robotics, transportation, and retrieval. In just a few hours, sensors used for long-term monitoring create petabytes of visual data. These data are geospatial and combined with other data to provide a clear picture of the present situation. Object detection is used in this procedure to monitor items such as people, cars, and suspicious objects in raw imaging data. Face detection applications include seeing and identifying wild animals in sterile zones such as industrial sites, as well as detecting cars parked in limited locations.

Thus, the CNN model is an efficient system for image detection which can further be used in various fields and applications for the benefit of science and mankind.

5.1 Challenges we face in Face detection-

- Classification and localization of objects are both top goals.
- Real-time detection speed
- Aspect ratios and many spatial scales
- Data is scarce.
- Inequality between classes

5.2 My Findings-

From the validation charts, we can say that the Loss graph is falling for training which is a good sign for a Neural network model. It has fallen from the 0th epoch to the 1st epoch and the value range has fallen from 1.94 to 1.94. Now, for the validation set, also the loss has fallen from 2 to 1.95 from the 0th epoch to the 1st epoch.

From the above chart, we can say that the Accuracy graph is rising for training which is a good sign for a Neural network model. It has risen from the 0th epoch to the 1st epoch and the value range has increased from 0.185 to 0.195. Now, for the validation set, also the loss has risen from 0.155 to 0.160 from the 0th epoch to the 1st epoch. In the last stage of this step, we evaluate the model and we get the test results as thus- Loss = 14 and Accuracy = 0.1675.

5.3 Future Scope of our study

- Face detection can be used in a variety of fields, including defence surveillance, human-computer interface, robotics, transportation, and retrieval.
- In just a few hours, sensors used for long-term monitoring create petabytes of visual data. These data are geospatialized and combined with other data to provide a clear picture of the present situation.
- Object detection is used in this procedure to monitor items such as people, cars, and suspicious objects in raw imaging data.
- Object detection applications include seeing and identifying wild animals in sterile zones such as industrial sites, as well as detecting cars parked in limited locations.

- When we are presented with a picture, our brain recognizes the items inside it very instantaneously. A machine, on the other hand, needs a lot of time and training data to recognize these items. However, because of recent advancements in technology and deep learning, the area of computer vision has gotten a lot simpler and more intuitive.
- Object detection technology has exploded in popularity across a wide range of sectors.
- It supports self-driving vehicles in safely navigating traffic, detecting aggressive behaviour in crowded areas, assisting sports teams in analyzing and creating scouting reports, and ensuring appropriate quality control of parts in production, among many other things.
- And this is only the tip of the iceberg in terms of what object detection technology can accomplish! There has been a huge and compelling development of Computer vision research lately. Portions of this accomplishment might be credited to the reception and transformation of AI techniques, while parts can be ascribed to the innovation of novel portrayals and models for explicit PC vision issues, just as the formation of proficient arrangements.
- Item location is one region where critical improvement has been made. The current investigation gives an outline of the article location considered.
- Because object detection and picture recognition are frequently confounded, it's critical to understand the differences between the two before moving forward.
- Object discovery is the underlying position in numerous PC vision frameworks since it permits more data about the recognized item and the scene to be gotten.
- Object discovery is the underlying position in numerous PC vision frameworks since it permits more data about the recognized item and the scene to be gotten. When an article case, like a face, has been identified, extra data can be gotten, for example,
- Recognizing the specific instance, such as identifying the subject's face.
- Tracking the object over an image sequence, such as tracking the face in a video,
- Extracting additional information about the object, such as determining the subject's gender.

5.4 Advantages and Disadvantages of our study

Face Detection Projects are no longer a concept, but a reality, thanks to years of study by some of the world's leading specialists. Face detection Projects and Object Detection Project Ideas have a bright future ahead of them. The breadth of technology is expanding all the time, as is the demand for specialists.

- Face detection programs have the huge benefit of being more accurate than human eyesight. The human brain is incredible, so much so that it can complete pictures with only a few bits of information. However, it can also prevent us from seeing what is truly there. Because our brains create assumptions, the whole image isn't always correct.
- Face detection projects, unlike the human brain, react to pictures based only on the data provided, not simply fragments of it. Although it may make inferences based on patterns, it lacks the human brain's inclination to jump to conclusions that aren't always correct. Object detection also works at a pixel level that the human brain is incapable of processing. Face detection projects can now offer more accurate findings as a result of this.
- Although the human brain is swift and efficient, computers are superior at multitasking, allowing object detection projects to give faster results in some cases. Face detection projects are capable of doing certain jobs for long periods.
- Using Face detection projects to complete projects not only gets the job done faster but also frees up time to focus on higher-level activities that demand human intellect. Using object detection projects to analyze X-ray pictures in a hospital context, for example, allows for speedier diagnosis, which might lead to faster treatment delivery at critical moments.
- After an object detection project has been taught, it can perform the same tasks with little effort, and it can even learn while doing so. This eliminates many hours of physical labor and the costs associated with it.
- Whether the resources saved by employing object detection projects are used to hire more employees to undertake higher-level jobs or to cover other costs associated with developing a firm, this technology saves money.
- When Face detection projects examine a picture with a specific objective in mind, all information that isn't linked to that goal is ignored. This reduces the bias that individuals, whether purposefully or inadvertently, may put into a process.
- Both online and in-store, Object detection initiatives have been utilized to improve the consumer experience. Based on photos in social media profiles, object detection can identify goods or brands that an individual is most likely to acquire via online platforms. In grocery shops, Amazon Go has revolutionized the shopping experience by recognizing goods in carts as consumers move forward in the line and charging them instantly, avoiding long checkout queues.

References

- J. Cassell, “Embodied Conversation: Integrating Face and Gesture into Automatic Spoken Dialogue Systems, In Luperfoy (ed.), Spoken Dialogue Systems, Cambridge, MA: MIT Press.
- F. Cassell, K.R. Thorisson, “The power of a nod and a glance: Envelope vs. Emotional Feedback in Animated Conversational Agents”, *Applied Artificial Intelligence* 13: 519- 538, 1999
- R.W. Picard, “Affective Computing”, MIT Press, Cambridge 1997.
- A. Young and H. Ellis (eds.), *Handbook of Research on Face Processing*, Elsevier Science Publishers 1989.
- G.W. Cottrell and J. Metcalfe, “EMPATH: Face, emotion, and gender recognition using holons”, in *Neural Information Processing Systems*, vol.3, pp. 564-571, 1991.
- C. Padgett, G. Cottrell, Identifying emotion in static face images, in *Proc. Of the 2nd Joint Symposium on Neural Computation*, Vol.5, pp.91-101, La Jolla, CA, University of California, San Diego.
- I. Essa and A. Pentland, Facial Expression Recognition using a Dynamic Model and Motion Energy, *Proc. of the International Conference on Computer Vision 1995*, Cambridge, MA, May 1995.
- Y. Yacoob, L.S. Davis, Computing Spatio-Temporal Representation of Human Faces, *IEEE Trans. On PAMI*, 18 (6), 636-642
- M. Black, Y. Yacoob, recognizing facial expressions in image sequences using local parameterized models of image motion, *Int. Journal of Computer Vision*, 25 (1), 1997, 23-48.
- M. Rosenblum, Y. Yacoob, L.S. Davis, Human Expression Recognition from Motion Using a Radial Basis Function Network Architecture, *IEEE Trans. On Neural Networks*, 7 (5), 1121-1138, 1996.
- I. Essa and A. Pentland, Facial Expression Recognition using a Dynamic Model and Motion Energy, *Proc. of the International Conference on Computer Vision 1995*, Cambridge, MA, May 1995.
- I. Essa, “Analysis, Interpretation, and Synthesis of Facial Expressions, Ph.D. Thesis, Massachusetts Institute of Technology (MIT Media Laboratory)
- J. Lien, T. Kanade, J. Cohn, C. Li, Subtly different Facial Expression Recognition And Expression Intensity Estimation, in *Proc. Of the IEEE Int. Conference on Computer Vision and Pattern Recognition*, pp. 853-859, Santa Barbara, Ca, June 1998.
- M. Pantic, L. Rothkrantz, “Automatic Analysis of Facial Expressions: The State of the Art”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, N. 12, pp. 1424-1445, 2000
- AboBakr, A. and Azer, M.A., 2017, December. IoT ethics challenges and legal issues. In *2017 12th International Conference on Computer Engineering and Systems (ICCES)* (pp. 233-237). IEEE.
- Bargshady, G., Soar, J., Zhou, X., Deo, R.C., Whittaker, F. and Wang, H., 2019, February. A joint deep neural network model for pain recognition from the face. In *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)* (pp. 52-56). IEEE.
- Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I. and Zafeiriou, S., 2019. Retinaface: Single-stage dense face localization in the wild. *arXiv preprint arXiv:1905.00641*.
- Eaton, A.A., and Stephens, D.P., 2020. Reproductive Justice Special Issue Introduction “Reproductive Justice: Moving the Margins to the Center in Social Issues Research”. *Journal of Social Issues*, 76(2), pp.208-218.
- Jang, Y., Gunes, H. and Patras, I., 2019. Registration-free face-SSD: Single-shot analysis of smiles, facial attributes, and affect in the wild. *Computer Vision and Image Understanding*, 182, pp.17-29.

- Jiang, W., Wang, Z., Jin, J.S., Han, X. and Li, C., 2019. Speech emotion recognition with heterogeneous feature unification of deep neural network. *Sensors*, 19(12), p.2730.
- Jyoti, S., Sharma, G. and Dhall, A., 2018, December. A single hierarchical network for face, action unit, and emotion detection. In *2018 Digital Image Computing: Techniques and Applications (DICTA)* (pp. 1-8). IEEE.
- Saha, T., Patra, A., Saha, S. and Bhattacharyya, P., 2020, July. Towards Emotion-aided Multi-modal Dialogue Act Classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 4361-4372).
- Thomas, D.R., Pastrana, S., Hutchings, A., Clayton, R. and Beresford, A.R., 2017, November. Ethical issues in research using datasets of illicit origin. In *Proceedings of the 2017 Internet Measurement Conference* (pp. 445-462).
- Vu, T.H., Dang, A. and Wang, J.C., 2019. A Deep Neural Network for Real-Time Driver Drowsiness Detection. *IEICE TRANSACTIONS on Information and Systems*, 102(12), pp.2637-2641.
- Yang, L., Jiang, D., Han, W. and Sahli, H., 2017, October. DCNN and DNN based multi-modal depression recognition. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ASCI)* (pp. 484-489). IEEE.
- Yang, L., Liu, Y. and Peng, J., 2019. An automatic detection and identification method of welded joints based on a deep neural network. *IEEE Access*, 7, pp.164952-164961.
- Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X. and Li, S.Z., 2017. S3fd: Single-shot scale-invariant face detector. In *Proceedings of the IEEE international conference on computer vision* (pp. 192-201).
- Zhuang, N., Yan, Y., Chen, S., Wang, H., and Shen, C., 2018. Multi-label learning-based deep transfer neural network for facial attribute classification. *Pattern Recognition*, 80, pp.225-240.
- Qian, H., Xu, J., and Zhou, J., 2018. Object Detection Using Deep Convolutional Neural Networks. 2018 Chinese Automation Congress (CAC),
- Tao, J., Wang, H., Zhang, X., Li, X., and Yang, H., 2017. An object detection system based on YOLO in traffic scene. 2017 6th International Conference on Computer Science and Network Technology (ICCSNT),
- Malhotra, P. and Garg, E., 2020. Object Detection Techniques: A Comparison. 2020 7th International Conference on Smart Structures and Systems (ICSSS).
- Marty, M., Banerjee, S., and Sinha Chaudhuri, S., 2021. Faster R-CNN and YOLO based Vehicle detection: A Survey. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC).
- Chandan, G., Jain, A., Jain, H. and Mohana, 2018. Real-Time Object Detection and Tracking Using Deep Learning and OpenCV. 2018 International Conference on Inventive Research in Computing Applications (CIRCA).
- Chen, P., Shi, Y., Zheng, Q. and Wu, Q., 2020. State-of-the-art Object Detection Model Based on YOLO. 2020 International Conference on Computer Network, Electronic and Automation (ICCNEA),.
- Li, K. and Cao, L., 2020. A Review of Object Detection Techniques. 2020 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT).
- Algarve, A., Garcia, A., and Hofmann, A., 2017. Real-Time Object Detection and Classification of Small and Similar Figures in Image Processing. 2017 International Conference on Computational Science and Computational Intelligence (CSCI),.

- Ahmad, T., Ma, Y., Yahya, M., Ahmad, B., Nazir, S. and Haq, A., 2020. Object Detection through Modified YOLO Neural Network. *Scientific Programming*, 2020, pp.1-10.
- Ajeet Ram Pathak, Manjusha Pandey, Siddharth Rautaray. "Application of Deep Learning for Object Detection", *Procedia Computer Science*, 2018
- Yongju Choi, Othmane Atif, Jonguk Lee, Daihee Park, Yongwha Chung. "Noise-Robust Sound-Event Classification System with Texture Analysis", *Symmetry*, 2018
- Makram Soui, Nesrine Mansouri, Raed Alhamad, Marouane Kessentini, Khaled Ghedira. "NSGA-II as feature selection technique and AdaBoost classifier for COVID-19 prediction using patient's symptoms", *Nonlinear Dynamics*, 2021
- M Sarosa, N Muna, E Rohadi. "Detection of natural disaster victims using You Only Look Once (YOLO)", *IOP Conference Series: Materials Science and Engineering*, 2021
- "Advances in Artificial Intelligence and Security", Springer Science and Business Media LLC, 2021
- Mariam L. Francies, Mohamed M. Ata, Mohamed A. Mohamed. " A robust multiclass object recognition based on modern deep learning algorithms ", *Concurrency and Computation: Practice and Experience*, 2021
- Prithvi N. Amin, Sayali S. Moghe, Sparsh N. Prabhakar, Charusheela M. Nehete. "Deep Learning-Based Face Mask Detection and Crowd Counting", 2021 6th International Conference for Convergence in Technology (I2CT), 2021
- G Chandan, Ayush Jain, Harsh Jain, Mohana. "Real-Time Object Detection and Tracking Using Deep Learning and OpenCV", 2018
- Weixing Zhang, Chandi Witharana, Anna Liljedahl, Mikhail Kanevskiy. "Deep Convolutional Neural Networks for Automated Characterization of Arctic Ice- Wedge Polygons in Very High Spatial Resolution Aerial Imagery", *Remote Sensing*, 2018
- Nashwan Adnan OTHMAN, Ilhan AYDIN. "A New Deep Learning Application Based on Movidius NCS for Embedded Object Detection and Recognition", 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2018
- Mayank Raj, Swet Chandan. "Real-Time Vehicle and Pedestrian Detection Through SSD in Indian Traffic Conditions", 2018 International Conference on Computing, Power and Communication Technologies (GUCON), 2018
- Tao Ye, Xi Zhang, Yi Zhang, Jie Liu. "Railway Traffic Object Detection Using Differential Feature Fusion Convolution Neural Network", *IEEE Transactions on Intelligent Transportation Systems*, 2021
- Widanagamaachchi, Wathsala. (2009). Emotion Recognition with Image Processing and Neural Networks. 27th National Information Technology Conference.
- Kim, & Joo, Young Hoon & Park, Jin. (2015). Emotion Detection Algorithm Using Frontal Face Image.
- R. S. Deshmukh, V. Jagtap and S. Paygude, "Facial emotion recognition system through machine learning approach," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 272-277, doi: 10.1109/ICCONS.2017.8250725.
- André, E., Rehm, M., Minker, W., and Bühler, D. (2004). "Endowing spoken language dialogue systems with emotional intelligence," in *Affective Dialogue Systems Tutorial and Research Workshop, ADS 2004*, eds E. Andre, L. Dybkjaer, P. Heisterkamp, and W. Minker (Germany: Kloster Irsee), 178–187.
- Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. doi:10.1145/3065386
- Mao, Qirong; Dong, Ming; Huang, Zhengwei; Zhan, Yongzhao (2014). Learning Salient Features for Speech Emotion Recognition Using Convolutional Neural Networks. *IEEE Transactions on Multimedia*, 16(8), 2203–2213. doi:10.1109/TMM.2014.2360798

- Agustsson, E., Timofte, R., Escalera, S., Baro, X., Guyon, I., Rothe, R., ´ 2017. Apparent and real age estimation in still images with deep residual regressors on the appa-real database, in FG, IEEE Computer Society. pp. 87–94.
- An, L., Yang, S., Bhanu, B., 2015. Efficient smile detection by an extreme learning machine. *Neurocomputing*. 149, 354–363.
- Chang, W., Hsu, S., Chien, J., 2017. Fatauva-net: An integrated deep learning framework for facial attribute recognition, action unit detection, and valence-arousal estimation, in 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1963–1971.
- Chen, J., Ou, Q., Chi, Z., Fu, H., 2017a. Smile detection in the wild with deep convolutional neural networks. *Machine Vision Applications* 28, 173–183.
- Chen, S., Zhang, C., Dong, M., Le, J., Rao, M., 2017b. Using ranking CNN for age estimation, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. ImageNet: A Large-Scale Hierarchical Image Database, in CVPR.
- Ehrlich, M., Shields, T.J., Almaev, T., Amer, M.R., 2016. Facial attributes classification using multi-task representation learning, in 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 752–760.
- Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision* 88, 303–338.
- Girshick, R., 2015. Fast R-CNN, in: Proceedings of the International Conference on Computer Vision (ICCV).
- Han, H., Jain, A.K., Wang, F., Shan, S., Chen, X., 2017. Heterogeneous face attribute estimation: A deep multi-task learning approach. *IEEE Trans. on PAMI*.
- Hand, E.M., Chellappa, R., 2017. Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification, in AAAI, AAAI Press. pp. 4068–4074.
- Hao, Z., Liu, Y., Qin, H., Yan, J., Li, X., Hu, X., 2017. Scale-aware face detection, in CVPR.
- He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., Li, X., 2017. Single-shot text detector with regional attention, in IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22–29, 2017, pp. 3066–3074.
- Hsu, G.S.J., Cheng, Y.T., Ng, C.C., Yap, M.H., 2017. Component biologically inspired features with moving segmentation for age estimation, in 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 540–547 <http://mplab.ucsd.edu>, 2009. The MPLab GENKI Database, GENKI-4K Subset.
- Jain, V., Crowley, J.L., 2013. Smile Detection Using Multi-scale Gaussian Derivatives, in 12th WSEAS International Conference on Signal Processing, Robotics, and Automation, Cambridge, United Kingdom. URL: <https://hal.inria.fr/hal-00807362>.
- Jang, Y., Gunes, H., Patras, I., 2017. silent: Registration-Free Smiling Face Detection in the Wild, in The IEEE International Conference on Computer Vision (ICCV) Workshops.
- Jourabloo, A., Liu, X., 2017. Pose-invariant face alignment via CNN-based dense 3d model fitting. *Int. J. Comput. Vision* 124, 187–203
- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.
- A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861, 2017.

- P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- D. Kollias, S. Cheng, M. Pantic, and S. Zafeiriou. Photorealistic facial synthesis in the dimensional effect space. In *European Conference on Computer Vision*, pages 475–491. Springer, 2018.
- M. Kuchnik and V. Smith. Efficient augmentation via data subsampling. *arXiv preprint arXiv:1810.05222*, 2018.
- Kuhnke, L. Rumberg, and J. Ostermann. Two-stream aural-visual effect analysis in the wild. *arXiv preprint arXiv:2002.03399*, 2020.
- Liu, J. Zeng, S. Shan, and X. Chen. Emotion recognition for in-the-wild videos. *arXiv preprint arXiv:2002.05447*, 2020.
- M. Najibi, P. Samangouei, R. Chellappa, and L. Davis. SSH: Single-stage headless face detector. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.
- J. Pahl, I. Rieger, and D. Seuss. Multi-label class balancing algorithm for action unit detection. *arXiv preprint arXiv:2002.03238*, 2020.
- K. Rapantzikos, N. Tsapatsoulis, Y. Avrithis, and S. Kollias. Bottom-up spatiotemporal visual attention model for video analysis. *IET Image Processing*, 1(2):237–248, 2007.
- Ringeval, B. Schuller, M. Valstar, N. Cummins, R. Cowie, L. Tavabi, M. Schmitt, S. Alisamir, S. Amiriparian, E.-M. Messner, et al. Avec 2019 workshop and challenge: state-of-mind, detecting depression with ai, and cross-cultural affect recognition. In *Proceedings of the 9th International on Audio/Visual Emotion Challenge and Workshop*, pages 3–12, 2019.
- J. A. Russell. Evidence of convergent validity on the dimensions of effect. *Journal of personality and social psychology*, 36(10):1152, 1978.
- M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- C. Whissel. *The dictionary of affect in language, emotion: Theory, research, and experience: vol. 4, the measurement of emotions*, r. Plutchik and H. Kellerman, Eds., New York: Academic, 1989.
- S. Yang, P. Luo, C. C. Loy, and X. Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- S. Zafeiriou, D. Kollias, M. A. Nicolaou, A. Papaioannou, G. Zhao, and I. Kotsia. Aff-wild: Valence and arousal in-the-wild challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 34–41, 2017.
- Y.-H. Zhang, R. Huang, J. Zeng, S. Shan, and X. Chen. m3 t: Multi-modal continuous valence-arousal estimation in the wild. *arXiv preprint arXiv:2002.02957*, 2020.
- H. Zhou, D. Meng, Y. Zhang, X. Peng, J. Du, K. Wang, and Y. Qiao. Exploring emotion features and fusion strategies for audio-video emotion recognition. In *2019 International Conference on Multimodal Interaction*, pages 562–566, 2019.
- J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.
- R. Alp Guler, N. Neverova, and I. Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 2, 3
- A. Bulat and G. Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? In *ICCV*, 2017. 6
- Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*, 2016. 5

D. Chen, G. Hua, F. Wen, and J. Sun. Supervised transformer network for efficient face detection. In ECCV, 2016. 2

D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In ECCV, 2014. 1, 2

T. Chen, T. Moreau, Z. Jiang, L. Zheng, E. Yan, H. Shen, M. Cowan, L. Wang, Y. Hu, L. Ceze, et al. Tvm: An automated end-to-end optimizing compiler for deep learning. In OSDI, 2018.

C. Chi, S. Zhang, J. Xing, Z. Lei, S. Z. Li, and X. Zou. Selective refinement network for high performance face detection. AAAI, 2019. 1, 5

Thank you Note:

I Would like to thank professor Dimitris in helping me with this project and providing me required guidelines, base papers for reference and Annotated Image data for building my model.