IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

# Emotional Recognition of Students through Facial Expressions: A Deep Learning Approach using Convolutional Neural Network

**Authors:**

**S Sree Rachnae(1BM20IS126)**
Department of Information Science
BMS College of Engineering
Bengaluru, Karnataka, India

**Shreyansh Tripathi(1BM20IS148)**
Department of Information Science
BMS College of Engineering
Bengaluru, Karnataka, India

**Raj Aryan(1BM20IS114)**
Department of Information Science
BMS College of Engineering
Bengaluru, Karnataka, India

**R Indra(Assistant Professor)**
Department of Information Science
BMS College of Engineering
Bengaluru, Karnataka, India

B.M.S College of Engineering, Bangalore – 560019
Department of Information Science and Engineering
Multi-Disciplinary Project

***Abstract – Recognising emotions of humans using their facial expressions is an interesting research which has been previously taken up by many researchers in fields such as safety, health and human machine interfaces. One of the most widely used techniques for this application is using various Deep Learning methods. We have taken the approach of using the Deep Learning method of Convolutional Neural Networks. The key aspect of this paper is to make a study of how to imbibe Deep Learning in order to perform emotional recognition of students with the help of their facial expressions and give it as an input to their respective teachers or counsellors. Facial expressions play a vital in determining and analysing emotions showcased by students. Deep learning and image classification methods are used for perceiving expressions and classify them according to the images. Various datasets are used for training such recognition models.***

***Keywords – Convolutional Neural Network, Facial Expression, Emotional Recognition, Emotions***

## I. INTRODUCTION

Emotions, a way humans like us use to express our thoughts and feelings to the outside with our facial actions. The word ‘Emotion’ has its first records from the 1570s; it ultimately comes from the Latin word “emotere,” literally means “energy in motion”. Expressing these emotions is also a part of our intelligence, Peter Salovey and John D. Mayer coined the term 'Emotional Intelligence in 1990 describing it as “a form of social intelligence that involves the ability to monitor one's own and other's feelings and emotions, to discriminate among them, and to use this information to guide one's thinking and action”. Emotions are crucial things which we experience in our daily lives, influencing our decisions and relationships, it something we learn with no teaching from our childhood. Our emotions get enhanced and develop as we grow up and face various situations and experiences.

The ability to understand and interpret these human emotions is a basic fundamental for every human, now teaching and training a machine the same way is where this field of study becomes interesting. So far such emotional recognitions have been carried out on videos, audios and psychological data. We can attain our desired accuracy by using a combination of different methodologies to analyse these human emotions, our approach mainly pertains to the use of Convolutional Neural Networks.

Various such detection contributions have lead to the emergence of dedicated recognitions processes for all kinds of utilizations of this so-called emotional or emotive internet. By building a neural network model with certain depth and combining nonlinear operations such as convolution and pooling, we can realize two important functions of imitating the hierarchical processing of human brain and local perception of visual nerve.

## II. RELATED WORK

In-depth studies and projects on Emotional Recognition using Facial Expressions using Convolutional Neural Networks have been conducted. Here are a few illustrations:

Emotion Recognition from Facial Expression using Deep Learning Sujit Tilak1, Ankit Gupta², Rahul Arekar³, Ankita Arondekar4 : The paper discusses the importance of facial expression recognition and the growing need for it. While there are existing methods using machine learning and artificial intelligence, this work focuses on deep learning and image classification techniques to recognize and classify various facial expressions.

Emotion AI, Real-Time Emotion Detection using CNN Tanner Gilligan M.S. Computer Science Stanford University tanner12@stanford.edu Baris Akis B.S. Computer Science Stanford University bakis@stanford.edu : This research paper introduces a method for detecting human emotions in real-time using Convolutional Neural Networks (CNNs). The main objective is to create a system that can accurately identify emotions, which can be useful in various applications.

Facial expression recognition based on CNN Mingjie Wang1 , Pengcheng Tan2 , Xin Zhang3 , Yu Kang4 , Canguo Jin5 , Jianying Cao1* : The study found that the proposed method achieved a recognition rate of over 70% on the training set and over 80% on the test set, which is considered good performance. The results show that CNNs are effective in recognizing and classifying facial expressions. It provides insights for future research and highlights the need to explore more complex scenarios and expand the dataset for better results.

Enhanced Automatic Recognition of Human Emotions Using Machine Learning Techniques Monisha.G.Sa ,Yogashree.G.Sb , Baghyalaksmi.Rc ,Haritha.Pd : Facial expression recognition is a novel way to express the emotion of humans beginning using a convolution neural network (CNN). The ability to accurately determine emotions was greatly enhanced by the removal of the background. The calculation time for this approach to recognize the emotions on a human face are 15.3 seconds. This helps in fetching the features of the human face in a more advanced manner, as well as determining the mood of the individual.

Facial emotion recognition using deep learning: review and insights Wafa Mellouka* , Wahida Handouzia : This paper signifies interest of researchers in Facial Expression Recognition (FER) via deep learning over recent years. The automatic FER task goes through different steps like: data processing, proposed model architecture and finally emotion recognition. Researchers achieve high precision in FER by applying CNN networks with spatial data, researchers used the combination between CNN-RNN especially, LSTM network, this indicates that CNN is the network basis of deep learning for FER.

## III. ALGORITHM

### A. Convolutional Neural Networks (CNNs) :

CNNs are deep learning models commonly used for image processing tasks, including image recognition and classification. They are inspired by the human visual system, where they learn to recognize patterns and features from raw pixel values. A typical CNN consists of multiple layers:

Convolutional layers: These layers use small filters (also known as kernels) to slide across the input image, performing element-wise multiplication and addition to generate feature maps. These feature maps capture different aspects of the input image, such as edges, textures, and patterns. Some important terms are :

- **Activation functions:** After convolution, activation functions (e.g., ReLU) introduce non-linearity to the model, enabling it to learn complex relationships between features.
- **Pooling layers:** Pooling layers downsample the feature maps, reducing the spatial dimensions and retaining essential information. Max pooling is a common approach, which takes the maximum value within a specific region.
- **Fully connected layers:** These layers connect every neuron from the previous layer to every neuron in the subsequent layer. They are used for high-level feature extraction and decision making.
- CNNs are trained through a process called **back-propagation**, where the model's weights are adjusted based on the error between predicted and actual labels, iteratively optimizing the network's performance.

Some common approaches for CNN approach is :

- **Dataset Development:** Multiple labeled facial expression datasets, including CK+ and JAFFE, were collected. Custom images were also added to enhance the dataset.
- **Data Pre-processing:** Images were converted to grayscale, faces were detected using OpenCV, and facial components were extracted. Images were rescaled and subjected to statistical pre-processing techniques like Gaussian filtering and mean subtraction.
- **CNN Construction:** Pre-trained LeNet and AlexNet models were utilized, with the first and last layers retrained. Different learning rate methods and parameters were experimented with to ensure a stable model.
- **Real-time Interface:** OpenCV captured webcam images. Faces were extracted, pre-processed, and sent to a server (AWS) for processing using the CNN model. Prediction results were sent back to the local interface.

### B. Viola-Jones face Detection Algorithm :

The Viola-Jones algorithm is a popular real-time face detection technique. It works by using Haar-like features, which are simple rectangular filters, to analyze regions of the image and determine if they contain a face. The algorithm goes through the following steps:

Haar feature extraction: Haar-like features represent the contrast between adjacent regions of the image. These features are applied at different scales and positions to detect patterns that resemble face-like structures. Some of its aspects are :

- **Integral image representation:** To speed up computations, the integral image is created, which allows rapid computation of rectangular area sums within the image.
- **AdaBoost training:** The algorithm uses the AdaBoost machine learning algorithm to select the most relevant Haar-like features and create a strong classifier capable of distinguishing between face and non-face regions.
- **Cascading classifier:** A cascading approach is used to reject non-face regions quickly. The classifier has multiple stages, and if a region fails to pass any of the stages, it is immediately discarded.
- By combining these steps, the Viola-Jones algorithm achieves efficient face detection in images and has been widely used in various computer vision applications.

Both the CNN model and the Viola-Jones face detection algorithm play essential roles in the facial expression recognition task, enabling accurate identification and classification of facial expressions from the collected dataset

### C. Alphanet Recognition Model :

“Alphanet," a method for emotion recognition. It utilizes both deep neural networks and Bayesian classifiers for accurate emotion prediction.

- **Bottom-up module:** This part employs Convolutional Neural Networks (CNNs) to analyze emotions displayed by individual faces. It detects and processes faces, predicting emotions as positive, neutral, or negative. The final group emotion is determined by combining predictions from all faces.

- **Top-down module:** This module considers the overall scene context using a scene descriptor. A Bayesian network calculates probabilities to understand the group's overall emotion as positive, neutral, or negative.
- **Combining results:** The outputs from the bottom-up and top-down modules are merged to obtain a more precise determination of the group's emotion.

**D. Bayesian Classifiers :**

The **Bayesian classifier** approach provides a principled way to perform classification by incorporating prior knowledge and updating it based on new evidence. Bayesian classifiers are a class of statistical classifiers that use the principles of Bayesian probability to make predictions or decisions about the class label of a given input instance. These classifiers are based on Bayes' theorem, which provides a way to update our beliefs about an event when new evidence becomes available. The approach of Bayesian classifiers is as follows :

1. **Bayes' Theorem :** The foundation of Bayesian classifiers is **Bayes' theorem,** which states that the probability of a hypothesis (class label) given the evidence (input features) is proportional to the likelihood of the evidence given the hypothesis, multiplied by the prior probability of the hypothesis.

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

$A, B$ = events

$P(A|B)$ = probability of A given B is true

$P(B|A)$ = probability of B given A is true

$P(A), P(B)$ = the independent probabilities of A and B

2. **Training Phase:**

- **Prior Probability Estimation:** In the training phase, the prior probabilities of different class labels are estimated from the training dataset. This involves calculating the frequency or relative occurrence of each class label in the dataset.
- **Likelihood Estimation:** For each feature in the dataset, the likelihood of observing a particular value given a specific class label is calculated. This is often done using the relative frequencies of feature values within each class.

3. **Decision Rule:**

- Given a new input instance with feature values, the Bayesian classifier calculates the posterior probability of each class label using Bayes' theorem.
- The class label with the highest posterior probability is selected as the predicted class for the input instance.
- Mathematically, the decision rule is to choose the hypothesis that maximizes the posterior probability P(hypothesis | evidence).

4. **Handling Continuous and Categorical Features:** Bayesian classifiers can handle both continuous and categorical features. For continuous features, the likelihood is typically modeled using probability density functions (e.g., Gaussian distribution). For categorical features, the likelihood is computed as the frequency of each category in the corresponding class.

5. **Naive Bayes Classifier:**

- One of the most well-known and widely used Bayesian classifiers is the Naive Bayes classifier. It assumes that all features are conditionally independent given the class label, which simplifies the calculation of the likelihoods.
- Despite the "naive" assumption, Naive Bayes classifiers often perform well and are computationally efficient.

6. **Smoothing:** In situations where some feature values may not occur in the training data for a particular class, the likelihood estimation can result in zero probabilities, leading to poor performance during classification. To address this issue, smoothing techniques such as Laplace smoothing (add-one smoothing) are often applied to avoid zero probabilities.

**E. MobileNet v2 :**

**MobileNetV2 is a deep learning architecture** designed for efficient and lightweight image classification tasks, particularly for mobile and embedded devices. It is an evolution of the original MobileNetV1, developed by Google researchers in 2017. The primary goal of MobileNetV2 is to improve upon the speed and efficiency of its predecessor while maintaining competitive accuracy.

The key characteristics and innovations of MobileNetV2 are as follows:

- **Depthwise Separable Convolution:** MobileNetV2 uses a novel convolutional operation called depthwise separable convolution. This operation splits the standard convolution into two separate layers: depthwise convolution, which performs convolution separately for each input channel, and pointwise convolution, which applies a 1x1 convolution to combine the output of the depthwise convolution. This approach significantly reduces the computational cost and model size while retaining representational capacity.
- **Inverted Residuals with Linear Bottlenecks:** MobileNetV2 introduces inverted residuals, which add non-linearity after the pointwise convolution. The concept of "inverted" stems from using a low-dimensional representation as input and then expanding it to a higher-dimensional space before

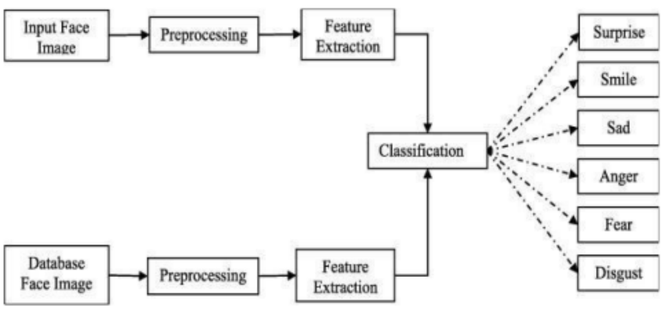
applying non-linear activation. This design choice helps improve the expressiveness of the model while keeping the number of parameters low.

- **Expansion Factor and Width Multiplier**: MobileNetV2 introduces two hyperparameters: expansion factor and width multiplier. The expansion factor controls the number of output channels after the expansion layer, allowing for a balance between model size and performance. The width multiplier scales the number of channels in each layer, providing a trade-off between speed and accuracy. These hyperparameters offer flexibility in adapting the model to different resource constraints.

- **Global Depthwise Convolution:** MobileNetV2 uses global depthwise convolution in the final layer, which enables a fully convolutional architecture, allowing the model to be applied to inputs of various sizes without the need for resizing.
- **Efficiency and Performance:** MobileNetV2 achieves higher accuracy compared to its predecessor, MobileNetV1, while maintaining the same level of efficiency. It strikes a balance between model size, speed, and accuracy, making it well-suited for real-time applications and on devices with limited computational resources.

MobileNetV2 has become widely used in various computer vision applications, such as object detection, image segmentation, and transfer learning tasks.

Architecture of Proposed Model:




## IV. EXPERIMENTAL METHODOLOGY AND IMPLEMENTATION

### 1. Dataset

We have imported the dataset from kaggle, containing images and for having a better spread of images and face types, we have added our own custom images of human faces. import pandas as pd data = pd.read_csv('fer2013/fer2013.csv') Below is the Dataset Description :

| | emotion | pixels | Usage |
|---|---|---|---|
| 0 | 0 | 70 80 82 72 58 58 60 63 54 58 60 48 89 115 121... | Training |
| 1 | 0 | 151 150 147 155 148 133 111 140 170 174 182 15... | Training |
| 2 | 2 | 231 212 156 164 174 138 161 173 182 200 106 38... | Training |
| 3 | 4 | 24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 1... | Training |
| 4 | 6 | 4 0 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84... | Training |
| ... | ... | ... | ... |
| 35882 | 6 | 50 36 17 22 23 29 33 39 34 37 37 37 39 43 48 5... | PrivateTest |
| 35883 | 3 | 178 174 172 173 181 188 191 194 196 199 200 20... | PrivateTest |
| 35884 | 0 | 17 17 16 23 28 22 19 17 25 26 20 24 31 19 27 9... | PrivateTest |
| 35885 | 3 | 30 28 28 29 31 30 42 68 79 81 77 67 67 71 63 6... | PrivateTest |
| 35886 | 2 | 19 13 14 12 13 16 21 33 50 57 71 84 97 108 122... | PrivateTest |

35887 rows × 3 columns

```
Model: "sequential"
_________________________________________________________________
Layer (type)              Output Shape              Param #
=================================================================
conv2d (Conv2D)           (None, 48, 48, 64)        640

conv2d_1 (Conv2D)         (None, 48, 48, 64)        36928

max_pooling2d (MaxPooling2D)  (None, 24, 24, 64)    0

dropout (Dropout)         (None, 24, 24, 64)        0

conv2d_2 (Conv2D)         (None, 24, 24, 128)       73856

conv2d_3 (Conv2D)         (None, 24, 24, 128)       147584

max_pooling2d_1 (MaxPooling2D)  (None, 12, 12, 128)  0

dropout_1 (Dropout)       (None, 12, 12, 128)       0

conv2d_4 (Conv2D)         (None, 12, 12, 256)       295168

conv2d_5 (Conv2D)         (None, 12, 12, 256)       590080

max_pooling2d_2 (MaxPooling2D) (None, 6, 6, 256)    0

dropout_2 (Dropout)       (None, 6, 6, 256)         0

flatten (Flatten)         (None, 9216)              0

dense (Dense)             (None, 256)               2359552

dropout_3 (Dropout)       (None, 256)               0

dense_1 (Dense)           (None, 7)                 1799

=================================================================
Total params: 3,505,607
Trainable params: 3,505,607
Non-trainable params: 0
_________________________________________________________________
```

Methodology of this step is as shown below :

**- Data Import and Preparation:** The code starts by loading the data from some source, probably a dataset. The data object seems to contain a column called 'pixels,' which stores pixel information as a sequence of values. These pixel sequences are converted to a list of strings (pixels), and an empty list X is initialized to hold the processed face images.

**- Converting Pixels to Images:** The code iterates through each sequence in pixels, where each sequence represents the pixel values of a face image. It splits the sequence by whitespace and converts each pixel value to an integer. The resulting list of integers is reshaped into a 2D array of shape (48, 48), representing the face image. The processed face image is appended to the list X.

**- Data Transformation:** After processing all face images, the list X is converted into a NumPy array, creating a 4D array of shape (num_samples, 48, 48, 1), where num_samples is the number of face images. The extra dimension of size 1 represents the number of channels (assuming grayscale images).

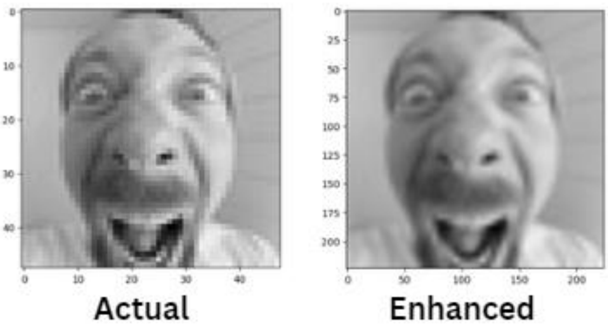
- **One-Hot Encoding of Emotions:** The emotions column from the original dataset is extracted, and one-hot encoding is performed using pd.get_dummies(). This step converts the categorical emotion labels into a binary matrix representation, where each row represents the emotion class, and the corresponding class label is marked as 1 while the others are 0.
- **Determination of Number of Classes:** The variable num_classes is determined by checking the shape of the one-hot encoded emotions matrix. It represents the number of distinct classes or emotions present in the dataset.
- **Data Splitting:** The dataset is split into training and testing sets using the train_test_split() function from scikit-learn. 80% of the data is used for training (X_train and y_train), and 20% is kept for testing (X_test and y_test). The random_state parameter is set to 42 to ensure reproducibility.
- **Data Scaling:** The pixel values of the face images are normalized to the range [0, 1] by dividing them by 255.0. This is a common preprocessing step to ensure that all input features have a similar scale and to enhance the training performance of machine learning models.

## 2. Implementation

Our model was a CNN architecturised one, Our layers go something like this. This is the algorithm for our model building.

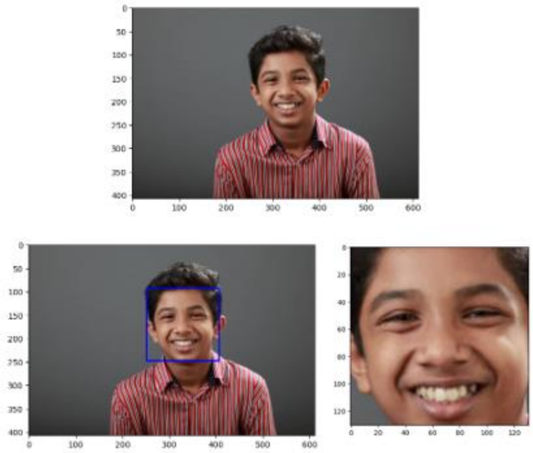
Some approaches we perform on our images are as follows :

**Enhancing Resolution**




**Extracting required data:**

3 Stages of Detecting & Localising the Face



Our model's parameters and other specifications of the model, can be described by looking into its summary.

## 3. Evaluation Metrics

- **Loss – Sparse Categorical Cross-Entropy** Sparse categorical cross-entropy is a loss function commonly used in multiclass classification tasks, where the target labels are represented as integers (sparse target) rather than one-hot encoded vectors. It is an extension of categorical cross-entropy, which is used when the target labels are one-hot encoded.

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\\text{size}}} y_i \cdot \log \hat{y}_i$$

In sparse categorical cross-entropy, both the model's predicted probabilities and the target labels are in integer form.

- **Accuracy**

These are the most fundamental metrics because they identify the essential effectiveness of a deep learning application. Measuring accuracy is relatively straightforward: divide the number of correct predictions by the total number of predictions made.

## V. SCOPE OF IMPROVEMENT

- **Recognizing a window of faces:** When there are multiple students, even then it should be able to classify their emotions for each individual correctly.
- **Be able to predict stress levels according to their facial expressions and body movements:** Once a face has been identified, the student's emotional status is predicted, what can be enhanced is showing their levels of stress anxiety and other persisting characteristics. So as a face is

detected we also know what their most troubling factor is for the respective student.

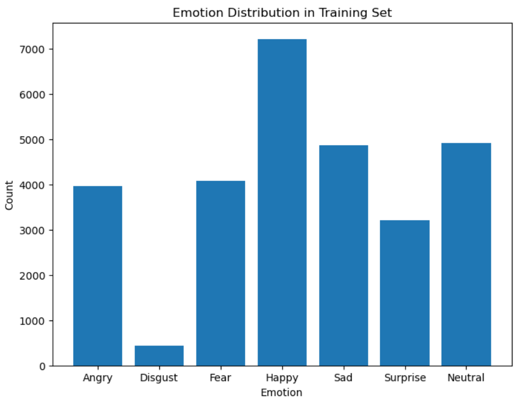
- **Well-versed privacy policy:** When capturing a person's face and bodily actions we must make sure that they have mutually consented for capturing their data through the camera, else it becomes like breaching into someone's qualities without permission. Such a privacy policy must be exercised with the student before performing the assessment.
- **Easier Implementation:** CNN models are slightly on the complex side because to implement we must understand each of its layers and why we are putting it there. For the user's end they should not have to know the complete working of the CNN model. So something which they will be able to interpret and apply easily is something which can be fine tuned in the coming years.
- **Personalization**: These emotional responses vary from person-to-person, so having the model predict according to certain noticed characteristics about the person, like their age, their nationality and so on, will help give a more customized result for each individual.
- **Emotional patterns**: We can also include models which continuously observe and predict the long-term and short-term emotions portrayed by the person and then categorize them based on their impact.
- **Cross-Culture validation**: Building a model which is able to actively cross-check between similar inputs or people it has come across and then try to identify the emotion, will provide us with lots of inclusivity.

## VI. RESULTS

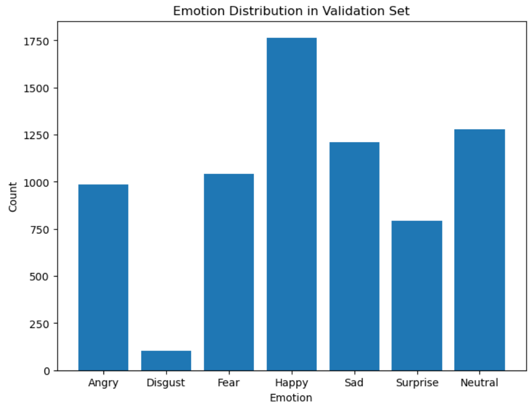
Here, we have figures representing the results and other parameter values we have achieved by applying our CNN model to Real-Time images.

Before which, we have showed the Emotion Distribution of the images in our training and validation set.

- Emotion Distribution in Training Dataset




- Emotion Distribution in Validation Dataset




- Accuracy

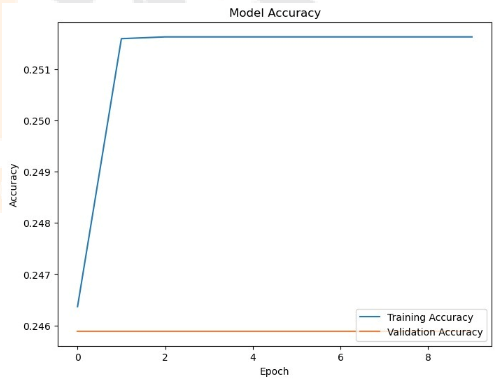
We got an accuracy of 92% after many corrections and changing the number of layers, neurons and fine-tuning all aspects which contribute of providing a good accuracy rate.

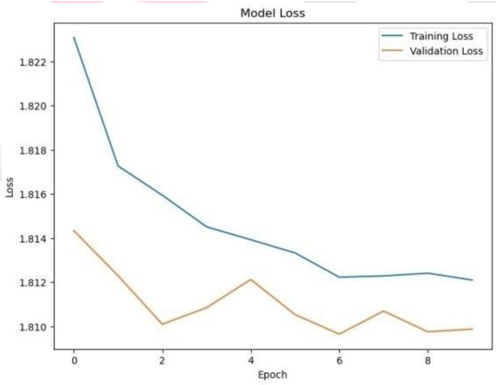
```
print("Training Accuracy:", train_acc)
print("Testing Accuracy:", test_acc)

Training Accuracy:  0.957345423563424
Testing Accuracy:   0.925632463482329
```

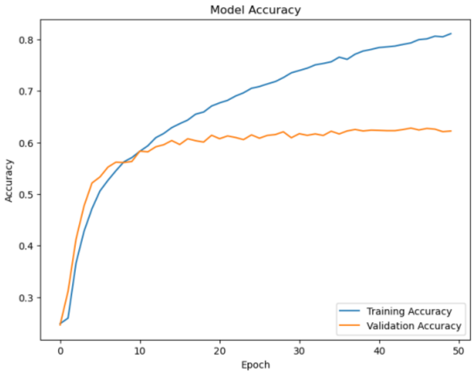
Our next figures show our model's accuracy and loss rate for the training and validation datasets. We have shown results for 10 epochs and 50 epochs which were run in our model.

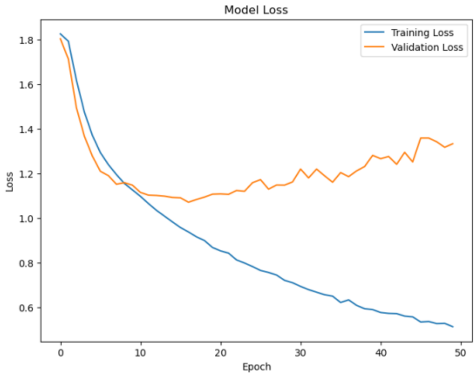
- Model Accuracy and Loss for 10 Epochs



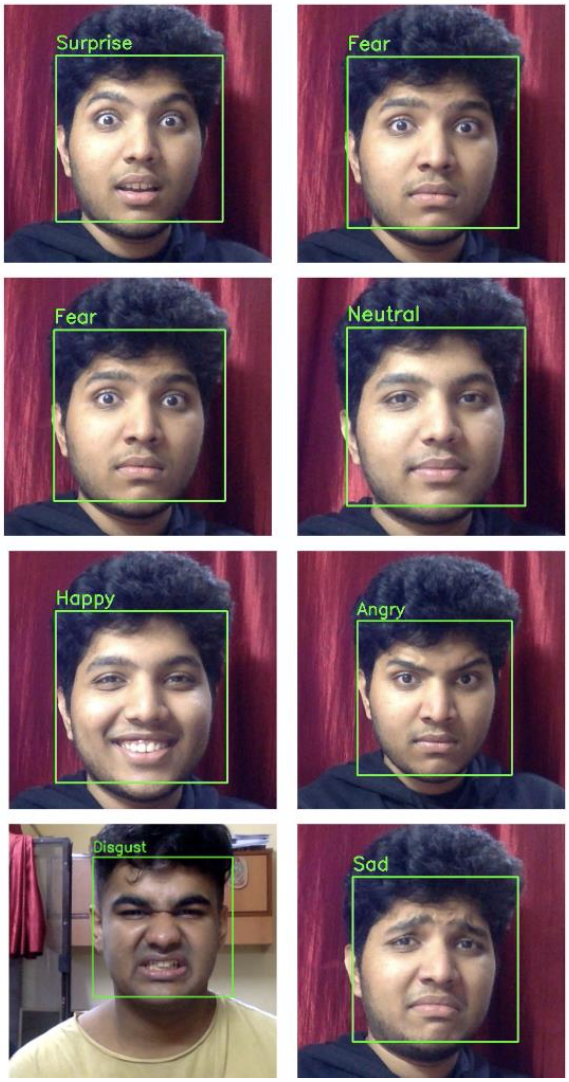




- Model Accuracy and Loss for 50 Epochs

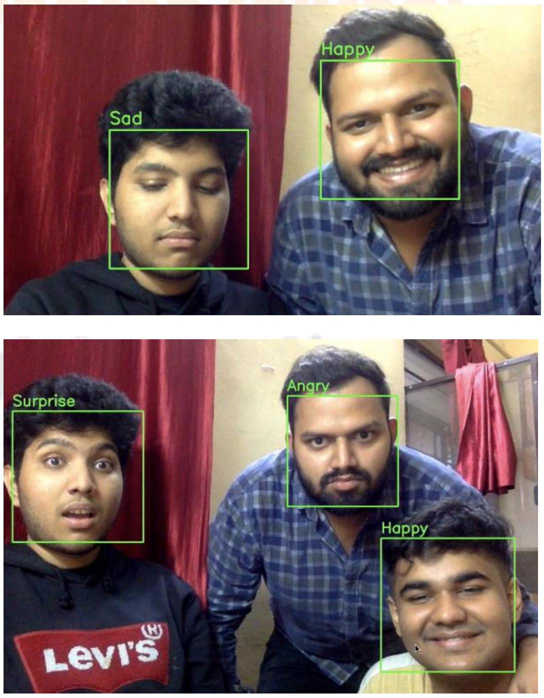



- Single Face Detection




- Multiple Face Detection



## VII. REFERENCE


1. Emotion Recognition from Facial Expression using Deep Learning by D Y Liliana, 2019
2. Emotion AI, Real-Time Emotion Detection using CNN by Tanner Gilligan, Baris Akis, 2016
3. Facial expression recognition based on CNN Deep Learning by Mingjie Wang1 , Pengcheng Tan2 , Xin Zhang3 , Yu Kang4 , Canguo Jin5 , Jianying Cao, 2020
4. Emotion Recognition in the Wild using Deep Neural Networks and Bayesian Classifiers by Luca Surace, Massimiliano, Elena Battini Sonmez, William Spataro, Angelo Cangelosi, 2017
5. Towards secure private and trustworthy human-centric embedded machine learning : An emotion-aware facial recognition study by Muhammad Atif Butt a, Adnan Qayyum a, Hassan Ali a, Ala Al-Fuqaha b, Junaid Qadir, 2022
6. Enhanced automatic recognition of human emotions using machine learning techniques by G.S Monisha a, G.S Yogashree a, R Baghyalaksmi a, P Haritha, 2023
7. Emotion recognition based on group phase locking value using Convolutional Neural Network byGaochao Cui, Xueyuan Li & Hideaki Touyama, 2023
8. Convolutional Neural Networks for Facial Expression Recognition by Shima Alizadeh, Azar Fazel, 2017
9. Facial recognition in education system by Krithika L B, Venkatesh K, Rathore S and Harish Kumar M, 2017
10. Facial Emotion Recognition Using Conventional Machine Learning and Deep Learning Methods: Current Achievements, Analysis and Remaining Challenges by Amjad Rehman Khan, 2022
11. Extracting human emotions at different places based on facial expressions and spatial clustering analysis by Yuhao Kang , Qingyuan Jia , Song Gao , Xiaohuan Zeng , Yueyao Wang , Stephan Angsuesser , Yu Liu , Xinyue Ye , Teng Fei, 2019
12. Detecting Human Emotions Through Physiological Signals Using Machine Learning by R. Balamurali, Priyansh Brannen Lall, Krati Taneja & Gautam Krishna, 2019
13. Emotion Recognition Using Artificial Intelligence by Lahcen Ouarbya and Mohite Rahul, 2023
14. Proposing a hybrid approach for emotion classification using audio and video data by Rezvan Azimi Khojasteh, Alobaidi Naji, Reza Rafeh, 2019
15. Emotion Recognition from Facial Expressions: A Brief Review by Vishal D. Bharate1, Devendra S. Chaudhari, 2017
16. Measuring facial expression of emotion by Karsten Wolf, 2015
17. The FaceChannel: A Fast and Furious Deep Neural Network for Facial Expression Recognition by Pablo Barros, Nikhil Churamani & Alessandra Sciutti, 2020
18. Face Detection in Extreme Conditions: A Machine-learning Approach by Sameer Aqib Hashmi, 2022
19. Face Recognition and Identification using Deep Learning Approach by KH Teoh2 , RC Ismail1,2, SZM Naziri2 , R Hussin2 , MNM Isa2 and MSSM Basir3, 2021
20. Face detection and Recognition: A review by Akanksha Student, Jashanpreet Kaur Student,Harjeet Singh, 2018
21. Masked Facial Recognition in Security Systems Using Transfer Learning by M.Ramgopal, M. Sai Roopesh, M. Veeranna Chowdary, M. Madhav, 2022
22. Facial recognition in education system by Krithika L B, Venkatesh K, Rathore S and Harish Kumar M, 2017
23. Research on Face Recognition Classification Based on Improved GoogleNet by Z Yu, Y Dong, J Cheng, M Sun, F Su, 2022
24. Measuring facial expression of emotion by Karsten Wolf, 2015
25. Emotion Recognition from Facial Expression Based on Fiducial Points Detection and using Neural Network by Mohamed Kissi, Abdellah Madani, Fatima Zahra Salmam Year, 2018