**EMOTION DETECTION FROM LIVE CAMERA USING CNN**

**ABSTRACT**

A Facial expression is the visible manifestation of the affective state, cognitive activity, intention, personality and psychopathology of a person and plays a communicative role in interpersonal relations. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be used in behavioral science and in clinical practice. An automatic Facial Expression Recognition system needs to perform detection and location of faces in a cluttered scene, facial feature extraction, and facial expression classification. Facial expression recognition system is implemented using Convolution Neural Network (CNN).

CNN model of the project is based on Facial expression dataset with some facial expression labels as happy, sad, surprise, anger, and neutral is used in this project.

**1. INTRODUCTION**

Facial expressions are important attributes in human communication that helps us to know the intentions of other people. In common, people are inferred to know the emotional states of others, such as happiness, sadness, disguist, anger, using facial expressions and vocal behavior. According to different number of surveys, actionable components will convey one third of the human communication, and non actionable components convey two thirds of the same. Among various non actionable components, by carrying emotional synonym, facial emotions are one of the main data channels in the one to one communication. Hence, it is quite common that research of facial emotion detection has been winning lot of attention over the past years with applications in wide range and not only permitting to perceptual and cognitive sciences, but also in Machine computing and computer graphics..

**1.1OBJECTIVE:**

To detect the emotions of the faces in live webcam video with high accuracy using convolutional neural networks and OPENCV library of computer vision. To use state-of-the-art machine learning capabilities to furnish efficient working of the model.

**2. FEASIBILITY STUDY**

Preliminary investigation examine project practicability, the chance the system are helpful to the organization. The most objective of the practicability study is to check the Technical, Operational and Economical practicability for adding new modules and debugging previous running system. All system is possible if they're unlimited resources and infinite time. There are unit aspects within the practicability study portion of the preliminary investigation

* Technical Feasibility
* Economical Feasibility
* Social Feasibility

**2.1 Technical Feasibility**

The technical issue typically raised throughout the practicableness stage of the investigation includes the following:

* Does the mandatory technology exist to try to what's suggested?
* Do the planned equipments have the technical capability to carry the info needed to use the new system?
* Will the planned system offer adequate response to inquiries, despite the amount or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, responsibleness, simple access and information security?

Earlier no system existed to cater to the requirements of ‘Secure Infrastructure Implementation System’. this system developed is technically possible. it's an internet primarily based interface for audit work flow at NIC-CSD. therefore it provides a simple access to the users. The database’s purpose is to make, establish and maintain a work flow among numerous entities so as to facilitate all involved users in their numerous capacities or roles. Permission to the users would be granted supported the roles nominative. Therefore, it provides the technical guarantee of accuracy, responsibleness and security. The package and laborious needs for the event of this project aren't several and area unit already out there in-house at NIC or area unit out there as free as open supply.

The work for the project is finished with this instrumentality and existing package technology. Necessary information measure exists for providing a quick feedback to the users no matter the amount of user’s victimization the system.

**2.2 Economical Feasibility**

A system is developed technically which are used if put in should still be an honest investment for the organization. within the economical practicableness, the event price in making the system is evaluated against the last word profit derived from the new systems. money advantages should equal or exceed the prices.

The system is economically possible. It doesn't need any addition hardware or code. Since the interface for this technique is developed mistreatment the prevailing resources and technologies out there at NIC, there's nominal expenditure and economical practicableness sure.

**2.3 Social Feasibility**

Proposed comes square measure useful given that they will be clad into data system. That may meet the organizations in operation needs. Operational feasibleness aspects of the project square measure to be taken as a vital a part of the project implementation. a number of the vital problems raised square measure to check the operational feasibleness of a project includes the following: -

* Is there spare support for the management from the users?
* Will the system be used and work properly if it's being developed and implemented?
* Will there be any resistance from the user that may undermine the potential application benefits?

This system is targeted to be in accordance with the above-named problems. Beforehand, the management problems and user needs are taken into thought. Therefore there's absolute confidence of resistance from the users that may undermine the potential application edges.

The well-planned style would make sure the optimum utilization of the pc resources and would facilitate within the improvement of performance standing.

**3. SYSTEM ANALYSIS**

**3.2 EXISTING SYSTEM:**

This model already existed in an image net classification based emotion detection system and it fails to detect more emotions from the video frame. In facial features were detected based on latent relations. In the another reference the authors proposed a intelligent facial emotion detection system to effectively detect the emotions from the live video feed using Recurrent neural networks and it takes lot of time to train the model and thus reduces the efficiency.

**3.2.1 DISADVANTAGES:**

* In the existing system, emotions in the faces were detected only from the static images (only from photos) and fail to detect facial expressions from live video frames.
* Emotions were detected from images accuracy is less.
* Facial expressions are limited only too few emotions like happiness and neutral.

**3.3 PROPOSED SYSTEM:**

We have developed a convolution neural network based model for classifying human facial emotions from dynamic facial expressions through live video frame in real time. We use transfer learning on the fully connected layers of an existing convolution neural network which was pre-trained for human emotion classification. Finally, a live video stream connected to a face detector system give feeding of images to the neural network. The results facilitate the easiness of implementing convolution neural networks in real time to detect human facial expression. The results demonstrate the feasibility of implementing neural networks in real time to detect human emotion

**3.3.1 ADVANTAGES:**

* The accuracy of the model is high compared with the existing model.
* Our model is very efficient in getting the emotions by training CNN.
* Our model detected facial expressions from live video frames.

**3.4 SYSTEM REQUIREMENTS**

**3.4.1 SOFTWARE REQUIREMENTS:**

* Operating system :Windows (above 7 64-bit),/Linux/ MAC
* Programming Language : Python

**3.4.2 HARDWARE REQUIREMENTS:**

* Processor :Intel core i3 processor
* RAM : 4GB(minimum)
* Hard disc :500GB

**3.5 TECHNOLOGIES:**

* CNN ( convolutional neural network )
* Open cv
* Python
* Tenserflow (TF)
* Keras

**3.5.1 CNN (CONVOLUTIONAL NEURAL NETWORK)**

In neural networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used.

When it is comes to Machine Learning, Artificial Neural Networks perform really well. Artificial Neural Networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use Convolution Neural Network. In this blog, we are going to build basic building block for CNN.

Before dividing into the Convolution Neural Network, let us first revisit some concepts of Neural networks. In a regular Neural Network.

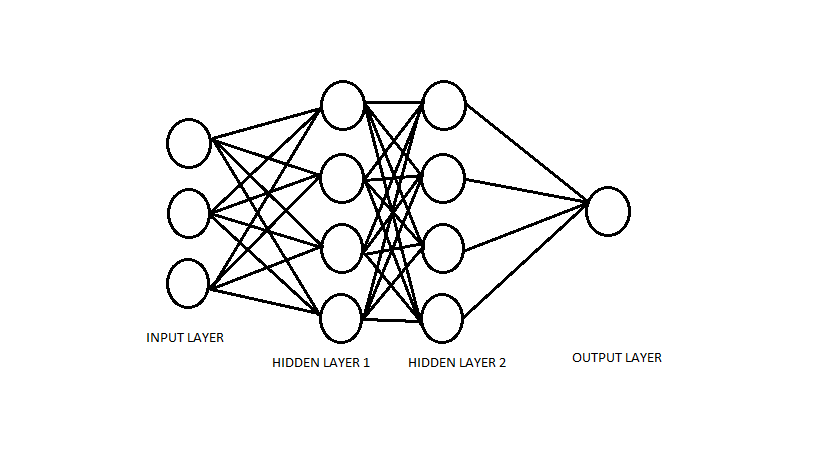
The layers of CNN are:

* **Input Layer:**

It’s the layer in which we given input to our model. The number of Neurons in this layer is equal to total number of features in our data

* **Hidden Layer:**

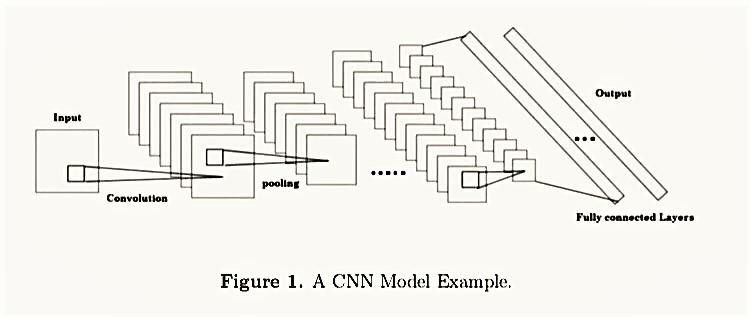
The input from input layer is then feed into hidden layer. there can be many hidden layers depending upon our model and data size. Each hidden layers can have different number of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with the learnable biases followedby activation function which makes the Network nonlinear.



There some of Hidden Layers are:

1. Convolution Layer
2. Activation Function Layer
3. Pool Layer
4. Fully-Connected Layer

* **Convolution Layer:** This layer computes the output volume by computing dot product between all filters and image patch.
* **Activation Function Layer:** This layer will apply element wise activation function to output of convolution layer. Some common activation functions are RELU:max(0,x), Sigmoid: 1/(1+e^\_ x),Tanh,LeakyRELU, etc.The volume remains unchanged hence output volume will have dimensions 32\*32\*12.
* **Pool Layer:** This layer is periodically inserted in the convets and its main function to reduce the size of volume which makes the computation fast reduces memory and also prevents from over fitting. Two common types of pooling layers are max pooling and average pooling.
* **Fully-Connected Layer:** This layer is regular neural network layer which takes input from the previous layer and computes the class scores and outputs the 1-D array to the equal number of classes.



**3.5.3 OPENCV:**

OpenCV (*Open source computer vision*) is a [library of programming functions](https://en.wikipedia.org/wiki/Library_(computing)) mainly aimed at real-time [computer vision](https://en.wikipedia.org/wiki/Computer_vision).Originally developed by [Intel](https://en.wikipedia.org/wiki/Intel_Corporation), it was later supported by [Willow Garage](https://en.wikipedia.org/wiki/Willow_Garage) then Itseez (which was later acquired by Intel). The library is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) and free for use under the [open-source](https://en.wikipedia.org/wiki/Open-source_software) [BSD license](https://en.wikipedia.org/wiki/BSD_license).

OpenCV runs on the following desktop operating systems: [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://en.wikipedia.org/wiki/Linux), [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS), [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD), [NetBSD](https://en.wikipedia.org/wiki/NetBSD" \o "NetBSD), [OpenBSD](https://en.wikipedia.org/wiki/OpenBSD" \o "OpenBSD). OpenCV runs on the following mobile operating systems: [Android](https://en.wikipedia.org/wiki/Android_(operating_system)), [iOS](https://en.wikipedia.org/wiki/IOS" \o "IOS). The user can get official releases from [SourceForge](https://en.wikipedia.org/wiki/SourceForge" \o "SourceForge) or take the latest sources from [GitHub](https://en.wikipedia.org/wiki/GitHub" \o "GitHub).OpenCV uses [CMake](https://en.wikipedia.org/wiki/CMake" \o "CMake)

**3.5.4 PYTHON**

Python is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language" \o "Interpreted language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991, Python has a design philosophy that emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability), notably using [significant whitespace](https://en.wikipedia.org/wiki/Significant_whitespace). It provides constructs that enable clear programming on both small and large scales.[[26]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-7-26) Van Rossum led the language community until stepping down as leader in July 2018.

Python features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type) system and automatic [memory management](https://en.wikipedia.org/wiki/Memory_management). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming) and [procedural](https://en.wikipedia.org/wiki/Procedural_programming). It also has a comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

**3.5.6 TENSORFLOW:**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library](https://en.wikipedia.org/wiki/Library_(computing)) for [dataflow](https://en.wikipedia.org/wiki/Dataflow_programming) and [differentiable](https://en.wikipedia.org/wiki/Differentiable_programming) programming across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google). It is a standard expectation in the industry to have experience in TensorFlow to work in machine learning.

TensorFlow can run on multiple [CPUs](https://en.wikipedia.org/wiki/Central_processing_unit) and [GPUs](https://en.wikipedia.org/wiki/GPU) (with optional [CUDA](https://en.wikipedia.org/wiki/CUDA) and [SYCL](https://en.wikipedia.org/wiki/SYCL) extensions for [general-purpose computing on graphics processing units](https://en.wikipedia.org/wiki/General-purpose_computing_on_graphics_processing_units)). TensorFlow is available on 64-bit [Linux](https://en.wikipedia.org/wiki/Linux), [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS), [Windows](https://en.wikipedia.org/wiki/Windows), and mobile computing platforms including [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) and [iOS](https://en.wikipedia.org/wiki/IOS" \o "IOS).

Tensorflow is an open source machine learning library that allows user to deploy computationto signed processing units with single API.The Tensorflow provides a high-level API for different typesof layers in neural networks, such as convolutional layer, pooling layer and fully connected layer.It also provides methods that adding activation function and applying dropout regularization.

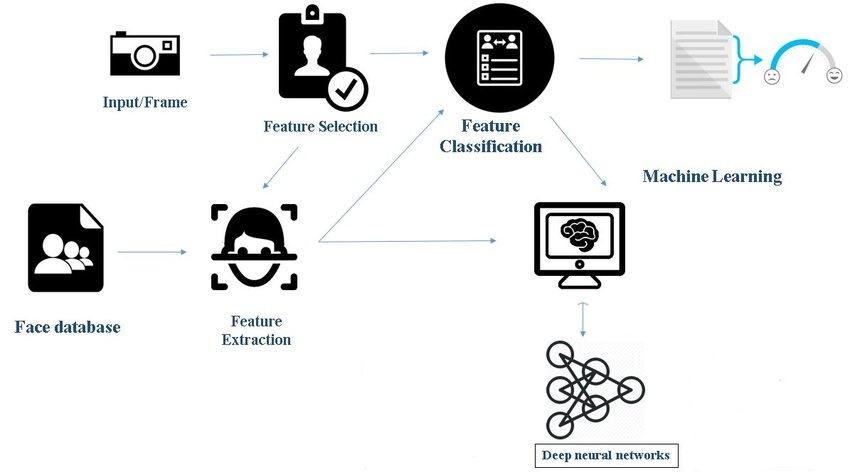
**3.5.7 KERAS:**

Keras is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) [neural-network](https://en.wikipedia.org/wiki/Artificial_neural_network) library written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)). It is capable of running on top of [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow" \o "TensorFlow), [Microsoft Cognitive Toolkit](https://en.wikipedia.org/wiki/Microsoft_Cognitive_Toolkit), [Theano](https://en.wikipedia.org/wiki/Theano_(software)" \o "Theano (software)). Designed to enable fast experimentation with [deep neural networks](https://en.wikipedia.org/wiki/Deep_learning), it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a [Google](https://en.wikipedia.org/wiki/Google) engineer.

Google's TensorFlow team decided to support Keras in TensorFlow's core library. Chollet explained that Keras was conceived to be an interface rather than a standalone machine-learning framework. It offers a higher-level, more intuitive set of abstractions that make it easy to develop deep learning models regardless of the computational backend used.

**4. SYSTEM DESIGN**

**4.1 SYSTEM ARCHITECTURE:**

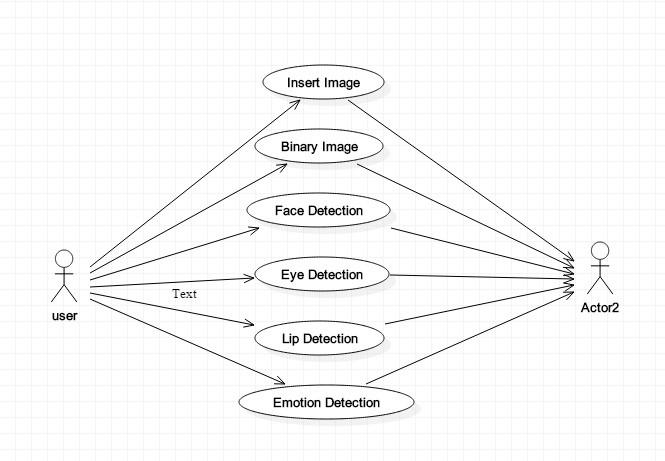


**FIG:1.1 SYSTEM ARCHITECTURE**

**4.3 UML DIAGRAMS**

**4.3.1 USE CASE DIAGRAM**

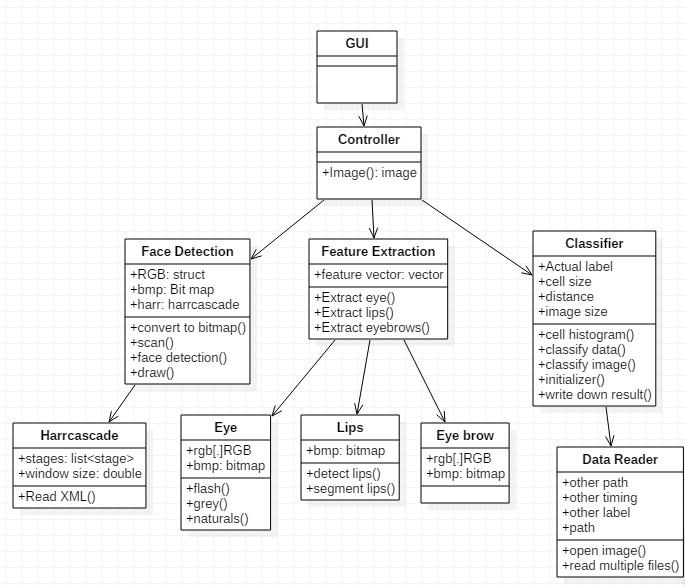
Use Cases are typically used to describe the typically visible interactions that the system will have with u sers and external systems. Typically, the are used to describe how a user would perfor m their role using the system.



**Fig 3.1 use case diagram for emotion detection**

**4.3.2 CLASS DIAG RAM**

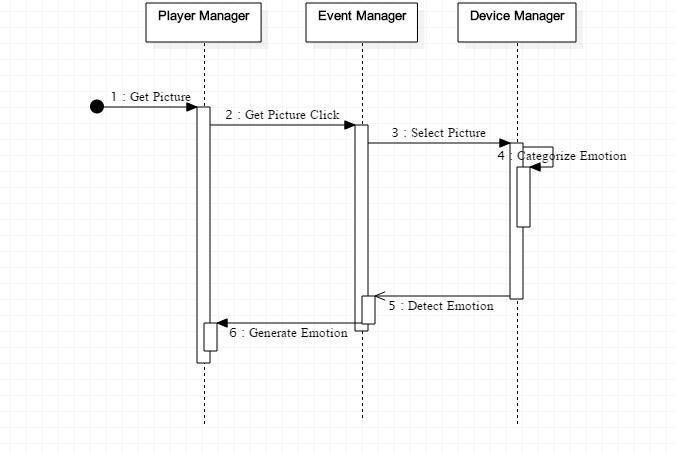
The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, de scribing and documenting different aspects of a system but also for constructing executable code of a software application.



**Fig 3.2 class diagram for emotion detection**

**4.3.3 SEQUENCE DIAGRAM**

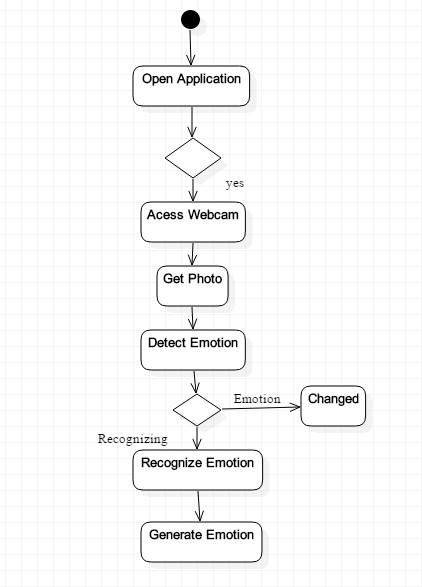
A Sequence diagram i s a kind of interaction diagram that s hows how processes operate with one anothe r and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and c lasses involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functional ities of the scenario.



**Fig 3.3 sequence diagram for emotion detection**

**4.3.4 ACTIVITY DIAGRAM**

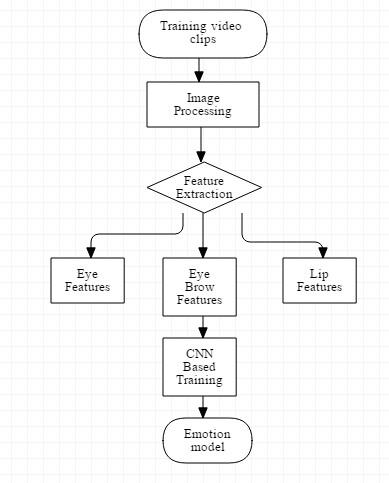
Activity diagram is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another.



**Fig 3.4 Activity diagram for emotion detection**

**4.3.5 DATA FLOW DIAGRAM**

The Data Flow diagram which represents the data of giving input frames with trained video clips . the extraction of various features based on CNN based algorithms to which recognizes the emotion og the given input frame operation.

****

**Fig 3.5 Data Flow diagram for emotion detection**

**5. IMPLEMENTATION**

**5.1 LIST OF MODULES:**

* Dataset
* Models
* Python control program
* Trainer
* Classifier

**5.1.1 DATASET:** We have used a face dataset named fer2013, which comprises of a pre-trained model trained with thousands of images available in open source. Those images were used to train our model which in later stages can be used to detect the emotion.

**5.1.2 MODELS:**We have used a convolutional neural network model, to detect and extract emotions from the face in live streaming web cam. CNN is mostly preferred in dealing with images and feature extraction.

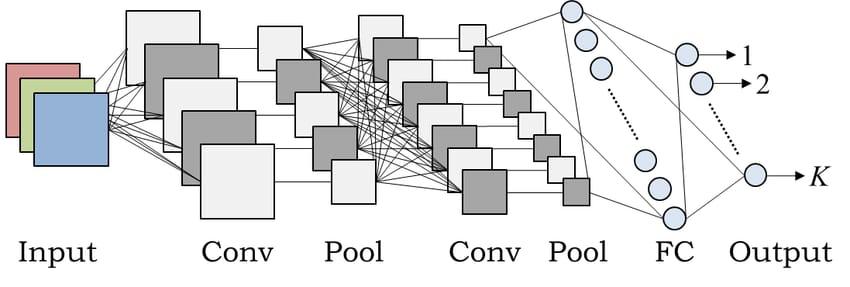
**5.1.3 PYTHON PROGRAM:**  We have used python programming to run our model, as python is very efficient in implementing machine learning and deep learning models.

**5.2 METHODOLOGY AND WORKING:**

Our model has the following sequence of methods to be followed:

1. A dataset is collected, which in our project is fer2013.

2. We trained our model using convolutional neural networks, which is efficient in dealing with Image processing tasks. The architecture of CNN is shown below.



3. After training with the dataset, some pre-trained models will be stored which will be used later in describing the facial emotions from live video frame.

4. By using the web camera, the system detects the faces initially using haarcascade frontal face classifier and later recognizes the emotion using pre-trained model of CNN.

**6. SYSTEM TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.

The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**6.1 Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produces valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**6.2 Functional Testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

**6.3 System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**6.4 Performance Testing**

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**6.5 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**6.6 Acceptance Testing**

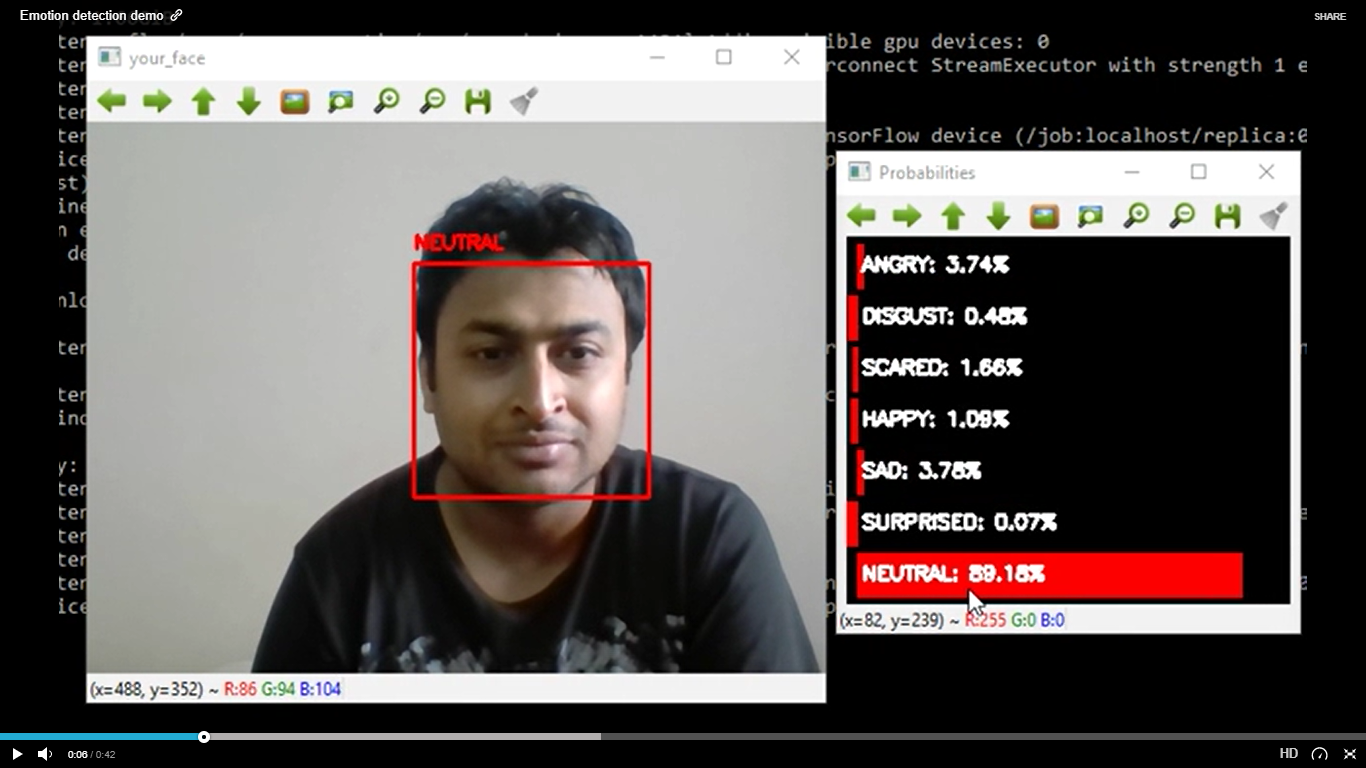
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Acceptance testing for Data Synchronization:**

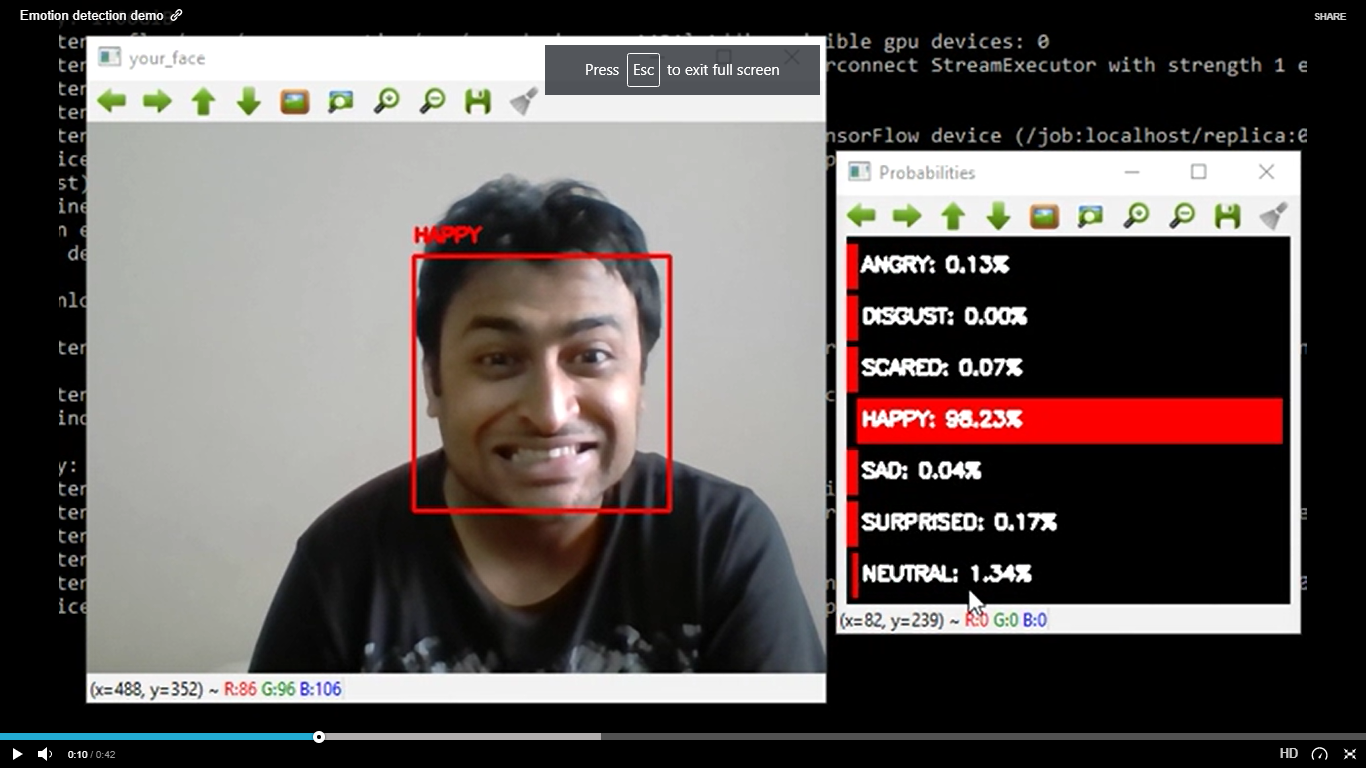
* The Acknowledgements will be received by the auditor after the data is received by the cloud server
* The auditors audit operation is done only when there is a request from user
* The Status of data information on the cloud is viewed only by the cloud server

**RESULT AND OUTPUT**

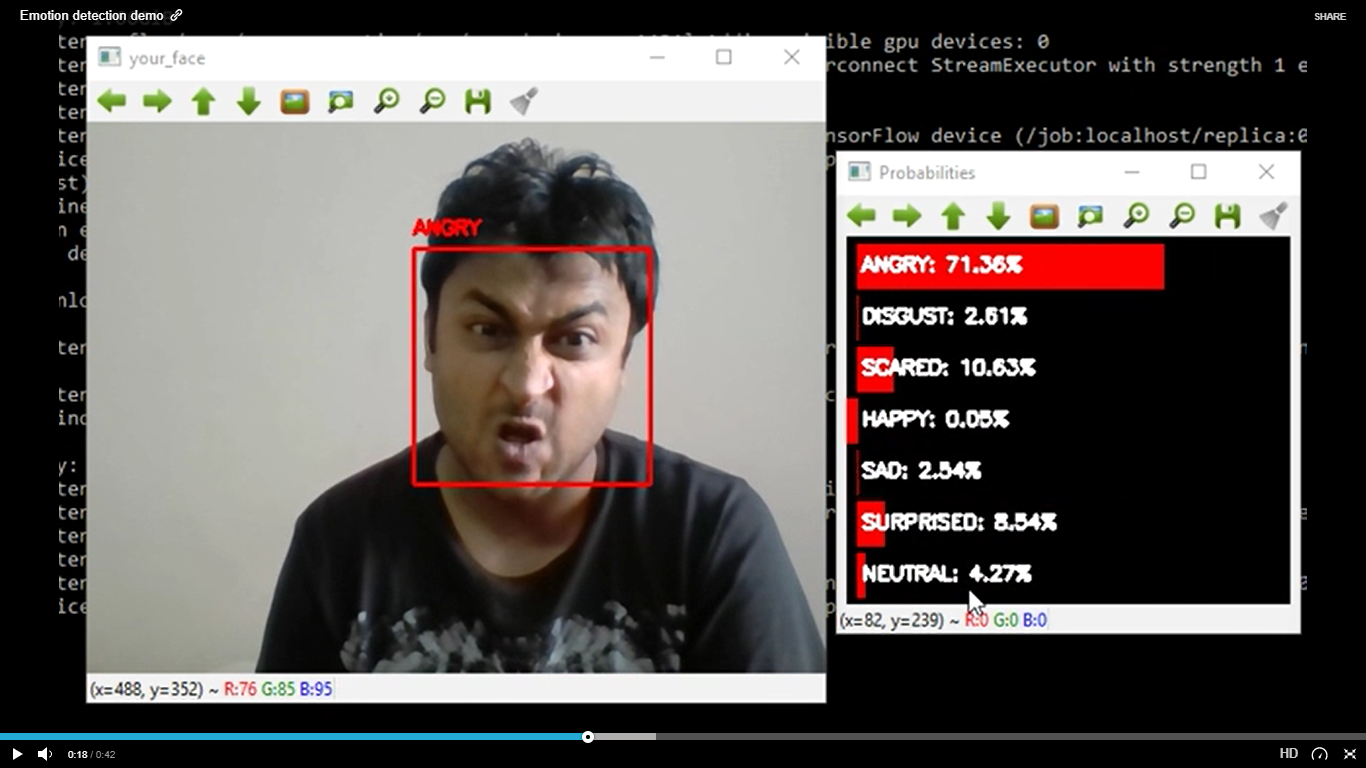
In this model the result can be said that the output will be shown in screenshots. This model which can detect multiple faces simultaneously .this will be accuracy that shows the emotion exactly of high accuracy. This can detect various emotions at a time by detecting multiple frames with their emotion recognition**.**

****

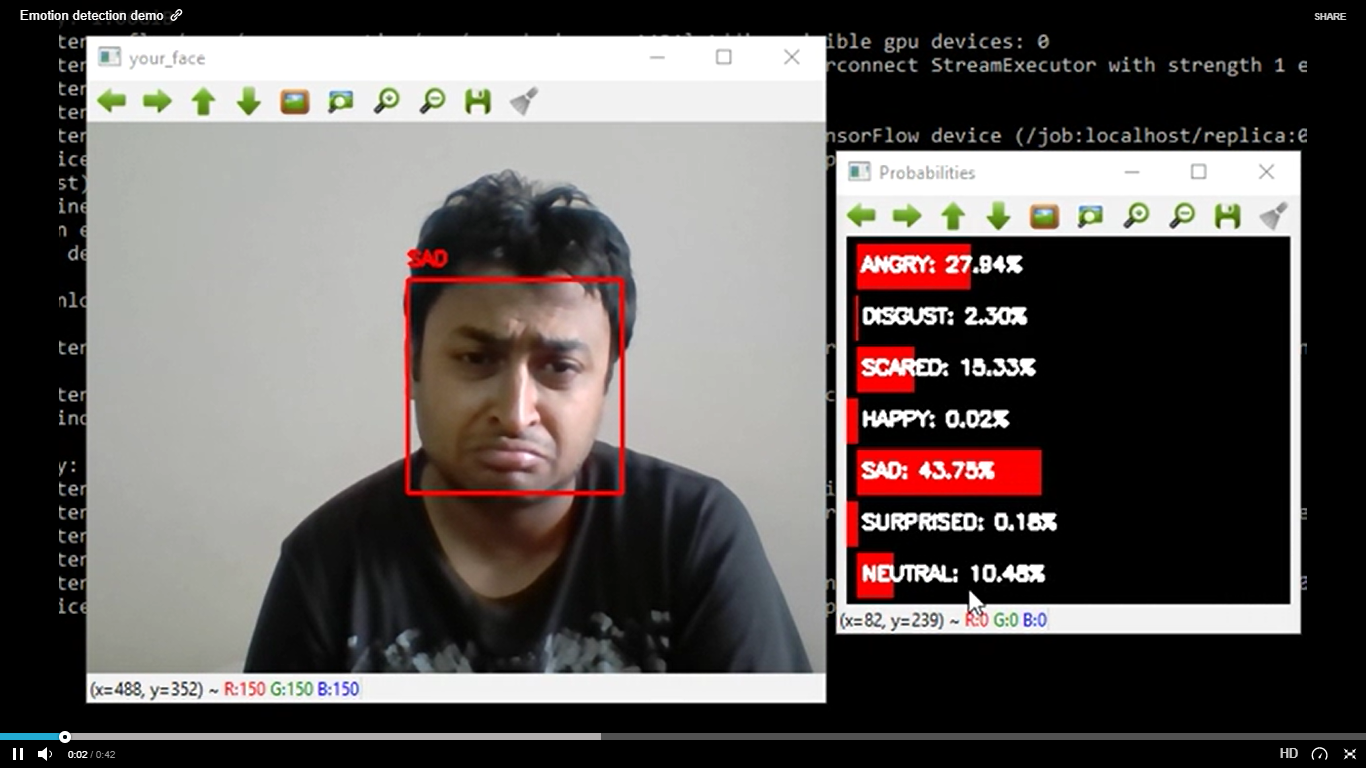
**FIG: 4.2 NEUTRAL**

****

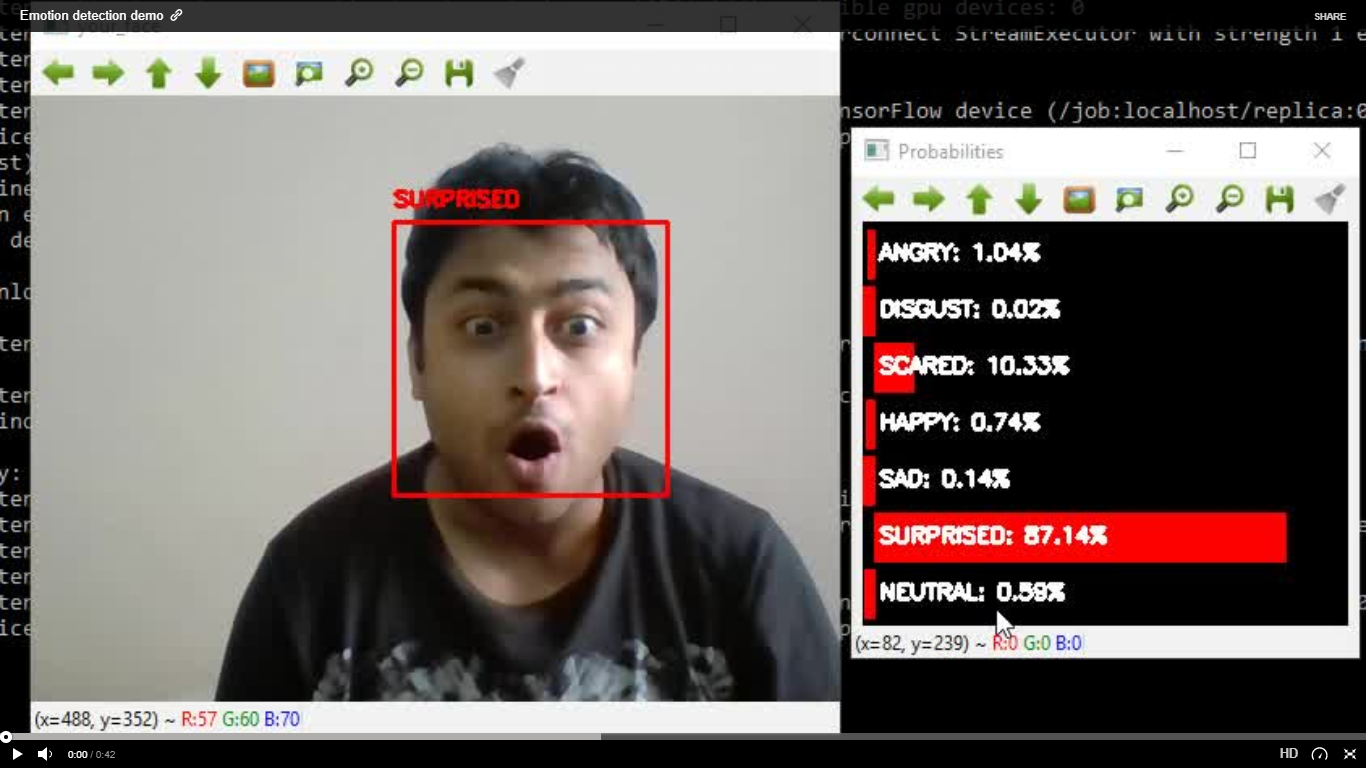
**FIG:4.3 HAPPY**



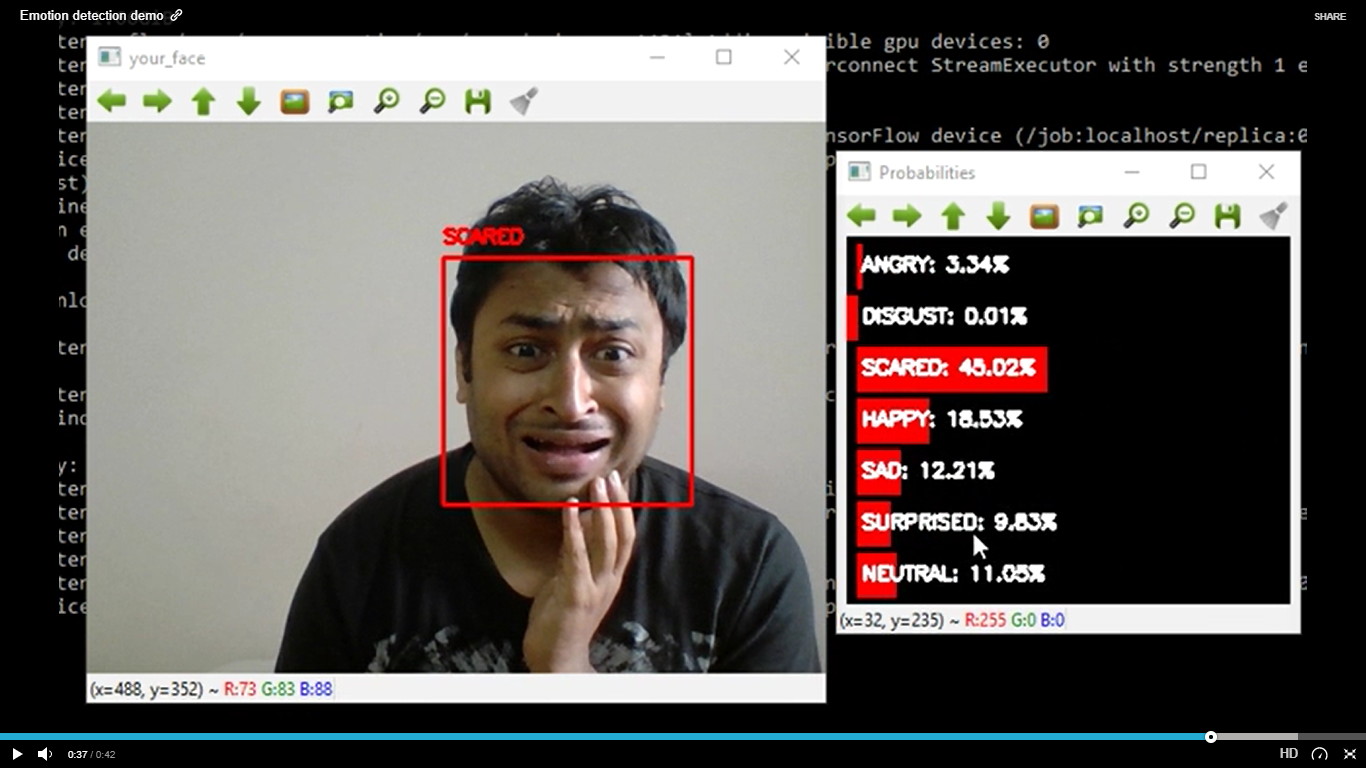
**FIG: 4.4 ANGRY**



**FIG: 4.5 SAD**



**FIG: 4.6 SURPRISE**

****

**FIG: 4.7 SCARED**

**CONCLUSION**

Emotion detection is never ending prolonged research as it has no perfect ending with the accuracy. We have tried a perfect solution to detect Even though, it is not 100% accurate, but it makes the most out of any other existing models. Our model can be used in various applications like humanoid robots, military, etc.

**FUTURE ENHANCEMENTS:**

Our future enhancement is to make our model more user friendly by adding emoticons or emoji's to the detected emotions. This can be done by making recurrent neural networks trained with different emoji's which matches with the detected facial expression. If suppose, the detected emotion is happy then a emoji with happy face is attached to the person's face.

**BIBLIOGRAPHY**

1. P. Abhang, S. Rao, B. W. Gawali, and P. Rokade, “Article: Emotion recognition using speech and eeg signal a review,” International Journal of Computer Applications, vol. 15, pp. 37–40, February 2011. Full text available.
2. P. Ekman, Universals and cultural differences in facial expressions of emotion. Nebraska, USA: Lincoln University of Nebraska Press, 1971.
3. P. Ekman and W. V. Friesen, “Universals and cultural differences in the judgements of facial expressions of emotion,” Journal of Personality and Social Psychology, vol. 53, 1987.
4. A. Kołakowska, A. Landowska, M. Szwoch, W. Szwoch, and M. R. Wrobel,´ Human-Computer Systems Interaction: Backgrounds and Applications 3, ch. Emotion Recognition and Its Applications, pp. 51–62. Cham: Springer International Publishing, 2014.
5. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” NIPS, vol. 1, p. 4, 2012.
6. A. Yao, J. Shao, N. Ma, and Y. Chen, “Capturing au-aware facial features and their latent relations for emotion recognition in the wild,” in Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, ICMI ’15, (New York, NY, USA), pp. 451–458, ACM, 2015.
7. C. Shan, S. Gong, and P. W. McOwan, “Facial expression recognition based on local binary patterns: A comprehensive study,” Image and Vision Computing, vol. 27, no. 6, pp. 803 – 816, 2009.
8. S. Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal, “Recurrent neural networks for emotion recognition in video,” ICMI, pp. 467–474, 2015.
9. Z. Yu and C. Zhang, “Image based static facial expression recognition with multiple deep network learning,” in Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, ICMI ’15, (New York, NY, USA), pp. 435–442, ACM, 2015.
10. B. Kim, J. Roh, S. Dong, and S. Lee, “Hierarchical committee of deep convolutional neural networks for robust facial expression recognition,” Journal on Multimodal User Interfaces, pp. 1–17, 2016.
11. G. Levi and T. Hassner, “Emotion recognition in the wild via convolutional neural networks and mapped binary patterns,” in Proc. ACM International Conference on Multimodal Interaction (ICMI), November 2015.
12. K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” in British Machine Vision Conference, 2014.
13. D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Learning face representation from scratch,” CoRR, vol. abs/1411.7923, 2014.
14. A. Dhall, R. Goecke, S. Lucey, and T. Gedeon, “Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark,” in Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, pp. 2106–2112, Nov 2011.
15. S. Ouellet, “Real-time emotion recognition for gaming using deep convolutional network features,”
16. P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression,” in Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pp. 94–101, June 2010.
17. T. Kanade, J. F. Cohn, and Y. Tian, “Comprehensive database for facial expression analysis,” in Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, pp. 46–53, 2000.
18. M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, “Coding facial expressions with gabor wavelets,” in Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on, pp. 200–205, Apr 1998.