**1.INTRODUCTION**

Stress and anxiety are common everyday life states of emotional strain that affects quality of life. These states consist of several complementary and interacting components (i.e., cognitive, affective, central and peripheral physiological) comprising the organism’s response to changing internal and/or external conditions and demands.Prolonged stress and anxiety can be associated with psychological and/or somatic diseases. Increased skeletal, smooth and cardiac muscle tension, disturbances are typical signs of stress and anxiety, which are linked to some of their most common symptoms and disorders (Headache, Hypertension, etc).

. Historically, the potential negative impact of stress/anxiety on body physiology and health has long been recognized. Although stress and, particularly, anxiety are subjective, multifaceted phenomena, which are difficult to measure comprehensively through objective means, more recent research is beginning to throw light on the factors that determine the degree and type of stress/anxiety-related impact on personal health, including characteristics of the stressor itself, and various biological and psychological vulnerabilities.

However there is evidence that they bear both direct and long term consequences on the person’s capacity to adapt to life events and function adequately, as well as to overall wellbeing .Stress is often described as a complex psychological, physiological and behavioral state triggered upon perceiving a signiﬁcant imbalance between the demands placed upon the person and their perceived capacity to meet those demands. Psychological Stress is becoming a Threat to People’s Health Nowadays. From an evolutionary standpoint, it is an adaptive process characterized by increased physiological and psychological arousal. The two key modes of the stress response (“ﬁght” and “ﬂight”) were presumably evolved to enhance the survival capacity of the organism.

Nevertheless, prolonged stress can be associated with psychological and/or somatic disease. Anxiety is the unpleasant mood characterized by thoughts of worry and fear, sometimes in the absence of real threat. When anxiety is experienced frequently and at intensity levels that appear disproportional to the actual threat level, it can evolve to a broad range of disorders .It should be noted that the terms stress and anxiety are often used interchangeably. Their main difference is that anxiety is usually a feeling not directly and apparently linked to external cursor objective threats. On the other hand, stress is an immediate response to daily demands and is considered to be more adaptive than anxiety. Nevertheless, anxiety and stress typically involve similar physical sensations, such as higher heart rate, sweaty palm and churning stomach, triggered by largely overlapping neuronal circuits when the brain fails to distinguish the difference between a perceived and a real threat. These similarities extend to the facial expressions associated with each state and, accordingly, they were considered as a single state.

Stress and anxiety heavily affect the human wellbeing and health. Under chronic stress, the human body and mind suffers by constantly mobilizing all of its resources for defense. Such a stress response can also be caused by anxiety. Moreover, excessive worrying and high anxiety can lead to depression and even suicidal thoughts. With the rapid pace of life, more and more people are feeling stressed. Though stress itself is non-clinical and common in our life, excessive and chronic stress can be rather harmful to people’s physical and mental health. According to existing research works, long-term stress has been found to be related to many diseases, e.g., clinical depressions, insomnia etc. Moreover, according to Chinese Center for Disease Control and Prevention, suicide has become the top cause of death among Chinese youth, and excessive stress is considered to be a major factor of suicide. All these reveal that the rapid increase of stress has become a great challenge to human health and life quality.

Thus, there is significant importance to detect stress before it turns into severe problems. Traditional psychological stress detection is mainly based on face-to face interviews, self-report questionnaires or wearable sensors. However, traditional methods are actually reactive, which are usually labor-consuming, time-costing and hysteretic. Are there any timely and proactive methods for stress detection?

Stress detection of humans has been a well researched topic in the area of speech signal processing; while very little attention has been paid to recognizing stress from faces. Recognizing stress from faces could complement speech-based techniques and also help in understanding recognition of emotions. The challenges in this domain are that the data from each person are continuous and dynamic and each person expresses stress differently. Therefore the recognition of stress and the associated development of the necessary software is a DDDAS.

Here we provide a medium which scans the face of a person’s and identifies what kind of a mood he/she is and indicate when a person is stressed out.

**1**.**1 Feature Extraction for Emotion Recognition**

1.4Feature Extraction for Emotion Recognition

The feature vector extraction method is most important key point in emotion recognition problem. Especially, it is necessary to get good feature vector to make better recognition

accuracy. In the facial feature extraction stage, we propose a new feature vector extraction method. The proposed method divide whole image into three feature region: eye region,

mouth region, and auxiliary region. Several information are extracted from each region:



Fig 1.1 Mouth template for comparision

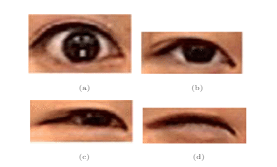


Fig 1.2 Eye template for comparision

**1.2 PROBLEM STATEMENT**

In this contemporary world where the primary focus is to get the job done on time pushes human to their limit which encumbers them with workload that not only effects their health but also their throughput to produce results.

**1.3 OBJECTIVE**

For the well being of a person as well as to be productive it becomes necessary to reduce the stress. Through our project we aim to identify if a person is stressed out by facial detection, through this we can take necessary steps in order to rejuvenate him/her.

**2. LITERATURE SURVEY**

# 2.1 Anxiety and Sensitivity to Gaze Direction in Emotionally Expressive Faces

  Study investigated the role of neutral, happy, fearful, and angry facial expressions in enhancing orienting to the direction of eye gaze. Photographs of faces with either direct or averted gaze were presented. A target letter (T or L) appeared unpredictably to the left or the right of the face, either 300 ms or 700 ms after gaze direction changed. Response times were faster in congruent conditions (i.e., when the eyes gazed toward the target) relative to incongruent conditions (when the eyes gazed away from the target letter). Facial expression did influence reaction times, but these effects were qualified by individual differences in self-reported anxiety. High trait-anxious participants showed an enhanced orienting to the eye gaze of faces with fearful expressions relative to all other expressions. In contrast, when the eyes stared straight ahead, trait anxiety was associated with slower responding when the facial expressions depicted anger. Thus, in anxiety-prone people attention is more likely to be held by an expression of anger, whereas attention is guided more potently by fearful facial expressions.

**Advantages:**

* The findings of this experiment are straightforward
* The current results highlight the importance of taking individual differences into account when examining fundamental aspects of social perception at both cognitive and neural levels.

**Disadvantages:**

* Enhanced orienting response was specific to fearful expressions and that the gaze cuing effect was not enhanced for angry or happy relative to neutral expressions.
* This pattern only emerged in a post hoc analysis.
* Research is required to clarify whether averted fearful expressions enhance attention orienting effects relative to angry facial expressions in anxious individuals.

# 2.2 The MPI facial expression database--a validated database of emotional and conversational facial expressions.

The ability to communicate is one of the core aspects of human life. For this, we use not only verbal but also nonverbal signals of remarkable complexity. Among the latter, facial expressions belong to the most important information channels. Despite the large variety of facial expressions we use in daily life, research on facial expressions has so far mostly focused on the emotional aspect. Consequently, most databases of facial expressions available to the research community also include only emotional expressions, neglecting the largely unexplored aspect of conversational expressions. To fill this gap, we present the MPI facial expression database, which contains a large variety of natural emotional and conversational expressions. The database contains 55 different facial expressions performed by 19 German participants.

Expressions were elicited with the help of a method-acting protocol, which guarantees both well-defined and natural facial expressions. The method-acting protocol was based on every-day scenarios, which are used to define the necessary context information for each expression. All facial expressions are available in three repetitions, in two intensities, as well as from three different camera angles. A detailed frame annotation is provided, from which a dynamic and a static version of the database have been created.

In addition to describing the database in detail, we also present the results of an experiment with two conditions that serve to validate the context scenarios as well as the naturalness and recognizability of the video sequences. Our results provide clear evidence that conversational expressions can be recognized surprisingly well from visual information alone. The MPI facial expression database will enable researchers from different research fields (including the perceptual and cognitive sciences, but also affective computing, as well as computer vision) to investigate the processing of a wider range of natural facial expressions.

**Disadvantages:**

* The expressions, although exhibiting a large degree of individual variation, were overall rated as being very natural.
* An additional problem is, still lacking a proper facial expression “vocabulary” in order to uniquely label the expressions.

# 2.3 Facial Emotion Recognition System through Machine learning approach

Data mining also sometimes called data or knowledge discovery is the process of analyzing data from different perspectives and summarizing it into useful information. Image processing is related to Computer vision, which is a high-level image processing out of which a machine/computer/software intends to decipher the physical contents of an image or a sequence of images. One of the ways to do this is by comparing selected facial features from the image and a facial database. Recognizing emotion from images has become one of the active research themes in image processing and in applications based on human-computer interaction.

This research conducts an experimental study on recognizing facial emotions. The flow of our emotion recognition system includes the basic process in FER system. These include image acquisition, preprocessing of an image, face detection, feature extraction, classification and then when the emotions are classified the system assigns the user particular music according to his emotion. Our system focuses on live images taken from the webcam. The aim of this research is to develop automatic facial emotion recognition system for stressed individuals thus assigning them music therapy so as to relief stress. The emotions considered for the experiments include happiness, Sadness, Surprise, Fear, Disgust, and Anger that are universally accepted.

# 2.4 Covariance Pooling for Facial Expression Recognition

Classifying facial expressions into different categories requires capturing regional distortions of facial landmarks. We believe that second-order statistics such as covariance is better able to capture such distortions in regional facial features. In this work, we explore the benefits of using a manifold network structure for covariance pooling to improve facial expression recognition. In particular, we first employ such kind of manifold networks in conjunction with traditional Convolutional networks for spatial pooling within individual image feature maps in an end-to-end deep learning manner.

By doing so, we are able to achieve a recognition accuracy of 58.14% on the validation set of Static Facial Expressions in the Wild (SFEW 2.0) and 87.0% on the validation set of Real-World Affective Faces (RAF) Database1. Both of these results are the best results we are aware of. Besides, we leverage covariance pooling to capture the temporal evolution of per-frame features for video-based facial expression recognition.

Our reported results demonstrate the advantage of pooling image-set features temporally by stacking the designed manifold network of covariance pooling on top of Convolutional network layers.

**2.5.** **Facial emotion detection using deep learning**

The use of machines to perform different tasks is constantly increasing in society. Providing machines with perception can lead them to perform a great variety of tasks; even very complex ones such as elderly care. Machine perception requires that machines understand about their environment and interlocutor’s intention.

Recognizing facial emotions might help in this regard. During the development of this work, deep learning techniques have been used over images displaying the following facial emotions: happiness, sadness, anger, surprise, disgust, and fear. In this research, a pure Convolutional neural network approach outperformed other statistical methods' results achieved by other authors that include feature engineering.

Utilizing Convolutional networks involves feature learning; which sounds very promising for this task where defining features is not trivial. Moreover, the network was evaluated using two different corpora: one was employed during network's training and it was also helpful for parameter tuning and for network's architecture definition. This corpus consisted of facial acted emotions. The network providing best classification accuracy results was tested against the second dataset. Even though the network was trained using only one corpus; the network reported auspicious results when tested on a different dataset, which displayed facial non-acted emotions.

While the results achieved were not state-of-the-art; the evidence gathered points out deep learning might be suitable to classify facial emotion expressions. Thus, deep learning has the potential to improve human-machine interaction because its ability to learn features will allow machines to develop perception. And by having perception, machines will potentially provide smoother responses, drastically improving the user experience.

# 2.6. Training deep networks for facial expression recognition with crowd-sourced label distribution

Crowd sourcing has become a widely adopted scheme to collect ground truth labels. However, it is a well-known problem that these labels can be very noisy. In this paper, we demonstrate how to learn a deep Convolutional neural network (DCNN) from noisy labels, using facial expression recognition as an example. More specifically, we have 10 taggers to label each input image, and compare four different approaches to utilizing the multiple labels: majority voting, multi-label learning, probabilistic label drawing, and cross-entropy loss. We show that the traditional majority voting scheme does not perform as well as the last two approaches that fully leverage the label distribution. An enhanced FER+ data set with multiple labels for each face image will also be shared with the research community.

**2.7.** **A Comprehensive Survey on Techniques for Facial Emotion Recognition**

The emotions, set in simple words are what people feel. Emotional aspects have huge impact on Social intelligence like communication understanding, decision making and helps in understanding behavioral aspect of human. Human faces provide various information about emotions. As per psychological researcher, a person expresses his emotions less by verbal talk and more by non-verbal body posture and gestures. Emotion recognition or Affective Computing (AC) being the AI related area imparts intelligence to computers in recognizing human emotions.

Emotion recognition is proved a popular research area topic in few decades. The aim of this paper is to report an illustrative and comprehensive study of most popular emotion recognition methods, which are generally used in emotion recognition problems. We are motivated by the lack of detailed study of all possible techniques implementations in available literature. This paper provides an up-to-date comprehensive survey of techniques available for emotion recognition**.**

**2.8. A Brief Review of Facial Emotion Recognition Based on Visual Information**

Facial emotion recognition (FER) is an important topic in the fields of computer vision and artificial intelligence owing to its significant academic and commercial potential. Although FER can be conducted using multiple sensors, this review focuses on studies that exclusively use facial images, because visual expressions are one of the main information channels in interpersonal communication. This paper provides a brief review of researches in the field of FER conducted over the past decades.

First, conventional FER approaches are described along with a summary of the representative categories of FER systems and their main algorithms. Deep-learning-based FER approaches using deep networks enabling “end-to-end” learning are then presented. This review also focuses on an up-to-date hybrid deep-learning approach combining a Convolutional neural network (CNN) for the spatial features of an individual frame and long short-term memory (LSTM) for temporal features of consecutive frames.

In the later part of this paper, a brief review of publicly available evaluation metrics is given, and a comparison with benchmark results, which are a standard for a quantitative comparison of FER researches, is described. This review can serve as a brief guide book to newcomers in the field of FER, providing basic knowledge and a general understanding of the latest state-of-the-art studies, as well as to experienced researchers looking for productive directions for future work.

**3. SYSTEM ANALYSIS**

**3.1. EXISTING SYSTEM**

* A existing system studied the emotion propagation problem, and found that anger has a stronger correlation among different users than joy, indicating that negative emotions could spread more quickly and broadly in the network. As stress is mostly considered as a negative emotion, this conclusion can help us.
* Involves through interaction with psychotherapist where a series of questions are asked to determine their mood.
* ECGS are taken to determine increase of heart rate which indicates high level of stress.
* Industries use 6 stages to identify and release stress: Hazard detection, Assessment of harm, Identification of likely risk factor, Employee support system and residual of risk.

**3.2 DISADVANTAGES OF EXISTING SYSTEM**

* Though the system is designed to yield correct results there is always a possibility of faulty detection.
* Traditional psychological stress detection is mainly based on face-to face interviews, self-report questionnaires or wearable sensors. However, traditional methods are actually reactive, which are usually labor-consuming, time-costing and hysteretic.
* Training takes lots of time for its completion.
* the images have a low resolution
* the faces are not in the same position
* some images have text written on them
* some people hide part of their faces with their hands
  1. **PROPOSED SYSTEM**
* Proposed system provides an innovative and faster way to determine if a person is in stress through their expressions.
* The program uses the dataset to train itself about various human emotions and uses the results of it to identify mood of a person.
* Using the combination of preceding points results an approximate result is generated which depicts if a person is stressed.

**3.4. ADVANTAGES OF PROPOSED SYSTEM**

Variety of semi voluntary facial features is jointly used for the detection of anxiety.

Reduces the time to identify if a person is in anxiety which creates opportunities to reduce anxiety before it takes a toll on body.

Not every person is open to talk about stress openly, companies can use this to identify how its employees are handling work load and can provide refreshments and activities to reduce this load.

Each recordings and prediction made by the system can be used to retrain the model in-order to improve the accuracy ultimately yielding much accurate detections of stress or anxiety.

**3.5 FEASIBILITY STUDY**

**3.5.1 TECHNICAL FEASIBILITY**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested? - The OpenCV library of Python provides the necessary LBPH Face Recognition module and tkinter library for user interface. The report generation is made possible using Excel.
* The technical capacity of the data?-The training data is a total of 14MB for 8 people requiring 4-12 training pictures per person (the more the better). The training image size can be 60 KB-400 KB per image.
* Will the proposed system provide adequate response to inquiries, regardless of the number? - The system will not only mark the attendance automatically but will also generate reports as requested by the user. For e.g.: weekly and monthly reports, per student attendance report to determine defaulters.
* Can the system be upgraded if developed?-The project has been implemented using basic Python OpenCV but can be improved using Tensor Flow, a deep-learning framework supported by OpenCV.
* Are there technical guarantees of accuracy, ease of access? - The accuracy may decrease as the number of students increases (our system being implemented for 20 students approximately). Access is easy due to understandable and simple user interface.

**3.5.2 OPERATIONAL FEASIBILITY**

* User-friendly**-**The GUI implemented using tkinter library in Python makes using the application very much user-friendly.
* Portability**-**The application will be developed using OpenCV (Open source computer vision) of Python which will work on Windows and Linux o/s. Hence portability problems will not arise.
* Availability**-** This software will be available always.
* Maintainability -The system uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database. The front-end can be run on different systems (clients). Users access these forms by using the user-ids and the passwords.

**3.5.3 ECONOMIC FEASIBILITY**

A computer system will take care of the requirements of this attendance system in terms of supporting the training set, training the data, running the code and report generation. For capturing the class image, a 12MP camera is required. (A Samsung 12MP camera approximately costing around Rs.9, 000/-)

**3.6 EFFORT AND COST ESTIMATION**

COCOMO (Constructive Cost Model) is a regression model based on LOC, i.e.  Number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models.

The key parameters which define the quality of any software products, which are also an outcome of the COCOMO, are primarily Effort & Schedule:

* **Effort:** Amount of labor that will be required to complete a task. It is measured in person-months units.
* **Schedule:** Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Different models of COCOMO have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below.

Boehm’s definition of organic, semidetached, and embedded systems:

1. **Organic –** A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.
2. **Semi-detached –** A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. Eg: Compilers or different Embedded Systems can be considered of Semi-Detached type.
3. **Embedded –** A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

**3.5.1 Basic COCOMO Model**

E=a\*(KLOC) ^b

D=c\*(E) ^d

Where E=Effort applied in person-months, D is the development time in chronological months and KLOC is estimated number of delivered lines of code for project expressed in per thousand. The coefficients a, b, c, d comes from the given table considering our project to be organic.

| **SOFTWARE PROJECTS** | **A** | **B** |
| --- | --- | --- |
| Organic | 2.4 | 1.05 |
| Semi Detached | 3.0 | 1.12 |
| Embedded | 3.6 | 1.20 |

Table 3.1 Coefficients for Basic COCOMO model

The KLOC for our project are as follows:

* 1. Emotion-Recognition-training.ipynb=0.057

Total=0.057

E=2.4\*(0.057) ^1.05

=0.118 person-months

D=2.5\*(0.118) ^0.38

=1.2 months

The number of people required to complete this project were 2.

**3.6 SYSTEM REQUIREMENT SPECIFICATION**

**3.6.1 PURPOSE**

In this contemporary world where the primary focus is to get the job done on time pushes human to their limit which encumbers them with workload that not only effects their health but also their throughput to produce results.

Stress detection using facial expressions is used to identify if a person is stressed out by facial detection, through this we can take necessary steps in order to rejuvenate him/her.

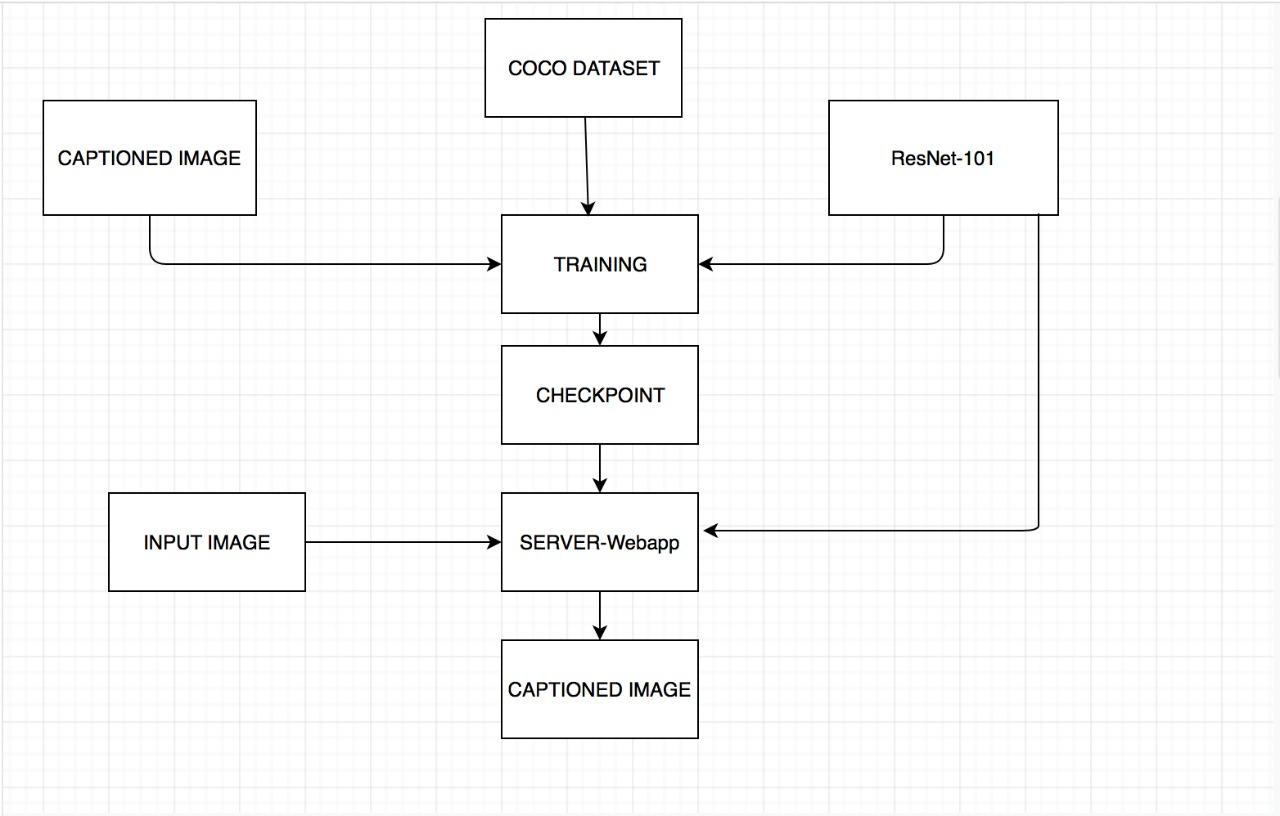
**3.6.2 SCOPE**

The aim of our project is to detect stress and take necessary steps to reduce it .This project can help every individual user and also helps in reducing the stress. . For the images that are stored in the database we apply system algorithm which includes face detection and face recognition steps.

**3.6.3. OVERALL DESCRIPTION**

Let us consider the various factors in overall description.

**3.6.3.1 PRODUCT PERSPECTIVE**



**3.6.3.2 PRODUCT FUNCTIONS**

● Allows users to detect stress based on their facial expressions.

● Can identify different types of expressions.

**3.6.3.3 OPERATING ENVIRONMENT**

**Software Requirements are:**

● Programming language: Python

● Windows XP or higher / Ubuntu

● CUDA and cudNN

● Tensorflow GPU

Python with NumPy , SciPy and FFMPEG are some important libraries.

**Python** - is one of those rare languages which can claim to be both *simple* and *powerful*. You

will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in.

Python is an easy to learn, powerful programming language. It has efficient high-level data

structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

**CUDA** - CUDA is NVIDIA’s parallel computing architecture that enables dramatic increases in computing performance by harnessing the power of the GPU (graphics processing unit).

With millions of CUDA-enabled GPUs sold to date, software developers, scientists and

researchers are finding broad-ranging uses for CUDA, including image and video processing,

computational biology and chemistry, fluid dynamics simulation, CT image reconstruction,

seismic analysis, ray tracing, and much more.

The CUDA parallel computing platform provides a few simple C and C++ extensions that enableexpressing fine-grained and coarse-grained data and task parallelism. The programmer canchoose to express the parallelism in high-level languages such as C, C++, Fortan or open

standards as OpenACC directive. The CUDA parallel computing platform is now widely deployed with 1000s of GPU-accelerated applications and 1000s of published research papers.

**CudNN** - The NVIDIA CUDA Deep Neural Network library (cuDNN) is a GPU-accelerated

library of primitives for deep neural networks. cuDNN provides highly tuned implementations for standard routines such as forward and backward convolution, pooling, normalization, and activation layers.

Deep learning researchers and framework developers worldwide rely on cuDNN for highperformance GPU acceleration. It allows them to focus on training neural networks and

developing software applications rather than spending time on low-level GPU performance

tuning. cuDNN accelerates widely used deep learning frameworks, including Caffe,Caffe2,

Chainer, Keras, MATLAB, MxNet, TensorFlow, and PyTorch. For access to NVIDIA optimized deep learning framework containers, that has cuDNN integrated into the frameworks.

**NumPy** - NumPy is the fundamental package for scientific computing with Python. It contains

among other things:

● A powerful N-dimensional array object

● sophisticated (broadcasting) functions

● tools for integrating C/C++ and Fortran code

● useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional

container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlesslyand speedily integrate with a wide variety of databases.

**SciPy** - SciPy (pronounced /ˈsaɪpaɪ'/ "Sigh Pie"[3]) is a free and open-source Python library usedfor scientific computing and technical computing.

SciPy contains modules for optimization, linear algebra, integration, interpolation, special

functions, FFT, signal and image processing, ODE solvers and other tasks common in science

and engineering.

SciPy builds on the NumPy array object and is part of the NumPy stack which includes tools like Matplotlib, pandas and SymPy, and an expanding set of scientific computing libraries. This NumPy stack has similar users to other applications such as MATLAB, GNU Octave, and Scilab.The NumPy stack is also sometimes referred to as the SciPy stack.

**FFMPEG** - FFmpeg is a free software project consisting of a vast software suite of libraries and programs for handling video, audio, and other multimedia files and streams. At its core is the FFmpeg program itself, designed for command-line-based processing of video and audio files, and widely used for format transcoding, basic editing (trimming and concatenation), video scaling, video post-production effects, and standards compliance (SMPTE, ITU). FFmpeg includes libavcodec, an audio/video codec library used by many commercial and free software products, libavformat (Lavf),[6] an audio/video container mux and demux library, and the core ffmpeg command line program for transcoding multimedia files.

**Tensorflow** - TensorFlow is an open source software library for numerical computation using

data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them. This flexible architecture lets you deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device without rewriting code.

TensorFlow was originally developed by researchers and engineers working on the Google Brain team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research. The system is general enough to be applicable in a wide variety of other domains, as well.

**3.6.4 EXTERNAL INTERFACE REQUIREMENT**

Using the tkinter library of python, the GUI was made which included appropriate buttons placed in a grid layout.

**3.6.5 SYSTEM FEATURES**

**3.6.5.1 Determining train loss, value loss and accuracy**

**3.6.5.1.1 Description and priority**

This feature involves using resnet34 a trained model and using it to create an image classifier from the available architecture and the path from where the images are to be loaded. We use this classifier to learn about data and train the resnet34 to work on raw input images.

**3.6.5.1.2 Stimulus/response sequences**

•This process list out the loss which occurs during training and loss in values along with accuracy.

•With each epoch the accuracy of model increases.

•At the end of 6th epoch model is fully trained with highest accuracy compared to the status of model in previous epoch.

**3.6.5.1.3 Functional requirements**

REQ-1: resnet34 trained model.

REQ-2: Images downloaded from google which is to be used as an input.

REQ-3: A system capable of running jypiter notebook or python installed on a system with all the system requirements pre-installed.

**3.6.5.2 Determining Emotions**

**3.6.5.2.1 Description and priority**

This feature involves identifying the emotions of a person x from the given input image.

**3.6.5.2.2 Stimulus/response sequences**

•Involves finding the training transformations and value transformations.

•The above recorded transformations are used to open image and predict the emotion or mood of the person.

**3.6.5.2.3 Functional requirements**

REQ-1: A trained model like resnet34.

REQ-2: Stable net connections to transfer images to the model.

1. **SYSTEM DESIGN**

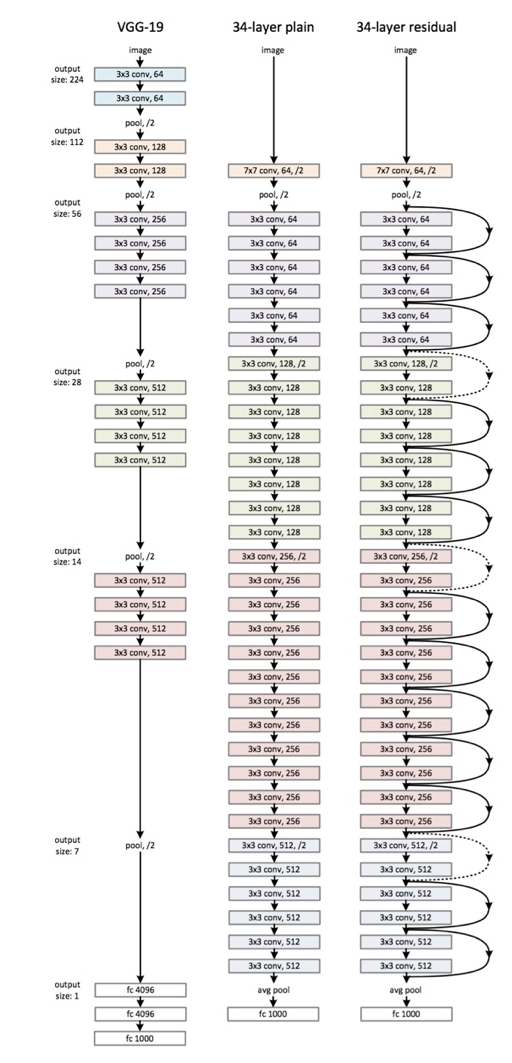
****

Fig.4.1 Resnet 34 architecture

**4.1 TRAINING PROCESS**

The pre-trained network used in this implementation is resnet 34. ResNet is a short name for a residual network.Pre-trained Model for PyTorch. ResNet has been trained with 34, 50, 101 and 152 layers. During training period, the residual network learns the weights of its layers such that if the identity mapping were optimal, all the weights get set to 0. In effect F(x) become 0, as in x gets directly mapped to H(x) and no corrections need to be made. Hence these become your identity mappings which help grow the network deep. And if there is a deviation from optimal identity mapping, weights and biases of F(x) are learned to adjust for it. Think of F(x) as learning how to adjust our predictions to match the actuals.

These networks are stacked together to arrive at a deep network architecture. For e.g., bellow is a ResNet arch with 34 layers.

**4.2 SYSTEM ARCHITECTURE**

**4.2.1.DATAFLOW DIAGRAMS**

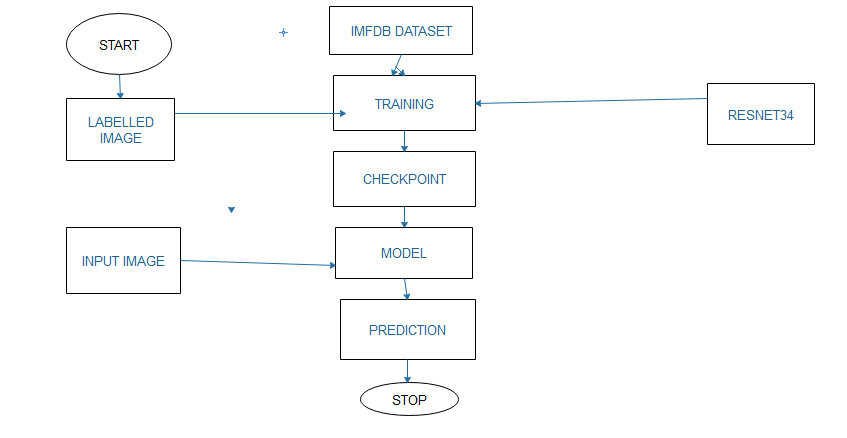
****

Fig 4.2 Data flow diagram for stress detection using facial expressions

**4.3. INTRODUCTION TO UML**

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part for developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**Goals of UML:**

1. Provide users with a ready-to-use, expressive visual modelling language so they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of the OO tools market.
6. Support higher-level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**4.3.1. USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioural diagram defined by and created from a use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

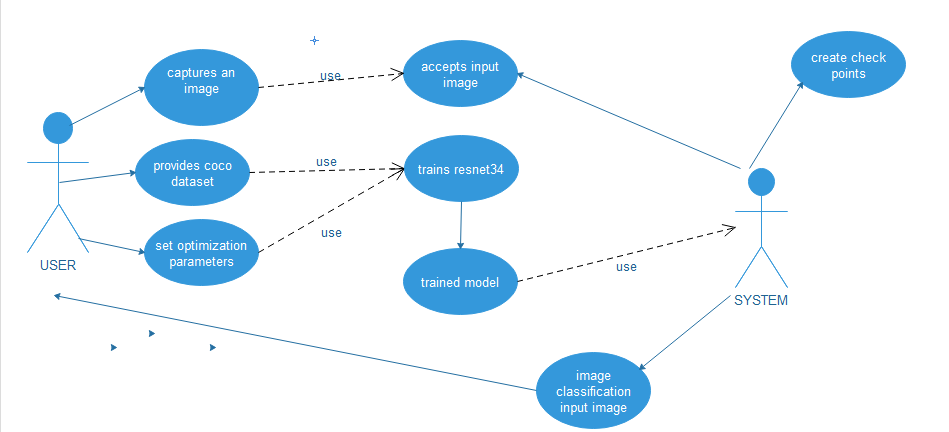


Fig 4.3 Use case diagram for stress detection using facial expressions

**4.3.2. CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML)is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

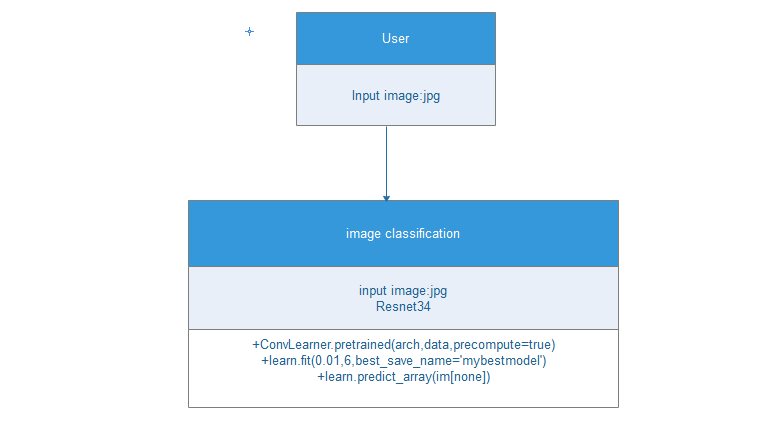
****

Fig 4.4 Class diagram for stress detection using facial expressions

**4.3.3,STATE CHART DIAGRAM:**

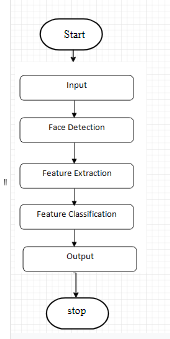
A state chart diagrams a state machine, emphasizing the flow of control from state tostate.

**Contents**

State chart diagrams commonly contain

Simply state and composite states

Transitions, including events and action

****

**Fig 4.5** State Chart diagram for stress detection using facial expressions

1. **IMPLEMENTATION**

Let us describe the steps and processes involved in the implementation if this system.

**5.1 Data Set**

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. Each image corresponds to a facial expression in one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The dataset contains approximately 36K images.

**5.1.1 About Cnn and use of it in our proposed work**

Classic NNs are usually composed of several fully connected layers. This means that every node of one layer is connected to all the nodes of the next layer.

Convolutional Neural Networks also have Convolutional layers that apply sliding functions to groups of pixels that are next to each other. Therefore those structures have a better understanding of patterns that we can observe in images. We will explain this in more details after.

The convolutional layers will extract relevant features from the images and the fully connected layers will focus on using these features to classify well our images.

Now let’s focus on how those convolution layers work. Each of them contain the following operations:

**•A convolution operator**: extracts features from the input image using sliding matrices to preserve the spatial relations between the pixels. The following image summarizes how it works:

The green matrix corresponds to the raw image values. The orange sliding matrix is called a ‘filter’ or ‘kernel’. This filter slides over the image by one pixel at each step (stride). During each step, we multiply the filter with the corresponding elements of the base matrix and sum the results. There are different types of filters and each one will be able to retrieve different image features:

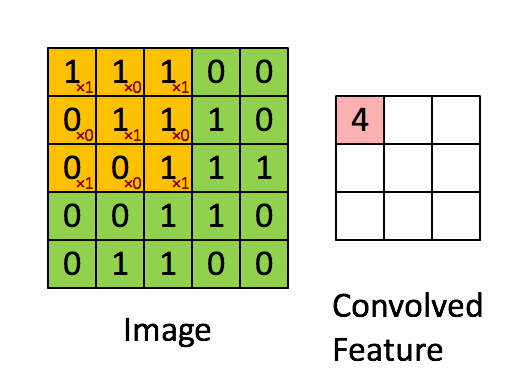
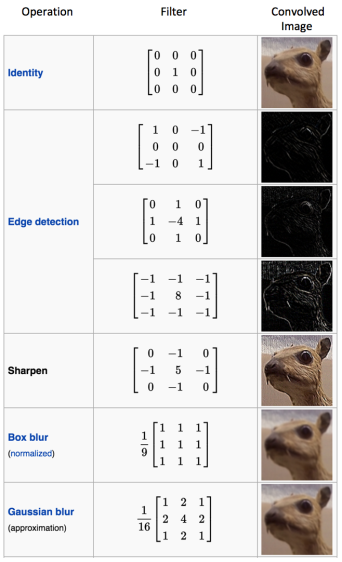


Fig 5.1. A convolution operator

 Fig 5.2 Different filter results

•We apply the ReLU function to introduce non linearity in our CNN. Other functions like tanh or sigmoid could also be used, but ReLU has been found to perform better in most situations.

•Pooling is used to reduce the dimensionality of each features while retaining the most important information. Like for the convolutional step, we apply a sliding function on our data. Different functions can be applied: max, sum, mean… The max function usually performs better.

We also use some common techniques for each layer:

**•Batch normalization**: improves the performance and stability of NNs by providing inputs with zero mean and unit variance.

**•Dropout**: reduces overfitting by randomly not updating the weights of some nodes. This helps prevent the NN from relying on one node in the layer too much.

We chose softmax as our last activation function as it is commonly used for multi-label classification.

Now that our CNN is defined, we can compile it with a few more parameters. We chose the Adam optimizer as it is one of the most computationally effective. We chose the categorical cross-entropy as our loss function as it is quite relevant for classification tasks. Our metric will be the accuracy, which is also quite informative for classification tasks on balanced datasets.

**5.1.2 At Training**

At each epoch, it is checked if our model performed better than the models of the previous epochs. If it is the case, the new best model weights are saved into a file. This will allow us to load directly the weights of our model without having to re-train it if we want to use it in another situation.

The training loss is slightly higher than the validation loss for the first epochs which can be surprising. Indeed we are used to see higher validation losses than training losses in machine learning. Here this is simply due to the presence of dropout, which is only applied during the training phase and not during the validation phase.

We can see that the training loss is becoming much smaller than the validation loss after the 20th iteration. This means that our model starts to overfit our training dataset after too much epochs. That is why the validation loss does not decrease a lot after. One solution consists in early-stopping the training of the model.

We could also use some different dropout values and performing data augmentation. Those methods were tested on this dataset, but they did not significantly increase the validation accuracy although they reduced the overfitting effect. Using them slightly increased the training duration of the model.

**5.2 SAMPLE CODE:**

**Step by Step go through**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import matplotlib.image as mpimg

from sklearn import linear\_model

from fastai.imports import \*

%reload\_ext autoreload

%autoreload 2

%matplotlib inline

from fastai.transforms import \*

from fastai.conv\_learner import \*

from fastai.model import \*

from fastai.dataset import \*

from fastai.sgdr import \*

from fastai.plots import \*

PATH = "/Users/syedimad/Desktop/Facial-Expression/images"

sz=224

arch=resnet34

data=ImageClassifierData.from\_paths(PATH,tfms=tfms\_from\_model(arch,sz))

learn=ConvLearner.pretrained(arch,data,precompute=True)

learn.fit(0.01,6,best\_save\_name='mybestmodel')

data.classes

# We'll pass in a raw image, not activations

learn.precompute=False

trn\_tfms, val\_tfms = tfms\_from\_model(arch,sz) # get transformations

im = val\_tfms(open\_image('/Users/test/sad.jpg'))

image = mpimg.imread("/Users/test/sad.jpg")

plt.imshow(image)

plt.show()

preds = learn.predict\_array(im[None])

pred=np.argmax(preds)

print("Looks like the person is: "+data.classes[pred])

im = val\_tfms(open\_image('/Users/test/angry.png'))

image = mpimg.imread("/Users/test/angry.png")

plt.imshow(image)

plt.show()

preds = learn.predict\_array(im[None])

pred=np.argmax(preds)

print("Looks like the person is: "+data.classes[pred])

im = val\_tfms(open\_image('/Users/test/surprised.jpg'))

image = mpimg.imread("/Users/test/surprised.jpg")

plt.imshow(image)

plt.show()

preds = learn.predict\_array(im[None])

pred=np.argmax(preds)

print("Looks like the person is: "+data.classes[pred])

im = val\_tfms(open\_image('/Users/test/test1.jpg'))

image = mpimg.imread("/Users/test/test1.jpg")

plt.imshow(image)

plt.show()

preds = learn.predict\_array(im[None])

pred=np.argmax(preds)

print("Looks like the person is: "+data.classes[pred])

**6. 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.

**6.1 Types of Testing**

**6.1.1 Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of 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 endure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Test Cases:**

A test case is a set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do.

Test Case 1:

Test Case: Training Module

Test Objective: To check whether the recognizer is training the image dataset or not.

Test Description: The user (i.e. in this case the admin) will run the training module code to train the images of the students with their respective labels.

Requirement verified: Yes

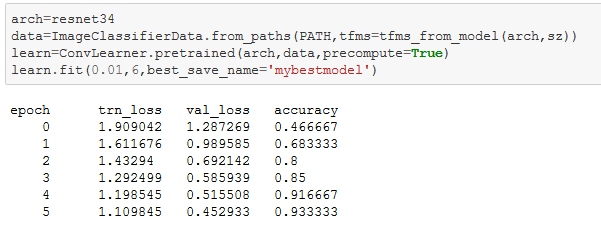
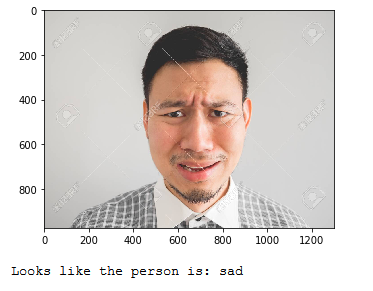


Fig 6.1: Training Module logs

Test Case 2:

Test Case: Stress detection

Test Description: to check whether it is able to recognize or not



Requirement Verified: Yes

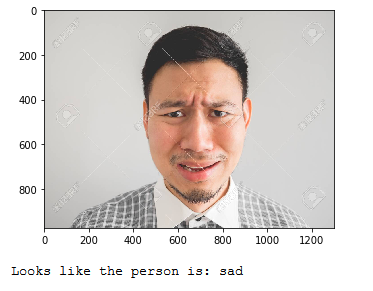
Test Successful.

**6.1.2 Integration Testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Although the components were individually tested successfully through unit testing, integration tests demonstrate whether the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Test case:Detecting stress

Test Result:All the components are working together to yield the results



Able to select best model and detect emotions.

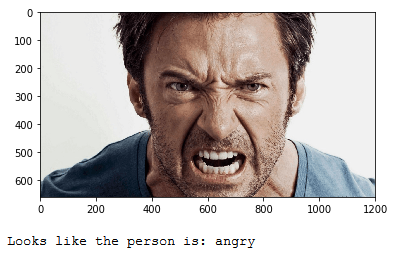
**6.1.3 Black Box Testing**

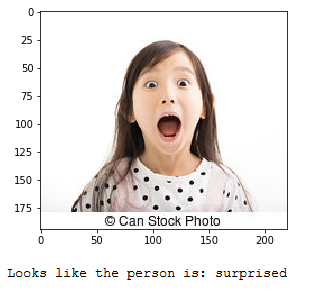
Black Box Testing is testing the software without any knowledge of the inner working, structure or knowledge of the module being tested. These tests can be functional or non-functional, though usually functional. This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in data structures or external database access
* Behavior or performance errors
* Initialization and termination errors

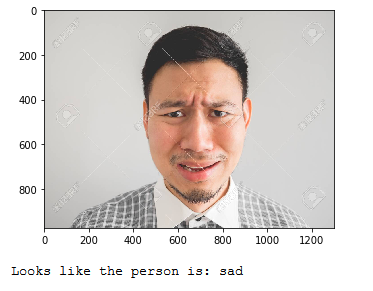
Test Case: Performance

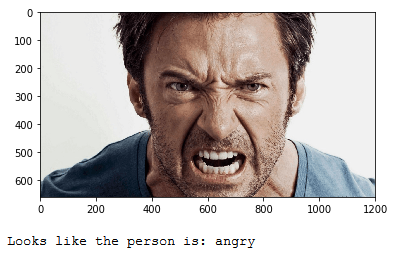
Test Objective: Giving random images to see if it is able to detect

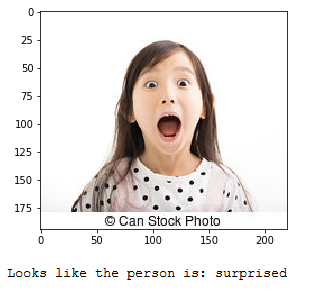




**7. SCREENSHOTS**



****

****

**8. CONCLUSION**

This system investigates the use of task elicited facial signs as indices of anxiety/stress in relation to neutral and relaxed states. Although there is much literature discussing recognition and analysis of the six basic emotions, i.e. anger, disgust, fear, happiness. Studies and used, among other features, facial features achieving 75%–88% and 90.5% classification accuracy respectively. This system uses facial signs as indices of anxiety or stress in relation to neutral and relaxed state. This system involved eye related features and head movement for stress detection. All these features provide a contactless approach of stress detection not interfering with human body in relation to other related studies employing semi-invasive measurements like ECG, EEG, galvanic skin response and skin temperature.

**9. REFERENCES**

[1]E. Fox, A. Mathews, A.J. Calder, J. Yiend, “Anxiety and sensitivity to gaze direction in emotionally expressive faces, Emotion waved the possibility of detecting stress through facial expressions” <https://www.ncbi.nlm.nih.gov/pubmed/17683204>

[2] N. Schneider man, G. Ironson, S.D. Siegel, Stress and health: psychological, behavioral, and biological determinants, Ann. Rev. Clin. Psychol. 1 (2005) 607.

[3] H. Selye, The stress syndrome, AJN Am. J. Nursing 65 (1965) 97–99.

[4] L. Vitetta, B. Anton, F. Cortizo, A. Sali, Mind-Body medicine: stress and its impact on overall health and longevity, Ann. N. Y. Acad. Sci. 1057 (2005)492–505.

[5] R., Donatelle (2013). My Health: An Outcomes Approach, Pearson Education

[6] <https://towardsdatascience.com/from-raw-images-to-real-time-predictions-with-deep-learning-ddbbda1be0e4>

[7]<https://www.google.com/search?q=resnet+architecture&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjW0vHj8sPhAhXTIqYKHW3BCV4Q_AUIDigB&biw=1066&bih=609#imgrc=phdenOMrAHMa0M>:

**APPENDIX I**

|  |  |
| --- | --- |
| **PROJECT TITLE** | **CATEGORY** |
| STRESS DETECTION USING FACIAL RECOGNITION | RESEARCH |

**ABSTRACT:**

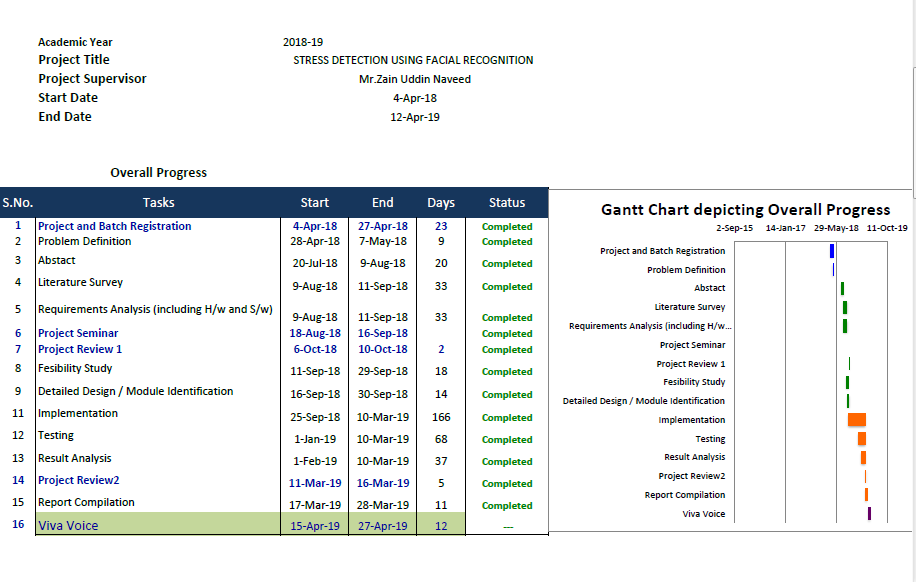
Psychological stress is threatening people's health. Stress and anxiety are common everyday life states of emotional strain that effects quality of life.These states consist of several complementary and interacting components (i.e., cognitive, affective, central and peripheral physiological).Prolonged stress and anxiety can be associated with psychological and/or somatic diseases.Increased skeletal, smooth and cardiac muscle tension, disturbances are typical signs of stress and anxiety, which are linked to some of their most common symptoms and disorders . (Headache, Hypertension,etc) It is non-trivial to detect stress timely for proactive care.

For the well being of a person as well as to be productive it becomes necessary to reduce the stress. .Here we provide a medium which scans the face of a persons and identifies what kind of a mood he/she is and indicate when a person is stressed out.Through our project we aim to identify if a person is stressed out by the facial detection ,through this we can take necessary steps in order to rejuvenate him/her.

**Mapping to POs and PSOs: (Mapping Scale [1-3]: 1-Slight 2-Moderate 3-Strong)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **PO**  **1** | **PO**  **2** | **PO**  **3** | **PO**  **4** | **PO**  **5** | **PO**  **6** | **PO**  **7** | **PO**  **8** | **PO**  **9** | **PO**  **10** | **PO**  **11** | **PSO**  **1** | **PSO**  **2** | **PSO**  **3** |
| **3** | **3** | **3** | **3** | **2** | **3** | **1** | **2** | **3** | **2** | **2** | **2** | **2** | **3** |

|  |  |
| --- | --- |
| **RELEVANCE DETAILS** | |
| **Implementation Details**  The project is implemented using Languages: Python | |
| **PO and PSO Justification** | |
| **PO1** | It is strongly mapped as the basic concepts of mathematics, physics and engineering fundamentals are used for designing a solution to the complex engineering problem to apply automatic brakes for avoiding collisions |
| **PO2** | It is strongly mapped as the high transmission rate a concept of natural and engineering sciences are used so prevent delay between two readings from sensors as a solution to complex engineering problem after doing the analysis. |
| **PO3** | It is strongly mapped as risks, responsiveness and efficiency are considered to cater to the societal needs, health and public safety. |
| **PO4** | It is strongly mapped as many investigations were conducted on different brake system to select the most appropriate one for the proposed system. Critical thinking and the ability to analyze and solve complex real-world problems is done for developing the product. |
| **PO5** | It is moderate mapped as the embedded kit concepts integrated with sensors is an example of new modern tool collaboration |
| **PO6** | It is strongly mapped as the role of the sensors in the project is to help the society for avoiding accidents |
| **PO7** | It is weakly mapped as the project needs to be sustainable to the environment but most of the projects designed are non-renewable and perishable. |
| **PO8** | It is strongly mapped because the plagiarism tool can be used to check the authenticity and originality of the work done by the students |
| **PO9** | Projects are used to inculcate group work and to manage a team for promoting knowledge, conceptualization and delivering same with varied complexity, therefore the mapping is strong. |
| **PO**  **10** | Demonstrate versatile and effective communication skills, both verbal and written with team members and present the product to the audience in comprehensive manner. Therefore the mapping is moderate. |
| **PO**  **11** | Project Management tools like CPM and Pert etc can be used to propose and work accordingly, hence the mapping is moderate. |
| **PSO**  **1** | It is strongly mapped as understanding of the principles and working of the hardware and software aspects of computer systems is required to decompose the system into phases and workflows. |
| **PSO**  **2** | It is moderately mapped as cases are explored to ensure reliability of the product ,optimal quality and tested thoroughly to validate the ongoing process |
| **PSO**  **3** | It is strongly mapped as self-learning skills are applied for a real world problem by using content knowledge and acquired intellectual skills. E.g.: The product is targeted for vehicles, 2-wheelers, in particular. Also applicable in 4-wheelers. |



Appendix-2