

**A MAJOR PROJECT REPORT
ON
REAL TIME HUMAN EMOTION RECOGNITION
BASED ON FACIAL EXPRESSION DETECTION USING
SOFTMAX CLASSIFIER AND PREDICT THE ERROR
LEVEL USING OPEN CV LIBRARY**

**Submitted to Sri Indu College of Engineering & Technology, Hyd
In partial fulfillment of the requirements for the award of degree of
BACHELOR OF TECHNOLOGY**

IN

COMPUTER SCIENCE ENGINEERING

Submitted by

K.SUDHEER	(17D41A05A4)
G.SAI SATHWIK	(17D41A0574)
G.BHARGAV	(17D41A0572)
K.KISHORE	(17D41A05A7)

Under the esteemed guidance of

Mrs B.SHWETHA(Asst.Prof)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution under UGC, Accredited by NBA&NAAC, Affiliated to JNTUH)
Sheriguda, Ibrahimpatnam (2019-2020)**

SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY
(An Autonomous Institution under UGC, Accredited by NBA&NAAC, Affiliated to JNTUH)

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING



CERTIFICATE

This is to certify that the major project work entitled

**“REAL TIME HUMAN EMOTION RECOGNISATION BASED ON FACIAL
EXPRESSION DETECTION USING SOFTMAX CLASSIFIER AND PREDICT
THE ERROR LEVEL USING OPENCV LIBRARY”**

Submitted by

K. SUDHEER	(17D41A05A4)
G.SAI SATHWIK	(17D41A0574)
G. BHARGAV	(17D41A0572)
K. KISHORE	(17D41A05A7)

In partial fulfilment for the award of Bachelor of Technology in Information Technology to the SICET, is a record of Bonafide work carried out by them under my guidance and supervision during academic year **2019-2020**.

Internal Guide:
Mrs.K.SHWETHA

HOD
(Dept of CSE)
DR.S.R.MUGUNTHAN

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the effort with success.

We are thankful to principal **Dr.G.Suresh**, for giving us the permission to carry out this project and for Providing necessary infrastructure and labs. We are highly indebted to **Dr.S.R.Mugunthan**, Head of the Department of Computer Science Engineering, for providing valuable guidance at every stage of this project.

We are grateful to our internal project guide **Mrs K.Shwetha (Asst Prof)** for his constant motivation and guidance given by him during the execution of this project work.

We would like to thank the Teaching & Non-Teaching staff of Department of Information Technology for sharing their knowledge with us, last but not least we express our sincere thanks to everyone who helped directly or indirectly for the completion of this project.

K.SUDHEER	(17D41A05A4)
G.SAI SATHWIK	(17D41A0574)
G.BHARGAV	(17D41A0572)
K.KISHORE	(17D41A05A7)

INDEX

TITLES	PAGE NO
<u>CONTENTS</u>	
ABSTRACT	vi
1. INTRODUCTION	1
2. LITERATURE SURVEY	5
2.1 KEY-DEDUPLICATION WITH IBBE	5
2.2 SERVER LESS DISTRIBUTEDFILESYSTEM	6
2.3 THE GOOGLEFILESYSTEM	7
2.4 CONVERGENTKEYMANAGEMENT	8
2.5 SOFTWAREENVIRONMENT	9
2.6 WHY CHOOSEPYTHON	10
3. SYSTEM ANALYSIS	14
3.1 EXISTING SYSTEM	16
3.2 PROPOSEDSYSTEM	16
4. FEASIBILITYSTUDY	17
4.1 ECONOMICALFEASIBILITY	17
4.2 TECHNICALFEASIBILITY	17
4.3 SOCIALFEASIBILITY	18
5. SYSTEMREQUIREMENTS	19

6. SYSTEMDESIGN	20
6.1 SYSTEMARCHITECTURE	20
6.2 DATAFLOW DIAGRAM	20
6.3 UMLDIAGRAMS	22
7. IMPLEMENTATION	27
7.1 MODULES	27
7.2 SAMPLE CODE	28
8. SYSTEMTESTING	29
8.1 UNITTESTING	31
8.2 INTEGRATION TESTING	32
8.3 ACCEPTANCETESTING	33
9. INPUT DESIGN ANDOUTPUTDESIGN	34
9.1 INPUTDESIGN	34
9.2 OUTPUTDESIGN	35
10. SCREENSHOTS	37
11. FUTUREWORK	50
12. CONCLUSION	51
13. BIBLOGRAPHY	52

ABSTRACT:

Facial Detection and recognition research has been widely studied in recent years. The facial recognition applications plays an important role in many areas such as security, camera surveillance, identity verification in modern electronic devices, criminal investigations, database management systems and smart card applications etc. This work presents deep learning algorithms used in facial recognition for accurate identification and detection. The main objective of facial recognition is to authenticate and identify the facial features. However, the facial features are captured in real time and processed using haar cascade detection. The sequential process of the work is defined in three different phases where in the first phase human face is detected from the camera and in the second phase, the captured input is analyzed based on the features and database used with support of keras convolutional neural network model. In the last phase human face is authenticated to classify the emotions of human as happy, neutral, angry, sad, disgust and surprise. The proposed work presented is simplified in three objectives as face detection, recognition and emotion classification. In support of this work Open CV library, dataset and python programming is used for computer vision techniques involved. In order to prove real time efficacy, an experiment was conducted for multiple students to identify their inner emotions and find physiological changes for each face. The results of the experiments demonstrates the perfections in face analysis system. Finally, the performance of automatic face detection and recognition is measured with Accuracy.

1. INTRODUCTION

Human computer interaction is a common trend and innate ability to distinguish among multiple faces. Until recent past computer vision problems were quite challenging but advent of modern technologies has trivially improved from the problems of varying light, changed by age, hair and other accessories [1]. However, face recognition applications are used improve access to identify and verify the people by their face features. Hence interpreting the facial features and their actions is much required. As these features and expressions helps in classify the emotions of human face. Recent advances in technology has resulted in the use of Artificial intelligence system as these systems are capable to understand and realize the emotion recognition through facial features. Hence this is an attempt to prove the existence of latest technological developments for human-computer interaction using deep learning or Convolution neural network models [2]. To recognize and classify the human face various methods are required but deep learning technique outperforms other methods by its large capabilities of different datasets and fast computation capabilities. Usually the process of face recognition and classification involves various steps such as preprocessing, detection, orientation, extraction of features and classification of emotion. These tasks are easily performed with deep learning keras model which outperforms the handy computations. Deep learning technique is a standard paradigm to represent the working of human brain with neurons [3]. This learning usually consists of neural network model where neurons act as inputs and each of them are connected to move as outputs. Deep learning is the subset of machine learning in which the algorithms are integrated similar to machine learning, but there are innumerable levels of these algorithms, each of which provides a different interpretation of the data they insert. This network of algorithms is called the network of artificial neurons, because their operation is a

source of inspiration, or we could say; an attempt to mimic the function of human neural networks in the brain [7]. Several hidden levels allow deep neural networks to study data functions in the so-called functional hierarchy, since simple functions, such as two pixels, combine from one level to another, forming, for example, functions that are more complex. Low level networks are not much capable of processing mathematical operations than multilevel networks as these networks are able to perform deep data processing. Deep learning models have very good characteristics of calculating the intensive calculus and has great demand in the form of minute chip called Graphical processing unit [9]. There are three different phases of this work such as facial detection, recognition and emotion classification which are outlined briefly in the below sections.

2. LITERATURE SURVEY

2.1 Adaptive fuzzy rule based Classification systems

Authors: Ken Nozaki, Hisao Ishibuchi

Abstract: This paper proposes an adaptive method to construct a fuzzy rule-based classification system with high performance for pattern classification problems. The proposed method consists of two procedures: an error correction-based learning procedure, and an additional learning procedure. The error correction-based learning procedure adjusts the grade of certainty of each fuzzy rule by its classification performance. That is, when a pattern is misclassified by a particular fuzzy rule, the grade of certainty of that rule is decreased. On the contrary, when a pattern is correctly classified, the grade of certainty is increased. Because the error correction-based learning procedure is not meaningful after all the given patterns are correctly classified, we cannot adjust a classification boundary in such a case. To acquire a more intuitively acceptable boundary, we propose an additional learning procedure. We also propose a method for selecting significant fuzzy rules by pruning unnecessary fuzzy rules, which consists of the error correction-based learning procedure and the concept of forgetting. We can construct a compact fuzzy rule-based classification system with high performance.

2.2 Image Steganalysis Using a Bee Colony Based Feature Selection Algorithm

Authors: Farid Ghareh Mohammadi, Mohammed Saniee Abadeh

Abstract: Feature selection is one of the most significant phases of pre-analysis processing, which can influence the performance of steganalysis. In this paper, we have proposed a new feature-based blind steganalysis method for detecting stego images from the cover images in JPEG images using a feature selection technique based on artificial bee colony (IFAB). Most usual techniques for feature selection are wrapper methods and filter methods which IFAB is one of the wrapper based feature selection methods. Artificial bee colony (ABC) algorithm is inspired by honey bees' social behavior in their search for perfect food sources. However, in the suggested algorithm, classifier performance and the dimension of the selected feature vector are dependent on heuristic information for ABC. As a result, we can choose the adaptive feature subset with respect to the shortest feature dimension and the improved performance of the classifier. The experimental results show that the proposed approach is easy to be employed for steganalysis purposes. Moreover, since IFAB is used as one of wrapper methods, as a result, its overall performance is better than several recent and well-known feature selection methods.

2.3 Significance of feature selection for image Steganalysis

Authors: Chhikara, M K,

Abstract: Steganalysis is capable of identifying the carrier(s) which have information hidden in them in such a way that their very existence is concealed. In this paper we propose a classification system with neural

networks which reduces computational complexity through a pre-processing step (feature selection) performed by Bhattacharyya distance for image steganalysis. This approach is able to identify relevant features which are a subset of original features extracted from spatial as well as transform domain. It helps in overcoming the problem of “curse of dimensionality” by removing redundant features by feature selection step before classifying the dataset. The experiments are performed on dataset obtained by four steganography algorithms outguess, steghide, PQ and nsF5 with two classifiers Support Vector Machine and Back Propagation neural networks. Classifier in combination with Bhattacharyya distance filter feature selection approach shows an improvement of 2-20% against total number of features.

2.4 A survey on Image steganography and Steganalysis,

Authors: Bin Li

Abstract: Steganography and steganalysis are important topics in information hiding. Steganography refers to the technology of hiding

data into digital media without drawing any suspicion, while steganalysis is the art of detecting the presence of steganography. This paper provides a survey on steganography and steganalysis for digital images, mainly covering the fundamental concepts, the progress of steganographic methods for images in spatial representation and in JPEG format, and the development of the corresponding steganalytic schemes. Some commonly used strategies for improving steganographic security and enhancing steganalytic capability are summarized and possible research trends are discussed.

2.5 SOFTWARE ENVIRONMENT

Python is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt.Sources newsgroup in 1991, and version 1.0 was released in 1994.

Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been back ported to Python 2. But in general, they remain not quite compatible.

Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End of Life date of January 1, 2020 has been established for Python 2, after which time it will no longer be maintained. If you are a newcomer to Python, it is recommended that you focus on Python 3, as this tutorial will do.

Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe Monty Python's Flying Circus, of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.

2.6 WHY CHOOSE PYTHON

If you're going to write programs, there are literally dozens of commonly used languages to choose from. Why choose Python? Here are some of the features that make Python an appealing choice.

Python is Popular

Python has been growing in popularity over the last few years. The 2018 [Stack Overflow Developer Survey](#) ranked Python as the 7th most popular and the number one most wanted technology of the year. [World-class software development countries around the globe use Python every single day.](#)

According to [research by Dice](#) Python is also one of the hottest skills to have and the most popular programming language in the world based on the [Popularity of Programming Language Index](#).

Due to the popularity and widespread use of Python as a programming language, Python developers are sought after and paid well. If you'd like to dig deeper into [Python salary statistics and job opportunities](#), you can do so here.

Python is interpreted

Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.

This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.

One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.

In practice, however, for most programs, the difference in execution speed is

measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

Python is Free

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes.

A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, macOS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

Python is Portable

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.)

Python is Simple

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way.

A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language.

Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect.

Python code has a simple and clean structure that is easy to learn and easy to read. In fact, as you will see, the language definition enforces code structure that is easy to read.

But It's Not That Simple

For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming.

Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.

Python accomplishes what many programming languages don't: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

Conclusion

This section gave an overview of the **Python** programming language, including:

- A brief history of the development of Python
- Some reasons why you might select Python as your language of choice

Python is a great option, whether you are a beginning programmer looking to learn the basics, an experienced programmer designing a large application, or anywhere in between. The basics of Python are easily grasped, and yet its capabilities are vast. Proceed to the next section to learn how to acquire and install Python on your computer.

Python is an open source programming language that was made to be easy-to-read and powerful. A Dutch programmer named Guido van Rossum made Python in 1991. He named it after the television show Monty Python's Flying Circus. Many Python examples and tutorials include jokes from the show.

Python is an interpreted language. Interpreted languages do not need to be compiled to run. A program called an interpreter runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like C, because it is not running machine code directly.

Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages.

Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp.

Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a semicolon at the end of each line and curly braces ({}) are used to group code. Using whitespace to delimit code makes Python a very easy-to-read language.

Python use [change / change source]

Python is used by hundreds of thousands of programmers and is used in many places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks.

Its standard library is made up of many functions that come with Python when it is installed. On the Internet there are many other libraries available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.

Some things that Python is often used for are:

- Web development
- Scientific programming
- Desktop GUIs
- Network programming
- Game programming

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

Research has been widely studied in recent years. The facial recognition applications plays an important role in many areas such as security, camera surveillance, identity verification in modern electronic devices,

criminal investigations, database management systems and smart card applications etc. This work presents deep learning algorithms used in facial recognition for accurate identification and detection. The main objective of facial recognition is to authenticate and identify the facial features.

3.2 PROPOSED SYSTEM:

The proposed work presented is simplified in three objectives as face detection, recognition and emotion classification. In support of this work Open CV library, dataset and python programming is used for computer vision techniques involved. In order to prove real time efficacy, an experiment was conducted for multiple students to identify their inner emotions and find physiological changes for each face. The results of the experiments demonstrates the perfections in face analysis system. Finally, the performance of automatic face detection and recognition is measured with Accuracy.

4. FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is

not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

4.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a

modest requirement, as only minimal or null changes are required for implementing this system.

4.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

5. SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS:

- System : Pentium Dual Core.
- Hard Disk : 120 GB.
- Monitor : 15'' LED
- Input Devices : Keyboard, Mouse

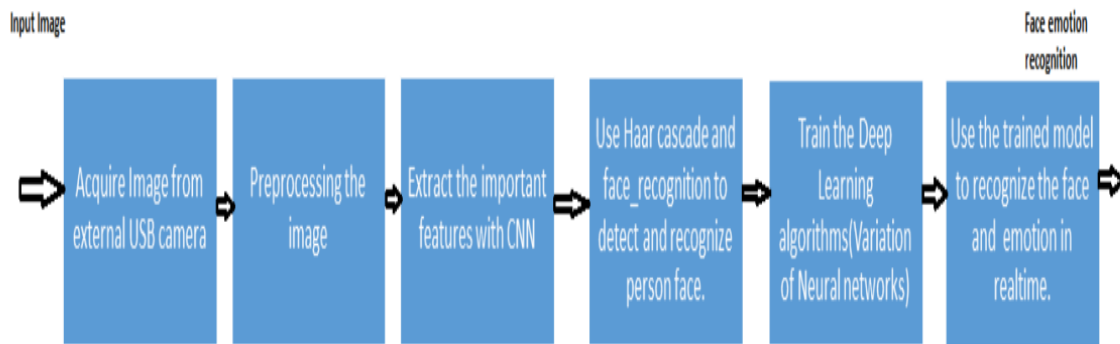
- Ram : 1 GB

5.2 SOFTWARE REQUIREMENTS:

- Operating system : Windows 10
- Coding Language : python
- Tool : PyCharm
- Database : MYSQL
- Server : Flask

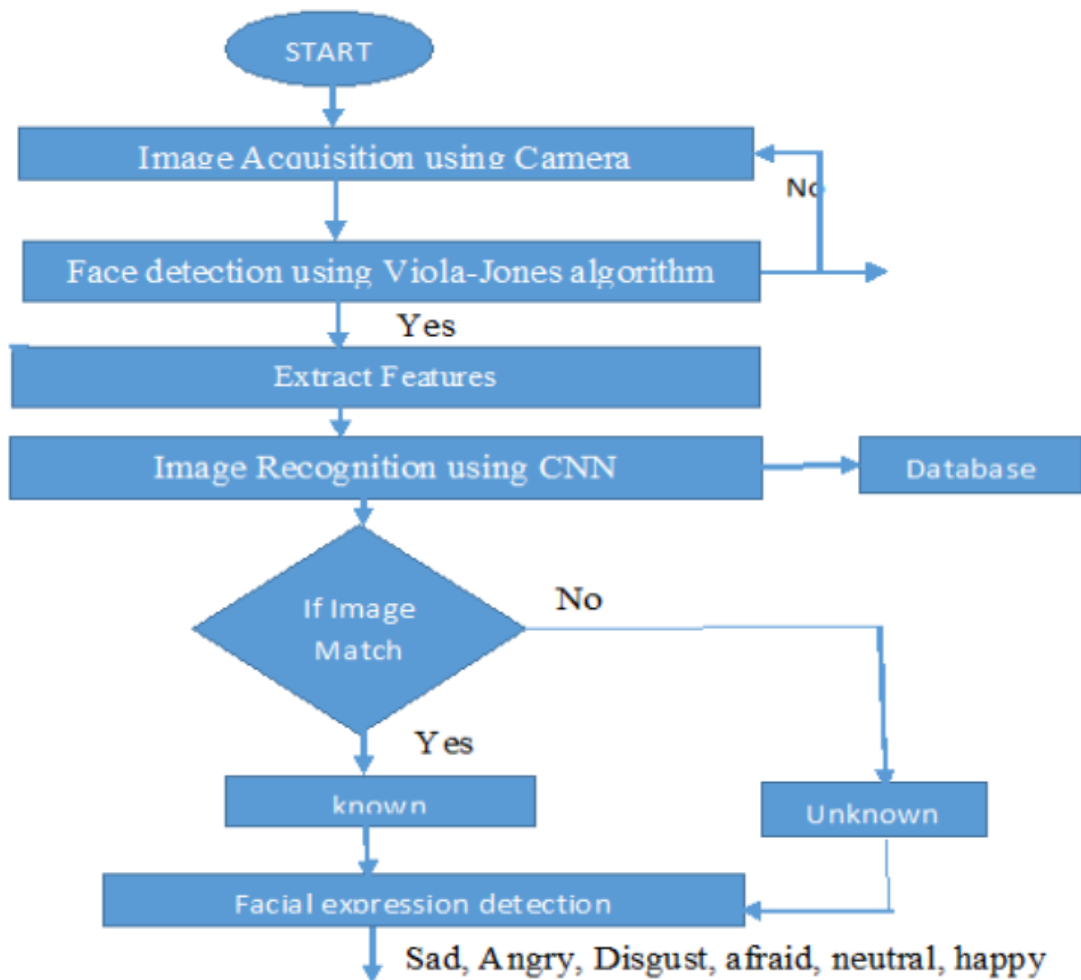
6. SYSTEM DESIGN

6.1 SYSTEM ARCHITECTURE:



6.2 DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical
4. Technique that depicts information flow and the transformations that are applied as data moves from input to output.
5. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



6.3 UML DIAGRAMS:

UML stands for Unified Modeling Language. UML 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 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 standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

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 of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

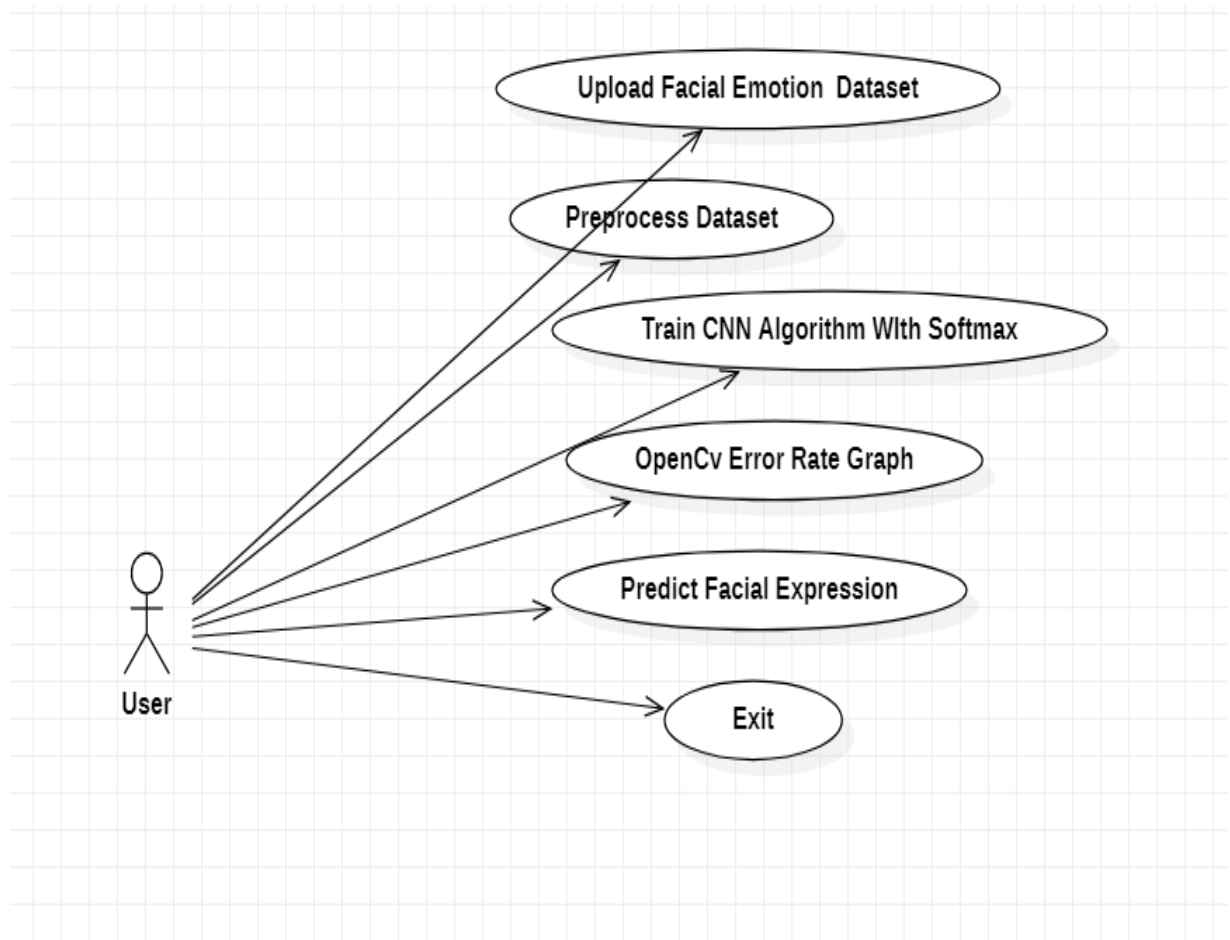
1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.

3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Integrate best practices.

USE CASE DIAGRAM:

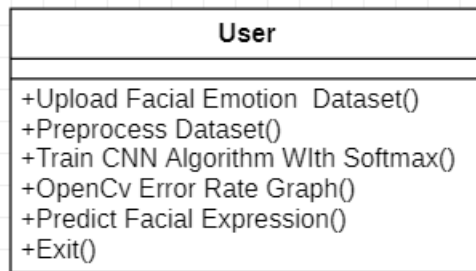
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies 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.



CLASS DIAGRAM:

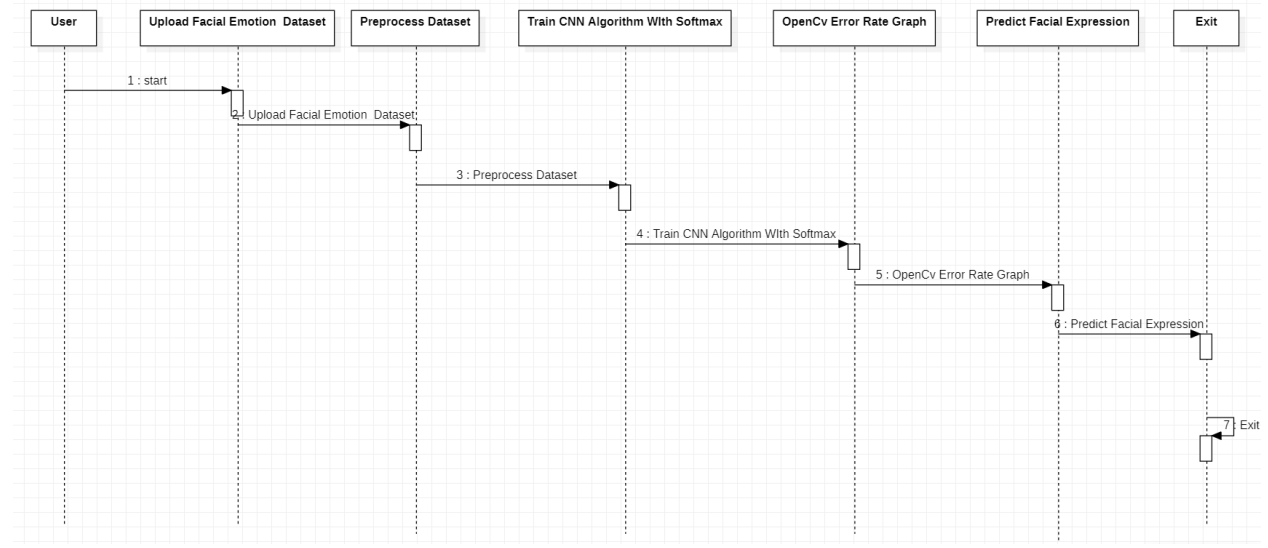
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains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



SEQUENCE DIAGRAM:

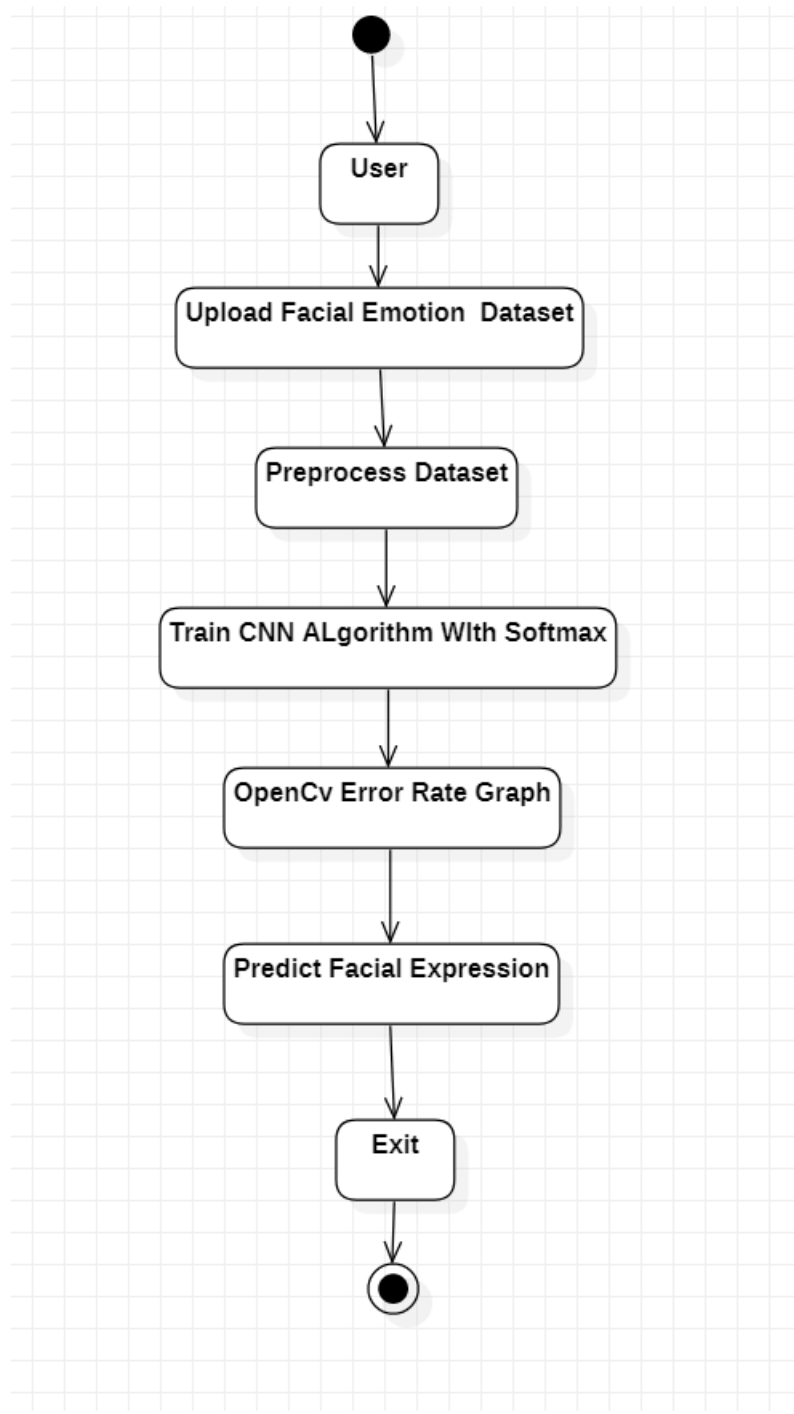
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.

Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



7. IMPLEMENTATION

7.1 MODULES:

- ❖ Upload Facial Emotion Dataset
- ❖ Preprocess Dataset
- ❖ Train CNN ALgorithm With Softmax
- ❖ OpenCv Error Rate Graph
- ❖ Predict Facial Expression
- ❖ Exit

MODULES DESCRIPTION:

Upload Facial Emotion Dataset

In Upload Facial Emotion Dataset to upload dataset here.

Preprocess Dataset

In Preprocess Dataset to read all images and then apply feature extraction algorithm called PCA to read important features from dataset.

Train CNN ALgorithm With Softmax

In this module train CNN with process image features.

OpenCv Error Rate Graph

In this module graph shown.

Predict Facial Expression

In this module upload new test image and the application predict emotion from it

Exit

Application closed here..

7.2 SAMPLE CODE

```
def upload():
    global filename
    filename = filedialog.askopenfilename(initialdir="model")
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n");

def processDataset():
    text.delete('1.0', END)
    global X, Y
    ""
    X = []
    Y = []
    for root, dirs, directory in
os.walk(filename):
    for j in range(len(directory)):
        name = os.path.basename(root)
        print(name+" "+root+"/"+directory[j])
        if 'Thumbs.db' not in directory[j]:
```



```
img =  
cv2.imread(root+"/"+directory[j])  
img = cv2.resize(img, (32,32))  
im2arr = np.array(img)  
im2arr = im2arr.reshape(32,32,3)  
X.append(im2arr)  
Y.append(getID(name))  
  
X = np.asarray(X)  
Y = np.asarray(Y)  
print(Y)
```

8. 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.

TYPES OF TESTS

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 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.

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. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test:

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.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields,

predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test:

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.

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

8.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach:

Field testing will be performed manually and functional tests will be written in detail.

Test objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

8.2 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.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.3 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.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9. INPUT DESIGN AND OUTPUT DESIGN

9.1 INPUT DESIGN:

The input design is the link between the information system

and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES:

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized

system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

9.2 OUTPUT DESIGN:

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the

requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

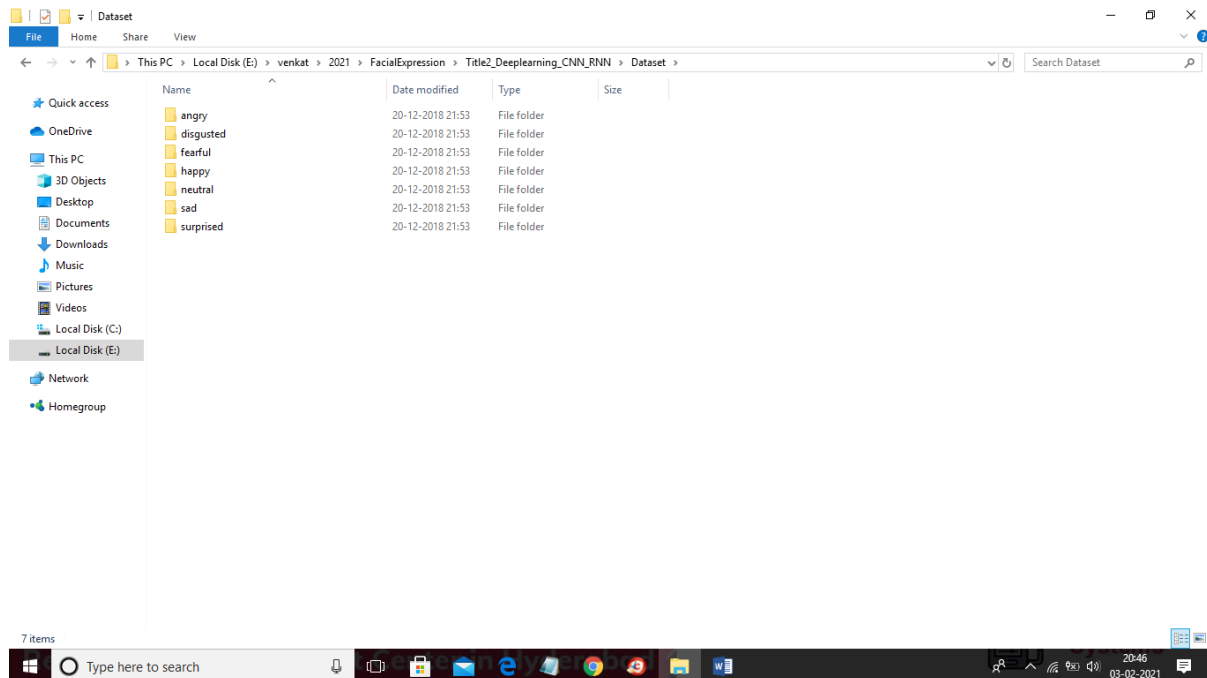
The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

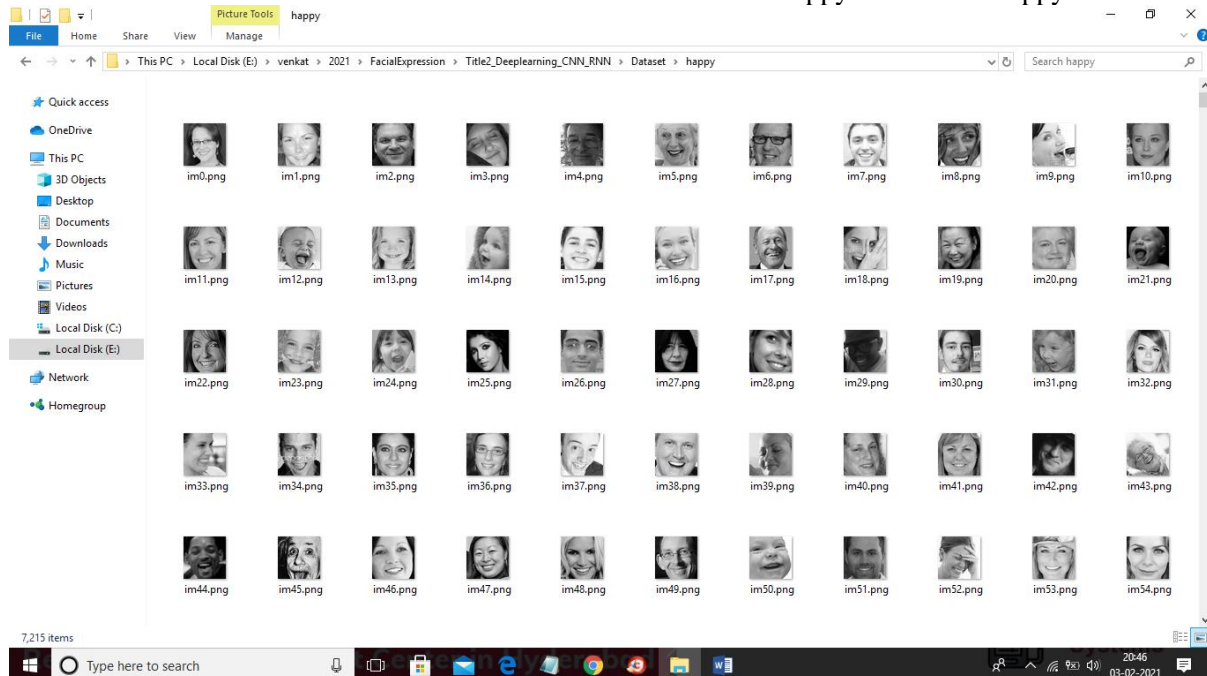
10. SCREENSHOTS

In this project we have built 4 projects with CNN and RNN, Feature extraction, YOLO and OPENCV to detect facial expressions or emotions from human faces.

To implement this project we are using facial expression dataset which contains faces of seven emotions. See below screen shots of dataset



Each folder contains its related faces which can be seen in below happy folder with happy faces



In above screen we can see faces from each emotion and using above dataset we have implemented following modules

- 1) In title 1 we are applying PCA feature extraction algorithm to extract important features from dataset and then train CNN and RNN algorithm and then compare accuracy of both algorithms

- 2) In title 2 we are using above dataset and deep learning CNN and RNN algorithms to build emotion prediction model and then calculate and compare accuracy between CNN and RNN
- 3) In title 3 we are using combination of YOLO face detection and CNN to build emotion prediction model and this title gave better prediction result and accuracy compare to title1 and title2.
- 4) In title 4 we are using combination OPENCV and CNN to build deep learning to predict human faces emotion. In this title we are reading dataset images using OPENCV and then training with CNN using SOFTMAX function and then calculate loss/error rate. This module give better prediction accuracy and less error rate with good emotion detection result compare to above 3 algorithms.

Below screen showing implementation of all 4 titles

11. FUTURE ENHANCEMENT

Future Enhancement is being planned to further analyze and enhance the

protocol to successfully working condition for analysis of emotion in everybody faces.They are happy or sad.

12. CONCLUSION

The proposed work is designed to develop a real time system to detect, recognize and classify human face. The classified expressions are represented in seven states as shown in above results. The softwares used to test the functionality are Anaconda and python 3.5. For Face detection viola jones and haar cascade algorithm was used. Similarly KDEF dataset and VGG 16 were used with convolution neural network model for face recognition and classification. The dlib and other libraries works in support with python programming. The performance measures are validated with the CNN model designed with an accuracy of 88%. However the results prove that the network architecture designed has better advancements than existing algorithms. This application is widely used in many areas such as education, industry, medical and electronics etc. VGG 16 and KDEF dataset is built to achieve face classification and recognition. The above

seven facial expressions shows different conditions of a person. Some of the applications directly related are Autism which can be helpful to interpret expressions of a person or child. The emotions of a student E-learning techniques could also be evaluated easily through the proposed work.

3. BIBLIOGRAPHY

- [1] Ken Nozaki, Hisao Ishibuchi 1996. Adaptive fuzzy rule based Classification systems, IEEE transactions on Fuzzy systems, Vol.4, No.3.
- [2] Farid Ghareh Mohammadi, Mohammed Saniee Abadeh 2014. Image Steganalysis using abee colony based feature selection Algorithm. Engineering Applications of AI. Science Direct, pp. 35-43.
- [3] Liu, F L C Y, 2015. Improving Steganalysis by Fusing SVM Classifiers for JPEG Images. IEEE, pp. 185-190.
- [4] Chhikara, M K, 2016. Significance of feature selection for image Steganalysis. IEEE.
- [5] Bin Li. 2011 A survey on Image steganography and Steganalysis, Journal of Information Hiding and Multimedia signal Processing, Ubiquitous International, vol2 No.3 pp. 142-172.
- [6] Cho, B H, 2013. Block-based image Steganalysis: Algorithm and performance evaluation. Elsevier, 24(7), pp. 846-856.
- [7] Alex Krizhevsky, I. S. a. G. E. H., 2012. Imagenet classification with deep convolutional neural networks. ACM DL, Volume 1, p. 1097–1105.
- [8] C. Shan, S. G. a. P. W. M., 2009. “Facial expression recognition based on local binary patterns: A comprehensive study. Image and Vision Computing, 27(6), p.

803–816.

[9] Dharma, A. S. A. a. D., April 2018. Novel directional patterns and a Generalized Supervised Dimension Reduction System (GSDRS) for facial emotion recognition. Springer Link, 77(8), p. 9455–9488 .

[10] Hao Wang, S. W. a. B. F., August 2018. Facial expression recognition using iterative fusion of MO-HOG and deep features. The Journal of Supercomputing.

[11] Ji, Y. Z. a. Q., May 2005. Active and Dynamic Information Fusion for Facial Expression Understanding from Image Sequences. IEEE, 27(5), p. 699–714.

[12] Mahdi Jampour, T. M. a. H. B., 2015. Multi-view Facial Expressions Recognition using Local Linear Regression of Sparse Codes. Seggau, Austria, Research Gate

[13] Mundher Al-Shabi, W. P. C. a. T. C., Aug 2019. Facial Expression Recognition Using a Hybrid CNN– SIFT Aggregator. ResearchGate(RG), pp. 139-149.

[14] Viraj Mavani, S. R. a. K. P. M., 2017. Facial Expression Recognition using Visual Saliency and Deep Learning. s.l., IEEE.

[15] CUDA Toolkit 10.1 Update 2 available from
<<https://developer.nvidia.com/cuda-downloads>> [2 September 2019]

[16] KDEF | The Kenya Dryland Education Fund (2019) available from
<<https://kdef.org/>> [2 September 2019].