

CHAPTER-I

1. INTRODUCTION

1.1 ABOUT THE PROJECT

Overview

The facial emotion may lead to understanding the current emotional or mental state of the user. Human beings regularly have different moods and facial expressions changes consequently. Human emotion recognition performs a completely crucial role in social relations. Facial Emotion Recognition is essential to recognize the emotions in many cases like in E-learning, gaming and psychology. E-learning, this can be used to conclude whether a student understands a concept or not

Objectives

The objective of this proposed system is can be used to recognize the facial expressions of human and classify 7 basic emotions. An automated emotion recognition system which examines the expressions through various steps like segmentation, feature extraction and identification of human emotions from image or video. This application inspired by image processing and machine learning algorithms.

Steps involved in this process are like Image Pre-processing, face detection, facial components, feature extraction and classification. This application uses the image captured from the webcam and the obtained image is compared with the trained dataset model that can classify 7 basic emotions such as Happy, Sad, Disgust, Anger, Contempt, Surprise, and Fear by using Convolutional neural network(CNN) algorithm.

1.2 ORGANIZATION PROFILE

INTECH DIGITAL SOLUTIONS

Company Profile.

At Intech digital solutions we go beyond providing software solutions. We work with our client's Solutions and business changes that shape their competitive advantages.

Founded in 2000, Intech is a software and service provider that helps organizations deploy, manage, and support their business-critical software more effectively. Utilizing a combination of proprietary software, services and specialized expertise, Intech digital solutions. helps mid-to-large enterprises, software companies and IT service providers improve consistency, speed, and transparency with service delivery at lower costs.

Intech digital solutions. helps companies avoid many of the delays, costs and risks associated with the distribution and support of software on desktops, servers and remote devices. Our automated solutions include rapid, touch-free deployments, ongoing software upgrades, fixes and security patches, technology asset inventory and tracking, software license optimization, application self-healing and policy management. At Intech digital solutions, we go beyond providing software solutions. We work with our clients' Solutions and business processes that shape their competitive advantages

About The People

As a team we have the prowess to have a clear vision and realize it too. As a statistical evaluation, the team has more than 40,000 hours of expertise in providing real-time solutions in the fields of Embedded Systems, Control systems, Micro-Controllers, c Based Interfacing, Programmable Logic Controller, VLSI Design And Implementation, Networking With C, ++, java, client Server Solutions in Java,(J2EE\J2ME\J2SE\EJB),VB & VC++, Oracle and operating system concepts with LINUX.

Website: www.intechdigitalads.com

1. 3. SYSTEM SPECIFICATION

1.3.1 HARDWARE SPECIFICATION

| | | |
|-----------------|---|--------------------|
| Processor | : | Intel Duel core |
| RAM | : | 4 GB |
| Hard Disk Drive | : | 500 GB |
| Printer | : | HP Ink Jet |
| Keyboard | : | Samsung |
| Mouse | : | Logitech (Optical) |

1.3.2 SOFTWARE SPECIFICATION

| | | |
|------------------|---|------------|
| IDE | : | Pycharm |
| Operating System | : | Windows 10 |
| Front End | : | Python |

1.3.3 SOFTWARE DESCRIPTION

INTRODUCTION TO PYTHON PROGRAMMING LANGUAGE

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language, including NASA, Google, YouTube, BitTorrent, etc. 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. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications. This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well. For a description of standard objects and modules, see [library-index](#). [Reference-index](#) gives a more formal definition of the language. To write extensions in C or C++, read [extending-index](#) and [c-api-index](#). There are also several books covering Python in depth. This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in [library-index](#).

Methods

Methods on objects are functions attached to the object's class; the syntax `instance.method(argument)` is, for normal methods and functions, syntactic sugar for `Class.method(instance, argument)`. Python methods have an explicit `self` parameter to access instance data, in contrast to the implicit `self` (or `this`) in some other object-oriented programming languages (e.g., C++, Java, Objective-C, or Ruby).

Typing

The standard type hierarchy in Python 3

Python uses duck typing and has typed objects but untyped variable names.

Type constraints are not checked at compile time; rather, operations on an object may fail, signifying that the given object is not of a suitable type. Despite being dynamically-typed, Python is strongly-typed, forbidding operations that are not well-defined (for example, adding a number to a string) rather than silently attempting to make sense of them. Python allows programmers to define their own types using classes, which are most often used for object-oriented programming. New instances of classes are constructed by calling the class (for example, `SpamClass()` or `EggsClass()`), and the classes are instances of the metaclass type (itself an instance of itself), allowing metaprogramming and reflection. Before version 3.0, Python had two kinds of classes: old-style and new-style.[96] The syntax of both styles is the same, the difference being whether the class object is inherited from, directly or indirectly (all new-style classes inherit from `object` and are instances of `type`). In versions of Python 2 from Python 2.2 onwards, both kinds of classes can be used. Old-style classes were eliminated in Python 3.0. The long-term plan is to support gradual typing and from Python 3.5, the syntax of the language allows specifying static types but they are not checked in the default implementation, CPython. An experimental optional static type checker named `mypy` supports compile-time type checking.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python Programming Characteristics

- It provides rich data types and easier to read syntax than any other programming languages
- It is a platform-independent scripted language with full access to operating system API's
- Compared to other programming languages, it allows more run-time flexibility
- It includes the basic text manipulation facilities of Perl and Awk
- A module in Python may have one or more classes and free functions
- Libraries in Python are cross-platform compatible with Linux, Macintosh, and Windows

- For building large applications, Python can be compiled to byte-code
- Python supports functional and structured programming as well as OOP
- It supports interactive mode that allows interacting Testing and debugging of snippets of code
- In Python, since there is no compilation step, editing, debugging, and testing are fast.

Applications of Python Programming Language

- Program video games
 - Build Artificial Intelligence algorithms
 - Program various scientific programs such as statistical models
-

Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes Ridiculously fast

Django was designed to help developers take applications from concept to completion as quickly as possible.

Reassuringly secure

Django takes security seriously and helps developers avoid many common security mistakes.

Exceedingly scalable

Some of the busiest sites on the Web leverage Django's ability to quickly and flexibly scale.

Django framework development process

To build such a complicated web system, we need three major parts for each component: database, user interface and the functions to interact in between. Django framework provides sufficient functionalities to implement these three parts.

Corresponding to database, user interface and functions in between, Django has model, template and view components to deal with each part respectively. Django's model

component helps programmer to define and maintain tables in the database, while its template component helps to write html files using a combination of both html syntax and Django syntax. For those functions in between, Django provides a view component which reads the input from user interface and makes corresponding changes in the database.

CHAPTER-II

2. SYSTEM STUDY

System analysis will be performed to determine if it is flexible to design information based on policies and plans of organization and on user requirements and to eliminate the weakness of present system. This chapter discusses the existing system, proposed system and highlights of the system requirements.

2.1 EXISTING SYSTEM

Facial emotional based emoji prediction and identification has become a wide range area of research, due to increase in number of research field. Several works have been done in the area of Facial emotional prediction systems and many are still trying to provide improvements for a better model for retrieving Facial emotional detection accuracy. Existing system SVM Support Vector Machine used to detect Facial emotional in images. But the problem is this will leads time consuming process need to apply more training data. This increase the detection time this algorithm fails to achieve Facial emotional detection accuracy

2.1.1 DRAWBACKS OF EXISTING SYSTEM

- Limited accuracy and robustness
- Simplistic machine learning algorithms may struggle to capture nuances of human emotions effectively.
- Challenges in accurately detecting emotions
- Suboptimal performance in classifying complex emotional states.
- Need for enhancement with advanced deep learning approaches for improved accuracy and generalization.
- Requirement for real-world applicability in recognizing and classifying facial expressions effectively.

2.2 PROPOSED SYSTEM

The drawbacks, which are faced during existing system, can be eradicated by using this application. The main objective of the proposed system is to provide a user-friendly application. This application inspired by image processing and machine learning algorithms. Steps involved in this process are like Image Pre-processing, face detection, facial components, feature extraction and classification. This application uses the image captured from the webcam and the obtained image is compared with the trained dataset model that can classify 7 basic emoji such as Happy, Sad, Disgust, Anger, Contempt, Surprise, and Fear by using Convolutional neural network(CNN) algorithm.

2.2.1 BENEFITS OF PROPOSED SYSTEM

- **Enhanced Accuracy:** Utilizing deep learning techniques like convolutional neural networks (CNNs) significantly improves the system's ability to accurately detect and classify facial expressions.
- **Increased Robustness:** Deep learning models are more resilient to variations in lighting, facial orientations, and demographics, ensuring consistent performance across diverse scenarios.
- **Improved Generalization:** The system can effectively recognize and classify a wide range of facial expressions, including subtle nuances, leading to better performance in different contexts.
- **Real-time Processing:** Capable of processing live video streams from webcams, enabling real-time emotion recognition in interactive applications.
- **Scalability and Adaptability:** The system's flexibility allows for easy integration of additional emotions or customization for specific use cases, ensuring relevance and effectiveness over time.
- **Automated Feature Extraction:** Deep learning models automatically learn relevant features from raw image data, eliminating the need for manual feature engineering and reducing bias.

CHAPTER-III

3. SYSTEM DESIGN AND DEVELOPMENT

OBJECTIVES

Software testing is a critical element of software quality assurance that represents the ultimate review of specifications, design and coding. The user tests the developed system and changes are made according to their needs. The testing phase involves the testing of developed system using various kinds of data. It involves user training, system testing and successful running of the developed system.

The changes are made according to their needs. The testing phase involves the testing of the developed system using various kinds of data. While testing, errors are noted and corrections are made system testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. The candidate system is subject to a variety of test: stress recovery, and security and usability tests.

3.1 INPUT DESIGN

Input Design converts the user-oriented inputs to computer-based formats. Inaccurate input data are the most common cause of errors in data processing. Error data entered by the data operator can be controlled by the input design. The goal of designing input is to make the data entry easy, logical and as free from errors as much as possible.

The proposed system is completely menu-driven. It is a powerful tool for interactive design. It helps the user comprehend the range of alternatives available and also prevents them from making an invalid selection. All entry screens are interactive in nature. It has been designed taking into account all the constraints of the end-user.

Some other features included are:

- The form title clearly states the purpose of the form
- Adequate space is given for data entry Data Validation is done for eliminating duplicate entries

3.2 OUTPUT DESIGN

Outputs are the most important and direct source of information to the customer and management. Intelligent output design will improve the system's relationship with the user and help in decision making. Outputs are used to make permanent hard copy of the results for later consultation. The output generated by the system is often regarded as the criteria for evaluating the performance of the system. The output design was based on the following factors.

- Usefulness determining the various outputs to be printed to the system user.
- Differentiating between the outputs to be displayed and those to be printed.
- The format for the presentation of the output.

For the proposed system, it is necessary that the output should be compatible with the existing manual reports. The outputs have been formatted with this consideration in mind. The outputs are obtained after all the phase, from the system can be displayed or can be produced in the hard copy. The hard copy is highly preferred since it can be used by the controller section for future reference and it can be used for maintaining the record.

Frame Extraction

The frame extraction module segments the video captured into different frames. It extracts every frame from video using extraction function. This extraction usually performs with some specific time interval.

Emotion Classification

Each and every Test image analysis using neural networks will predict the emotion effectively using our model. Finally test image is going to match with model it takes to perform detection on an image. Training is not done in a single step. There are different models for doing different parts which make the training process . Finally This application uses the image captured from the webcam and the obtained image is compared with the trained dataset model that can classify 7 basic emotions such as Happy, Sad, Disgust, Anger, Contempt, Surprise, and Fear by using Convolutional neural network(CNN) algorithm.

CHAPTER-IV

4. TESTING AND IMPLEMENTATION

4.1 SYSTEM TESTING

Objectives of Testing

Software testing is a critical element of software quality assurance that represents the ultimate review of specifications, design and coding. The user tests the developed system and changes are made according to their needs. The testing phase involves the testing of developed system using various kinds of data. It involves user training, system testing and successful running of the developed system.

The changes are made according to their needs. The testing phase involves the testing of the developed system using various kinds of data. While testing, errors are noted and corrections are made system testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. The candidate system is subject to a variety of test: stress recovery, and security and usability tests.

Test Plan

Testing is the process of executing a program with the intent of finding any errors. A good test of course has the high probability of finding a yet undiscovered error. A successful testing is the one that uncovers a yet undiscovered error.

A test is vital to the success of the system; system test makes a logical assumption that if all parts of the system are correct, then goal will be successfully achieved. The candidate system is subjected to a verity of tests online like responsiveness, its value, stress and security. A series of tests are performed before the system is ready for user acceptance testing.

TESTING METHODS

The different types of testing are:-

- Unit Testing
- Integration Testing
- Validation Testing
- Output Testing

UNIT TESTING

Unit testing focuses verification efforts on the smallest unit of software design, the module. This is also known as “Module Testing” The modules are tested separately this testing is carried out during programming stage itself. In this step each module is found to be working satisfaction as regard to the expected output from the module.

INTEGRATION TESTING

Integration testing focuses on the design and construction of the software architecture. Data can be lost across an interface, one module can have adverse effect on another sub functions and show on. Thus integration testing is a systematic technique for constructing test to uncover errors associated with in the interface. In this project, all the modules are companied and then the entire program is tested as a whole.

VALIDATION TESTING

Validation testing is the requirement established as a part of software requirement analysis is validated against the software that has been constructed. This test provides the final assurance whether the software needs all functional, behavioral and performance requirements

Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactory.

OUTPUT TESTING

After performing the validation testing, the next step is the output testing of the proposed system, since no system could be useful if it does not produce required output in the specific format. Tested asking the users about the format required by them, the output is considered into two ways: one is on the screen and the other is printed format.

The output format on the screen is found to be correct as the format designed according to the user needs, for the hard copy also, the output comes as specified by the user. Hence output testing does not result in correction in the system.

4.2 IMPLEMENTATION

Implementation is the stage where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system and in giving confidence on the new system for the users that it will work efficiently and effectively. The system can be implemented only after thorough testing is done and if it is found to work according to the specification. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over and an evaluation of change over methods a part from planning. Two major tasks of preparing the implementation are education and training of the users and testing of the system.

The more complex the system being implemented, the more involved will be the systems analysis and design effort required just for implementation. The implementation phase comprises of several activities. The required hardware and software acquisition is carried out. The system may require some software to be developed. This application is implemented in python as front end Mysql as back end.

CHAPTER-V

5. CONCLUSION AND FUTURE ENHANCEMENT

5.1 CONCLUSION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. This proposed application successfully image captured from the webcam and the obtained image is compared with the trained dataset model available and then emotional state of the image will be displayed. Facial Emotion Recognition application is implemented using Convolution Neural Network (CNN). System has been tested with the dataset which contains the various emotions of the humans.

5.2 SCOPE FOR FUTURE ENHANCEMENT

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. In future we can develop this application with advanced features and we can implement voice based emotion detection.

CHAPTER-VI

6. BIBLIOGRAPHY

Books Referred:

1. Python Tricks: A Buffet of Awesome Python Features by Dan Bader Python
2. Tricks: A Buffet of Awesome Python Features by Dan Bader Python Tricks:
3. A Buffet of Awesome Python Features by Dan Bader EPUB Mark Lutz ,”
4. core Python language with this hands-on book.”,2013
5. <https://creatorpdf.com/B0785Q7GSY> <https://creatorpdf.com/B0785Q7GSY>
- 6.

WEBSITES

1. <https://www.w3schools.com/python>
2. <https://www.w3schools.com/python>
3. www.tutorialspoint.com

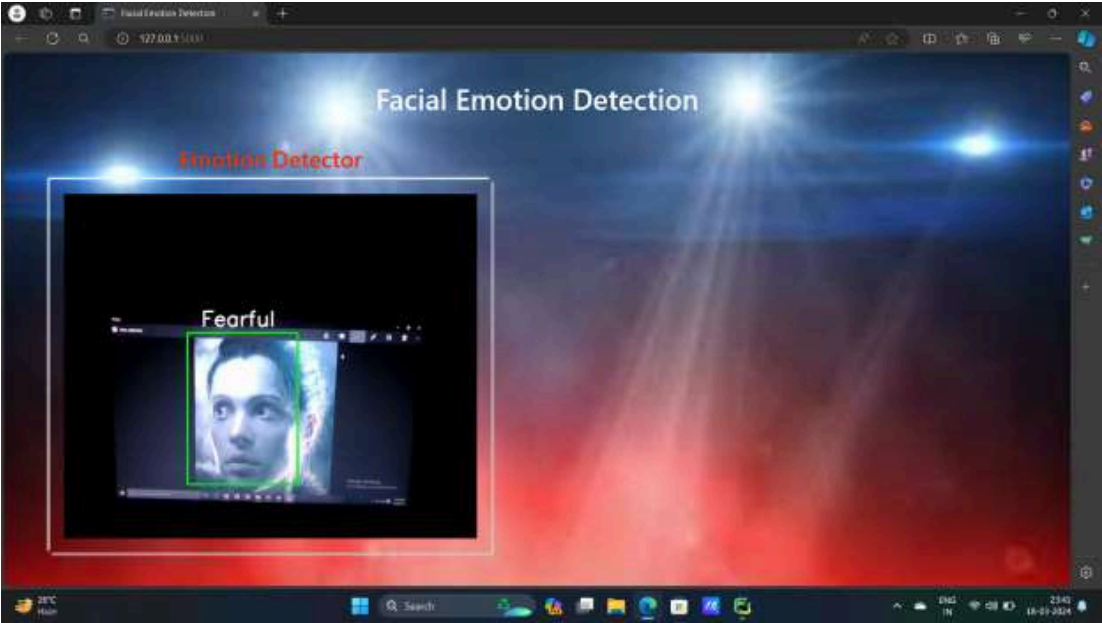
C. SAMPLE INPUT



SAD



FEAR



SURPRISE

