

MULTIMEDIA RETRIEVAL USING EMOJI PREDICTION, ANTICIPATION, AND RETRIEVAL

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

V.VENKATA SUBBARAYUDU

(Reg. No. 37110805)

V. GIRISH GUPTHA

(Reg. No. 37110812)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600 119**

MARCH - 2020



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **Vaygundala GIRISH GUPTHA (Reg. No. 37111812)** and **MRIGANKA VANAMALA VENKATA SUBBARAYUDU (Reg. No. 37110805)** who carried out the project entitled **"ONLINE CRIME REPORTING SYSTEM"** under my supervision from **DECEMBER 2020 to March 2021**.

Internal Guide

Dr. M. Saravanan, M.E., Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **VANAMALA VENKATA SUBBARAYUDU** hereby declare that the Project Report entitled “**MULTIMEDIA RETRIEVAL USING EMOJI PREDICTION, ANTICIPATION, AND RETRIEVAL**” is done by me under the guidance of **Dr. M. Saravanan, M.E., Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D.,** and **Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. M. Saravanan, M.E., Ph.D., Assistant Professor**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Over the past decade, emoji have emerged as a new and widespread form of digital communication, spanning diverse social networks and spoken languages. We propose treating these ideograms as a new modality in their own right, distinct in their semantic structure from both the text in which they are often embedded as well as the images which they resemble. As a new modality, emoji present rich novel possibilities for representation and interaction. In this paper, we explore the challenges that arise naturally from considering the emoji modality through the lens of multimedia research, specifically the ways in which emoji can be related to other common modalities such as text and images. To do so, we first present a large-scale data set of real-world emoji usage collected from Twitter. This data set contains examples of both textemoji and image-emoji relationships within tweets. We present baseline results on the challenge of predicting emoji from both text and images, using state-of-the-art neural networks. Further, we offer a first consideration into the problem of how to account for new, unseen emoji—a relevant issue as the emoji vocabulary continues to expand on a yearly basis. Finally, we present results for multimedia retrieval using emoji as queries.

TABLE OF CONTENTS

ABSTRACT	v
LIST OF FIGURES	viii

CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	1
	1.1 TECHNIQUES USED	3
2.	AIM AND SCOPE	5
	2.1 AIM OF PROJECT	5
	2.2 OBJECTIVES	6
	2.3 SCOPES	6
3.	METHODOLOGY	9
	3.1 METHODS USED	9
	3.2 MODULES	22
	3.3 ARCHITECHURE	24
	3.4 SEQUENCE DIAGRAM	24
	3.5 USECASE	25
	3.6 ACTIVITY	26
	3.7 DESIGN AND IMPLEMENTATION CONSTRAINTS	27
	3.8 NON-FUNCTIONAL REQUIREMENTS	28
	3.8 REQUIREMENT SPECIFICATIONS	30
4.	RESULTS & DISCUSSION	31
	4.1 CODING	31
	4.2 RESULTS	33
	4.3 SUMMARY	36

CONCLUSION & FUTUREWORK	38
REFERENCES	40
APPENDIX	
A. PAPER ACCEPTANCE MAIL	-
B. PLAGIARISM REPORT	47
C. JOURNAL PAPER	48
D. SOURCE CODE	59

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.1	BLOCK DIAGRAM	24
4.2	ACTIVITY DIAGRAM	28
5.1	HOME PAGE	34
5.2	NEWS FEED	35
5.3	LOG-IN PAGE	36
5.4	REGISTRATION FORM	37
5.5	CRIME REPORT FORM	38
5.6	MISSING VEHICLE FORM	39
5.7	ALERT PAGE	40
5.8	APPLICATION FORM	41
5.9	DASHBOARD	42

CHAPTER 1

INTRODUCTION

INTRODUCTION

Emoji, small ideograms depicting objects, people, and scenes, have exploded in popularity. They are now available on all major mobile phone platforms and social media websites, as well as many other places. According to the Oxford English Dictionary, the term emoji is a Japanese coinage meaning ‘pictogram’, created by combining e (picture) with moji (letter or character). Emoji as we know them were first introduced as a set of 176 pictogram available to users of mobile phones. The available range of ideograms has expanded greatly over the previous years, with 1,144 single emoji characters defined in Unicode 10.0 and many more defined through combinations of two or more emoji characters.

Nowadays, emoji have emerged as a new and widespread form of digital communication, spanning diverse social networks and spoken languages. We propose treating these ideograms as a new modality in their own right, distinct in their semantic structure from both the text in which they are often embedded as well as the images which they resemble. As a new modality, emoji present rich novel possibilities for representation and interaction.

Emojis or avatars are ways to indicate nonverbal cues. These cues have become an essential part of online chatting, product review, brand emotion, and many more. It also leads to increasing data science research dedicated to emoji-driven storytelling. With advancements in computer vision and deep learning, it is now possible to detect human emotions from images. In this deep learning project, we will classify human facial expressions to filter and map corresponding emojis or avatars.

The word emoji has originated from the Japanese language with the letter “e” (meaning picture) and “moji” (meaning character). Emojis are considered to be the 21st-century transformation of emoticons. They were initially developed in 1999 as a 12*12 pixel grid by Shigetaka Kurita as a part of a Japanese team working on an early version of a mobile internet platform. But present emoji images can be scaled to unlimited resolution with the help of vector graphics¹. The utilization of emojis in social media has visually perceived a rapid increase over the last few years as they have become a way to integrate

a tone and non-verbal context to daily communication. According to the latest statistics released by Twitter, tweets with images links get 2x the engagement rate of those without² .

The study of emoji usage in social media platforms has been one of the exciting research elds, Instagram has reported that the emoji usage on their website has seen an increase of 10% in a single month after the release of Emoji keyboard on iOS mobiles in 2011, and it also reported that more than 50% of all captions and comments include an emoji or two³ . Hence considering the extensive usage of images on their platform, Instagram has started analyzing proles of users who use emojis, and they reported that 53% of users use emojis in their posts⁴ . Figure 1 shows the usage of different emojis in Instagram. Images being more expressive than text, sending a message utilizing one diminutive emoji is more efcacious than a text message is the other important feature which increased emoji usage in social media. According to the latest statistics released by Emojipedia, there are 2666 emojis which are further divided into different subcategories. Earlier in 2016 most search engines namely Google and Bing supported emoji search⁵ , but in 2017 Twitter has also enabled.

Due to limited linguistic concepts, earlier manually annotated patterns were used as external knowledge concepts to enhance NLP systems. With the advent of advanced knowledge base construction, large amounts of semantic and syntactic information are made available which helped researchers enhance the performance of most NLP systems namely word embedding models and many other prediction and classication tasks. In recent years several noteworthy large, cross-domain knowledge graphs have been developed. Researchers have also worked on incrementally populating knowledge graph from unstructured data which encompasses problems of extraction, cleaning, and integration. This research in the development of advanced knowledge graphs has enabled many researchers to leverage external domain knowledge in Natural Language Processing (NLP) systems to improve machine understanding. External knowledge has also been effective to improve the accuracies of emoji understanding tasks including but not limited to emoji similarity, emoji sense disambiguation . In this paper, we investigate whether external knowledge from EmojiNet can enhance the performance of emoji recommendation task in the context of images.

1.1 TECHNIQUES USED:

The techniques of the visual technology for the conversions of the expressions into the art of facial graphical style conversion are making a great contribution and advancement in the digital graphics designing. The initial tools used from the people associated with this field were using the histograms as the mode of representing the pyramids of the facial gradients (Duncan et al., n.d., p. 1). In 2015, Emotions in the Wild (EmotiW 2015) contest, for descriptions, such as the emotions, utilized static types of images that deeply experienced the effect of the convolutional neural networks (CNNs) (Duncan et al., n.d., p. 2). Their accuracy was about 62%, but recent progress by Levi & Hassner has shown substantial growth using a CNN in facial emotion recognition .

Communication is an important part of everyday life. Verbal or non-verbal communication allows one to engage in conversations. Today is the era of communication technologies. The internet and other communication devices have made it possible to engage in the fast, dynamic, and affective communication. The emojis are being used for the visual depictions of human emotions. This paper presents the visual expressions of humans using emojis. Therefore, the efforts have always been made to enhance the experience of communication in human beings. The present study has three major objectives to be accomplished. These three major objectives are to investigate emotional recognition using facial expression by emojis. Moreover, another objective is to develop the parameters of measuring the facial expression by emoji texting, as well as to understand the facial emotion recognition.

Therefore, the present study in that realm focuses on the use of the emojis as an important technique that can be used in the field. Hence, the present research will be designed to assess human emotions and their visual depiction in the form of six emotions. These emotions include neutral, fear, anger, happy, sad, and surprise emotions. These are important human expressions and, thus, are quite important. The channels show the convolution of 3D delineation models. These models are prone to the use of layers, which records clamor in process. Moreover, these

applications will be developed in the study to assess its impacts on the emojis.

This research study presents how the facial looks and discovery for the facial communication is delivered. Therefore, it can be said that the facial expression is important for the review of the investigation of the facial activity coding framework for the sake of the communication that is being watched. Moreover, the facial expressions are critical as they help incorporate and communicate feelings to discover the activity units. Therefore, this study develops an application that impacts communication channels, which are designed for human conduct, as well as to identity someone.

Therefore, the use of the emoji provides the feelings that have a profound impact that has to be delivered, and with the expression and the facial recognition, which is being conducted, our general identity is restored through the emoji. These are important things that must be considered and investigated in current research fields. The emojis will be developed, and the human expressions will be tested through the use of such emojis.

CHAPTER 2

AIM AND SCOPE

2.1 AIM OF PROJECT

The main aim of this project is to predict emoji's according to the input such as image, emoji and text using convolution neural networks shotly called as CNN Algorithm. This program predicts the emojis for text input and image inputs and cetaes a HTML document as output. The main idea behind doing this project is that over the past decade, emoji have emerged as a new and widespread form of digital communication, spanning diverse social networks and spoken languages. We propose treating these ideograms as a new modality in their own right, distinct in their semantic structure from both the text in which they are often embedded as well as the images which they resemble. As a new modality, emoji present rich novel possibilities for representation and interaction.

Previously in the existing system there were **to** facilitate further research on emoji, we propose three emoji challenge problems and present state-of-the- art neural network baselines for them, as well as a dataset for evaluation. Previously cannot find a Emoji correctly and not accuracy. To found a unrelated emoji are displayed. So, we will move to the proposed model.

The primary aim of this proposed system Multimedia retrieval using emoji Prediction, Anticipation, and Retrieval is to predict the emoji's according to the input such as image, emoji and text using convolution neural networks shotly called as CNN Algorithm. This program predicts the emojis for text input and image inputs and cetaes a HTML document as output. This makes

2.2 OBJECTIVES

There are certain objectives of the thesis necessary to understand the core aspects for the identification of research outcomes which are listed below:

- To investigate emotional recognition using facial expression by emoji in real time
 - To develop the parameters of measuring the facial expression by emoji texting
 - To understand the facial emotion recognition in real time

The stated objectives of this thesis highlighted the reasons behinds the facial recognition emotions in the real time. The main goal of this project is to implement the recognition of the facial emotion for real time. The implementation of an application process is to identify the facial emotion recognition reasons with highlighted emoji indicator for the identification of six expressions. Such expressions are neutral, fear, anger, happy, sad, and surprise. Furthermore, facial expressions are investigated to realize the real impact of the emoji. The expressions are the actual predictor of human behavior. If expressions are good, then it means the person is in a good mood or has a joyful personality. If the person's expressions show anger or sorrow that means the person is not feeling well or his or her personality does not have joyful traits. The expressions are related with the person's behavior and personality. Understanding expressions are important because they give a lot of information regarding behavior and moods of people through expressions; one can know what is going inside the person's mind and how it can be handled

2.3 SCOPE

This model will help to show emoji very accurately and efficiently according to face expression of an individual using Convolutional Neural Network algorithm which is shortly known as CNN and it is known for image processing and sentimental analysis.

EMOJI, small ideograms depicting objects, people, and scenes, have

exploded in popularity. They are now available on all major mobile phone platforms and social media websites, as well as many other places. According to the Oxford English Dictionary, the term *memoji* is a Japanese coinage meaning ‘pictogram’, created by combining *e* (picture) with *moji* (letter or character). Emoji as we know them were first introduced as a set of 176 pictogram available to users of Japanese mobile phones. The available range of ideograms has expanded greatly over the previous years, with 1,144 single emoji characters defined in Unicode 10.0 and many more defined through combinations of two or more emoji characters. In this paper, we approach emoji as a modality related to, but not contained within, text and images.

We investigate the properties and challenges of relating these modalities to emoji, as well as the multimedia retrieval opportunities that emoji present. The identification and benchmarking of novel modalities has a rich history in the multimedia community. When new modalities are identified, it is important to make first attempts to understand their relationship with already established information channels. One way in which to do this is to explore the crossmodal relationships between the modality and other modalities. When Lee et al. [19] identified nonverbal head nods as an information-rich and overlooked modality, they sought to provide understanding through prediction of them based on semantic understanding of the accompanying conversation transcript. Like emoji, new modalities are sometimes the result of a newly developed technology, as with 3D models [15] or the growth of microblogging [2]. Though ideograms are ancient, emoji are a modern technological evolution of that ancient idea.

The march of technology sometimes facilitates new looks at old problems, such as the use of infrared imagery for facial recognition instead of natural images [43]. Often, the presentation of new tasks as research challenges can accelerate research progress, as it did with acoustic scenes [39] and video concepts [38]. We look to this history of multimedia challenge problems and identify emoji as an emerging modality worthy of a similar treatment. To facilitate further research on emoji, we propose three emoji challenge problems and present state-of-the-art neural network baselines for them, as well as a dataset for evaluation.

Despite their prevalence, research into emoji remains limited. The majority of prior research concerning emoji has focused on descriptive analysis, such as identifying how patterns of emoji usage shift among different demographics or has used them as a signal to indicate the emotional affect of accompanying media. The focus on sentiment is likely a result of there being a number of “face emoji” (e.g.) which are designed to exhibit a particular emotion or reaction.

These face emoji are by far the most visible emoji and among the most widely used [33], but the focus on them ignores the hundreds of other emoji which are worthy of study in their own right. Beyond these face emoji, the full set of emoji also contains a wide range of other objects, such as foods (), signs (), and scenes () which may lack a strong sentimental signal. Recently, Apple has introduced Animoji which allow users to animate select emoji with facial expressions, further broadening their range

CHAPTER 3

METHODOLOGY

This project uses Convolutional Neural Network (CNN) and Python interpreter to execute the code and the data sets or images collected from the website named kaagle which is well known for data collection. The data set of emojis are collected from kaggle website which is well known for data collection. The collected data sets consists of lot of human faces with different emotions and emoji's pictures and both data sets are trained by using convolutional neural networks algorithm. Detailed explanation about software, hardware requirements and envirolment needed to run program and algorithm used to predict emojis are given below.

3.1 METHODS USED:

The strategies of the visual innovation for the transformations of the articulations into the craft of facial graphical style change are making an incredible commitment and headway in the advanced illustrations planning. The underlying apparatuses utilized from individuals related with this field were utilizing the histograms as the method of addressing the pyramids of the facial angles (Duncan et al., n.d., p. 1). In 2015, Emotions in the Wild (Emotion 2015) challenge, for depictions, like the feelings, used static sorts of pictures that profoundly encountered the impact of the convolutional neural organizations (CNNs) (Duncan et al., n.d., p. 2). Their exactness was about 62%, yet late advancement by Levi and Hassner has shown generous development utilizing a CNN in facial feeling acknowledgment .

The process of the recognitions for different types of the human emotions reveal two relevant challenges, or the problems. First, the availability of limited data used analytically for the training, or the procedures for orientation for the CNN. Second, the variation affected by an illumination, which usually appears in the local emotional patterns using the techniques for binary invariants would part of the set of data. The use of the 3D technology in manipulating the existing models of the emotions graphics are usually linked with the expression of the face.

The web face from the challenge through the technology as emoji we could have the level of accuracy by five percent while the improvements from the previous results would yield to only 10-16% (Duncan et al., n.d., p. 2).

The use of the graphical emotional neural tool like the VGG-S would be necessary as the initiating factor of executing the novel facial representation model. Using the sources of database that reveals the advanced version of the CNN is implementing a theory of game as the moving average would be taken. The emotions must be identified or detected in manner that would demonstrate the stream of images or videos while transforming into the input level for graphical streams. The descriptors feature for the LTB would use the database illumination guidelines for detecting the presence of invariants portions of the images (Russell, 1994, p. 109).

The filters, which are common, practice in this discussion of the convolution 3D illustration of the layers that would record the noise of the sounds from the background while emotions are transformed. The visualization process of the emotions that converts the database of the recognition of the expression the person is making with theory facial gestures are clearly optimization element (Sashikar, Murali, & Henderson, 2015, p. 2).

The set of the data, such as the web face uses the features of this graphical portray of emotions in facial translating images as if one of CASIA is dependent on the availability of the resources (Duncan et al., n.d., p. 2). The usual approach in this expression transmission through graphical designing projects or the improvements considerations would link up with the static facial expressions in the wild (SFEW) dataset or the tool of statistical techniques for detecting the expression for the facial wild for revealing the natural effects upon the emotions of the images of people (Duncan et al., n.d., p. 2).

The image, which has the static advancements in the input emotions detection, are illustrating in the applications of the visual posture for the body language. The

face moods that highlight the EEG are complex stature for the methods for detecting the wide ranges of the pyramids of computer database applications. The networks such as the CNN is also integrated with this small-scale training framework or the models that could relate the facial impressions or the emotions for optimization of the facial defection of emotions in the layer based technology, such as VGS (Oberman, Winkielman, & Ramachandran, 2007, p. 176).

3.1.2 Python:

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.

- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way or a functional way.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- 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 backported 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.
- 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.
- It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

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.
- 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.

3.1.3 Introduction Machine learning:

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through learning. In its application across business problems, machine learning is also referred to as predictive analytics.

3.1.3.1 Machine learning tasks:

Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. Semi algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels.

Classification algorithms and regression algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the Boolean values `true` and `false`. Regression algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object.

In unsupervised learning, the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in feature learning. Dimensionality reduction is the process of reducing the number of "features", or inputs, in a set of data.

Active learning algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. Reinforcement learning algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment and are used in autonomous vehicles or in learning to play a game against a human opponent.

Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set of natural language documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable probability density function in density estimation problems. Meta learning algorithms learn their own inductive bias based on previous experience. In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and

social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

3.1.3.2 Types of learning algorithms:

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

Supervised learning:

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

Supervised learning algorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

In the case of semi-supervised learning algorithms, some of the training examples are missing training labels, but they can nevertheless be used to improve the quality of a model. In weakly supervised learning, the training labels are noisy,

limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger effective training sets.

Unsupervised learning:

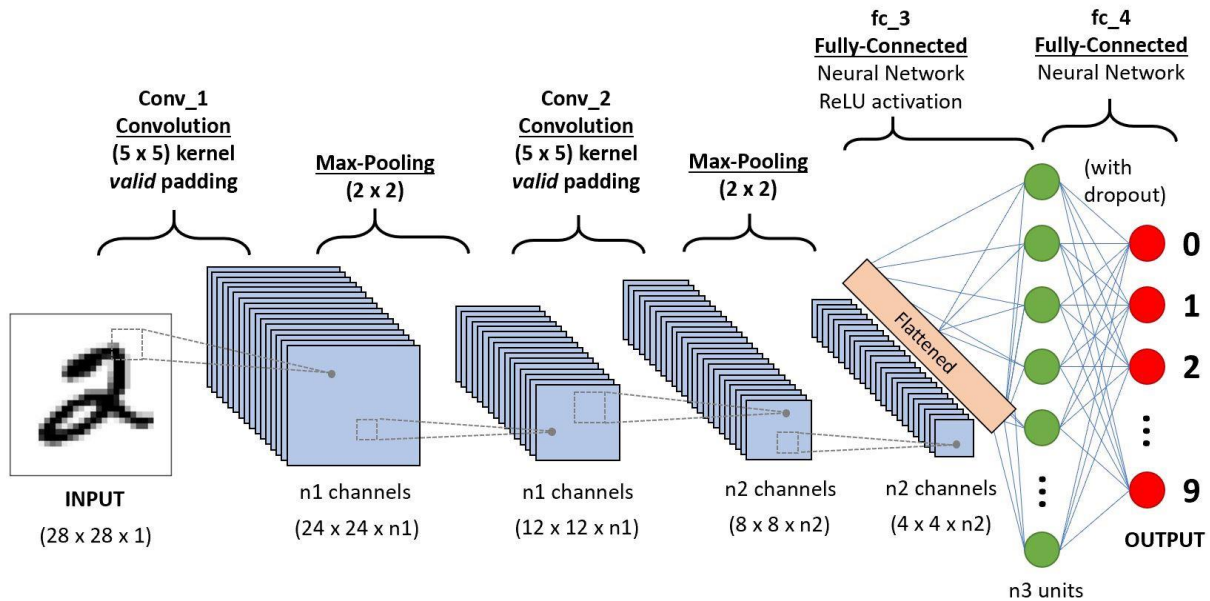
Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, though unsupervised learning encompasses other domains involving summarizing and explaining data features.

Cluster analysis is the assignment of a set of observations into subsets (called *clusters*) so that observations within the same cluster are similar according to one or more pre designated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some *similarity metric* and evaluated, for example, by *internal compactness*, or the similarity between members of the same cluster, and *separation*, the difference between clusters. Other methods are based on *estimated density* and *graph connectivity*.

Semi-supervised learning:

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy

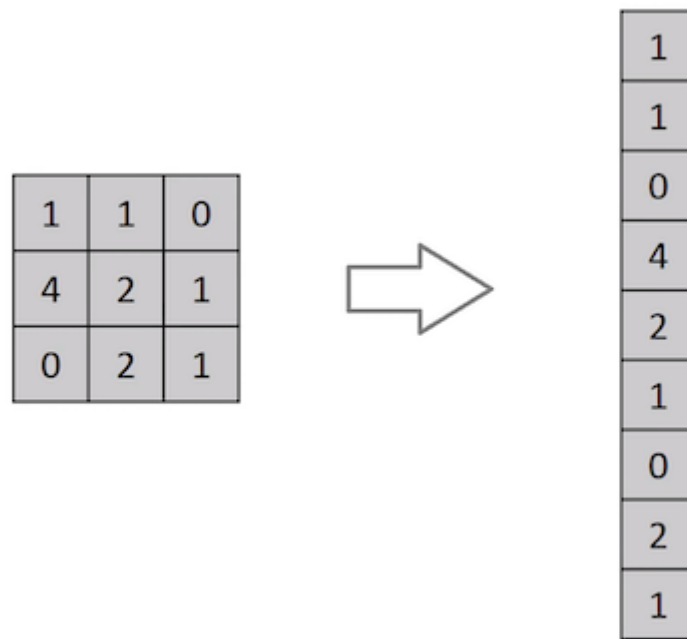
3.1.1 Convolutional Neural Networks:



A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

Why ConvNets over Feed-Forward Neural Nets?



An image is nothing but a matrix of pixel values, right? So why not just flatten the image (e.g. 3x3 image matrix into a 9x1 vector) and feed it to a Multi-Level Perceptron for classification purposes? Uh.. not really.

In cases of extremely basic binary images, the method might show an average precision score while performing prediction of classes but would have little to no accuracy when it comes to complex images having pixel dependencies throughout.

A ConvNet is able to **successfully capture the Spatial and Temporal dependencies** in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

3.1.1.1 Convolutional layers:

A convolutional neural network consists of an input layer, hidden layers and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and

final convolution. In a convolutional neural network, the hidden layers include layers that perform convolutions. Typically this includes a layer that does multiplication or other dot product, and its activation function is commonly ReLU. This is followed by other convolution layers such as pooling layers, fully connected layers and normalization layers.

In a CNN, the input is a tensor with shape (number of images) x (image height) x (image width) x (input channels). After passing through a convolutional layer, the image becomes abstracted to a feature map, with shape (number of images) x (feature map height) x (feature map width) x (feature map channels). A convolutional layer within a neural network should have the following attributes:

- Convolutional filters/kernels defined by a width and height (hyper-parameters).
- The number of input channels and output channels (hyper-parameter).
- The depth of the convolution kernel/filter (the input channels) must equal the number channels (depth) of the input feature map.
- The hyperparameters of the convolution operation, like padding size and stride.

Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its receptive field. Although fully connected feedforward neural networks can be used to learn features and classify data, this architecture is impractical for images. It would require a very high number of neurons, even in a shallow architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10,000 weights for each neuron in the second layer. Instead, convolution reduces the number of free parameters, allowing the network to be deeper. For example, regardless of image size, tiling 5 x 5 region, each with the same shared weights, requires only 25 learnable parameters. Using regularized weights over fewer parameters avoids the vanishing gradient and exploding gradient problems seen during backpropagation in traditional neural networks.

3.1.1.2 Pooling Layers:

Convolutional networks may include local or global pooling layers to streamline the underlying computation. Pooling layers reduce the dimensions of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, typically 2 x 2. Global pooling acts on all the neurons of the convolutional layer. There are two common types of pooling: max and average. Max pooling uses the maximum value of each cluster of neurons at the prior layer, while average pooling instead uses the average value.

3.1.1.3 Fully Connected Layers

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is the same as a traditional multi-layer perceptron neural network (MLP). The flattened matrix goes through a fully connected layer to classify the images.

3.1.1.4 Receptive field:

In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from every neuron of the previous layer. In a convolutional layer, each neuron receives input from only a restricted area of the previous layer called the neuron's receptive field. Typically the area is a square (e.g., 5 by 5 neurons). (So, in a fully connected layer, the receptive field is the entire previous layer.) Thus in each convolutional layer, each neuron takes input from a larger area of pixels in the input image than previous layers. This is due to applying the convolution over and over, which takes into account the value of a pixel and its surrounding pixels. In dilated CNNs, the number of pixels in the receptive field remains invariant but the field gets more sparsely populated as its dimensions grow when assessing the influence of different layers.

3.1.1.5 Weights:

Each neuron in a neural network computes an output value by applying a specific function to the input values coming from the receptive field in the previous layer. The function that is applied to the input values is determined by a vector of weights and a bias (typically real numbers). Learning consists of iteratively adjusting these biases and weights.

The vector of weights and the bias are called filters and represent particular features of the input (e.g., a particular shape). A distinguishing feature of CNNs is that many neurons can share the same filter. This reduces memory footprint because a single bias and a single vector of weights are used across all receptive fields sharing that filter, as opposed to each receptive field having its own bias and vector weighting.

3.2 MODULES

This project consists of mainly three modules named as follows –

- Emoji Prediction
- Emoji Anticipation
- Query-by-Emoji

3.2.1 MODULE EXPLANATION:

➤ Emoji Prediction:

- A system is suggesting some k emoji that the user may wish to include during message composition, and the system should try to ensure that at least some of these emoji are relevant.
- The main aim of these module is to predict the emoji if we given a image as input. It uses convolutional neural network which is also known as CNN algorithm which we discussed above in this documentation.

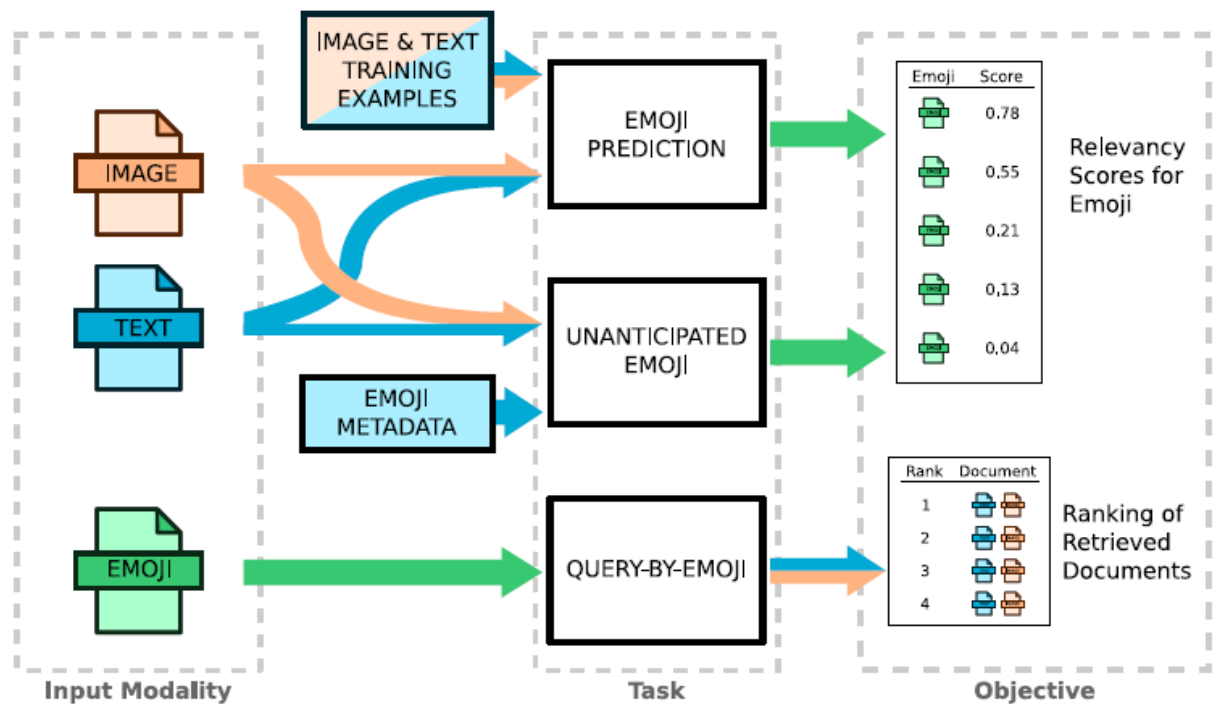
➤ Emoji Anticipation:

- Emoji Anticipation differs from Emoji Prediction in its absence of training data, but the test set and goal of the challenge is shared with Emoji Prediction.

➤ **Query by Emoji:**

- Given a query emoji, the goal is to retrieve a ranked list of documents considered relevant due to their text or image content.
- This module fetches the emoji's related to the image or photo provided as the input, the program emojis by using convolutional neural network.

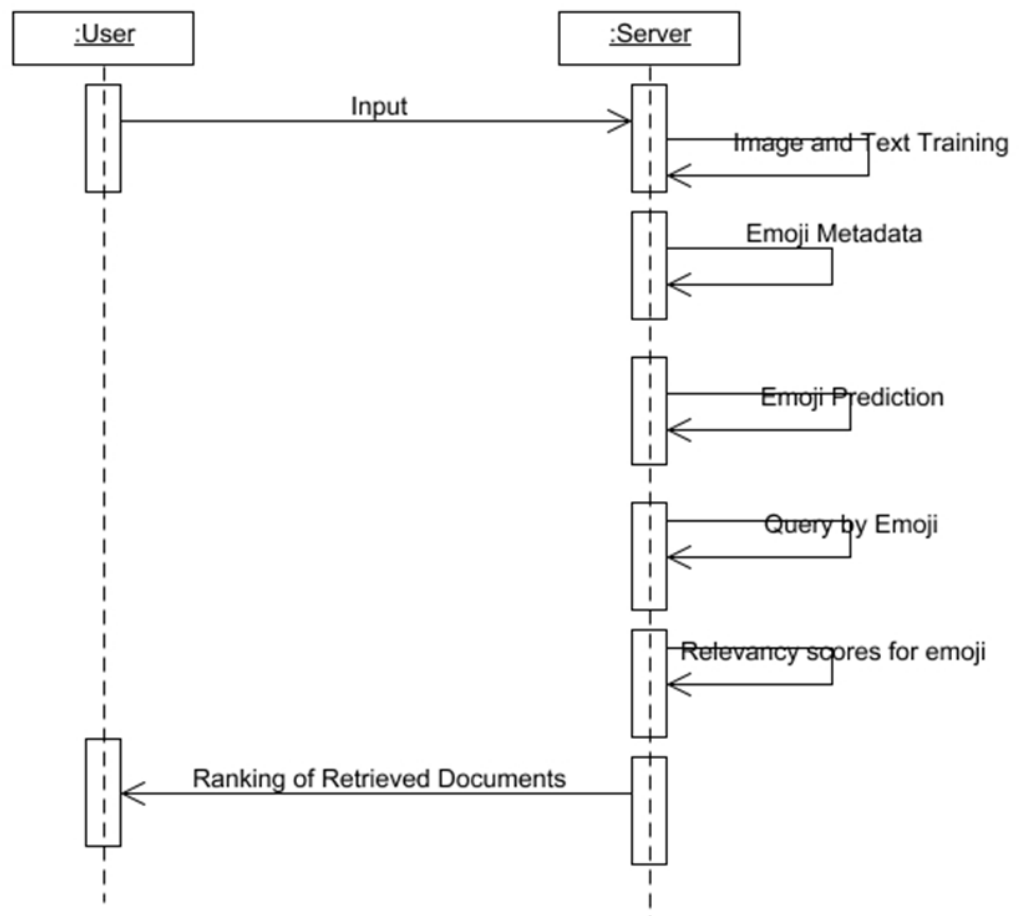
3.3 Architecture Diagrams:



3.4.1: ARCHITECTURE DIAGRAM

3.4 Sequence Diagram:

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



3.3.1 SEQUENCE DIAGRAM

3.5 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

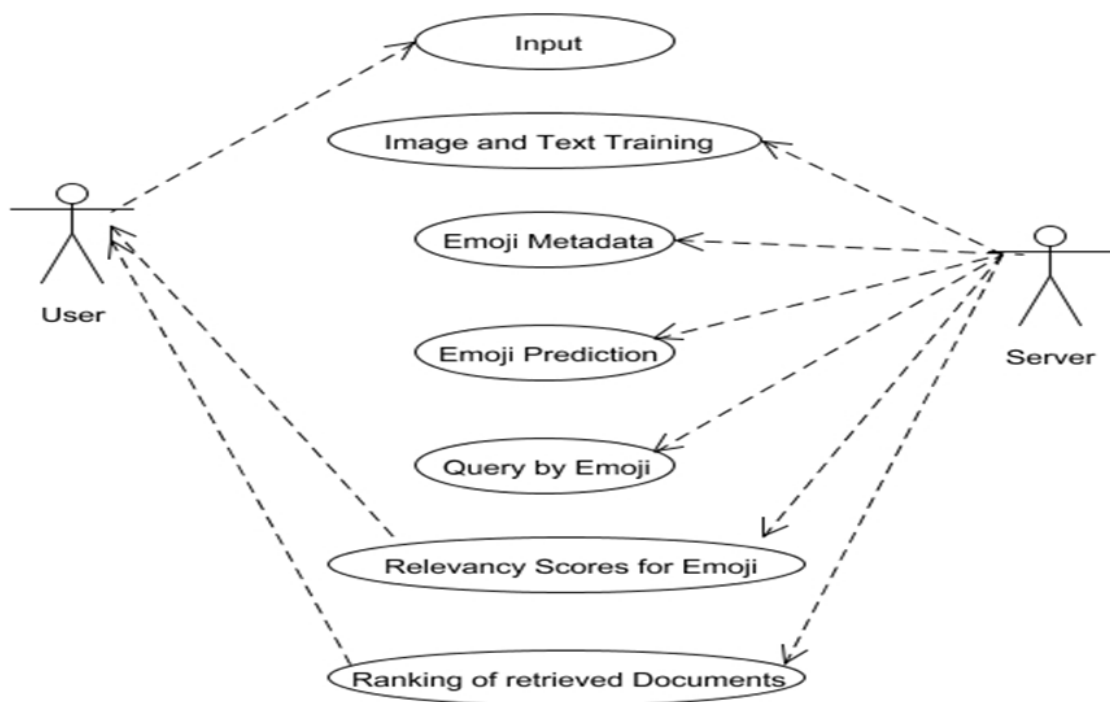
3.5.1 USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.



3.6.2: USE CASE DIAGRAM

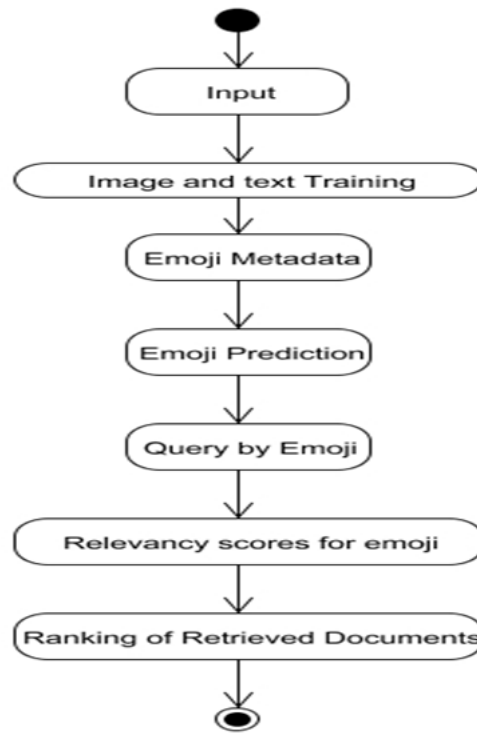
3.6 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.

- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.



3.7: ACTIVITY DIAGRAM

3.7 DESIGN AND IMPLEMENTATION CONSTRAINTS

3.7.1 Constraints in Analysis

- Constraints as Informal Text
- Constraints as Operational Restrictions
- Constraints Integrated in Existing Model Concepts
- Constraints as a Separate Concept
- Constraints Implied by the Model Structure
-

3.7.2 Constraints in Design

- Determination of the Involved Classes
- Determination of the Involved Objects
- Determination of the Involved Actions
- Determination of the Require Clauses

- Global actions and Constraint Realization

3.7.3 Constraints in Implementation

- A hierarchical structuring of relations may result in more classes and a more
- complicated structure to implement. Therefore it is advisable to transform the hierarchical
- relation structure to a simpler structure such as a classical flat one. It is rather
- straightforward to transform the developed hierarchical model into a bipartite, flat model,
- consisting of classes on the one hand and flat relations on the other. Flat relations are
- preferred at the design level for reasons of simplicity and implementation ease. There is
- no identity or functionality associated with a flat relation. A flat relation corresponds with
- the relation concept of entity-relationship modeling and many object oriented methods.

3.8 Other Nonfunctional Requirements

3.8.1 Performance Requirements

The application at this side controls and communicates with the following three main general components.

- embedded browser in charge of the navigation and accessing to the web service;
- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.
- Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine,
- Database System, Verification Engine, Output Module.

3.8.2 Safety Requirements

- The software may be safety-critical. If so, there are issues associated with its
- integrity level
- The software may not be safety-critical although it forms part of a safety-critical
- system. For example, software may simply log transactions.
- If a system must be of a high integrity level and if the software is shown to be of that
- integrity level, then the hardware must be at least of the same integrity level.
- There is little point in producing 'perfect' code in some language if hardware and
- system software (in widest sense) are not reliable.
- If a computer system is to run software of a high integrity level then that system
- should not at the same time accommodate software of a lower integrity level.
- Systems with different requirements for safety levels must be separated.
- Otherwise, the highest level of integrity required must be applied to all systems in
- the same environment.

3.9 HARDWARE AND SOFTWARE SPECIFICATION

3.10.1 HARDWARE REQUIREMENTS

- Hard disk : 500 GB and above.
- Processor : i3 and above.
- Ram : 4GB and above.

3.9.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 7 and above (64-bit).
- Python : 3.6

3.9.3 TECHNOLOGIES USED

- Convolutional Neural Network

CHAPTER 4

RESULTS AND DISCUSSION

4.1 CODING

Once the design aspect of the system is finalized, the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required; it easily screws into the system.

4.1.1 CODING STANDARDS

Coding standards are guidelines to programming that focus on the physical structure and appearance of the program. They make the code easier to read, understand, and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standards needed to achieve the above-mentioned objectives are as follows:

- Program should be simple, clear, and easy to understand.
- Naming conventions
- Value conventions
- Script and comment procedure
- Message box format
- Exception and error handling

4.1.2 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be self-descriptive. One should even get the meaning and scope of the variable by its name. The conventions are adopted for easy understanding of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names:

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name:

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

4.1.3 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

4.1.4 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

4.1.5 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X – User has performed illegal operation.
- ! – Information to the user.

4.2 RESULTS

As we discussed above the main aim of this project is to predict emoji's according to the input such as image, emoji and text using convolution neural networks shotly called as CNN Algorithm. This program predicts the emojis for text input and image inputs and cetaes a HTML document as output. The main idea behind doing this project is that over the past decade, emoji have emerged as a new and widespread form of digital communication, spanning diverse social networks and spoken languages. We propose treating these ideograms as a new modality in their own right, distinct in their semantic structure from both the text in which they are often embedded as well as the images which they resemble. As a new modality, emoji present rich novel possibilities for representation and interaction. The input and outputs of project is given below.

```

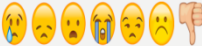
200     for text_list in x_all:
201         for words in text_list:
202             features_list.append(words)
203
204     features_list = list(dict.fromkeys(features_list))
205     print ( ' Creating Feature list vector')
206
207     for label,text in category_input:
208         text = text.lower()
209         line_encoded_list = []
210         for feature in features_list:
211             cnt = text.count(feature)
212             line_encoded_list.append(cnt)
213             feature_encoded_list.append(line_encoded_list)
214     print (" Completed Creating feature vector list")
215     return features_list,feature_encoded_list,y_all
216
217
218 training_dataset = read_label_file(TRAINING_FILE_MAIN)
219
220 y_all = []
221 features_list = []
222 feature_encoded_list = []
223
224 print("\n Creating feature list for prediction main category ")
225 features_list,feature_encoded_list, y_all = create_feature_vector( training_dataset.main_category , EMOJI_MAIN_GROUPS )
226 print(' Completed creating feature list vector. {} / {}'.format(len(feature_encoded_list),len(y_all)) )
227
228 input_stmts = ['Happy to work in Python', 'Afraid on Data Science', 'When I failed the S.S.C examination I felt very ashamed',
229 'Do not like Angular', 'Yellow , Yellow dirty fellow']
230
231 input_statement = InputStatement()
232 print('')
233 print(" Tagging main categories")
234 svcmodel = svm.LinearSVC(random_state=10,C=1)
235 svcmodel.fit(feature_encoded_list,y_all)

```

4.2.1: INPUT FOR PREDICTION OF EMOJIES USING TEXT

Here, we are providing the input to the program to understand the emotion in the text and then give the emojis according to the text emotion like if you are giving input as Happy to work in python it will give all happy emojis in the out put HTML page.

Emoji Prediction

Input Statement	Main Category	Sub Category	Emoji
Happy to work in Python	smileys_people	joy	
Afraid on Data Science	smileys_people	fear	
When I failed the S.S.C examination I felt very ashamed	smileys_people	shame	
Do not like Angular	smileys_people	sad	
Yellow , Yellow dirty fellow	smileys_people	sad	

4.2.2: OUTPUT FOR THE GIVEN ABOVE INPUT TEXT

In the above picture it is showing output of the given input, input statement is in the column of input statement in the picture and sub category refers to the emotion in the picture such as Do not like angular is an input in the program, it analyzed and then showed the emotion in the picture as sad and in final section it showed all related emojis in emoji column.

Emoji Prediction, Anticipation and Retrieval			
Project Name	Multimedia Retrieval using emoji Prediction, Anticipation, and Retrieval	Total Items Predicted	32
College	Sathyabama Institute Of Science And Technology	Average Accuracy rate in Kaggle Db	55%
Team	Girish Gupta & Subbarayadu	Our Model Accuracy rate	Check / Uncheck accuracy flag
Technology	AI/ML using Python		

Input Data	Machine Predicted Category	Machine Predicted Emoji's	Accuracy Flag
Input Image : 	fear	   	☒
Input Image : 	sad	      	☒
Input Image : 	sad	      	☒
Input Image : 	fear	   	☒
Input Image : 	angry	   	☒
Input Image : 	sad	      	☒
Input Image : 	neutral		☒

4.2.3: Image to emoji retrieval

In the given figure, it is representing the output of image to emoji retrieval module by analyzing the emotion in the given face of a human being and also predicted category and accuracy of this particular module.

Input Image	Predicted Emotion	Suggested Emojis	Accuracy
	happy	😊 😄 😁 😂 😃 😆 😇 😊	☒
	happy	😊 😄 😁 😂 😃 😆 😇 😊	☒
	happy	😊 😄 😁 😂 😃 😆 😇 😊	☒
	neutral	😐	☒
	angry	😡 😠 😡 😠	☒
	happy	😊 😄 😁 😂 😃 😆 😇 😊	☒
	sad	😞 😓 😔 😖 😗 😘 😙	☒
	neutral	😐	☒
	happy	😊 😄 😁 😂 😃 😆 😇 😊	☒
	happy	😊 😄 😁 😂 😃 😆 😇 😊	☒
	happy	😊 😄 😁 😂 😃 😆 😇 😊	☒

4.2.4 : BOTH IMAGE TO EMOJI AND EMOJI TO EMOJI RETRIVAL

In this figure, it is representing the output of image to emoji and emoji to emoji retrieval module by analyzing the emotion in the given face of a human being and also predicted category and accuracy of this particular module.

4.3 SUMMARY

The facial expression is specifically based on the emoticon identification system. It is identified as an open source extension to the tracker.js (A modern approach for Computer Vision on the web) framework that converts into expression of human facial to match best emoticon . The emoticons are the major parts towards the digital communication system. Emoticons, are also called, emoji's that "are used to express the emotions of a person through text in a way that is not possible with just words". As mentioned by Sashikar, "the importance of emoji has become so huge that they [also have been specifically] annotated with the WordNets" (p. 1). The vision of computer has grown at large scale that use as commercial product for the benefits of high speed.

The key task at hand ensures the face expression and face detection for the recognition of emotion. The formed task behind the face detection and emotional recognized gained from the eigen faces, fisher faces, and viola jones are the detections framework that prologs the hausdorff distance. The study exploration of the facial action coding system that observed the facial muscles, that plays an important role in the compiled and expressing emotions to find out the action units. Such action units are raising from the outer eyebrow and inner eyebrow, which is important for the quantification of human expressions.

CHAPTER 5

CONCLUSION AND FUTURE WORK

Conclusion:

In this project, we have approached emoji in different ways from text and images. There is sufficient motivation for doing so, and considerable future opportunities for research and applications with the emoji. We have proposed a large scale dataset of real-world emoji usage, containing the semantic relationships between emoji and text as well as emoji and images and emoji's. We have defined three challenge tasks with evaluation on this dataset, and provided baseline results for all three. We have looked at the problem of predicting emoji from text and/or images, both with the use of ample training data and in the absence of any. We have also looked at the problem of using emoji as queries for cross-modal retrieval.

Nowadays, Emoji are everywhere, and are becoming more pervasive. They already possess a distinct semantic space that can be utilized as a strong information signal as well as a novel means of interaction with data, through both query-by-emoji as well as emoji summarization of content. Furthermore, their semantic richness will only increase as new emoji continue to be introduced. It is our hope that this work and the challenge tasks defined within will spur further research and understanding of emoji within the multimedia community.

Future Work:

As mentioned above emojis are becoming the way to express emotions in social media like Instagram, Whatsapp and facebook and etc. So, we are thinking to optimize the mechanism used in this program by introducing a graphical user interface (GUI) by using stranded libraries of python such as TKInter. In the future, we will introduce real time emoji prediction by using OpenCV, which is an open source computer vision and machine learning software library. This algorithm can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to

produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

REFERENCES

- [1] W. Ai et al., “Untangling emoji popularity through semantic embeddings,” in *Proc. 11th Int. AAAI Conf. Web Social Media*, 2017, pp. 2–11.
- [2] L. M. Aiello et al., “Sensing trending topics in twitter,” *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1268–1282, Oct. 2013.
- [3] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, “Label-embedding for image classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1425–1438, Jul. 2016.
- [4] F. Barbieri, M. Ballesteros, and H. Saggion, “Are emojis predictable? In *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics: vol. 2, Short Papers*, 2017, pp. 105–111.
- [5] F. Barbieri, G. Kruszewski, F. Ronzano, and H. Saggion, “How cosmopolitan are emojis?: Exploring emojis usage and meaning over different languages with distributional semantics,” in *Proc. ACM Conf. Multimedia*, 2016, pp. 531–535.
- [6] F. Barbieri, F. Ronzano, and H. Saggion, “What does this emoji mean? A vector space skip-gram model for twitter emojis,” in *Proc. Lang. Resources Eval. Conf.*, 2016.
- [7] J. Berengueres and D. Castro, “Sentiment perception of readers and writers in emoji use,” 2017, arXiv:1710.00888.
- [8] S. Cappallo, T. Mensink, and C. G. M. Snoek, “Image2emoji: Zero-shot emoji prediction for visual media,” in *Proc. ACM Conf. Multimedia*, 2015, pp. 1311–1314.
- [9] S. Cappallo, T. Mensink, and C. G. M. Snoek, “Query-by-emoji video search,” in *Proc. ACM Conf. Multimedia*, 2015, pp. 735–736.
- [10] J. Chen, Y. Cui, G. Ye, D. Liu, and S.-F. Chang, “Event-driven semantic concept discovery by exploiting weakly tagged internet images,” in *Proc. Int. Conf. Multimedia Retrieval*, 2014, [11] Z. Chen et al., “Through a gender lens: An empirical study of emoji usage over large-scale android users,” 2017, arXiv:1705.05546.
- [12] G. Donato and P. Paggio, “Investigating redundancy in emoji use: Study on a twitter based corpus,” in *Proc. 8th Workshop Comput. Approaches Subjectivity*,

Sentiment Social Media Anal., 2017, pp. 118–126.

[13] A. El Ali, T. Wallbaum, M. Wasmann, W. Heuten, and S. C. Boll, “Face2emoji: Using facial emotional expressions to filter emojis,” in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2017, pp. 1577–1584.

[14] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, “Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm,” in *Proc. Conf. Empirical Methods Natural Language Process.*, 2017, pp. 1615–1625.

[15] Y. Gao and Q. Dai, “View-based 3d object retrieval: Challenges and approaches,” *IEEE MultiMedia*, vol. 21, no. 3, pp. 52–57, Jul./Sep. 2014.

[16] B. Guthier, K. Ho, and A. El Saddik, “Language-independent data set annotation for machine learning-based sentiment analysis,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2017, pp. 2105–2110.

[17] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[18] L. Jiang, T. Mitamura, S.-I. Yu, and A. G. Hauptmann, “Zero-example event search using multimodal pseudo relevance feedback,” in *Proc. Int. Conf. Multimedia Retrieval*, 2014, p. 297.

[19] J. Lee and S. C. Marsella, “Predicting speaker head nods and the effects of affective information,” *IEEE Trans. Multimedia*, vol. 12, no. 6, pp. 552–562, Oct. 2010.

[20] X. Li, R. Yan, and M. Zhang, “Joint emoji classification and embedding learning,” in *Proc. Asia-Pacific Web Web-Age Inf. Manage. Joint Conf. Web Big Data*, 2017, pp. 48–63.

[21] N. Ljubešić and D. Fišer, “A global analysis of emoji usage,” in *Proc. 10th Web as Corpus Workshop*, 2016, pp. 82–89.

[22] R. P. L’opez and F. Cap, “Did you ever read about frogs drinking coffee? investigating the compositionality of multi-emoji expressions,” in *Proc. 8th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, 2017, pp. 113–117.

[23] X. Lu et al., “Learning from the ubiquitous language: An empirical analysis of emoji usage of smartphone users,” in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 770–780.

[24] P. Mettes, D. C. Koelma, and C. G. M. Snoek, “The imagenet shuffle: Reorganized pre-training for video event detection,” in *Proc. Int. Conf. Multimedia Retrieval*, 2016, pp. 175–182.

- [25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, arXiv:1301.3781.
- [26] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Adv. Int Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
- [27] H. Miller *et al.*, "blissfully happy or ready to fight: Varying interpretations of emoji," in *Proc. 10th Int. Conf. Web Social Media*, 2016, pp. 259–268.
- [28] H. J. Miller, D. Kluver, J. Thebault-Spieker, L. G. Terveen, and B. J. Hecht, "Understanding emoji ambiguity in context: The role of text in emoji-related miscommunication," in *Proc. 11th Int. Conf. Web Social Media*, 2017, pp. 152–161.
- [29] N. Na'aman, H. Provenza, and O. Montoya, "Varying linguistic purposes of emoji in (twitter) context," in *Proc. Assoc. Comput. Linguistics*, 2017, pp. 136–141.
- [30] K. Njenga, "Social media information security threats: Anthropomorphic emoji analysis on social engineering," in *Proc. IT Convergence Security*, 2017, pp. 185–192.
- [31] M. Norouzi *et al.*, "Zero-shot learning by convex combination of semantic embeddings," in *Proc. Int. Conf. Learn. Represent.*, 2014.
- [32] P. K. Novak, J. Smailović, B. Sluban, and I. Mozetič, "Sentiment of emojis," *PloS one*, vol. 10, no. 12, 2015, Art. no. e0144296.
- [33] H. Pohl, C. Domin, and M. Rohs, "Beyond just text: Semantic emoji similarity modeling to support expressive communication," *ACM Trans. Comput.-Human Interact.*, vol. 24, no. 1, 2017, Art. no. 6.
- [34] M. Rathan, V. R. Hulipalled, K. Venugopal, and L. Patnaik, "Consumer insight mining: Aspect based twitter opinion mining of mobile phone reviews," *Appl. Soft Comput.*, vol. 68, pp. 765–773, 2017.
- [35] M. A. Riordan, "The communicative role of non-face emojis: Affect and disambiguation," *Comput. Human Behav.*, vol. 76, pp. 75–86, 2017.
- [36] D. Rodrigues, D. Lopes, M. Prada, D. Thompson, and M. V. Garrido, "A frown emoji can be worth a thousand words: Perceptions of emoji use in text messages exchanged between romantic partners," *Telematics Inf.*, vol. 34, pp. 1532–1543, 2017.
- [37] M. Shiha and S. Ayvaz, "The effects of emoji in sentiment analysis," *IJCEE*, vol. 9, pp. 360–369, 2017.
- [38] C. G. M. Snoek, M. Worring, J. C. Van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *Proc. 14th ACM Int. Conf. Multimedia*

- [39] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1733–1746, Oct. 2015.
- [40] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–6.
- [41] B. Thomee *et al.*, "The new data and new challenges in multimedia research," 2015, arXiv:1503.01817.
- [42] G. W. Tigwell and D. R. Flatla, "Oh that's what you meant!: Reducing emoji misunderstanding," in *Proc. 18th Int. Conf. Human–Comput. Interact. With Mobile Devices Services Adjunct*, 2016, pp. 859–866.
- [43] S. Wang *et al.*, "A natural visible and infrared facial expression database for expression recognition and emotion inference," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 682–691, Nov. 2010.
- [44] S. Wijeratne, L. Balasuriya, A. Sheth, and D. Doran, "Emojinet: Building a machine readable sense inventory for emoji," in *Proc. 8th Int. Conf. Social Inf.*, 2016, pp. 527–541.
- [45] S. Wijeratne, L. Balasuriya, A. Sheth, and D. Doran, "A semantics-based measure of emoji similarity," in *Proc. Int. Conf. Web Intell.*, 2017, pp. 646–653.
- [46] R. Zhou, J. Hentschel, and N. Kumar, "Goodbye text, hello emoji: Mobile communication on wechat in china," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2017, pp. 748–759.

PLAGARISM REPORT

MULTIMEDIA RETRIEVAL USING EMOJI PREDICTION,
ANTICIPATION, AND RETRIEVAL.docx

ORIGINALITY REPORT

5%

SIMILARITY INDEX

3%

INTERNET SOURCES

4%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to PES University

Student Paper

2%

2

pure.uva.nl

Internet Source

1%

3

[Spencer Cappallo](#), [Stacey Svetlichnaya](#), [Pierre Garrigues](#), [Thomas Mensink](#), [Cees G. M. Snoek](#). "New Modality: Emoji Challenges in Prediction, Anticipation, and Retrieval", IEEE Transactions on Multimedia, 2019

Publication

1%

4

Submitted to University of Hull

Student Paper

1%

C. JOURNAL PAPER

SOURCE CODE

```
import tensorflow as tf
import keras
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, AveragePooling2D
from keras.layers import Dense, Activation, Dropout, Flatten
from keras.preprocessing import image
from keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
from os import listdir
from os.path import isfile, join
def train_facialmodel():
    print ('\n*****WARNING *****
***** \n' )

    print ('RETRAIN_MODEL flage set to true. \n' )
    print ('Retraining model is a cpu intensive process. It might take 30 mins to 1 hour
depending on various factors ' )
    print ('like CPU, Ram, BatchSize etc. Unless any new image is added to training
data set, its recommended not to ' )
    print ('select this option.' )
    print ('\n*****WARNING
***** \n' )

    print ('Reading Training data')
    with open('training_data/face_emotion_recognition.dat') as f:
        content = f.readlines()
        lines = np.array(content)

    num_of_instances = lines.size
    print("number of instances: ",num_of_instances)
    print("instance length: ",len(lines[1].split(",")[1].split(" ")))
    content = "
x_train, y_train, x_test, y_test = [], [], [], []
```

```

#-----
#transfer train and test set data
for i in range(1,num_of_instances):
    try:
        emotion, img, usage = lines[i].split(",")

        val = img.split(" ")
        pixels = np.array(val, 'float32')

        # if ( emotion in training_vector_count ):
        # training_vector_count [emotion] = training_vector_count [emotion] + 1
        # else:
        # training_vector_count [emotion] = 1
        emotion = keras.utils.to_categorical(emotion, num_classes)

        if 'Training' in usage:
            y_train.append(emotion)
            x_train.append(pixels)
        elif 'PublicTest' in usage:
            y_test.append(emotion)
            x_test.append(pixels)
    except:
        print("",end="")
    lines = []
x_train = np.array(x_train, 'float32')
y_train = np.array(y_train, 'float32')
x_test = np.array(x_test, 'float32')
y_test = np.array(y_test, 'float32')

x_train /= 255 #normalize inputs between [0, 1]
x_test /= 255
x_train = x_train.reshape(x_train.shape[0], 48, 48, 1)
x_train = x_train.astype('float32')
x_test = x_test.reshape(x_test.shape[0], 48, 48, 1)
x_test = x_test.astype('float32')
print(x_train.shape[0], 'train samples')

```

```

print(x_test.shape[0], 'test samples')
return x_train,y_train,x_test,y_test
def emotion_analysis(emotions):
    objects = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')
    y_pos = np.arange(len(objects))
    plt.bar(y_pos, emotions, align='center', alpha=0.5)
    plt.xticks(y_pos, objects)
    plt.ylabel('percentage')
    plt.title('emotion')
    print ( y_pos )
    print ( objects )
    print ( emotions )
    plt.show()
    #maxpos = emotions.where(max(emotions))
    maxpos = np.argmax ( emotions )
    return objects[maxpos]
def image2emoji( input_folder_path ):
    complete_prediction_results = {}
    prediction_result = ""
    for img_file in listdir(input_folder_path):
        if isfile(join(input_folder_path, img_file) ):
            print ( 'Predicting : {}'.format(input_folder_path + img_file) )
            #input_image_files.append ( input_folder_path + img_file)
            img = image.load_img(input_folder_path + img_file, color_mode = "grayscale",
target_size=(48, 48))
            x = image.img_to_array(img)
            x = np.expand_dims(x, axis = 0)
            x /= 255
            custom = model.predict(x)
            # prediction_result = emotion_analysis(custom[0])
            prediction_result = EMOTION_OBJECTS[ np.argmax ( custom[0] ) ]
            complete_prediction_results [ input_folder_path + img_file ] = prediction_result
            # x = np.array(x, 'float32')
            # x = x.reshape([48, 48])

            # plt.gray()

```

```

    # plt.imshow(x)
    # plt.show()
    return complete_prediction_results
INPUT_FILE_PATH = 'testing/images_input/'
EMOTION_OBJECTS = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')
config = tf.ConfigProto( device_count = {'GPU': 0 , 'CPU': 56} ) #max: 1 gpu, 56 cpu
sess = tf.Session(config=config)
x_train, y_train, x_test, y_test = [], [], [], []
keras.backend.set_session(sess)
print ( ' TF backend set successfully ' )
num_classes = 7 #angry, disgust, fear, happy, sad, surprise, neutral
batch_size = 256
epochs = 30
#construct CNN structure
model = Sequential()
#1st convolution layer
model.add(Conv2D(64, (5, 5), activation='relu', input_shape=(48,48,1)))
model.add(MaxPooling2D(pool_size=(5,5), strides=(2, 2)))
#2nd convolution layer
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))
#3rd convolution layer
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(AveragePooling2D(pool_size=(3,3), strides=(2, 2)))
model.add(Flatten())
#fully connected neural networks
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
model.compile(loss='categorical_crossentropy'
    , optimizer=keras.optimizers.Adam()
    , metrics=['accuracy'])

```

```

)
fit = True
if fit == True:
    x_train,y_train,x_test,y_test = train_facialmodel()
    #-----
    #batch process
    gen = ImageDataGenerator()
    train_generator = gen.flow(x_train, y_train, batch_size=batch_size)
    #-----
    #model.fit_generator(x_train, y_train, epochs=epochs) #train for all trainset
    model.fit_generator(train_generator, steps_per_epoch=batch_size, epochs=epochs)
#train for randomly selected one
    model.save_weights("model.h5")
else:
    print ( 'Loading model from already trained model file.')
    model.load_weights('training_data/facial_expression_model_weights.h5')
prediction_results = image2emoji ( INPUT_FILE_PATH )
print (prediction_results)

for img in prediction_results:
    print('{} --> {}'.format( img, prediction_results[img]) )
print ( 'Please check the above output.. Did the model predict the image category
correct ? ')
print ( ' If the prediction accuracy is wrong then we need to INCREASE THE EPOCHS
in line number 139 ' )
print ( ' More the epoch more the iteration and more time. ' )

```