EMOJIFY - CREATE YOUR OWN EMOJI

A PROJECT REPORT

Submittedby

Gurleen Kaur
Varun Singh

in partial fulfillment for the award of the degree of

BACHLORS OF ENGINEERING

IN
BIG DATA ANALYTICS

Under the Supervision of:

Ms. Shaveta Jain

Chandigarh University

MAY 2023

BONAFIDE CERTIFICATE

Certified that this project report **"Emojify - Create your own emoji"** is the bonafide work of "Gurleen Kaur and Varun Singh" who carried out the project work under my/our supervision.

<<Signature of the HoD>>
SIGNATURE

HEAD OF THE DEPARTMENT

<<Signature of the Supervisor>>
SIGNATURE

Ms. Shaveta Jain
SUPERVISOR

Submitted for the project viva-voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENTS

List of Figures

List of Tables

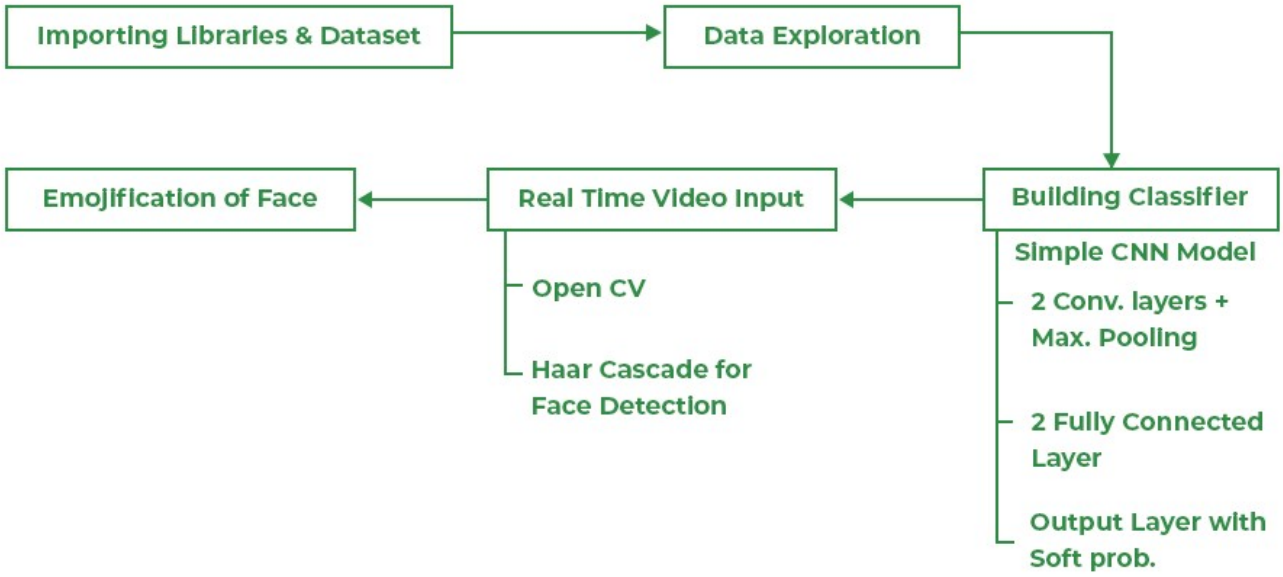
Abstract

Today's world is computer science world and technology is omnipresent. Everything is getting shifted to Online mode or simple say internet/network. Even day to day human communication had also transformed into digital communication through applications like whats app, Facebook, Instagram, Twitter, You tube and many more. Due to this modification the way of representing individual's thoughts and emotions had also been digitalized and transfigured and combination of visual and textual content in form of emojis and chat messages respectively. The computer science has been upgrading its field since long period of time. However, because of this enhancement, it is now achievable to get the idea about human sentiments from their facial posture and mutate them into Emojis/Avatars/Sticker accordingly. This project is going to fabricate emojis/avatars from user's facial formation. This software will allow us to understand the emotions of an individual in more coloured and cherished way in the form of emojis.

Keywords:

Convolutional Neural Network(CNN), Python, Visual Studio.

Graphical Abstract



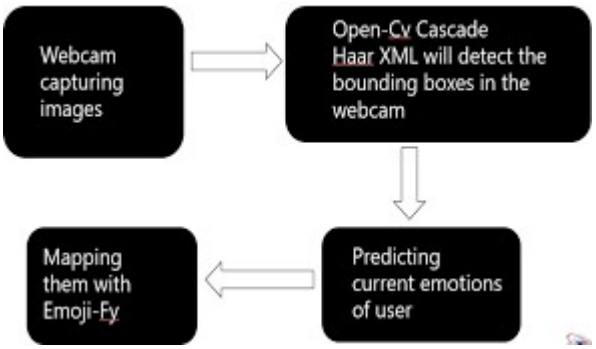


Fig-1: Graphical Flowchart of Emojify Process

Chapter 1 – Introduction

1.1. Background of Predictive Analytics

Predictive analytics incorporates the use of statistics but is a subject that requires a differentapproach and one that has different ideals than statistics. First and foremost, statistics is a field thatoperates based on a uniquely defined set of rules and a foundation of theory, whereas with predictive analytics, that may not always be the case. For instance, there are algorithms used, e.g. from fields like machine learning and artificial intelligence, within predictive analytics that do not have a best possible solution or such a solution that can even be proved. Furthermore, professionals that use predictive analytics are more lenient when it comes to models and less particular with model parameters. That is, when fitting a predictive model to the data, the focus is on optimizing predictive accuracy of some target. Again, predictive models are "data- driven" or in other words, predictive models are formed from the given data on the basis of being able to make and influence decisions. In general, the methods used to create a predictive model tend to be less rigorous in comparison to many statistical analysis techniques.

The algorithms that are used in predictive analytics can be categorized as either supervised learning methods or unsupervised learning methods. Supervised learning models aim to predict a target variable, represented by a single column in the dataset, by using the other variables or columns in the dataset. Supervised learning is also known as predictive modeling. The most common predictive modeling algorithms are classification when dealing with a categorical target variable or regression in the context of a continuous target variable. Unsupervised learning does not have a target variable, but rather builds a model using clusters of the data. Unsupervised learning models are referred to as descriptive modeling.

The emojify project is a popular application of predictive analytics that aims to predict the most suitable emoji to use for a given test or sentence. Predictive analytics is a subset of data analytics that uses statistical algorithms, machine learning techniques, and data mining methods to analyze historical data and make predictions about future events.

The use of predictive analytics in the emojify project using images/video involves the use of machine learning algorithms and deep learning models to analyze large datasets of labeled images and learn from past examples. This process typically involves the use of convolutional neural networks (CNNs), which are specialized deep learning models designed to recognize patterns and features in images.

In the context of the emojify - create your own emoji project, predictive analytics is used to analyze large amounts of images data and identify patters and relationships between images and their associated emojis. This involves the use of CNNs that are trained on large datasets of labeled images and associated emojis, allowing them to learn the relationship between specific visual features in the images and the appropriate emojis to use. Once trained, the CNNs can be used to predict the most suitable emoji for a given image by analyzing its visual features and matching it to similar patterns and relationships in the training data.

1.2. Introduction to Emojis:

An **emoji** (plural **emoji** or **emojis**) is pictogram, logogram, ideogram or smiley embedded in text and used in electronic messages and web pages. The primary function of emoji is to fill in emotional cues otherwise missing from typed conversation. Examples of emoji are

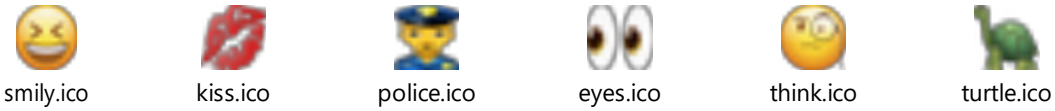


Fig-1.2.1: Emojis

Emoji exist in various genres, including facial expressions, common objects, places and types of weather, and animals. They are much like emoticons, except emoji are pictures rather than typographic approximations; the term "emoji" in the strict sense refers to such pictures which can be represented as encoded characters, but it is sometimes applied to messaging stickers by extension. Originally meaning pictograph, the word *emoji* comes from Japanese *e* (絵, 'picture') + *moji* (文 字 , 'character'); the resemblance to the English words *emotion* and *emoticon* is purely coincidental. The ISO 15924 script code for emoji is Zsye.

Originating on Japanese mobile phones in 1997, emoji became increasingly popular worldwide in the 2010s after being added to several mobile operating systems. They are now considered to be a large part of popular culture in the West and around the world. In 2015, Oxford Dictionaries named the Face with Tears of Joy emoji

the word of the year

1.2.1. Evolution from emoticons (1990s)

The emoji was predated by the emoticon, a concept implemented in 1982 by computer scientist Scott Fahlman when he suggested text-based symbols such as :-) and :-(could be used to replace language. Theories about language replacement can be traced back to the 1960s, when Russian novelist and professor Vladimir Nabokov stated in an interview with *The New York Times*: "I often think there should exist a special typographical sign for a smile — some sort of concave mark, a supine round bracket." It did not become a mainstream concept until the 1990s when Japanese, American and European companies began developing Fahlman's idea. Mary Kalantzis and Bill Cope point out that similar symbology was incorporated by Bruce Parello, a student at the University of Illinois, into PLATO IV, the first e-learning system, in 1972. The PLATO system was not considered mainstream, and therefore Parello's pictograms were only used by a small number of people. Scott Fahlman's emoticons importantly used common alphabet symbols, and aimed to replace language/text to express emotion, and for that reason are seen as the actual origin of emoticons.

Wingdings, a font invented by Charles Bigelow and Kris Holmes, was released by Microsoft in 1990. It could be used to send pictographs in rich text messages, but would only load on devices with the Wingdings font installed. In 1995, the French newspaper *Le Monde* announced that Alcatel would be launching a new phone, the BC 600. Its welcome screen displayed a digital smiley face, replacing the usual text seen as part of the "welcome message" often seen on other devices at the time. In 1997, J-Phone launched the SkyWalker DP-211SW, which contained a set of 90 emoji. It is thought to be the first set of its kind. Its designs, each measuring 12 by 12 pixels were monochrome, depicting numbers, sports, the time, moon phases and the weather. It contained the Pile of Poo emoji in particular. The J-Phone model experienced low sales, and the emoji set was thus rarely used.

In 1999, Shigetaka Kurita created 176 emoji as part of NTT DoCoMo's i-mode, used on its mobile platform. They were intended to help facilitate electronic communication, and to serve as a distinguishing feature from other services. Due to their influence, Kurita's designs were once claimed to be the first cellular emoji; however, Kurita has denied that this is the case. According to interviews, he took inspiration from Japanese manga where characters are often drawn with symbolic representations called *manpu* (such as a water drop on a face representing nervousness or confusion), and weather pictograms used to depict the weather conditions at any given time. He also drew inspiration from Chinese characters and street sign pictograms. The DoCoMo i-Mode set included facial expressions, such as smiley faces, derived from a Japanese visual style commonly found in manga and anime, combined with *kaomoji* and smiley elements. Kurita's work is displayed in the Museum of Modern Art in New York City.

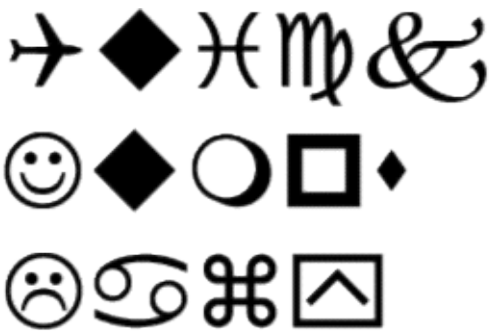


Fig-1.2.2: WingDings

Kurita's emoji were brightly colored, albeit with a single color per glyph. General-use emoji, such as sports, actions and weather, can readily be traced back to Kurita's emoji set. Notably absent from the set were pictograms that demonstrated emotion. The yellow-faced emoji in current use evolved from other emoticon sets and cannot be traced back to Kurita's work. His set also had generic images much like the J-Phones. Elsewhere in the 1990s, Nokia phones began including preset pictograms in its text messaging app, which they defined as "smileys and symbols". A third notable emoji set was introduced by Japanese mobile phone brand au by KDDI.

1.2.2. Development of emoji sets (2000-2007)

The basic 12-by-12-pixel emoji in Japan grew in popularity across various platforms over the next decade. This was aided by the popularity of DoCoMo i-mode, which for many was the origins of the smartphone. The i-mode service also saw the introduction of emoji in conversation form on messenger apps. By 2004, i-mode had 40 million subscribers, exposing numerous people to emoji for the first time between 2000 and 2004. The popularity of i-mode led to other manufacturers offering their own emoji sets. While emoji adoption was high in Japan during this time, the competitors failed to collaborate to create a uniform set of emoji to be used across all platforms in the country.

The Universal Coded Character Set (Unicode), controlled by the Unicode Consortium and ISO/IEC JTC 1/SC 2, had already been established as the international standard for text representation (ISO/IEC 10646) since 1993, although variants of Shift JIS remained relatively common in Japan. Unicode included several characters which would subsequently be classified as emoji, including some from North American or Western European sources such as DOS code page 437, ITC Zapf Dingbats or the WordPerfect Iconic Symbols set. Unicode coverage of written characters was extended several times by new editions during the 2000s, with little interest in incorporating the Japanese cellular emoji sets (deemed out of scope), although symbol characters which would subsequently be classified

Fig-1.2.3: Smiley faces from DOC code page

as emoji continued to be added. For example, Unicode 4.0 contained 16 new emoji, which included direction arrows, a warning triangle, and an eject button. Besides Zapf Dingbats, other dingbat fonts such as Wingdings or Webdings also included additional pictographic symbols in their own custom pi font encodings; unlike Zapf Dingbats, however, many of these would not be available as Unicode emoji until 2014.

The Smiley Company developed The Smiley Dictionary, which was launched in 2001. The desktop platform was aimed at allowing people to insert smileys as text when sending emails and writing on a desktop computer. The smiley toolbar offered a variety of symbols and smileys and was used on platforms such as MSN Messenger. Nokia, then one of the largest global telecom companies, was still referring to today's emoji sets as smileys in 2001. The digital smiley movement was headed up by Nicolas Loufrani, the CEO of The Smiley Company. He created a smiley toolbar, which was available at smileydictionary.com during the early 2000s to be sent as emoji are today.

1.2.3. Beginnings of Unicode emoji (2008–2014)

Mobile providers in both the United States and Europe began discussions on how to introduce their own emoji sets from 2004 onwards. Many companies did not begin to take emoji seriously until Google employees requested that Unicode look into the possibility of a uniform emoji set. Apple quickly followed and began to collaborate with not only Google, but also providers in Europe and Japan. In August 2007, Mark Davis and his colleagues Kat Momoi and Markus Scherer wrote the first draft for consideration by the Unicode Technical Committee (UTC), to introduce emoji into the Unicode standard. The UTC, having previously deemed emoji to be out of scope for Unicode, made the decision to broaden its scope to enable compatibility with the Japanese cellular carrier formats which were becoming more widespread. Peter Edberg and Yasuo Kida joined the collaborative effort from Apple Inc. shortly after, and their official UTC proposal came in January 2009.

Pending the assignment of standard Unicode code points, Google and Apple implemented emoji support via Private Use Area schemes. Google first introduced emoji in Gmail in October 2008, in collaboration with au by KDDI, and Apple introduced the first release of Apple Color Emoji to iPhone OS on 21 November 2008. Initially, Apple's emoji support was implemented for holders of a SoftBank SIM card; the emoji themselves were represented using SoftBank's Private Use Area scheme and mostly resembled the SoftBank designs. Gmail emoji used their own Private Use Area scheme, in a supplementary Private Use plane.

Separately, a proposal had been submitted in 2008 to add the ARIB extended characters used in broadcasting in Japan to Unicode. This included several pictographic symbols. These were added in Unicode 5.2 in 2009, a year before the cellular emoji sets were fully added; they include several characters which either also appeared amongst the cellular emoji or were subsequently classified as emoji.

After iPhone users in the United States discovered that downloading Japanese apps allowed access to the keyboard, pressure grew to expand the availability of the emoji keyboard beyond Japan. The Emoji application for iOS, which altered the Settings app to allow access to the emoji keyboard, was created by Josh Gare in February 2010. Before the existence of Gare's Emoji app, Apple had intended for the emoji keyboard to only be available in Japan in iOS version 2.2.

Throughout 2009, members of the Unicode Consortium and national standardization bodies of various countries gave feedback and proposed changes to the international standardization of the emoji. The feedback from various bodies in the United States, Europe, and Japan agreed on a set of 722 emoji as the standard set. This would be released in October 2010 in Unicode 6.0. Apple made the emoji keyboard available to those outside of Japan in iOS version 5.0 in 2011. Later, Unicode 7.0 (June 2014) added the character repertoires of the Webdings and Wingdings fonts to Unicode, resulting in approximately 250 more Unicode emoji.

The Unicode emoji whose code points were assigned in 2014 or earlier are therefore taken from several sources. A single character could exist in multiple sources, and characters from a source were unified with existing characters where appropriate: for example, the "shower" weather symbol

1000146-umbrell
a-rain-drops-emoj

from the ARIB source was unified with an existing umbrella with raindrops character, which had been added for KPS 9566 compatibility. The emoji characters named "Rain" ("雨", *ame*) from all three Japanese carriers were in turn unified with the ARIB character. However, the Unicode Consortium groups the most significant sources of emoji into four categories:

Table-1.2.1: Categories of emojis

Source category	Abbreviations	Unicode version (year)	Included sources	Example
Zapf Dingbats	ZDings, z	1.0 (1991)	ITC Zapf Dingbats Series 100	❣ (U+2763 ← 0xA3)[60]
ARIB	ARIB, a	5.2 (2008)	ARIB STD-B24 Volume 1 extended Shift JIS	⛩ (U+26E9 ← 0xEE4B)[61]
Japanese carriers	JCarrier, j	6.0 (2010)	NTT DoCoMo mobile Shift JIS	🐎 (U+1F3A0 ← 0xF8DA)[62]
			au by KDDI mobile Shift JIS	📌 (U+1F4CC ← 0xF78A)[62]
			SoftBank 3G mobile Shift JIS	🏘 (U+1F492 ← 0xFB7D)[62]
Wingdings and Webdings	WDings, w	7.0 (2014)	Webdings	🚅 (U+1F6F3 ← 0x54)[63]
			Wingdings	🏵 (U+1F3F5 ← 0x7B)[63]
			Wingdings 2	🖉 (U+1F58D ← 0x24)[63]
			Wingdings 3	▶ (U+25B6 ← 0x75)[63][a]

1.2.4. UTS #51 and modern emoji (2015–present)

In late 2014, a Public Review Issue was created by the Unicode Technical Committee, seeking feedback on a proposed Unicode Technical Report (UTR) titled "Unicode Emoji". This was intended to improve interoperability of emoji between vendors, and define a means of supporting multiple skin tones. The feedback period closed in January 2015. Also in January 2015, the use of the zero width joiner to indicate that a sequence of emoji could be shown as a single equivalent glyph (analogous to a ligature) as a means of implementing emoji without atomic code points, such as varied compositions of families, was discussed within the "emoji ad-hoc committee".

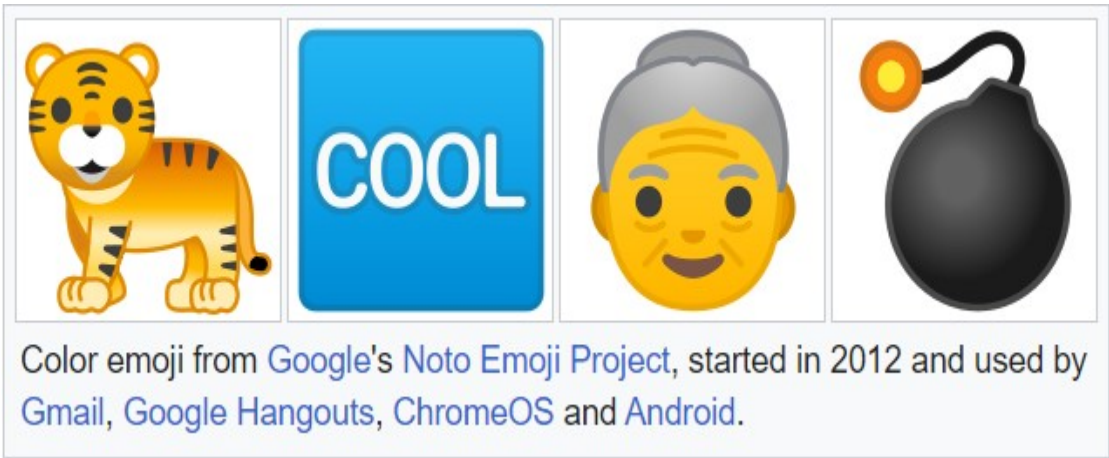


Fig-1.2.4: Color emojis

Unicode 8.0 (June 2015) added another 41 emoji, including articles of sports equipment such as the cricket bat, food items such as the taco, new facial expressions, and symbols for places of worship, as well as five characters (crab, scorpion, lion face, bow and arrow, amphora) to improve support for pictorial rather than symbolic representations of the signs of the Zodiac.

Also in June 2015, the first approved version ("Emoji 1.0") of the Unicode Emoji report was published as Unicode Technical Report #51 (UTR #51). This introduced the mechanism of skin tone indicators, the first official recommendations about which Unicode characters were to be considered emoji, and the first official recommendations about which characters were to be displayed in an emoji font in absence of a variation selector, and listed the zero width joiner sequences for families and couples that were implemented by existing vendors. Maintenance of UTR #51, taking emoji requests, and creating proposals for emoji characters and emoji mechanisms was made the responsibility of the Unicode Emoji Subcommittee (ESC), operating as a subcommittee of the Unicode Technical Committee.

With the release of version 5.0 in May 2017 alongside Unicode 10.0, UTR #51 was redesignated a Unicode Technical Standard (UTS #51), making it an independent specification rather than merely an informative document. As of July 2017, there were 2,666 Unicode emoji listed. The next version of UTS #51 (published in May 2018) skipped to the version number Emoji 11.0, so as to synchronise its major version number with the corresponding version of the Unicode Standard.

The popularity of emoji has caused pressure from vendors and international markets to add additional designs into the Unicode standard to meet the demands of different cultures. Some characters now defined as emoji are inherited from a variety of pre-Unicode messenger systems not only used in Japan, including Yahoo and MSN Messenger.

Corporate demand for emoji standardization has placed pressures on the Unicode Consortium, with some members complaining that it had overtaken the group's traditional focus on standardizing characters used for minority languages and transcribing historical records. Conversely, the Consortium recognises that public desire for emoji support has put pressure on vendors to improve their Unicode support, which is especially true for characters outside the Basic Multilingual Plane, thus leading to better support for Unicode's historic and minority scripts in deployed software.

1.2.5. Linguistic function of emojis

Linguistically, emoji are used to indicate emotional state, they tend to be used more in positive communication. Some researchers believe emoji can be used for visual rhetoric. Emoji can be used to set emotional tone in messages. Emoji tend not to have their own meaning but act as a paralanguage adding meaning to text. Emoji can add clarity and credibility to text.

Sociolinguistically, the use of emoji differ depending on speaker and setting. Women use emoji more than men. Men use a wider variety of emoji. Women are more likely to use emoji in public communication than private communication. Extraversion and agreeableness are positively correlated with emoji use, neuroticism is negative correlated. Emoji use differ between cultures: studies in terms of Hofstede's cultural dimensions theory found that cultures with high power distance and tolerance to indulgence used more negative emojis, while those with high uncertainty avoidance, individualism, and long-term orientation use more positive emojis.

1.2.6. Emoji communication problems

Research has shown that emoji are often misunderstood. In some cases, this misunderstanding is related to how the actual emoji design is interpreted by the viewer; in other cases, the emoji that was sent is not shown in the same way on the receiving side.

The first issue relates to the cultural or contextual interpretation of the emoji. When the author picks an emoji, they think about it in a certain way, but the same character may not trigger the same thoughts in the mind of the receiver (see also Models of communication).

For example, people in China have developed a system for using emoji subversively, so that a smiley face could be sent to convey a despising, mocking, and even obnoxious attitude, as the orbicularis oculi (the muscle near that upper eye corner) on the face of the emoji does not move, and the orbicularis oris (the one near the mouth) tightens, which is believed to be a sign of suppressing a smile.

The second problem relates to technology and branding. When an author of a message picks an emoji from a list, it is normally encoded in a non-graphical manner during the transmission, and if the author and the reader do not use the same software or operating system for their devices, the reader's device may visualize the same emoji in a different way. Small changes to a character's look may completely alter its perceived meaning with the receiver. As an example, in April 2020, British actress and presenter Jameela Jamil posted a tweet from her iPhone using the Face with Hand Over Mouth emoji (COVID-19 pandemic. On Apple's iOS, the emoji expression is neutral and pensive, but on other platforms the emoji shows as a giggling face.

Many fans were initially upset thinking that she, as a well off celebrity, was mocking poor people, but this was not her intended meaning.

Researchers from German Studies Institute at Ruhr-Universität Bochum found that most people can easily understand an emoji when it replaces a word directly – like an icon for a rose instead of the word 'rose' – yet it takes people about 50 percent longer to comprehend the emoji.

1.2.7. Background and Preliminaries

Face Detection and Face Expression Recognition are the main tasks available. We have different techniques for the former assignment, such as fisherfaces, eigenfaces, viola jones object detection system, hausdorff distance, etc. The visual technology techniques for translating expressions into the art of transforming facial graphic style are making a significant contribution and development in the design of digital graphics. Histograms were used as the method of representing the pyramids of the facial gradients by the initial resources used by the people associated with this area.

Ekman and Friesen observed the muscles of the face which is necessary to convey emotions and compile their emotions. A system of 46 Action Units (AUs) results. Such intervention Modules, some of which are the inner eyebrow and eyebrow raising units, the elevation of the outer eyebrow was extremely significant, in quantifying expressions of humans. Before this framework was developed,facial expression analysis,relied heavily on Human labelling of example expressions and many researchers were worried about the Bias Relevant to the cultural context or the cultural context or the labeler's emotional state at The time.The advent of Ekmanand Feisen's facial action coding system in 1977 put these issues to rest and became the golden standard easily.

.

The units of facial movement are closely related to musculature of the face and can therefore be mixed in ways that either independent or communicating with each other to form various appearances inherently. In 2015, the Emotions in the Wild (EmotiW 2015) contest used static forms of images that profoundly experienced the influence of convolutionary neural networks (CNNs) for definitions, such as emotions (Duncan et al., n.d., p. 2). Their precision was around 62 percent, but Levi & Hassner's recent development has shown significant improvement in facial emotion recognition using a CNN.

The process of identification of various forms of human emotions shows two difficulties, or issues, that are important. First, the availability of restricted data used analytically for instruction or CNN training procedures. Second, the variation of an illumination, which typically occurs in local emotional patterns using binary invariant techniques, will be part of the data collection. In manipulating the current models of emotion graphics, the use of 3D technology is typically correlated with the expression of the face. We could have the level of accuracy by five percent from the web face the challenge via technology as emoji, while the changes from the previous results will yield only 10-16%.

As the initiating factor for the implementation of the novel facial representation model, the use of the graphical emotional neural method such as the VGG-S will be mandatory. Using the database sources that reveal the advanced version of CNN, a game theory is applied as the moving average will be taken. The feelings must be defined or sensed in a way that would illustrate the stream of images or videos when translating graphical streams into the input stage. The LTB's descriptor function will use the guidelines for database illumination to detect the existence of invariant portions of the images.

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 philtres, which are normal practise. The visualisation process of the emotions that converts the recognition database of the person's expression with theoretical facial movements is a simple optimization feature.

The data collection, such as the web face, uses the features of this graphic representation of emotions in the translation of facial images as if one of CASIA depends on the availability of resources. The usual approach in transmitting this expression through graphical design projects or considerations of improvements would interact with the static facial expressions in the wild (SFEW) dataset or the statistical techniques to detect the expression for the facial wild to show the natural effects on the emotions of people's images.

The picture that has the static developments in the identification of input emotions is demonstrated in the visual posture applications for the language of the body. For the methods of detecting the large range of pyramids of computer database applications, the face moods highlighting the EEG are complex. This small-scale training system or the models that could relate facial perceptions or emotions to optimise the facial defection of emotions in layer- based technology, such as VGS, are also integrated with networks such as CNN.

1.3. Introduction to Project

Emojis are nothing but tiny pictures of emotions or we can also refer them as stickers. Emojis are the present day channel of chatting with one another online. Emojis/Avatars have completely changed the way people used to communicate and now emojis are ruling the online mode of conveying the emotions. Emojis are becoming the new language, which is being used by people all over the world. Simply, it is the emerging language of universe. Emojis - to enrich the written form of communication.

The reason of emojis getting used world wide is the simplicity they have in themselves. As already mentioned they are just little pictures like cartoons but they convey all the information about someone's state of mind, someone's reaction, or someone's response that simple text message can't pass to other individual's message sitting on the other side of medium in communication. Straightforwardly, emojis are nonverbal method of expressing feelings of a person.

The creator of emoji, Shigetaka Kurita, wanted to enable communication of "thoughts or emotions without inspiring strong likes or dislikes in the way a picture might". An emoji is a graphic symbol, ideogram, which represents not only facial expressions, but also concepts and ideas, such as celebration, weather, vehicles and buildings, food and drink, animals and plants, or emotions, feelings, and activities.

Every so often, text messages are not proved to transmit the emotions sender wants to show to the receiver. In the rescue of text messages, emojis are there! Emojis can reflect ideas, state of mind, emotions or sentiments effectively and precisely. The logical formula that i use to state emojis is:

Emoji = nonverbal communication + current state of mind

Since it is nonverbal so user doesn't have to write long-long paragraphs to express in which state he or she is! Hence, it is also not very tedious task to do so, just one click on the emoji that matches with the user's present state of feeling and there he/she goes! In support of my point, i would like to point out famous character's in history of Television shows.

- **Charlie Chaplin** had not spoken a single word in his entire TV show and yet he was able to make everyone laugh by expressing his emotions through his facial formation that everyone loved. He believed that his comedy would not translate to audiences via having talk. He did believed that emotions are best way to communicate.

- **Mr. Bean,** a cartoon character loved by children, youngsters and also by adults. It has been characterized to only make facial expressions and not speak a word throughout the cartoon show, still everyone including children, youngsters and adults are able to figure it out what he is conveying.

Above two points shows the power of facial expressions, the strength they carry to communicate the sentiments and feelings.

Emojis are available for use in almost all the digital platforms like WhatsApp, Facebook and a lot more. After looking at the increasing interest and affection of present generation people in communicating through ways that consists of emojis, the organizations have modified their way of surveying the people. The organizations and industries have started to take feedback's in form of emojis. Emojis are getting turned into global language and there are researches going on to build emoji-driven storytelling.

In a nutshell, it is committed that present generation love to communicate through emojis and avatars. So we have developed our project which is supposed to build our own customized emojis similar to human face formation. Our Project is including seven human expressions that consists of emotions such as happy, neutral, sad, surprise, angry, fearful, disgusted.

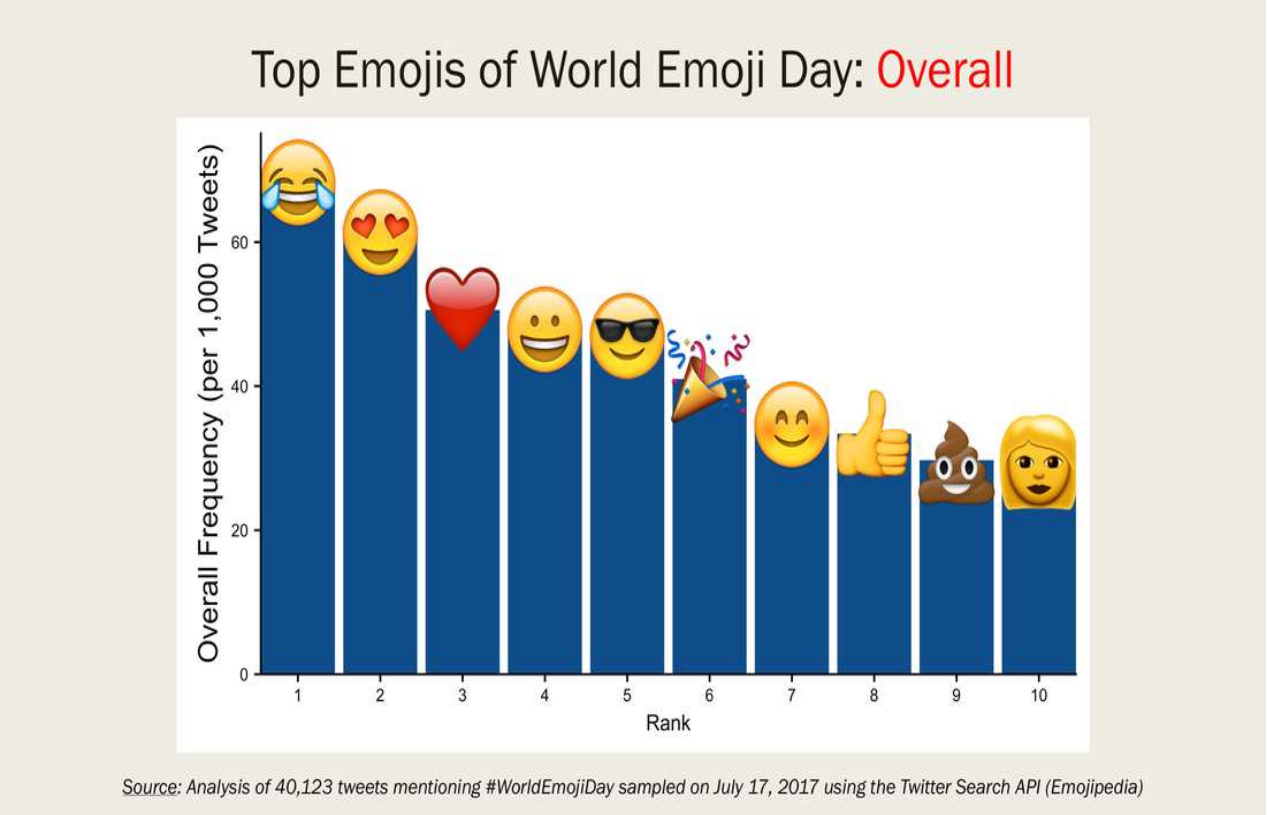


Fig-1.3.1: Uses of Emojis

1.4. Objective

The objective of the Emojify Project is to develop a system that can analyze an input image and generate an appropriate emoji that reflects the emotions conveyed in the image. The objective of an emojify project is to generate relevant set of emojis based on the visual content of an image. This type of project is commonly used in computer vision and image processing applications, where goal is to automatically extract the most relevant features or objects from and image and map them to corresponding emojis.This project aims to enhance communication and expression by providing a fun and intuitive way for users to convey their emotions through digital means. By using computer vision techniques and machine learning algorithms, the Emojify project can help bridge the gap between human emotions and digital communication, allowing users to convey their emotions more accurately and effectively. Overall, the objective of this project is to create a user friendly and accurate system that can generate emojis that accurately reflect the emotions displayed in input, thereby enhancing digital communication.

The current work will therefore be focused to explore seven major facial expressions including:

- ✧ Angry
- ✧ Disgust
- ✧ Fearful
- ✧ Happy
- ✧ Neutral
- ✧ Sad
- ✧ Surprise

Chapter 2 – Methods

2.1. Problem Definition

Emoji helps individual's to express feelings and their identities more "authentically" by increasing the semantic quality of visual messages. Emojis are also now being used in feedback forms. The feelings represented by the text or its severity are changed by emojis. Indeed, by simulating facial gestures, emojis can be used in Informal Text Communication (ITC) to express feelings such as sarcasm, irony or non-textual humour.

Emoji let user's to choose from large lists. It is one way to display nonverbal signs. In this project we will be exploring "Emotional recognition" using facial expression through emoji. The created software program contains seven human expressions that include emotions that are happy, neutral , sad, surprise, angry, fearful, disgusted. The real expressions that are being expressed are the expressions transmitted by human beings, because of their capacity to better communicate emotional responses and the way they promote contact between individuals, the investigations of such speech are important.

In today's age, the use of communication through various platforms, such as cell phones and computers, is very popular. Any of the ways of contact which are very popular today are e-mails, instant messages and blog posts. All of these include emojis. All of the above facts and observations have made emojis the necessity to explore them more.

So we are indulged to develop a software that will be using algorithm that can automatically assign the most appropriate emoji to a given input image.

2.2. Problem Formulation

The problem formulation for this project can be broken down into following steps:

Data Collection: Collect a large dataset of images that contain human faces displaying different emotions, such as happiness, sadness, anger, surprise, fear and disgust.

Pre-processing: Pre-process the input images to ensure that they are in a suitable format for analysis. This may include re-sizing, cropping and converting images to grayscale.

Emotion Detection: Develop and algorithm or use existing models to detect the primary emotion displayed in the input image. This may involve using computer vision techniques, such as feature extraction and machine learning, to analyze the facial expressions, body language and other visual cues.

Emoji Generation: Develop a system that can generate an appropriate emoji that reflects the detected emotion. This may involve using pre-trained models or training new models to match the detected emotion with corresponding emoji.

User Interface: Design a user interface that allows users to upload an image, analyze it for emotions, and display the corresponding emoji. This may involve developing a web application, mobile application, or desktop application that integrates the emotion detection and emoji generation algorithms.

Testing and Evaluation: Test the system using a diverse range of images and evaluate its performance based on metrics such as accuracy, speed, and user satisfaction.

2.3. Hardware Requirements

- PC
- Good internet connection
- Visual studio or any other compiler installed on system

Software Requirements

- TensorFlow.
- Keras.
- Numpy.

2.4. Convolutional Neural Network

A convolutional neural network (CNN) is a type of Deep Learning neural network architecture commonly used in computer vision. Computer vision is a field of Artificial Intelligence that enables a computer to understand and interpret the image or visual data.

When it comes to Machine Learning, Artificial Neural Networks perform really well. Neural Networks are used in various datasets like images, audio, and text. Different types of neural networks are used for different purposes, for example for predicting the sequence of words we use Recurrent Neural Networks more precisely an LSTM, similarly for image classification we use convolution neural networks. Now er are going to discuss the basic building block for CNN.

In regular Neural Network there are three types of layers.

1. **Input Layer:** It's the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in case of an image)

2. **Hidden Layer:** The input from the input layer is then feed into the hidden layer. There can be many hidden layers depending upon our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.

3. **Output Layer:** The output from the hidden layer is then feed into logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

The data is fed into the model and output from each layer is obtained from the above step is called feed-forward, we then calculate the error using error function, some common error functions are cross-entropy, square-loss error etc. The error function measures how well the network is performing. After that, we back propagate into the model by calculating derivatives. This step is called Backpropagation which basically is used to minimize the loss.

Convolutional Neural Network

Convolutional Neural Network (CNN) is the extended version of artificial neural networks (ANN) which is predominately used to extract the features form the grid-like matrix dataset. For example visual datasets like images or videos where data patters play and extensive role.

2.4.1. CNN Architecture:

Convolutional Neural Network consists of multiple layers like the input layer, convolutional layer, pooling layer, and fully connected layers.

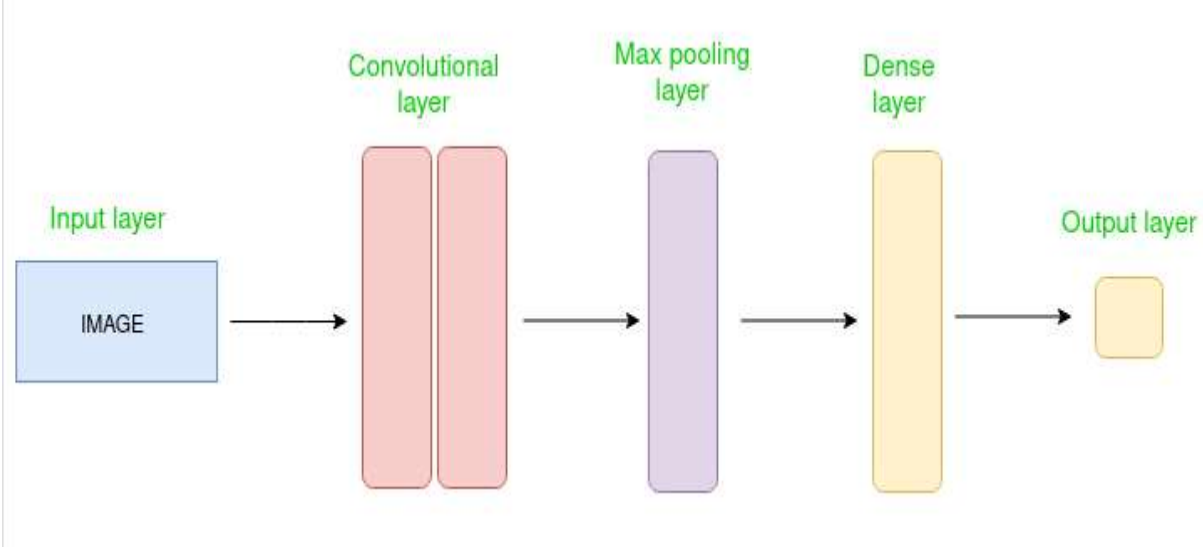


Fig-2.4.1: CNN

The convolutional layer applies filters to the input image to extract features, the pooling layer down-samples the image to reduce computation, and the fully connected layer makes the final prediction. The network learns the optimal filters through Backpropagation and gradient descent.

2.5. How Convolutional Layers Works

Convolutional neural networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of image), and height (i.e. the channel as images generally have red, green, and blue channels).

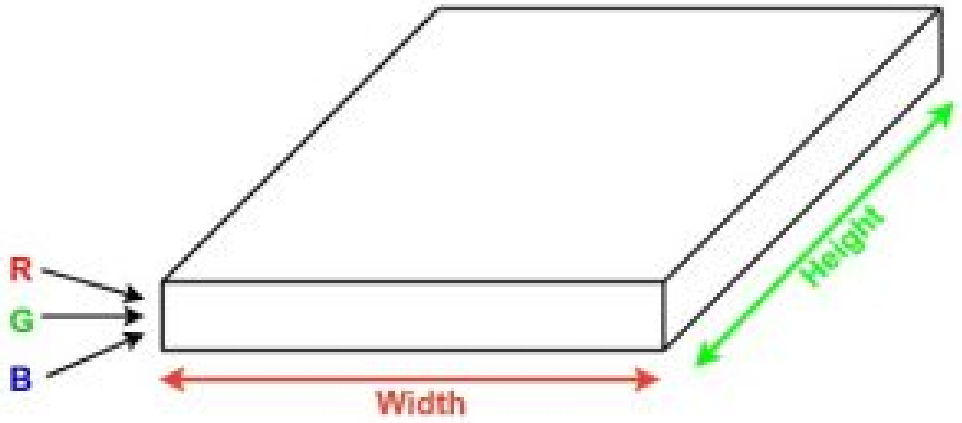


Fig-2.5.1: rectangle

Now imagine taking a small patch of this image and running a small neural network, called a filter or kernel on it, with say, K outputs and representing them vertically. Now slide the neural network across the whole image, as a result, we will get another image with different widths, heights and depths. Instead of just R, G, and B channels now we have more channels but lesser width and height. This operation is called Convolution. If the patch size if the same as that of image it will be regular neural network. Because of this small patch, we have fewer weights.

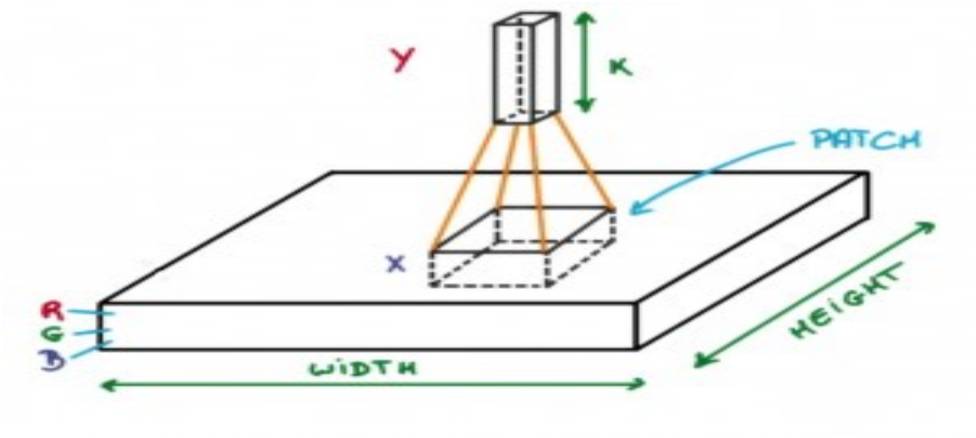


Fig-2.5.2: Deep learning udacity

Now let's talk about a bit of mathematics that is involved in the whole convolution process.

- Convolution layers consist of a set of learnable filters (or kernels) having small widths and heights and the same depth as that of input volume (3 if the input layer is image input)
- For example, if we have to run convolution on an image with dimensions 34 X 34 X 3. The possible size of filters can be axax3, where 'a' can be anything like 3,5, or 7 but smaller as compared to the image dimension.
- During the forward pass, we slide each filter across the whole input volume step by step where each step is called stride (which can have a value of 2,3, or even 4 foe high-dimensional images) and compute the dot product between the kernel weights and patch from input volume.
- As we slide our filters we'll get 2-D output for each filter and we'll stack them together as a result, we'll get output volume having depth equal to the number of filters. The network will learn all the filters.

Layers used to build CovNets

A complete convolutional neural networks architecture is also known as covnets. A covnets is a sequence of layers and every layer transforms one volume to another through a differentiable function.

Type of layers: datasets

Let's take an example by running a covnets on of image of dimension 32 X 32 X 3

Input Layers: it's the layer in which we give input to out model. In CNN, generally, the input will be an image of a sequence of images. This layer hold the raw input of the image with width 32, height 32, and depth 3.

Convolutional Layers: This is the layer, which is used to extract the feature from the input dataset. It applies a set of learnable filters known as the kernels to the input images. The filters/kernels are smaller matrices usually 2 X 2, 3 X 3 or 5 X 5 shape. It slides over the input image data and computes the dot product between kernel weight and the corresponding input image patch. The output of this layer is referred ad feature maps. Suppose we use a total of 12 filters for this layer we'll get and output volume of dimension 32 X 32 X 12.

Activation Layer: By adding an activation function to the output of the preceding layer, activation layers add non-linearity to the network. It will apply and element wise activation function to the output of the convolution layer. Some common activation functions are RELU:
Max(0, x). Tanh, Leaky RELU etc. The volume remains unchanged hence output volume will have dimensions 32 X 32 X 12.

Pooling Layer: This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents over-fitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 X 2 filters and stride 2, the resultant volume will be of dimension 16 X 16 X 12.

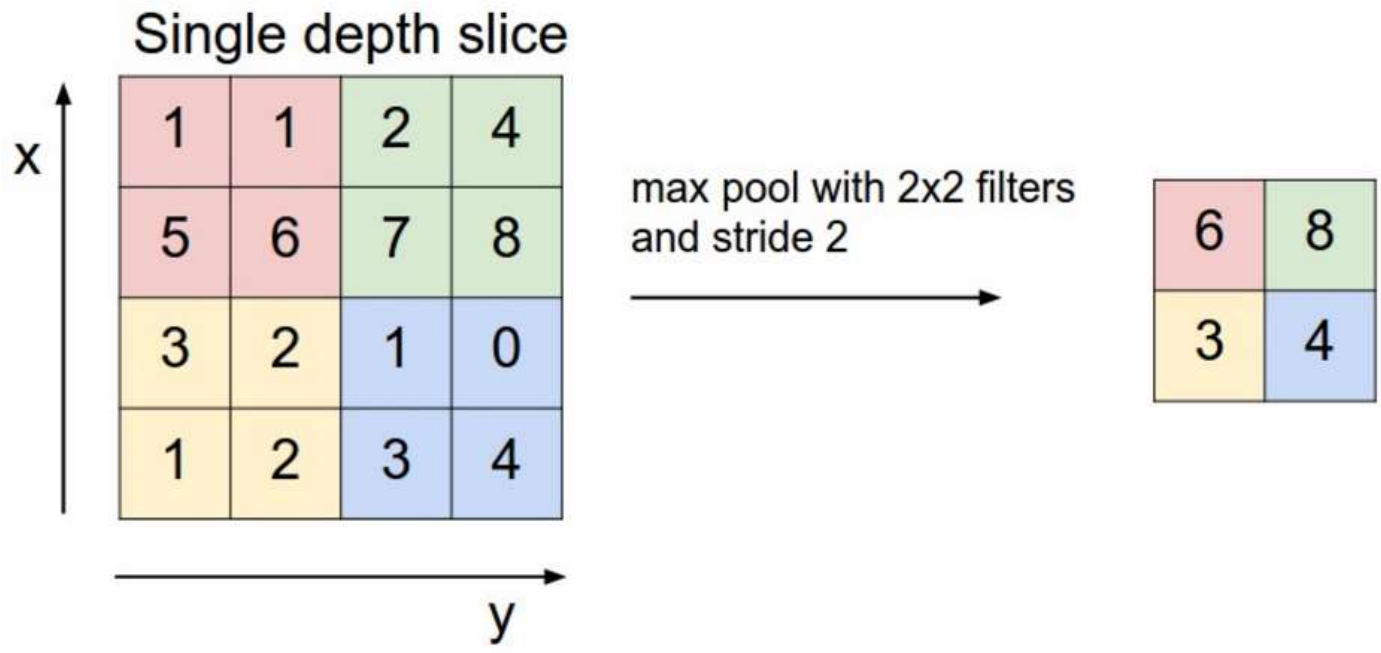


Fig-2.5.3: cs231.stanford.edu

Flattening: The resulting feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they can be passed into a completely linked layer for categorization or regression.

Fully Connected Layers: It takes the input from the previous layer and computes the final classification or regression task.

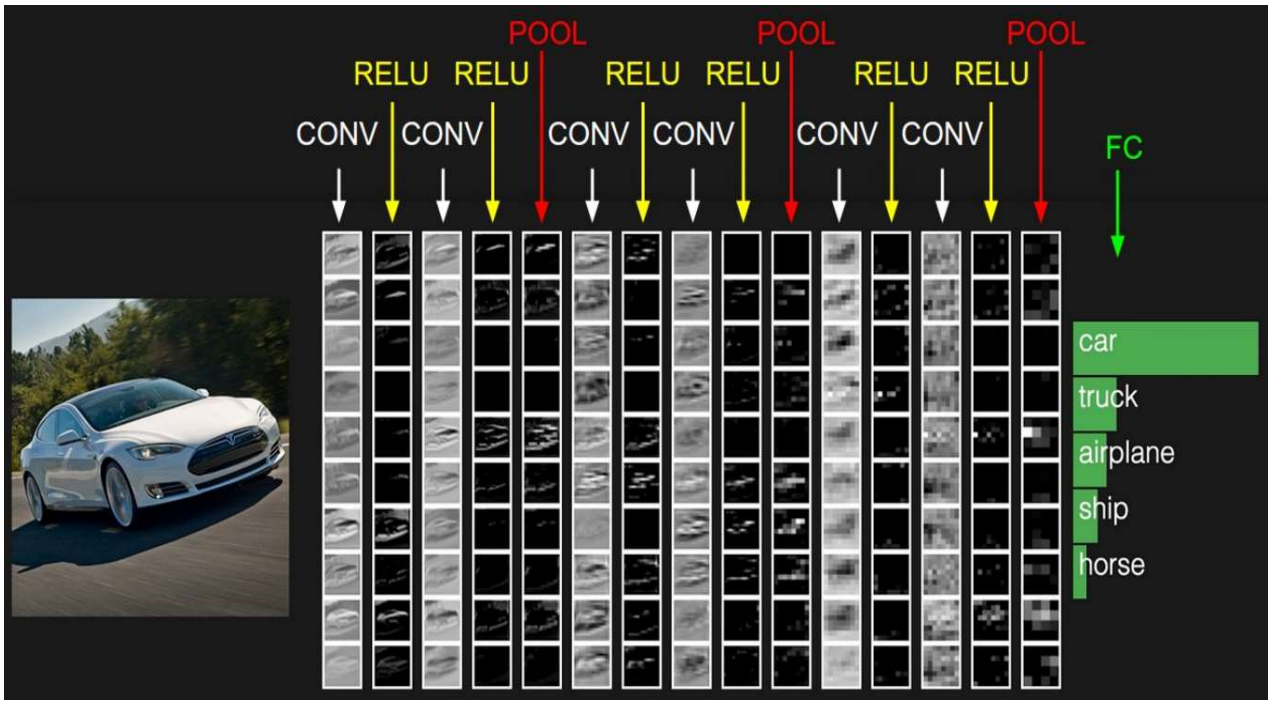


Fig-2.5.4: cs231.stanford.edu

Output Layer: The output from the fully connected layers is then fed into a logistic function for classification tasks like sigmoid or softmax which converts the output of each class into the probability score of each class.

Example:

Let’s consider an image and apply the convolution layer, activation layer, and pooling layer operation to extract the inside feature.

Fig-2.5.5: Ganesh

Step:

- import the necessary libraries
- set the parameter
- define the kernel
- Load the image and plot it.
- Reformat the image
- Apply convolution layer operation and plot the output image.
- Apply activation layer operation and plot the output image.
- Apply pooling layer operation and plot the output image.

Fig-2.5.6: Gray scale

2.6. Methdology

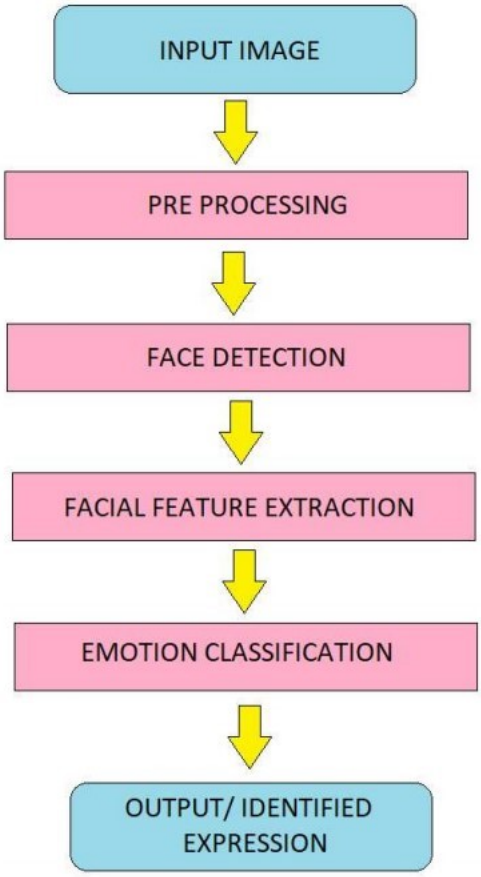


Fig-2.6.1: Flowchart

Chapter 3 – Results

3.1 Source Code

In the below steps will build a convolution neural network architecture and train the model on FER2013 dataset for Emotion recognition from images.

Step1:- Make a file train.py and follow the steps

1. Imports:

```
1. import numpy as np
2. import cv2
3.
4. from keras.emotion_models import Sequential
5. from keras.layers import Dense, Dropout, Flatten
6. from keras.layers import Conv2D
7. from keras.optimizers import Adam
8. from keras.layers import MaxPooling2D
9. from keras.preprocessing.image import ImageDataGenerator
```

2. Initialize the training and validation generators:

```
train_dir = 'data/train'
val_dir = 'data/test'
train_datagen = ImageDataGenerator(rescale=1./255)
val_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
        train_dir,
        target_size=(48,48),
        batch_size=64,
        color_mode="gray_framescale",
        class_mode='categorical')

validation_generator = val_datagen.flow_from_directory(
        val_dir,
        target_size=(48,48),
        batch_size=64,
        color_mode="gray_framescale",
        class_mode='categorical')
```

3. Build the convolution network architecture:

```
1.  emotion_model = Sequential()
2.
3.  emotion_model.add(Conv2D(32, kernel_size=(3, 3),
    activation='relu', input_shape=(48,48,1)))
4.  emotion_model.add(Conv2D(64, kernel_size=(3, 3),
    activation='relu'))
5.  emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
6.  emotion_model.add(Dropout(0.25))
7.
8.  emotion_model.add(Conv2D(128, kernel_size=(3, 3),
    activation='relu'))
9.  emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
10. emotion_model.add(Conv2D(128, kernel_size=(3, 3),
    activation='relu'))
11. emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
12. emotion_model.add(Dropout(0.25))
13.
14. emotion_model.add(Flatten())
15. emotion_model.add(Dense(1024, activation='relu'))
16. emotion_model.add(Dropout(0.5))
17. emotion_model.add(Dense(7, activation='softmax'))
```

4. Compile and train model:

```
emotion_model.compile(loss='categorical_crossentropy',optimizer=Adam(lr=0.0001, decay=1e-6),metrics=['accuracy'])

emotion_model_info = emotion_model.fit_generator(
        train_generator,
        steps_per_epoch=28709 // 64,
        epochs=50,
        validation_data=validation_generator,
        validation_steps=7178 // 64)
```

5. Save the model weights:

```
emotion_model.save_weights('model.h5')
```

6. Using openCV harcascade xml detect the bounding boxes of face in the webcam and predict the emotions:

```
cv2.ocl.setUseOpenCL(False)

emotion_dict = {0: "Angry", 1: "Disgusted", 2: "Fearful", 3: "Happy", 4: "Neutral", 5: "Sad", 6: "Surprised"}

cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    if not ret:
        break
    bounding_box = cv2.CascadeClassifier('/home/shivam/.local/lib/python3.6/site-packages/cv2/data/haarcascade_frontalface_default.xml')
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2gray_frame)
    num_faces = bounding_box.detectMultiScale(gray_frame,scaleFactor=1.3, minNeighbors=5)

    for (x, y, w, h) in num_faces:
        cv2.rectangle(frame, (x, y-50), (x+w, y+h+10), (255, 0, 0), 2)
        roi_gray_frame = gray_frame[y:y + h, x:x + w]
        cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray_frame, (48, 48)), -1), 0)
        emotion_prediction = emotion_model.predict(cropped_img)
        maxindex = int(np.argmax(emotion_prediction))
        cv2.putText(frame, emotion_dict[maxindex], (x+20, y-60), cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2, cv2.LINE_AA)

    cv2.imshow('Video', cv2.resize(frame,(1200,860),interpolation = cv2.INTER_CUBIC))
    if cv2.waitKey(1) & 0xFF == ord('q'):
cap.release()
cv2.destroyAllWindows()
    break
```

Step2:- Code for GUI and mapping with emojis:

```
import tkinter as tk
from tkinter import *
import cv2
from PIL import Image, ImageTk
import os
import numpy as np
import cv2
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D
from keras.optimizers import Adam
from keras.layers import MaxPooling2D
from keras.preprocessing.image import ImageDataGenerator

emotion_model = Sequential()

emotion_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu
emotion_model.add(Conv2D(64, kernel_size=(3, 3), activation='relu
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='rel
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='rel
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Flatten())
```

```
28.  emotion_model.add(Flatten())
29.  emotion_model.add(Dense(1024, activation='relu'))
30.  emotion_model.add(Dropout(0.5))
31.  emotion_model.add(Dense(7, activation='softmax'))
32.  emotion_model.load_weights('model.h5')
33.
34.  cv2.ocl.setUseOpenCL(False)
35.
36.  emotion_dict = {0: "   Angry   ", 1: "Disgusted", 2: "  Fearful
37.
38.
39.  emoji_dist=
     {0:"./emojis/angry.png",2:"./emojis/disgusted.png",2:"./emojis/fea
40.
41.  global last_frame1
42.  last_frame1 = np.zeros((480, 640, 3), dtype=np.uint8)
43.  global cap1
44.  show_text=[0]
45.  def show_vid():
46.      cap1 = cv2.VideoCapture(0)
47.      if not cap1.isOpened():
48.          print("cant open the camera1")
49.      flag1, frame1 = cap1.read()
50.      frame1 = cv2.resize(frame1,(600,500))
51.
52.      bounding_box = cv2.CascadeClassifier('/home/shivam/.local/lib
53.      gray_frame = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
54.      num_faces = bounding_box.detectMultiScale(gray_frame,scaleFac
55.
56.      for (x, y, w, h) in num_faces:
57.          cv2.rectangle(frame1, (x, y-50), (x+w, y+h+10), (255, 0,
```

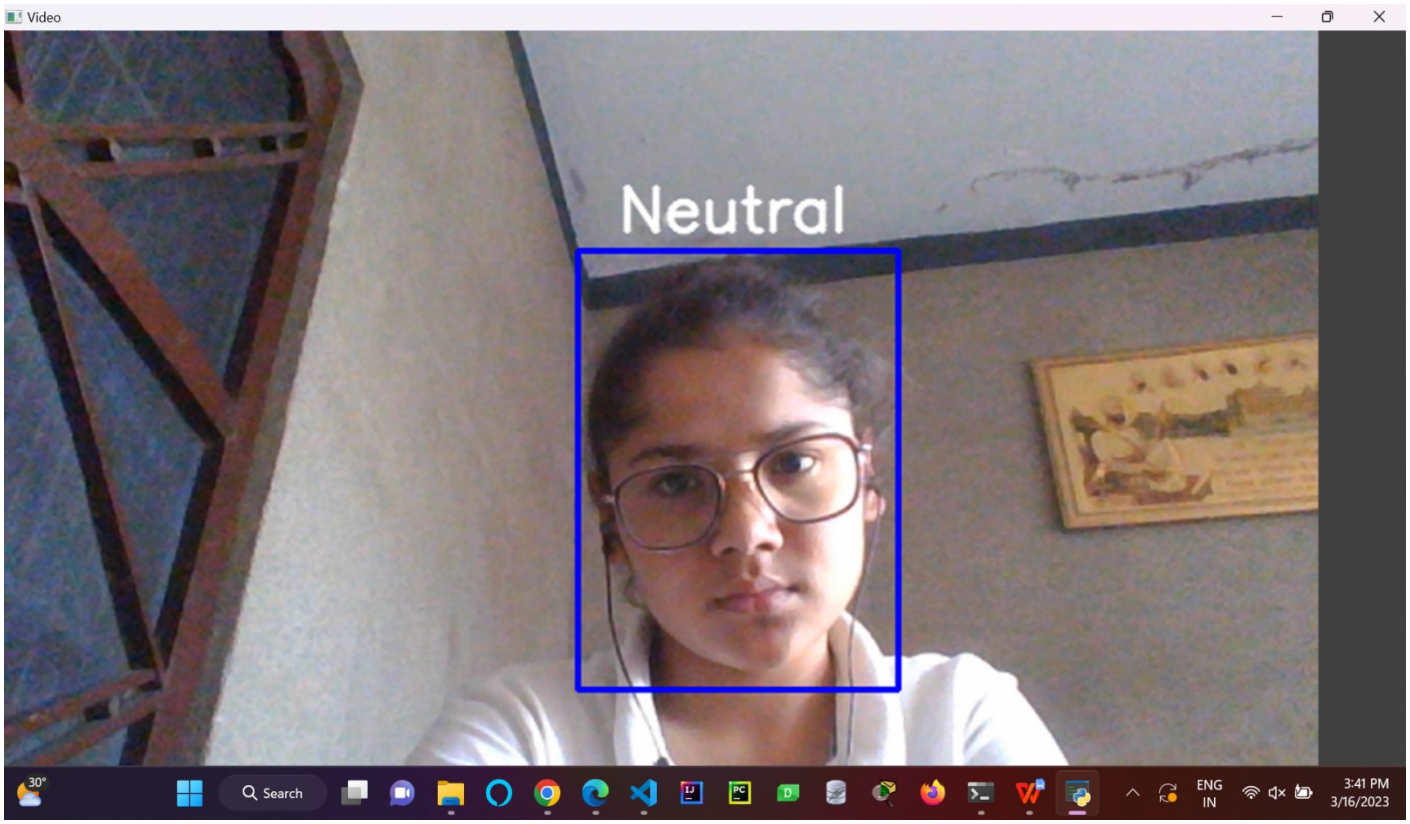
```
45.  def show_vid():
46.      cap1 = cv2.VideoCapture(0)
47.      if not cap1.isOpened():
48.          print("cant open the camera1")
49.      flag1, frame1 = cap1.read()
50.      frame1 = cv2.resize(frame1,(600,500))
51.
52.      bounding_box = cv2.CascadeClassifier('/home/shivam/.local/lib
53.      gray_frame = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
54.      num_faces = bounding_box.detectMultiScale(gray_frame,scaleFac
55.
56.      for (x, y, w, h) in num_faces:
57.          cv2.rectangle(frame1, (x, y-50), (x+w, y+h+10), (255, 0,
58.          roi_gray_frame = gray_frame[y:y + h, x:x + w]
59.          cropped_img = np.expand_dims(np.expand_dims(cv2.resize(ro
60.          prediction = emotion_model.predict(cropped_img)
61.
62.          maxindex = int(np.argmax(prediction))
63.          cv2.putText(frame1, emotion_dict[maxindex], (x+20, y-60),
64.          show_text[0]=maxindex
65.      if flag1 is None:
66.          print ("Major error!")
67.      elif flag1:
68.          global last_frame1
69.          last_frame1 = frame1.copy()
70.          pic = cv2.cvtColor(last_frame1, cv2.COLOR_BGR2RGB)
71.          img = Image.fromarray(pic)
72.          imgtk = ImageTk.PhotoImage(image=img)
73.          lmain.imgtk = imgtk
74.          lmain.configure(image=imgtk)
75.          lmain.after(10, show_vid)
```

```
            print ("Major error!")
        elif flag1:
            global last_frame1
            last_frame1 = frame1.copy()
            pic = cv2.cvtColor(last_frame1, cv2.COLOR_BGR2RGB)
            img = Image.fromarray(pic)
            imgtk = ImageTk.PhotoImage(image=img)
            lmain.imgtk = imgtk
            lmain.configure(image=imgtk)
            lmain.after(10, show_vid)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            exit()


def show_vid2():
    frame2=cv2.imread(emoji_dist[show_text[0]])
    pic2=cv2.cvtColor(frame2,cv2.COLOR_BGR2RGB)
    img2=Image.fromarray(frame2)
    imgtk2=ImageTk.PhotoImage(image=img2)
    lmain2.imgtk2=imgtk2
    lmain3.configure(text=emotion_dict[show_text[0]],font=('arial

    lmain2.configure(image=imgtk2)
    lmain2.after(10, show_vid2)

if __name__ == '__main__':
    root=tk.Tk()
    img = ImageTk.PhotoImage(Image.open("logo.png"))
    heading = Label(root,image=img,bg='black')

    heading.pack()
```

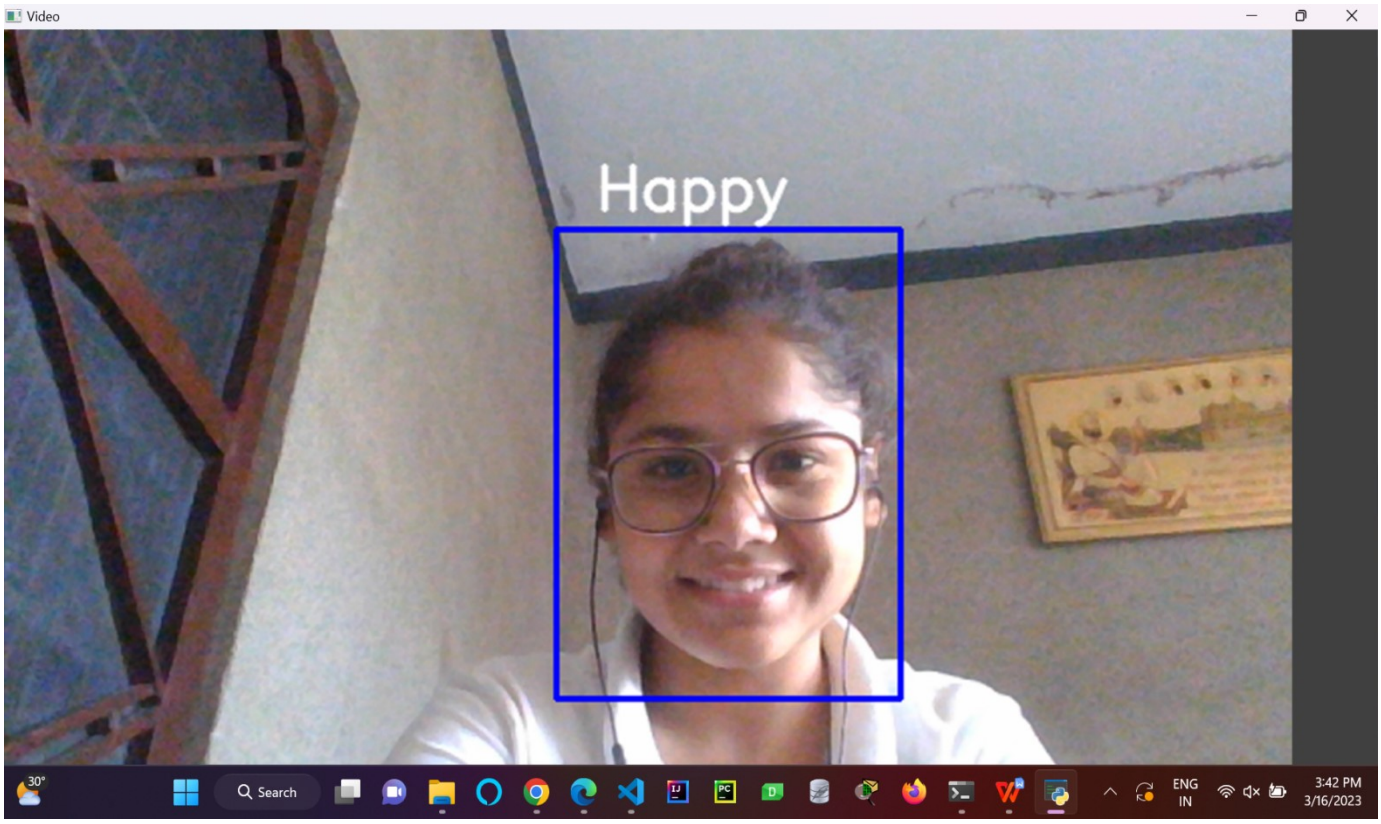
```
90.
91.  if __name__ == '__main__':
92.      root=tk.Tk()
93.      img = ImageTk.PhotoImage(Image.open("logo.png"))
94.      heading = Label(root,image=img,bg='black')
95.
96.      heading.pack()
97.      heading2=Label(root,text="Photo to Emoji",pady=20, font=('ari
98.
99.      heading2.pack()
100.     lmain = tk.Label(master=root,padx=50,bd=10)
101.     lmain2 = tk.Label(master=root,bd=10)
102.
103.     lmain3=tk.Label(master=root,bd=10,fg="#CDCDCD",bg='black')
104.     lmain.pack(side=LEFT)
105.     lmain.place(x=50,y=250)
106.     lmain3.pack()
107.     lmain3.place(x=960,y=250)
108.     lmain2.pack(side=RIGHT)
109.     lmain2.place(x=900,y=350)
110.
111.
112.
113.     root.title("Photo To Emoji")
114.     root.geometry("1400x900+100+10")
115.     root['bg']='black'
116.     exitbutton = Button(root, text='Quit',fg="red",command=root.d
117.     show_vid()
118.     show_vid2()
119.     root.mainloop()
```

3.2 After training

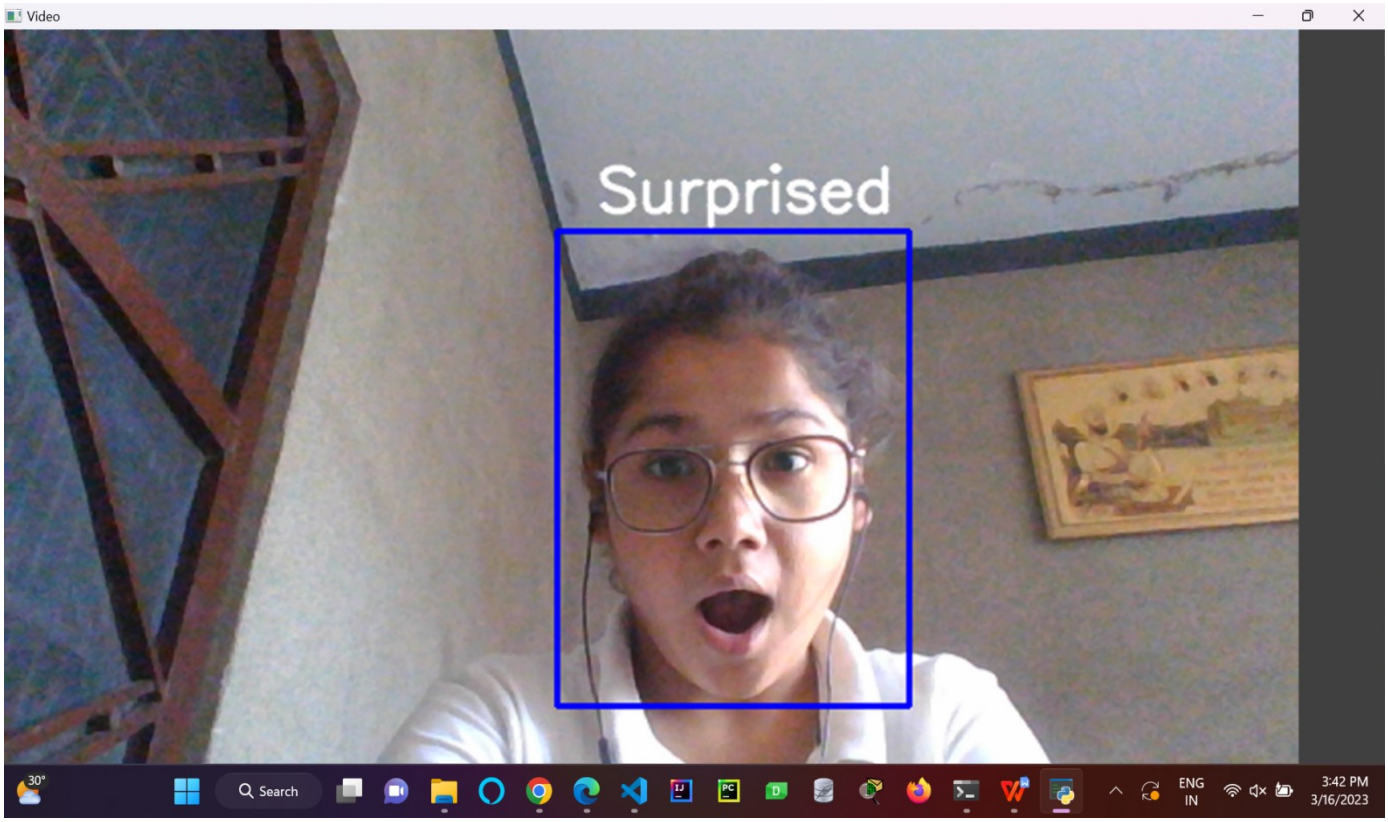
1. Neutral



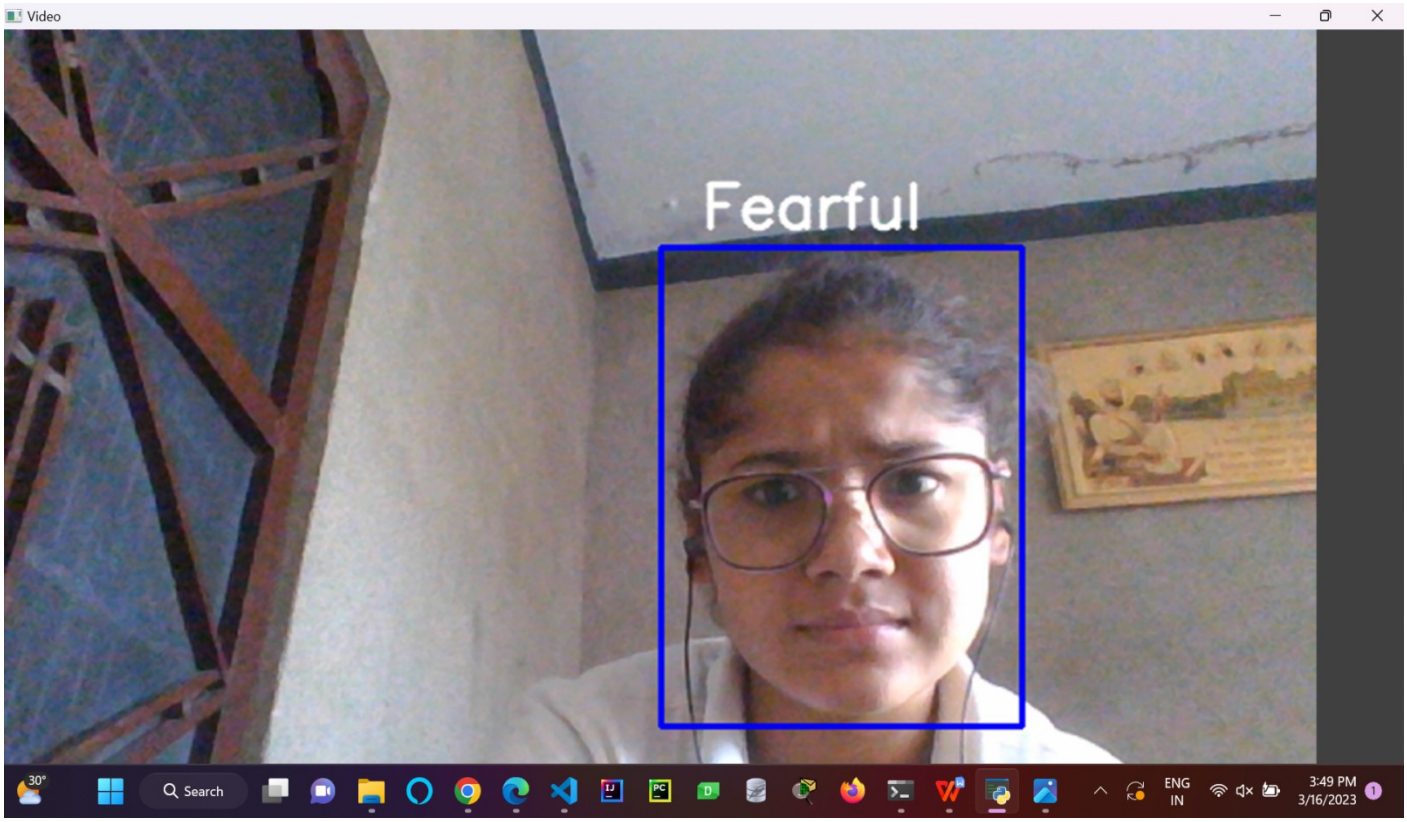
2. Happy



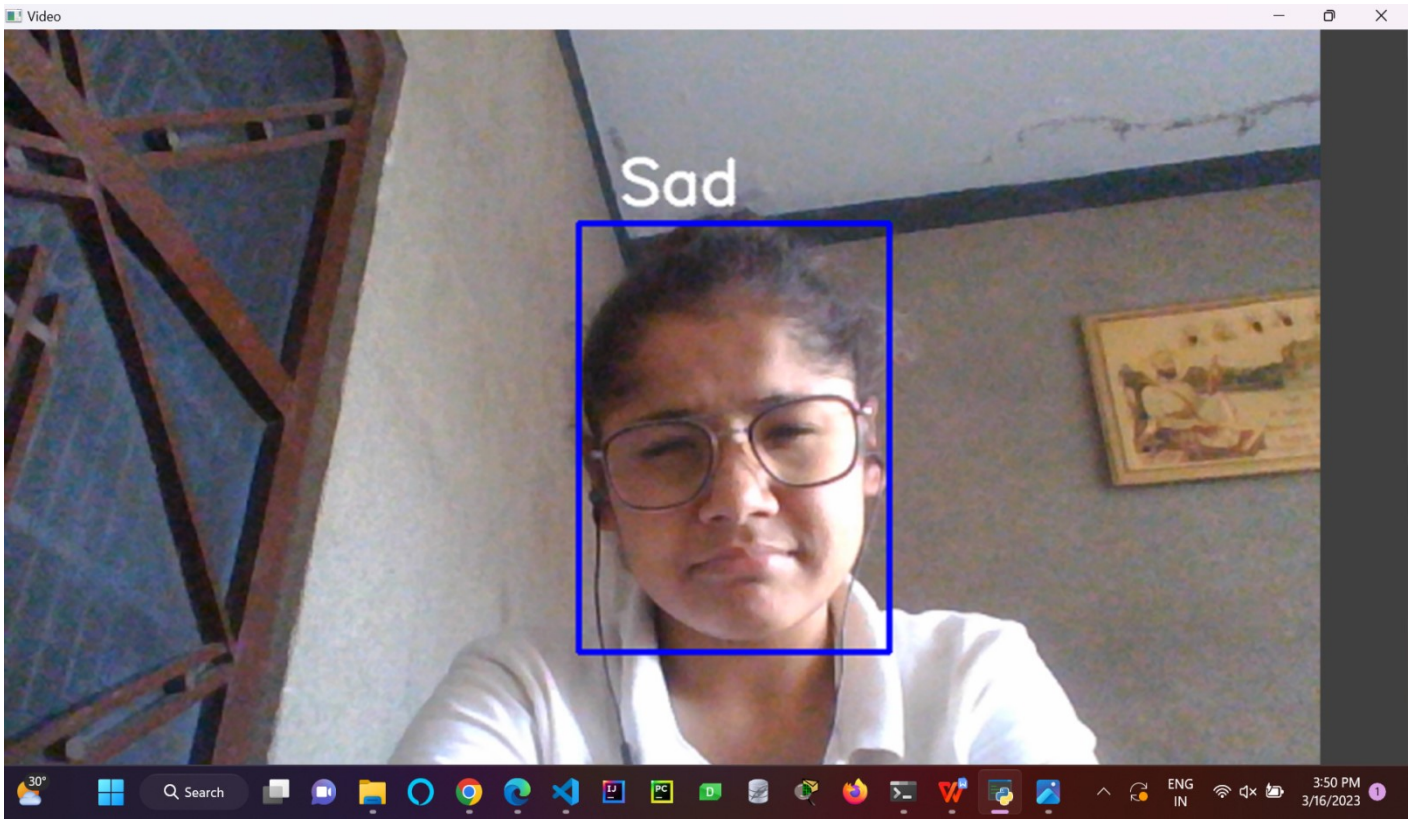
3. Surprised



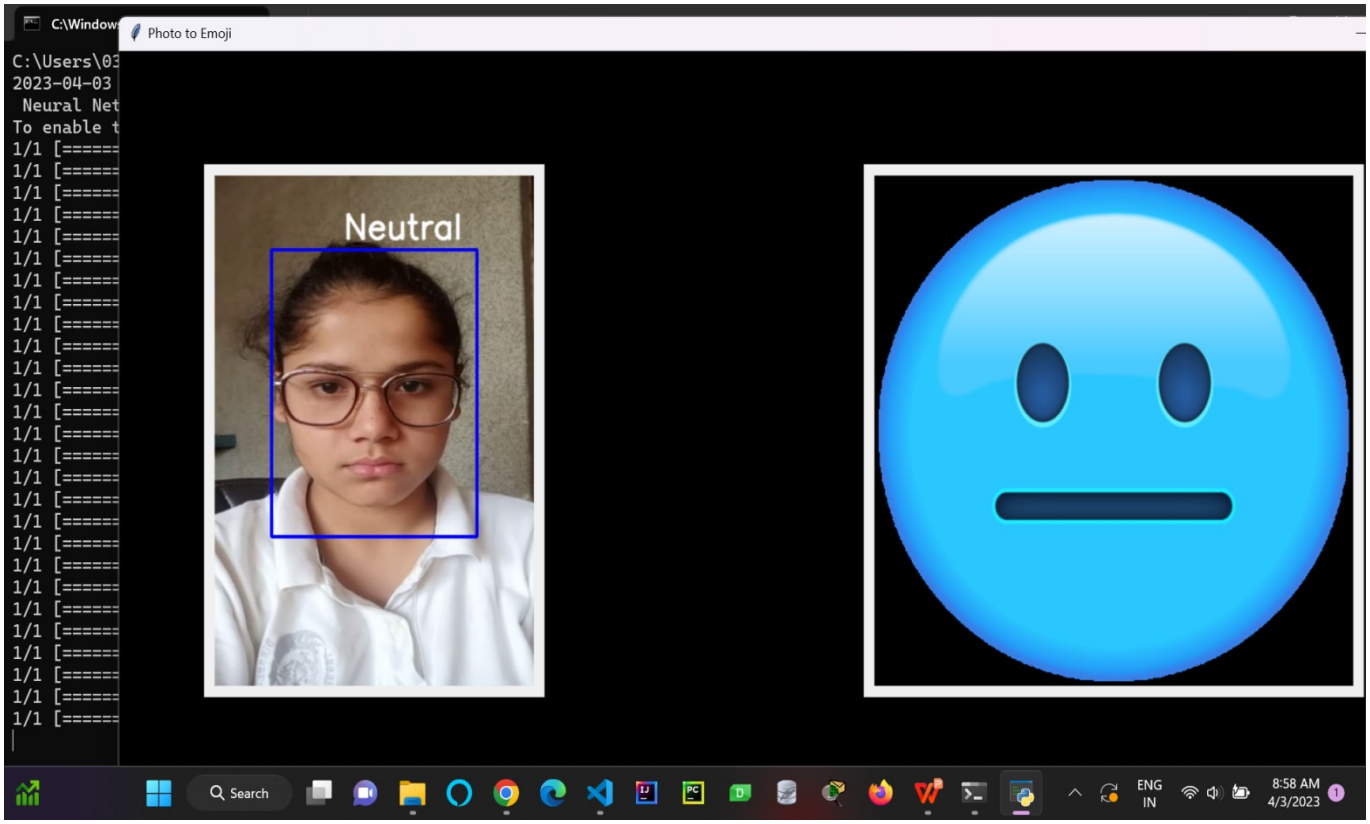
4. Fearful

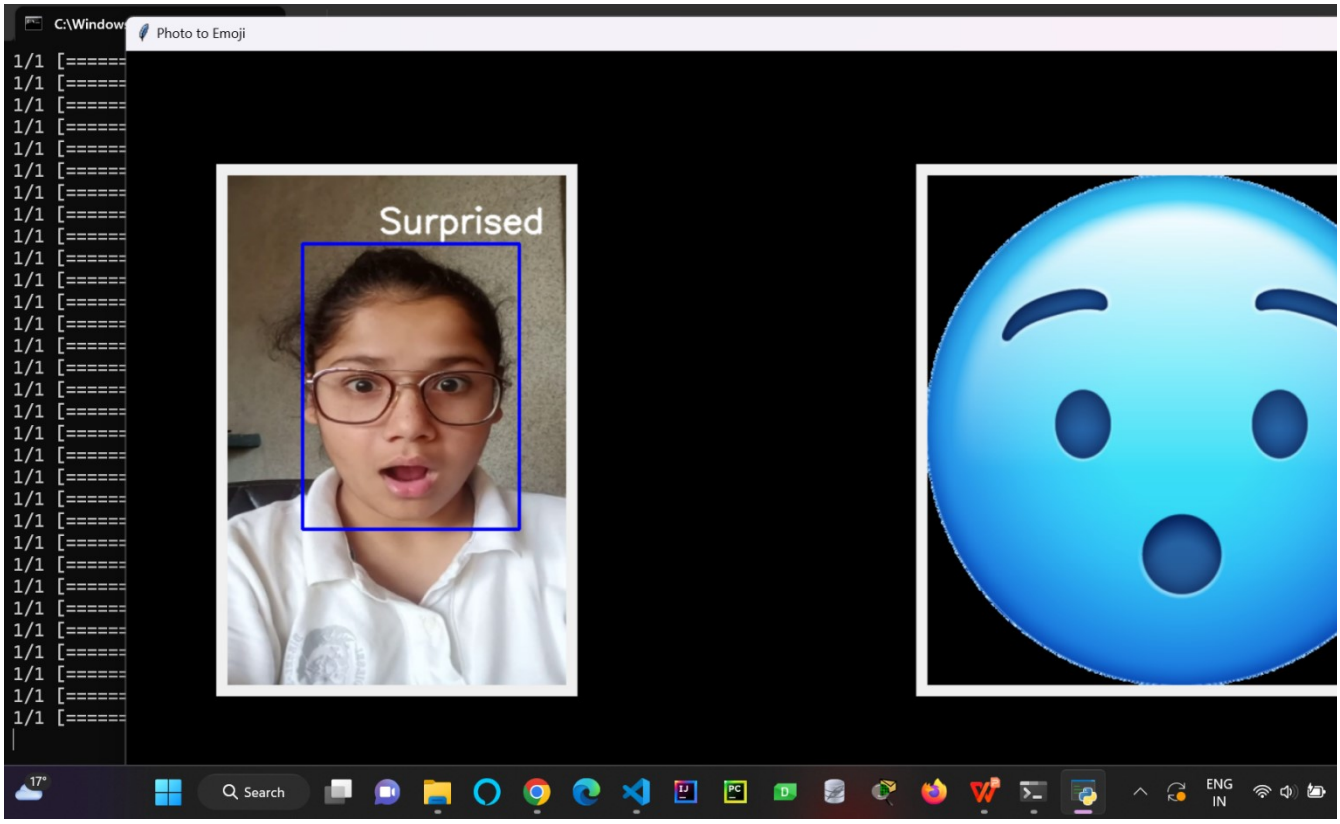


5. Sad



3.3 After integrating emojis with emotion





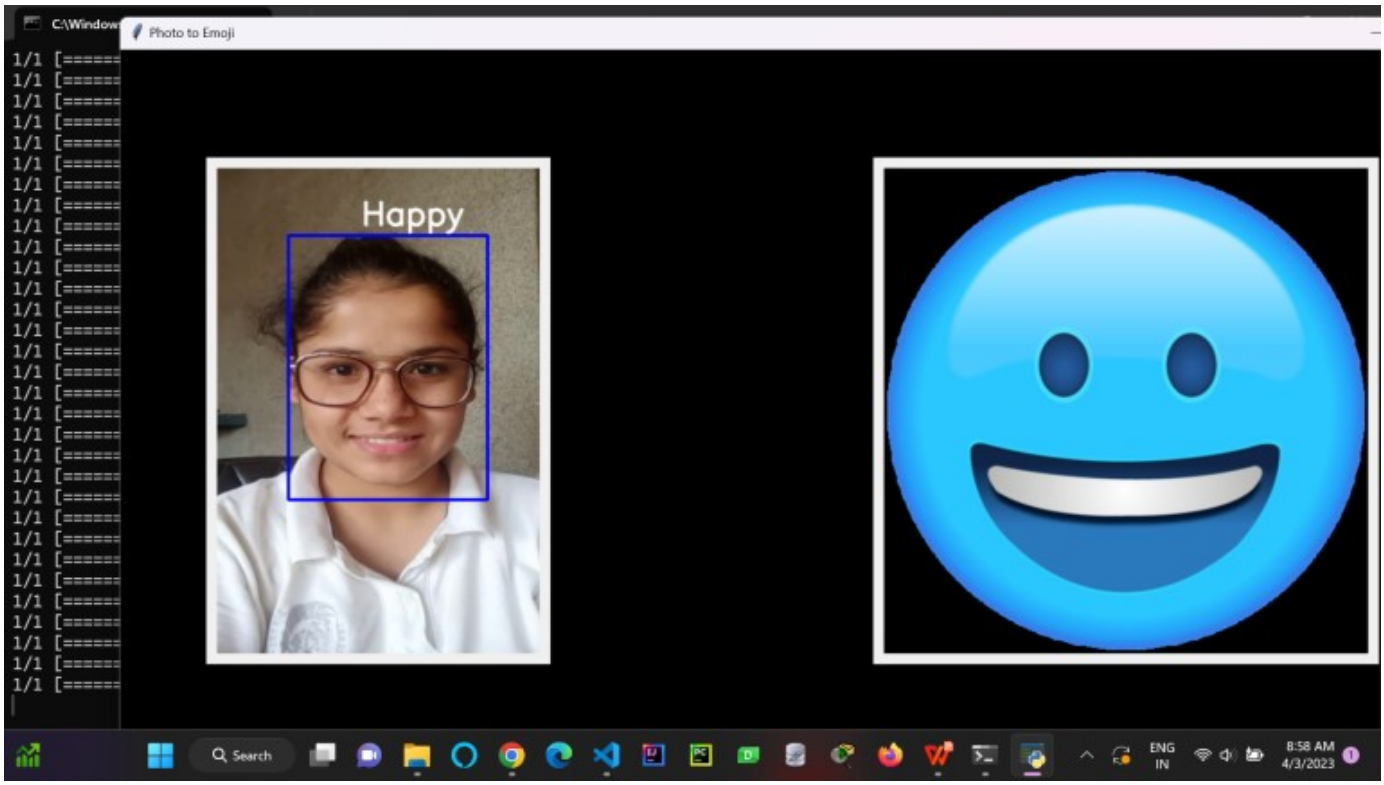


Photo to Emoji
Happy
Search
ENG
IN
8:58 AM
4/3/2023

Chapter 4 – Conclusion

The emojify Project using image is an interesting and innovative project that leverages computer vision and machine learning techniques to detect and classify facial expressions in images, and then generate an appropriate emoji that represents the corresponding emotions. Emojify project using images has the potential to revolutionize digital communication by allowing users to express their emotions more accurately and effectively. The project requires the integration of computer vision and natural language processing techniques to analyze an image, extract relevant features, and map them to the appropriate emoji. While there have been several studies and research on the use of emojis in natural language processing and computer vision tasks, there is still much to be explored in the field of emojify project.

The literature survey has provided an overview of the current state of the art in emojify projects using images. The survey revealed that existing research has focused on developing methods for learning vector representations of emojis using textual descriptions and for predicting emojis in social media posts.

However, there are still several challenges that need to be addressed in emojify projects, including improving the accuracy and robustness of image analysis algorithms, developing better methods for mapping image features to emojis, and expanding the set of available emojis. Future research can also explore use of multi modal techniques, such as combining facial expressions and body language, to improve the accuracy of emotion recognition in emojify projects.

Chapter 5 - Literature Survey

In this literature survey, we will explore the existing research and techniques used in emojify projects that use images. We will review the most recent and relevant papers and articles that have investigated this topic. The survey aims to provide an overview of the current state-of-art and identify gaps in the research that need further exploration. Ultimately, the goal of this literature survey is to provide a comprehensive guide to researchers interested in the Emojify-Create your own emoji project.

[1] "EmojiNet: A machine readable sense inventory for Emoji" by Sanjaya Wijeratne et al. This paper introduces EmojiNet, a machine-readable sense inventory for emoji that can be used to improve emoji-related NLP tasks.

[2] "Emoji2Vec: Learning Emoji Representations from their Description" by Bjarke Felbo et al. This paper proposes Emoji2Vec, a method for learning vector representations of emoji using their textual descriptions. The authors show that these representations can be used to perform emoji prediction in social media posts.

[3] "Emoji prediction using a hybrid neural network" by Muhammad Alshahrani at al. This paper proposes a hybrid neural network model that combines convolutional neural networks (CNNs) and long short -term memory(LSTM) networks for emoji prediction in tweets.

[4] "Emoji-CNN: Learning Emojis Representation with convolutional neural networks" by Jialong Tang et al. This paper proposes Emoji-CNN, a method for learning emoji representations using convolutional neural networks. The authors show that these representations can be used to improve sentiment analysis on social media posts.

[5] “Emoji-based emotion recognition using convolutional neural network and bidirectional LSTM” by Jing Chen et al. This paper proposes a method for emotion recognition in tweets using emoji-based features and combination of convolutional neural networks and bidirectional LSTM networks.

[6] “Multi-modal Emotion Recognition in the Wild” by Fabien Ringeval et al. This paper presents the results of the EmoReact challenge, which aimed to recognize emotions in videos using both audio and visual cues, including facial expressions and body language.

A) Existing System

There are existing systems and application for emojify that use image recognition techniques to assign the most appropriate emoji to an input image. Here are some examples:

1. **Emojer:** Emojer is a mobile application that allows users to take a photo or upload an existing photo and receive a suggested emoji based on the content of the image. The application uses deep learning algorithms to recognize the objects, people, and scenes in the image and match them with the most appropriate emoji.

2. **Emoji Scavenger Hunt:** Emoji Scavenger Hunt is a web-based game that challenges users to find real-world objects that match a given emoji within a certain amount of time. The game uses machine learning to recognize the objects in the user's camera feed and assign the appropriate emoji.

B) Proposed System

The proposed architecture for the system is a convolutional neural network that is pre-trained on a large dataset of images, such as FER-2013 dataset. The pre-trained CNN would serve as the feature extractor for the input image, and the extracted features would be passed through a fully connected layer to predict the most appropriate emoji. The training data for the model would consist of a large dataset of images and their corresponding emojis. The model would be trained using unsupervised learning techniques. The performance of the model would be evaluated using metrics such as accuracy.

5.1. Literature Review Summary:

Table-5.1: Literature summary

Year and Citation	Article/ Author	Tools/ Software	Technique	Source	Evaluation Parameter
2020	"EmojifyGAN: A Conditional Generative Adversarial Network for Emoji Generation from Images" by S. Roy	EmojifyGAN, a cGAN-based approach for generating emojis from input images.	The author shows that their model can generate high-quality emojis that match the input image's style and emotion.	Dataset	Accuracy, Speed, Diversity, Robustness.
2019	"Emojify: A Genrative Model for Emojis with Personalized Styles" by Y.Zhang	A generative model for creating emojis with personalized styles.	The author used a conditional generative adversarial network to generate emojis that match the input image's style and emotion.	Dataset	Accuracy, Diversity, Robustness, User satisfaction.
2019	"Emojify: A Genrative Model for Emojis with Personalized Styles" by Y.Zhang	A generative model for creating emojis with personalized styles.	The author used a conditional generative adversarial network to generate emojis that match the input image's style and emotion.	Dataset	Accuracy, Diversity, Robustness, User satisfaction.

2017	"EmojiNet: A Machine Learning Approach for Emoji Prediction" by B.Felbo	EmojiNet, a deep learning model that predicts relevant emojis for input text.	The author train their model on a large dataset of text and emojis and show that it outperforms the existing methods.	Dataset	Accuracy, Speed, Diversity.
2017	"Deep Emoji Classification" by Z.Yang	This paper proposes a deep learning bases approach from emoji classification.	The author uses a Convolutional neural network to extract features from the input image and then classify it into one of the several emoji categories.	Dataset	Accuracy, Speed, Robustness.

Chapter 6 - EXPERIMENTAL SETUP

This is how we will approach the problem:

- Data collection: Collect a dataset of images that includes faces or expressions that you want to emojify. This dataset includes faces or expressions and emotions to train your model to recognize and emojify.

- Data preprocessing: Preprocess the images in the dataset by resizing them to a consistent size and normalizing the pixel values. You may also need to perform data augmentation techniques such as rotation, flipping, and zooming to increase the size of your dataset.

- Model Selection: Choose a model that is suitable for image classification tasks. For example, you can use a convolutional neural network (CNN) with a pre-trained model such as ResNet or VGG.

- Model training: Train your model using the preprocessed dataset. Use a validation set to monitor the performance of your model and avoid overfitting.

- Emojify output selection: Choose the set of emojis that you want to use for emojifying the images. This set should include emojis that correspond to the different expressions and emotions in your dataset.

- Emojify Algorithm: Develop an algorithm that maps the predicted emotion from your trained model to the corresponding emoji in your chosen emoji set.

- Testing and evaluation: Test your model on a separate test set of images and evaluate its performance using metrics such as accuracy, precision, and recall. You can also perform qualitative analysis by visually inspecting the emojified images.

- Deployment: Deploy your model and algorithm as a web application or mobile app, allowing users to upload their images and receive and emojified version of the image.

Bibliography

Andral M. and Larroque A. 2016. “The emojis consumer perception in the online advertising” Halmstad University Bachelor’s Valiation

Bliss-Carroll N.L. 2016. “The Nature, Function, and Value of Emojis as Contemporary Tools of Digital Interpersonal Communication” Doctoral Dissertation, Gardner-Webb University

Chairunnisa S. and Benedictus A. S. 2017. “Analysis of Emoji and Emoticon Usage in Interpersonal Communication of Blackberry Messenger and WhatsApp Application User” International Journal of Social Sciences and Management Volume 4, Issue 2: 120-126

Eisenstein, J. and Pavalanathan, U. (2015). “Emoticons vs. Emojis on Twitter: A Casual Inference Approach” CoRR abs/1510.08480

Gullberg K. 2016. “Laughing Face with Tears of Joy: A Study of the Production and Interpretation of emojis among Swedish University Students” Bachelor Thesis, Lund University

Kyle L. K., Malone S. A. and Wall H. J. 2017. “Emojis: Insights, Affordances and Possibilites for Psychological Science” Trends in Cognitive Sciences Volume 21, Issue 2 : 66-68

Kelly, R. and Watts, L. (2015). “Characterising the Inventive Appropriation of Emoji as Relationall [67] ningful in Mediated Close Personal Relationships” Experiences of Technology Appropriation: Unanticipated Users, Usage, Circumstances, and Design

REFERENCES

[1] Lu X., Ai W., Liu X., Li Q., Wang N., Huang G. and Mei Q. 2016. "Learning from the Ubiquitous Language: An Empirical Analysis of Emoji Usage of Smartphone Users" Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing : 770-780 Emojis: The Language of the future? – A Literature Review 139

[2] Negishi, M. (2014). "Meet Shigetaka Kurita, The Father of Emoji" The Wall Street Journal https://blogs.wsj.com/japanrealtime/2014/03/26/meet-shigetakakurita-the-father-of-emoji/ (Accessed September 1, 2017)

[3] Novak, P. K., Smailović, J., Sluban, B. and Mozetiĉ, I. (2015). "Sentiment of Emojis" CoRR abs/1509.07761

[4] Pele A. 2016. "From English to Emojis: A New, Simpler, Digital Language" The 11th International Conference on Virtual Learning ICVL 2016 :396-401

[5] Stark, L. and Crawford, K. (2015). "The Conservatism of Emoji: Work, Affect and Communication" Social Media+Society 1.2: 2056305115604853

[6] Tian Y., Galery T., Dulcinati G., Molimpakis E. and Sun C. 2017. "Facebook Sentiment: Reactions and Emojis" Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media 10.18653/v1/W17-1102 : 11-16

[7] Walther, J. B. and D'Addario, K. P. (2001). "The Impacts of Emoticons on Message Interpretation in Computer-Aided Communication" Social Science Review Volume 19, Issue 3: 324-327

[8] Zhou, R., Hentschel, J. and Kumar N. (2017). “Goodbye Text, Hello Emoji: Mobile Communication on WeChat in China” Proceedings of the 2017 CHI Conference on Human Factors in Computer Systems :748-759

[9] Zhu, X. (2015). “A Symbolism Study of Expression in Text-Based Communication” Iowa State University Graduate Theses and Dissertations