



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Emojify: A Deep Learning Approach For Custom Emoji Creation And Recognition

Author Info

1) Venkata Ravi Kiran Kolla
Sr. Software Engineer
Department of Information Technology
USA

Abstract:- Emojis are an inevitable record that will grow over the rest of the year, from marketing, especially virtual verbal interactions, to recovering statistics related to sentiment assessment and perspective mining. Making semantic exceptions allows people to make certain emotions and identities more “real”. Emojis are also used in the comment form. Comment form fees with emojis are more than just different strategies for commenting. The content of the text and the emotion expressed by its severity are modified using emojis. In fact, emoji can be used to simulate facial gestures in informal text communication (ITC) for certain emotions such as sarcasm, sarcasm, and non-text humor. Emojis allow customers to choose from an extensive list and are a way of indicating non-verbal cues. Emojis are small images often included in text messages on social media. Combining visual and textual content in the same message forms a modern way of communicating. Emojis, or avatars, are a way of displaying non-verbal cues. These cues have become essential in online chats, product reviews, brand sentiment, and more. It has also led to an increase in data science research focused on storytelling with emojis. Advances in computer vision and deep learning have made it possible to detect human emotions from images. This deep learning project classifies human facial expressions to filter and map corresponding emojis and avatars. This project isn't meant to solve real problems, but it lets you see the world of chat in a more colorful way. Emojify is dedicated software for creating emojis and avatars. This paper explores the emotional popularity that is the use of facial features via current emoji. In addition, criteria for evaluating facial features and current beliefs about the popularity of facial emotions have been developed. The software created contains her five human emotions: happy, neutral, sad, surprised and neutral. The actual term that can be expressed is the transferred term.

Introduction

People use emoji every day. Emoji have developed a whole new language that allows us to express concepts and emotions well. This visible language is now a trend in verbal communication online, and is now found not on Twitter, but on other major online platforms such as Facebook and Instagram. In today's generation, people tend to use emoticons to talk to each other. So we have an idea to create a personal customized emoji. Emojiiify is a software program that responds to the appearance of emojis and avatars. Neural communities are emerging software in many areas as research declines. This paper is primarily based on devices implementing convolutional neural networks and the Fer2013 dataset to detect emotions from facial expressions and convert them into personalized emojis. Build a neural convolutional community that captures facial emotions. You can train your version on the FER2013 dataset. Then assign the emotion to the corresponding emoji or avatar. Fer2013 contains about 30,000 RGB face images with various expressions, whose length is limited to 48×48 , and their main labels can be categorized into five types: are marked as

0 = sad, 1 = surprised, 2 = scared, 3 = happy, 4 = normal.

METHODOLOGY

- Proposed system:
- Facial emotion recognition with CNN:
 1. Create a CNN architecture.
- Now upload all the required libraries you need for your version. Then initialize the education and validation mills. H. First rescale all images to tell the version, then change them to grayscale images.

- **Imports:**

```
import numpy as np
import cv2
from tensorflow.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
```

- **Build the CNN architecture:**

```

emotion_model = Sequential()

emotion_model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1)))
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='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2, 2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Flatten())
emotion_model.add(Dense(1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(7, activation='softmax'))

```

2. Train your version on the Fer2013 dataset.

This is the place to let the community know about all the images we have. Fer2013 data set and save the weights to the face prediction version. Then with OpenCV, you'll encounter the limited field of view of your webcam and expect emotions.

```

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="grayscale",
    class_mode='categorical')

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

```

- Transfer Train and set records.
- Emotions

```

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

cur_path=os.path.dirname(os.path.abspath(__file__))

emoji_dist={0:cur_path+"/emojis/angry.png",2:cur_path+"/emojis/disgusted.png",
            2:cur_path+"/emojis/fearful.png",3:cur_path+"/emojis/happy.png",
            4:cur_path+"/emojis/neutral.png",5:cur_path+"/emojis/sad.png",
            6:cur_path+"/emojis/surpriced.png"}

```

```

for (x, y, w, h) in num_faces:
    cv2.rectangle(frame1, (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)

    prediction = emotion_model.predict(cropped_img)

    maxindex = int(np.argmax(prediction))
    cv2.putText(frame1, emotion_dict[maxindex], (x+20, y-60), cv2.FONT_HERSHEY_SIMPLEX, 1,
                (255, 255, 255), 2, cv2.LINE_AA)
    show_text[0]=maxindex

```

1. GUI

```

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
import threading

if __name__ == '__main__':
    frame_number=0
    root=tk.Tk()

    lmain = tk.Label(master=root, padx=50, bd=10)
    lmain2 = tk.Label(master=root, bd=10)
    lmain3=tk.Label(master=root, bd=10, fg="#CDCDCD", bg='black')
    lmain.pack(side=LEFT)
    lmain.place(x=50, y=250)
    lmain3.pack()
    lmain3.place(x=960, y=250)
    lmain2.pack(side=RIGHT)
    lmain2.place(x=900, y=350)

    root.title("Photo To Emoji")
    root.geometry("1400x900+100+10")
    root['bg']='black'
    exitbutton = Button(root, text='Quit', fg="red",
                        command=root.destroy, font=('arial', 25, 'bold')).pack(side = BOTTOM)
    threading.Thread(target=show_subject).start()
    threading.Thread(target=show_avatar).start()

    root.mainloop()

```

•final version.

Few pictures show what this task looks like. This dataset contains the following categories of facial emotions:

0:

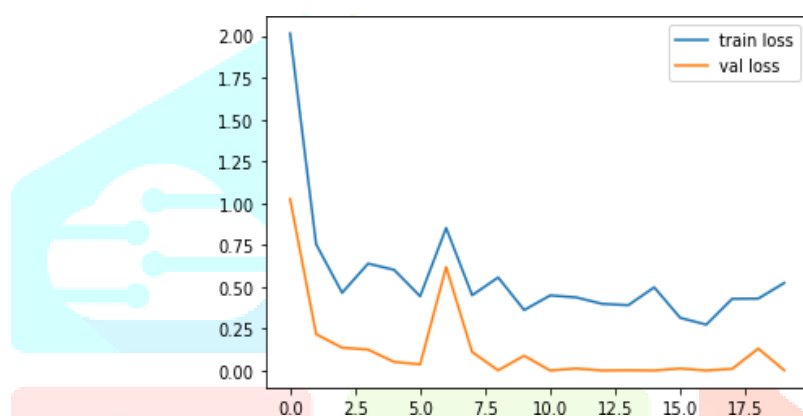
Sad 1:

Surprise 2:

Fear 3:

Happy4:

neutral



➤ GUI development and mapping with pictograms

Create a folder called Emojis and store the emoji corresponding to each of the seven emotions in your dataset. A trained model is tested on a set of images. A random image is introduced into the network and the first tag is compared to the original known tag of the image. The parameters used for scoring are F1 score, precision and recall. Accuracy is the percentage of positive outcomes predicted to be true positives.

• Test



Fig : Loss plot

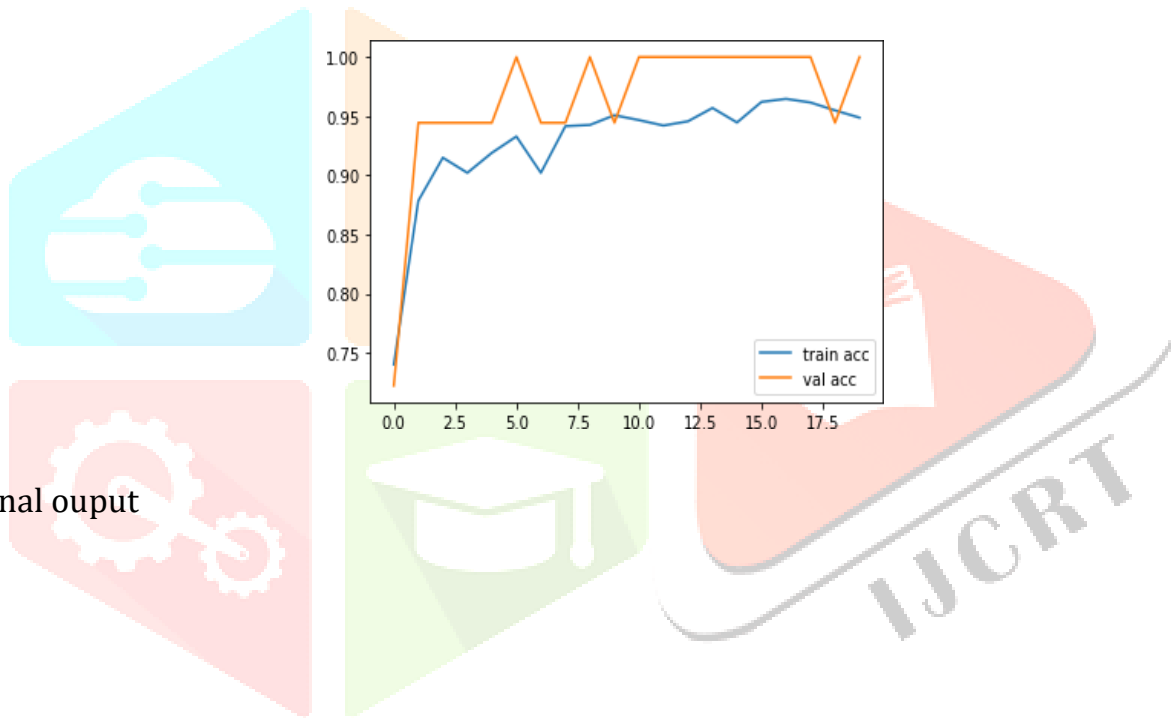
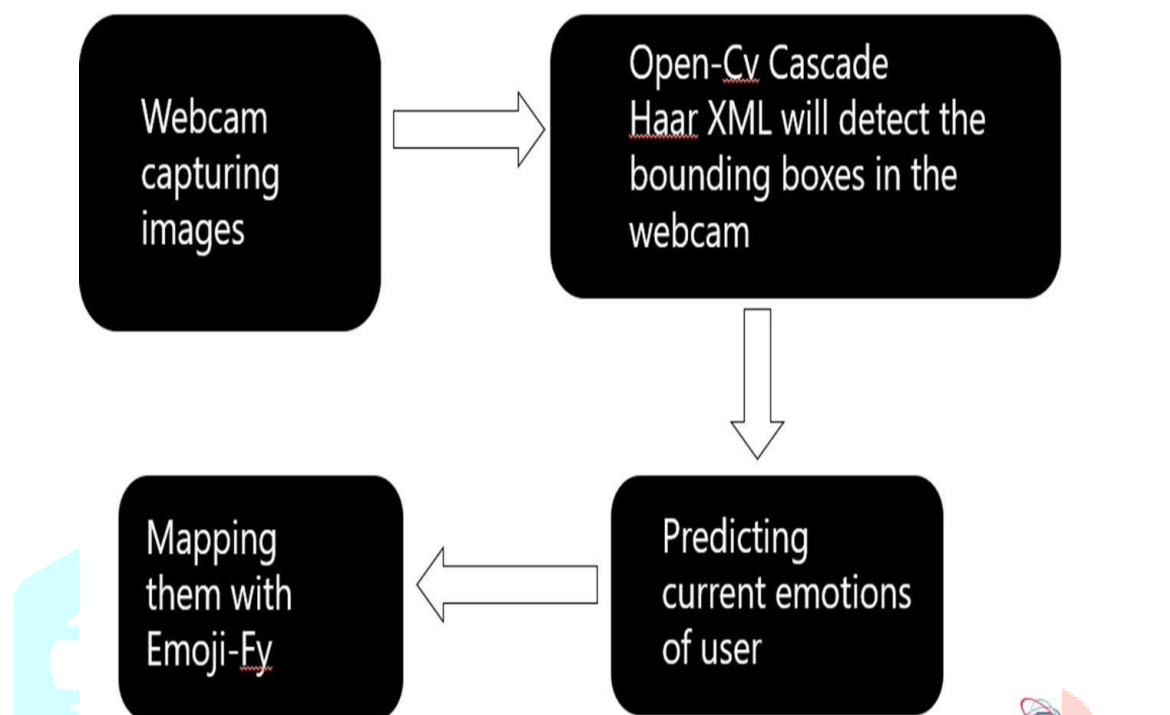


Fig : Final ouput

I. BLOCK DIAGRAM



I. TOOLS USED

We used various libraries related to recording technology such as Keras, TensorFlow, OpenCV and NumPy. A sequential modeling approach was used for the motif in building the Keras version.

II. CONCLUSION

Today's techies prefer a fashion to speak with non-verbal cues like emoticons, so we know why they don't currently ship personalized emojis. Imagination from laptops and improvements in deep learning have made it possible to identify human emotions from images. This Deep His learning task can classify human facial expressions to erase and match corresponding emojis and avatars. The end result we expect is the use of emojis in the chat world. You need someone who speaks with a personally customizable emoticon. This task captures the current emotion and converts the emoji for that emotion so that the consumer receives a face emoji to use when chatting. Advances in computer vision and deep learning have made it possible to detect human emotions in images. This deep learning project classifies human facial expressions to filter and map corresponding emojis and avatars.

The expected result is the use of emojis in the chat world. We want people to be able to communicate with their customizable emoticons. The project recognizes the current emotion and converts that emotion emoji so that customers can get their own face emoji to use when chatting.

The result we are expected is the use of emoji in chatting world. We want people to communicate with their own customisable emoticon. The project will recognize one's current emotion and convert that emotion's emoji so that the customer gets emoji of their face and use it in chatting.

III. REFERENCES

- [1] A Novel Multi-Attribute Face-to-Cartoon Model for Human-Computer Interaction, Chengzhi Cai, Chien-Shiung Wu College, Southeast University, Nanjing, China,
- [2] <https://data-flair.training/blogs/create-emoji-with-deep-learning>
- [3] "Evolution of Deep Convolutional Neural Networks" (2020)
- [4] Milad Mohammad Taghi Zadeh Institut für Elektrotechnik Universität Khatam Teheran, Iran, "The Popularity of Fast Face Emotions Using Convolutional Neural Networks and Gabor Filters" 2019
- [5] Yong Yang Chongqing Key Laboratory of Computational and Intelligence Chongqing University of Posts and Telecommunications Chongqing, China, "Facial Expression Recognition Based on Arousal-Valence Emotion Model and Deep Learning Method" 2019
- [6] "Facial Emotion Recognition on Datasets Using Convolutional Neural Networks" 2017.
- [7] R. Yamashita, M. Nishio, R. Do und K. Togashi "Convolutional Neural Networks: an overview and application in radiology", Insights into Imaging, Vol. 9, No. 4, pp. 611-629, 2018.
- [8] D. Meena and R. Sharan, "Facial Recognition and Approaches to Recognition", International Conference 2016 (ICRAIE), pp. 1-6, 2016
- [9] Expanded Cohn-Kanade database. Available online:
<http://www.consortium.ri.cmu.edu/ckagree/>
- [10] Database of facial expressions of Japanese women. Available online:
<http://www.kasrl.org/jaffe.html>.