Project seminar on

# “EMOTION RECOGNITION USING GABOR FILTERS”


Submitted By

**MAYUR S SHARMA [1DS16CS720]**
**K VISHAL WARRIER [1DS16CS717]**

[Eighth Semester B.E (CSE)]

Under the guidance of

**Dr. Preeti Satish**
Associate Professor
Dept. of CSE
DSCE, Bangalore

# CONTENTS

# ABSTRACT

The emotions evolved in human face have a great influence on decisions and arguments about various subjects. In psychological theory, emotional states of a person can be classified into six main categories: surprise, fear, disgust, anger, happiness and sadness.

The progression of the decades of scientific research has been conducted for developing methods for automated emotion recognition. Now, there is an extensive literature proposing and evaluating various methods, leveraging techniques from multiple fields, such as signal processing, computer vision, machine learning, and speech processing. Given an image of arbitrary size, the job is to identify an emotion of a human face appearing in the image. Face detection in complex environments is disputing since the faces may appear in different scales, different head poses, and orientations. External factors also play a vital role; for instance, the lighting conditions, facial expressions, and shadows are few other sources of variations that need to be taken into account. The approach is to yield a better classification performance implemented in real-world scenarios with fewer exemptions and more generalized accuracy.

A method for detecting facial regions by combining a Gabor filter and a convolutional neural network. The first stage uses the Gabor filter which extracts intrinsic facial features. The second stage of the method concerns the application of the convolutional neural network to these four images. The proposed framework uses the Gabor filters for feature extraction and then a Convolutional Neural Network (CNN) for classification. The experimental results show that the proposed methodology increases both of the speed training process of CNN and the recognition accuracy. The approach yields better classification performance in comparison to the results obtained by the convolutional neural network alone.

# INTRODUCTION

Emotion recognition is a method of identifying human emotion. People vary widely in accuracy at recognizing the emotions of all people. The use of modern technology to help people with emotion identification is a relatively nascent research area. Nowadays, most of the work has been going on in the area of automating the recognition of facial expressions from video, audio, written expressions from the text, and physiology as measured by wearables technology.

The progression of the decades of scientific research has been conducted for developing methods for automated emotion recognition. Now, there is an extensive literature proposing and evaluating various methods, leveraging techniques from multiple fields, such as signal processing, computer vision, machine learning, and speech processing. These different methodologies and techniques are employed to interpret emotions such as Gaussian Mixture Models, Hidden Markov Models and Bayesian networks.

The correctness of emotion recognition is usually increased when it uses the analysis of human expressions from multimodal forms such as texts, physiology, audio, or video. Emotion types are detected through the combination of information from expressions, body movement, gestures, and speech. The upcoming technologies are said to contribute to the emergence of the emotional or emotive Internet.

**STATISTICAL METHOD APPROACH**

This is the primary approach used in this project to identify, classify, and label the emotions.

Statistical methods usually involve supervised machine learning algorithms in which a broad set of interpreted data is used in the algorithms for the computer to learn and predict the relevant emotion types. Machine learning algorithms provide more rational classification accuracy when compared to other strategies, but the challenges in achieving valid results in the exact classification process require an extensive training set.

Some of the most regularly used learning algorithms are:

- Support Vector Machines (SVM),
- Naive Bayes, and
- Maximum Entropy.
- Deep learning is the unsupervised family of machine learning, so it is widely employed in emotion recognition.

Prominent Deep learning algorithms include:

- Artificial Neural Network (ANN) such as Convolutional Neural Network (CNN),
- Long Short-term Memory (LSTM), and
- Extreme Learning Machine (ELM).

The popularity of these deep learning approaches may be mainly attributed to its success in german applications such as in computer vision, Natural Language Processing (NLP), and speech recognition.

# APPLICATIONS

**1. Making Cars Safer and Personalized**

Car manufacturers around the world are increasingly focusing on making cars more personal and safe for us to drive. In their pursuit to build more smart car features, it makes sense for makers to use AI to help them understand the human emotions. Using facial emotion detection smart cars can alert the driver when he is feeling drowsy.Facial Emotion Detection can find subtle changes in facial micro-expressions that precedes drowsiness and send personalized alerts to the driver asking him to stop for a coffee break, change music or temperature.

**2. Facial Emotion Detection in Interviews**

A candidate-interviewer interaction is susceptible to many categories of judgment and subjectivity. Such subjectivity makes it hard to determine whether candidate's personality is a good fit for the job. Identifying what a candidate is trying to say is out of our hands because of the multiple layers of language interpretation, cognitive biases, and context that lie in between. That's where AI comes in, which can measure candidate's facial expressions to capture their moods and further assess their personality traits.With this technology, a recruiter will be able to know, say, the overall confidence level of an interviewee and make a decision about whether or not this candidate will be able to perform well at a client-facing job. Similarly, it will be possible to find whether the candidate is honestly replying to all the questions by measuring the change in emotions during his responses and correlating it the vast amount of knowledge available in this area.

3. Testing for Video Games

During the testing phase, users are asked to play the game for a given period and their feedback is incorporated to make the final product. Using facial emotion detection can aid in understanding which emotions a user is going through in real-time as he is playing without analyzing the complete video manually.Such product feedback can be taken by analyzing a live feed of the user and detecting his facial emotions. While feelings of frustration and anger are commonly experienced in advanced video games, making use of facial emotion detection will help understand which emotions are experienced at what points in the game.

4. Market Research

Traditional market research companies have employed verbal methods mostly in the form of surveys to find the consumers wants and needs. However, such methods assume that consumers can formulate their preferences verbally and the stated preferences correspond to future actions which may not always be right.

5. **Music recommendation** where the suitable music is played according to the persons emotion.

6. **In psychological counselling sessions** where patient can be diagnosed from many sociological and emotional conditions.

# CHALLENGES

1. **Illumination**

A slight change in lighting conditions has always been known to cause a major impact on its results. If the illumination tends to vary, then; even if the same individual gets captured with the same sensor and with an almost identical facial expression and pose, the results that emerge may appear quite different.

2. **Background**

The placement of the subject also serves as a significant contributor to the limitations. A facial recognition system might not produce the same results outdoors compared to what it produces indoors because the factors - impacting its performance - change as soon as the locations change.

3. **Pose**

Facial Recognition Systems are highly sensitive to pose variations. The movements of head or differing POV of a camera can invariably cause changes in face appearance and generate intra-class variations making automated face recognition across pose a tough nut to crack.

4. **Occlusion**

Occlusions of the face such as beard, moustache, accessories (goggles, caps, mask etc.) also meddle with the evaluation of a face recognition system. Presence of such components make the subject diverse and hence it becomes difficult for the system to operate in a non-simulated environment.

# MOTIVATION

Being a multidisciplinary project, motivation for this project came from many areas:

1. Emotion recognition is a hot topic of discussion in image processing and in the broad field of Neural Network.

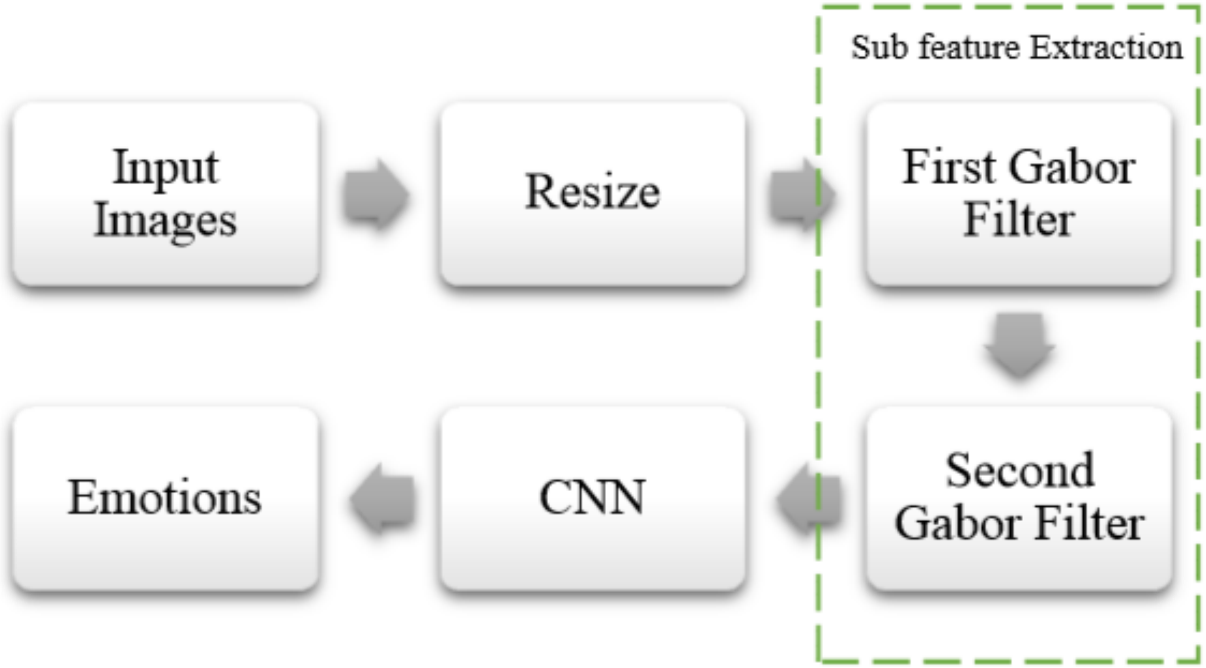
2. The project being a learning curve in areas like Image Processing, AI, and Team management which could be of a potential value for our future endeavours.

3. Probability of the project being incorporated into/with another project from our fellow peers and made into a project with real time applications.

4. Explore our areas of interest in the workings Artificial Intelligence and Neural Network.

# BLOCK DIAGRAM

## SYSTEM OVERVIEW




The progress of the system is first to apply a Gabor filter to the images and then dispatch the output results as inputs to the neural network. The output of the Gabor filter is given to the convolutional neural network.

The Gabor filters extract the essential features by plotting down points at strategic locations on the face and measuring the distance between these points to determine the emotion expressed ultimately.

# METHODOLOGY

The main methods used here are:

1. Haar Cascade Classifier
2. Gabor Filters
3. Convoluted Neural Nerworks

**1. Haar Cascade Classifier**

Haar Cascade is a machine learning object detection algorithm used to identify objects in an image or video. It is a machine learning algorithm based on the approach where a cascade function is trained from positive and negative images. Later afterward, it can be used to detect objects in other images.

**2. Gabor Filters**

Gabor filter are used in texture analysis with edge detection
and feature extraction. When it is applied to an image it returns the highest response at edges and points where texture changes.

The Gabor wavelet id used which is an equation maintaining a real part and an imaginary part. For obtaining 2 filters, we pass two different values each time.

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma)=\exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)\exp\left(i\left(2\pi\frac{x'}{\lambda}+\psi\right)\right)$$

in which the real part is:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma)=\exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\psi\right)$$

and the imaginary part is:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma)=\exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)\sin\left(2\pi\frac{x'}{\lambda}+\psi\right)$$

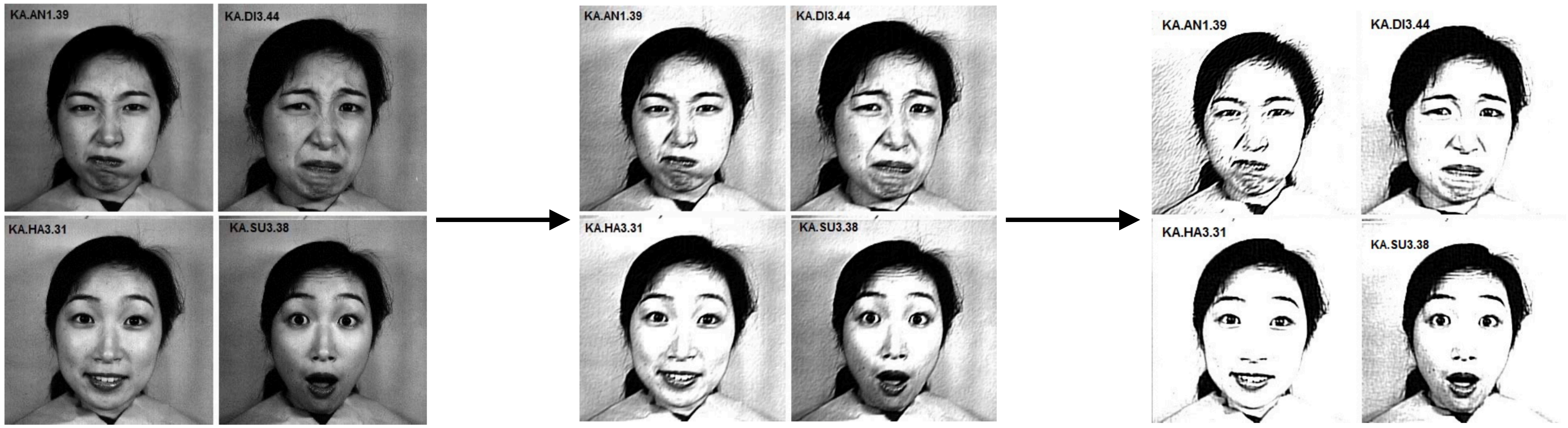
where:

$$x'=x\cos\theta+y\sin\theta$$

and

$$y'=-x\sin\theta+y\cos\theta$$

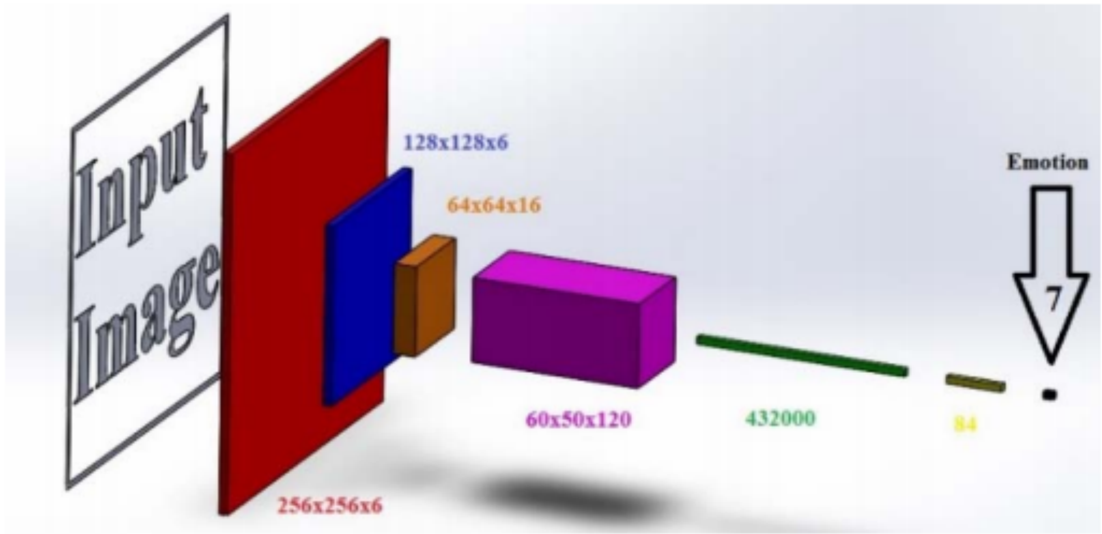
Following set of images shows how the filtering works:




### 3. Convoluted Neural Network

In brief, the main functions applied in the network are:

- MaxPooling Function
- Flatten Function
- Softmax Function
- Rectified Linear Unit (Relu) function



# Brief Code Explanation

**real_time_video.py**

```
from keras.preprocessing.image import img_to_array
import imutils
import cv2
from keras.models import load_model
import numpy as np

# parameters for loading data and images
detection_model_path =
    'haarcascade_files/haarcascade_frontalface_default.xml'
emotion_model_path = 'models/_mini_XCEPTION.106-0.65.hdf5'
```

```
# starting video streaming
cv2.namedWindow('your_face')
camera = cv2.VideoCapture(0)
while True:
    frame = camera.read()[1]
    #reading the frame
    frame = imutils.resize(frame,width=400)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces =
        face_detection.detectMultiScale(gray,scaleFactor=1
        .1,minNeighbors=5,minSize=(30,30),flags=cv2
        .CASCADE_SCALE_IMAGE)
```

```
    canvas = np.zeros((250, 300, 3), dtype="uint8")
    frameClone = frame.copy()
    if len(faces) > 0:
        faces = sorted(faces, reverse=True,
        key=lambda x: (x[2] - x[0]) * (x[3] - x[1]))[0]
        (fX, fY, fW, fH) = faces
                    # Extract the ROI of the face from the grayscale
                        image, resize it to a fixed 48x48 pixels,
                        and then prepare
            # the ROI for classification via the CNN
        roi = gray[fY:fY + fH, fX:fX + fW]
        roi = cv2.resize(roi, (48, 48))
        roi = roi.astype("float") / 255.0
        roi = img_to_array(roi)
        roi = np.expand_dims(roi, axis=0)
```

```
        preds = emotion_classifier.predict(roi)[0]
        emotion_probability = np.max(preds)
        label = EMOTIONS[preds.argmax()]
```

```
    for (i, (emotion, prob)) in enumerate(zip(EMOTIONS, preds)):
                # construct the label text
                text = "{}: {:.2f}%".format(emotion, prob * 100)
                w = int(prob * 300)
                cv2.rectangle(canvas, (7, (i * 35) + 5),
                (w, (i * 35) + 35), (0, 0, 255), -1)
                cv2.putText(canvas, text, (10, (i * 35) + 23),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45,
                (255, 255, 255), 2)
                cv2.putText(frameClone, label, (fX, fY - 10),
                cv2.FONT_HERSHEY_SIMPLEX, 0.45, (0, 0, 255), 2)
                cv2.rectangle(frameClone, (fX, fY), (fX + fW, fY +
                    fH),
                              (0, 0, 255), 2)

    cv2.imshow('your_face', frameClone)
    cv2.imshow("Probabilities", canvas)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

camera.release()
cv2.destroyAllWindows()
```

## train_emotion_classifer.py

```
from keras.callbacks import CSVLogger, ModelCheckpoint, EarlyStopping
from keras.callbacks import ReduceLROnPlateau
from keras.preprocessing.image import ImageDataGenerator
from sklearn.model_selection import train_test_split
from keras.layers import Activation, Convolution2D, Dropout, Conv2D
from keras.layers import AveragePooling2D, BatchNormalization
from keras.layers import GlobalAveragePooling2D
from keras.models import Sequential
from keras.layers import Flatten
from keras.models import Model
from keras.layers import Input
from keras.layers import MaxPooling2D
from keras.layers import SeparableConv2D
from keras import layers
from keras.regularizers import l2
import pandas as pd
import cv2
import numpy as np
```

```
dataset_path = 'fer2013/fer2013.csv'
image_size=(48,48)
```

```
def load_fer2013():
    data = pd.read_csv(dataset_path) #reads the dataset
        pixels = data['pixels'].tolist() #enumerates a new list
        width, height = 48, 48 #frame size
    faces = []
    for pixel_sequence in pixels:
        face = [int(pixel) for pixel in pixel_sequence.split(' ')]
        face = np.asarray(face).reshape(width, height)
        face = cv2.resize(face.astype('uint8'),image_size)
        faces.append(face.astype('float32'))
```

```
# base
img_input = Input(input_shape)
x = Conv2D(8, (3, 3), strides=(1, 1),
    kernel_regularizer=regularization, use_bias=False)(img_input)
x = BatchNormalization()(x)
x = Activation('relu')(x)
x = Conv2D(8, (3, 3), strides=(1, 1),
    kernel_regularizer=regularization, use_bias=False)(x)
x = BatchNormalization()(x)
x = Activation('relu')(x)
```

| Layer type | Details | Output Shape |
|---|---|---|
| Conv | Conv (6x6) | 256, 256, 6 |
| Activation | Relu | 256, 256, 6 |
| MaxPooling | Pool size (2,2) | 128, 128, 6 |
| Conv | Conv (16x16) | 128, 128, 16 |
| Activation | Relu | 128, 128, 16 |
| MaxPooling | Pool size (2,2) | 64, 64, 16 |
| Conv | Conv (120x120) | 60, 60, 120 |
| Activation | Relu | 60, 60, 120 |
| Dropout | -------------------- | 60, 60, 120 |
| Flatten | Flatten to a vector | 432000 |
| Dense | Input →84 | 84 |
| Activation | Relu | 84 |
| Dropout | -------------------- | 84 |
| Dense | Input → Classese Num =7 | 7 |
| activation | softmax | 7 |

```
x = BatchNormalization()(x)
x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
x = layers.add([x, residual])
x = Conv2D(num_classes, (3, 3), padding='same')(x)
x = GlobalAveragePooling2D()(x)
output = Activation('softmax',name='predictions')(x)
```

```
# callbacks : function done during train,test,predict
log_file_path = base_path + '_emotion_training.log'
csv_logger = CSVLogger(log_file_path, append=False)
early_stop = EarlyStopping('val_loss', patience=patience)
reduce_lr = ReduceLROnPlateau('val_loss', factor=0.1,
    patience=int(patience/4), verbose=1)
trained_models_path = base_path + '_mini_XCEPTION'
model_names = trained_models_path + '.{epoch:02d}-{val_acc:.2f}.hdf5'
model_checkpoint = ModelCheckpoint(model_names, 'val_loss',
    verbose=1,save_best_only=True)
callbacks = [model_checkpoint, csv_logger, early_stop, reduce_lr]
```

```
# loading dataset
faces, emotions = load_fer2013()
faces = preprocess_input(faces)
num_samples, num_classes = emotions.shape
xtrain, xtest,ytrain,ytest = train_test_split(faces,
    emotions,test_size=0.2,shuffle=True)
model.fit_generator(data_generator.flow(xtrain, ytrain,
                                        batch_size),
                        steps_per_epoch=len(xtrain) / batch_size,
                        epochs=num_epochs, verbose=1,
                            callbacks=callbacks,
                        validation_data=(xtest,ytest))
```

# RESULTS

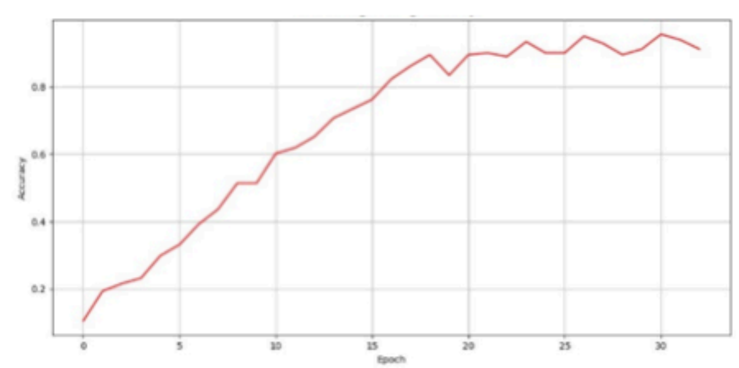
The experiment is conducted and the number of trials is converted into ‘epochs’ where at certain number of trails epochs are taken.

| Epoch | CNN Method Accuracy | 2Gabor + CNN Method |
|---|---|---|
| 1 | 0.1050 | 0.1326 |
| 10 | 0.5138 | 0.8619 |
| 15 | 0.7348 | 0.9227 |
| 20 | 0.8343 | 0.9558 |
| 25 | 0.9006 | 0.9779 |
| 30 | 0.9116 | 0.9716 |

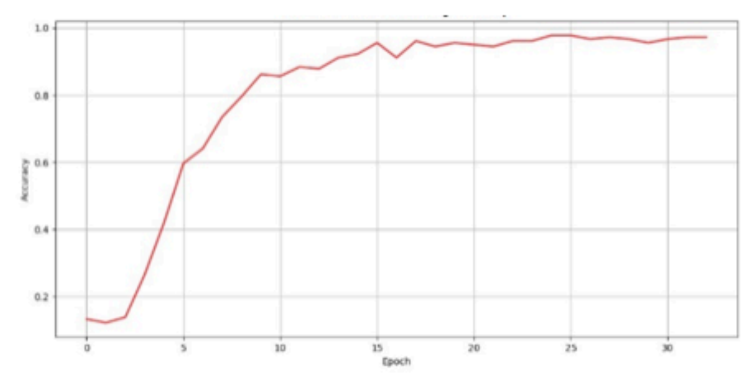
Comparison accuracy

| Accuracy | CNN Method | | 2Gabor + CNN | |
|---|---|---|---|---|
| | Epoch | Time(s) | Epoch | Time(s) |
| 10-15% | 1 | 42 | 1 | 41 |
| 50-55% | 9 | 330 | 6 | 232 |
| 70-73% | 14 | 521 | 8 | 308 |
| 86-87% | 18 | 683 | 10 | 390 |
| 91-92% | 30 | 1189 | 14 | 541 |

Comparison of Speed



CNN method



2Gabor Filter+CNN method

## Confusion of emotions

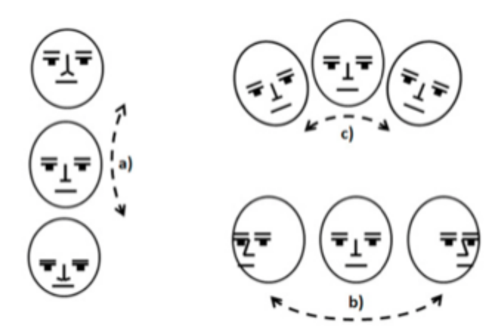
In order to determine which emotions are the easiest and which the most difficult to distinguish, it is necessary to calculate the confusion matrices.
Here, we can see that:

- Sadness was confused with neutral emotion.
- Fear was confused with surprised emotion.

| Emotions | neutral | joy | surprise | anger | sadness | fear | disgust |
|---|---|---|---|---|---|---|---|
| neutral | - | 0.86 | 0.94 | 0.85 | **0.71** | 0.93 | 0.85 |
| joy | 0.86 | - | 0.98 | 0.84 | 0.84 | 0.98 | 0.83 |
| surprise | 0.94 | 0.98 | - | 0.98 | 0.94 | **0.75** | 0.95 |
| anger | 0.85 | 0.84 | 0.98 | - | 0.87 | 0.97 | 0.84 |
| sadness | **0.71** | 0.84 | 0.94 | 0.87 | - | 0,89 | 0.86 |
| fear | 0.93 | 0.98 | **0.75** | 0.97 | 0.89 | - | 0.94 |
| disgust | 0.85 | 0.83 | 0.95 | 0.84 | 0.86 | 0.94 | - |

The classification accuracy of emotions in pairs

## Effect of head rotation, lighting conditions and object on the face




Rotation of head in a)Y- axis b)X-axis and c)Z-axis

# CONCLUSION

After the Gabor filter applied, the system learning became faster and the accuracy has improved.  The learning speed of the convolutional neural network has increased immensely. This is because the Gabor filter actually extracts the image sub feature and gives the neural network. By doing this, the convolutional neural network receives a number of sub feature and takes one step further in extracting the emotions from the faces.

Accuracy was influenced by the way users play facial expressions.

Certainly, the classification accuracy was influenced by the way users play specific facial expressions. In real conditions the classification accuracy can be affected by many additional factors. When you feel real emotions, facial expressions can vary greatly - may be exposed to a greater or lesser extent.

# FUTURE WORK

The proposed system of emotion classification has already founds its niche in the market in several industry. But there is still more innovative and motivating ways to incorporate these into growing future technologically advanced areas

Humans show plenty of cognitive changes in their abilities to recognize emotion over the generations. A pivotal point to keep in mind when studying automatic emotion recognition is that there are many sources of truth about real emotion. For example, Kyle may feel sad, but he has a big, happy face that most people say he looks happy. If an automated method results in the same conclusion as a group of observers, it may be concluded as accurate, even if it does not actually show Alex 'truly' feels. In general, getting to the truth of which emotion is actually expressed can take some work. It can vary depending on the criteria that are selected and will involve sustaining some level of uncertainty.

# REFERENCES


[1] S. L. Happy and A. Routray, "Automatic facial expression recognition using features of salient facial patches," in IEEE Transactions on Affective Computing, vol. 6, no. 1, pp. 1-12, 1 Jan.-March 2015, doi: 10.1109/TAFFC.2014.2386334.

[2] A. Nicolai and A. Choi, "Facial Emotion Recognition Using Fuzzy Systems," 2015 IEEE In- ternational Conference on Systems, Man, and Cybernetics, Kowloon, 2015, pp. 2216-2221, doi: 10.1109/SMC.2015.387.

[3] M. Imani and G. A. Montazer, "GLCM features and fuzzy nearest neighbor classifier for emo- tion recognition from face," 2017 7th International Conference on Computer and Knowledge Engi- neering (ICCKE), Mashhad, 2017, pp. 8-13, doi: 10.1109/ICCKE.2017.8167879.

[4] Kwolek B. (2005) Face Detection Using Convolutional Neural Networks and Gabor Filters. In: Duch W., Kacprzyk J., Oja E., Zadrożny S. (eds) Artificial Neural Networks: Biological Inspirations – ICANN 2005. ICANN 2005. Lecture Notes in Computer Science, vol 3696. Springer, Berlin, Heidelberg.

[5] H. Li, J. Sun, Z. Xu and L. Chen, "Multimodal 2D+3D Facial Expression Recognition With Deep Fusion Convolutional Neural Network," in IEEE Transactions on Multimedia, vol. 19, no. 12, pp. 2816-2831, Dec. 2017, doi: 10.1109/TMM.2017.2713408.

[6] Tzirakis, Panagiotis & Trigeorgis, George & Nicolaou, Mihalis & Schuller, Björn & Zafeiriou, Stefanos. (2017). End-to-End Multimodal Emotion Recognition Using Deep Neural Networks. IEEE Journal of Selected Topics in Signal Processing. PP. 10.1109/JSTSP.2017.2764438.

[7] G. Pons and D. Masip, "Supervised Committee of Convolutional Neural Networks in Automated Facial Expression Analysis," in IEEE Transactions on Affective Computing, vol. 9, no. 3, pp. 343-350, 1 July-Sept. 2018, doi: 10.1109/TAFFC.2017.2753235.

[8] A. Verma, P. Singh and J. S. Rani Alex, "Modified Convolutional Neural Network Architecture Analysis for Facial Emotion Recognition," 2019 International Conference on Systems, Signals and Image Processing (IWSSIP), Osijek, Croatia, 2019, pp. 169-173, doi: 10.1109/ IWSSIP.2019.8787215.

[9] Tarnowski, Paweł & Kołodziej, Marcin & Majkowski, Andrzej & Rak, Remigiusz. (2017). Emotion recognition using facial expressions. Procedia Computer Science. 108. 1175-1184. 10.1016/j.procs.2017.05.025.

[10] M. M. Taghi Zadeh, M. Imani and B. Majidi, "Fast Facial emotion recognition Using Convolu- tional Neural Networks and Gabor Filters," 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Tehran, Iran, 2019, pp. 577-581, doi: 10.1109/KBEI.2019.8734943.

END