Islamic University of Technology ( IUT **)**

# Face Emotion Recognition Using Deep Learning

Authors :

Tareq Khaled – **170041067**<br>
Ahmed Camara – **170041070**<br>
Mahamat Djibrine – **170041072**<br>
Malick Sow – **170041073**

Teacher:

**Sabbir Ahmed**<br>
Lecturer, Dept. of CSE, IUT.

Course Name:

**Pattern Recognition**<br>
**CSE (4835/4836)**

Department of Computer Science and Engineering (CSE)<br>
Islamic University of Technology (IUT)

# Abstract:

Facial emotion recognition has gained significant attention in recent years, as it is an important and a core element of human communication. Understanding the cause and consequences of facial emotions can be useful in many applications, requiring accurate and effective recognition algorithms. State-of-the-art methods for recognizing basic emotions like happiness or anger have been developed using deep learning techniques, but these systems are limited to only a few emotions.

In this work, by using **FER2013 dataset** that contains seven emotions (Angry, Surprise, Disgust, Happy, Sad, Fear,  or Neutral) we proposed a method based on **transfer learning** to train a model that would detect the emotion on the face of a human being. The architecture we used was well-known architecture: **EfficientNetB1**. At the end of the training, we got an accuracy of **64%** on the test data with a validation of **0.97**.

# Introduction:

Facial emotion recognition is actually the process of detecting or analysising human emotions through facial expressions, which is a topic that has been gaining more attention over the past few years.
The process of facial emotion recognition has been done in two ways: one is by capturing a video and then using computer vision to detect the person's facial expression. The other way is by detecting the person's expressions from a still image.

When it comes to expressing emotions, people often describe what they are feeling with facial expressions. Facial emotion detection is based on the idea that the eyes and mouth are the two main areas of our face that convey subtle but important changes in mood. For example, blinking or closing one's eyes can be seen as a sign of sadness, while smiling or frowning depicts joy. This technique has been used in research to detect whether someone is experiencing pain by looking at their facial expressions when presented with a painful stimulus.[1]

One of the latest developments in deep learning is the ability to evaluate emotions, or rather detect facial expressions. , we have seen many advances in recent years that have improved our driving experience and security systems. These findings are related since emotions are an important factor that can impact a person’s decision-making process. The study published a few years ago by MIT researchers in conjunction with those at Brown University and Facebook suggests that deep learning has now been used to detect the six universal human emotions: happiness, disgust, anger, fear, sadness, and surprise.

Today's society is extremely focused not only on improved communication but also on improving our ability to read each other's emotions through facial expression reading devices. The use of this technology could be applied to many different filed and areas, such as marketing, healthcare, education, and security. Some of the use cases are:

**Marketing**: Facial emotion recognition can be used to measure the customer's emotional state during an advertisement or to see if they are enjoying it or not.

**Healthcare**: It can be used to detect different types of mental disorders such as depression or anxiety.

It can also be used for entertainment, such as **video games and movies**. It can also be used for commercial purposes, such as **marketing, advertising and security**. In these cases, the application of facial emotion recognition would allow computers to recognize key aspects of a person's face that humans would not otherwise be able to see. Therefore, they could detect someone's mood or state based on their facial expressions.

# Literature Review:

This literature review examines the research and study of how machines identify emotions in faces using different deep learning approaches. It reviews the types of experiments done to test facial recognition, as well as a summary of the findings.

Pradnya Kedari, Mihir Kapile, Divya Kadole, Sagar Jaikar have used deep learning approach to detect face emotion from multiple datasets (FER-2013 and CK+) and their experiment architecture have convolution operation , Batch normalization ,ReLU ,Max pooling and they got **60%** accuracy for FER-2013 dataset, 99.1% accuracy for CK+ dataset [2], The reason behind this low performance on FER2013 is that some classes have more images samples than others hence feature extraction more challenging.

Shruti Jaiswal and G. C. Nandi have also done paper in Neural Computing and Applications called "Robust real-time emotion detection system using CNN architecture", In this paper they actually proposed a CNN-based model and the try to compared its computation cost and efficiency with 8 different datasets to make sure its robustness, One of the dataset they used is Fer2013, and achieved around **65%** accuracy, to reduce dimensionality and showing use of 1*1 convolutions they used Inception module.[3]

Sarmela A/P Raja Sekaran , Chin Poo Lee and Kian Ming Lim, In this paper they implemented a model finetuning on the Alexnet network, which was previously trained on the Imagenet dataset, using emotion datasets. Their final model was trained and tested on two

CK+ and FER2013 datasets. The proposed model performed better existing state-of-the-art methods in facial emotion recognition by achieving the accuracy of 99.44% for the CK+ dataset and 70.52% and the FER dataset. The major contribution of this paper includes first the image Augmentation using two well-known methods namely random rotation and horizontal flipping Then , The second phase of the implementation used full model finetuning, in this method the model's lasts layers are replaced,then entire model is unfrozen and finetuned. At the end of the layer there is a fully connected layer with an output shape as 7 because they are classifying 7 emotions classes. Two optimisers, namely Adam and RAdam was used to optimize the implemented model to improve the accuracy . Their model with the trained RAdam optimiser got a higher accuracy than the model with Adam. While training the model they performed an early stopping method to avoid overfitting. While training model with lowest validation accuracy was set as best model and saved.When the accuracy starts to decreasing during the training the training will stop. [4]

Gede Putra Kusuma* , Jonathan, Andreas Pangestu Lim, In this study they actually perform using a standalone-based modified Convolutional Neural Network (CNN) based on the Visual Geometry Group – 16 (VGG-16) classification model that was pre-trained on the ImageNet dataset and fine-tuned for emotion detection. They used same dataset FER-2013 and they could achieved around **69.40%.**[5]

Adrian Vulpe-Grigoraşi and Ovidiu Grigore have implemented a method of optimizing the hyperparameters of a convolutional neural community on the way to increase accuracy inside the context of facial emotion reputation. The best hyperparameters of the network were decided by way of producing and educating fashions based on the Random Search set of rules applied on a search space described via discrete values of hyperparameters. The best model resulted become educated and evaluated using FER2013 database, acquiring an accuracy of 72.16%.[6]

Yousif Khaireddin and Zhuofa Chen have worked also in FER2013 database in their paper and achieved the highest single network classification accuracy. Their work trying to adopt the VGGNet architecture, its hyperparameters, rigorously fine-tune, and also they experiment with various optimization methods, their model could achieved state-of-theart single-network getting accuracy of 73.28 % on same dataset FER2013 without using extra training data.[7]

The study found that deep learning is more accurate than traditional machine learning when it comes to recognizing emotions on faces. This is because of the fact that deep neural networks are able to take into account all of the possible layers in a human face, whereas traditional machine-learning algorithms do not have this capability. Deep neural networks also have an advantage over traditional machine-learning algorithms because they are able to learn from examples, while traditional algorithms can only be programmed by humans.

# Methodology:

The project was done under five phases:

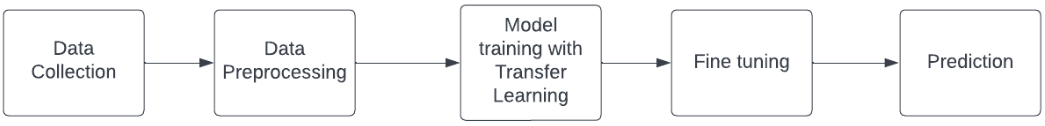


Figure 1: Five phases of our project

And we are going to see each of the section in details:

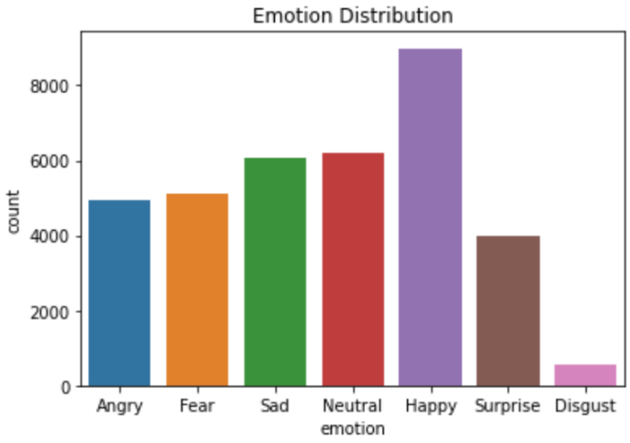
## 1) Description of Data: FER2013

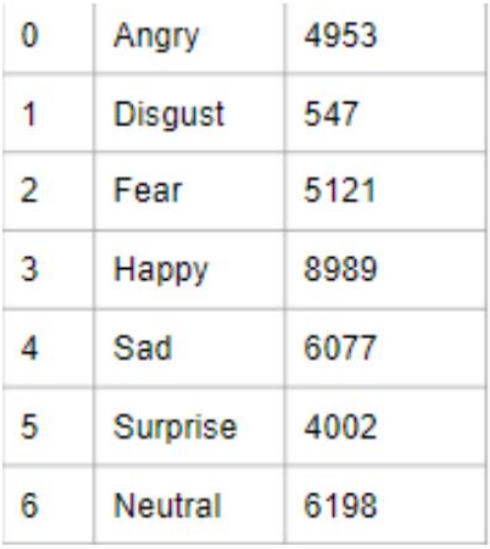
The dataset we used in this project is the famous dataset knows as **FER2013**, it was collected from **Kaggle** [8].

It has **seven** emotions expression (0=Angry, 1=Disgust, 2=Fear, 3=Sad, 4=Happy, 5=Surprise, 6=Neutral). The dataset consists of **28709 images** distributed under **7 classes**. We used **22968** images for the **training** phase, and **5741** for the **validation** phase. The **test dataset** consists of **7178** images.

## 2) Exploratory Data Analysis:

We can see from the bellow figure the distribution of the emotions classes in the dataset:





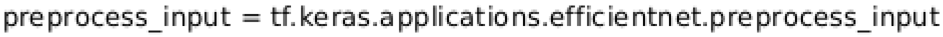
| | | |
|---|---|---|
| 0 | Angry | 4953 |
| 1 | Disgust | 547 |
| 2 | Fear | 5121 |
| 3 | Happy | 8989 |
| 4 | Sad | 6077 |
| 5 | Surprise | 4002 |
| 6 | Neutral | 6198 |

Figure 2: Distribution of dataset

We can clearly understand that the dataset that we are dealing with it is actually **imbalance dataset** mean there are **minority classes** like disgust and surprise that have very low numbers of images in the dataset and those classes will be difficult to train them in the model.

## 3) Data Pre-processing:

The next step consists of data preprocessing. At this point, the pixel values in our dataset are in [0,255], we need to match the pixel value with the model's expectation. Therefore, we need to rescale the pixel into the appropriate format.

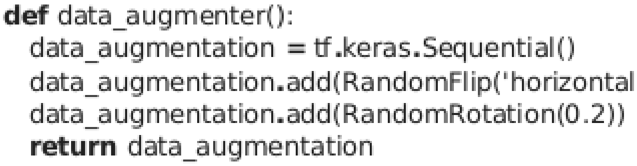


```
preprocess_input = tf.keras.applications.efficientnet.preprocess_input
```

## 4) Data Augmentation:

Image augmentation is a technique for modifying original images by applying various transformations to those images, resulting in several different versions of the same image. Each copy is unique in certain respects depending on the augmentation techniques utilized, such as shifting, rotating, flipping, and so on.

So in order to deal with this imbalance dataset then we have to perform data augmentation. We have to augment our dataset, to make the learning process more generalized. This prevents overfitting by training the model to various elements of the training data



```
def data_augmenter():
  data_augmentation = tf.keras.Sequential()
  data_augmentation.add(RandomFlip('horizontal'))
  data_augmentation.add(RandomRotation(0.2))
  return data_augmentation
```

- The **RandomFllip function** helps us to flip the images. Depending on the mode parameter, this layer will flip the pictures up or down. The outcome will then be similar to the input at inference time.

- Each image will be shuffled at random by the **RandomRotation layer**. Random rotations should only be used during learning.  By default, the layer will not do anything during the prediction phase.

Bellow the image of image augmentation:

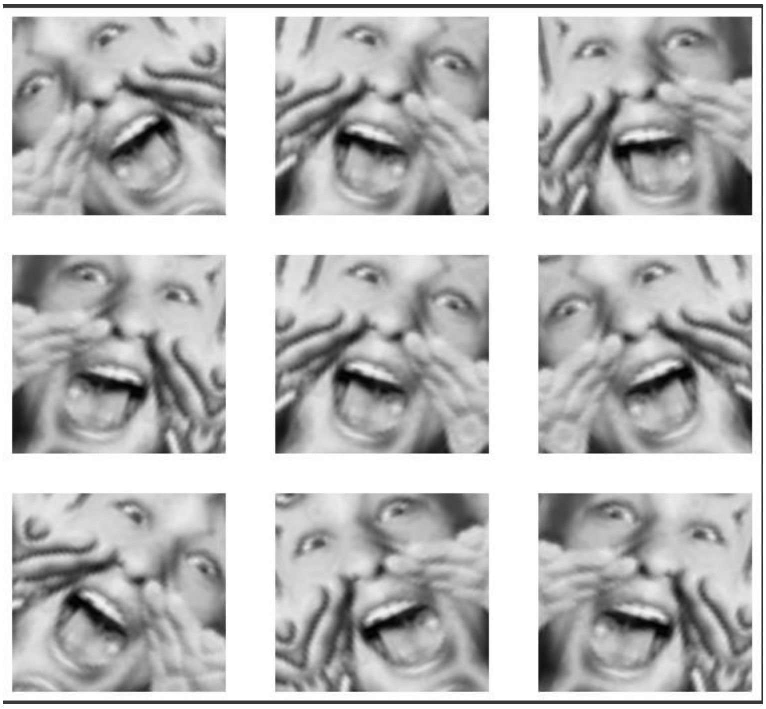
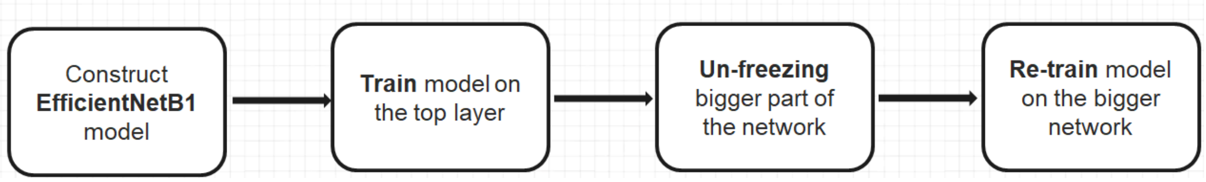


Figure 3: Data Augmentation

## 5) Model Building:



Construct **EfficientNetB1** model
**Train** model on the top layer
**Un-freezing** bigger part of the network
**Re-train** model on the bigger network

Figure 4: Methodology

We built a custom CNN architecture and trained the model, the accuracy we got was about **50%.** In order to increase the accuracy, we used transfer learning.
Before going deeper, let’s try to understand what is transfer learning and how it works.

- Transfer Learning is the technique of using an already pre-trained model on our data for a specific problem. Transfer Learning is done in two ways.

- The first approach consists of only un-freezing the top two layer, and keep the previous layers frozen, and train our model. Once the model have converged, we can re-train the model on a bigger network. This process is known as fine-tuning.

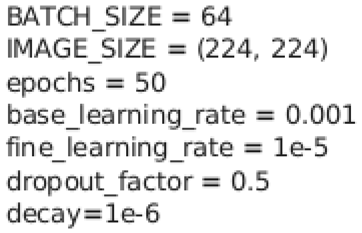
- Fine-tuning the weights of the upper layers of the base model alongside the learning of the classifier we introduced is one way to improve performance much farther. The weights will be obliged to be tuned from generalized extracted features to features specific to the dataset during the learning phase.

For our experiment, we let the first 200 layers frozen, and un-freeze the others one to train the model. We did that because, in CNN the first layers will not learn much about the data, the deeper we goes, the more specific our model become. That’s why we did not start the fine tuning from the very beginning of the architecture.

The final step consists of making the prediction. Once we have trained the model, we need to test the model on the testing data which are about 7178 images.

In this project we used only one architecture: **EfficientNetB1.**

The parameters used to train the model are given below:



```
BATCH_SIZE = 64
IMAGE_SIZE = (224, 224)
epochs = 50
base_learning_rate = 0.001
fine_learning_rate = 1e-5
dropout_factor = 0.5
decay=1e-6
```

- In Deep Learning, the **batch size** usually used by engineers is in [32, 64,128]. It does not mean these are the only values to be taken. In our project, we took different values for the batch size, and the value of **64** gave us the optimal result.

- The **image size given is 224** because the efficient architecture accept data in the shape of **(224,224,3)**

- As we trained the model on top of the softmax layer for the first time, we used a small epochs value. We did this because, as we have already explained, the goal here is only to make the model converge. After this phase, in the fine-tuning phase we trained the model for a longer time with a high epoch value of 100.

- We took **1e-3 as the base learning rate** and after for the fine tuning phase, we took a very small learning rate 1e-5. The reason behind this idea is that, as the architecture was already trained on a bigger different dataset, we want the model to generalize extracted features to features specific to the dataset during the learning phase. The main challenge here is that we need to be very careful because, the model quickly starts to overfit at some point.

- The **dropout layer** is a technique for regularization when our model is overfitting. It is a technique that will randomly delete some neurons in the neural network during the training phase to give us a simple network as shown in the figure below.

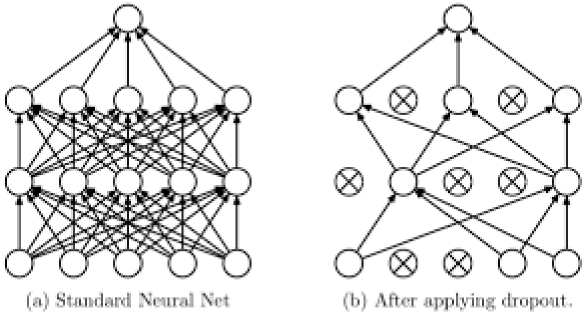


Figure 4: Dropout layer

- The **dropout factor** gives the probability for each neuron to be randomly kept or removed in the training phase.

All these parameters mentioned have to tune in order to get better results.

# Results Analysis:

As stated above, the model training was done in two phases. The first phase which consisted of training the model on only the top layers gave us an **accuracy of 49%.** Then when doing the fine tuning we were able to reach an **accuracy of 64%** for the validation data.

From the picture below, the graph represents the training with only the top layers. We can clearly see that the model converges.

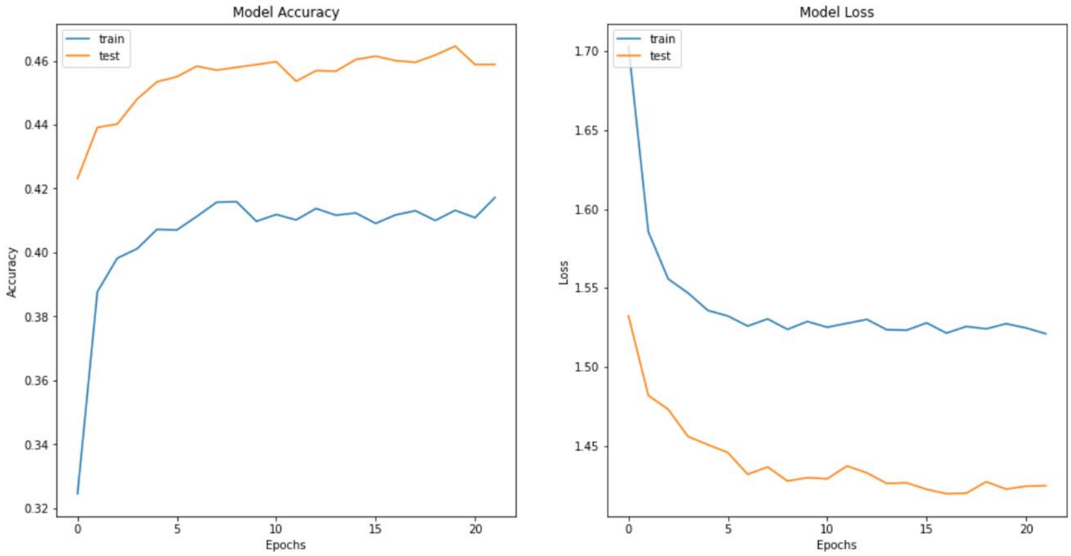


Figure 5: Result Analysis of transfer learning

Once this is done, we trained the model on a bigger network, and the **accuracy is 46%,** and the **validation is 0.97** even though the model overfitted at some point.

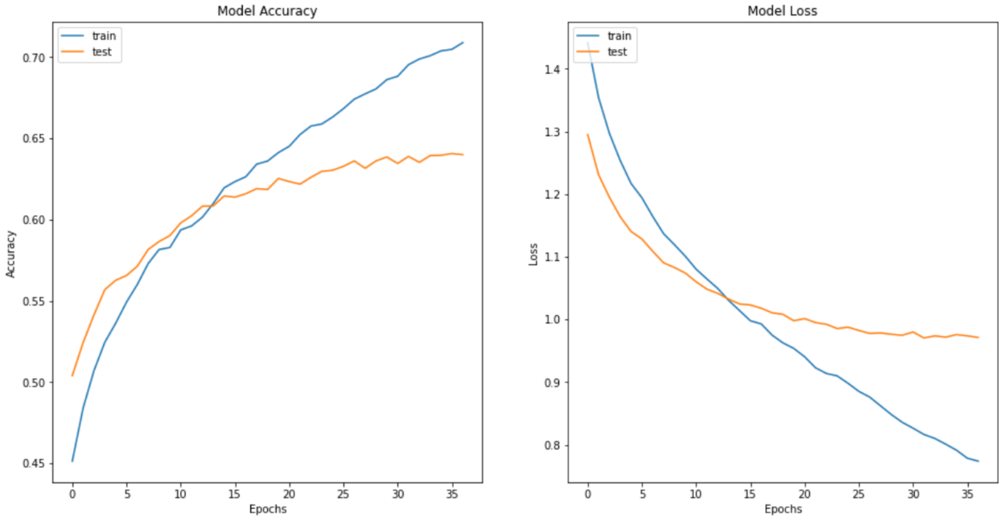


Figure 6: Result Analysis after using fine-tuning

We can see that the results for the model to predict on the testing data are **0.64**. With this result, we were able to outperform the result found [name of paper and title]

### Real Time Emotion Detection:

Real time emotion detection is actually an application of facial emotion recognition in which the prediction by the model happens on the live webcam feed.
By using open source python libraries such as OpenCV-python the detector was created and here are few results from locally testing the application.

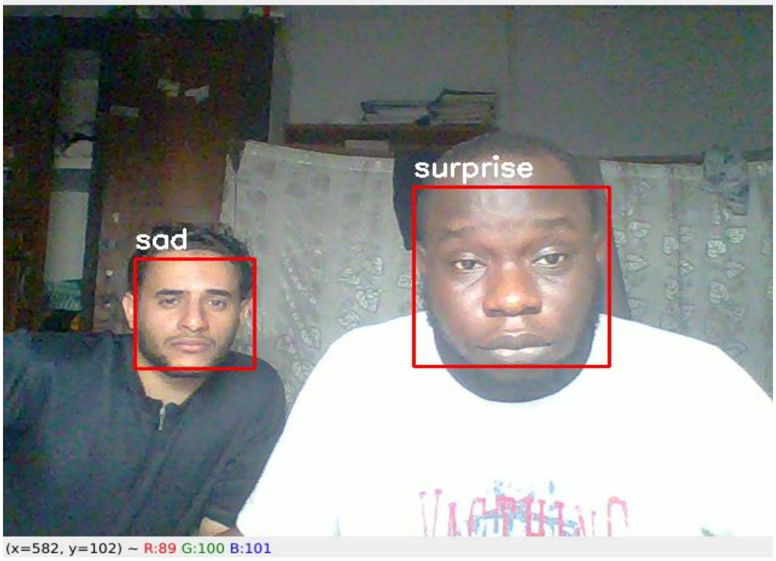


Figure 7: Real Time Emotion Detection

## Conclusion:

- Facial Emotion Recognition (FER) is the technology that analyses facial expressions from both static images and videos in order to reveal information on one's emotional state.
- Using Data preprocessing and data augmentation for handling imbalance dataset
- We train Transfer Learning model to get 45% accuracy after we done Fine-tuning in model we could achieved 64% accuracy.
- We implement Real Time Emotion Detection by using OpenCV-python.

# Future Work:

- Using different optimization techniques.
- Try out more different hyper parameters tuning.
- Explore other techniques to deal with imbalance dataset.
- Increase model architecture to extract complex features..
- More data preprocessing.

## References:

[1] Illiana Azizan and Fatimah Khalid, “Facial Emotion Recognition: A Brief Review”, *ICSETM* -2018.

[2] Pradnya Kedari , Mihir Kapile, Divya Kadole , and Sagar Jaikar “Face Emotion Detection Using Deep Learning”, 2021 (ACCESS).

[3] Shruti Jaiswal , G. C. Nandi “Neural Computing and Applications” : https://doi.org/10.1007/s00521-019-04564-4

[4] Sarmela A/P Raja Sekaran , Chin Poo Lee and Kian Ming Lim “Facial Emotion Recognition Using Transfer Learning of AlexNet” 2021 (ICoICT).

[5] Gede Putra Kusuma* , Jonathan, Andreas Pangestu Lim, “Emotion Recognition on FER-2013 Face Images Using Fine-Tuned VGG-16” 2020 ASTESJ https://www.astesj.com/publications/ASTESJ_050638.pdf

[6] Adrian Vulpe-Grigoraşi and Ovidiu Grigore, “Convolutional Neural Network Hyperparameters optimization for Facial Emotion Recognition” in International Symposium on Advanced Topics in Electrical Engineering (ATEE)

[7] Yousif Khaireddin and Zhuofa, **“**Facial Emotion Recognition: State of the Art Performance on FER2013” arXiv:2105.03588

[8] https://www.kaggle.com/datasets/msambare/fer2013