Emotion Recognition System

1st Soniya Maharjan
Computer and Electronics Engineering Department
Kantipur Engineering College, Tribhuvan University
Lalitpur, Nepal
maharjansoniya36@gmail.com

2nd Rinky Maharjan
Computer and Electronics Engineering Department
Kantipur Engineering College, Tribhuvan University
Lalitpur, Nepal
maharjanrinky@gmail.com

3rd Sawarni Ghimire
Computer and Electronics Engineering Department
Kantipur Engineering College, Tribhuvan University
Lalitpur, Nepal
sawarnighimire@gmail.com

4th Nisha Bhattarai
Computer and Electronics Engineering Department
Kantipur Engineering College, Tribhuvan University
Lalitpur, Nepal
nishabhattarai1905@gmail.com

***Abstract*—Human facial expressions transmit a great deal of information visually. In the field of human-machine interaction, facial expression recognition is critical. Many applications exist for automatic facial expression recognition systems, including, but not limited to, human behavior comprehensions and synthetic human emotions. Facial expression recognition by computer with a high recognition rate remains a difficult challenge. Face detection, feature extraction, and expression classification are the three phases of facial expression recognition. We used deep learning approaches (convolutional neural networks) in this study to detect the key seven human emotions: angry, disgust, fear, happy, sad, surprise, and neutral. A dataset, FER2013 was utilized for pre-processing of images. Haar algorithm, Local Binary Pattern, and Convolution Neural Network have been implemented in various stages of our project. User satisfaction is a significant advantage of implementing emotion detection. The goal of this system is to classify the captured image and determine the emotion.**

***Keywords*—Emotion detection, Feature Extraction, Local Binary Pattern (LBP), Convolution Neural Network (CNN), FER2013**

I. Introduction

Emotion recognition is a critical area of research in the field of computer vision and artificial intelligence, as it has various applications in human-computer interaction, medical diagnosis, marketing, data-driven animations, human-robot communication, and many more. Emotion recognition is the process of identifying human emotions from facial expressions, vocal cues, physiological signals, and other modalities. In this project, facial expression is used for classification of different emotions.

This project is focused on extracting the distinctive and precise facial features which later helps in classifying the image of the individual to identify them. It is a feature learning technique in which model learns from the features extracted from the images. Also it uses supervised learning method in which the model is trained on a labeled dataset where the emotion label is provided for each image in the dataset. FER2013 dataset is utilized for this project. New, unseen images' emotion can be classified accurately. To achieve this aim, the fusion of Local Binary Patterns (LBP) and Convolution Neural Networks (CNN) for emotion detection is proposed. LBP feature map is used as the input of CNN to improve the understanding and learning of CNN which will provide guidance for the selection of CNN learning data [1].

LBP is a texture descriptor that characterizes local patterns in an image. LBP algorithm is implemented for its high effectiveness and robustness in handling the illumination effect in captured images. It extracts local textures, edges, and patterns within an image. LBP capture details such as wrinkles, bumps, and contours in facial expressions that are the indicative of emotions.

CNNs, on the other hand, are a type of neural network that can learn hierarchical representations of features from input data. This approach of feeding image generated from LBP instead of normal image improve the performance of the CNN by providing it with richer, more informative input features. Convolutional Neural Network is focused on extracting high-level distinctive features from the local 1features obtained by LBP. The high-level features from CNN are then passed through the fully connected layer with Softmax classifier to identify the individuals' emotion. Backward propagation is used to update the weights of the network based on the error between the identified output and the true output during the training phase. After the model is trained, it is evaluated on a test dataset to measure its performance. Common metrics for evaluation: accuracy, precision, recall, and f1-score is calculated which is then deployed for real-world applications for detection of emotion of images.

II. Methodology

The dataset, used for training the model is Facial Expression Recognition Challenge(FER2013) [8]. The dataset consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The training set consists of 28,709 images. The test set consists of 7,178 images. Emotion labels in the dataset:

0: 4593 images- Angry
1: 547 images- Disgust
2: 5121 images- Fear
3: 8989 images- Happy
4: 6077 images- Sad
5: 4002 images- Surprise
6: 6198 images- Neutral

Oversampling is the process of increasing the number of instances or samples of a minority class by creating new instances or repeating some instances. We have an imbalanced classification problem when there is a significant difference in the class distribution of our training data. It is regarded as a source of concern because it may have an impact on the performance of our Machine Learning algorithms. FER2013 has an imbalanced classification problem. It's majority class, happy contains 8989 images and the minority class, disgust contains only 547 images. Hence to solve this problem, oversampling was done. The oversampler used for oversampling was random oversampler.

In general, classifiers can differentiate between normal image data and LBP image data because they are trained on different feature representations of the images. Normal image data represents an image as a matrix of pixel values that correspond to the intensity or color of each pixel. LBP, on the other hand, encodes information about the local texture and patterns within the image into a binary pattern that is calculated based on the relationships between the intensity values of neighboring pixels.

Local Binary Pattern (LBP) is a method to describe texture characteristics of a surface. The LBP divides the examined window into cells of pixel and compare the pixel to each of its neighbour's and follows the pixel along a circle, clockwise direction.The LBP operator is defined as in equation

$$LBP_{n,c} = \sum_{n=0}^{7} s(i_n - i_c)2^n \tag{1}$$

When n is the neighbors, i_n, i_c are gray intensity values of neighboring pixel and sampling center pixel and s(x) is defined as

$$s(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{2}$$

The CNN architecture is inspired by the visual cortex in the brain. CNN designs vary largely, but they always have convolutional and pooling layers that are organized into modules. These modules are followed by one or more fully linked layers, similar to a normal feed-forward neural network. To create a deep model, modules are frequently stacked on top of one another. The network receives an image directly, which is then processed through various rounds of convolution and pooling. The results of these processes are then sent into one or more fully linked layers. Backpropagation occurs which alters the parameters of a network to reduce errors that impact performance by computing the gradient of an objective function.

Some of the hyper-parameter of CNN are:
Epochs: If all images in the dataset are processed one time individually, forward and backward to the network, then that is one epoch. The number of epochs used in this project was 60, this is because epochs greater that 60 caused early stopping to prevent overfitting.
Batch size: It refers to the number of training examples used in one forward/backward pass. 64 batches of images were used. Larger batch size requires more memory since the dataset is large and smaller batch size are slower.
Learning rate: It refers to the step size at which the model is updated during training. A larger learning rate are faster while smaller learning rate improves model's stability. Here an optimizer (function or algorithm that modifies the neural network's attributes such as weights and learning rates) namely Adam optimizer was used which is an adaptive learning rate method that adjusts the learning rate dynamically during training.
The layers of CNN architecture used are:
1. Input layer
2. Convolution layer
(32,64,128,256 size filters 4 layers with zero padding and ReLU activation)
3. Normalization Layer (Batch Normalization)
4. Pooling layer(Max Pooling)
5. Dropout layer
6. Flatten layer
7. Fully connected layers (softmax classifier)
Face detection is the first and most important step in emotion recognition; it detects faces in photos. It is a type of object detection that can be used in a variety of applications, including security, biometrics, law enforcement, entertainment, and personal safety. Haar cascade was the face detection algorithm used

in this project.The Haar cascade algorithm can detect objects in images regardless of their scale or location. This algorithm is not overly complex and can be executed in real time.

The detected face from Haar Cascade was converted into a grayscale image. This was done because grayscale images require less computation time than colored images, as it has only one channel. Then, the face of the image was cropped since only the features of the face is required to classify emotion of the image. The trained model used 48*48 pixels grayscale images. So, as to maintain compatibility with the input layer of the model used, the cropped image was resized to 48*48 pixels. The obtained image was then fed to LBP image generator.

III. RESULT & DISCUSSION

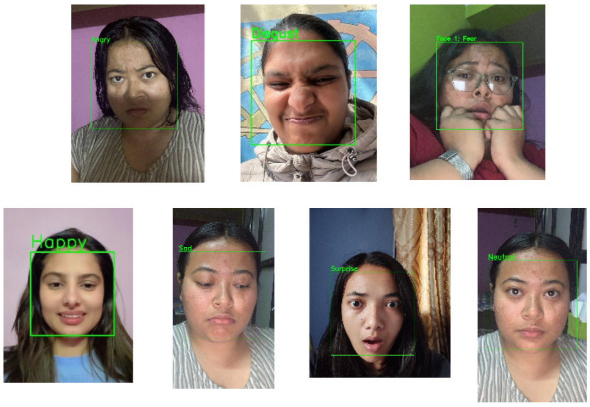

Fig. 1. Predicted emotions

Seven classes of emotion were detected from the trained model. For the detection of emotion, firstly the training(28709) and testing(7178) data of FER2013 was separated and the training dataset was oversampled (50505) after which the LBP of the dataset was generated using LBP generator. The data separated for training purpose was split into train data set(40404) and validation data(10101) in 8:2 ratio, then fed into CNN for building a model. The training accuracy obtained from the model was 77.23% and the validation accuracy was 76.03 %. After that the test data was fed to the model which resulted in 65.617% of testing accuracy. The obtained model was then used to predict the emotion in real time by using image files. The images were first fed into haar cascade model, then were sent to LBP generator if faces were detected by it. The LBP image obtained was fed to the trained model.

The result obtained are shown in figure 1 . The model was able to detect most of the emotions correctly. Thus the results obtained were satisfactory.

Confusion matrix

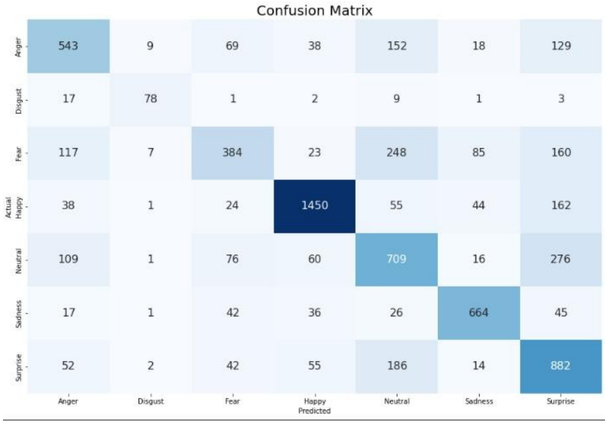

Fig. 2. Confusion matrix

The confusion Matrix gives a comparison between actual and predicted values. It is used for the optimization of machine learning models.The confusion matrix is a N x N matrix, where N is the number of classes or outputs. For 7 classes, we get a 7 x 7 confusion matrix. There are 4 terms needed to understand in order to correctly interpret or read a Confusion Matrix: True Positive(TP), False Positive(FP), True Negative(TN), and False Negative(FN). These terms are explained below:

True Positive: It means the actual value and also the predicted values are the same.

False Negative: This means the actual value is positive but the model gives the wrong prediction. Whatever the negative output we get is false; hence the name False Negative.

False Positive: This means the actual value is negative but the model predicts it as positive. So the model has given the wrong prediction. Whatever the positive output we get is false; hence the name False Positive.

True Negative: It is an outcome where the model correctly predicts the negative class.

The confusion Matrix allows us to measure Recall, Precision and F1 Score, which are the metrics used to measure the performance of the model. In figure 2, row shows the actual classes and column shows the predicted classes. The classifier made a total of 7178 predictions where the classifier predicted anger for 893 times , disgust for 99 times, fear for 638 times, happy for 1664 times, neutral for 1385 times, sadness for 842 times and surprise for 1657 times. Whereas in reality 543 cases was anger, 78 was disgust, 384 was fear, 1450 was happy, 709 was neutral, 664 was sadness and 882 was surprise.

A precision score is used to assess the model's performance in counting true positives correctly out

TABLE I
EVALUATION METRICS TABLE

	precision	recall	f1-score	support
Angry	0.608	0.567	0.587	958
Disgust	0.788	0.703	0.743	111
Fear	0.602	0.375	0.462	1024
Happy	0.871	0.817	0.844	1774
Sad	0.512	0.569	0.539	1247
Surprise	0.789	0.799	0.794	831
Neutral	0.532	0.715	0.610	1233
accuracy			0.656	7178
macro avg.	0.672	0.649	0.654	7178
macro avg.	0.666	0.656	0.655	7178

of all positive predictions made. The recall score is used to assess model performance by counting the number of true positives that are correctly identified among all positive values. When the classes are very imbalanced, the Precision-Recall score is a useful measure of prediction success.

Accuracy score is used to assess model performance by calculating the ratio of the sum of true positive and true negative predictions out of all predictions made. F1-score is the harmonic mean of precision and recall score. Their values have been presented in table I. Satisfactory f1-scores for emotions disgust, happy, surprise and neutral have been achieved.

$$precision = \frac{TP}{TP + FP} \tag{3}$$

$$recall = \frac{TP}{TP + FN} \tag{4}$$

$$f1score = \frac{2 * precision * recall}{recall + precision} \tag{5}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{6}$$

During the training process, the model was presented with a set of inputs (i.e. pixels) and their corresponding targets (i.e. emotion). The model then used an optimization algorithm (i.e. Adam optimizer) to adjust its parameters (i.e. weights and biases) to minimize the difference between predicted and the actual output. This difference was measured using a loss function (categorical cross-entropy), which quantifies how far off the model's predictions are from the targets. As the model iteratively updates its parameters i.e. after each batch is fed forward and backward to the model during backpropagation which minimize the loss on the training set and it gradually becomes better at making predictions on the training set. This leads to a decrease in the training loss. However, the model's goal is not just to make accurate predictions on the training set, but also on unseen data. Therefore, during the training process, a portion of the data (the validation set) was set aside to evaluate the model's performance. This allowed to monitor the model's generalization performance, and detect overfitting. If the model starts to overfit the training data (i.e. it starts to fit noise in the data), it will perform poorly on the validation set, and the validation loss will increase. However, if the model is able to generalize well, it will perform well on both the training and validation sets, leading to a decrease in both the training and validation loss as shown in figure 3.

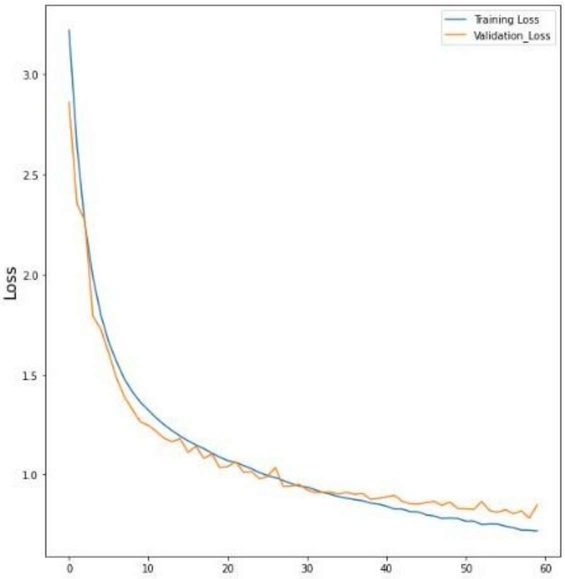

Fig. 3. Training and validation loss

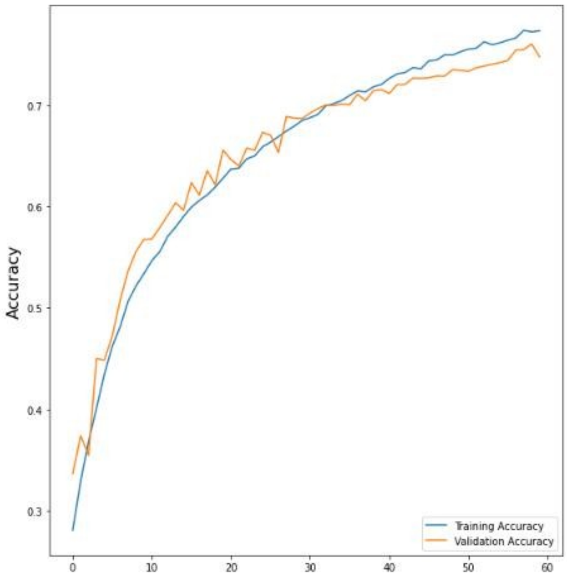

Fig. 4. Training and validation accuracy

Accuracy is a metric that measures the percentage of correctly classified instances (e.g. images, texts, etc.) by the model. It is used to evaluate the performance of classification models. During the training process, the accuracy of the model on the training set increase as the model's parameters are updated iteratively to better fit the training data during backward propagation. This occurs when the model learns to correctly classify more instances in the training set. However, it is important to note that a high accuracy on the training set does not necessarily mean that the model will perform well on new, unseen data. To evaluate the model's performance on new data, model's accuracy on the validation set was calculated by feeding the validation data through the trained model and comparing the model's predictions to the true labels. As the model trains, the accuracy on the validation set can increase or decrease. Here the model is performing well on both training and testing data so the accuracy is increasing as shown in figure 4.

IV. FUTURE ENHANCEMENT

The system that we have developed is able to correctly predict seven of the major human emotions. But since, human emotions can be ambiguous, emotions like disgust and fear, sad and angry can sometimes look same. In order for the system to provide results with maximum accuracy, a more refined dataset can be used. The dataset used in our project FER2013 has class imbalance. So using dataset having enough distinct datas for all the classes result in the model learning more distinct features such that it can classify subtle expressions and increase the model accuracy.

V. CONCLUSION

The emotion recognition system has been built which yields satisfactory result. We were able to build a system that classifies seven of the major human emotions. The project has been extremely helpful in giving us insights about machine learning and to a large extent neural network. The successful implementation of the system and meeting the project objective has provided us extra motivation for further undertaking in the field of machine learning. Considering all the features and performances of our system, we can conclude that our system can be implemented easily in the application field and will prove to be effective in classifying emotion in order to predict ones' emotional state.

ACKNOWLEDGMENT

We would like to express deep gratitude to our project coordinator Er. Bishal Thapa for his helpful guidance and encouragement which has been the key for bringing the project in its present form. We are thankful to the Department of Computer and Electronics Engineering, Kantipur Engineering College, for providing us support throughout the project period. We would like to express our sincere thanks to Er. Rabindra Khati, Er. Pralhad Chapagain and Er. Dipesh Shrestha for providing us their valuable suggestions and support.

Finally, we would like to thank all of our friends and teachers for their direct and indirect help.

REFERENCES

[1] S. Sawardekar and S. R. Naik, "Facial expression recognition using efficient lbp and cnn," Int Res J Eng Technol (IRJET), vol. 5, no. 6, pp. 2273–2277, 2018.

[2] N. Matang, S. Sunuwar, S. Shrestha, and S. Parajuli, "A facial expression recogni- tion system," Ph.D. dissertation, Tribhuvan University, 2016.

[3] S. Gaur, M. Dixit, S. N. Hasan, A. Wani, T. Kazi, and A. Z. Rizvi, "Comparative studies for the human facial expressions recognition techniques," Int J Trend Sci Res Dev Int J Trend Sci Res Dev, vol. 3, pp. 2421–2442, 2019.

[4] S. S. Kulkarni, N. P. Reddy, and S. Hariharan, "Facial expression (mood) recogni- tion from facial images using committee neural networks," Biomedical engineering online, vol. 8, no. 1, pp. 1–12, 2009.

[5] M. Bhatt, H. Drashti, M. Rathod, R. Kirit, M. Agravat, and J. Shardul, "A study of local binary pattern method for facial expression detection," arXiv preprint arXiv:1405.6130, 2014.

[6] A. Sarirete, "Sentiment analysis tracking of covid-19 vaccine through tweets," Jour- nal of Ambient Intelligence and Humanized Computing, pp. 1–9, 2022.

[7] J. Shao and Y. Qian, "Three convolutional neural network models for facial expres- sion recognition in the wild," Neurocomputing, vol. 355, pp. 82–92, 2019.

[8] Kaggle, "Face expression recognition dataset," 2019, https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition- dataset.