Indian Institute of Information Technology, Allahabad

Department of Information Technology and Information Technology Business Informatics

Semester - 5th

B.Tech Mini Project

Facial Emotion Recognition

Under the supervision of

Dr. Ramesh K Bhukya

Dr. Anshu S. Anand

Group Members:

Priyamvada Priyadarshini IIB2020037

Shalini Agrawal IIT2020236

Prakhar Chitravanshi IIT2020235

Anshuman Jain IIT2020239

Siddhant Agrawal IIT2020228

Abstract

Machine Learning is now a ubiquitous feature of our everyday life whether we notice it or not. It works in the background making our lives a tad bit easier to navigate. It is used in our mobile phones, laptops, and even in everyday objects through the IoT. It gathers data from our actions and learns specific patterns in our behavior to better predict what we want at a certain point in time. One of its many uses is in Facial Recognition. Detecting faces automatically is an incredibly important task that finds its usefulness in a plethora of ways. It may be Crime prevention, attendance systems, or biometrics among others. Our report is on one of those very important features of Facial recognition which is Facial emotion recognition. In this project, we try to detect various types of emotion as depicted through their faces. We use various types of Neural Network models like CNN, Vgg, ResNet, LSTM, etc. to detect emotion and compare their results to conclude which model performs the best under a given condition.

Introduction:

Facial Expression Recognition:

In recent years, the research on facial emotion recognition has become extensive. Facial emotion recognition aims to help identify the state of human emotion (neutral, happy, sad, surprise, fear, anger, disgust, contempt) based on particular facial images.

There are mainly 6-class and 7-class FER. 6-class include Anger, Disgust, Fear, Joy, Sadness, and Surprise. 7-class includes Neutral as well.

Challenges of FER so far:

The challenge lies in recognizing the different curves and expressions, different people express the same emotion through their faces with high accuracy.

People across the globe from different races, sex, culture, body architect, and environment express the same emotion through a range of different muscle movements.

One challenge for facial expression recognition is recognizing facial expressions at low resolutions, as only compressed low-resolution video input is available in real-world applications.

Literature Review:

Year	Paper	Method	Dataset	Accuracy
2016	Facial Expression Recognition with CNN Ensemble Kuang Liu; Mingmin Zhang; Zhigeng Pan	subnet ensemble RBM	FER2013	69.7% 65.03%
2017	A New Approach for Automatic Face Emotion Recognition and Classification Based on Deep Networks Vibha. V. Salunke; C.G. Patil	Three convolutional layers each followed by max pooling and ReLU	FER2013 trained and tested on RaFD	68%
2020	Facial Emotion Recognition System With Improved Preprocessing and Feature Extraction: Ansamma John; Abhishek MC	Pre-processing methods like facial landmark and HOG were incorporated into a convolutional neural network (CNN)	FER2013	74.4%
2020	Deep Convolution Neural Network Implementation for Emotion Recognition System	Xception" model using the Fer2013 datasets on a GPU (NVIDIA GeForce MX230) to improve our facial expression recognition system and then the improved system is evaluated using an embedded system Raspberry Pi 4.	FER2013	84%
2021	Facial Expression Recognition using Machine Learning models in FER2013; Jinyu Luo; Zhuocheng Xie; Feiyao Zhu; Xiaohu Zhu	HOG+SVM AlexNet+SVM VGG16 ResNet18	FER2013	46.0% 52.7% 67.0% 58.6%

Terminologies:

Before going further into FER approaches, we first present current related terminologies other than emotions and expressions supporting the theoretical basis of FER technology.

Facial Landmarks:
They[10, 11] are facial feature points, which are specific points on a face that are detected and characterized by a computer vision system as shown in figure 1.0. They are utilized to pinpoint and examine various features of a face, such as the location of the eyes, nose, mouth, and other facial components. Here are some examples of facial landmarks:

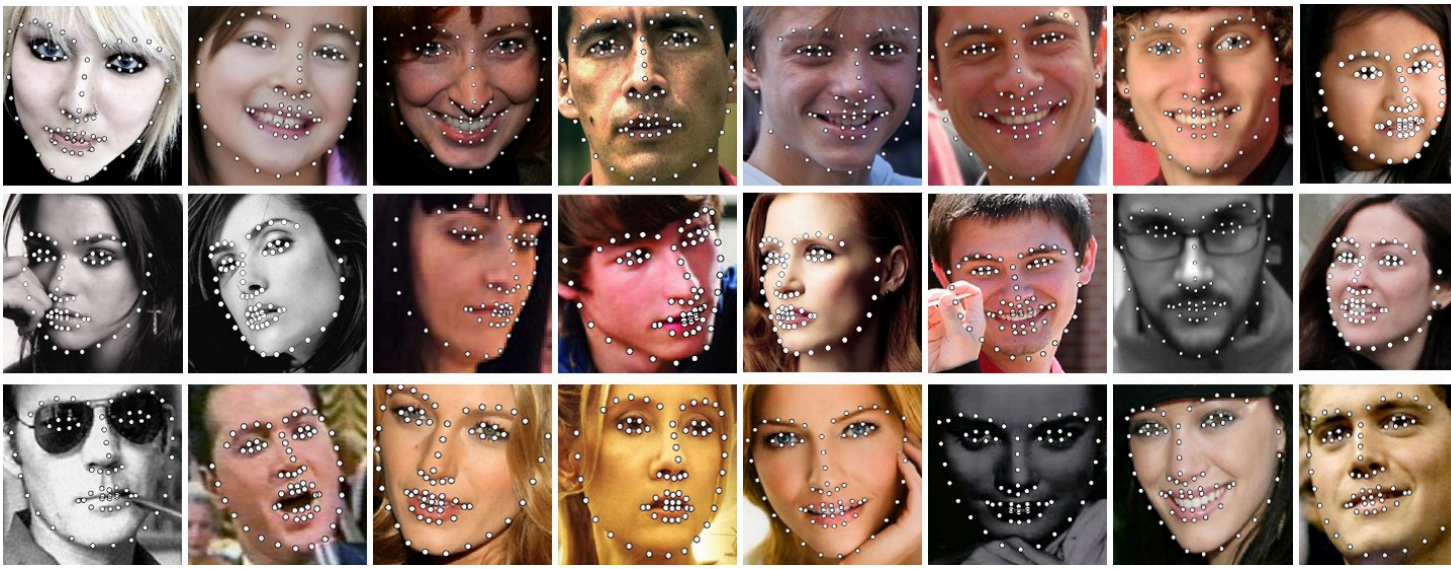


Figure 1.0 Facial Landmark detection algorithm results in a live demo[10]

- Eye landmarks: These landmarks indicate the location of the pupils, corners of the eyes, and eyelids.

- Nose landmarks: These landmarks indicate the position and shape of the nose, including the nostrils and the tip of the nose.

- Mouth landmarks: These landmarks indicate the position and shape of the lips, including the corners of the mouth, the upper and lower lips, and the philtrum (the vertical groove between the nose and the upper lip).
- Face contour landmarks: These landmarks indicate the outline of the face, including the chin, cheeks, and jawline.

Facial Action Units (AUs):

AUs (Action Units)[4] are a coding system used in facial expression analysis to describe the movements of individual muscles in the face.

Here are some examples of Action Units:

AU1 - Inner Brow Raiser: raising the inner corners of the eyebrows
AU2 - Outer Brow Raiser: raising the outer edges of the eyebrows
AU4 - Brow Lowerer: lowering the eyebrows
AU5 - Upper Lid Raiser: raising the upper eyelids
AU6 - Cheek Raiser: raising the cheeks
AU7 - Lid Tightener: tightening the eyelids
AU9 - Nose Wrinkler: wrinkling the nose
AU12 - Lip Corner Puller: pulling the corners of the mouth sideways
AU15 - Lip Corner Depressor: lowering the corners of the mouth
AU17 - Chin Raiser: raising the chin

These are just a few examples, and there are many more Action Units used to describe facial expressions.

Description of Dataset:

- **FER-2013:** FER-2013 is a dataset composed of 35,953 images in seven classes (fear, disgust, sad, happy, neutral, surprise, angry). Images are 48 × 48 pixels in size with a grey-scaled color palette. The classes' variations and feature distributions are helpful in the merging phase for other classes to obtain a good distribution and normalize the amount of data variation. According to the final classification, the contempt class was missed for this type of data. Its composition already divides its original samples into training and validation sets using different folders with the function of labeling.

- **FER-2018**: FER-2018 is a dataset composed of 34,034 images in seven classes (fear, disgust, sad, happy, neutral, surprise, angry). Images are 48 × 48 pixels in size with a grey-scaled color palette. The classes' variations and feature distributions are helpful in the merging phase for other classes to obtain a good distribution and normalize the amount of data variation. According to the final classification, the contempt class was missed for this type of data. Its composition already divides its original samples into training and validation sets using different folders with the function of labeling.

- **CK+48:** CK+48 is a small dataset composed of 981 images in seven classes (fear, disgust, sad, happy, neutral, surprise, angry). Images are 48 × 48 in size with a grey-scaled color palette. The classes' variations and feature distributions are helpful in the merging phase for other classes to obtain a good distribution and normalize the amount of data variation. Generally, images taken from video frames did not have much variation, and the total number of elements is negligible compared to other datasets.

Compared with the FER-2013, images are in frontal view with a clean pattern for facial expression. Usually cropped version of CK+48 is used with 5 emotions namely happy, surprise, fear, angry and sad.

ADAM:

An approach for the gradient descent optimization technique is called adaptive moment estimation. When dealing with complex problems involving a lot of data or factors, the strategy is incredibly effective. It is effective and uses little memory. It combines the "gradient descent with momentum" algorithm and the "RMSP" algorithm, intuitively.

Adam optimizer involves a combination of two gradient descent methodologies:

- **Momentum:** This algorithm is used to accelerate the gradient descent algorithm by taking into consideration the ‘exponentially weighted average’ of the gradients. Using averages makes the algorithm converge towards the minima in a faster pace.

- **Root Mean Square Propagation (RMSP):** Root mean square prop or RMSprop is an adaptive learning algorithm that tries to improve AdaGrad. Instead of taking the cumulative sum of squared gradients like in AdaGrad, it takes the ‘exponential moving average’.

ReLU:

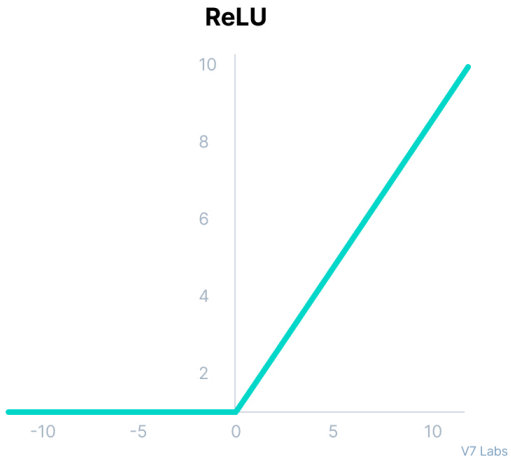
ReLU stands for Rectified Linear Unit. being computationally efficient, despite giving the impression of being a linear function. The fundamental issue here is that not all of the neurons are activated simultaneously by the ReLU function. Only if the output of the linear transformation is less than 0 will the neurons become inactive.

The following are some benefits of employing ReLU as an activation function:

- The ReLU function is far more computationally efficient than the sigmoid and tanh functions since it only activates a subset of neurons.
- Due to ReLU's linear, non-saturating nature, gradient descent's convergence towards the loss function's global minimum is sped up.

$$ReLU$$

$$f(x) = max\ (0, x)$$



ELU:

The function known as the exponential linear unit, or ELU for short, tends to converge to zero faster and yield more precise results. ELU features an additional alpha constant that must be a positive quantity, unlike other activation functions.

Except for negative inputs, ELU and RELU are quite similar. For non-negative inputs, they are both in the form of an identity function. As opposed to RELU, which smoothes sharply, ELU becomes smooth gradually until its output equals -α.

Benefits of ELU include:

- While RELU becomes significantly smoothed, ELU becomes smooth gradually until its output equals -α.
- ELU is a capable substitute for ReLU.
- ELU can result in negative outputs, unlike ReLU.

Our Implementation:

CNN: A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

CNNs have two components:

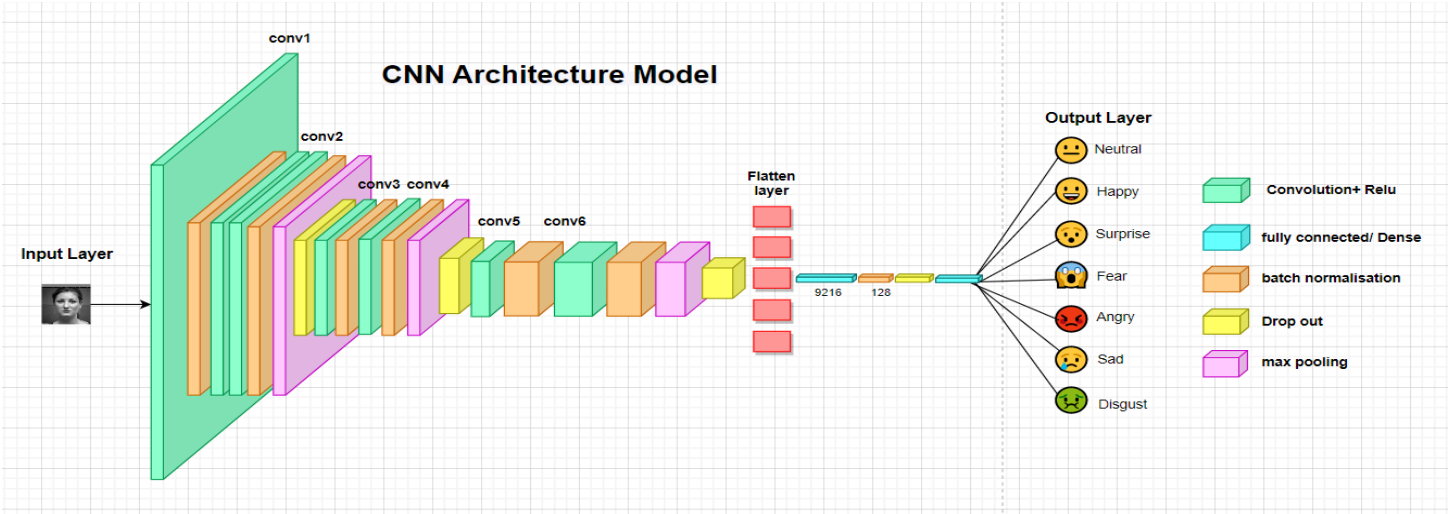
- **The Hidden layers/Feature extraction part**: The network will perform a series of **convolutions** and **pooling** operations during which the **features are detected**.

● **The Classification part**: The fully connected layers will serve as a **classifier** on top of these extracted features. They will assign a **probability** for the object on the image being what the algorithm predicts it is.

CNN Steps include:

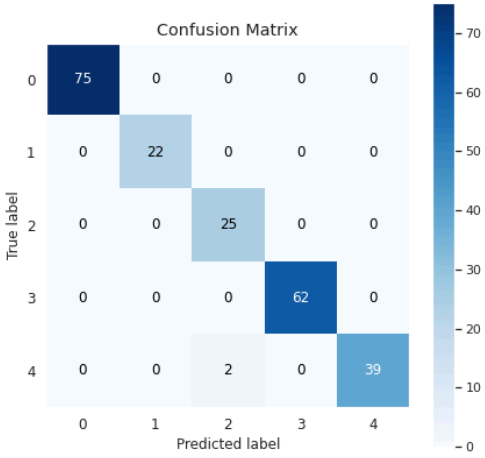
1. Convolution: Apply filters to generate feature maps (tf.keras.layers.conv2D)
2. Non Linearity: ReLU (tf.keras.activations)
3. Pooling: Down sampling operation on each feature map (tf.keras.MaxPool2D). It is used for reducing dimensionality and making spatial invariance.
4. Fully Connected Layer: Classify the image based on the features extracted.

Our Implementation:



6 Conv2D layers with ReLU activation function layer followed by Batch Normalization and dropouts which is followed by max pooling layer to reshape data and input to flatten layer. Finally a Dense layer which finds the probability of each image being happy, surprise, fear, angry and sad and gives the prediction accordingly.

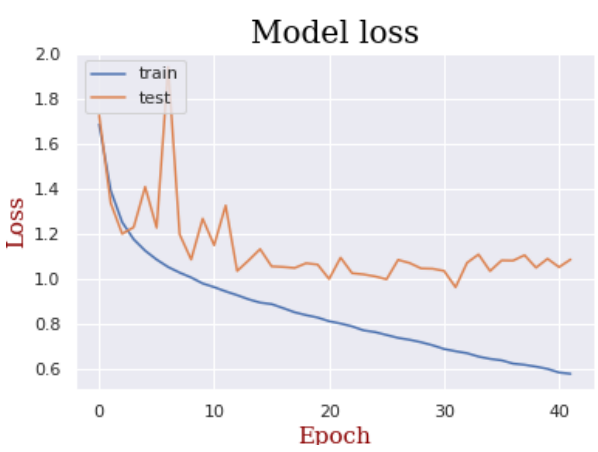
Our result:

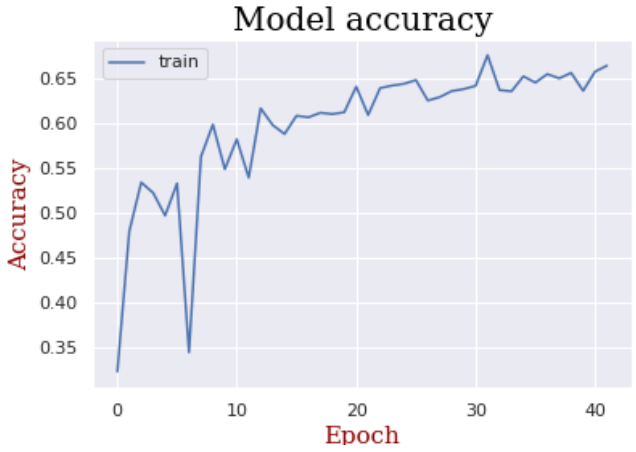


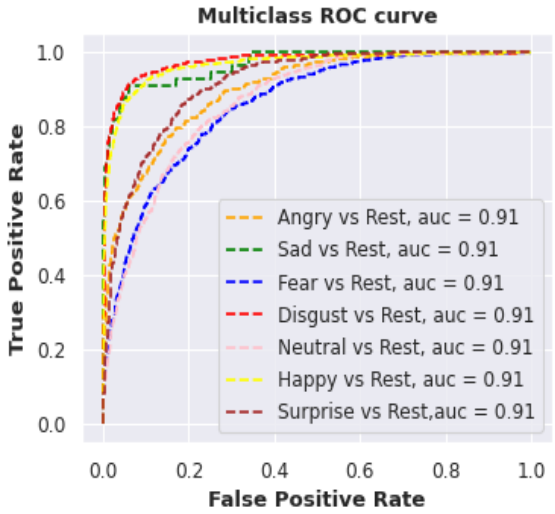
test accuracy: 99.1111 %

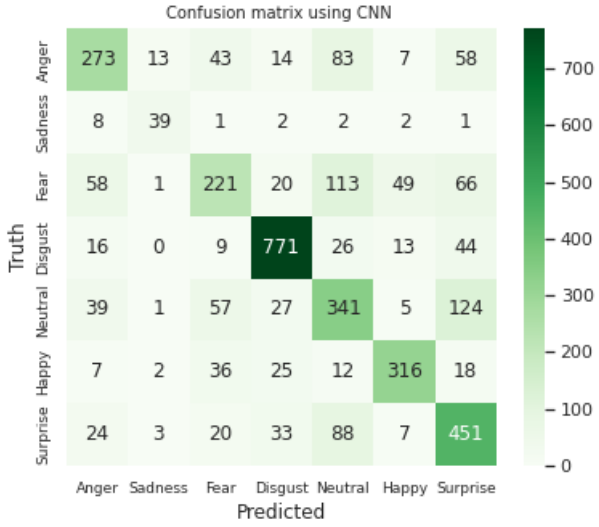
Emotions	Precision	Recall	f1-score	support
surprise	1.00	1.00	1.00	75
fear	1.00	1.00	1.00	22
sadness	0.93	1.00	0.96	25
happy	1.00	1.00	1.00	62
anger	1.00	0.95	0.97	41
accuracy			**0.99**	225
Macro avg	0.99	0.99	0.99	225
Weighted avg	0.99	0.99	0.99	225

FER2013:

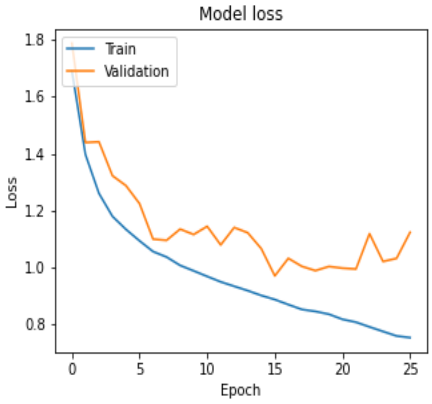


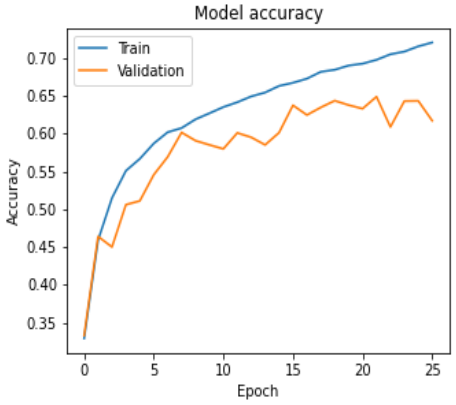


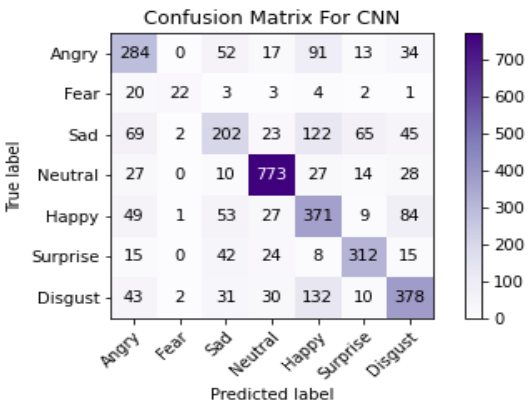




FER2018:



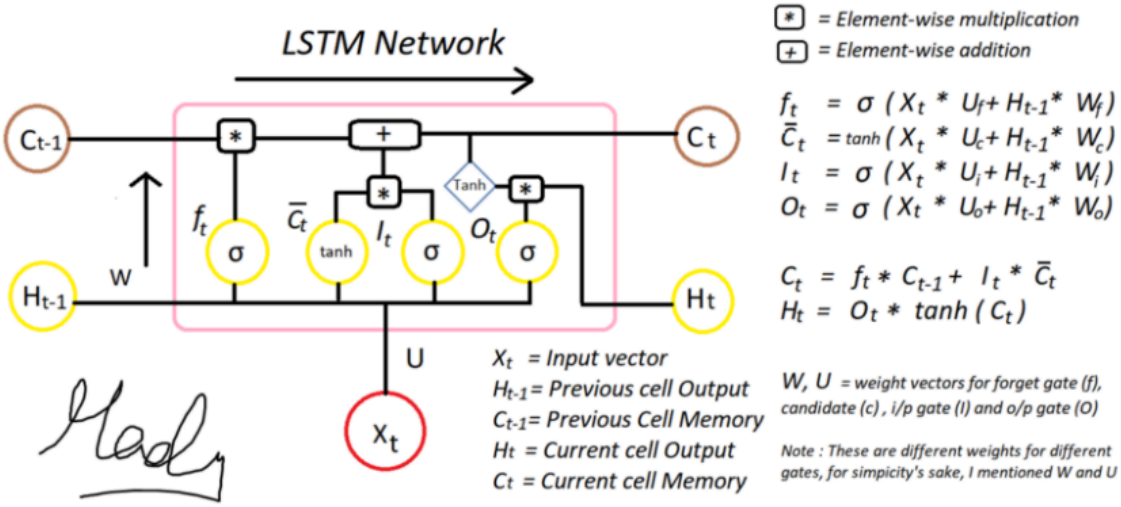




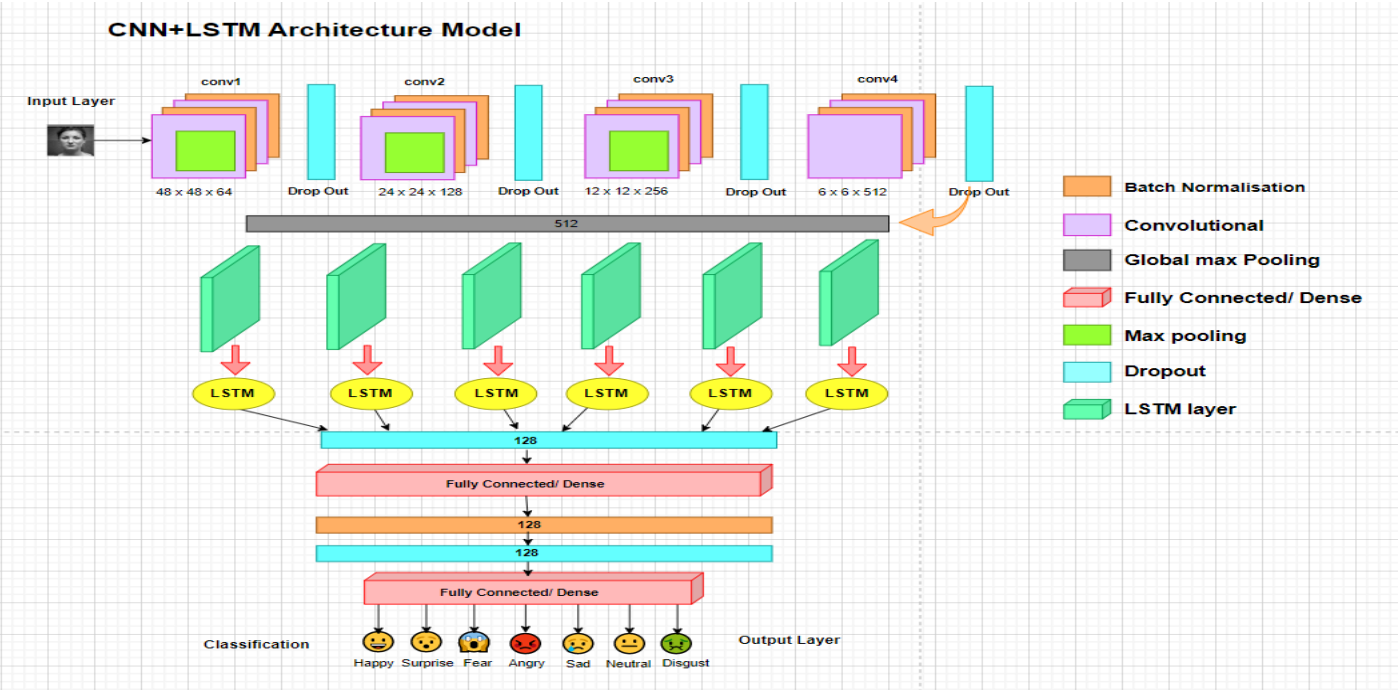
CNN+LSTM:

Architecture:
LSTMs are unfolded RNNs. Unlike simple recurrent neural networks where only a tanh function is used to remember the past layers, LSTMs have hidden layers containing gated unit or gated cell. Four layers interact to produce the output of that cell along with the cell state which is like a conveyer belt that carries information forward in the most smooth manner. LSTMs comprise three logistic sigmoid gates and one tanh layer. The output is usually in the range of 0-1 where '0' means 'reject all' and '1' means 'include all'. Gates limit the information that is passed through the cell and determine which part of the information will be needed by the next cell and which part will be discarded using weights assigned to them.
The hidden state, cell state or memory, and current data or input are three inputs put into the forget gate which consists of a sigmoid layer and pointwise multiplication.

The second sigmoid layer is the input gate that decides what new information is to be added to the cell. It takes two inputs h_{t-1} and x_t . The tanh layer creates a vector C_t of the new candidate values. Together, these two layers determine the information to be stored in the cell state. Their pointwise multiplication tells us the amount of information to be added to the cell state. The result is then added with the result of the forget gate multiplied with the previous cell state to produce the current cell state. Next, the output of the cell is calculated using a sigmoid and a tanh layer. The sigmoid layer decides which part of the cell state will be present in the output whereas the tanh layer shifts the output in the range of [-1,1]. The results of the two layers undergo pointwise multiplication to produce the output ht of the cell.



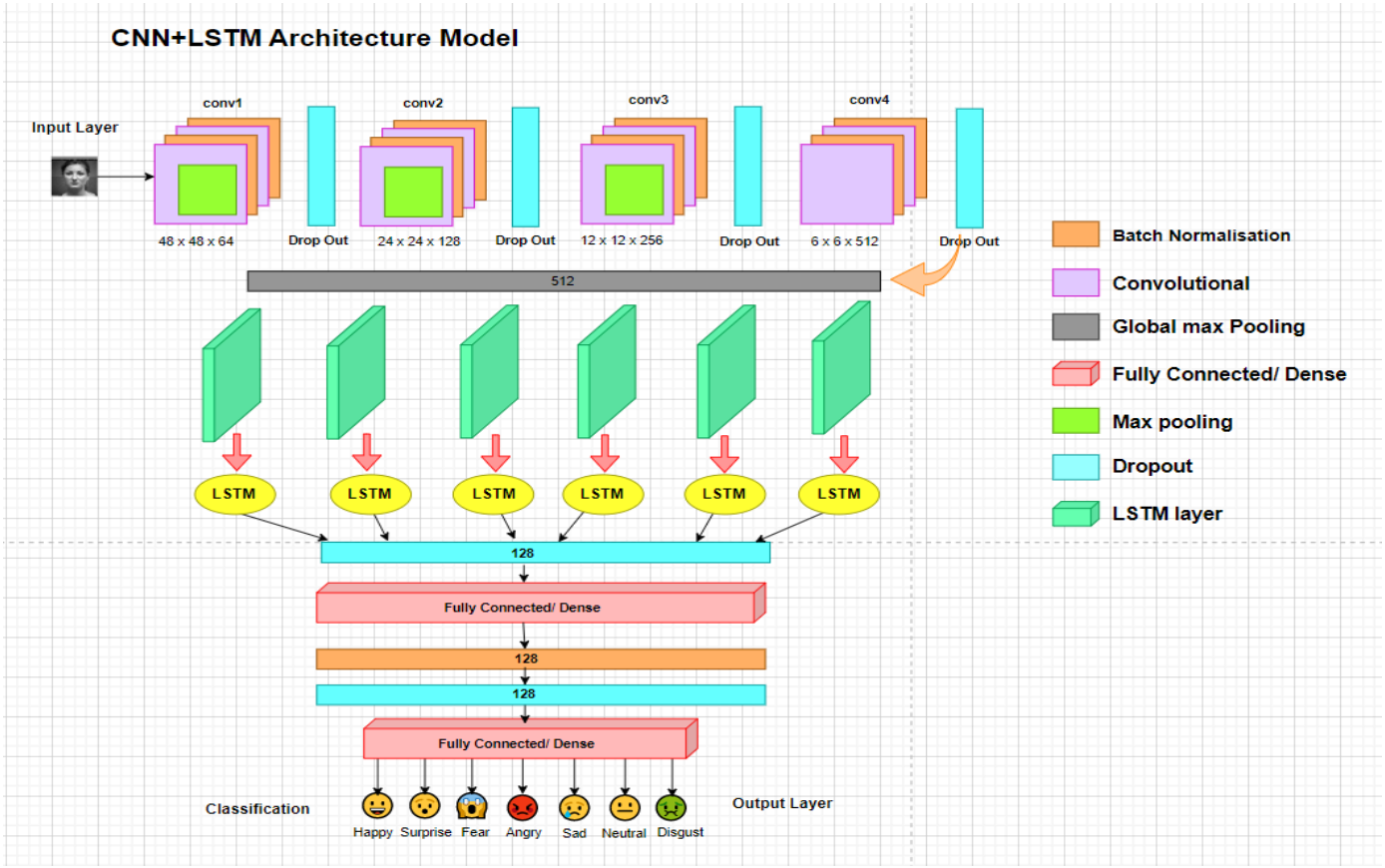
Source: [ImageLink]

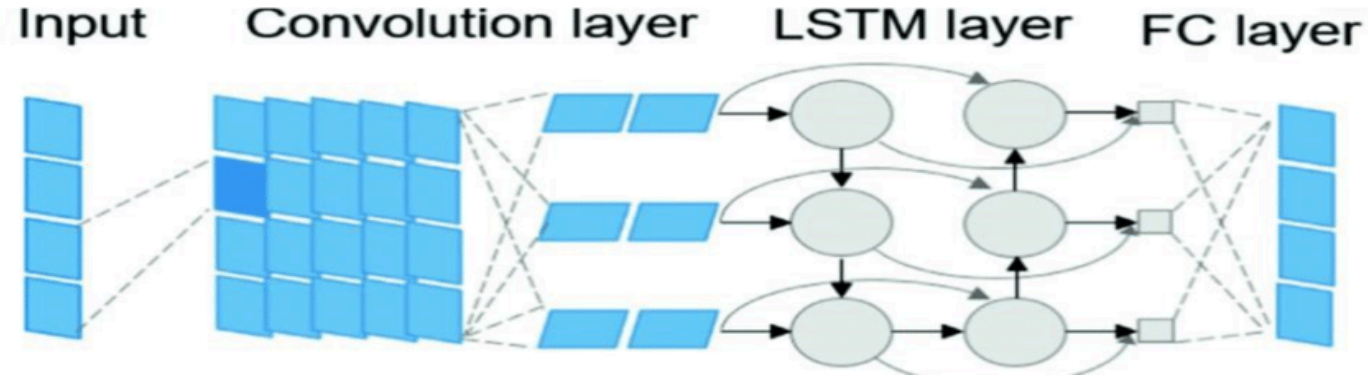


Our code:

CK+48:

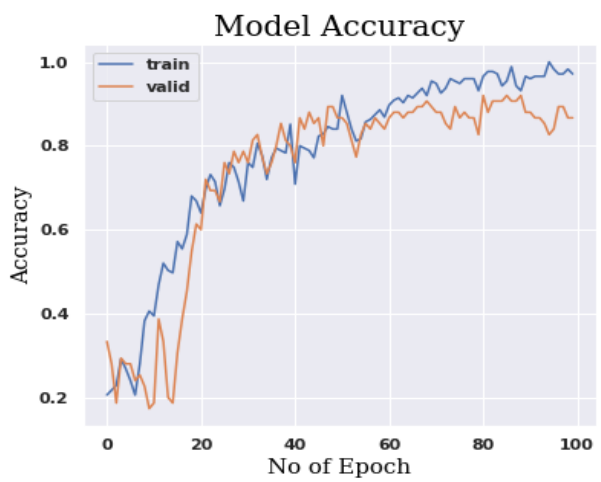
8 Conv2D layers with elu activation layer followed by Batch Normalization and dropouts which is followed by Global max pooling layer to reshape data and input to two LSTM layer. Finally a Dense layer with softmax activation is used.

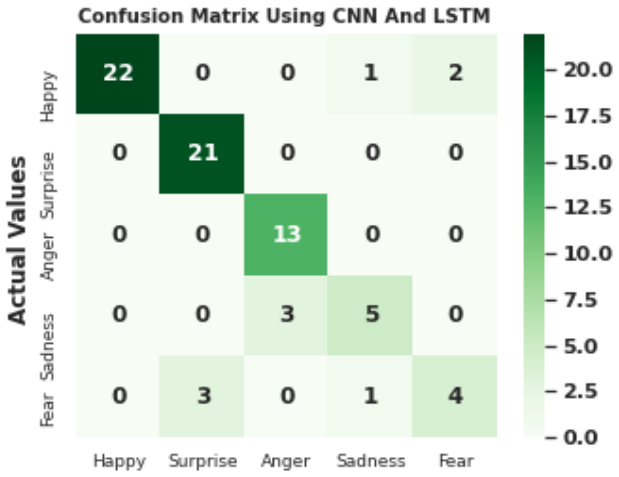

CNN+LSTM Architecture Model
Input Layer
conv1
conv2
conv3
conv4
48 x 48 x 64
Drop Out
24 x 24 x 128
Drop Out
12 x 12 x 256
Drop Out
6 x 6 x 512
Drop Out
512
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
128
Fully Connected/ Dense
128
128
Fully Connected/ Dense
Classification
Happy
Surprise
Fear
Angry
Sad
Neutral
Disgust
Output Layer
Batch Normalisation
Convolutional
Global max Pooling
Fully Connected/ Dense
Max pooling
Dropout
LSTM layer

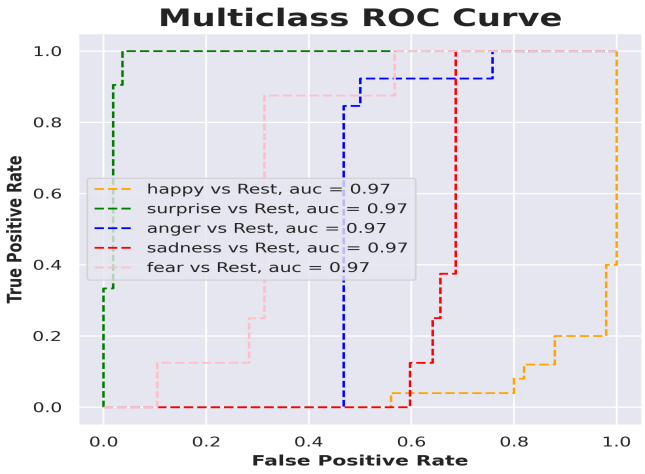

Input
Convolution layer
LSTM layer
FC layer

Results:

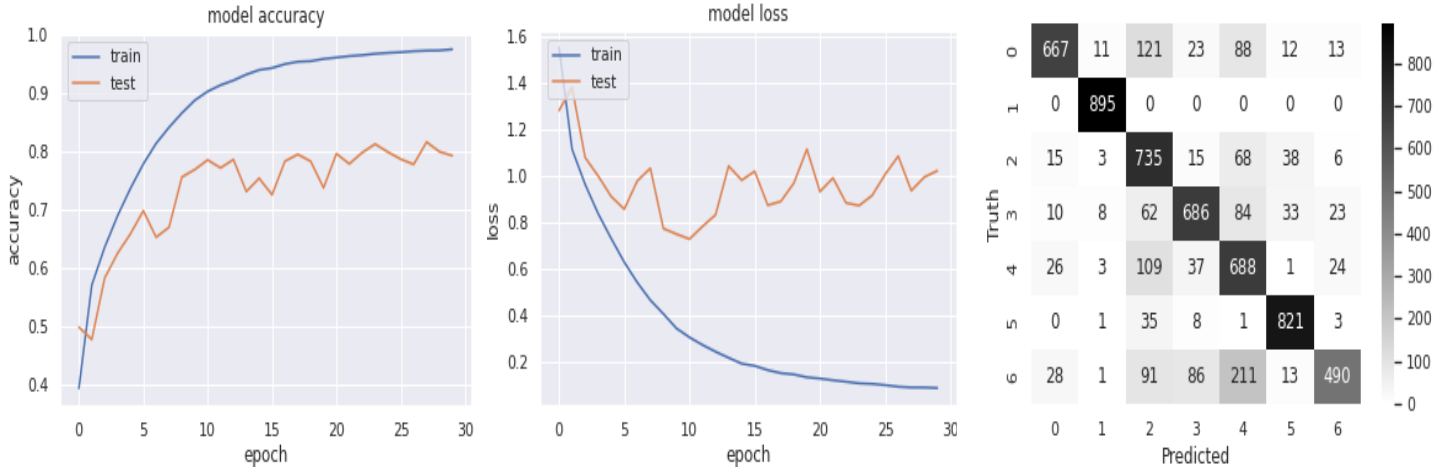
CK+48:



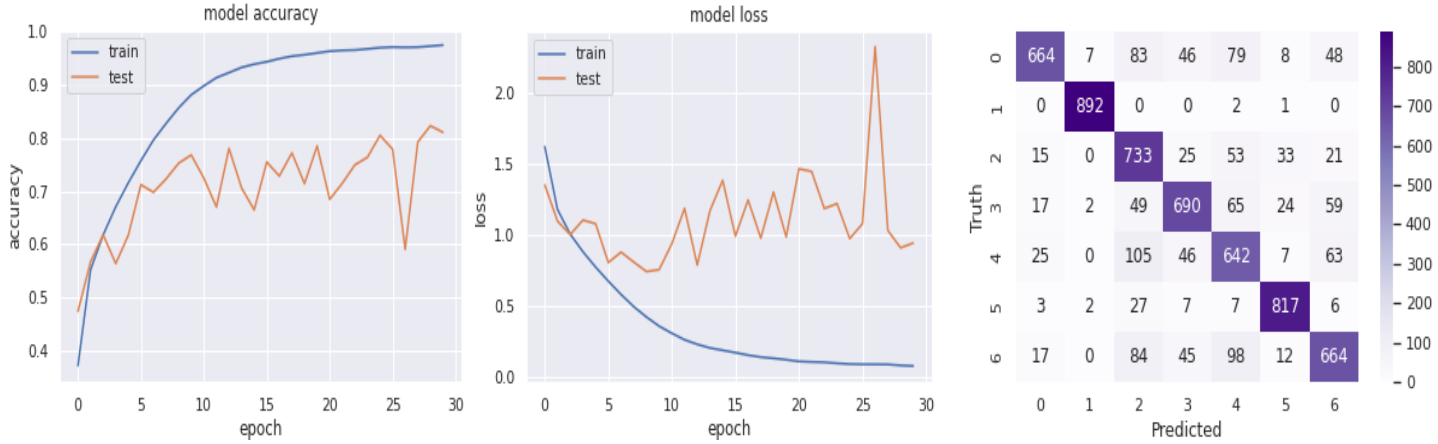




FER2013



FER2018:



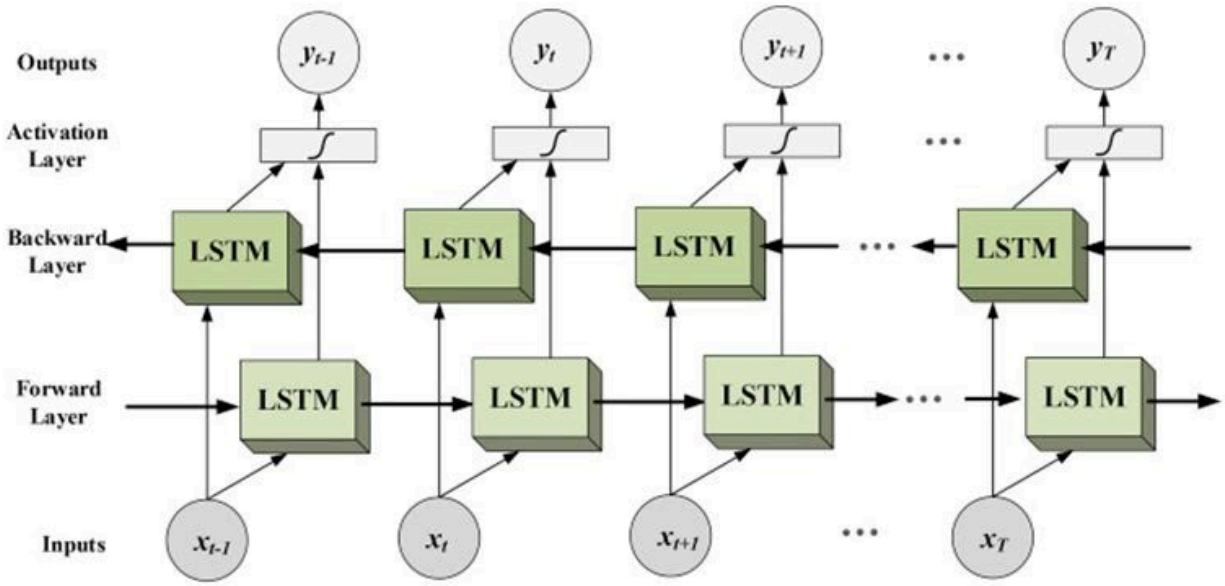
CNN+BiLSTM:

Each training sequence is presented forwards and backward to two independent recurrent nets, coupled to the same output layer in Bidirectional RNN (BRNN)
Comprehensive, sequential knowledge about all points before and after each point in a given sequence
Conventional RNNs: only being able to use the previous contexts. Bidirectional RNNs (BRNNs) process data in both ways with two hidden layers that feed forward to the same output layer.
BRNN + LSTM = LSTM that can access long-range context in both input directions.

Though what they are suited for is a very complicated question, BiLSTMs show very good results as they can understand the context better.
Let's say we try to predict the next word in a sentence, on a high level what a unidirectional LSTM will see is “The boys went to ….”

And will try to predict the next word only by this context, with bidirectional LSTM you will be able to see information further down the road for example
Forward LSTM:
“The boys went to …”
Backward LSTM:
“… and then they got out of the pool”
One can see that by using the information from the future it could be easier for the network to understand what the next word is. Similarly in Facial expressions, it can be used to predict facial emotion more accurately

Bidirectional recurrent neural networks(RNN) are really just putting two independent RNNs together. This structure allows the networks to have both backward and forward information about the sequence at every time step

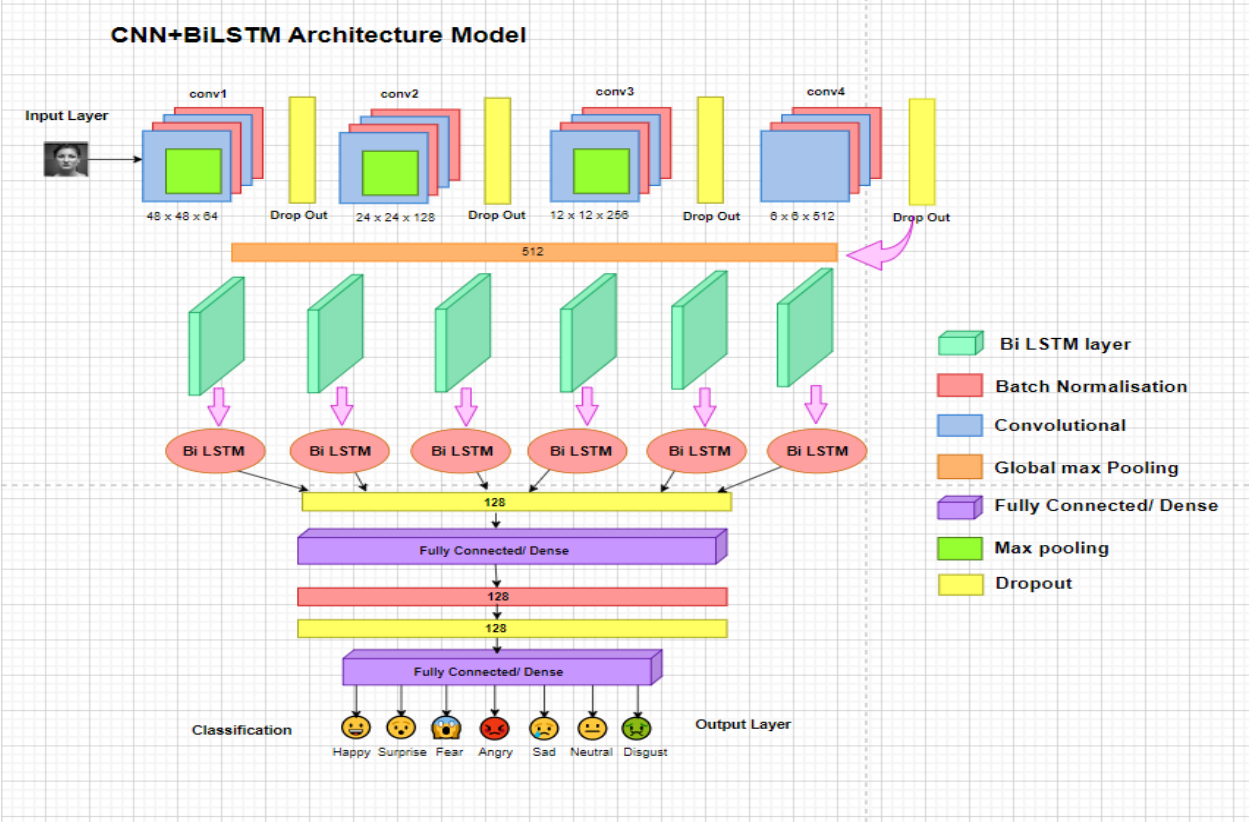


Source: ImageLink

Using bidirectional will run your inputs in two ways, one from past to future and one from future to past what differs this approach from unidirectional is that in the LSTM that runs backward you preserve information from the future and using the two hidden states combined you are able in any point in time to preserve information from both past and future.
Both activations(forward, backward) would be considered to calculate the output y^ at time t

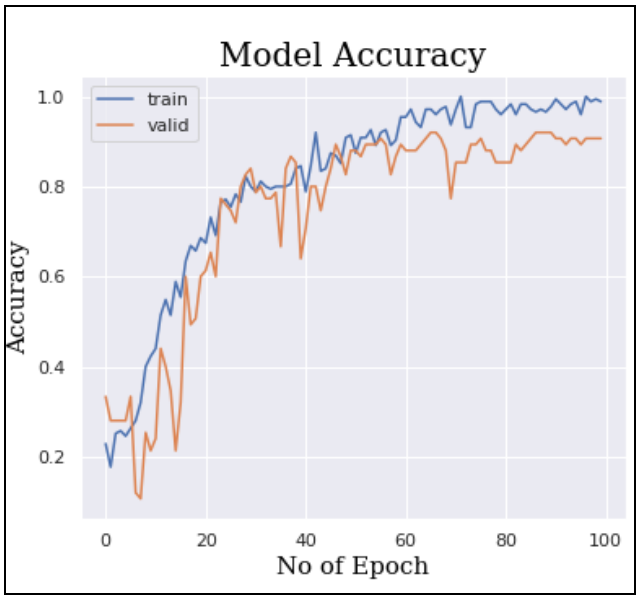
Our Implementation:

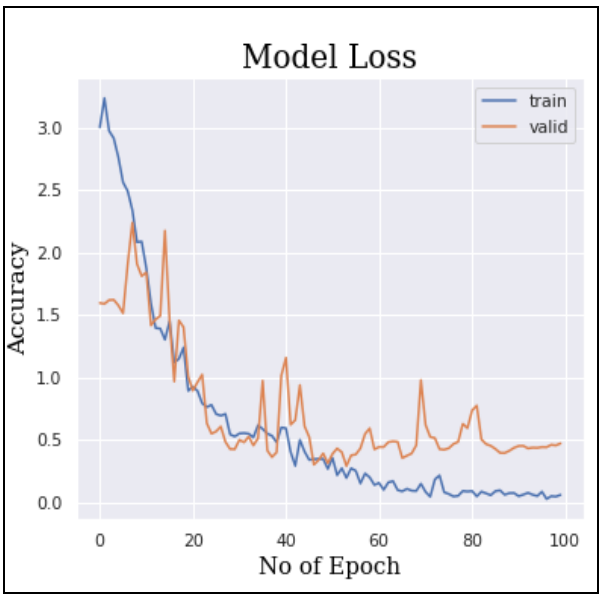
8 Conv2D layers with elu activation layer followed by Batch Normalization and dropouts which is followed by Global max pooling layer to reshape data and input to two Bidirectional LSTM layers. Finally a Dense layer with softmax activation is used.

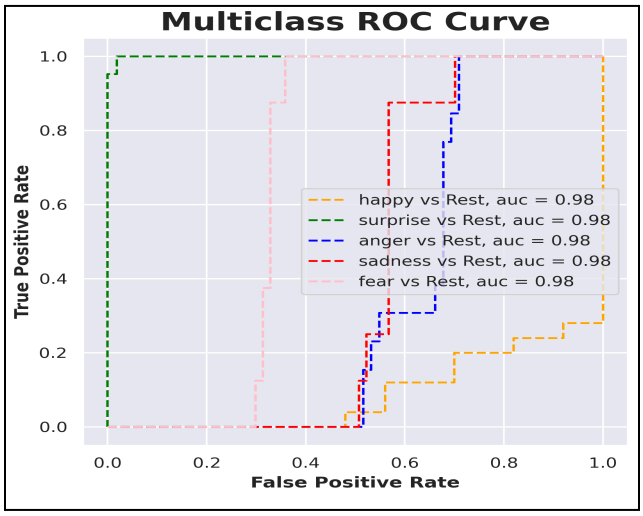


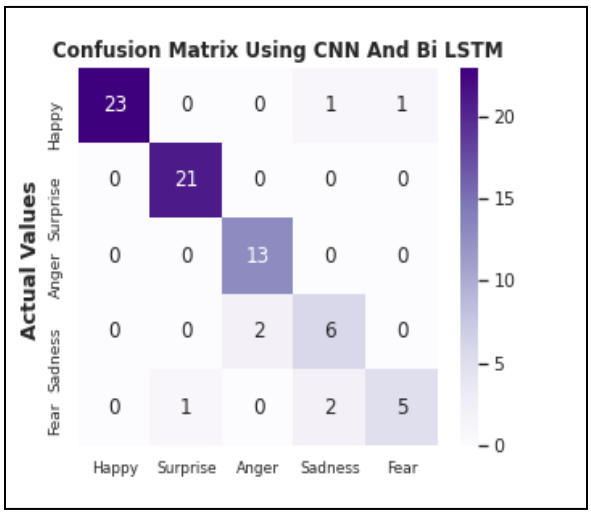
Our result:

CK+48:


Model Accuracy
train
valid
Accuracy
No of Epoch

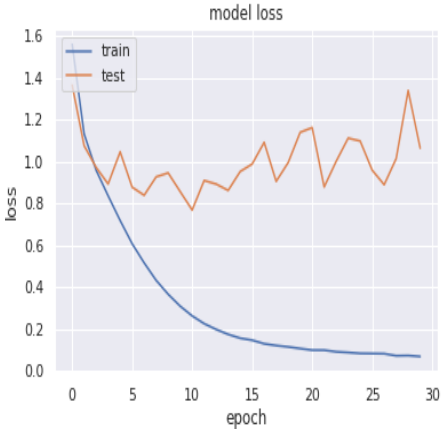

Model Loss
train
valid
Accuracy
No of Epoch

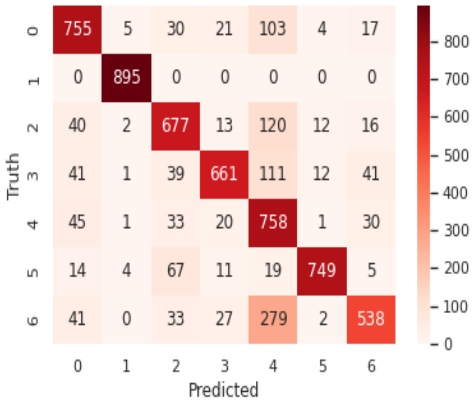

Multiclass ROC Curve
True Positive Rate
False Positive Rate
happy vs Rest, auc = 0.98
surprise vs Rest, auc = 0.98
anger vs Rest, auc = 0.98
sadness vs Rest, auc = 0.98
fear vs Rest, auc = 0.98


Confusion Matrix Using CNN And Bi LSTM
Actual Values
Happy
Surprise
Anger
Sadness
Fear

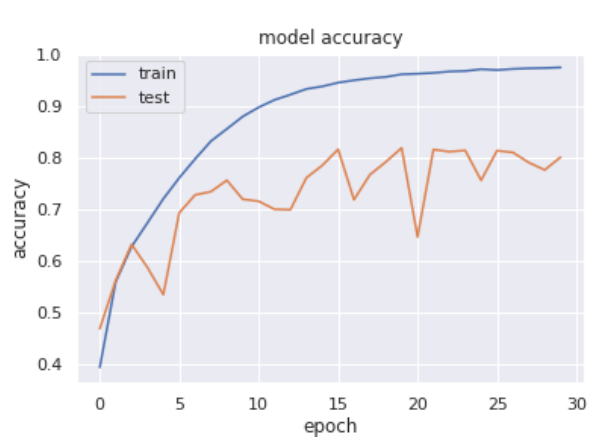
FER2013:

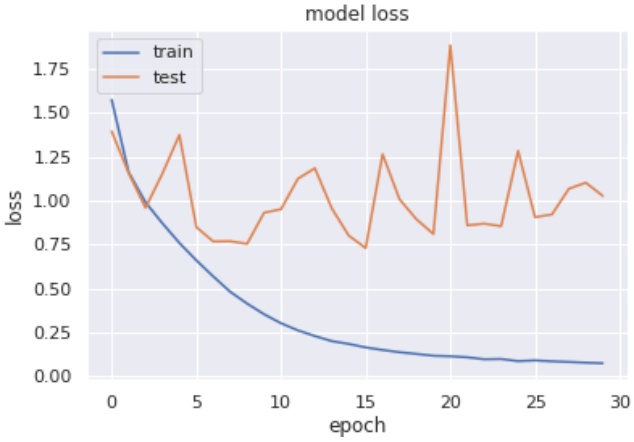

model accuracy
train
test
accuracy
epoch

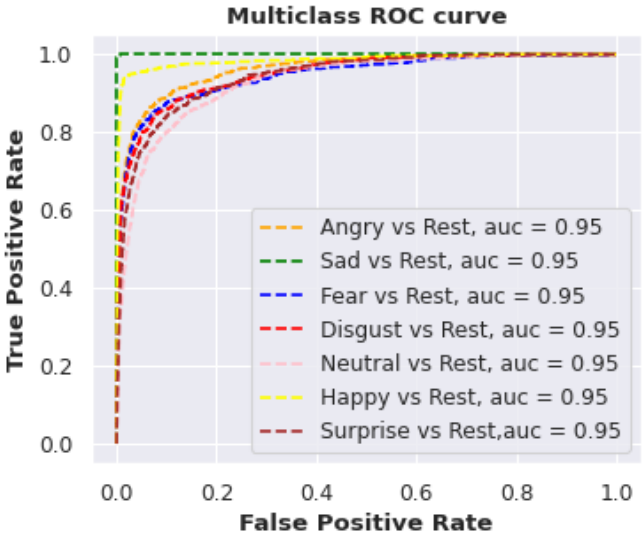

model loss
train
test
loss
epoch

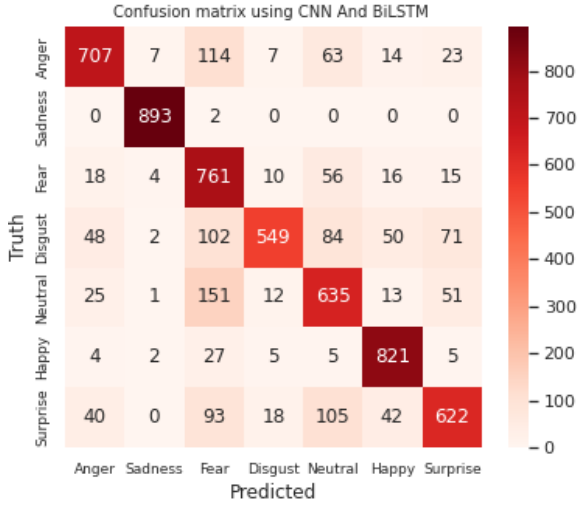

Truth
Predicted

FER2018:

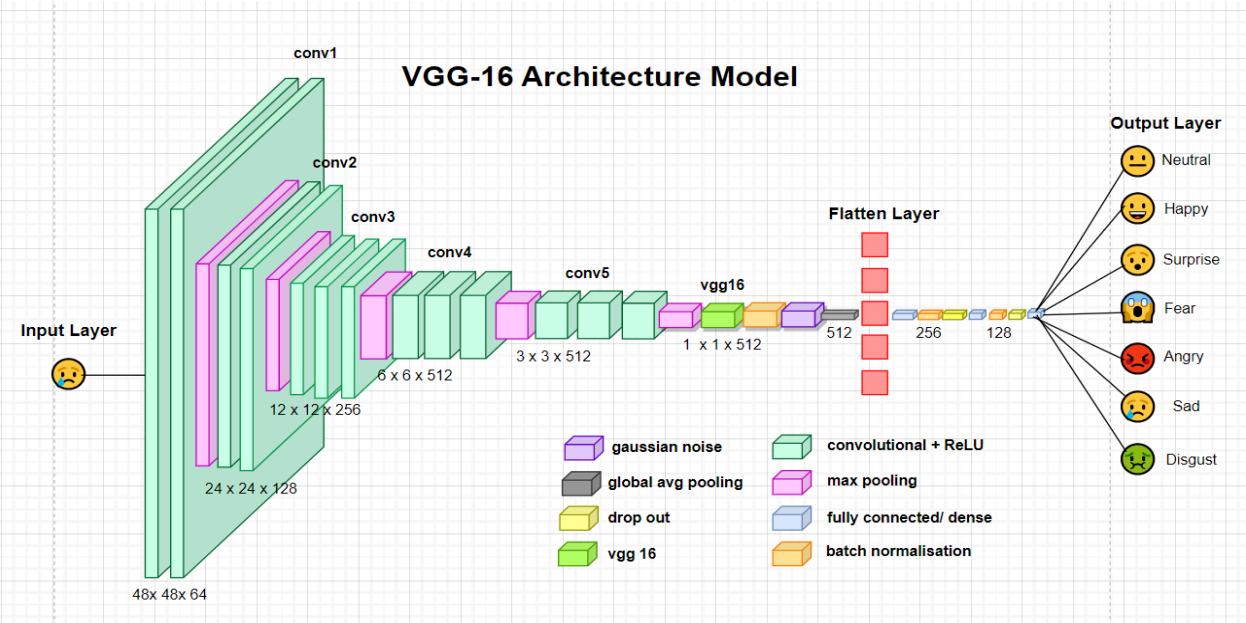

model accuracy
train
test
accuracy
epoch


model loss
train
test
loss
epoch


Multiclass ROC curve
True Positive Rate
False Positive Rate
Angry vs Rest, auc = 0.95
Sad vs Rest, auc = 0.95
Fear vs Rest, auc = 0.95
Disgust vs Rest, auc = 0.95
Neutral vs Rest, auc = 0.95
Happy vs Rest, auc = 0.95
Surprise vs Rest,auc = 0.95


Confusion matrix using CNN And BiLSTM
Truth
Predicted
Anger Sadness Fear Disgust Neutral Happy Surprise
707 7 114 7 63 14 23
0 893 2 0 0 0 0
18 4 761 10 56 16 15
48 2 102 549 84 50 71
25 1 151 12 635 13 51
4 2 27 5 5 821 5
40 0 93 18 105 42 622

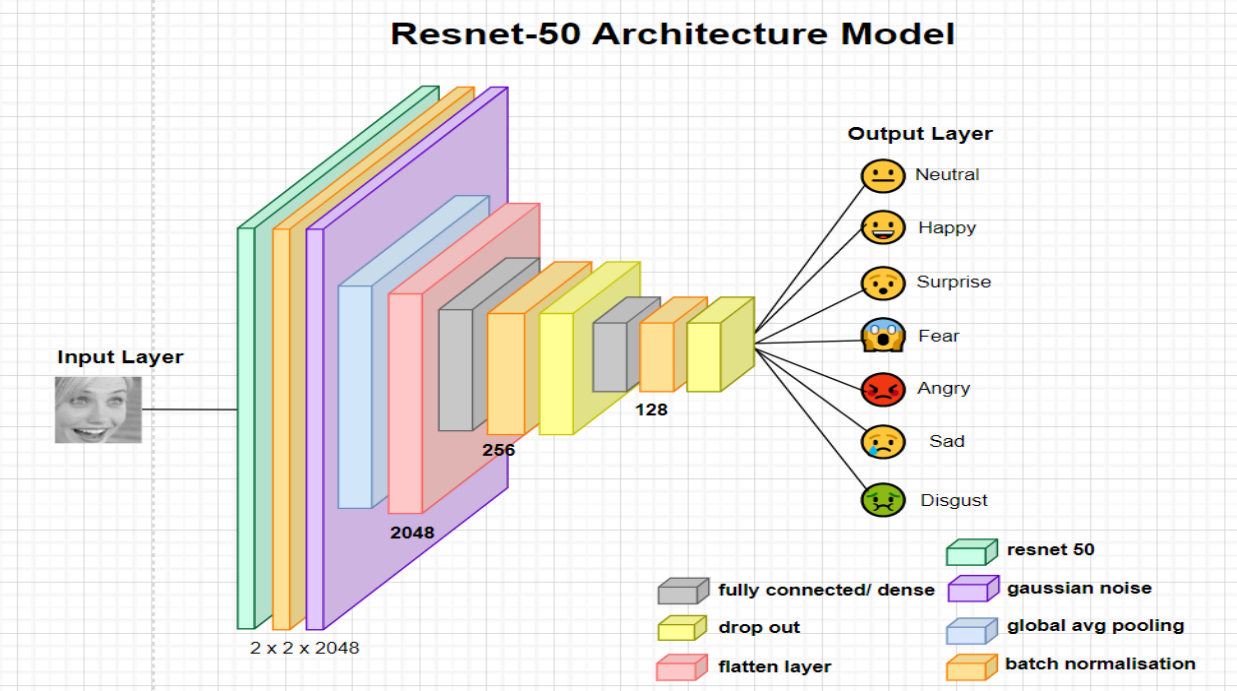
VGG16:



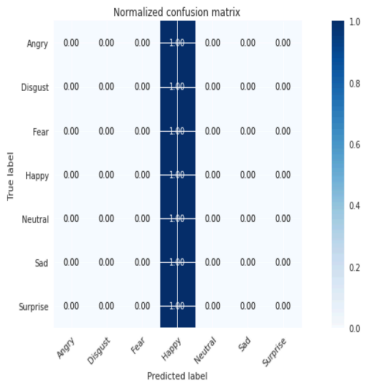
VGG-16 is a convolutional neura l network that is 16 layers deep. Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2.It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it having 16 layers that have weights.

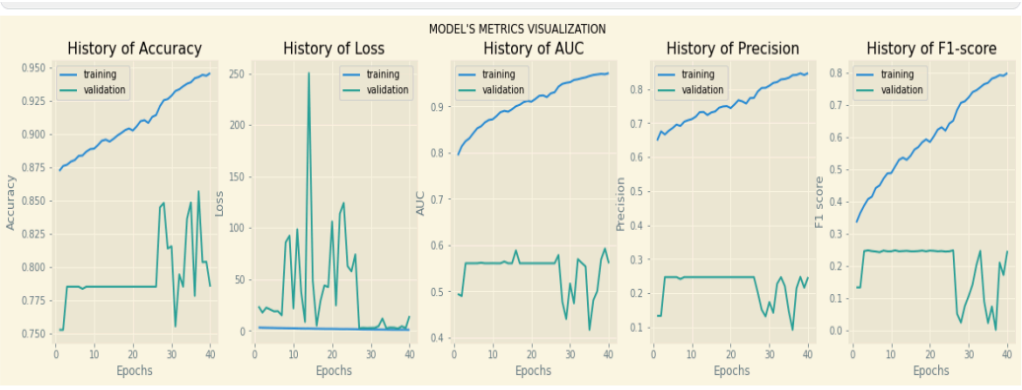
With the help of VGG16 we were able to achieve validation accuracy of 70%. To increase the accuracy we add a sequence of layers -batch_normalization,gaussian_noise, global_average_pooling, flatten_1,dense,dropout and achieved an accuracy of 90.3%.

ResNet50:



ResNet is a short-hand for Residual Neural Networks. It was introduced to solve the problem of Vanishing/Exploding gradients. It uses a concept called Residual blocks. It uses skip connections to skip between certain layers in the network when one layer is activated. The advantage of adding this type of skip connection is that if any layer hurt the performance of architecture then it will be skipped by regularization





Results & comparison:

METHOD	DATASET	TRAIN_ACCURACY	VAL_ACCURACY	WEIGHTED AVG PRECISION	WEIGHTED AVG RECALL	WEIGHTED AVG F1 SCORE
CNN	CK+48	89%	84%	0.88	0.88	0.87
	FER2013*	76.4%	66.9%	0.68	0.67	0.67
	FER2018	70%	64%	0.67	0.67	0.66
CNN+LSTM	CK+48	98%	87%	0.87	0.87	0.86
	FER2013	97.40%	79.7%	0.81	0.80	0.79
	FER2018	97.33%	78.33%	0.80	0.78	0.78
CNN + BiLSTM	CK+48	98%	93%	0.93	0.92	0.92
	FER2013	97.77%	79.97%	0.83	0.80	0.80
	FER2018	97.54%	80.10%	0.83	0.80	0.81
VGG16	CK+48	98%	92%	0.58	0.56	0.58
	FER2013	95%	82.8%	0.62	0.62	0.64
	FER2018	95%	83%	0.66	0.65	0.68
VGG19	CK+48	98%	92%	0.56	0.56	0.54
	FER2013	95%	84%	0.68	0.58	0.61
	FER2018	96%	84%	0.70	0.62	0.66
RESNET50	CK+48	96%	85%	0.69	0.55	0.63
	FER2013	78%	78%	0.25	0.25	0.245
	FER2018	76.4%	66.9%	0.23	0.22	0.24

Conclusion:

We tuned our parameters, increased layers, and observed so many things are not necessary so we have applied max-pooling, dropout layers to obtain good accuracy. We have used softmax to get probabilistic value to get more information about the expressions.

We observed a combination of CNN and LSTM provides a slighter better result than simple CNN model. However the results of Bidirectional LSTM with CNN didn't show much different results from CNN+LSTM. Then, we applied the VGG16, VGG19, and ResNet50 transfer learning methods where we observed the best results.

Facial emotion recognition has been a hot issue of debate in the past years regarding its usability and relevance.

Experts argue that even if we truncate our database or range to average adults from urban society, ignoring the anomalies that might increase if we consider children, "It is not possible to confidently infer happiness from a smile, anger from a scowl, or sadness from a frown, as much of current technology tries to do when applying what are mistakenly believed to be the scientific facts" as famously concluded by the psychologist and neuroscientist Lisa Feldman Barrett.

They found limited reliability (emotions don't always generate the same facial movements), lack of specificity (emotion-facial movement pairs don't present unique mappings), and limited generalizability (cultural and contextual factors haven't been sufficiently documented). The exposure of biases in face and emotion recognition technologies gave way to a more crucial debate.

However, even though it may not be the most accurate predictor there are questions as to whether it is a safe practice and if we are ready for our emotions continuously analyzed and read by the government or other sources.

Keeping aside the drawbacks and dilemma of FER, if we focus on how much it has been contributing and how it can be revolutionizing the world in coming years.

Emotion recognition is already used by schools and other institutions since it can help prevent violence and improves the overall security of a place. Some companies use AI with emotion recognition API capabilities as HR assistants. The system helps determine whether the candidate is honest and truly interested in the position by evaluating intonations, facial expressions, and keywords, and creating a report for the human recruiters for final assessment. It is a powerful tool in the field of business and marketing where customer emotion is the essence of a business. The Healthcare industry is using it for physicians to know which patient to prioritize seeing first.

FER can be useful for autistic patients and mentally disabled people who have difficulty analyzing, understanding, and reading facial expressions.

FER may contribute a lot to the interaction with Human-Robot Interface (HRI), and turn how Robots can take on jobs where a surface knowledge of user emotion is required and can be useful in making AI pets.

References:

- Zhang, T. (2017, June). Facial expression recognition based on deep learning: a survey. In International Conference on Intelligent and Interactive Systems and Applications (pp. 345-352). Springer, Cham
- An Overview on Long Short Term Memory (LSTM) - Analytics Vidhya
- Artificial Intelligence, A guide to Intelligent Systems, second edition Michael Negnevitsky.
- Raghuvanshi, A., & Choksi, V. (2016). Facial expression recognition with convolutional neural networks. CS231n Course Projects, 362.
- Pramerdorfer, C., & Kampel, M. (2016). Facial expression recognition using convolutional neural networks: state of the art. arXiv preprint arXiv:1612.02903.
- Facial Expression Recognition System Using Machine Learning by Sanghyuk Kim, Gwon Hwan An, and Suk-Ju Kang
- J. Kim, J. K. Lee, and K. M. Lee, “Deeply-recursive convolutional network for image super-resolution,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, US, June 26-July 1, 2016
- Facial Expression Recognition Using CNN-LSTM Approach Md. Mohsin Kabir1 , Tanvir Ahamed Anik1 , Md. Shahnewaz Abid1 , M. F. Mridha1 , Md. Abdul Hamid
- P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, 2010, pp. 94-101, doi: 10.1109/CVPRW.2010.5543262.