A Project Report

On

Emotional Analysis using Audio, Visual and Text Inputs

BY

Anirudh Bhupalarao

F2018AAPS1242H

Under the supervision of

Dr. Jabez Christopher

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

CS F366: LABORATORY PROJECT

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)

HYDERABAD CAMPUS

(March 2021)

ACKNOWLEDGMENTS

A special thanks to Prof. Jabez Christopher for advising me through this project and guiding me throughout my journey. I would also like to thank my peers as well for helping me whenever I needed it and providing valuable input to the discussions regarding the research we had. The project "Emotional Analysis using Audio, Video and Text Inputs" has been an excellent experience for me and it will definitely act as a stepping stone in my Artificial Intelligence research career.

Anirudh Bhupalarao

2018AAPS1242H

Birla Institute of Technology and Science-Pilani,

Hyderabad Campus

Certificate

This is to certify that the project report entitled "**Emotional Analysis using Audio, Video and Text Inputs**" submitted by **Mr. Anirudh Bhupalarao** (ID No. 2018AAPS1242H) in partial fulfillment of the requirements of the course CS F366, Laboratory Project Course, embodies the work done by him under my supervision and guidance.

Date: 28th April 2021 **(Dr. Jabez Christopher)**

BITS- Pilani, Hyderabad Campus

CONTENTS

ABSTRACT

The main objective behind this project is to provide a pathway for analysis of student attentiveness in a classroom, based on their actions and responses. Human beings communicate their feelings in several ways, including facial expressions, speech, writing, gestures and many more. Hence, for a robust model to accurately perform emotional analysis, a combination of two or more of these features must be taken as inputs. Emotion Recognition has become a significant field of study in the Artificial Intelligence area and due to social media, there is an abundance of datasets available to researchers to tackle this problem. Researchers from diverse fields are using a wide variety of implementations ranging from traditional machine learning models to deep learning models that consist of deep multi-layered neural networks. Out of all the novel methods that are currently being used to aid in emotion recognition, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are two of the most commonly used, simply because of their network depth, structure and robustness. In this project, we focus on three of the main emotion input features, i.e., visual, audio and text. The algorithms we implement for text-based emotion recognition outperforms many state of the art models by a good margin, achieving around 83% on the test dataset, while the algorithms we implemented for visual-based emotion recognition also scored around 80%, outperforming a few novel methods. We also do a literature review on audio-based emotion recognition and conduct a thorough analysis on the results offered by state-of-the-art methods involved in the field. Ultimately, this paper provides the necessary knowledge and insights for researchers to understand better the research state of Emotional Analysis, remaining challenges, and future directions in this field.

LITERATURE SURVEY

Emotion Recognition has sparked tremendous interest in researchers from many fields. From natural language processing, computer vision/image processing, to psychology and cognitive sciences, new models/methods are being designed frequently. However, the development in Artificial Intelligence when it comes to emotional analysis is tremendous.

Surveys by Acheampong, Wenyu and Mensah [1] and Deng and Ren [2] adequately sum up the work done and datasets available in Textual Emotion Recognition (TER). [1] starts by listing various emotion models – Discrete Emotion Models (DEMs) and Dimensional Emotion Models (DiEMs). DEMs include Paul Ekman's model (that distinguishes emotions based on six basic categories), Robert Plutchik's model (according to Ekman, there are only a few primary emotions that appear in opposite pairs and contain complex emotions when they are combined) and Orthony, Clore, and Collin's (OCC) model (stated that emotions emerged as a result of how people viewed experiences and that emotions differed in intensity). Plutchik and Russel both presented two-dimensional DiEMs that presuppose that emotions are not autonomous and that there exists a connection between them hence the need to position them in a spatial space. Thus, dimensional models position emotions on a dimensional space depicting how related emotions are and usually reflecting the two central fundamental behavioral states of good and bad. Both the papers actively review around 20 textual datasets and perform a thorough analysis of the best detection approaches (ML, Rule Construction, Word Embedding) for each of those datasets. [2] also goes further to mention the challenges brought about by each of the proposed approaches. High-quality datasets, incomplete emotional information in textual representation, language barriers are a few challenges highlighted in their paper.

Yoon, Byun and Jung [3] proposed a novel deep dual recurrent encoder model that utilizes both text and audio data in emotion recognition. They performed experiments on the IEMOCAP dataset and got better results than many previous state-of-the-art methods (68.8% to 71.8% accuracies). The paper explores two models, one with attention and one without. To overcome the limitations of the neutral class misclassification bias frequently exhibited by previous models, they built a multimodal dual recurrent encoder (MDRE) that uses MFCC features, prosodic features and transcripts for sequential audio information, statistical audio information and textual information, respectively. This MDRE further connects two models – Audio Recurrent Encoder and a Text Recurrent Encoder. Both these encoders use Gated Recurrent Units (GRUs) as they yield comparable performance to LSTMs and include a smaller number of weight parameters. They use a max encoder step of 750 for the audio input and 128 for the text input since it covers the longest transcripts. For each model, the number of hidden units and layers in the RNN are chosen based on detailed hyperparameter search experiments. The dataset's vocabulary contains 3,747 words, including the "UNK" token, which represents unknown words, and the "PAD" token, which indicates padding information added while preparing mini-batch data. As for feature extraction, to extract speech information from audio signals, they used MFCC values, which are widely used in analyzing audio signals. Prosodic features were also used to extract additional information from the dataset. The Opensmile toolkit was used for most of the feature extraction part. Yoon and co. split the IEMOCAP dataset in each fold into training, testing and development datasets (8:1.5:0.5),

respectively. They achieved around 60-70% scores after analyzing model performances using mean scores and standard deviations.

The current leading approach for Facial Expression Recognition (FER) is convolutional neural networks (CNNs). In the paper, “Going Deeper with Convolutions” [4], they proposed Inception, a deep convolutional neural network architecture that set the current state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014. For CNNs, the more number of layers, the better. However, this increases the number of parameters to be trained in the model, making the enlarged network more prone to overfitting, especially if the number of labeled examples in the training set is limited. The Inception model consists of around 27 layers, including convolutional, pooling and inception layers. The paper also highlighted another major drawback of increased network sizes, computational resources. If two convolutional layers are chained, any uniform increase in the number of their filters results in a quadratic increase of computation. If the added capacity is used inefficiently, then a lot of computation is wasted. After thorough experimentation, they concluded that moving from fully connected architectures to sparsely connected architectures would solve this issue. By utilizing a carefully crafted architecture that allows for increasing the depth and width of the network while keeping the computational budget constant, they were able to improve the usage of computing resources. The architectural decisions were based on the Hebbian theory and the intuition of multi-scale processing to maximise output. In the Inception model, the use of average pooling before the classifier enables adapting and fine-tuning our networks for other label sets easily.

A survey by Li and Deng [5] summarizes all the novel methods and their architectures for Facial Expression Recognition (FER) in immense detail. The paper reviews all static and dynamic facial expression datasets while also reviewing the best preprocessing techniques (Face alignment, Data augmentation, Face normalization) for FER. Li and Deng also examine the existing novel deep neural networks built for FER, as well as the associated training methods suggested to solve expression-specific issues. They elaborately highlight the results and explain each novel method in detail while also listing the pros and cons of each framework. From GANs to Cascaded Networks to Ensemble Networks, the paper discusses many state-of-the-art methods by splitting them into two main groups: deep FER networks for static images and deep FER networks for dynamic image sequences. Since some of the datasets under consideration lack explicit data classes for training, validation and testing, the paper summarizes the expression recognition performance along with information about the data selection and grouping methods. The existing deep FER systems focus on two key issues: the lack of plentiful, diverse training data and expression-unrelated variations, such as illumination, identity and head alignment. To deal with inadequate training data and overfitting, deep FER now uses pre-training and fine-tuning techniques. Pre-training and fine-tuning the network in various stages with auxiliary data ranging from large-scale objection or facial recognition datasets to small-scale FER datasets has proven to be a practical strategy shown to be particularly useful. Comparisons are also made of different types of methods for static and dynamic images in terms of variations, computational efficiency, accuracy and other parameters.

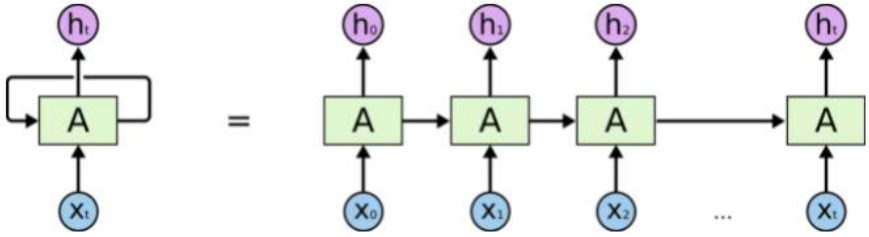
Simonyan and Zisserman's [6] main contribution to FER is a thorough evaluation of networks of increasing depth using an architecture that consists of small (3×3) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 16–19 weight layers. VGG's architecture consists of a stack of convolutional layers followed by fully connected layers. The paper incorporates 1×1 convolutional layers to increase the nonlinearity of the decision function without affecting the convolutional layers' receptive fields. It attained a top-1 validation error of around 25.5% and a top-5 validation error of around 8.0%. Given that FER is a data-driven task and that training a sufficiently deep network to catch subtle expression-related deformations necessitates a significant volume of training data, the lack of both quantity and accurate training data is a major challenge for deep FER networks. In addition to this, in facial datasets, outliers are bound to exist and since face expression is shown and interpreted differently by individuals of various ages, races, and genders, an optimal dataset should contain a large number of sample images with precise face attribute identifiers, not just expressions but also other attributes. Hence, the need for additional parameters/features. Individually, all these studies have achieved high accuracies and low error rates by tweaking and tuning layers according to their inputs and parameters. However, creating an ensemble of networks that combines and utilizes all three inputs – audio, video and text, can significantly increase the performance of a final model.

METHODOLOGY

I. Text-Based Emotion Recognition (TER)

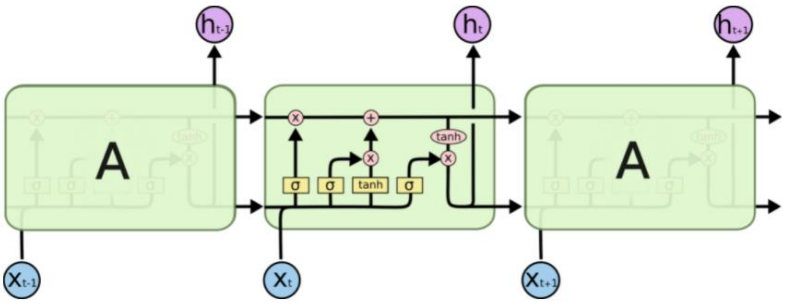
This section describes the methodologies that are applied to aid us in TER. We can break this down further into four sections: (i) Data Preprocessing (ii) Model Architecture and Training (iii) Results of other novel methods.

Based on the eight to nine research papers reviewed, our main focus has come to be Recurrent Neural Networks (RNNs). These neural networks withhold information using loops in their networks; hence they are best used for Natural Language Processing, Time Series Analysis etc.



Long Short Term Memory networks or LSTMs for short, are one of the most widely used and researched RNNs due to their capability of learning long-term dependencies. They were introduced by Hochreiter and Schmidhuber [7] to tackle the prevalent issue at the time i.e., remembering information over long periods. As depicted in the picture below, LSTMs have a chain-like structure but have four neural networks in each repeating unit.

The first sigmoid layer is called "forget gate layer" and handles which information to 'forget'. It outputs binary values in the cell state to keep or throw away the information learnt. The second sigmoid layer is called "input gate layer". It decides the values to be updated and is connected to the tanh layer that creates a vector of the new values. Then the network combines these two equations to make an update to the state.



Below are the equations involved leading to the update state in a standard LSTM network:

$$f_t = \sigma\left(W_f * [h_t, x_t] + b_f\right)$$

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh\left(W_C * [h_{t-1}, x_t] + b_C\right)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

Data Preprocessing: Sentiment Analysis is the branch of natural language processing that deals with identifying and extracting information from sentences, online forms, dialogues etc. The polarity of a given text in a document or sentence - whether the conveyed emotion in a document or a sentence is optimistic, negative, or neutral - plays a fundamental role in sentiment analysis. Joy, Anger and Disgust are all examples of advanced sentiment classification statements. There are several steps involved in preprocessing the data for sentiment analysis. For this we use the DailyDialogue dataset which contains 11318 conversations between two random speakers. As the dataset is not explicitly split into training, test and validation sets, we divide this into a 11118 train set, 1000 test set and 1000 validation set conversations. All the sentences in these conversations are labelled according to the emotion they portray i.e., Neutral, Anger, Disgust, Fear, Happiness, Sadness and Surprise, and are numbered in that exact order. To input data into our model, we convert the dataset into vector form and perform NLP techniques using the NLTK library to remove punctuations, stop words (a, an, that, is) and further take care of missing data or any blank spaces in the data inputs. Tokenization is done to count the frequency of words in a sentence and to further give weight to each word. Each token is then passed through a word-embedding layer, which transforms a word index into a multi-dimensional vector containing additional context between words. The final preprocessing dataset looks as shown in the figure below. After converting the sentences into vector format and encoding all the y-labels, the data preprocessing step is complete and we can pass it through our model.

```
['Say , Jim , how about going for a few beers after dinner ?', ' You know that is tempting but is
really not good for our fitness .', ' What do you mean ? It will help us to relax .', " Do you really
think so ? I don't . It will just make us fat and act silly . Remember last time ?", " I guess you are
right.But what shall we do ? I don't feel like sitting at home .", ' I suggest a walk over to the gym
```

```
['say jim how about going for few beers after dinner', 'you know that is tempting but is really not
good for our fitness', 'what do you mean will help us relax', 'do you really think so don t will just
make us fat and act silly remember last time', 'guess you are right but what shall we do don t feel
like sitting at home', 'suggest walk over gym where we can play singsong and meet some our friends',
```

Model Architecture and Training: A recurrent neural network is a type of neural network that has been shown to perform well when dealing with sequential data. Since text comprises of a series of sentences, using a recurrent neural network to solve text-related problems is a must. To solve the sentiment classification dilemma, we will use LSTM networks, which are a kind of RNN. We begin by creating the embedding layer with a limit of 10000 unique words and an embedding dimension of 200 to match the review length threshold that we use. The embedding layer is primarily used in models and datasets when the one-hot encoded vectors have high dimensionality and are sparse. This helps in visualizing relationships between words in sentences easily. Each term in the embedding

matrix is replaced by an index that is used to look up the vector. We investigate what terms are close to each other in a multi-dimensional space since the embedded vectors are updated during the deep neural network's training period. Before initialising the LSTM layers, we include a spatial dropout layer to promote independence throughout feature maps. This is recommended over normal dropout layers because a standard dropout will not regularise the activations if neighbouring frames within function maps are closely correlated, which will also result in an overall learning rate decrease. There are two LSTM layers, the first layer having 250 neurons and the second having 100 neurons. The recurrent dropout rates are selected after conducting thorough hyperparameter tuning and given to the model to provide optimal results. We also added a Dense layer to our model that uses the softmax function for predicting the emotion type from the text-RNN's last hidden state. As shown in the table below, there are over 2 million trainable parameters.

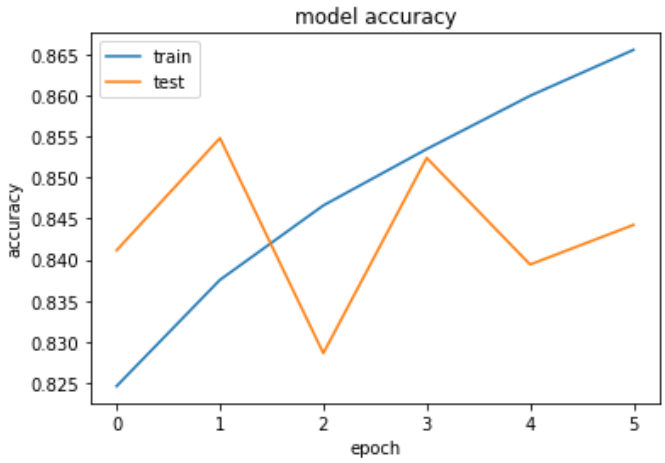
Layer (type)	***Output Shape***	***Param #***
embedding (Embedding)	(None, 200, 200)	2000000
spatial_dropout(SpatialDropout)	(None, 200, 200)	0
lstm (LSTM)	(None, 200, 250)	451000
lstm_1 (LSTM)	(None, 100)	140400
dense (Dense)	(None, 7)	707

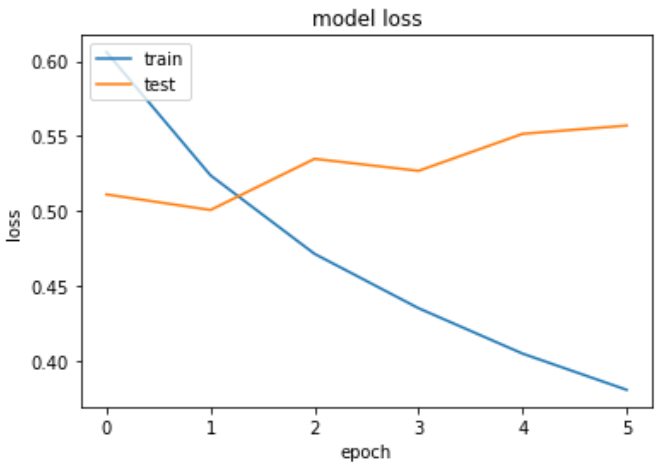
Total Params: 2,592,107 Trainable Params: 2,592,107

We train our model over 6-7 epochs, with a batch size of 128 giving the most optimum results. Categorical cross-entropy is used as the loss function which incorporates a cross-entropy function as well as a softmax function to calculate the loss of the model.

$$CE = -\log\left(\frac{e^{S_p}}{\sum_j^C e^{S_j}}\right)$$

Where S_p is the LSTM score for the positive class. Since the labels are one-hot in Multi-Class classification, only the positive class S_p holds the term in the loss. Over the training phase, the training accuracy fluctuates around 83-86% with a constant reducing loss of around 0.4-0.5. Furthermore, our model outperforms the many novel methods mentioned in the literary review by a significant margin by achieving a test accuracy of 84% and a model loss of ~0.53.





Results of other Novel Methods: We further compare the results we got through our RNN model with other state-of-the-art machine learning models to prove our theory of LSTMs being the best fit for sentiment analysis. For this comparison, we use a Convolutional Neural Network and an ordinary Neural Network.

CNN Model:

Layer (Type)	***Output Shape***	***Param #***
embedding (Embedding)	(None, 200, 200)	2000000
conv1d (Conv)	(None, 196, 128)	128128
global_max_pooling1d (Pooling)	(None, 128)	0
dense (Dense)	(None, 7)	903

Total params: 2,192,031 Trainable params: 2,192,031

As shown in the table above, our CNN model uses global max pooling to act as a replacement to the fully connected layers. In the last mlpconv layer, the idea is to create one function map for each corresponding segment of the classification task. We take the average of each feature map and feed the resulting vector directly into the softmax layer, rather than adding fully connected layers on top of the feature maps. One benefit of global average pooling is that there are no parameters to optimise, so overfitting is avoided at this layer. While we used categorical cross-entropy loss function for our LSTM model, we observed that a binary cross-entropy loss function provided the most optimum results for our CNN model. The remaining hyperparameters are the same values as used in the LSTM i.e., we train the model over 6-7 epochs with a batch size of 128. We summarise the results in table 1 in the next section. As seen in the table, we get a model loss of around approximately 0.16; however, the model accuracy underperforms while compared to the LSTM model.

NN Model:

Layer (Type)	*Output Shape*	*Param #*
embedding (Embedding)	(None, 200, 200)	2000000
flatten (Flatten)	(None, 40000)	0
dense (Dense)	(None, 7)	280007

Total params: 2,280,007 Trainable params: 2,280,007

We build a basic Neural Network model with a flattening and a fully connected layer. The flattening layer is used to flatten (convert into a single column) the pooled feature map matrix to pass into the neural network's fully connected layer. The predicted classes are found in the output layer. The data is sent across the network, and the prediction error is measured. To boost the prediction, the error is backpropagated via the system.

Model	Training Acc.	Training Loss	Test Acc.	Test Loss
CNN	80.29%	0.138	78.66%	0.159
NN	78.63%	0.232	75.64%	0.541

(Table 1)

Error Analysis: In the LSTM model, the high level of cross-predictions among similar emotions is a visible pattern in the predictions. The majority of the model's misclassifications for the happiness emotion are for the neutral class. In addition to this, anger and disgust have similar misclassifications. We believe this is due to a slight distinction between certain emotion pairs, making disambiguation more difficult. In both the CNN and NN models, the neutral class is another class with a high number of false-positives. The primary explanation for this may be its majority in the class distribution of the emotions under consideration. Overall, we found that most emotion classes are often confused with the neutral class, which is situated in the middle of the activation-valence space, making it difficult to distinguish from the other classes.

Interestingly, when compared to the other two models, the LSTM model reveals more considerable prediction benefits in predicting the happiness class. This finding is possible because the model will benefit from variations in word distributions in happiness and neutral expressions, which gives the LSTM networks more emotional information. On the other hand, the CNN model falsely predicts instances of the sadness class as happiness, despite the fact that both emotional states are diametrically opposed. Another explanation for all of these misclassifications, we conclude, is the lack of context to each of these networks i.e., discontinuation of emotion flow from the previous utterance, while predicting the emotion class.

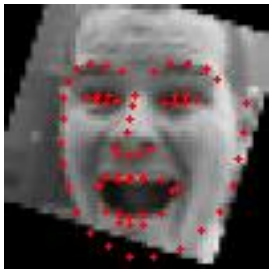
II. Facial Expression Recognition (FER)

We break down the method used into three parts: (i) Data preprocessing, (ii) CNN architecture and (iii) CNN training.

Data Preprocessing: Preprocessing data when it comes to facial expression recognition is primarily face detection and registration, facial alignment and facial landmarking. In unconstrained scenarios, patterns that are unrelated to facial expressions, such as surroundings, illuminations, and head poses, are reasonably common. Therefore, preprocessing is necessary to align and normalize semantic information transmitted by the face before training the deep neural network to learn meaningful features.

One of the main preprocessing techniques used for expression recognition is Facial Landmarking. They are used to localize and represent salient features of the face (nose, mouth, eyebrows, eyes, and outline of the face). Landmarking helps in facial alignment, which we have used later as another preprocessing technique. Detecting facial landmarks consists of localizing the face and detecting the key facial features in the ROI (Region of Interest). A training set of labeled facial landmarks of images is used. These images are manually labeled, specifying (x, y) coordinates to each facial landmark on the face. Along with this, we also calculate the probability of distances between pairs of input pixels. Given this training data, an ensemble of regression trees is trained to estimate the landmark positions from the pixel intensities. There are 68 such coordinates that map to features on the face.

Facial alignment is also a commonly used preprocessing step in FER-related work. It can be used as a form of data normalization. This is done with the help of facial landmarks where the angle of the line connecting the landmarks of the eyes is set to zero i.e., horizontal.



CNN architecture: We mainly work with two novel models – Inception v1 and VGG. Firstly, the Inception model is a convolutional neural network that is 27 layers deep. This model contains a layer called the inception layer, which is a combination of 1x1, 3x3, 5x5 convolution layers with their outputs concatenated into a single input for the next hidden layer. It helps the internal layers choose which filter size is important for the information needed to be learned. So, even though the ROI (face) is different, the layers function to recognize the face accurately. As we are dealing with grayscale images of size 48x48, we removed the 5a and 5b layers to prevent feature loss.

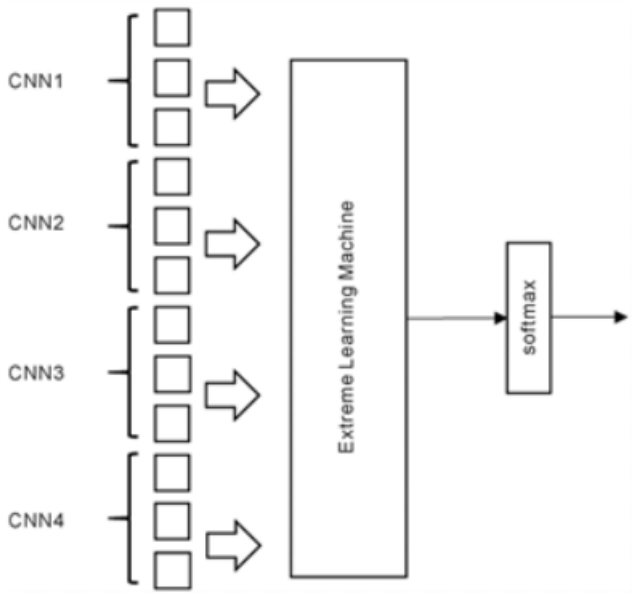
type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

The second model we are going to use is based on the VGG model. The VGG-Face model contains 22 layers with 37 deep units. We decrease a few layers from this for the same reason as before i.e., to avoid feature loss; however, the basic structure of the Deep CNN remains the same.

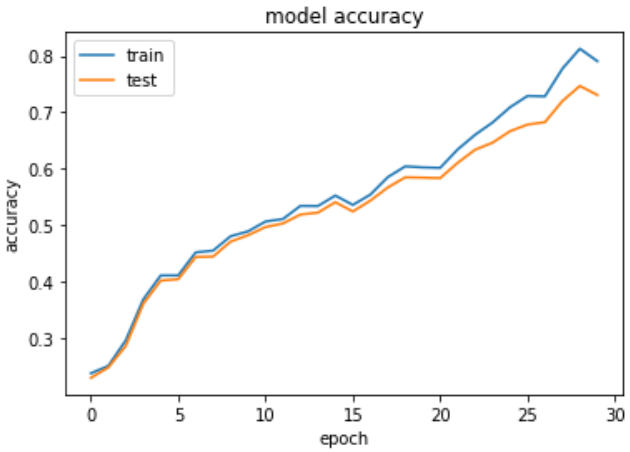
layer	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
type	input	conv	relu	conv	relu	mpool	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv
name	–	conv1_1	relu1_1	conv1_2	relu1_2	pool1	conv2_1	relu2_1	conv2_2	relu2_2	pool2	conv3_1	relu3_1	conv3_2	relu3_2	conv3_3	relu3_3	pool3	conv4_1
support	–	3	1	3	1	2	3	1	3	1	2	3	1	3	1	3	1	2	3
filt dim	–	3	–	64	–	–	64	–	128	–	–	128	–	256	–	256	–	–	256
num filts	–	64	–	64	–	–	128	–	128	–	–	256	–	256	–	256	–	–	512
stride	–	1	1	1	1	2	1	1	1	1	2	1	1	1	1	1	1	2	1
pad	–	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	1

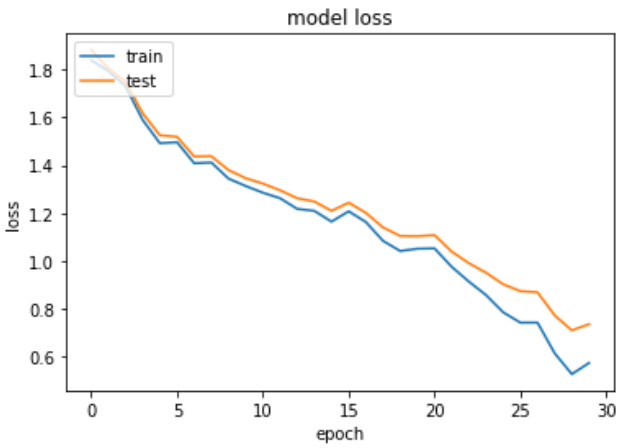
layer	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
type	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	relu	mpool	conv	relu	conv	relu	conv	softmx
name	relu4_1	conv4_2	relu4_2	conv4_3	relu4_3	pool4	conv5_1	relu5_1	conv5_2	relu5_2	conv5_3	relu5_3	pool5	fc6	relu6	fc7	relu7	fc8	prob
support	1	3	1	3	1	2	3	1	3	1	3	1	2	7	1	1	1	1	1
filt dim	–	512	–	512	–	–	512	–	512	–	512	–	–	512	–	4096	–	4096	–
num filts	–	512	–	512	–	–	512	–	512	–	512	–	–	4096	–	4096	–	2622	–
stride	1	1	1	1	1	2	1	1	1	1	1	1	2	1	1	1	1	1	1
pad	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0

CNN training: We use a network ensemble of both the models presented in the previous section. Research has shown that assemblies of multiple networks can outperform individual networks. For feature level ensembles, concatenated input features learned from different networks is the most common strategy. In our final model, we use a network ensemble to weigh and average the outputs of both the CNN models to get a better result. The image shown below is an accurate representation of how modern network ensembles work and are mainly used along with Deep CNNs to outperform novel individual models.



The models are trained for around 25-30 epochs each based and hyperparameter tuning is done to ensure that there is no overfitting. The initial learning rate, batch size and decay are fixed at 0.0005, 256 and 0.96, respectively. An optimal dropout rate and softmax activation are used for both the models while training. Over several runs of the final model, the average training accuracy was around 80-82% with the preprocessing techniques and around 67-68% without preprocessing. This outperforms many state-of-the-art models by almost 10-12% under similar conditions. The overall loss was dropping to around 0.499–0.6. A batch size of 256 for a dataset this large seemed to be the sweet spot. Below are the graphs of the results obtained. The training time on a CPU averaged about 15-20 mins simply because of the number of parameters being trained.





CONCLUSIONS

We have presented models that significantly outperform many state-of-the-art models both in accuracies and loss values. An LSTM-based neural architecture for textual emotion recognition, which in comparison to current approaches, considers the speaker's features when processing incoming expressions, providing a more nuanced meaning for the expression. And a CNN-based neural architecture for facial emotion recognition that uses preprocessed data and facial landmarking to accurately detect facial features and improve the results of the novel models significantly. Both of the models prove to be advances in the Emotional Analysis field of Artificial Intelligence. We agree that multi-party emotional engagement, personality simulation, and complex emotional monitoring will lead to new research avenues. We hope that this paper's thorough analysis of the most recent TER and FER technologies will offer new perspectives for future studies in this area and want to expand the modalities to include Speech Emotion Recognition in future work. We also want to look at the application of the attention mechanism to data obtained from a variety of sources. This method seems to be a good bet for uncovering improved learning schemes that will improve success in speech expression detection and other multimodal classification activities.

The research presented in this paper also found that the use of TER and FER for essential or life-saving applications has not been sufficiently investigated. These areas include crime prevention and deterrence by analysis of victim messages or facial features to recognize threatening terms, analysis of patient messages to assess patient depression levels, and so on. Finally, even though they are limited, there also exists other resources that cover other rich cultures like Hindi, German, French etc. that can greatly encourage further research in that direction. This we suggest because an individual's cultural affiliations have a significant impact on their expressed feelings toward circumstances.

REFERENCES

[1] F. A. Acheampong, C. Wenyu, H. N. Mensah, “Text-Based Emotion Detection: Advances, Challenges, and Opportunities”, 2020.

[2] J. Deng, F. Ren, “A Survey of Textual Emotion Recognition and Its Challenges”, IEEE Transactions on Affective Computing, 2021.

[3] S. Yoon, S. Byun, K. Jung, “Multimodal Speech Emotion Recognition using Audio and Text”, IEEE Spoken Language Technology Workshop, 2018.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, “Going Deeper with Convolutions”, IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[5] S. Li, W. Deng, “Deep Facial Expression Recognition: A Survey”, IEEE Transactions on Affective Computing, 2020.

[6] K. Simonyan, A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, 2014.

[7] S. Hochreiter, J. Schmidhuber, “Long Short-Term Memory”, 1997.

[8] Chen, M. Wang, S. Liang, P. P. Baltrusaitis, T. Zadeh, A. Morency, “Multimodal Sentiment Analysis with Word-Level Fusion and Reinforcement Learning”, 19th ACM International Conference on Multimodal Interaction, 2017.

[9] K. Sailunaz, M. Dhaliwal, J. Rokne, and R. Alhajj, “Emotion Detection from Text and Speech: A Survey”, Social Networking Analysis and Mining, 2018.

[10] N. Majumdar, S. Poria, D. Hazarika, R. Mihalcea, A. Gelbukh, E. Cambria, “DialogueRNN: An Attentive RNN for Emotion Detection in Conversations”, AAAI Conference on Artificial Intelligence, 2019.

[11] E. Cambria, B. White, “Jumping NLP Curves: A Review of Natural Language Processing Research”, IEEE Computer Intelligence Mag., 2014.

[12] Allouch M, Azaria A, Azoulay R, Ben-Izchak E, Zwilling M, Zachor DA, “Automatic Detection of Insulting Sentences in Conversation”, IEEE International Conference on the Science of Electrical Engineering, Israel, 2018.

[13] Kao E. C. C, Liu C. C, Yang T. H, Hsieh C. T, Soo V. W, “Towards Text-Based Emotion Detection: A Survey and Possible Improvements”, International Conference on Information Management and Engineering, 2009.

[14] J. Lee, I. Tashev, “High-Level Feature Representation using Recurrent Neural Network for Speech Emotion Recognition”, Annual Conference of the International Speech Communication Association, 2015.

[15] S. Mirsamadi, E. Barsoum, C. Zhang, “Automatic Speech Emotion Recognition using Recurrent Neural Networks with Local Attention”, Acoustics, Speech and Signal Processing, 2017 IEEE International Conference.

[16] l. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet Classification with Deep Convolutional Neural Networks”, Advances in Neural Information Processing Systems, 2012.