Static Facial Expression Analysis using Bidirectional Long Short-term Memory (LSTM) Model

Kanza Zahid Sahban
u6999472@anu.edu.au

Research School of Computer Science, Australian National University

Abstract. Human emotion recognition plays a vital role not just in interpersonal relations but has a wide variety of applications in real life. Human brain recognizes emotional state of a person using visual cues and facial expression and also visualizes a facial image related to a specific emotion when it hears or reads the labels of those emotions. This paper is aimed at assessing as to whether a neural network can classify emotions using facial expressions bidirectionally just like humans do. This paper uses *Static Facial Expressions in the Wild* (SFEW) database, which has extracts of Pyramid of Histogram of Gradients (PHOG) and Local Phase Quantization (LPQ) features, to recognize seven emotions. It builds on our previous work in which experiments were carried out on the dataset using two different models, Artificial Neural Network (ANN) and Bidirectional Neural Network (BDNN) and reported average testing accuracy of 67.58% and 24.69% respectively. It was assessed that within the realm of bidirectionality and deep learning, a different architecture may produce better results. Therefore, this paper analyzes static facial expressions using Bidirectional Long Short-term Memory (LSTM) model which provides competitive average test accuracy exceeding 56.90%. The results have established that emotions can be reverse classified using images to a large extent. Moreover, the paper also discusses the use of LSTM on non-sequential data.

Keywords: Long short-term memory (LSTM), Recurrent Neural Network (RNN), Bidirectional Neural Network (BDNN), Neural Network, Local Phase Quantization (LPQ), Pyramid of Histogram of Gradients (PHOG) features, Emotion recognition, Facial expression analysis, Static Facial Expressions, Static Facial Expressions in the Wild (SFEW) dataset, Non-sequential data

1 Introduction

Emotion Recognition Technology is a multibillion-dollar industry which is extracting facial features from images and videos to detect emotions using Artificial Intelligence and the results are being used by large multinational companies to understand customers' reaction to their products, security organizations to flag suspicious faces, employers to scan the candidate's face to read his/her reactions, teachers to assess involvement of students, doctor to find diagnostic clues, clinical psychologists and psychiatrists for behavioral assessment, police and intelligence agencies for lie detection; the applications are endless. Changes in facial expression are reflective of human emotions. Advances in technology have enabled researchers to read the measurements of facial motions changing with facial muscles, and facilitate automatic analysis of facial expression, thus making it free of human error and skill level.

The paper uses a dataset called Static Facial Expressions in the Wild (SFEW) which contains a subset of static facial expression images extracted from a temporal facial expressions database Acted Facial Expressions in the Wild (AFEW), which were extracted from movies. The database covers unconstrained facial expressions, varied head poses, large age range, occlusions, varied focus, different resolution of face and close to real world illumination [1]. Although movies are shot in a predicted and controlled environment, they provide situations close to reality and facial expressions of the actors are much more realistic than the ones recorded in lab-controlled environment. Hence it is a quality data recorded in varied realistic environment and seems perfect for studies regarding human face expression analysis.

All muscles in human body are connected by nerves. The nerves receive input from spinal cord and brain and these connections are bidirectional i.e. just like the way nerve triggers a muscle contraction after receiving brain signals, the same information can also travel back to the brain in reverse order. This paper takes inspiration from the working of human brain and aims to assess bidirectionality in facial expression analysis just like it is done by human brain. The motivation behind our work is its application in assistive applications. If implemented correctly, bidirectional analysis of facial expression can also be useful in the field of psychology, computer vision, human-computer interaction, virtual reality, education, entertainment, and pattern recognition.

Previously, we presented Static Facial Expression Analysis using Bidirectional Neural Network (BDNN) and Artificial Neural Network (ANN). The models predicted emotions using the facial features present in SFEW dataset. The results of the experiment were meant to determine whether emotion recognition can be done in both directions e.g. specific facial features can determine that a person is happy, but the label should also predict the values of facial features that correspond to a happy emotion. It was observed that ANN performed much better than BDNN. With fine tuning of hyperparameters, ANN reported a testing accuracy of 67.58%. BDNN's testing accuracy was reported to be

15% after thorough experimentation. An alternate solution however raised it to 24.69%. The limitation of results motivated to explore more ways to assess if emotions can be classified bidirectionally.

Bidirectional Long Short-Term Memory (LSTM) models perform quite well on classification problems on sequential data and process both backward and forward information of the sequence at every step. The data being used in this experiment is not sequential. However, Ba et al [7] and Gregor et al [8] have shown that sequential processing can be done in the absence of sequence. Esteban et al. combined static clinical information (e.g. patients demographic information, blood type, etc) and sequential information obtained from multiple medical tests and hospital visits to predict the next event, e.g. death of patient, rejection of organ, loss of organ, using Recurrent Neural Networks (RNN) [19]. Chopra et al. performed a similar task with non-sequential data to predict hospital readmission of diabetic patients using Recurrent Neural Networks and compared its accuracy with basic classifiers like Simple Vector Machine, Random Forest, and Simple Neural Network. Their experiment showed that RNN had the highest prediction power among all the methods [20]. This implies that RNN offers a lot of flexibility. Apart from analyzing facial expression, we are also interested in determining the extent of efficiency of Bidirectional LSTM when applied on fixed input or non-sequential data like SFEW dataset. LSTMs are very powerful models which can learn very quickly. There are several problems which are solved by applying machine learning on static data or a combination of sequential and non-sequential data. Data obtained from hospitals and immigration offices is one such example. Analyzing the limitations of applying LSTM on non-sequential data will help understand the challenges and identify future work that needs to be done to allow better modes and methods to merge data for machine learning and this is a big motivation behind using LSTM for this task.

The following sections will provide a brief discussion about related work that has been done in the field. It will also explain the experiments performed previously with ANN and BDNN, and recently with Bidirectional LSTM. The rest of the paper will evaluate the results and compare with related work. It will be followed by identification of possible issues, challenges in the work, conclusion, and recommendation for future work.

1.1 Related work

Automated Facial Expression Recognition (FER) has been an active area of research and a challenge for researchers. Many approaches have been adopted based on the engineered features like PHOG, LPQ, Gabor, etc., models are then developed and fine tuned to recognize the emotions with the best possible accuracy. Mollahosseini et al. implemented a network consisting of two convolutional layers each followed by max pooling and then four Inception layers. They tested their model on seven publicly available facial expression databases and found their results to be comparable or better than the traditional Convolutional Neural Networks (CNN) [9]. Sun et. al trained a linear Support Vector Machine (SVM) and partial least square classifiers for features in SFEW dataset. They proposed a fusion network to combine all the extracted features at decision level and gained an accuracy of 56.23% on SFEW which was much higher than the baseline recognition rate of 35.96% [10]. Deep neural networks have been greatly used to solve the problem of facial expression recognition. Following an initial framework of CNN by LeCunn in 1990s for handwritten digits [14], Krizhevsky et al. proposed Alexnet [15], Simonyam et al. proposed VGGNet [16], and Szegedy et al introduced an Inception module which could analyze images with multiple convolutional filters parallelly [17]. Recently a lot of hybrid models are being experimented which combine existing ones with new techniques. Emotion recognition in the wild (EmotiW) is a leading competition for facial expression recognition from videos [18]. Commencing in 2016, the participants have since then proposed a lot of models considering the data types and their work has added a lot of value to this research area.

RNNs are a type of Neural Network where the hidden state of one time step is computed by combining the current input with the hidden state of the previous time step. However, it is not possible for standard RNNs to capture long-term dependencies from very far in the past due to the vanishing gradient problem. The vanishing gradient problem means that, as we propagate the error through the network to earlier time steps, the gradient of such error with respect to the weights of the network will exponentially decay with the depth of the network. To counter this problem, LSTM was developed with a gating mechanism that dictates when and how the hidden state of an RNN has to be updated. [19]. Bidirectional LSTMs train two, instead of one, LSTMs on the input. Bidirectional LSTMs are an extension of traditional LSTM with improved performance.

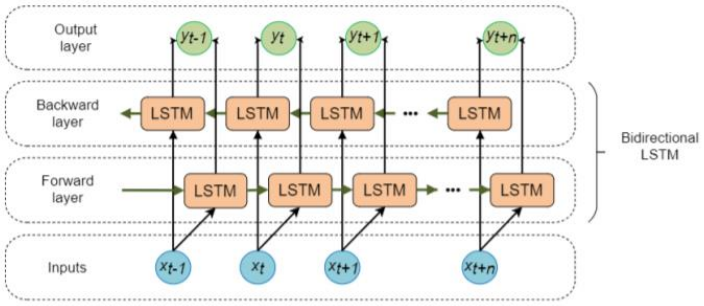

Fig. 1. An illustration of Bidirectional LSTM [21]

2 Database Details

Availability of realistic face data is highly crucial when it comes to analyzing facial expression. This paper makes use of *Static Facial Expressions in the Wild* (SFEW) database. Several analysis methods and descriptors are being used in this field to assess their efficiency and effectiveness in assessing the emotion. SFEW is a static dataset, which captures facial expressions in tough conditions [1]. The original data consists of 700 images which are labelled for six basic expressions and a neutral class. However, for this paper the data being used consists of 675 images. Apart from image files, statistical features called pyramid of histogram of gradients (PHOG) and local phase quantization (LPQ) had also been extracted from the images; this dataset was contained in a Microsoft Excel file and LSTM model was applied on it. Applying LSTM on statistical data allows us to assess the efficiency of model which has been trained using fixed input vector to process the data sequentially. Moreover, there can be several kinds of limitations that a process faces, including limitations in performance and limitations in processing resources or available data. Using statistical data instead of the visual data enables to determine the effect on performance when working with limited computational resources.

The classes in the dataset are balanced, with 6 labels having 100 observations each, and only label 2 – Disgust having 75 observations. The simplified version of SFEW data contains 5 columns of PHOG, 5 columns of LPQ features, and 1 column containing label. There is another column with image titles, which was not used in the analysis. The label consists of 7 values from 1 till 7, each depicting an emotion such as 1 – Angry, 2 – Disgust, 3 – Fear, 4 – Happy, 5 – Neutral, 6 – Sad, and 7 – Surprise. All 675 observations were used, out of which 540 (80%) were used in training and the rest were set apart for testing purpose. The data was magnified for better learning; the details will be discussed in later sections.

2.1 Database Preprocessing

The database didn't require much pre-processing, though one row with NAN values for PHOG was imputed with the group mean of PHOG corresponding to the specific label. The dataset does not have missing values, and the values of all features are centered around zero. The column with image names has 674 levels instead of 675. Upon examination of data, it was revealed that image titled "HarryPotter_Deathly_Hallows_1_001610960_00000028.mat" appears twice in the dataset with exactly same LPQ and PHOG features. However, both observations are labelled differently, with label 3 assigned for the first occurrence and label 7 for the second. Taking group mean of every feature with respect to label 3 and label 7 did not help in deciding which label does the image belong to. It was assumed that this image was possibly close to decision boundary of label 3 and 7 which caused the confusion. Since LSTM is sensitive to the scale of the input data, the data was normalized before being passed through the model. StandardScaler() method showed a higher testing accuracy than MinMaxScaler().

2.2 Database Distribution

LPQ2 till LPQ4 and all PHOG features follow the same distribution, but LPQ1 has a different cumulative distribution due to a few values which need to be investigated. Distribution of LPQ1 by label show at least two values which seem to outliers.

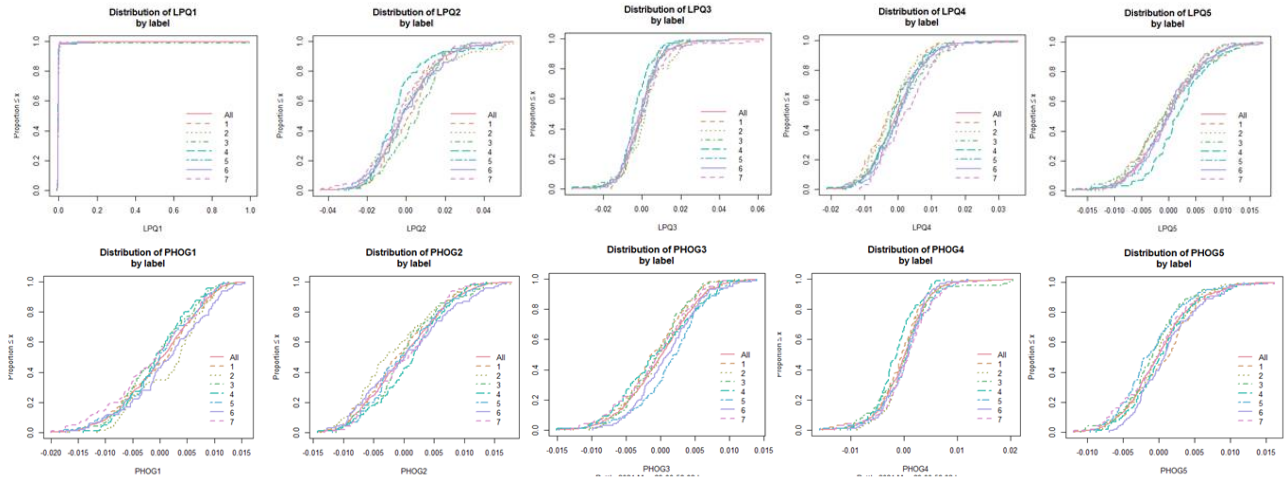

Fig. 2. Images showing distribution of LPQ and PHOG features

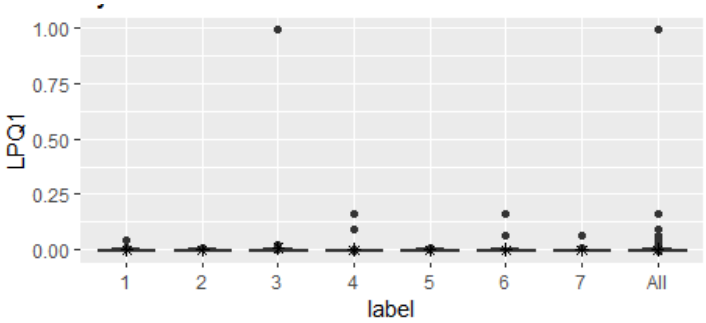

Fig. 3. Box plot of LPQ1 indicates potential outliers

3 Training different types of models to analyze static facial expressions

3.1 Artificial Neural Network (ANN)

Different settings, approaches, and values for hyperparameters were used to improve accuracy of ANN using Pytorch. ANN was tested with using either LPQ or POHG values, or both which corresponds to 5, 5, or 10 input neurons, respectively. Similarly, it was also trialed for single, double, and triple hidden layers with varying number of hidden neurons. The model worked the best with taking all 10 input neurons and using two hidden layers with 35 and 20 hidden neurons, respectively. A single hidden layer of 50-100 neurons signaled overfitting as testing accuracy achieved 100% even with half the number of epochs. In that case dropout was used to avoid overfitting. However, as multiple hidden layers showed better result, dropout was not needed. 700 number of epochs with a batch size of 150 showed better result; other experimented values e.g., batch size of 25, 50, 100 with 500 or 1000 number of epochs resulted either in oscillating accuracy or in a negligible effect on performance at the expense of run-time speed. Learning rate was decided to be 0.01 because a lower learning rate (0.001) gave a steeper training accuracy and low testing accuracy curve.

With the above-mentioned settings of hyperparameters, the model's testing accuracy ranged between 22%-28% which was very low. The accuracy did not show any change with varying the model parameters while keeping the same hyperparameters. However, magnifying the data by repeating the splitting process intro training and testing batches k-times increased the testing accuracy almost three times. It can be inferred that since it is a relatively small dataset with 675 observation, sampling the same data k-times improved the learning process. It was observed that testing accuracy improved from 54% to 69.84% as k increased from 5 to 10. However, it did not show a significant change when k was set to 15. As increasing k implies increasing number of epochs which results in slower performance, k value of 10 was chosen to be the optimal value.

It was also observed that with splitting the data k=10 times, using only LPQ values (5 inputs) gave a lesser accuracy (42.44%) as compared to the time when only PHOG values were used (60.43%). In case of SFEW, it was determined that PHOG was a better predictor of emotion as compared to LPQ. The best testing accuracy was obtained using all 10 input neurons (72.73%).

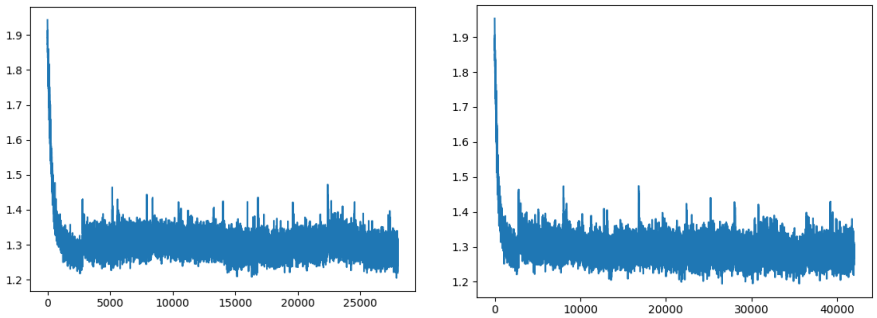

Fig. 4. Loss with k=10 (left) and k=15 (right)

Similar trials were done for model parameters. Different activation functions like Relu, Sigmoid, Softmax, and Tanh were used. It was observed that using Tanh on the hidden layers and Sigmoid on the output layer gave better results. This is because Tanh centres around 0 instead of Sigmoid's 0.5 and this makes learning easier and better for the next

layer. Convergence is faster when average of each input layer centres around zero. However, if all input vectors have the same sign the weights will increase or decrease together for a given input pattern and it results in a very inefficient and oscillating performance. This is the reason we chose to normalize the inputs before passing it on to the feedforward network. Various optimizers like SGD, Momentum, and Adam were used. Adam as an optimizer and Cross Entropy Loss as the loss function turned out to be the most appropriate choices.

To improve the performance of the neural network, heuristic approach was used and various measures were taken like increasing hidden layers, changing activation functions, changing activation function in the output layer, increasing the number of neurons, increasing the number of folds in data, normalizing the data and changing learning algorithm parameter.

3.2 Bidirectional Neural Network (BDNN)

Error-back propagation technique was applied in forward and reverse direction to adjust the weight matrix of our model. Training started on SFEW dataset using the same parameters as ANN and a testing accuracy of 12% - 15% was attained. There is no rule of thumb to select the parameters and the selection is often made on the basis of performance. Different parameters were tested e.g. learning rate reduced to 0.001, batch_size increased to 200, hidden layers reduced to one layer, and no. of neurons increased and decreased, but all these changes yielded lower accuracy, the lowest one recorded to be 1.02%. Just like ANN, after trying different Activation Functions and Optimizers, the dataset was repeatedly sampled for splitting between testing and training so as to imrpove learning. BDNN's accuracy plummeted to 7.02% with repeated sampling in a significantly slow process. Therefore, data was split only once in traning and testing data.

In the Forward pass, Tanh showed the best results with activation. After Forward pass, One-hot encoding was used to prepare tensor for Cross Entropy Loss in reverse pass. This was required because the prediction is being done for mutually-exclusive outcomes, and in such cases the prediction vector is supposed to have non-negative elements and the sum of the elements must be equal to 1. Since Softmax has asymptotes at 0 and 1, it works better with one-hot encoding. In the reverse pass, softmax improved the loss but yielded a low accuracy of 3%. After several trials and experimentation, it was established that our BDNN model does not analyse facial expression well.

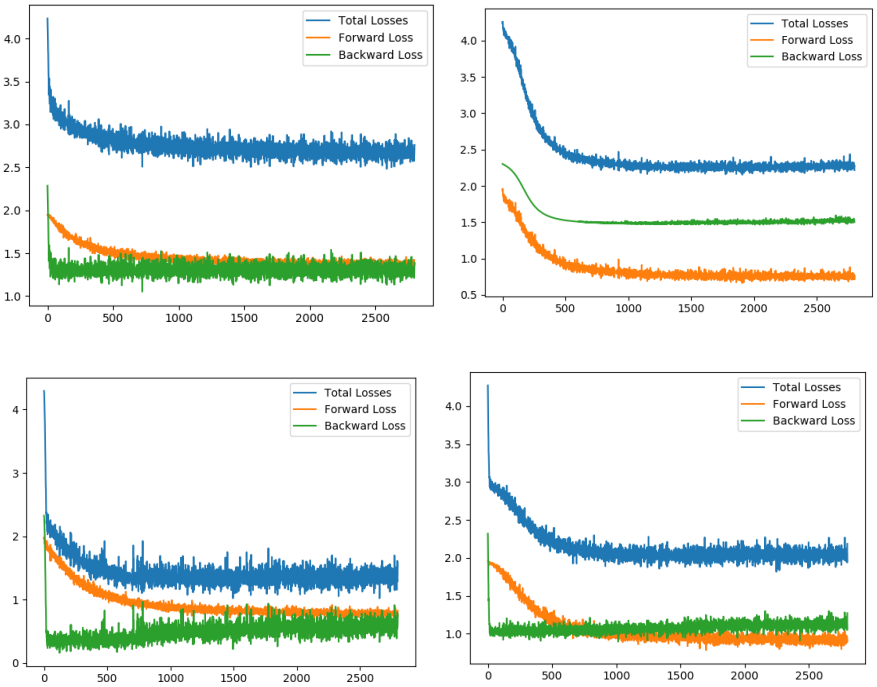

Fig. 5. Loss curve using Relu (top left), Softmax (top right) and Loss curve using Tanh (bottom)

An alternate implementation of BDNN using an extra node was also used to classify bidirectionally. Using two models in every epoch, one model was used for reverse assignment of weight of backward model and the other model was used for reverse assignment of weight of forward model. The forward and backward model were trained this way. This approach showed better results as compared to the other approach of BDNN. However, average testing accuracy of 24.69% does not qualify it for a reasonable performance.

3.3 Bidirectional Long Short-term Memory Model (LSTM)

A Bidirectional LSTM was trained on SFEW data duplicating the first recurrent layer in the network. In this way there were two layers side by side. Then input was provided to the first layer and a reversed copy of that input was fed to the second. Taking an input of 10 neurons, and using 3 layers with a learning rate of 0.01, the model was trained over 300 epochs using a batch size of 150. In this process 80 hidden neurons were used. These hyperparameters were selected heuristically after trying several other combinations. Similar to ANN, the data was magnified *k* times. It was seen that with data splitting was done iteratively thrice i.e. k=3, testing accuracy rose from 40% to more than 80% in some cases. However, there was a lot of variation in the results. The model was found to be prone to overfitting; therefore, Dropout of 5% was added to the network. For activation function, Sigmoid and Hyperbolic Tangent returned almost the same results. tanh() was used as activation of cell states and output node. Two tanh activations in LSTMS blocks squash the block input and output and therefore they had a significant impact on the overall performance. After thorough testing, tanh() was removed, as it blocked the testing accuracy around 45% and the model performed better without it. Adam as an optimizer and Cross Entropy Loss as the loss function turned out to be the most appropriate choices.

It was observed that accuracy took a dip when data was normalized using StandardScaler(). MinMaxScaler() provided much better results. This further demonstrates that LSTM is sensitive to scaling of input data. The recurrent network aggregates information extracted from the individual glimpses and combines the information in a coherent manner. We used Long-Short-Term Memory because of its ability to learn long-range dependencies and stable learning dynamics. The use of Bidirectional LSTM may not make sense for all sequence prediction problems, but it surely offers benefits in terms of better results. Because the model works bidirectionally, the model learns the problem faster which is apparent from training of the model over time.

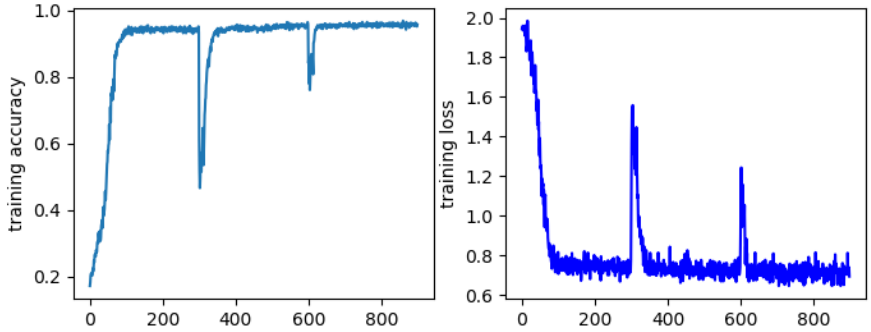

Fig. 6. Training loss curve with k=3. The dip in the curve marks the next training iteration. It can be seen that training improves with every iteration

4 Comparison

4.1 Comparison between the models

Bidirectional LSTM outperformed BDNN due to its ability to learn long range dependencies and to duplicate processing chains to enable the input to be processed in forward and reverse direction. ANN performed better as compared to BDNN. Both models had almost the same hyperparameters but different model parameters. However, for better learning ANN and Bidirectional LSTM magnified input data, as data was split between training and testing data *k* times. In case of BDNN, accuracy went downhill and made the model much slower. Due to reverse pass, activation functions were changed. In both cases, forward loss showed a similar trend; ANN performed accurately around 67% which is a decent figure but can be improved. To have a better learning experience in reverse forward pass it will be helpful to have a high accuracy in the forward pass. Fu achieved accuracy of 20.01% using both features without magnifying the input and assumed that the Principal Component Analysis does not keep all information from raw input, resulting in the loss of accuracy [5].

Table 1. Testing accuracy statistics of all models

Model	Hyperparameters	Testing accuracy
ANN	input neurons = 10 hidden neurons = [35, 20] epochs = 700 batch size = 150 learning rate = 0.01 k = 10	Average: 67.58% Testing accuracies of 5 random executions: [66.18%, 65.47%, 68.97%, 69.84%, 67.43%]
BDNN	input neurons = 10 hidden neurons = [35, 20] epochs = 700 batch size = 150 learning rate = 0.01	Average: 15% (BDNN Alternate solution)Average: 24.69% Testing accuracies of 5 random executions: [23.40%, 27.58%, 25.19%, 24.66%, 22.60%]
Bi-LSTM	input neurons = 10 hidden neurons = 80 number of layers = 3 epochs = 300 batch size = 150 learning rate = 0.01 k = 3	Average: 45.42% Testing accuracies of 10 random executions using tanh as activation function: [43.15%, 47.26%, 44.52%, 52.38%, 47.18%, 33.77%, 44.22%, 44.27%, 39.31%, 58.16%] Average: 56.90% Testing accuracies of 15 random execution without tanh: [81.30%, 48.97%, 64.55%, 37.40%, 54.89%, 49.32%, 45.89%, 52.52%, 65.25%, 66.90%, 68.38%, 49.02%, 51.66%, 52.03%, 65.47%]

4.2 Comparison with the previous work

Our experiments showed that while emotions could be classified using the input features of LPQ and PHOG values, the same was not accomplished as much accurately the other way round in case of BDNN. Dhall et. al performed Static Facial Feature Analysis on several datasets and concluded that LPQ and PHOG are not suitable measures to analyze facial features in an uncontrolled environment. In case of the stated paper, LPQ gave a classification accuracy of 53.07% and PHOG gave it around 57.18% [1]. This is very much comparable to our results for ANN where LPQ values gave a lesser accuracy (42.44%) as compared to PHOG (60.43%). While in case of BDNN our testing accuracy remained quite low, Bidirectional LSTM performed comparatively better with average testing accuracy of 56.90% in the absence of tanh as activation function. However, there was a large variation in the results and accuracy was evern reported as high as 81.30%. This, and the results from other datasets in Dhall et al.'s experiments also show that LPQ and PHOG have a significantly lower accuracy for SFEW. This is due to the close to real world conditions in the SFEW database. SFEW contains both high and very low-resolution faces, which adds to the complexity of the problem [1]. Furthermore, LPQ and PHOG along with other similar measures have been developed and previously experimented on lab-controlled data whereas the provided SFEW dataset comprises of images which have been extracted from movies and are hence very close to reality. The loss in accuracy can also be because the baseline paper used both cross validation and SPI protocol and implemented a support vector machine algorithm.

A similar task was successfully done by Nejad et. al for various cases in which the function is invertible. However, their work proposes not using Bidirectional neural network for classes of problems which are inherently not invertible [2]. Instead, they experimented BDNN with some changes like adding an extra node in the output in order to make the function invertible. Our alternate implementation of BDNN incorporated this technique but still could not attain a reasonable accuracy. However, Nejad et. al also experimented with the sequence of training direction. They maintained a direction of training of network until overall normalized error got lower than the error of previous direction of training. An alternate way is to train network in one direction for a maximum number of epochs and then switch the direction. We chose to experiment with the later method and did not see a significant improvement in the performance. On the contrary, Nejad et. al were able to achieve an accuracy of 74% when they implemented BDNN for predicting student final marks, and character recognition.

In case of Bidirectional LSTM, the results were comparable to the baseline model and previous work. This suggests that LSTM was able to process the data sequentially as the data was fixed. As mentioned earlier, the reason for using this model was to capture long-term dependencies. The ability to remember combination of features which correspond to a specific emotion can be useful in analysing future observations. However, as RNN is not capable of capturing long-term dependencies due to vanishing gradient problem, LSTM was used. In comparison with RNN, LSTM use more network parameters and are hence, computationally expensive. The experiment was repeated 10 times with and 15

times without using tanh as activation function. It was observed that using tanh as activation function produced more consistent but slightly lower accuracy; average around 45.42% but the results showed a lot of variation without the activation function. It was also observed that LSTM is very sensitive to scaling of data as MaxMinScaler() produced a lot more accurate results as compared to StandardScaler().

Some improvement may be seen if other techniques and suggestions by Nejad et. al are also implemented e.g., if training data is improved by removing outliers, and using alternate measures for static facial expressions. Training data can also be manipulated for statistical calculation like using measures of central tendency for a specific feature, say LPQ1, for an emotion, say, anger will indicate a typical value in the distribution, and it can be used to extract rules or derive equations. These values can also be used to manipulate the weight matrix and help us establish which feature plays a better role in prediction.

4.3 Discussion and Limitations

There are several explanations for the effective results produced by LSTM on a non-sequential data. One of the biggest drawbacks of LSTMs is that it cannot handle an infinitely long sequence. SFEW dataset did not offer such challenge to LSTM which would have compromised its working. Karpathy in his article "The Unreasonable Effectiveness of Recurrent Neural Networks" stresses that even if the data does not form a sequence, the model can still formulate and train powerful models that learn to process it sequentially. The model learns stateful programs that process the fixed size data [22]. It has been established that LSTMs memorize sequences extremely well but they do not necessarily generalize in the correct way. The generalizing ability of the model could not be verified in this experiment. There are seven publicly available datasets for facial expressions and replacing SFEW with them will demonstrate that. Another issue with LSTMs is the computational complexity that they have due to matrix multiplication and unnecessary duplication of representation size in each step.

There is also a possibility that since everything is working well in the model the recurrent paths and state variables are being ignored in the model, which, if true, indicates computational resources are not being used to the optimum level. In other words, this can be considered as a random noise which neither supported nor affected the performance. If over long series of training data, the input does not correlate to previous inputs or outputs, or the error propagating backwards, eventually weight decay will take place and gradually reduce to zero. Hence, input be ignored. If the training observations are independent of one another or they follow a subtle pattern, the RNN will quickly learn that and produce the output according to the sequence being followed as in the previous input cases. SFEW dataset apparently does not have any such sequence but exploring the data further may reveal more links to working of LSTM and limitations associated with it. Since the main goal was to analyze facial expressions bidirectionally, with average testing accuracy exceeding 55%, it is safe to say that static facial expressions can be analyzed with a reasonable accuracy and future work in this area will focus on improving the results. Another goal was to assess the limitations of using LSTM on non-sequential data. It has been observed that the model fits the data well and performs better than a simple BDNN. However, the chances of computational resources being over-burdened in this procedure cannot be ruled out.

5 Conclusion and further work

The analysis of the human face characteristics and the recognition of its emotional state are very challenging and difficult tasks. The main difficulty comes from the non-uniform nature of the human face and various limitations such as lightening, shadows, facial pose and orientation conditions [3]. The interaction between human beings and computers will be more natural if computers are able to perceive and respond to human non-verbal communication such as emotions [4]. Although a lot of work has been done to predict emotions using facial features, and an equivalent amount of work is being done in defining features which are a better predictor for the task, there is still a lot to be done to explore how to make it work in the reverse direction.

In this paper we conducted experiment on SFEW dataset applying bidirectionality on classification by training the networks in parallel simultaneously. The task was performed first using an Artificial Neural Network and then using Bidirectional Neural Network and Bidirectional Long Short-term Memory models. The results were then compared with each other and with previous work done using the same dataset and implementing the same technique. There are some limitations in implementations and further experimentation can yield better results. Extracting rules from data or using statistical manipulations may improve the learning process. Training and testing the model using some other features may show slightly different results. Training data can also be improved by removing outliers. Experimentation with the sequence of training direction can also be done in case of BDNN. Liu's work using SFEW dataset proposed to combine BDNN with facial emotion recognition using a Hierarchical Convoluted Neural Network (CNN) [6] which indicates that fusing another technique with BDNN can yield better results in analysis of static facial expressions. Further work in combining different feature extracts with advanced CNN and BDNN may provide better performance.

Using visual data, Deep Convolutional LSTM Networks will enable processing spatial and temporal information. In this scenario, using convolutional architectures like ResNet with Convolutional LSTMs or Transfer Learning will prove to be powerful tools. Training and testing the model on similar datasets, i.e. using other six publicly available facial expression databases i.e. viz. MultiPIE, MMI, CK+, DISFA, FERA, and FER2013, will improve generalization ability and enable the network to tackle the problem well.

6 References

1. Dhall, A., Goecke, R., Lucey, S., & Gedeon, T. (2011, November). Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)* (pp. 2106-2112). IEEE.
2. Nejad, A. F., & Gedeon, T. D. (1995). Bidirectional neural networks and class prototypes. In *Proceedings of ICNN'95-International Conference on Neural Networks* (Vol. 3, pp. 1322-1327). IEEE.
3. Koutlas, A., Fotiadis, D.I.: An automatic region based methodology for facial expression recognition. In: *IEEE International Conference on Systems, Man and Cybernetics, SMC* 2008, pp. 662–666 (2008).
4. Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C. M., Kazemzadeh, A., ... & Narayanan, S. (2004, October). Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th international conference on Multimodal interfaces* (pp. 205-211).
5. Fu, C. Emotion Classification and Input Encoding Technique Analysis using SFEW Dataset.
6. Jiaxu, L. I. U. Facial Emotion Classification in SFEW Database Based on Hierarchical CNN with Bidirectionality and Data Augmentation.
7. Ba, J., Mnih, V., & Kavukcuoglu, K. (2014). Multiple object recognition with visual attention. arXiv preprint arXiv:1412.7755.
8. Gregor, K., Danihelka, I., Graves, A., Rezende, D., & Wierstra, D. (2015, June). Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning* (pp. 1462-1471). PMLR
9. Mollahosseini, A., Chan, D., & Mahoor, M. H. (2016, March). Going deeper in facial expression recognition using deep neural networks. In 2016 *IEEE Winter conference on applications of computer vision (WACV)* (pp. 1-10). IEEE.
10. Sun, B., Li, L., Zhou, G., & He, J. (2016). Facial expression recognition in the wild based on multimodal texture features. *Journal of Electronic Imaging*, 25(6), 061407.
11. Guo, Y., Tao, D., Yu, J., Xiong, H., Li, Y., & Tao, D. (2016, July). Deep neural networks with relativity learning for facial expression recognition. In *2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (pp. 1-6). IEEE.
12. Acharya, D., Huang, Z., Pani Paudel, D., & Van Gool, L. (2018). Covariance pooling for facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 367-374).
13. Zheng, H., Wang, R., Ji, W., Zong, M., Wong, W. K., Lai, Z., & Lv, H. (2020). Discriminative deep multi-task learning for facial expression recognition. *Information Sciences*, *533*, 60-71.
14. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.
15. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, 1097-1105.
16. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
17. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
18. Dhall, A., Goecke, R., Joshi, J., Hoey, J., & Gedeon, T. (2016, October). Emotiw 2016: Video and group-level emotion recognition challenges. In *Proceedings of the 18th ACM international conference on multimodal interaction* (pp. 427-432)
19. Esteban, C., Staeck, O., Baier, S., Yang, Y., & Tresp, V. (2016, October). Predicting clinical events by combining static and dynamic information using recurrent neural networks. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)* (pp. 93-101). IEEE
20. Chopra, C., Sinha, S., Jaroli, S., Shukla, A., & Maheshwari, S. (2017, October). Recurrent neural networks with non-sequential data to predict hospital readmission of diabetic patients. In *Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics* (pp. 18-23).
21. Kulevome, D. K., Wang, H., & Wang, X. (2021). A Bidirectional LSTM-Based Prognostication of Electrolytic Capacitor. *Progress In Electromagnetics Research C*, *109*, 139-152
22. http://karpathy.github.io/2015/05/21/rnn-effectiveness/