University of Ibadan
Journal of Science and Logics in ICT Research

A Model for Identifying Deceptive Acts from Non-Verbal Cues In Visual Video Using Bidirectional Long Short Term Memory (BiLSTM) With Convolutional Neural Network (CNN) Features

[1] Ajibade, F. D. and [2]Akinola, S. O.

University of Ibadan
[1] faithajibade20@gmail.com, [2] akinola.olalekan@dlc.ui.edu.ng

Abstract

Automatic identification of deception is crucial in so many areas like security, police investigations, court trials, political debates, relationships, and workplace and so on. Techniques for deception detection range from detecting deception through verbal, nonverbal and vocal clues. Existing works have been thoroughly dependent on combining different modalities of videos like audio and text for identifying deceptive behaviours These approaches have improved the overall accuracy of deception detection systems but there are exceptions where videos do not have accompanying audio and text. The aim of this research is to develop a model that can identify deceptive behaviours through non-verbal cues gotten from the visual modality of videos. The Real Life Deception Dataset created by Perez *et al.* (2015) was used for the purpose of this research. It contained labelled videos of deceptive and truthful court cases. Image frames were extracted from each video and pre-processed. A Convolutional Neural Network (CNN) was used to learn the different behavioural gestures and cues exhibited in these image frames before passing these learned features to the Bidirectional Long Short Term Memory (BiLSTM) algorithm which then classifies as either deceptive or truthful. Training and testing was also done using BiLSTM and evaluated with existing works. The model performed well in identifying deception from visual videos using CNN features learned from image frames. It gave an accuracy of 61% with a loss of 0.2 after running for three epochs. Sourcing local data from surveillance cameras and security feeds can be further explored to validate this work.

Keywords: *Nonverbal cues, Visual videos, CNN, BiLSTM*

1. INTRODUCTION

Deception is a message knowingly transmitted by a sender to foster false belief or conclusion by the receiver. It is a deliberate attempt to mislead other people which can manifest in various forms like falsification, concealment and equivocation [1]. On the other hand, one can also be suspicious of being deceived. Believing that one is being deceived without proof or sufficient evidence to prove certainty is termed "Suspicion" [1]. This person then goes ahead to look for signs or clues that indicates if one is being deceived or not. Ekman and Friesen [2] stated that during communication, there are some unconscious and unintentional behaviours that the participants exhibit that can serve as clues to deception. These unconscious behaviours or nonverbal behaviours which they term leakages, usually reflect the perceptual, cognitive and emotional processes that accompany the way communicators encode and decode the messages in a communication.

Automatic identification of deception is crucial in so many areas like security, police investigations, court trials, relationships, and workplace and so on. But from research done so far [3], it has been suggested that machines have a better chance of identifying deception than humans.

Although, there is no universal standard or behaviours that

Ajibade, F. D. and Akinola, S. O. (2021). A Model For Identifying Deceptive Acts From Non-Verbal Cues In Visual Video Using Bidirectional Long Short Term Memory (BiLSTM) With Convolutional Neural Network (CNN) Features. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 7 No. 1, pp. 39 - 47

are indicators of deception as deception may vary from situation to situation. This is because the motive for deceiving others may vary depending on the deceiver's motive [2].

Existing research works have been thoroughly dependent on combining different modalities of videos like audio and text for identifying deceptive behaviours. This work is leveraged on videos that do not have accompanying audio and text (e.g. surveillance cameras) which are massive data thereby making sure they are still useful and can be sufficient to identify deceptive behaviours with improved accuracy.

The objectives of this research include:

i. To develop a deep CNN model to extract spatial features from video frames.
ii. To develop a BiLSTM model that will capture the temporal correlations in the sequence of the CNN extracted features to identify nonverbal cues gotten from these features.
iii. To test and evaluate with other existing models.

The rest of the paper is organized as follows: Section 2 discusses some of the literature reviewed relative to this work, Section 3 explains the methodology adopted to achieve our research objectives. In Section 4, the results obtained from this research are presented and a conclusion is drawn in Section 5.

2. Related Works

Jensen *et al.* [3] examined how humans express themselves and subtle behavioural patterns in order to develop an automated system for flagging hostile, deceptive or suspicious communications. They developed a taxonomy of fundamental meta-messages of interpersonal communication which was done using the Brunswikian lens model. This model maps behavioural indicators to communicative meanings that most time elude human observers. This model consisted of distal indicators or cues, a perceived judgment called proximal percept and subjective attributions or meanings. They applied this model to identify patterns of micro-level deception cues that predict mid-level percepts which in turn predict attributions. The deception cues were gotten through kinesics analysis, proximal percepts were derived from judgments made by third-party observers and attributions was gotten from the prediction of an individual's level of honesty using proximal percepts as predictors. Although factors such as motivation to succeed at deception, individual characteristics and so on were not considered when evaluating the honesty score of an individual which accounted for the low variance in their model.

Lu *et al.* [4] proposed a model for automatically extracting blobs from the hands and head which could be used to identify deceptive behaviours. This model called blob analysis analysed the movement of the head and hands based on the identification of its skin colour which was possible through validation with numerous skin tones. Their work was made possible by the work done by the Computational Biomedicine Imaging and Modelling Centre (CBIM) at Rutgers University which made it possible to track human body parts. They used colour analysis, eigenspace-based shape segmentation and Kalman filters to track the position, size, and angle of different body parts throughout a video segment. Blob analysis was able to extract hand and head features using a Look-Up-Table (LUT) with three colour components (red, green, and blue) that was created based on the colour distribution of the face and hands. It was built in advance of any analysis and was formed using skin colour sample. Although it is time consuming creating training skin samples and we can have issues like incorrectly identifying regions having the same colour as the skin colour. Blob analysis requires that the individual be positioned in front of the camera with the face clearly visible which is not always realistic.

Tsechpenakis *et al.* [5] proposed a Hidden Markov Model (HMM) which used visual cues that were extracted from videos using blob analysis to explore behavioural state identification in the detection of deception. They were mainly concerned with the detection of agitated and over-controlled behaviours. Their method involved using movement descriptors mined from blob analysis and other movements such as the positions, velocities and variances of the blobs being observed, to recognize illustrators and adaptors movements to recognize possible detection. These movements were used as indicators of agitated or over-controlled behaviours. They implemented this method using a two-layered

hierarchical HMM model and they tested this model on 9 videos of real interviews which gave them an accuracy of 87.5%. Their model performed less when tested with real data than testing with acted interviews. More work still needs to be done in identifying deceptive acts in real life scenarios.

Pérez-Rosas *et al.* [6] addressed the problem that had persisted in research studies by introducing a novel multimodal dataset which was a database consisting of real-life court trial videos. Since court trials is conducted in a high-stake situation, there is always the likelihood that deception will occur which has made this dataset a big milestone in the deception research study. For their data collection, three different trial outcomes were used to correctly label a certain trial video clip as deceptive or truthful which are: guilty verdict, non-guilty verdict, and exoneration.

For guilty verdicts, deceptive clips were collected from a defendant in a trial, and truthful videos were collected from witnesses in the same trial. In some cases, deceptive videos were collected from a suspect denying a crime he committed and truthful clips were taken from the same suspect when answering questions concerning some facts that were verified by the police as truthful. Features extracted from their data include: unigrams, bigrams, facial displays (which were manually annotated using the MUMIN coding scheme) and hand gestures. They tested their dataset using Decision Tree and Random Forest classifiers whereby Decision Tree gave the best result of 75.20% when all the features were used and Random Forest gave the best result 73.55% on classifier trained with just nonverbal features.

They also presented a human deception detection study where they evaluated the human capability of detecting deception in trial hearings. Their system outperformed the human capability of identifying deceit by a range of 16% which indicates that detecting deception is a difficult task for humans and further verified previous findings where human ability to spot liars were found to be slightly better than chance [7].

Wu *et al.* [8] combined audio, visual and text modalities to develop an automated deception detection system. For the visual modality, they used classifiers trained on low level video features to recognize micro expressions and fused the score with Improved Dense Trajectory features to improve performance. For the audio modality, Mel-frequency Cepstral Coefficients (MFCC) features were extracted from the audio domain which also provided a significant boost in performance but they discovered that information from transcripts did not contribute much to the performance of the system.

Four different binary classifiers were trained for visual, audio and text respectively, while the fourth classifier used the pooled scores gotten from the micro expression detectors. They tested their models using the dataset proposed by Pérez-Rosas *et al.* [6], which is a database consisting of 120 court trial videos. They used a subset of 104 videos consisting of 54 deceptive and 50 truthful videos. Their micro expression detectors had a performance of 65.11% which they said could be improved by using deep learning. Individual classifiers had accuracy of 77%, 75%, 64% and 76% respectively. They combined the scores of their different classifiers using late fusion which gave them an accuracy of 83.47% which led them to conclude that combining different modalities improve accuracy.

Avola *et al.* [9] in 2019 presented a paper on detecting deception by extraction of facial action units from video frames of a subject in question. These facial action unit features were then classified as either deceptive or truthful using an SVM classifier. Features like facial landmarks, head pose estimation and eye gaze estimation were also used. They tested their method using the dataset created by Pérez-Rosas *et al.* [6] but pruned it down by discarding videos where the face of the subject is covered, hidden or difficult to detect or recognize and also videos which had more than one subject. Videos which had more than one person was cut to include just one person as the action unit extractor they used could only work with one subject. Videos in which there was the voice of the interviewed subject, but his interlocutor was shown instead, were also discarded. Openface toolkit [10] was used for

their action unit extraction. They got the best accuracy when SVM was used with a radial basis function kernel (RBF) with an accuracy score of 76.84%. They plan on improving their proposed method by integrating other heterogeneous features, such as speech and intonation, to make their method even more robust.

3. Methodology

As shown in Figure 3, the Bi-LSTM model has two layers for each time step which is the forward and backward layer. The input to this layer is our flattened feature vectors gotten from the CNN layer. The spatial features gotten from the CNN is fed into the Bi-LSTM layer to capture temporal relations between video frames. This is done with the aid of a sigmoid layer which produces the log probabilities for each output label. The behaviour class as seen in the figure represents the output label of either truthful or deceptive.

3.1 Dataset

Real Life Deception Detection dataset which was sourced from Pérez-Rosas *et al.* [6] from the University of Michigan was used for the purpose of this research. The dataset was downloaded with permission from the authors from the following website: http://lit.eecs.umich.edu/downloads.html#Real-life%20Deception. The motivation for creating this dataset by these authors was to build a dataset that could provide real life data that portrays situations where deceptive motives are real. This dataset consists of 121 court trial videos divided into 61 deceptive videos and 60 truthful videos.

3.2 Data Preprocessing

Before the classification stage of our model, there are processes involved:

3.2.1 Video conversion to frames

To convert each video to their corresponding image frames, the OpenCV library in python was utilized. This library is very useful for images and video processing. First, the function cv2.VideoCapture( ) was used to get the path to where the videos are stored. Second, cap.get (cv2.CAP_PROP_FPS) was used to get the frame per second (fps) rate which is a property of the video. The video is converted into frames using the rate of the frame per second.

3.2.2 Resizing

The image frames were resized using a function in the OpenCV library in python. The function cv2.resize() was used to resize each of the frames to 100x100 in terms of width and height to ensure uniformity across all the frames as the dimensionality varies for each video.

3.2.3 Grayscaling

The image was grayscaled using the function cv2.cvtColor (frame, cv2.COLOR_BGR2GRAY). This function takes the coloured image and converts them to a grayscale. The essence of Grayscaling is to reduce the complexity of the model as less information is to be processed for each pixel.

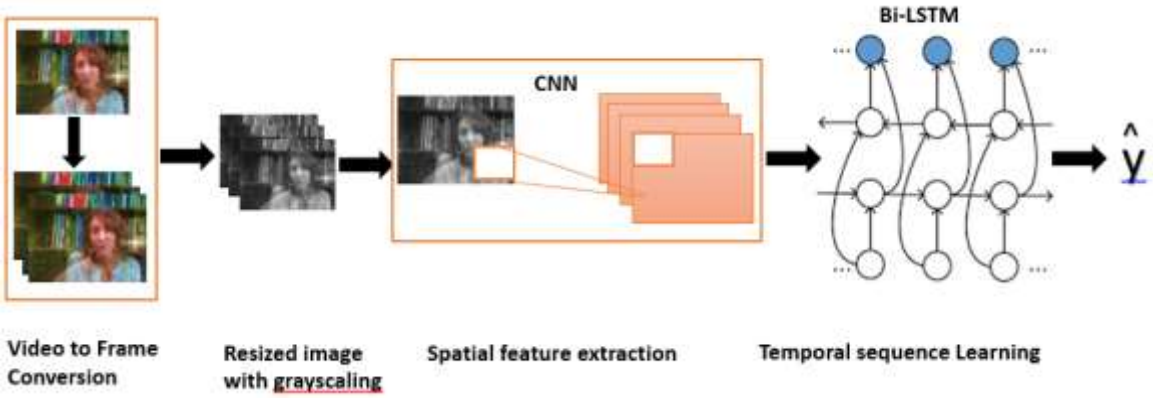

Figure 1: Model Overview

3.3 Spatial Feature Extraction

Convolutional Neural Network was introduced by Fukushima [11] and was improved by Lecun *et al.* [12]. From that time till now, CNN has broken a lot of grounds in image processing and is considered the state of the art for image classification [13]. Moreover, CNN have been regarded as a powerful model for solving visual recognition problems as it exhibits a high capacity of generalization and learning capability [13]. We employed a convolutional neural network model as illustrated in Figure 2 to extract spatial information from our image frames. The model used had 6 local connections, 4 pooling layers and a flatten layer.

Given an image frame containing n *pixels* $\{p_1, p_2, \ldots, p_n\}$, a convolution operation was applied to the surrounding pixels in a window to generate the feature map. Multiple kernels with different sizes were utilized to extract features of various granularities. Then max pooling was performed over each map so that only the largest number of each feature map was recorded. The property of pooling which produces a fixed size output vector allows us to apply variable kernel sizes. And by performing the max operation, the most salient information are kept. Finally, the fixed length output vector *cp, was taken* as a representation of the spatial feature of a frame.

This operation was performed consecutively until we got our final output vector, which was then flattened before being fed to the BiLSTM model as shown in Figure 2.

3.4 Temporal Feature Extraction

The founding idea of Bidirectional Long Short Term Memory is to present each training sequence forward and backward, both of which are connected to the same output layer. According to the keras website (https://keras.io/api/layers/recurrent_layers/bidirectional), the modes by which outputs of the forward and backward training sequence can be combined are one of the following: sum, mul, concat, ave, none. For every point in the training sequence, it has a complete and sequential information about all points before and after it.

As shown in Figure 3, the Bi-LSTM model has two layers for each time step which is the forward and backward layer. The input to this layer is our flattened feature vectors gotten from the CNN layer. The spatial features gotten from the CNN is fed into the Bi-LSTM layer to capture temporal relations between video frames. This is done with the aid of a sigmoid layer which produces the log probabilities for each output label. The behaviour class as seen in the figure represents the output label of either truthful or deceptive.

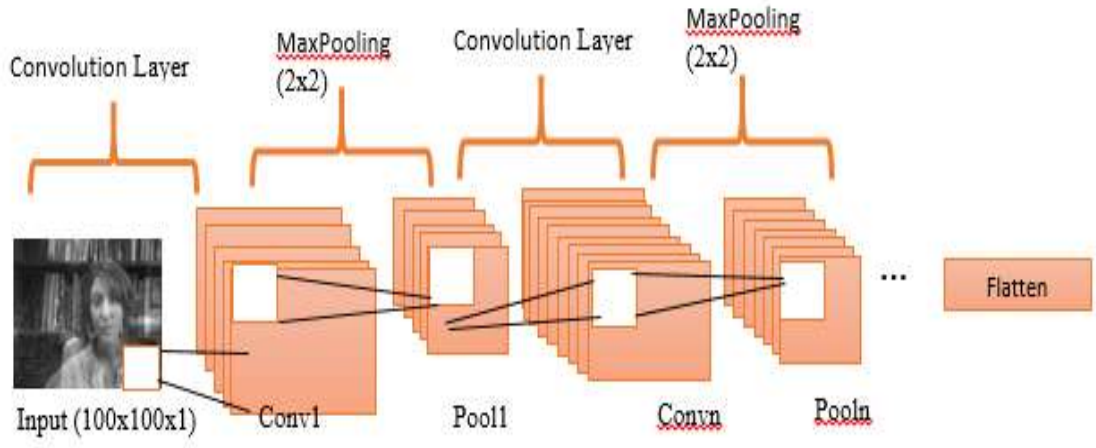

Figure 2: CNN Model

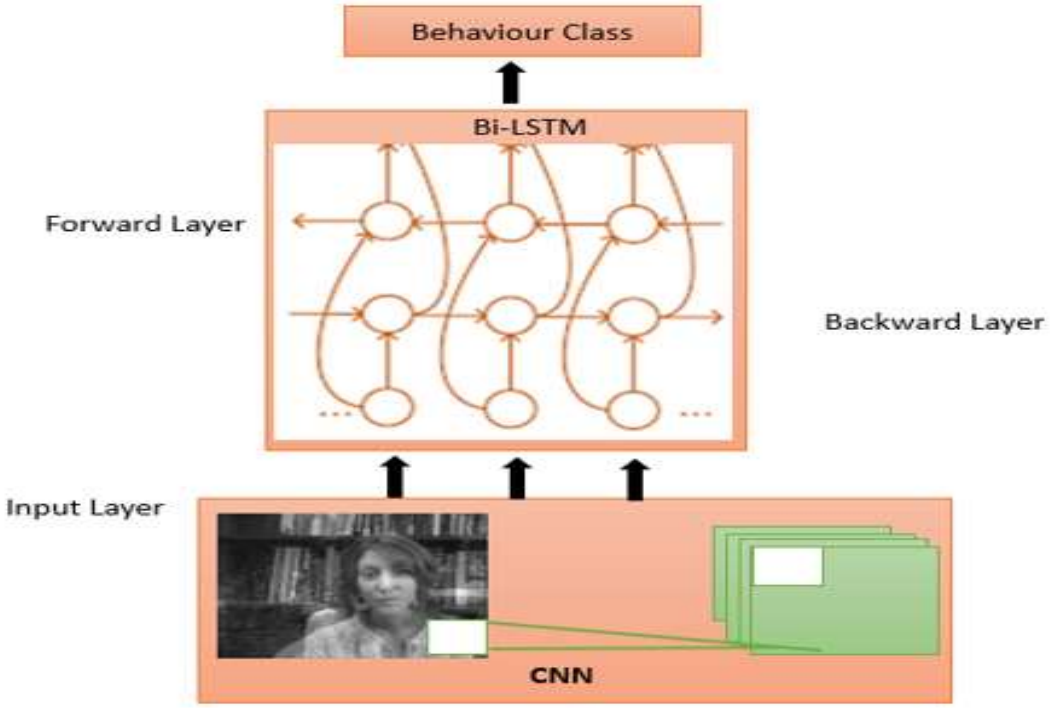

Figure 3: BiLSTM Model

3.5 Output Layer

The hidden state of the Bi-LSTM with spatial features extracted from the CNN model at each timestep was fed into a sigmoid layer to produce the log probabilities for each output label.

3.6 Performance Metrics

Accuracy of the prediction was used as our performance measuring metrics. The accuracy is the error rate between ground truth and predicted output i.e., how well the model was able to classify out-of-sample data

4. Implementation and Result

Implementation of this study was carried out using a number of python library tools, which include:

- OpenCV library: for image processing and manipulation. It is also for extracting 3D channels of the images.
- Keras: (with Tensorflow backend) for developing and evaluating deep learning models
- Pandas: To hold our Image frames in a table like structure called a Data Frame.
- Matplotlib: For visualizing our model

The hardware specification of the system used for implementing this work is a RAM size of 12GB, Hard Disk Drive of 107GB as provided by Google Colab.

4.1 CNN+BiLSTM Model

The CNN model was wrapped in a Time Distributed layer which applies a layer to every temporal slice of the input. The spatial features gotten from the model was passed into a bidirectional model for classification.

4.1.2 Training

The model was trained for 3 epochs and at the end of the training an accuracy of 94% with a loss of 0.25 was obtained. Figures 4 and 5 show the graph of both accuracy and loss.

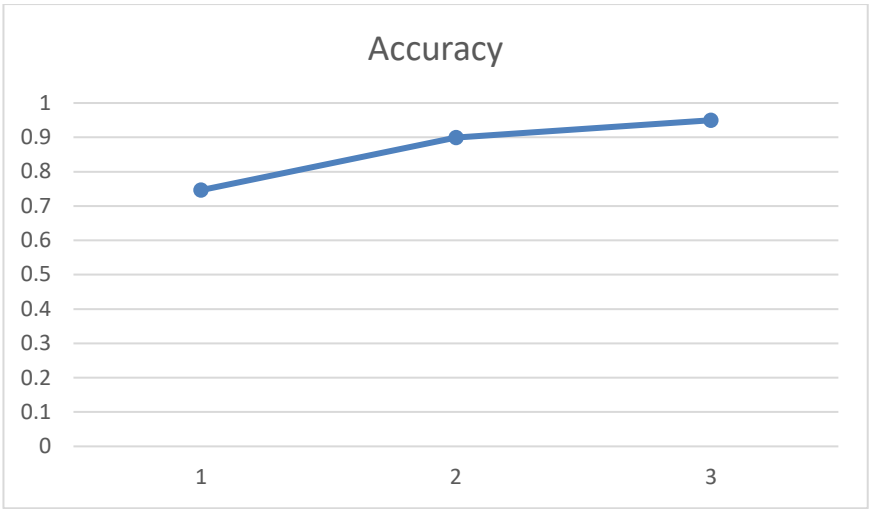

Figure 4: Training accuracy of the Model

Figure 5: Loss value of the Model

4.2 Model Evaluation

Our model was evaluated with the Real Life Deception Dataset, a publically available dataset obtained from Pérez-Rosas *et al.* [6] to test how well our model can generalize. It has 121 videos labelled as either deceptive or truthful. Perez-Rosas e*t al.* (2015) implemented their model using two classifiers: Decision tree and Random Forest.

Our model performed with an accuracy of 61% with a loss of 0.25% as shown in Table 1 compared to the Real Life Dataset with an accuracy of 68% for Random Forest and 74% for Decision Tree Classifiers.

However, our model performed above baseline of 50.4% despite insufficient training and computation time. We cannot clearly define if a person is being truthful or deceptive but we establish the use of CNN features to identify gestures that can serve as cues to deception and how deception can easily be detected if these cues are known.

Table 1: Comparative analysis with other models

Model	**Accuracy**
CNN+BiLSTM	0.61
RF	0.68
DT	0.74

4.3 Discussion of Result

From the implementation of the research done by Pérez-Rosas *et al.* [6], it can be found that:

i. There is a baseline threshold of 50%
ii. Using a total of 121 videos, an accuracy of 68% and 74% was gotten for Random Forest and Decision Tree Classifiers respectively.
iii. For detecting nonverbal cues, they employed the MUNIM coding scheme to annotate gestures found in the videos.

However, this study for detecting the nonverbal cues and the relationship between gestures and deceptive acts, a Convolutional Neural Network. Neural networks have the ability to learn from the given data, draw inferences and recognize patterns from what it has learnt. CNN was used to learn the different behavioural gestures and cues exhibited in the videos before passing the learned gestures to the BiLSTM algorithm, which then classifies as either deceptive or truthful. Using a total of 24 videos and running at 3 epochs, an accuracy of 61% was obtained. Figure 6 shows the training and test accuracy of the model. This is comparable to the result gotten by Pérez-Rosas *et al.* [6] and is above the baseline required. Figure7 shows a comparison between our model and that of Pérez-Rosas *et al.* [6].

It can be established from this research work that the proposed combination of CNN and BiLSTM algorithm worked efficiently on identifying deceptive acts from visual videos.

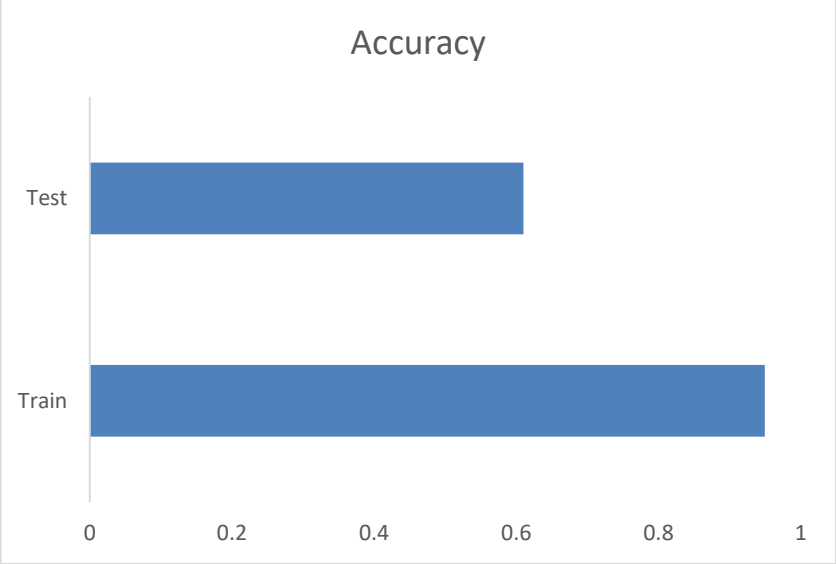

Figure 6: Train and Test accuracy

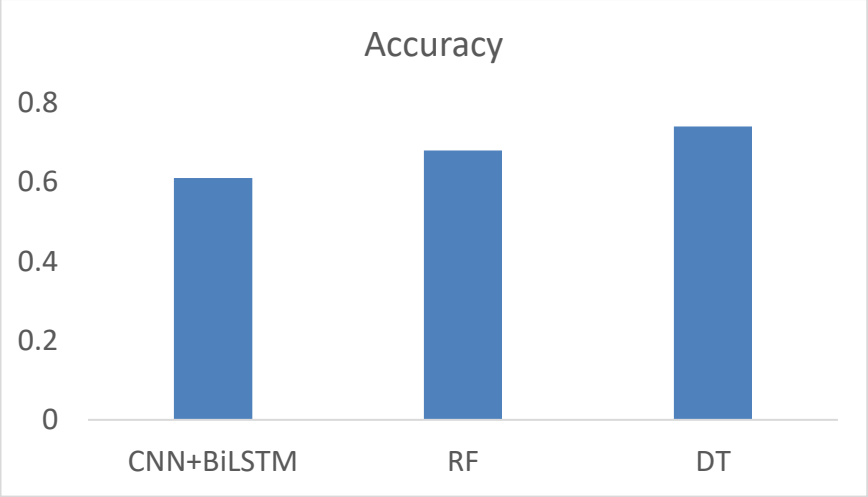

Figure 7: Evaluation with other models

5. Conclusion

In this study, we developed a model that uses CNN features to predict if a video contains deceptive acts or not by taking into cognizance the non-verbal cues exhibited in the visual images of that video. Using CNN features, we got a test accuracy of 61%.

From the research done, it was proven that it is possible to identify deceptive acts from visual videos. Although the accuracy gotten was 61%, we see that improving the processing time and the size of the dataset will lead to an improved accuracy and overall performance of the system. Future work will include sourcing local data from surveillance cameras and security feeds to validate this work.

References

[1] Buller, D. B., & Burgoon, J. K. (1996). Interpersonal Deception Theory. *Communication Theory*, 203-242.

[2] Ekman, P., & Friesen, W. V. (1969). Nonverbal Leakage and Clues to Deception. *Phsychiatry Journal for the Study of Interpersonal Processes, vol 32*, 88-105.

[3] Jensen, M. L., Meservy, T. O., Burgoon, J. K., & Nunamaker, J. F. (2008). Video-Based Deception Detection. *Intelligence and Security Informatics, SCI 135*, 425-441.

[4] Lu, S., Tsechpenakis, G., Metaxas, D. N., Jensen, M. L., & J. Kruse. (2005). Blob Analysis of the Head and Hands: A Method for Deception Detection. *Hawaii International Conference on System Science*.

[5] Tsechpenakis, G., Metaxas, D., Adkins, M., Kruse, J., Burgoon, J., Jensen, M., Nunamaker, J. (2005). HMM-Based Deception Recognition from Visual Cues. *2005 IEEE International Conference on Multimedia and Expo, Amsterdam*, 824-827.

[6] Pérez-Rosas, V., Abouelenien, M., Mihalcea, R., & Burzo, M. (2015). Deception Detection using Real-life Trial Data. *2015 ACM on International Conference on Multimodal Interaction*, 59-66.

[7] Aamodt, M., & Custer, H. (2006). Who can best catch a liar? a meta-analysis of individual differences in detecting deception. *Forensic Examiner , 15(1)*, 6-11.

[8] Wu, Z., Singh, B., Davis, L. S., & Subrahmanian, V. (2018). Deception detection in videos. *Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, 1-8.

[9] Avola, D., Cinque, L., Foresti, G. L., & Pannone, D. (2019). Automatic Deception Detection in RGB videos using Facial Action Units. *13th International Conference on Distributed Smart Cameras (ICDSC 2019)*.

[10] Baltrusaitis, T., Zadeh, A., Lim, Y. C., & Morency, L.-P. (2018). OpenFace 2.0: Facial Behavior Analysis Toolkit. *13th IEEE International Conference on Automatic Face & Gesture Recognition*, 59-66.

[11] Fukushima, K. (1980). Neocognitron: A self-organizing neural network for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics, 36(4)*, 193-202.

[12] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE (Volume: 86, Issue: 11, Nov. 1998)* , 2278 - 2324.

[13] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature 521*, 436–444.