Detecting Deception through Face Reading ML Model

Richard Shum, Zhuyin Lyu, Christy Yung

Anonymous submission

Paper ID

1. Introduction

Detecting deceptions can be helpful in various contexts. In earnings calls, accurate information is vital for investors and stakeholders to make informed financial decisions. A model capable of identifying deception could help uncover discrepancies or misleading statements, thereby safeguarding against financial manipulation. For politicians, whose words carry considerable weight in shaping policies and public opinion, detecting dishonesty can foster accountability and trust in governance. By developing a face reading ML model to detect lies, we aim to uphold integrity and accountability across various domains, ultimately contributing to a more informed and trustworthy society.

2. Related Work

Detecting deception is challenging. Most researchers agree that there is currently no single, clear indicator akin to "Pinocchio's nose" that reliably indicates deceit. [1] Nevertheless, certain observable facial expressions can, to some degree, detect lying, including but not limited to:

- Lip Biting: Individuals disclose emotional distress by biting thier lips, either externally or internally. [2]
- Micro-expressions: Negative emotions, such as anxiety and fear, last only around $\frac{1}{8}$ of a second before transitioning to a positive or neutral expression [2]
- Dilated pupils: Pupils tend to enlarge when a person is under mental strain [2]
- Vocal tones: When people are stressed or being deceptive, the average pitch and fundamental frequency of their voice go up [3]

Emotional facial expressions are subtle, and these minor movements might be unnoticed by human observers. Differences in these subtle features between lying and truth-telling situations are more effectively detected by computer vision, as it can discern movements that are beyond human perception. [1]

3. Methods and Objectives

We used two datasets to train our machine learning models: the Miami University Deception Detection Dataset (MU3D)[1] and the University of Michigan Real Life Court Trial Dataset (Trial)[2]. The MU3D dataset, which was collected in a lab environment, includes participants' honest and deceptive statements about their social interactions. The Trial dataset, in contrast, contains actual trial testimonies, which carry higher stakes. Due to the superior visual quality of the MU3D dataset, we initially focused on using it for the development of our earlier models.

Dataset	# of videos	Labels	Stakes
MU3D	320	120 Truths, 120 Lies	Low
Trial	121	60 Truths, 61 Lies	High

3.1. Data Preprocessing

To ensure that we are able to capture micro-expressions from the data, we first carried out preliminary video analysis. For both datasets, we determined the average frame rate, which is the number of frames displayed per second in a video, and the duration, which is the total length of time it plays from start to finish.

Dataset	# Avg FPS	Max FPS	Min FPS
MU3D	29.98	30.00	29.97
Trial	26.66	30.00	10.00

Ideally, we would want to capture as many frames as possible because micro-expressions last only a brief time. However, given the varying frame rates across videos, we have decided to capture a frame every 0.1 seconds, capping the number of frames collected to 240 per video. This strategy ensures consistent sampling of these brief expressions across all datasets.

[1] https://sc.lib.miamioh.edu/handle/2374.MIA/6067

[2] https://lit.eecs.umich.edu/downloads.html#Real-life%20Deception

After capturing each video frame, we processed them using Google MediaPipe, a Python package for feature extraction. Each image yields 478 facial features, which are then converted into an array. Each facial landmark consists of XY screen coordinates that are normalized to [0,1]. Therefore, for each video, we obtain a numpy array of size [240 frames, 478 landmarks, 2 normalized XY coordinates]

3.2. Model 1 - MLP Approach

We randomly selected 40 frames from each video and detected 478 landmarks from these selected frames. There are two reasons for choosing this approach. First, using fewer features can expedite model training. With each video containing at least 240 frames, and each frame comprising 478 landmarks (2 coordinates per landmark), we would end up with 229,440 features. Secondly, lie detection differs from traditional classification tasks. While detecting activities like running involves continuous motion, cues or signs of deception may occur intermittently in a video. Hence, we opted to collect frames randomly to accommodate the sporadic nature of deception cues in videos.

The model processes this high-dimensional input starting with a 32-neuron layer, which compresses the data into a more abstract form. This is followed by a 16-neuron layer that further refines the data, honing in on the most relevant features for detecting deception. Both layers use sigmoid activation functions, which are ideal for binary classification as they help the model output probabilities between 0 and 1.

Despite its straightforward structure, the model struggles with performance, achieving only 50% accuracy on both training and test sets. This result suggests that it does not effectively capture deceptive behavior. This limitation underscores the potential need for more sophisticated approaches, such as LSTM (Long Short-Term Memory) networks, which can better handle sequential data like time-series data.

3.3. Model 2 - LSTM with Facial Landmarks

In reviewing Usama's work on Deception Detection in Videos[3], we adopted a similar approach using the LSTM model: we grouped the results of 30 consecutive video frames as one input (or chunk). For example, if the video consists of 240 frames, it will be divided into eight chunks, which are subsequently inputted into the LSTM model. We chose LSTM because a single image or video frame is insufficient to determine whether a person is lying. The LSTM model takes sequences of facial landmarks as input and outputs a probability indicating the likelihood of the person telling the truth. The model architecture consists of an LSTM layer followed by a dropout layer, a linear layer, and a sigmoid activation function.

The LSTM's architecture, with its various gates, determines the retention or omission of information at each timestep. This design allows the LSTM to retain memory of significant occurrences in prior frames while discarding non-essential data, enabling effective capture of temporal dependencies within the input sequences.

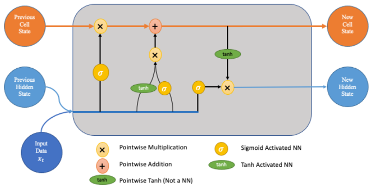


Figure 1. LSTM Model

After passing through the LSTM layers, the output is routed through a dropout layer with a dropout rate of 0.20. This dropout layer aids in preventing overfitting by randomly dropping a subset of features during training, thereby promoting better generalization. The output of the dropout layer then enters a linear layer with an output size of 1. This linear layer transforms the LSTM outputs into a single prediction score. Finally, the output of the linear layer undergoes a sigmoid activation function. This activation function compresses the output scores into the range [0, 1], which indicate the likelihood of the subject telling the truth.

3.4. Model 3 - LSTM with a focus on Eye Landmark

For this model, we focused only on the eye facial landmarks. Our hypothesis was that eye landmarks could be a stronger indicator of deceit than other facial landmarks. Using the eye landmarks, we aim to calculate sequences of blink ratios, which are fed into the LSTM model.

Using the coordinates of specific eye landmarks[4], we are able to calculate the average blink ratio for each video frame. This metric determines the probability that the eyes in each frame are open, based on the eyes' geometric configuration. We first identified the most extreme left, right, top, and bottom points of each eye, then calculated eye width and height using Euclidean distances. Following this, we computed the ratio of horizontal to vertical distances for each eye. To ensure there is no discrepancy, we averaged the blink ratios from both eyes. A blink ratio exceeding the

[3] Ahmed, H. U., Bajwa, U. I., Zhang, F., & Anwar, M. W. (2021). Deception Detection in Videos using the Facial Action Coding System. ArXiv. /abs/2105.13659.

[4] https://github.com/Asadullah-Dal17/Eyes-Position-Estimator-Mediapipe/blob/master/Eye$_{T}racking_{p}art2/main.py$

threshold of 5.5 typically indicates that the eyes are either closing or already closed.

In addition to calculating blink ratio, we also attempted to capture the gaze direction using iris positions (See Code for description). To do so, we derived a ratio of the center-to-right point distance relative to the eye width. However, we encountered an issue where the ratio, expected to be between 0 and 1, exceeded 2, consistently indicating that an incorrect leftward gaze. Upon reviewing our data, we suspect that an erroneous index for one of the eye's data points may have led to an inaccurate distance measurement.

3.5. Model 4 - LSTM with Action Units

Instead of taking the facial landmarks, we used OpenFace to extract the Action Units (AUs) for each frame. We inputted each chunk (30 frames) of AUs into the model.

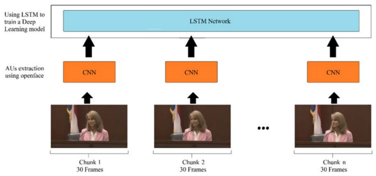


Figure 2. LSTM Model

Action Units (AUs) track the movements of individual or group facial muscles. Using OpenFace, we were able to detect 35 Action Units (existence of 18 AUs and intensity of 17 AUs). Using AUs derived from the Facial Action Coding System (FACS) instead of facial landmarks offers specific benefits for deception detection:

- Specificity in Expression Detection: AUs are direct indicators of specific facial muscle movements. AUs allow for a more precise analysis of subtle facial movements, which are crucial in identifying microexpressions often associated with deception.
- Reduced Noise in Data: By focusing on specific muscle movements, AUs reduce the data noise that might come from irrelevant movements or changes in facial landmarks not directly linked to deceptive behavior.
- Enhanced Analytical Accuracy: Studies have shown that AUs directly correspond to underlying emotional states, which can reveal inhibited emotions.

While each AU is identified based on the facial landmarks at a specific moment, the concept of an AU inherently includes the idea of movement or change in facial muscle position, even if this movement is inferred from a single snapshot. In a single image, an AU is detected by observing the configuration and relative positioning of facial landmarks that correspond to the activation of specific facial muscles. For example, AU12 (Lip Corner Puller) is detected by noting the relative expansion or pulling of the corners of the lips, which can be observed even in a still image. With this approach, we were able to significantly increase our accuracy from the previous models.

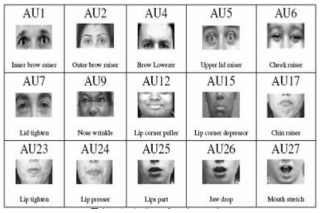


Figure 3. Action Units Examples

4. Experiments

As described in Section 3, we used 2 datasets, the MU3D dataset and the Trial dataset.

The following table shows the hyperparameters we used for different models:

Models	Learning Rate	Dropout Rate
Model 1	0.0001%	0.20
Model 2	0.01%	0.20
Model 3	0.01%	0.20
Model 4	0.01%	0.17

Table 1. Hyperparameters

The model is evaluated based on its performance in training and test data. Each dataset is split into 80% training data and 20% testing data. We also ensure an equal representation of truths and lies within both the training and testing datasets. Throughout each epoch of training, loss and accuracy metrics are computed for both the training and test sets. Loss is calculated using the Binary cross-entropy function, which measures the difference between predicted probabilities and actual binary outcomes. Accuracy is calculated as the number of true positives and true negatives divided by the total number of instances. These metrics offer insight into the model's ability to learn patterns within the data and its generalization to unseen data.

Detecting lying is a challenging task, even for humans. Therefore, our objective was to simply develop a lie detection model capable of surpassing random guessing (baseline=50% accuracy).

Throughout the four models we built, the training accuracy has gradually improved. In Model 1, both the training and test accuracy stand at 50%, which is no different from coin flipping. Examining the graph reveals a very slow decrease in loss, which eventually plateaus. The slow decline suggests that this model may not be the most effective way to capture microexpressions. Consequently, we decided to employ LSTMs, as capturing long-term patterns inherent in sequential data could yield more promising results.

In Models 2 and 3, where we integrated an LSTM model with facial landmarks and eye landmarks, the training and test accuracy showed an improvement of around 1-3%, which is better than random guessing but still not very high. From the graph of Model 2, at around Epoch 26, the test loss no longer decreases (0.6932), with training and test accuracies plateauing at 51% and 52%, respectively.

The results improved significantly using an LSTM with Action Units (Model 4). When the model is trained on the MU3D dataset, the testing accuracy is 7% higher than random guessing, specifically, with a training accuracy of 82% and a test accuracy of 57% at Epoch 19, which are much higher than previous models and fall within the range of many of the previous research results. However, when using the real-world Trial dataset, the accuracy increases even further. At Epoch 26, we have a train accuracy of 96% and test accuracy of 86%. These results significantly outperform the benchmark, demonstrating the model's effectiveness in lie detection

5. Conclusions

Unlike traditional classification tasks, detecting microexpressions associated with lying poses a unique challenge. These fleeting expressions can be hard to detect. Therefore, our decision to employ LSTM networks is crucial. It helps to capture long-term dependencies in sequential data, making them well-suited for analyzing the subtle and intricate patterns inherent in the dynamics of human behavior, especially when it comes to detecting deception.

Our project has highlighted the significant role that the dataset plays in training a model. Initially, we focused on the MU3D dataset, attracted by its standardized setting and high resolution. However, despite our efforts with various data processing methods, we only achieved around 50%-60% accuracy in training or testing. This led us to question whether micro-expressions could reliably indicate deception. Upon transitioning to the trial dataset, our test accuracy surged to 86%. This contrast underscores the crucial role of dataset selection in achieving accurate outcomes. It suggests that when individuals face higher consequences for being caught lying, their micro-expressions are more apparent.

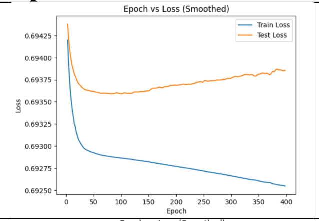
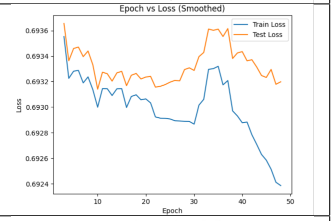
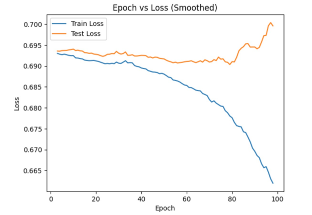
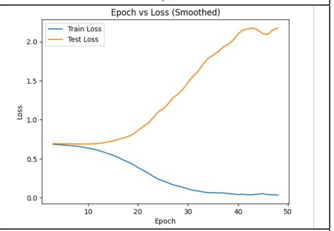
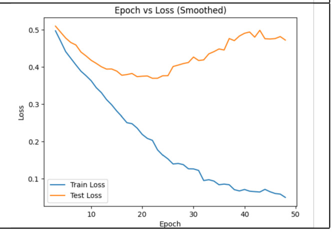
Method	Accuracy Rate	Epoch vs Loss Plot
Model 1 (MU3D Dataset)	train accuracy = 50% test accuracy = 50%	
Model 2 (MU3D Dataset)	train accuracy = 51% test accuracy = 51%	
Model 3 (MU3D Dataset)	train accuracy = 54% test accuracy = 53%	
Model 4 (MU3D Dataset)	train accuracy = 82% test accuracy = 57%	
Model 4 (Trial Dataset)	train accuracy = 96% test accuracy = 86%	

Table 2. Experiment Results

Future research could be done to refine this project. For example, we can investigate whether the increase in accuracy is merely due to a change in dataset or a change in methods (AUs vs facial landmarks). Incorporating text and audio data is also promising. Audio signals, such as pitch, tone, and speech content, can convey emotional states or psychological stressors indicative of dishonesty. While our current focus is on facial expressions, expanding to other body movements could potentially enhance accuracy assessments.

6. Contributions

Member	Coding	Report Writing
Richard	Data Processing; Model1 MLP; Model4 LSTM with Action Units (on both datasets)	3.2 Model 1; 3.5 Model 4
Christy	Model3 LSTM with Eye Landmarks (on MU3D dataset)	3.1 Data Processing; 3.4 Model 3; 5. Conclusion
Zhuyin	Modeld2 LSTM with Facial Landmarks (on MU3D dataset)	3.3 Model 2; 4. Experiments
All		1. Introduction; 2. Related Work

Github Repository: https://github.com/shumh/Deception-Detection-through-Machine-Learning

7. References

[1] Shen, X., Fan, G., Niu, C., Chen, Z. (2021, May 17). Catching a liar through facial expression of fear. Frontiers. https://www.frontiersin.org/journals/ psychology/articles/10.3389/fpsyg.2021.675097/full.

[2] Moi Hoon Yap, Rajoub, B., Ugail, H., Zwiggelaar, R. (2011). Visual cues of facial behaviour in deception detection. 2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE), 294–299. https://doi.org/10.1109/ICCAIE.2011.6162148

[3] S. Sondhi, R. Vijay, M. Khan and A. K. Salhan, ”Voice analysis for detection of deception,” 2016 11th International Conference on Knowledge, Information and Creativity Support Systems (KICSS), Yogyakarta, Indonesia, 2016, pp. 1-6, doi: 10.1109/KICSS.2016.7951455.

[4] B. Rajoub, M. H. Yap, and H. Ugail, Face reading technology for lie detection, https://www.cl.cam.ac.uk/research/security/seminars/ archive/slides/2011-10-25.pdf (accessed Feb. 29, 2024).

[5] Ahmed Khan, H.U.D., Bajwa, U.I., Ratyal, N.I. et al. Deception detection in videos using the facial action coding system. Multimed Tools Appl (2024). https://doi.org/10.1007/s11042-024-19153-4

[6] Burgoon, J. K. (2018). Microexpressions Are Not the Best Way to Catch a Liar. Frontiers in Psychology, 9, 1672–1672. https://doi.org/10.3389/fpsyg.2018.01672

[7] Gallardo-Antolín, A., Montero, J.M. (2021). Detecting Deception from Gaze and Speech Using a Multimodal Attention LSTM-Based Framework. Applied Sciences.

[8] Monaro, M., Maldera, S., Scarpazza, C., Sartori, G., Navarin, N. (2022). Detecting deception through facial expressions in a dataset of videotaped interviews: A comparison between human judges and machine learning models. Computers in Human Behavior, 127, Article 107063.