CS230

DeepLie: Detect Lies with Facial Expression (Computer Vision)

Kai (Jiabo) Feng
jiabo@stanford.edu

Abstract

Lie detection is a highly valued topic in both public and private sectors. We propose a computer vision-based approach to detecting lies in video streams, improving upon previous work. By detecting micro-expressions universal to human, we use deep learning to recognize patterns of human emotions when telling lies in front of a camera.

1 Introduction

There are many different approaches to lie detection in videos, including uni-modal approaches such as audio, text, video (micro expression), and multi-modal fusion of audio, text and video. Although multi-modal approach would be able to utilize the most amount of information to detect deception, in real world scenarios we often need to detect lies in real-time, and thus micro-expression based visual-only approach is worth exploring.

Detecting lies in videos with facial expressions requires several key building blocks, including recognizing human faces, identifying the face that we are interested in detecting lies from (in videos where more than one human face is present), recognizing the micro-expressions not immediately visible to the human eyes due to the short duration of appearance and subtle facial muscle movements, and finally using time series of data to detect lies. The outcome will be a model that detects a time series of facial expressions on the target human face and uses the expression vector to categorize the video as either a lie or a truth statement.

2 Related Work

Previous work on deception detection has focused on a combination of different factors including verbal and non-verbal aspects. Text / audio only approaches alone using RNN or LSTM architecture were able to achieve only moderate amount of accuracy 76% - 84% [1]. Micro-expression only approaches achieved higher accuracy of 77% - 88% [1]. A 2018 paper [2] focused on "Deception Detection in Videos" concluded that micro-expression was the best performing approach among different methods they explored in detecting deception in videos. In a paper published on CVIP 2019 [3], a group of researchers explored an visual-only approach by extracting visual features using CNN from video input and then feed them into an LSTM sequence learning model to output binary classification. We will further expand on previous work to focus exclusively on micro expressions and using the sequence of micro-expressions as the input for our deep learning model.

CS230: Deep Learning, Spring 2021, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Dataset and Features

There are two main datasets required for this learning task. The first is faces labeled with corresponding expressions. The second is video clips of human telling truth / lies.

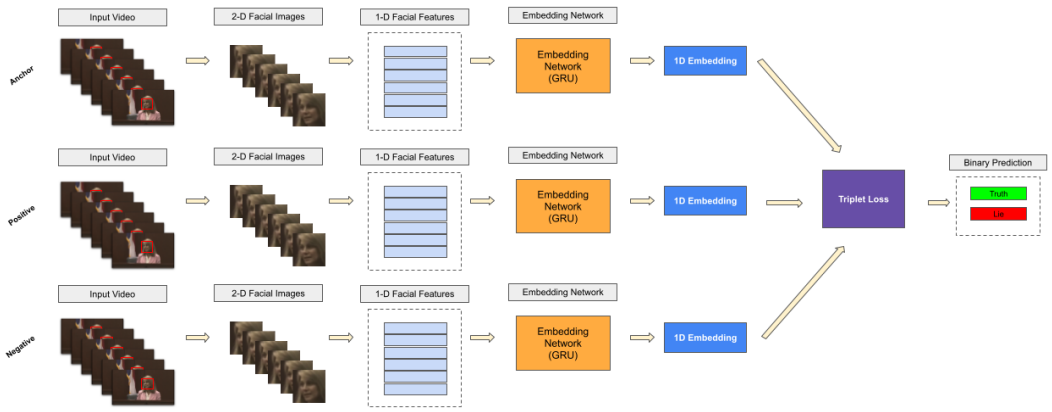
To train the expression recognizer, we use FER-2013 ("Learn facial expressions from an image") available on Kaggle. The training set contains 28,709 examples and the test set contains 3,589 examples. Each image is pre-cropped to contain only the human face and is labeled using numbers 0-6 which stand for 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral.

For the video clip data, we are using a labeled video clip set consisting of 60 truths and 61 lies, which is the same dataset used by a previous paper "Deception detection in videos", in AAAI 2018 [2]. The average video is about 2 minutes in length, and all video files are in mp4 format with a frame rate of 30 frames per second. Only the person making a statement is visible in each video (there are occasional appearance of the judge or other people but such occurrences are very rare and can be ignored for the purpose of training).

4 Methods

Figure 1 shows the model architecture of DeepLie network for reliable deception detection in videos. There are three key components to this micro-expression based lie detection algorithm. First, we turn video clips into time-stacked images of human facial expressions. Second, we use computer vision approach to convert images of human faces captured on camera into encoding vectors. Finally, we train a classification network taking the encoding vectors as input to train a binary classifier and output prediction on whether or not the video input is truthful.

Figure 1: Model architecture



Using a Siamese network architecture with triplet loss, the model is able to effectively leverage the limited amount of labeled truth / lie video data to reliably detect lies in videos. Below are a detailed explanation of the full algorithm.

In step one, we pre-process the data by reading in all videos one-by-one, and pass the images of each frame through a face detector based on OpenCV to crop the image to contain only the primary human face (assuming only one face is present in each frame of the video). Subsequently each cropped facial image is converted to grayscale and a standard 48 by 48 pixel format before stored as a numpy array. The data which consists of 121 videos (61 lies and 60 truths) are split into dev-test datasets following a 90-10 ratio; the dev dataset is subsequently split into train-valid datasets following a 90-10 ratio. The test data is stored and cast aside for final evaluation at the end of the project.

In step two, we read the facial images represented as numpy arrays into memory and then use a pre-trained facial expression recognizer CNN model to convert each image into an encoding vector which captures the most important features extracted by the facial expression recognizer.

The facial expression recognizer is trained on the FER-2013 dataset with a CNN model having 8 convolutional layers (with intermediate layers for max-pooling and batch normalization plus dropouts) followed by 4 fully connected layers leading to the final output using softmax activation. To save time, initially we have adopted a pre-built model which achieved 66.4% accuracy on test dataset. In comparison, the FER-2013 Kaggle contest's winner team had 71% accuracy, so this model performance is good enough as a baseline.

Finally, in step three, we construct an embedding model using a two-layer gated recurrent unit network (similar to how trigger word detection algorithms work) in order to detect complex patterns of facial expression / micro-movement in facial muscles that give away any trace of deception. We then use this embedding model to optimize triplet loss in a Siamese network architecture.

To construct a triplet, we randomly select a training example as the anchor, then randomly select another training example which shares the same label as the anchor as the positive, and finally randomly select a training example which has a different label from the anchor as the negative.

Triplet loss is the loss function which optimizes the embedding model weights such that the embedding vectors of same-category (truth/lie) videos are closer together while videos that are different are further away in the embedding space. Given 3 video encoding vectors generated from step 2, A, P, N, the loss for this triplet is:

$$l(A, P, N) = \max(||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + \alpha, 0)$$

And the overall loss function:

$$J = \Sigma_{i=1}^{m} l(A^{(i)}, P^{(i)}, N^{(i)})$$

Using early stopping with patience = 10 epochs and restoring the best weights, we are able to efficiently train this model without iterating through the entire possible universe of combinations of $120 \times 60 \times 60 = 432,000$ (anchor, positive, negative) triplets. The model effectively combines the advantages of both the Siamese network / triplet learning in solving one-shot learning problems (with limited data) and the GRU-based RNN network's ability to detect recurring patterns through time to detect possible "tells" of lies.

5 Experiments/Results/Discussion

In this section, we discuss the experimental results of all distinctive algorithms / approaches we have tried, primarily concerning step 3 of the DeepLie network: using deep neural networks to classify videos as truthful or deceptive. To evaluate performance of different models, we have chosen accuracy / correct classification rate (CCR) as the primary metrics. Further, we evaluate the model results using confusion matrix to understand the balance between precision and recall.

To begin with, we have trained two different classification networks to get a baseline performance for this algorithm. The first is a basic deep neutral network consisting of **5 fully-connected layers** ([linear-> relu]$\times 4$ -> linear -> sigmoid). We selected binary cross entropy as the loss function, Adam optimizer with $\alpha = 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$, with $\epsilon = 10^{-7}$, and evaluation metrics being accuracy. After training for 100 epochs with a batch size of 64, the basic 5-layer network achieved 75.19% train accuracy and 72.73% validation accuracy. This is an acceptable result to use as the baseline performance. Due to its simple nature, there are many promising directions to try in order to increase the accuracy.

For the binary classification task, we have also tried a **3-dimensional CNN** architecture, taking the 2-dimensional image plus time as the third dimension, in the hope that our 3-D filters would capture important "tells" of lies as people make the statement in videos. In this model, we also skipped using facial emotional recognizer module to encode facial images, and instead used the raw, preprocessed facial images stacked together as the the input to the Conv3D model. The Conv3D model closely mimicks the structure of the FER CNN model described in step two. Here we passed the input through 8 layers of Conv3D, with intermediate max-pooling layers and dropouts applied, before passing through 4 fully-connected dense layers leading to a sigmoid activation which outputs truth / lie predictions. Similarly, here we have chosen the binary cross entropy loss, Adam optimizer and

binary accuracy as evaluation metrics. Training for 100 epochs with a small batch size of 5 (due to memory limits), the highest results achieved is similar to the linear model (70% validation accuracy). However the performance varied a lot from training to training and the performance is not very reliable.

We then constructed an **GRU/RNN model** consisting of 2-layers of uni-directional Gated Recurrent Units (GRU) to detect patterns in time series data (after passing input through Conv1D and Max-Pooling layers). We chose 2-layers of GRU because that is sufficient to detect complex patterns in data without loss of performance (Adding additional layers of GRU-based layers only hurt the performance). This model achieves 81.82% CCR on validation set, and 94.31% on training set.

As we traversed the above experimental models, we came to realize that the one big weakness / limitation of this project is the limited amount of training data (videos of people lying or telling the truth in high-stake environments, such as the court of law). To solve this problem, we further propose the **DeepLie network** - a Siamese network with triplet loss to solve this one-shot learning problem. To do that, we treat all truthful statements as one class and all lies as another, and draw at random from the training data to build triplets consisting of (anchor, positive, negative) tuples as inputs. Since we have 60 truths and 61 lies, we will be able to access a dataset having approximately $120 \times 60 \times 60 = 432,000$ samples.

The proposed DeepLie model achieves 81.82% accuracy on validation dataset, and 100% correct classification rate (CCR) when evaluated using leave-one-out cross validation with 25 trials, which puts it on par with the model performance set forth in the CVIP 2019 paper, "Video Based Deception Detection Using Deep Recurrent Convolutional Neural Network". The training time of this model was surprisingly not very long, taking only around 50 epochs before automatically stopped by early stopping mechanics which we have employed in our model with a patience setting of 10 epochs (meaning it will stop the training early if the objective function didn't improve in 10 consecutive epochs) and restoring the best weights when stopped.

Figure 2: PCA and Loss vs. Epochs($\alpha = 0.6$)

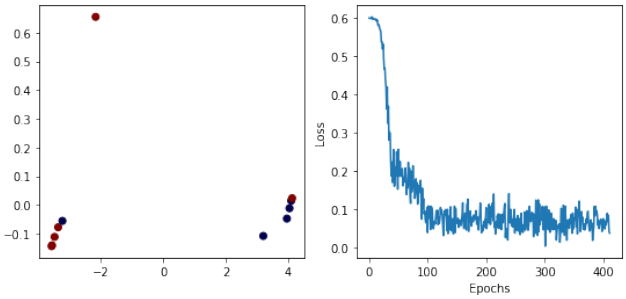


Figure 3: PCA and Loss vs. Epochs($\alpha = 0.2$, Early Stopping)

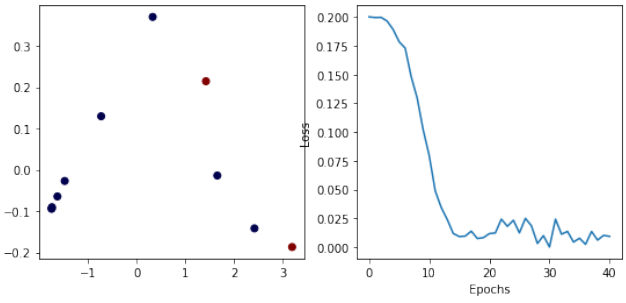
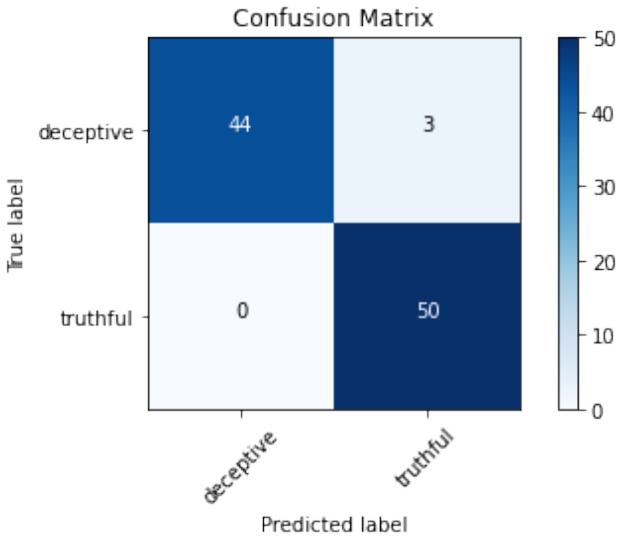


Figure 2/3 shows the principle component analysis (PCA) plot of the validation data in the top 2 dimensions of the embedding space, as well as the loss function evaluated after each epoch. We tried different values of α ranging from 0.2 to 0.8 and it didn't seem to make a big difference in the validation / test CCR, thus we landed on $\alpha = 0.2$ since it is a common value to use and also not so

large that we risk over-fitting training data by altering the embedding too much with each training sample.

Looking at the confusion matrix as shown in Figure 4, the DeepLie model labeled 3 lies as truthful statements but none of the truthful as lies. This suggests a lower recall but higher precision in detecting lies, making it a potential fit for helping the justice system to detect lies since although we might miss some lies, we are less likely to mistakenly declare truthful statements as lies.

Figure 4: Model architecture



6 Conclusion/Future Work

In this work, we have proposed an integrated end-to-end algorithm for detecting lies in videos, based primarily on the facial expressions extracted from each frame of the video. The proposed method is based on an underlying 2-layer GRU network model which effectively detects any recurring patterns of facial expression and micro-movements of facial muscles to detect lies. Using a Siamese network architecture with triplet loss constructed from a dataset of 61 truths and 60 lies in court videos, DeepLie algorithm solves the one-shot learning problem and avoids over-fitting data according to idiosyncratic traits of individual faces, optimizing for an embedding space that effectively separates truthful and deceptive videos from one another. DeepLie, the best performing algorithm, combines the Siamese Network and uni-directional RNN model to function as a real-time lie detector which errors on the side of missing lies to avoid wrongfully accusing an honest person of telling a lie.

Further work could be done to collect more diverse dataset and expand the occasions of the training videos to outside of the courtroom into more casual and diverse scenarios. Also, a multi-modal approach which expand DeepLie model to also use audio, text and gesture / body movement could yield further improvements. Lastly, extensive testing should be conducted (especially looking into classification results of minorities) to evaluate the fairness of this model before it can be deployed to real-world applications.

7 Contributions

As the only team member, I contributed to 100% of this project.

Appendix

The full project code can be accessed on Google Colab using this link: `https://colab.research.google.com/drive/1sjIOcNrolQEws2sKKuLPMhtKj9-uAW3b?usp=sharing`

Figure 5: Example FER results using a CNN model

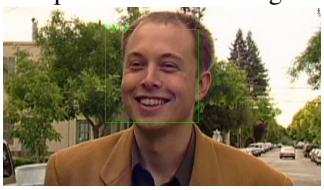
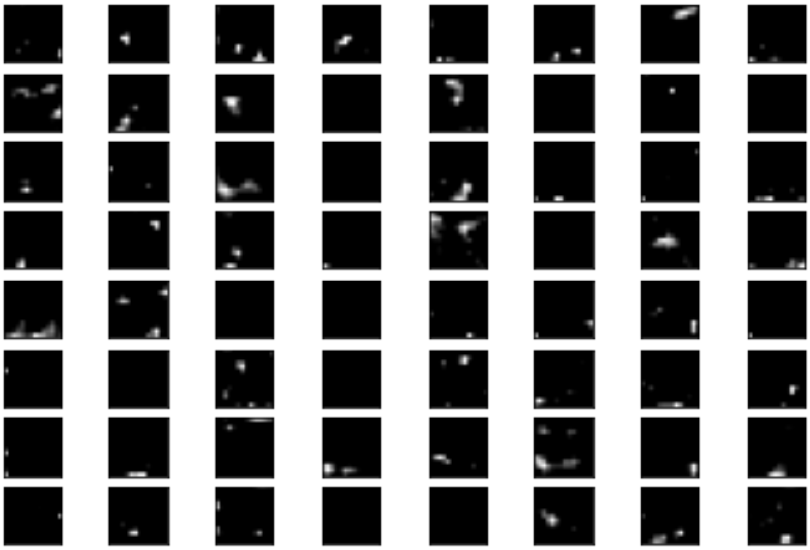


Figure 6: Visualization of a deep layer in VGG-16 model



References

[1] Venkatesh, S. & Ramachandra, R. & Bours, P., (2019) *Robust algorithm for multimodal deception detection*. In: 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 534–537, March 2019

[2] Zhe Wu & Bharat Singh & Larry S. Davis & V. S. Subrahmanian, (2018) *Deception Detection in Videos*. AAAI.

[3] Sushma Venkatesh & Raghavendra Ramachandra(B) & Patrick Bours, (2019) *Video Based Deception Detection Using Deep Recurrent Convolutional Neural Network*. CVIP 2019.

[4] Gangeshwar Krishnamurthy & Navonil Majumder & Soujanya Poria & Erik Cambria, (2018) *A Deep Learning Approach for Multimodal Deception Detection*.