

Enhancing Lie Detection Accuracy: A Comparative Study of Classic ML, CNN, and GCN Models using Audio-Visual Features

Abdelrahman Abdelwahab
abdelrahman.a.abdelaziz28@gmail.com
STEM high school for boys 6-October

Akshaj Vishnubhatla*
akshajvishnubhatla@gmail.com
Briar Woods HS

Ayaan Vaswani*
ayaanvaswani@gmail.com
Bellarmine College Preparatory

Advait Bharathulwar*
advaitbharathulwar@gmail.com
John P Stevens High School

Arnav Kommaraju
arnavkommaraju@gmail.com
Edison High School

Bohan Yu
ybhtim@berkeley.edu
UC Berkeley

Abstract—Inaccuracies in polygraph tests often lead to wrongful convictions, false information, and bias, all of which have significant consequences for both legal and political systems. Recently, analyzing facial micro-expressions has emerged as a method for detecting deception; however, current models have not reached high accuracy and generalizability. The purpose of this study is to aid in remedying these problems. The unique multimodal transformer architecture used in this study improves upon previous approaches by using auditory inputs, visual facial micro-expressions, and manually transcribed gesture annotations, moving closer to a reliable non-invasive lie detection model. Visual and auditory features were extracted using the Vision Transformer and OpenSmile models respectively, which were then concatenated with the transcriptions of participants’ micro-expressions and gestures. Various models were trained for the classification of lies and truths using these processed and concatenated features. The CNN Conv1D multimodal model achieved an average accuracy of 95.4%. However, further research is still required to create higher-quality datasets and even more generalized models for more diverse applications.

Index Terms—multimodal, polygraph, GCN, facial micro-expressions, multimodal model, deception detection, conv1d, CNN.

I. INTRODUCTION

Lie detection is a recurring focus of research and technological innovation in law enforcement and criminal justice. According to a survey conducted by the University of Wisconsin-La Crosse, about 75% of survey respondents reported telling zero to two lies per day; lying comprised 7% of the total communication, with 79% of the lies being told face-to-face and 21% being mediated [3].

Current technologies, such as polygraphs, have focused on biological responses such as blood pressure to detect lies. However, these methods are unpredictable and flawed easily. Recently, research has begun to focus on various indicators of deception, including facial micro-expressions and audio cues [4]. Facial micro-expressions (ME) are intentional or involuntary localized and momentary movements of the face, usually lasting less than 500 ms [2].

Despite advances in lie detection techniques, traditional methods remain intrusive, subjective, and often inaccurate. Detecting deception through ME and speech analysis presents a significant challenge owing to the subtle and brief nature of these cues. As shown in Table I traditional methods have high variance and relatively low accuracy. This study aimed to address these limitations by developing a non-intrusive, objective, and highly accurate method for detecting deception using both ME and audio signals. Accurate lie detection is crucial in various fields, including security, legal systems, and psychological evaluation. The primary objective of this study was to establish an AI model that can differentiate between truth and deception with high accuracy by analyzing audio, visual cues in videos, and extracted gestures. Audio dialogue, visuals, and gestures help distinguish between deception and truthfulness, making them important features to consider [23]. Therefore, the Real-life Deception Detection Dataset from the University of Michigan was used, which includes 121 videos of deception and truthfulness and a CSV file for gestures. Visuals and audio were extracted from the videos, and OpenSMILE and Vision Transformer (ViT) were used to extract features from the audio and video, respectively. Classical machine learning models, such as Random Forest Classifiers and Logistic Regression can serve as accurate baseline references for binary classification tasks, such as truth and lie. Yet to build off of that, by leveraging advanced neural network models, such as Conv1D, Graph Convolutional Networks (GCN), and Convolutional Neural Network Long Short-Term Memory (CNN LSTM), the accuracy can be increased.

This study addresses the following research questions: How effective is the proposed AI model for detecting lies compared with traditional methods and some recent AI models? Which features carry the highest weights in prediction? Deception detection technology has the potential to revolutionize various fields. Law enforcement can improve interrogation outcomes and border security by identifying deceptive behaviors. In

TABLE I
ESTIMATED ACCURACY OF DIFFERENT TEST TYPES IN DECEPTION AND TRUTHFULNESS

Test type	Detecting deception	Detecting truthfulness
Laboratory studies		
CQT – Polygraph	74%–82%	60%–83%
CIT – Polygraph	76%–88%	83%–97%
ERP	68%	82%
fMRI	84%	81%
Field studies		
CQT – Polygraph	84%–89%	59%–75%
CIT – Polygraph	42%–76%	94%–98%

the legal system, it can be used to assess the credibility of courtroom testimonies and negotiations. Additionally, applying this technology to financial services could aid in detecting fraudulent claims and in reducing the risk of financial fraud.

Previous studies have experimented with various machine-learning models. For instance, a study by Soldner et al. implemented the Random Forest model and achieved the best accuracy of 69%, as shown in II [5]. Insights from this study suggest expanding our dataset and exploring additional modalities to enhance the model’s accuracy and reliability in lie detection. Furthermore, Random Forest, which is a machine learning technique, cannot handle complex relations or multi-modal data, which is a limitation of the aforementioned study. Moreover, most traditional AI models fall short of reliability and accuracy, often leading to false positives or negatives [1]. A study conducted by the University of Michigan in 2015 analyzed trial videos using facial expressions and achieved a rudimentary accuracy rate of 83.05% using neural networks [6]. Aligning different data types and achieving 83.05% accuracy were the two main advantages of the study.

TABLE II
BEST RESULTS OF STUDY [2].

Features	Acc.
Linguistic	66%
Dialog	57%
Non-verbal	61%
All Features	69%

This paper is organized as follows: analyzing previous work, discussing the paper’s methods (data collection, data analysis, feature extraction, and implementation guide for the tested models), presenting the results of different tested models, comparing the paper’s results with those of other studies using the same dataset, and providing a discussion including limitations and recommendations. The paper concludes with a summary of the key findings and a look forward.

II. LITERATURE REVIEW

A. Prior solutions

The paper, titled *Facial Micro-Expression Recognition Based on Deep Local-Holistic Networks*, introduces a Deep

Local-Holistic Network (DLHN) for micro-expression recognition, comprising two sub-networks: the Hierarchical Convolutional Recurrent Neural Network (HCRNN) and the Robust Principal Component Analysis Recurrent Neural Network (RPRNN). The HCRNN captures local spatiotemporal features using CNNs and BRNNs, whereas the RPRNN extracts global sparse features using RPCA and BLSTM networks. The DLHN was evaluated on four combined datasets (CASME I, CASME II, CAS(ME)2, and SAMM) and achieved an accuracy of 60.31%, outperforming several state-of-the-art methods [1].

In a study by Feng (2021), titled *DeepLie: Detect Lies with Facial Expression (Computer Vision)*, the author developed a deep learning-based approach to lie detection using facial micro-expressions in video streams. This method employs a Siamese network architecture with triplet loss to effectively distinguish between truthful and deceptive expressions. The key components include the use of CNNs for feature extraction and a GRU-based RNN for sequence learning. The model achieved an 81.82% accuracy on the validation dataset. However, the study highlighted limitations due to the small size of the dataset, which may hinder the generalizability of the model. The authors suggest that future work should focus on incorporating multi-modal data (e.g., audio and text) and expanding the dataset to include more diverse scenarios [18].

The *Hybrid Machine Learning Model for Lie Detection* research dataset included thermal images. This study employed a hybrid machine-learning approach that combines the strengths of CNNs and SVMs. CNNs were used to automatically extract relevant features from the input data automatically. The extracted features were then fed into an SVM for classification. The hybrid model demonstrated an accuracy of 58%. However, the complexity of the hybrid model, which combines CNNs and SVMs, can lead to higher computational costs and increased difficulty in scaling the approach to larger datasets [24].

Audio-Visual Deception Detection Using the DOLOS Dataset study used a DOLOS dataset that combined synchronized audio and video signals. The study employed ImageNet pre-trained ViT as the backbone network for the visual modality and tokenized face images with a 2D-CNN module, resulting in a feature with dimensions of 64×256 . For audio modality, the study adopted a pre-trained W2V2 model. The raw audio was tokenized using the 1D-CNN module, resulting in a feature size of 64×512 for each audio sample. The plugin audio visual fusion model and multi-task learning achieved an accuracy of 66.84% [19].

Numerous studies have been conducted using the Real-Life Trial Dataset. Camara et al(2024) summarized many studies conducted using this dataset. Ding et al. (2019) used a CNN model. ResNet served as the backbone of the facial expression and computed the temporal feature maps. This was used to achieve accuracy in conjunction with a Generative Adversarial Network (GAN). This model was reported to achieve a 97% accuracy, which is the highest among current deep learning models [21]. Among the non-deep learning models, the use

of SVM in Carissimi et al. (2018) provided an accuracy of 99%. This model uses features from AlexNet. Wu et al (2018) reported that a logistic regression model using Improved Dense Trajectory (IDT) features had an accuracy of 92.21%. Other studies have also been conducted on this dataset with high accuracy; however, their methodology is vague and can not be readily replicated [22]. A key problem to note is that current studies have used varying features when working with different models, making comparison between models difficult. This study attempts to better display this comparison by using the same features across models.

B. Models Overview

1) *Logistic Regression*: Two classical machine learning models were considered for this study: Logistic Regression and Random Forest Classifier. Logistic regression is a binary classification technique based on a sigmoid function. This function is used to weigh features in a way that returns a value from zero to one [32]. The logistic regression function can be defined using equation [33]

$$P(y = 1 | X) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Where:

- $P(X)$ is the probability that the outcome y is 1 given the input X .
- z is the linear combination of input features and their corresponding weights, defined as:

$$z = \mathbf{w}^T \mathbf{X} + b = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

where:

- $\mathbf{X} = [x_1, x_2, \dots, x_n]$ are input features.
- $\mathbf{w} = [w_1, w_2, \dots, w_n]$ are the weights (parameters) associated with each feature.
- b is the bias (intercept term).
- $\sigma(z)$ is the *sigmoid function*, which squashes the output into the range (0, 1):

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

This function creates an S-shaped curve that determines the binary classification. Figure 1 shows the sigmoid function. As the value approaches zero or one, the probability of a certain classification increases. This model was tested in this study because of its application in sentiment analysis, and it has high success in facial expressions recognition, as shown by Goyani et al [32]. This may mean that the model will be useful in detecting facial micro-expressions, which were previously mentioned as being vital for detecting deception.

2) *Random Forest Classifier*: The second model is the Random Forest Classifier (RF). In brief, Random Forest is an ensemble model that chain multiple decision trees during training to merge results, improve accuracy and reducing overfitting [30]. The decision trees were generated by using a bagging algorithm (voting majority). Many decision trees that can enter RF predict an output by forming a “forest”

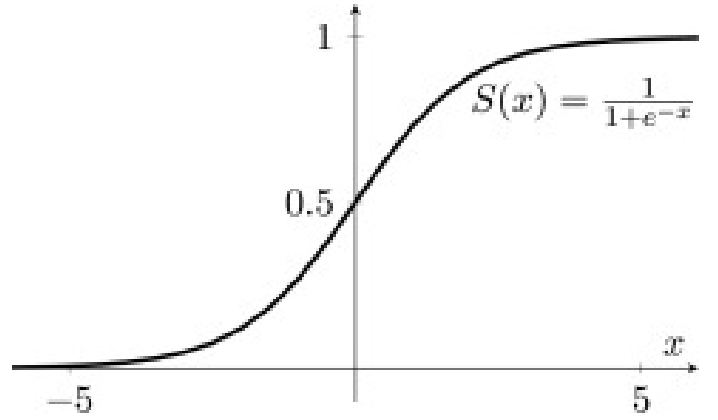


Fig. 1. Diagram of Sigmoid function [33]

of classifiers that vote for the classification of an input. The RF classifier was considered for this study particularly for the applications of decision trees in sentiment analysis, specifically for speech emotion recognition [32]. Considering that speech plays a considerable part in lie detection, the RF classifier has potential to increase deception detection accuracy.

3) *Graph convolutional Network (GCN)*: A graph consists of nodes (vertices) and edges (connections between nodes). In a GCN, each node represents an entity, and the edges represent the relationships between these entities. The primary goal of GCNs is to learn node embeddings, which are vector representations of nodes that capture the graph’s structural and feature information. Figure 2 shows a diagram for GCN.

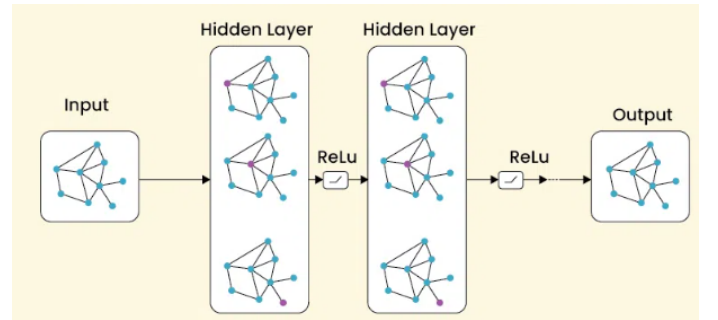


Fig. 2. shows a diagram for the GCN [34]

The graph captures the structural relations among data, harvesting more insights than analyzing data in isolation. However, it is often very challenging to solve learning problems on graphs, because (1) many types of data are not originally structured as graphs, and (2) for graph-structured data, the underlying connectivity patterns are often complex and diverse [31].

A typical GCN architecture contains the following: input layer (initializes the node features) and hidden layers (performs the graph convolution operations, progressively aggregates and transforms node features); output layer (produces the final node embeddings or predictions); and fully connected layer (is used at the end of the network to perform tasks such

as classification) [34]. Figure 3 illustrates the architecture of the GCN.

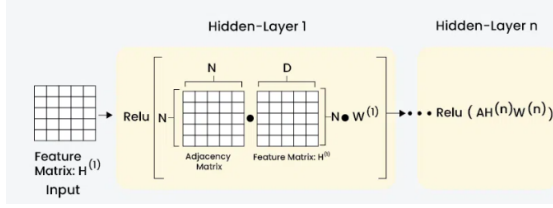


Fig. 3. shows the basic architecture for a GCN [34]

Spectral-based GCNs leverage graph Laplacian eigenvalues and eigenvectors for convolution operations, providing a strong theoretical foundation and capturing the global graph structure. The graph Laplacian eigenvalues represent the frequencies of graph signals, whereas the eigenvectors form a basis for representing functions over the graph, allowing for smooth and meaningful convolutions across the entire graph. This spectral approach effectively captures the global structural information of the graph but can be computationally intensive. Spatial-based GCNs, however, perform convolutions directly on the graph's local neighborhoods, offering greater flexibility and scalability, making them more suitable for handling large graphs and integrating with other data types [31].

4) CNN conv1d:

”Generally, 1D-CNNs are designed to handle one-dimensional data, such as time-series data, sequences (e.g., text), or any data where the primary structure is along a single axis. The kernel (or filter) in a 1D-CNN moves in one dimension. If the data are represented as vectors $[x_1, x_2, \dots, x_n]$, the kernel slides over this vector to detect the patterns within the sequence. The shape of the kernel is a 1D array with dimension $(k,)$, where k is the size of the kernel.”

– A. O. Ige and M. Sibiya, ”State-of-the-art in 1D Convolutional Neural Networks: A Survey,” [35]

In a 1D Convolutional Neural Network, the kernel moves along a single axis of the input vector, effectively processing the data. The receptive field of a 1D-CNN kernel involves a contiguous segment of 1-D input. As the kernel slides across the input, it aggregates the information from k consecutive elements. For a given input sequence x and kernel w , the convolution operation in a 1D-CNN layer can be expressed as [35]:

$$(x * w)(t) = \sum_{i=0}^{k-1} x(t+i) \cdot w(i)$$

where:

- x is the 1-d input,
- w is the kernel (or filter),
- $(x * w)(t)$ denotes the convolution of x and w at position t ,
- k is the size of the kernel,

- $x(t+i)$ is the element of the input sequence at position $t+i$,
- $w(i)$ is the element of the kernel at position i .

An illustration of three consecutive convolutional layers is presented in Fig. 4. As seen in [36], where x_i^k is used to denote the input, b_i^k is the bias of the neuron at k th position of layer l , and the output of the i th neuron in the incoming layer $l-1$ is given as $s_i^{(l-1)}$, and $w_{ik}^{(l-1)}$ denotes the kernel assigned from the neuron in the i th position of the first convolutional layer $l-1$ to the k th neuron in the second layer l . y_k^l is the intermediate output, SS is the scalar factor used in down sampling, and f is the activation function [35].

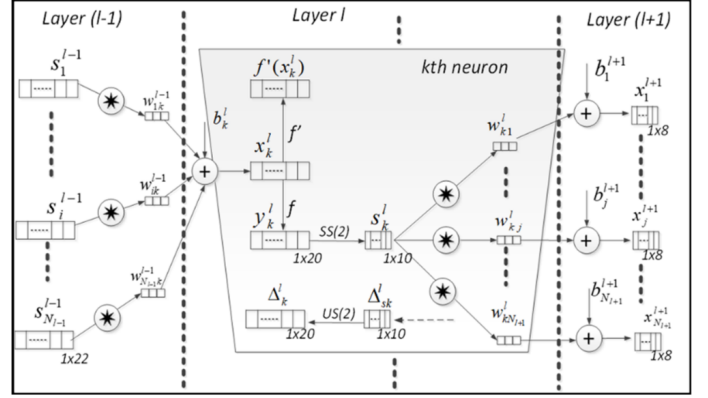


Fig. 4. Illustration of three consecutive layers in 1D-CNN [35]

Forward propagation in 1D-CNN involves passing the input through one or more convolutional layers, pooling layers, and fully connected layers, such that feature map Z_c is given as [35]:

$$Z_c = f_c(X * W_c + b_c)$$

where X is the input, W_c denotes the filter weights, b_c is the bias term, and f_c is the activation function of the convolution, and $X * W_c$ is the convolutional operation between filter weights and bias terms. The spatial dimension of Z_c is reduced by aggregating information from nearby values through pooling, which is given as [35]:

$$A_p = P(Z_c)$$

where P denotes the pooling operation. Subsequently, a fully connected layer Z_f combines the features learned from the convolutional and pooling operations, and the final activation function Y is used to obtain the output of the network. In addition, backward propagation in 1D-CNN involves computing the gradients of the loss function with respect to the network's parameters, which are used to update the weights and biases. The backward propagation in the fully connected layer is as follows [35]:

$$\frac{\partial L}{\partial Z_f} = \frac{\partial L}{\partial Y} \cdot f'_f(Z_f) \quad (1)$$

Here, L represents the loss function, and f'_f denotes the activation function in the fully connected layer. The gradient

of the loss with respect to the fully connected weights, $\frac{\partial L}{\partial W_f}$, is given by [35]:

$$\frac{\partial L}{\partial W_f} = \frac{1}{m} \frac{\partial L}{\partial Z_f} \cdot A_p^T \quad (2)$$

Similarly, the gradient of the loss with respect to the fully connected biases, $\frac{\partial L}{\partial b_f}$, is calculated as:

$$\frac{\partial L}{\partial b_f} = \frac{1}{m} \sum \left(\frac{\partial L}{\partial Z_f} \right) \quad (3)$$

Backpropagation through the pooling layer is performed as outlined in Equation (3) (which is not shown in the image). For the convolutional layer, the backpropagation is given by [35]:

$$\frac{\partial L}{\partial Z_c} = \frac{\partial L}{\partial A_p} \cdot P'(Z_c) \quad (4)$$

The gradient of the loss with respect to the convolutional weights, $\frac{\partial L}{\partial W_c}$, is expressed as:

$$\frac{\partial L}{\partial W_c} = \frac{1}{m} X * \frac{\partial L}{\partial Z_c} \quad (5)$$

Finally, the gradient of the loss with respect to the convolutional biases, $\frac{\partial L}{\partial b_c}$, is computed as:

$$\frac{\partial L}{\partial b_c} = \frac{1}{m} \sum \left(\frac{\partial L}{\partial Z_c} \right) \quad (6)$$

In these equations, m denotes the batch size, and $P'(Z_c)$ represents the gradient of the pooling operation. The terms $\frac{\partial L}{\partial W_c}$ and $\frac{\partial L}{\partial b_c}$ correspond to the gradients of the loss with respect to the convolutional weights and biases, respectively. These gradients are then used to update the convolutional weights W_c and biases b_c using gradient descent [35].

III. METHODS

A. Dataset Collection

The experiment was conducted under the following guiding question: *Can a multimodal model of facial microexpressions and speech be used to classify human deception accurately?* To successfully research a multimodal model that encompasses both a visual and speech encoder, a dataset containing both video and audio must be found.

The *Multimodal Real Life Trial dataset* was used in our experiments, which includes videos and handwritten elements, which are ideal for in-depth analysis. Each clip was labeled as deceptive or truthful and had visibility of the face of the speaker as well as the statements spoken by them during the duration of the clip as seen in the frames in Fig. 5 [25]. The dataset was composed of 121 testimonies, both truthful and deceptive, which were also manually transcribed and annotated with facial reactions.

The videos in this dataset had an average length of 28.0 seconds, with the deceptive videos averaging 27.7 seconds and the truthful videos averaging 28.3 seconds. The 56 distinct speakers in these clips were made up of 21 female and 35



Fig. 5. The dataset sample frames, pulled from [6]’s dataset, display hand movements, microfacial expressions, and facial reactions.

male speakers, each between the ages of 16 to 60 [25]. This dataset was found on the University of Michigan’s Deception Detection and Misinformation datasets list. Fig. 6 shows the distribution of the recorded annotations in the dataset.

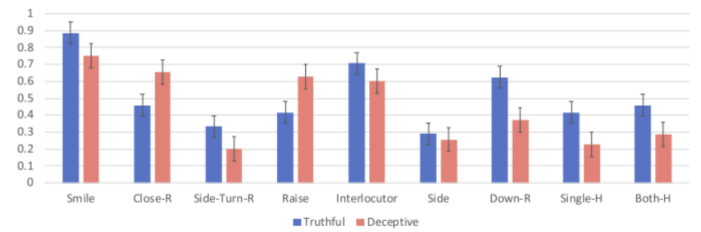


Fig. 6. The dataset sample frames, pulled from [6]’s dataset, display hand movements, microfacial expressions, and facial reactions.

B. Data Preprocessing (OpenSmile, ViT, manual annotations)

In this case, using audio, video, and textual analysis to predict the result can provide us a more accurate picture than rudimentary polygraphs and more freedom to innovate over related works and previous model pipelines. Primarily, to be able to fully grasp the importance of audio and video from the datasets, visual and auditory patterns can be extracted using two models: Vision Transformer (ViT) and OpenSmile.

ViT is an image recognition encoder. This was used to extract features from the visual data. ViT split the individual image frames of the video at a sampling rate of 50 Hz into different patches. These patches were linearly embedded with both patch and position embeddings. ViT’s benefits of computational efficiency and accuracy come to light when iterating over large datasets, and thus, we use a pretrained ViT to fit this task so that it can outperform models such as ResNet and other CNNs as mentioned in [26]. The vectors from this linear projection were interpolated to match the dimensions of the audio and text features, which were then saved for concatenation [26].

Furthermore, to satisfy the multimodal label, OpenSmile is our chosen processor, working in tandem with models such as a CNN and the simpler models mentioned below. OpenSmile, or open-source speech and music interpretation using large-space extraction, was first developed at the Technical University of Munich to create SEMAINE, a fully socially conscious software [27]. OpenSmile’s main function for that software

is for audio and emotional analysis and feature extraction, which in this study’s use case works well for identifying abnormalities in tone, hesitation, and emotional nuance between inputs of lie and truth [28]. Via the clips provided in the dataset, the “ffmpeg” package is used to successfully extract audio from the videos in a “.wav” format at a sampling rate of 50 Hz, which will aid in concatenation with other features. In the context of this research, OpenSmile takes in wave-form input audio files and analyses features such as pitch, loudness energy, and mel-frequency cepstral coefficients (MFCC). Using the aforementioned features, aspects such as tone, rhythm, and timbre are saved in vectors to help classify our data into buckets of truths and lies [28].

Finally, the features saved in the vector format from OpenSmile, ViT, and handwritten annotations can be combined using simple concatenation. These were then converted into a tensor format for experimentation with various models.

C. Dataset Curation and Analysis

The dataset was further curated to meet the project requirements. Because there was an imbalance in the number of truthful and deceptive videos, one of the deceptive videos was dropped from the dataset at random to create an even ½ split of 60 truthful to 60 deceptive videos. The dataset was then processed by splitting the audio files from the trial videos using FFmpeg, followed by OpenSmile to extract features from the audio. ViT was used to extract individual frames and extract features from videos.

The names of the different extracted feature files were placed in different split files for training, testing, and validation. The training set contained 70% of the dataset, the validation set contained 10% of the dataset, and the test set contained 20% of the dataset. These file names were used to identify the extracted feature files that were pulled during training, validating, and testing.

For CSV, the data were analyzed using pandas and seaborn. A heatmap was created to determine features that did not have a strong correlation with the target, as shown in Fig. 7. In addition, KDE using seaborn was performed while making hue = ‘class’ to observe the data distribution as illustrated in Fig 8. According to the analysis, any column that correlated less than 0.05, as shown in the column samples in Fig. 7, and the distribution of the classes (deception and truthful (0,1)) was approximately the same, as shown in the column samples in Fig. 8, was dropped. (Note: The analysis (heatmap and KDE) was performed on all columns, but here some samples were mentioned instead of all because if all columns were mentioned in the heatmap or KDE, it would take a big place. The full heatmap and KDE are in the file named ”data analysis” that was uploaded to GitHub; you can access the link in the appendix section)

D. Models

1) *Convolutional Neural Networks:* First, a Convolutional Neural Long Short-Term Memory Network (CNN LSTM), was used in our experiments. CNN LSTM models focus on

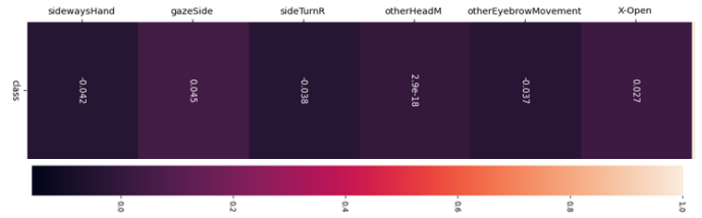


Fig. 7. shows a correlation heatmap for some of the sample columns.

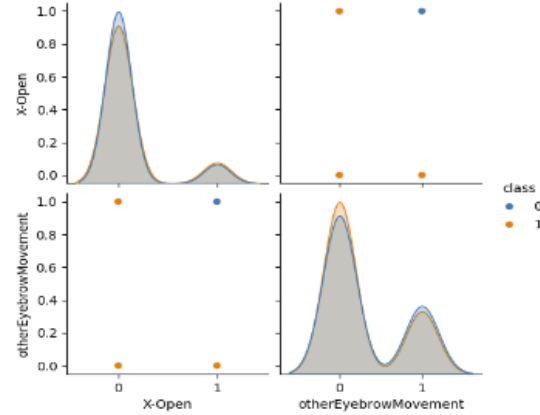


Fig. 8. shows the KDE for several columns

spatial relations in images, describing videos, actions with text, and classification [29]. In this case, the CNN LSTM takes concatenated data of audio, visual, and annotated features and then classifies the data as truth or lie.

The second iteration of CNNs is a basic custom Convolutional Neural Network. Using a large variety of layers within a 1-dimensional CNN framework, the model matches well with the results of the data preprocessing. Furthermore, to regulate and standardize the data to avoid overfitting due to lack of samples, a dropout function and Max Pooling function, as shown in Equation 7, are used when needed in the layers.

Further, for the classification aspect of the complex CNN, a classical sigmoidal function is used from the neural network python module because sigmoid is good in binary classification; similarly, the loss function is Binary Cross Entropy. Equation 8 shows the sigmoid function, and Equation 9 shows the binary cross entropy loss function.

$$\mathbf{F}_{\max(\mathbf{x})} = \max \{x_i\}_{i=0}^N \quad (7)$$

$$\text{Sigmoid}(x) = \sigma(x) = \frac{1}{1 + \exp(-x)} \quad (8)$$

$$\text{BCE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (9)$$

2) *Classical Machine Learning Models:* Furthermore, based on the characteristics of the dataset, pre-processing output tensors, and the number of samples, more simplistic

classification models can be used in this pipeline to gain insight into their effectiveness with lie detection. A Random Forest Classifier was used initially, in tandem with previous papers such as [5] mentioned previously.

Logistic Regression is a statistical binary classification tool that applies a logistical function to a linear arrangement of input features. Doing so serves as a baseline for further models and is easily interpretable.

3) *Experimental FiLM and Graph Convolutional Networks:* The first experimental method integrates audio and visual data using speech and vision encoders combined with Feature-wise Linear Modulation (FiLM) [25]. In this approach, audio signals are processed using a speech encoder that combines CNNs and Transformers to extract a sequence of hidden vectors for classification [8]. This fusion approach enhances the system’s ability to capture nuanced interactions between audio and visual cues by gaining the capability to use visual reasoning and classification, which are critical for accurate lie detection [7].

The second method employs Graph Convolutional Networks (GCNs) focusing on scattered (spatially apart) features in the image by creating graphs from the features supplied. The processed audio features are integrated with the GCN and Transformer outputs using a suitable fusion method. This combination leverages the strengths of GCNs in handling complex graph structures and Transformers to long-range dependencies, making it highly effective for detailed facial movement analysis [9]. In this instance, specific iterations and spectral-based GCNs were used. First, Spectral-based GCNs take advantage of graph Laplacian’s eigenvalues and eigenvectors to define convolutions in the spectral domain, capturing global graph structures and relationships. Second, their usage for spectral graph theory provides a robust theoretical foundation for processing graph data, helping to optimize the model for detecting subtle patterns in multimodal data. Third, because the multimodal model integrates features from different sources (e.g., auditory and visual), spectral-based GCNs can facilitate feature integration by providing a global view of the graph structure. Although model training may be computationally expensive, the model was trained on a server to overcome this challenge, increasing the thorough nature of this approach [31]. This specific adaptation of the Graph Convolutional Networks is fully experimental.

IV. RESULTS

TABLE III
CLASSIFICATION REPORTS FOR DIFFERENT MODELS

Model	Class	Precision	Recall	F1-score
Random Forest	Deception	0.77	0.91	0.83
	Truthful	0.83	0.73	0.8
Logistic Regression	Deception	0.77	0.91	0.83
	Truthful	0.89	0.73	0.8
GCN	Deception	1	0.07	0.14
	Truthful	0.48	1	0.65
CNN conv1d	Deception	0.91	1	0.95
	Truthful	1	0.9	0.95

TABLE IV
TEST ACCURACY FOR EACH MODEL AFTER EACH FOLD

Model	1 st fold	2 nd fold	3 rd fold	4 th fold	5 th fold	Mean	Std
Random forest	0.59	0.8636	0.6818	0.619	0.8095	0.712	0.107
Logistic regression	0.772	0.7727	0.7727	0.7142	0.7142	0.7402	0.0269
CNN conv1d	0.909	0.909	1	0.95	1	0.954	0.04

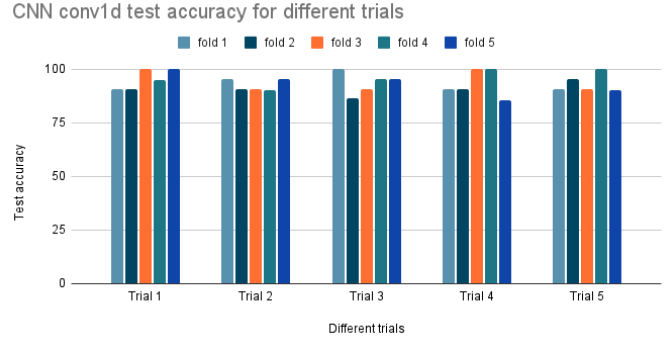


Fig. 9. Shows different trials for the best model (conv1d)

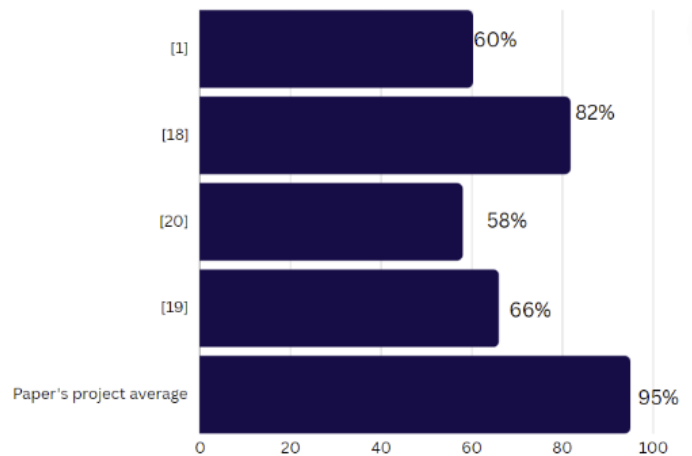


Fig. 10. shows the different accuracies of the previous papers

As discussed in the Methods section, this study examined multiple models and K-folds. Table IV shows the models used with the accuracy of each fold and the mean and standard deviation of all folds of each model. Table III presents the classification report of each model. Figure 9 shows different trials for the best model (conv1d).

Figures 10 and 9, along with tables III, IV and I, answer the first research question of the paper that is "How effective is the proposed AI model in detecting lies compared to traditional methods and some recent AI models?"

Figures 12, 13, and 11 answer the second research question of the paper "Which features carry the highest weights in prediction?"

V. DISCUSSION

A. Interpretation of the collected results

The following are the commonly used evaluation metrics in classification reports: Precision is the ratio of correctly predicted positive observations to the total number of predicted positives. The metric helps determine the true pinpointed accuracy of positive results as well as gauging false positives. Second, recall is the ratio of correctly predicted positive observations to all observations in an actual class. In the context of the task, it answers the question: "Of all the instances that were lies, how many were correctly predicted as a lie?" Finally, the F1-score is the harmonic mean of Precision and Recall. It provides a single metric that balances both concerns, especially when there is an imbalanced dataset in which one class is more prevalent.

The results of the conv1d model are presented in Tables III and IV. This demonstrate the classification report and the accuracy's mean and standard deviation, and the illustrated performance in each trial in Fig. 9, where each trial has five folds, ensuring the model's reliability and accuracy. For example, the classification report of conv1d showed no bias in the model, with a standard deviation of 0.04, which is relatively low, with a mean of 95.4%, which is a respected accuracy in the context of lie detection using AI. In contrast, more typical models such as Logistic Regression or Random Forest Classifiers have lower average precision. The F-1 scores of both were close to equal, with the lie class at 83% and the truth class at 80%. Furthermore, the spectral GCN model was not proven to be successful, as seen by the difference in precision, one class reached 100% whereas the other was significantly lower, hinting at an extreme bias towards a lie or a truth. However, when we tried CNN conv1d on the training set without manual annotation (only audio and visuals were used), average accuracy of the model was still 95+%.

B. The proposed model vs. previous studies

Fig. 10 shows a comparison of our model with those of previous studies. The figure contains the accuracy of different studies, considering that they may use the same data or different data. These figures help to answer the first research question of this paper: "How effective is the proposed AI model in detecting lies compared to traditional methods and some recent AI models?" For example, although the study [20] used a fusion of two different AI models, CNN and support vector machine, increasing the complexity and the need for high computational resources, our solution's best model used only conv1d and achieved higher accuracy. Its simplicity, along with its higher accuracy proves it to be a more effective model overall, especially in real-world situations.

C. Interpretation of the proposed model

Explainable AI (XAI) is used to interpret the model. For example, to explain the predictions of a machine learning model, SHAP (SHapley Additive exPlanations), a popular method for interpreting complex models, was used. It was used to generate a summary plot of the SHAP values, which

visually represented the impact of each feature on the model's predictions. The plot helps to understand which features are the most influential and how they contribute to the model's output. The output is shown in Fig. 11, illustrating that features 3925 and 4054 are the most important. These are the audio features. (Note: Because this requires a lot of computation, we used a subset (which contains both target classes) of the data for visualization.)

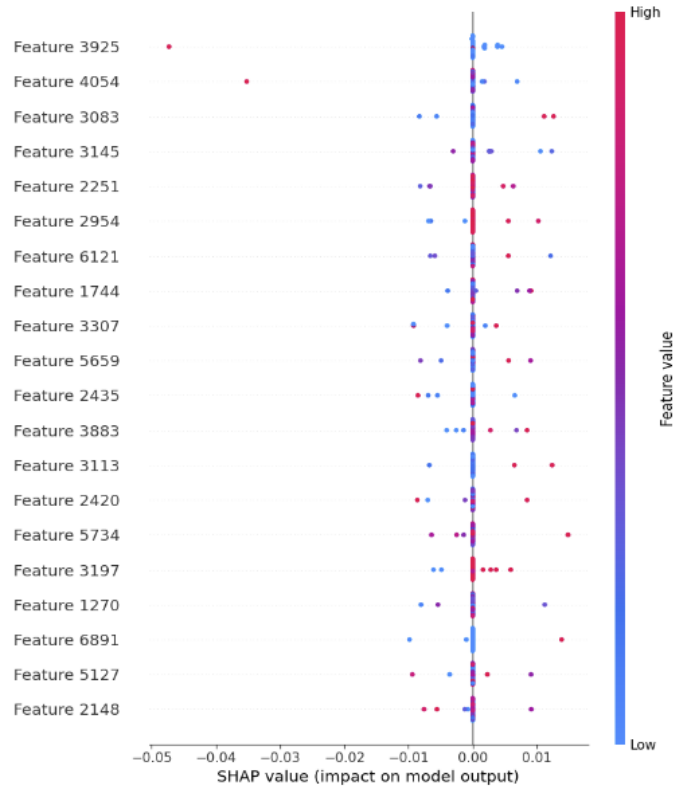


Fig. 11. illustrate the most and least significant features regarding the paper's model

The Local Interpretable Model-agnostic Explanations (LIME) framework, an XAI technique, explains the predictions of a machine learning model. LIME provides insight into why a model makes a specific prediction by approximating the model locally with an interpretable model. It provides a local interpretation of a specific prediction using a neural network model. By focusing on one instance and showing how different features contribute to the prediction, LIME helps to make complex models more understandable, especially in a classification context. We used two samples, one for deception (see Fig. 12) and one for truthful (see Fig. 13).

By analyzing Fig. 12, feature 949 had the highest positive contribution (1.57) towards the "Negative" (truthful) class, indicating that when this feature is at a higher value, the model is more confident that the instance is not associated with lying. On the other hand, features 692 and 6827 had significant negative contributions (-0.74 and -0.60, respectively), suggesting that when these features have lower values, the model is more

likely to classify the instance as "Negative" (truthful). The model is highly confident in predicting the "Negative" class (with a probability of 1.0), which is visualized by the zero probability for the "Positive" class. This suggests that a clear decision is made by the model based on a combination of feature values.

From Fig. 13, 3672, 2656, and 4835 features had values that strongly contributed to the "Positive" prediction, indicating that they were key indicators of deception according to the model. Features 1866 and 5882 contributed towards a "Negative" prediction (indicating truth) but were outweighed by the features supporting a "Positive" prediction.

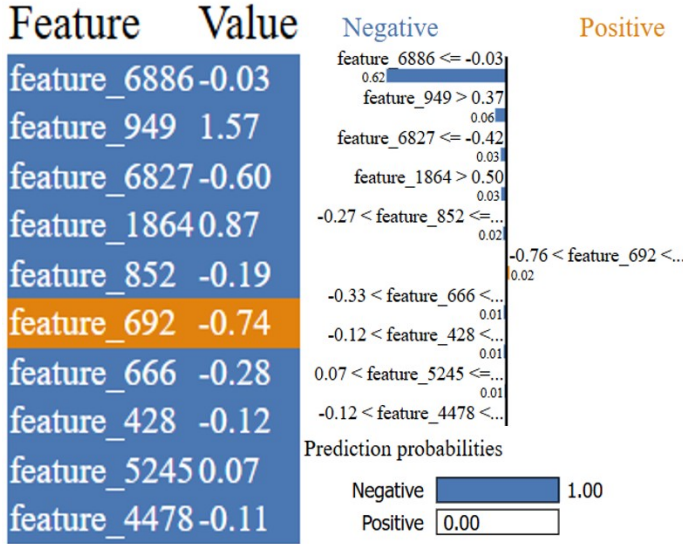


Fig. 12. LIME's output on a deceptive sample

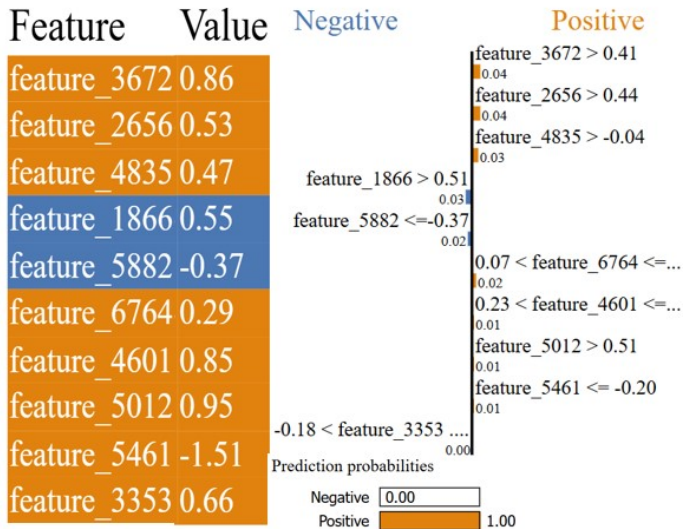


Fig. 13. LIME's output on a truthful sample

Features 3925, 4054, 949, 3672, 2656, 4835, 1866, and 5882 are related to audio. On the other hand, feature 6827 is

related to the visual extracted from the videos. CNN conv1d specializes in audio-related tasks; thus, it relays more audio features than visual features.

D. Limitations and Recommendations

The study had certain limitations that affected the results of this experiment. Primarily, the lack of high-quality datasets with both video and audio data was profound. The shortage of data in the Real-life Trial dataset may have decreased the overall accuracy of the model on the dataset. The dataset contained only 121 videos, of which 120 videos were used. Five K-folds were used to split the training and validation to avoid overfitting, but the small dataset still proved difficult when trying to train the data. Furthermore, as Mambreyan et al. (2022) showed, the Real-life Deception Dataset has significant gender bias that classifiers may exploit. Other large datasets usually have features that are manually annotated or have already been extracted for features. Expanding sampling to other scenarios, and diversifying subjects based on gender, ethnicity, and beliefs or ideologies could be vital to a more universal model. Finding these higher-quality datasets with more data samples would increase the amount of training data available and reduce social bias, thereby increasing the accuracy of the experiment.

However, bias appeared in the tests of the GCN model as stated above. Based on the data in Table III, it is evident that the GCN model shows a preference for class 1, as shown by its recall but low precision for class 1 and notably low recall for class 0, indicating extreme class bias. The spectral-based GCN showed much promise in the previous experiments but fell short, yet we encourage it to be experimented on further. To enhance the accuracy of the model, especially when utilizing the spectral GCN model, it will be beneficial to present more data, employ resampling methods, and provide higher computing power for those tasks.

VI. CONCLUSION

The results of this study highlight the critical role of multi-modal feature extraction techniques in advancing lie detection technologies. By leveraging audio features via OpenSmile, visual data through a Vision Transformer, and transcriptions of gestures and micro-expressions, the CNN Conv1D model achieved a high accuracy of 95.4 %, surpassing many state-of-the-art approaches. Even without manual transcriptions, the model performed admirably at 95 %, demonstrating the robustness of the architecture, particularly with a limited dataset. Both audio and visual features were vital to the performance of CNN Conv1D, and its success. Furthermore, the paper addressed the recommendation, which focuses on incorporating multi-modal data, mentioned by a study's [24] author. Despite these promising results, the limitations of this study, particularly the small and homogeneous dataset—underscore the necessity for further research. It is crucial to expand dataset diversity to include participants from various demographic groups and scenarios. Explainable AI (XAI) revealed that audio and visual features were the most significant contributors

to the model's decisions, indicating the importance of focusing on these modalities in future studies. However, integrating additional modalities such as thermal imaging, heart rate monitoring, and moisture tracking could further enhance model performance, especially in complex real-world applications. Addressing these challenges will not only improve model generalizability but also help mitigate ethical concerns related to biases in facial micro-expression recognition across different racial and gender groups. Future work should explore ablation studies and alternative architectures to deepen our understanding of how multimodal learning can be optimized. By continuing to build on this research, we move closer to creating an accurate, reliable, and ethical alternative to traditional polygraph tests, with potential applications in criminal justice and law enforcement.

REFERENCES

- [1] Li, J., Wang, T., & Wang, S.-J. (2022). Facial Micro-Expression Recognition Based on Deep Local-Holistic Network. *Applied Sciences*, 12(9), 4643. <https://doi.org/10.3390/app12094643>
- [2] Merghani, W., Davison, A. K., & Yap, M. H. (2018). A review on facial micro-expressions analysis: datasets, features and metrics. arXiv preprint arXiv:1805.02397.
- [3] Morgan, Tony Duncan. "How often do people lie?" UW-La Crosse, <https://www.uwlax.edu/currents/how-often-do-people-lie/>
- [4] Mahon, J. E. (2007). A definition of deceiving. *International Journal of Applied Philosophy*, 21(2), 181–194.
- [5] Soldner, F., Pérez-Rosas, V., & Mihalcea, R. (2019). Box of lies: Multimodal deception detection in dialogues. *Proceedings of the 2019 Conference of the North*. <https://doi.org/10.18653/v1/n19-1175>
- [6] Mohamed, Z., et al. "IEEETrans2022.pdf." *University of Michigan*, 1 Mar. 2022, <https://public.websites.umich.edu/~zmohamed/PDFs/IEEETrans2022.pdf>
- [7] Sun, B., & Wu, H. (2021). Deep Learning for Micro-expression Recognition: A Survey. arXiv. <https://arxiv.org/abs/2107.02823>
- [8] Reddy, S., Karri, S. T., Dubey, S. R., & Mukherjee, S. (2019). Spontaneous facial micro-expression recognition using 3D spatiotemporal convolutional neural networks. *SpringerLink*. <https://doi.org/10.1007/s00138-019-01056-1>
- [9] Liu, Y., Zhang, H., Zhao, Y., Chen, F., Mi, B., Zhou, J., Chen, Y., Wang, D., & Pei, L. (2021). Geographical variations in maternal dietary patterns during pregnancy associated with birth weight in Shaanxi province, Northwestern China. *PLOS ONE*, 16(7), e0254891. <https://doi.org/10.1371/journal.pone.0254891>
- [10] Sánchez-Monedero, Javier, and Lina Dencik. "The politics of deceptive borders: 'biomarkers of deceit' and the case of iBorderCtrl". School of Journalism, Media and Culture, Cardiff University, Cardiff, United Kingdom, <https://arxiv.org/pdf/1911.09156>
- [11] Chen, Jia, et al. "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing." arXiv, 26 Oct. 2021, <https://arxiv.org/pdf/2110.13900>
- [12] Perez, Ethan, et al. "FiLM: Visual Reasoning with a General Conditioning Layer." arXiv, 22 Sept. 2017, <https://arxiv.org/abs/1709.07871>
- [13] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." arXiv, 18 May 2015, <https://arXiv:1505.04597>
- [14] McNeely-White, D., Beveridge, J. R., Draper, B. A. (2020). Inception and ResNet features are (almost) equivalent. *Cognitive Systems Research*, 59, 312–318. <https://doi.org/10.1016/j.cogsys.2019.10.004>
- [15] GeeksforGeeks. (2023, January 10). Residual networks (resnet) - deep learning. <https://www.geeksforgeeks.org/residual-networks-resnet-deep-learning/> <https://doi.org/10.1016/j.cogsys.2019.10.004>
- [16] Zhang, G., Zhang, H., Yao, Y., & Shen, Q. (2022). Attention-guided feature extraction and multiscale feature fusion 3D resnet for automated pulmonary nodule detection. *IEEE Access*, 10, 61530–61543. <https://doi.org/10.1109/access.2022.3182104>
- [17] GeeksforGeeks. (2024, January 10). Explanation of Bert Model - NLP. <https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/>
- [18] Feng, J. (2021). DeepLie: Detect Lies with Facial Expression (Computer Vision). Retrieved from <https://www.example.com>
- [19] Guo, X., Selvaraj, N. M., Yu, Z., Kong, A. W.-K., Shen, B., I& Kot, A. (2023). Audio-Visual Deception Detection: DOLOS Dataset and Parameter-Efficient Crossmodal Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, Oct. 2023. Retrieved from https://openaccess.thecvf.com/content/ICCV2023/papers/Guo_Audio-Visual_Deception_Detection_DOLOS_Dataset_and_Parameter-Efficient_Crossmodal_Learning_ICCV_2023_paper.pdf
- [20] Dhabarde, R., Kodawade, D., I& Zalte-Gaikwad, S. (2023). Hybrid Machine Learning Model for Lie-Detection. In *2023 IEEE International Conference on Communication, Information and Computing Technology (ICCICT)*, pp. 1-5. doi:10.1109/I2CT57861.2023.10126460. Retrieved from https://www.researchgate.net/publication/369225522_Hybrid_Machine_Learning_Model_for_Lie-Detection
- [21] Ding, M., Zhao, A., Lu, Z., Xiang, T., I& Wen, J. (2019). Face-Focused Cross-Stream Network for Deception Detection in Videos. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 7794-7803. <https://doi.org/10.1109/CVPR.2019.00799>
- [22] Camara, M. K., Postal, A., Maul, T. H., I& Paetzold, G. H. (2024). Can lies be faked? Comparing low-stakes and high-stakes deception video datasets from a Machine Learning perspective. *Expert Systems with Applications*, Volume 249, Part C. <https://doi.org/10.1016/j.eswa.2024.123684>
- [23] Galinsky, A., I& Schweitzer, M. (2022, February 25). Recognizing deception: How to spot a lie. Wharton Executive Education. <https://executiveeducation.wharton.upenn.edu/thought-leadership/wharton-at-work/2021/08/recognizing-deception/#:~:text=Watch%20for%20inappropriate%2C%20unusual%2C%20or%20uncommon%20behavior.&text=Also%20watch%20for%20common%20liars,is%20serious%2C%20for%20example>
- [24] Dhabarde, R., Kodawade, D., & Zalte-Gaikwad, S. (2023). Hybrid Machine Learning Model for Lie-Detection. https://www.researchgate.net/publication/371002374_Hybrid_Machine_Learning_Model_for_Lie-Detection
- [25] Perez, Ethan, et al. "FiLM: Visual Reasoning with a General Conditioning Layer." arXiv, 22 Sept. 2017, arxiv.org/abs/1709.07871.
- [26] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. [arXiv.org/abs/2010.11929](https://arxiv.org/abs/2010.11929)
- [27] Schröder, Marc, The SEMAINE API: Towards a Standards-Based Framework for Building Emotion-Oriented Systems, *Advances in Human-Computer Interaction*, 2010, 319406, 21 pages, 2010. <https://doi.org/10.1155/2010/319406>
- [28] Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the Munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM International Conference on Multimedia (MM '10)*. Association for Computing Machinery, New York, NY, USA, 1459–1462. 2010. <https://doi.org/10.1145/1873951.1874246>
- [29] Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., & Darrell, T. (2014). Long-term Recurrent Convolutional Networks for Visual Recognition and Description. [ArXiv. /abs/1411.4389](https://arxiv.org/abs/1411.4389)
- [30] Breiman, L. Random Forests. *Machine Learning* 45, 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
- [31] Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph Convolutional Networks: A comprehensive review. *Computational Social Networks*, 6(1). <https://doi.org/10.1186/s40649-019-0069-y>
- [32] Saidi, A., Ben Othman, S., Dhoubi, M., & Ben Saoud, S. (2021). FPGA-based implementation of classification techniques: A survey. *Integration*, 81, 280–299. <https://doi.org/10.1016/j.vlsi.2021.08.004>
- [33] Yang, X.-S. (2019). Logistic Regression, PCA, Lda, and ica. *Introduction to Algorithms for Data Mining and Machine Learning*, 91–108. <https://doi.org/10.1016/b978-0-12-817216-2.00012-0>
- [34] GeeksforGeeks. (2024b, June 21). Graph Convolutional Networks (gcn): Architectural Insights and Applications. <https://www.geeksforgeeks.org/graph-convolutional-networks-gcn-architectural-insights-and-applications/>
- [35] Ige, A. O., & Sibiya, M. (2024). State-of-the-art in 1d Convolutional Neural Networks: A survey. *IEEE Access*, 1–1. <https://doi.org/10.1109/access.2024.3433513>
- [36] S. Kiranyaz, T. Ince, R. Hamila, and M. Gabbouj, "Convolutional Neural Networks for patient-specific ECG classification," in 2015

37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015, pp. 2608–2611, doi: 10.1109/EMBC.2015.7318926.

APPENDIX

You can find the code for the Multi-modal Lie Detection Project at the following GitHub repository:

<https://github.com/AbdelrahmanAbdelwahab1/Multi-modal-Lie-detection-project>