DECEPTION DETECTION FROM SPEECH

Tien Nguyen
Data Science

Advisor : Prof. Faranak Abri
Committee Member : Prof. Nada Attar, Prof. Fabio Di Troia

San Jose State University
Spring 2024

Overview of Presentation

Introduction

- **Problem:** Deception can harm individuals and society, making detection crucial.

- **Traditional Methods:** Polygraph tests have limitations.

- **Modern Approach:** Machine learning and deep learning offer new ways to detect deception using speech data.

Project Overview

Goal: Develop models using textual and audio data to detect deceptive contexts.

Contributions:

- Feature extraction from textual data
- Feature selection techniques
- Conventional and deep models for textual data
- Deep models for audio data
- Late fusion technique
- Testing on a real-life dataset

Related Work

Dataset

1

Real-Life Scenarios:

- Court Trials (e.g., 61 deceptive, 60 truthful videos, avg. 28 seconds each) [1],[2],[3],[4]
- Political Debates (claims labeled true, half-true, false) [5]

2

Staged Scenarios:

- Actors prompted with questions designed to elicit deception [6]
- CXD corpus (deceptive/non-deceptive speech in English & Mandarin) [7]

3

Game-Based Scenarios:

- "Werewolf Killing" game (liars vs. honest characters) [8]
- "Werewolves of Miller's Hollow" competitions (clips with deception) [9]

Conventional Models

- Wawer et al. utilized a **Support Vector Machine (SVM)** and **XGBoost** to detect deception [10]
- Bareeda et al. developed **SVM** classifiers with Gaussian and polynomial kernels [2].
- Tao et al. employed **SVM** to analyze normalized acoustic features for deception [8].
- Sen et al. experimented with multimodal features (verbal, acoustic, visual) in deception detection using **SVM** and **Random Forest (RF)** [1].
- Chebbi et al. developed **K-nearest neighbor (KNN)** models for individual modalities with selected features and integrated them using decision-level fusion [11].

Deep Models

- Sehrawat et al. proposed a multimodal approach to detect deception using **LSTM**, **BiLSTM**, **CNN**, and **ResNet50** [4].

- Marcolla et al. created their dataset and employed MFCC features extracted via Librosa for audio data, which were processed using an **LSTM** model [12].

- Hsiao and Sun used **BiLSTM** to integrate textual and visual data [3].

- Antolín and Montero developed a model leveraging attention **LSTM** mechanisms, using gaze and speech features from the Gazepoint GP3 Eye Tracker and LibROSA for speech [13].

Dataset

- **Source**: Public court trials (University of Michigan) [1]
- **Videos**: 121 total (61 deceptive, 60 truthful)
- **Average Length**: Approximately 28 seconds
- **Content**: Defendants or witnesses speaking

Additional Data: Transcripts of each video and Gesture annotations (smile, laugh, scowl)

Deceptive	Truthful
He had told me that he had had a dream that, ammm ...he was in a forest and that he had killed Laura, and that if I didn't help him get rid of her, that he ... that I was gonna be next.	All of us, who have represented people for years in the system get letters from prisioners um ... and their families. You know, this person is improperly convicted, you need to do something about it. But, its no one else's job to do it, other than the innocence project .... and they do it, they do itelse marvelously.

Model Evaluation

- **Evaluation Method**: 5-fold Cross-Validation
  - + The dataset is split into 5 subsets.
  - + Iterative training and testing on different folds.
  - + Ensures robustness and reduces overfitting.
- **Performance Metrics**: Accuracy and F1 Score

True Positives (TP): Correctly classified deceptions.
False Positives (FP): Incorrectly classified truths as deceptions.
True Negatives (TN): Correctly classified truths.
False Negatives (FN): Incorrectly classified deceptions as truths.

Accuracy: Proportion of correctly predicted instances.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

F1 Score: Harmonic mean of Precision and Recall.

+ Precision: Proportion of true positives among predicted positives.
+ Recall: Proportion of true positives identified correctly.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$\textbf{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Textual Models

Text Preprocessing

Step 1: Removing Non-alphabetic Characters like punctuation and special characters.

Step 2 (for Conventional Models): Feature Extraction

Step 2 (for Deep Learning Models): Stemming

- Reduces words to their root form (e.g., "running" -> "run").

Step 3: One-Hot Encoding

Step 4: Padding

- Sequences are padded with zeros to a fixed length (221 words).

Features Extraction

Feature Name	Description
Word Count	The total number of words in the text
Sentence Count	The total number of sentences in the text.
Sentiment Score	A numerical score indicating the overall sentiment of the text
Average Word Length	The average length of words in the text.
Vocabulary Diversity	The ratio of unique words to the total number of words in the text.
Adjective Frequency	The proportion of adjectives in the text
Adverb Frequency	The proportion of adverbs in the text.

Feature Name	Description
Conjunction Frequency	The proportion of conjunctions .
Negation Count	The number of negations in the text.
Repetition Count	The proportion of words that appear more than once in the text.
Self-Reference Count	The number of self-referential words in the text (e.g., ”I,” ”me,” ”myself”).
Filler Word Count	The number of common filler words ('um', 'uh', 'hmm' or 'like',)
Pronoun Frequency	The proportion of pronouns in the text
Past/Present/Future Tense Frequency	The proportion of verbs in the past/present/future tense in the text.

Feature Selection

Overlapping Probability Density Functions (OVL):

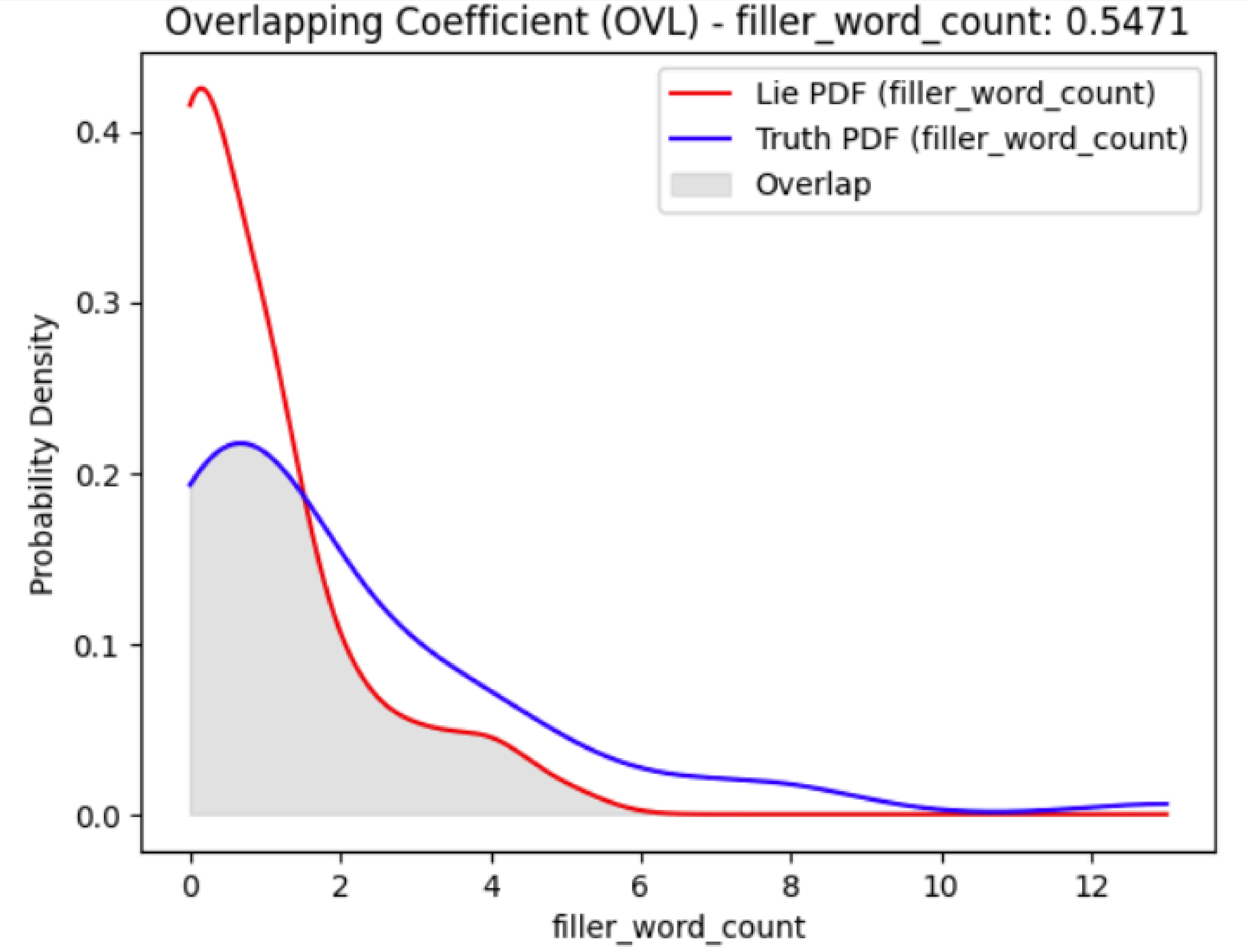
- Compares probability density functions (PDFs) of features for deceptive vs. truthful categories.
- Low OVL indicates a feature effectively separates categories (minimal PDF overlap).
- High OVL suggests the feature may not be useful (significant PDF overlap).

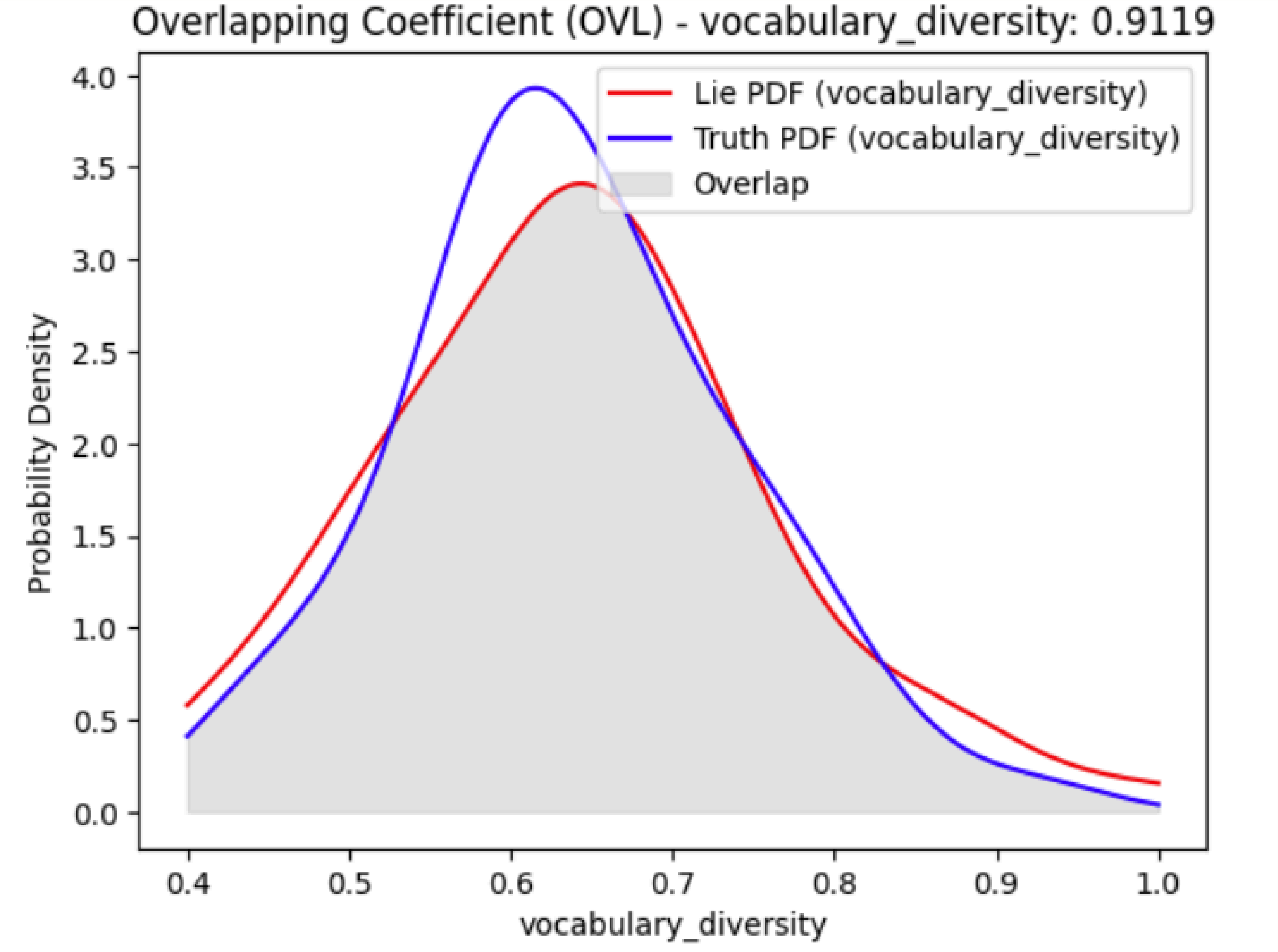
$$OVL = \int_{Rn} min(f(x), g(x))dx$$

where f(x) and g(x): two real probability density functions

Rn : n-dimensional space of real numbers [17]

Overlapping Probability Density Functions (OVL):





Overlapping Probability Density Functions (OVL):

Feature	OVL Score
Filter word count	0.5471
Future tense frequency	0.5517
Negation count	0.6097
Adverb frequency	0.7367
Present tense frequency	0.7512
Sentence count	0.7811

Feature	OVL Score
Pronoun frequency	0.8299
Past tense frequency	0.8345
Avg word length	0.8479
Repetition count	0.8497
Conjunction frequency	0.9001
Vocabulary diversity	0.9119

Feature Selection

Stepwise Regression

- Iterative feature selection technique for machine learning.
- Start with an empty feature set.
- Evaluate each feature's impact on model performance (e.g., accuracy, F1 score).
- Features are iteratively added or removed based on their contribution.
- Stop when a predefined criteria is met (e.g., optimal model performance).

- Average word length
- Vocabulary diversity
- Adjective frequency
- Adverb frequency and
- Filler word count.

Conventional Models

Models Used:

- Support Vector Machine (SVM) - Model 1
- K-Nearest Neighbors (KNN) - Model 2
- Logistic Regression (LR) - Model 3

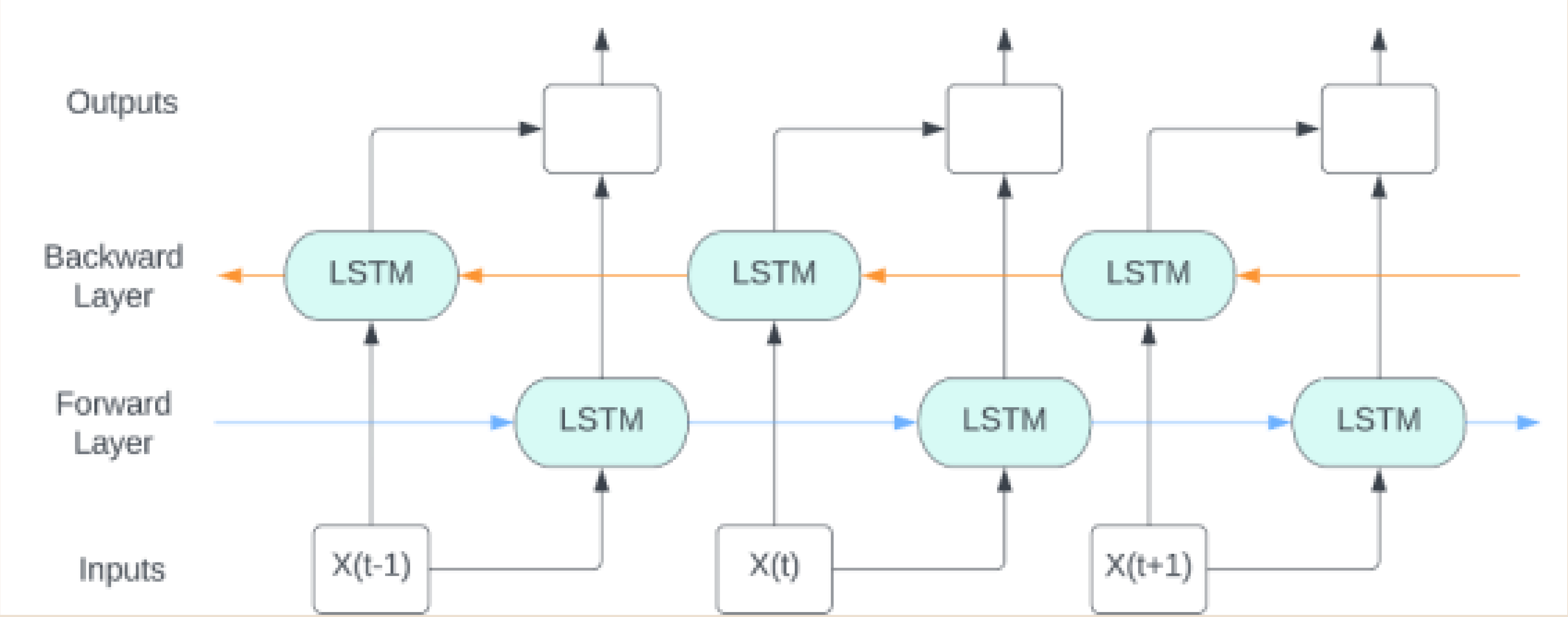
Hyperparameter Tuning:

- Grid Search technique for model optimization
- Improved model performance and predictive power

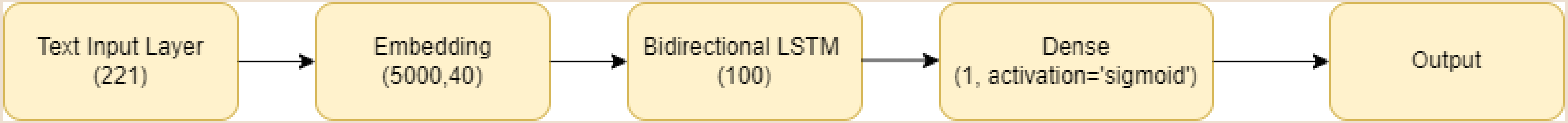
Result of Conventional Models

Model	Train Accuaracy	Test Accuracy	F1 Score
Model 1a: SVM + OVL	61.12	59.37	70.44
Model 1b: SMV + Stepwise	64.46	63.77	69.8
Model 2a: KNN + OVL	70.65	63.6	65.47
Model 2b: KNN + Stepwise	71.69	62.83	63.07
Model 3a: LR + OVL	66.12	63.63	67.89
Model 3b: LR + Stepwise	66.11	**68.53**	**71.69**

Deep Model- Bidirectional LSTM



The Architecture of a BiLSTM block [4]



- Model 4: 1 BiLSTM
- Model 5: 1 BiLSTM + Dropout Layer
- Model 6: 1 BiLSTM + Early Stopping

In addition, other pre-train models: Bert, GPT-2, Roberta

Results

Model	Train Accuaracy	Test Accuracy	F1 Score
Model 4: 1 BiLSTM	100	67.73	69.83
Model 5: 1 BiLSTM + Dropout	100	66.9	66.18
Model 6: 1 BiLSTM + Early Stopping	**100**	**93.57**	**94.48**
Model 7: Bert + Early Stopping + Dropout	83.54	68.73	64.63
Model 8: Pretrained GPT2	99.79	58.73	60.12
Model 9: Pretrained Roberta	88.18	71.2	73.71

Got accepted from COMPSAC 2024

Audio Models

Preprocessing

Step 1: Convert MP4 video files to WAV audio format

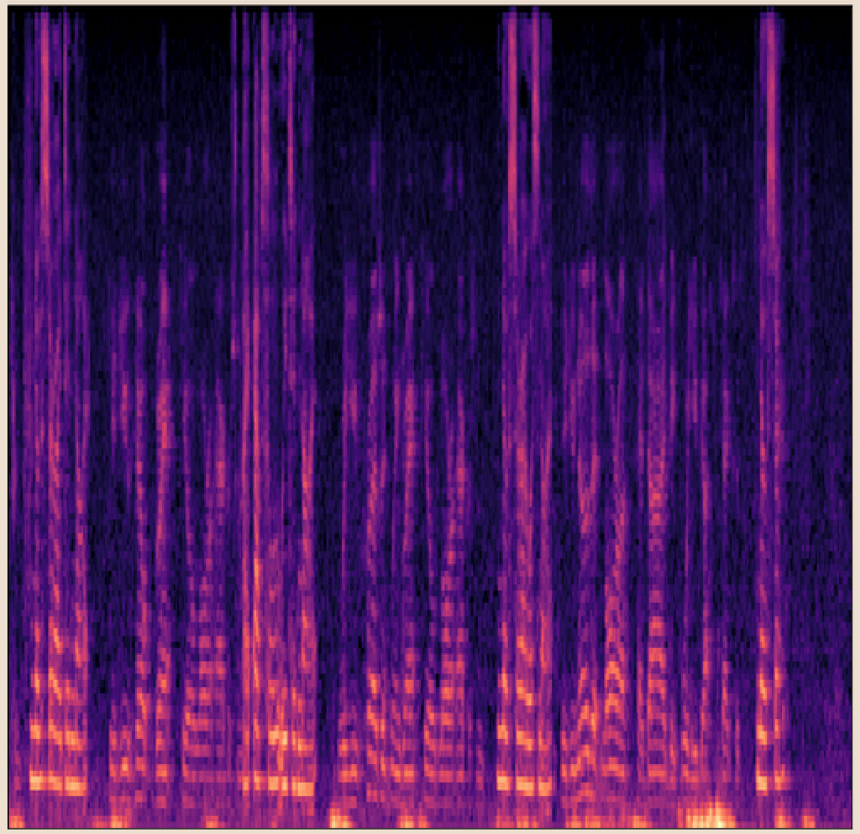
Step 2: Transform raw audio into Mel Spectrograms

- Mel spectrograms are visual representations of sound that consider human auditory perception.
- Use Librosa library

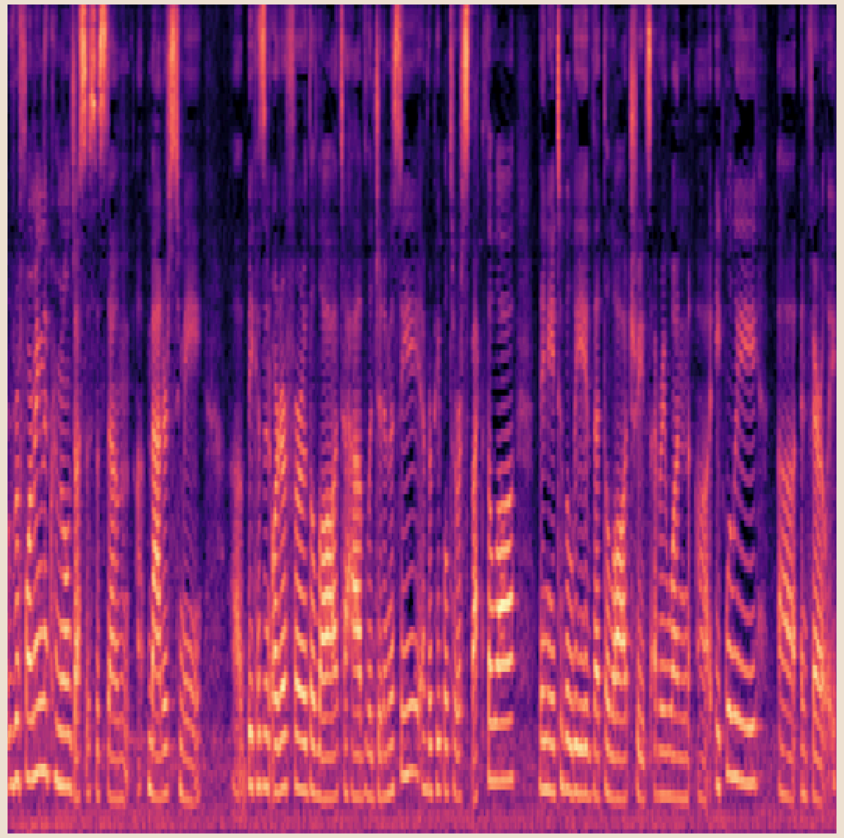
Step 3: Convert to RGB and Resize

- Convert the Mel spectrogram from grayscale to RGB color space
- Resize spectrograms to a uniform dimension (224x224 pixels) for consistent input size

Deceptive Image Example



Truthful Image Example



Residual Network 50 (ResNet50)

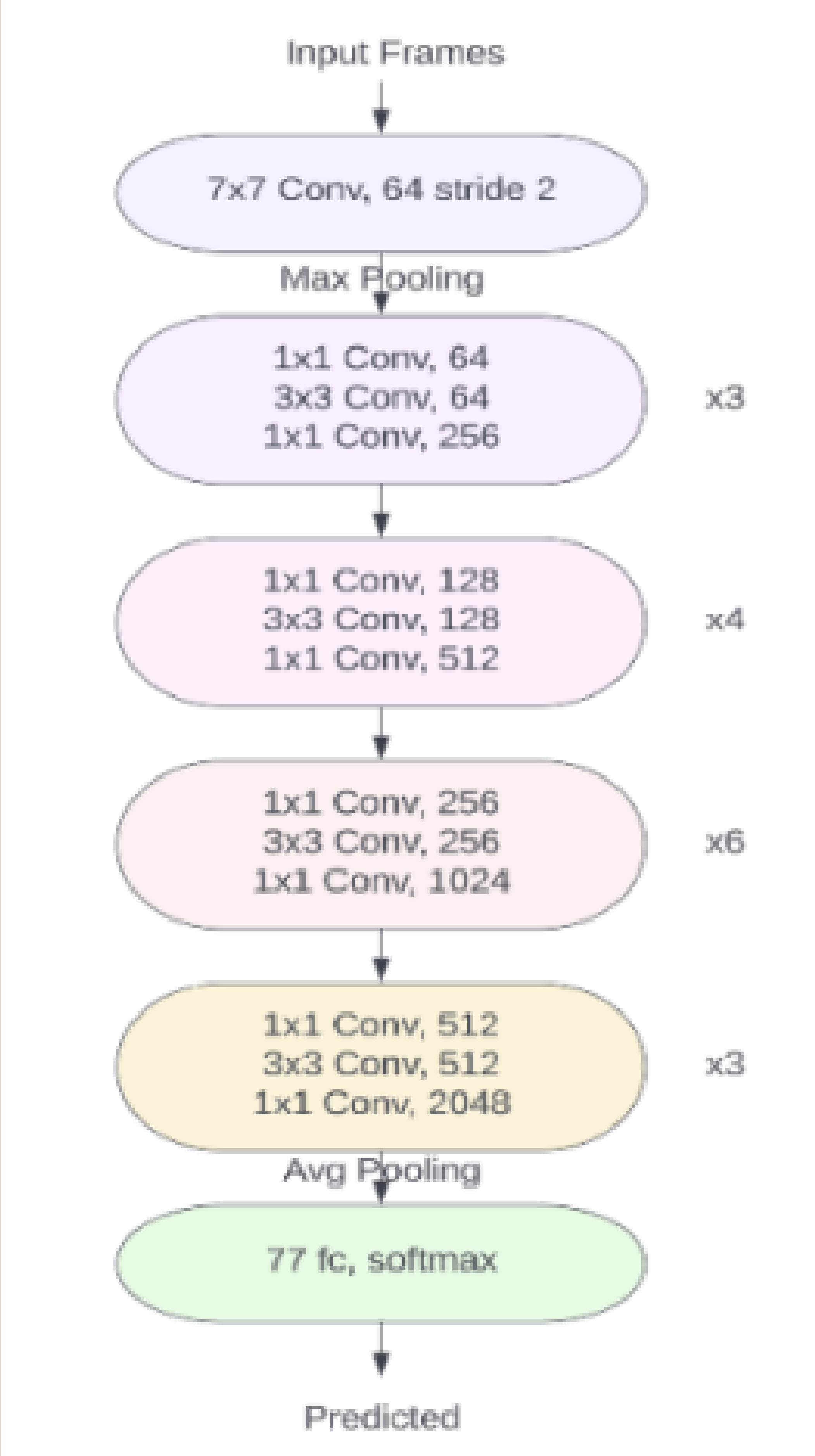
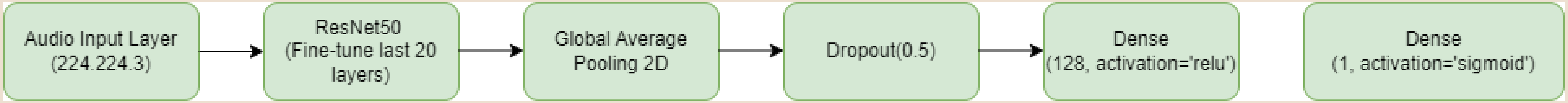


Figure: Architecture of a RetNes50 [4]

Residual Network 50 (ResNet50)



Regularization Techniques:

- Global Average Pooling 2D layer for dimensionality reduction.
- 0.5 Dropout layer to prevent overfitting.
- Early Stopping to halt training when validation performance plateaus.

Optimization:

- Adam optimizer (learning rate 0.0001): Efficient optimization for training.
- Binary cross-entropy loss: Suitable for binary classification tasks.

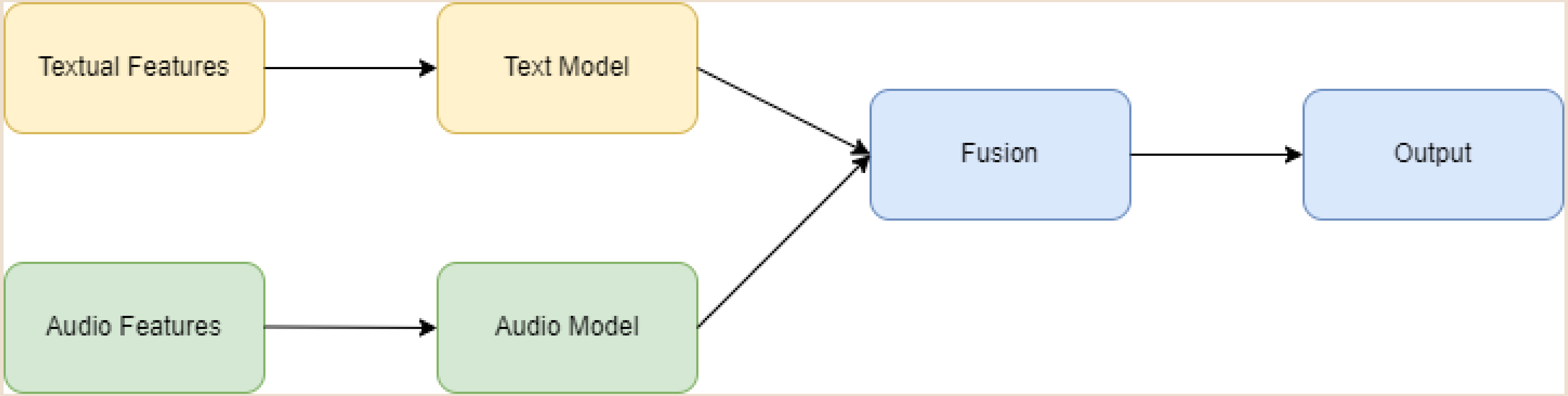
Results

Model	Train Accuaracy	Test Accuracy	F1 Score
Model 10: ResNet50	96.04	**93.57**	**92.16**
Model 11: VGG16	96.01	87.63	89.25

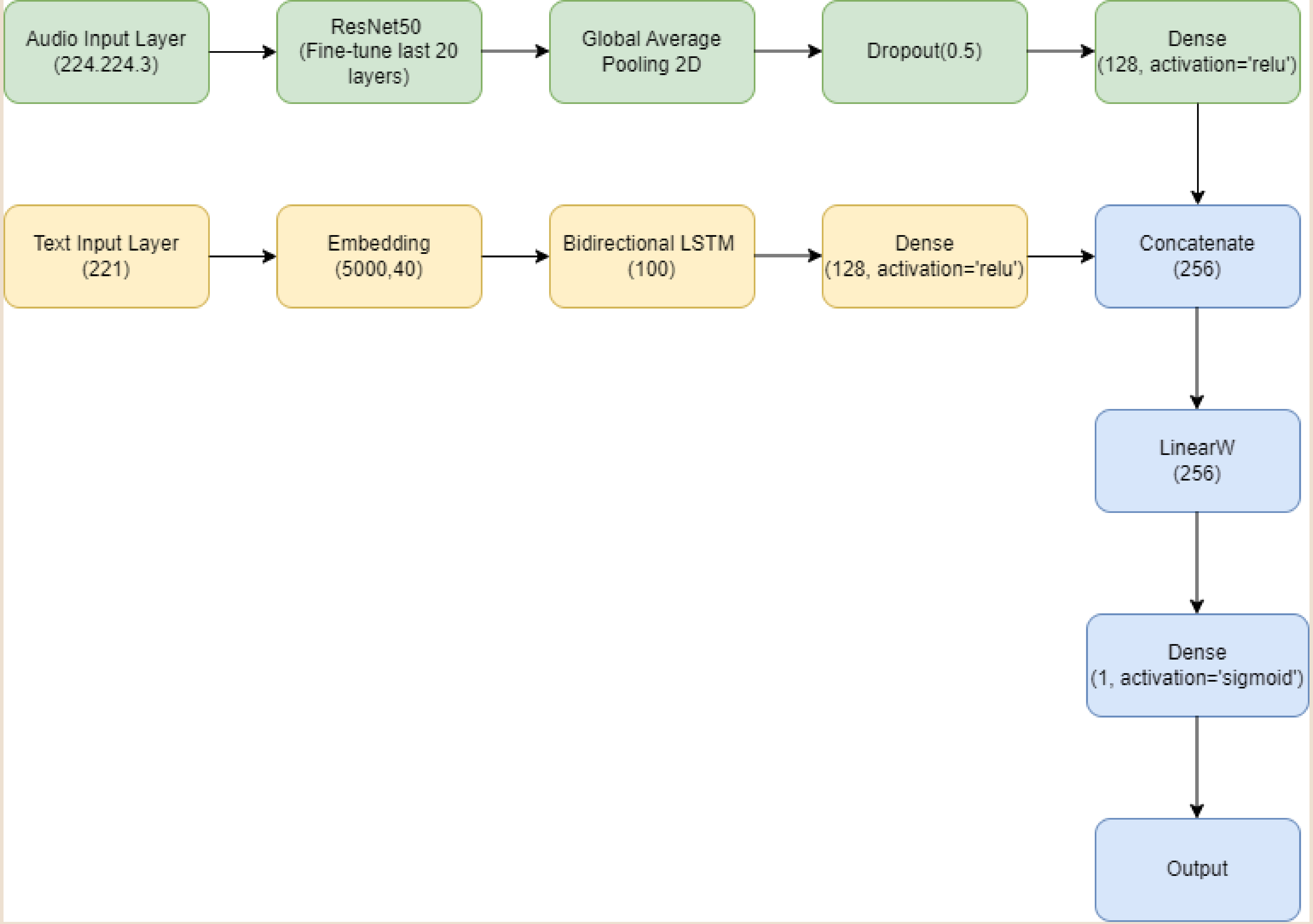
- Research by Sehrawat et al. [4] aligns with our findings, indicating ResNet50's suitability for audio data.
- ResNet50 also offered a significant advantage in terms of training time (approximately 8 hours faster).

Late Fusion Models

Architecture



- **Combining Audio and Text Modalities:** Leverage information from both audio and textual cues [1].
- **Separate Neural Network Architectures:** Individual models process audio and text data.
- **Late Fusion Approach:** Combines model outputs into a single vector [4].
- **Fusion Layer:** + Learns optimal weights for each model's contribution (trained weights).
  + Softmax function ensures weights sum to 1 (probabilistic interpretation).



The Architecture of Late Fusion [4]

Results

Fold	Audio Weight	Textual Weight	Test Accuracy	F1 Score
1	0.49731752	0.50268245	92	90.9
2	0.5032117	0.49678826	95.83	96
3	0.516298	0.4837021	91.67	88.89
4	0.5161787	0.48382124	**95.83**	**96.77**
5	0.49938968	0.5006103	79.12	82.76
Average	0.5064791	0.4935209	90.9	91.07

Comparison with Previous Work

Model	Test Accuracy	F1 Score
Late Fusion Model	**90.9**	**91.07**
Sehrawat et al. [4]	80	xxx
Zhang [18]	84.40	70.80

Conclusion
&
Future Work

Conclusion

- **Textual Feature Extraction:** Identified significant features using Stepwise and OVL methods.
- **Textual Model Performance:**
  - + LR achieved the highest accuracy (68.53%) among conventional models.
  - + Deep model with BiLSTM outperformed all textual models (93.57% accuracy, 94.48% F1)
- **Audio Model Performance**: The ResNet50 model achieved 93.57% accuracy and 92.16% F1
- **Late Fusion Model**: achieved 90.9% accuracy and 91.07% F1

Future Work:

- Extract new features from video data.
- Develop improved models combining video, audio, and textual features.
- Utilize larger datasets to enhance robustness and generalizability.

Reference

[1] M. U. S¸en, V. Perez-Rosas, B. Yanikoglu, M. Abouelenien, M. Burzo, and R. Mihalcea, “Multimodal deception detection using real-life trial data,” IEEE Transactions on Affective Computing, vol. 13, no. 1, pp. 306–319, 2020

[2] E. F. Bareeda, B. S. Mohan, and K. A. Muneer, “Lie detection using speech processing techniques,” in Journal of Physics: Conference Series, vol. 1921, no. 1. IOP Publishing, 2021, p. 012028.

[3] S.-W. Hsiao and C.-Y. Sun, “Attention-aware multi-modal rnn for deception detection,” in 2022 IEEE International Conference on Big Data (Big Data). IEEE, 2022, pp. 3593–3596

[4] P. K. Sehrawat, R. Kumar, N. Kumar, and D. K. Vishwakarma, “Deception detection using a multimodal stacked bi-lstm model,” in 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA). IEEE, 2023, pp.318–326

[5] D. Kopev, A. Ali, I. Koychev, and P. Nakov, “Detecting deception in political debates using acoustic and textual features,” in 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2019, pp. 652–659

[6] Sarzynska-Wawer, Justyna, et al. "Truth or lie: Exploring the language of deception." Plos one 18.2 (2023): e0281179.

[7] G. Mendels, S. I. Levitan, K.-Z. Lee, and J. Hirschberg, “Hybrid acoustic-lexical deep learning approach for deception detection.” in Interspeech, 2017, pp. 1472–1476.

[8] H. Tao, P. Lei, M. Wang, J. Wang, and H. Fu, “Speech deception detection algorithm based on svm and acoustic features,” in 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT). IEEE, 2019, pp. 31–33.

Reference

[9] H. Fu, H. Yu, X. Wang, X. Lu, and C. Zhu, “A semi-supervised speech deception detection algorithm combining acoustic statistical features and time-frequency two dimensional features,” Brain Sciences, vol. 13, no. 5, p. 725, 2023.

[10] J. Sarzynska-Wawer, A. Pawlak, J. Szymanowska, K. Hanusz, and A. Wawer, “Truth or lie: Exploring the language of deception,” Plos one, vol. 18, no. 2, p. e0281179, 2023

[11] S. Chebbi and S. B. Jebara, “Deception detection using multimodal fusion approaches,” Multimedia Tools and Applications, pp. 1–30, 2021.

[12] F. M. Marcolla, R. de Santiago, and R. L. Dazzi, “Novel lie speech classification by using voice stress.” in ICAART (2), 2020, pp. 742–749.

[13] A. Gallardo-Antolı́ın and J. M. Montero, “Detecting deception from gaze and speech using a multimodal attention lstm-based framework,” Applied Sciences, vol. 11, no. 14, p. 6393, 2021.

14] G. Van Houdt, C. Mosquera, and G. Ń apoles, “A review on the long short-term memory model,” Artificial Intelligence Review, vol. 53, no. 8, pp. 5929–5955, 2020.

[15] “Bidirectional lstm in nlp.” [Online]. Available: https://www.geeksforgeeks.org/bidirectional-lstm-in-nlp/

[16] “Introduction to Convolution Neural Network” [Online]. Available https://www.geeksforgeeks.org/introduction-convolution-neural-network/

Reference

[17] Namin, Akbar Siami, et al. "Linguistic features for detecting fake reviews." (2020)

[18] H. Zhang, Y. Ding, L. Cao, X. Wang, and L. Feng, “Fine-grained question-level deception detection via graph-based learning and cross-modal fusion,” IEEE Transactions on Information Forensics and Security, vol. 17, pp. 2452–2467, 2022

THANK YOU

BERT

Overview of BERT:

- Advanced pre-trained model developed by Google.
- Built on the Transformer architecture, which focuses on self-attention mechanisms.
- Processes words in full context relative to all other words in a sentence.

Core Features:

- Bidirectional Processing: Gathers context from both sides of each word simultaneously.
- Transformer Blocks: Comprised of layers that compute attention scores, vital for determining word relevance and context.

BERT

BERT Configuration for Project:

- Model Variant: bert-base-uncased – A general-purpose BERT model that is case insensitive.
- TFBertForSequenceClassification: Adapted for binary classification tasks

Fine-Tuning Process:

- Initialization: Utilizes TFBertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=2) to load the pre-trained model with a tailored output layer.
- Compilation: Employs the Adam optimizer and Sparse Categorical Crossentropy loss, which is perfect for logit handling.
- Training Regime: Conducted over 15 epochs, adjusting weights and biases to optimize for deception detection based on training data insights.

Robustly Optimized BERT Approach (RoBERTa)

Masked Language Modeling:

- RoBERTa is trained using a technique where random words in a sentence are hidden (masked), and the model learns to predict these hidden words. This helps it understand language deeply.

Bidirectional Context:

- RoBERTa reads text from both left to right and right to left, gaining a fuller understanding of each word's context.

Large Scale Training: It learns from a massive amount of text, far more than a human could read in their lifetime, which helps it grasp a wide variety of linguistic nuances and styles.

GPT-2

Unsupervised Learning:

- GPT-2 learns by reading a vast amount of text without specific instructions on what to learn. It simply looks for patterns and structures in how words are put together.

Generative Model:

- GPT-2 doesn’t just understand text; it generates new text based on the patterns seen during training. It’s like writing a novel where each new sentence needs to follow logically from the last.

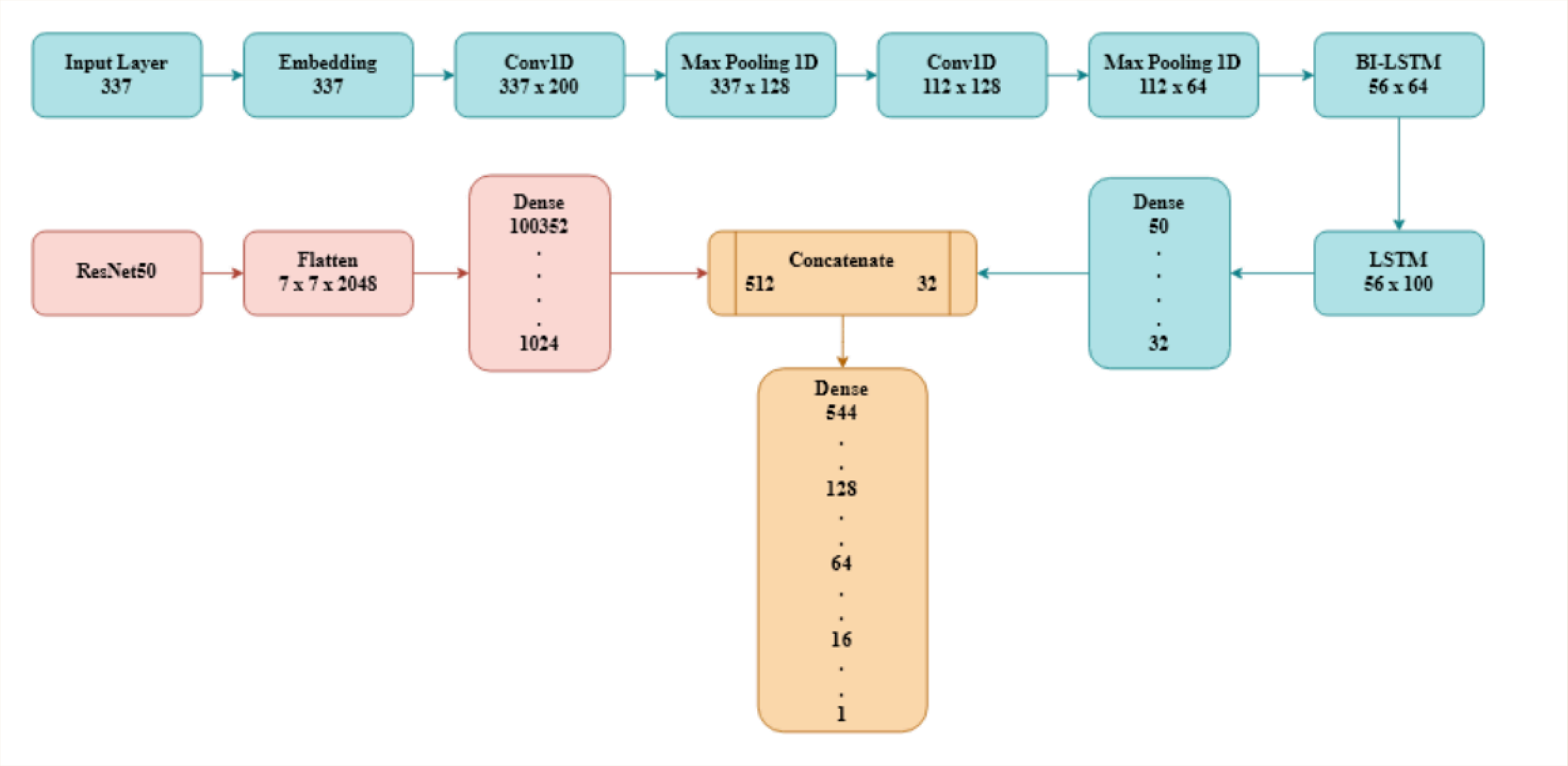
Attention Mechanism:

- At the core of GPT-2 is the transformer architecture, which uses an "attention mechanism." => Focus on different parts of a sentence as it is written, ensuring that each word it adds makes sense in the current context.

TABLE 4
Individual Feature Performance: Accuracy (%) and AUC Scores

Feature Set (dimension)	SVM		RF		NN	
	Accuracy	AUC	Accuracy	AUC	Accuracy	AUC
Visual						
Facial displays (32)	76.27 $\pm$ 0.00	0.8581	76.27 $\pm$ 1.69	0.9270	**80.79** $\pm$ 0.98	0.9416
Hand gestures (7)	50.28 $\pm$ 3.53	0.7232	**64.97** $\pm$ 3.91	0.6671	61.58 $\pm$ 0.98	0.6930
All visual (39)	58.19 $\pm$ 0.98	0.8641	77.40 $\pm$ 0.98	0.9187	**78.53** $\pm$ 1.96	0.9377
Acoustic						
Pitch (std-f_0) (1)	61.58 $\pm$ 0.98	0.6507	**71.19** $\pm$ 3.39	0.7939	51.41 $\pm$ 0.98	0.7427
Pitch (mean-f_0) (1)	54.24 $\pm$ 1.69	0.5223	53.11 $\pm$ 0.98	0.5465	**61.02** $\pm$ 0.00	0.5235
Sil.Sp.Hist (50)	57.63 $\pm$ 0.00	0.4159	**59.32** $\pm$ 2.94	0.7069	55.93 $\pm$ 1.69	0.6483
All Acoustic (52)	56.50 $\pm$ 2.59	0.5864	**63.28** $\pm$ 0.98	0.7059	61.02 $\pm$ 4.48	0.6589
Linguistic						
Unigrams (134)	53.11 $\pm$ 1.96	0.7275	**64.41** $\pm$ 4.48	0.6173	63.28 $\pm$ 0.98	0.7651
Unigrams - LIWC (100)	52.54 $\pm$ 4.48	0.5906	**63.84** $\pm$ 2.59	0.6764	55.93 $\pm$ 1.69	0.7729
All Linguistic (234)	53.11 $\pm$ 4.27	0.6765	**61.58** $\pm$ 2.59	0.6605	57.63 $\pm$ 1.69	0.7655

Best results in each line are shown in bold.



Input Layer
337
Embedding
337
Conv1D
337 x 200
Max Pooling 1D
337 x 128
Conv1D
112 x 128
Max Pooling 1D
112 x 64
BI-LSTM
56 x 64
ResNet50
Flatten
7 x 7 x 2048
Dense
100352
1024
Concatenate
512
32
Dense
50
32
LSTM
56 x 100
Dense
544
128
64
16
1