AI Interview Coach (Video)

## Overview -

The AI Interview Coach is a deep learning–based system designed to analyse a candidate's behaviour during mock interviews and provide feedback on performance attributes such as confidence, engagement, and expression. The project combines computer vision, temporal sequence modelling, and behavioural analytics to performance scores and feedback on how the candidate performed.

## Objectives -

- Capture and analyse facial expressions and movements from recorded or live interview videos. Extract temporal behavioural features for model training.
- Build a time-series model that learns from these features to predict perceived confidence or performance quality.
- Evaluate model robustness and generalization using appropriate validation strategies for sequential data.

## System Architecture -

Video Input → OpenFace Feature Extraction → OpenSMILE Feature Extraction → Merged Time-Series Dataset → Preprocessing → Model Training (LSTM / 1D CNN) → Evaluation & Bias

1) Analysis Components:
   a) OpenFace: Facial action units, gaze, head pose
   b) OpenSMILE: Pitch, Probability of voicing, Loudness (via prosodyShs.conf)
2) Data Preprocessing: Flattening to fit scaler and padding sequences to same length
3) Modelling: LSTM and 1D CNN for sequence learning
4) Evaluation: Walk-forward validation respecting time order

## Tools and Technologies -

OpenFace – Facial behaviour analysis
LSTM – Sequence modelling for time-series data
1D CNN– Temporal feature baseline (less stable)
Google Colab – GPU training environment
Docker - Attempted containerization (GPU issue, abandoned)
Python (Keras, NumPy, Pandas, Matplotlib)– Implementation and analysis

## Implementation Journey-

1. Local Setup: Encountered dependency and CUDA issues running OpenFace locally.
2. Docker Deployment: Worked on CPU but failed to access GPU via nvidia-docker. DLIB couldn't recognize system GPU.
3. Migration to Google Colab: Resolved dependency issues and enabled GPU acceleration.

AI Interview Coach (Video)

## Data Augmentation -

Added 4 different augmentation functions

1. Jitter: Adds Gaussian noise to the time series data.
2. Scale: Multiplies the time series data by a random scalar
3. Time Warp: Stretches or compresses the time axis of the data
4. Window Slice: Takes a random sub-sequence from the data

Number of augmentations per file = 5

## Model Development -

Data:

- Extracted OpenFace features (AUs, head pose) per frame.
- Extracted OpenSMILE prosodic features.

Preprocessing: Grouped features into sequences, normalized inputs, 80:20 train: test split.

Models:

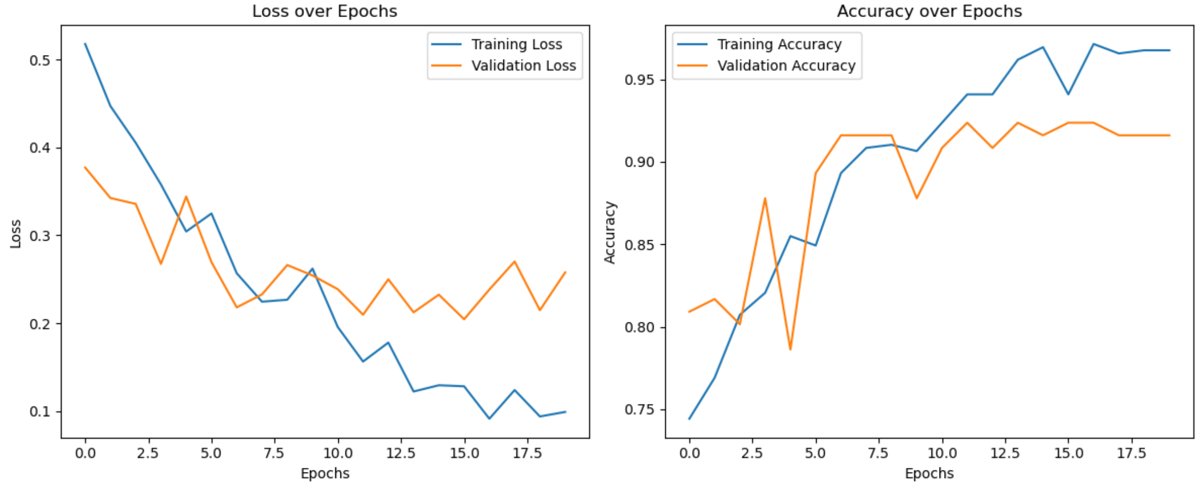
- LSTM: Stable and effective for temporal dependencies.
- 1D CNN: Faster but less stable generalization.

## Validation Strategy -

Standard k-fold cross-validation was avoided due to temporal data dependency. Instead, walk-forward validation was implemented to ensure that the model trained only on past data and predicted future behaviour, preventing data leakage and ensuring realistic performance estimation.

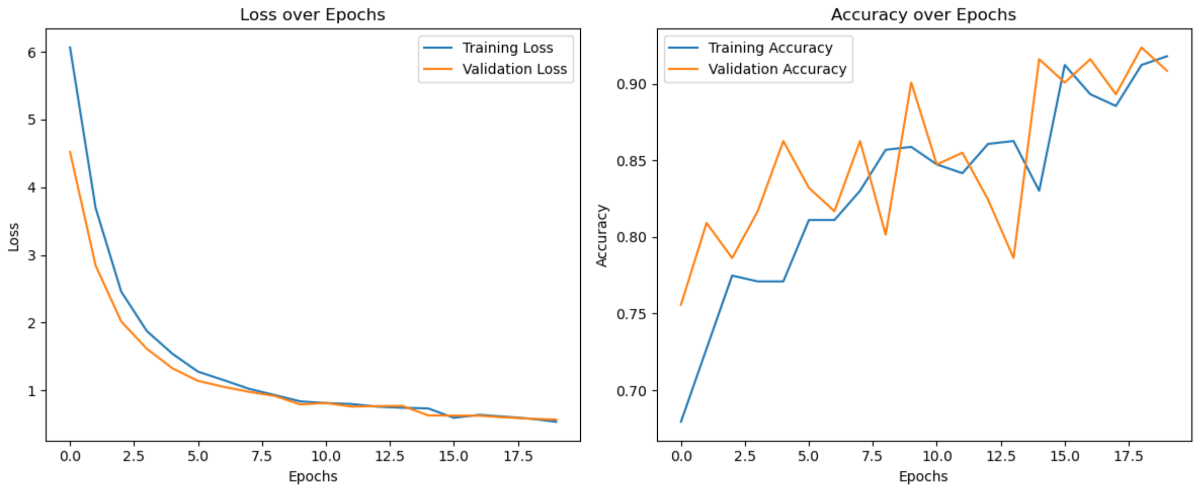
## Results and Observations

**LSTM:** Ran a basic LSTM just with the 819 Augmented dataset, It's training history shows the training loss decreases but validation loss plateus showing the model is highly overfitting.



AI Interview Coach (Video)

LSTM demonstrated stable learning upon adding **L2 regularization** and showed an AUC greater than 95%



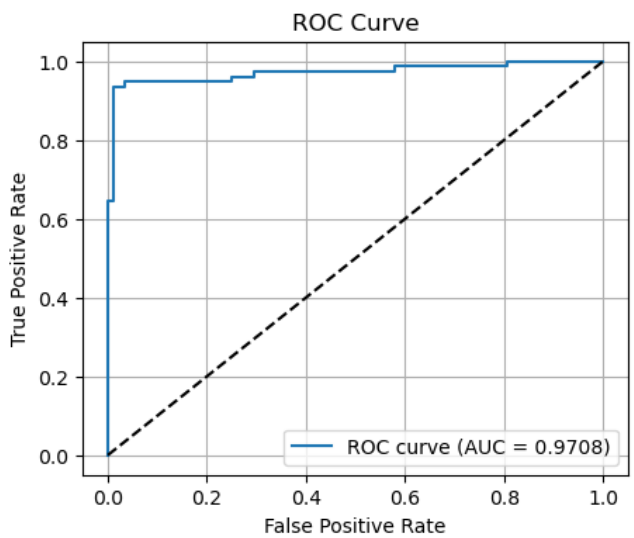

Key observations and what they mean:

Decreasing Loss: In the "Loss over Epochs" graph, both the training loss and validation loss are decreasing over time. This is a positive sign that the model is learning from the data and getting better at its task.

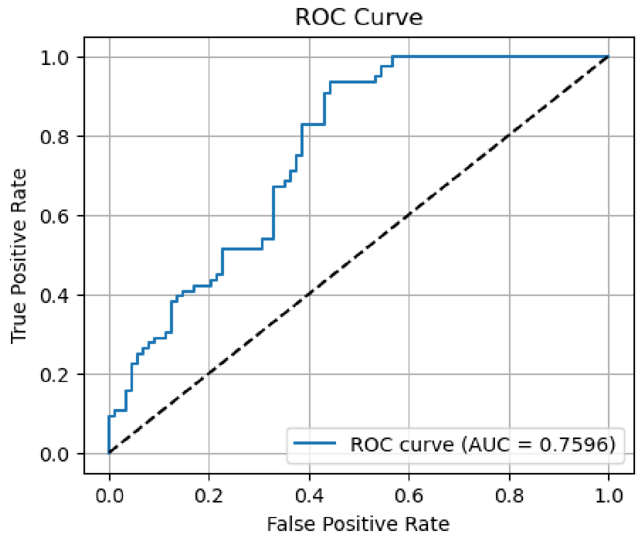
Increasing Accuracy: The "Accuracy over Epochs" graph shows that both the training accuracy and validation accuracy are generally increasing. This means the model is making more correct predictions as it trains.

Converging Curves: The training and validation curves for both loss and accuracy are very close to each other. When the curves are close, it indicates that the model is generalizing well and is not overfitting to the training data. Overfitting would be indicated by a large gap where the training loss is low and the validation loss is high, and the training accuracy is high while the validation accuracy is low.

Stability: The curves eventually flatten out or "plateau" towards the end, suggesting that the model has learned as much as it can from the given data and has reached a stable or optimal state.




The ROC curve in the image demonstrates a highly effective model with excellent performance, as its curve is very close to the top-left corner and its AUC value is nearly perfect.

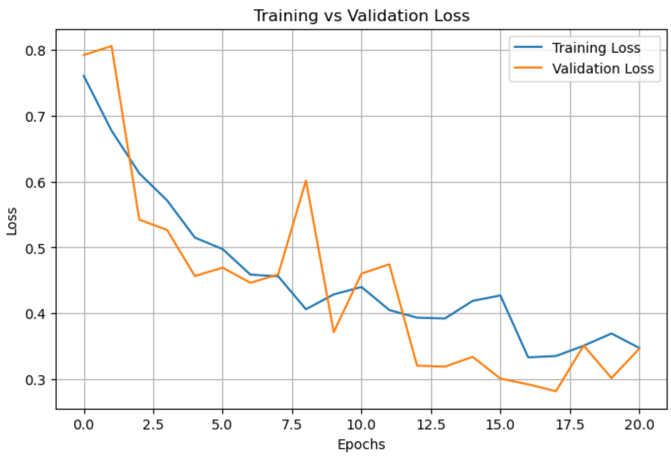



But then, after running the walk-forward validation, **AUC was down to 76%** indicating when randomly picking one positive and one negative sample, there is a 76% chance your model will correctly rank the positive sample higher **showing bias toward predicting high confidence levels** consistently. This was likely due to dataset imbalance.

| VALIDATION METHOD | PRECISION | RECALL | ROC AUC | INTERPRETATION |
|---|---|---|---|---|
| **STANDARD (L2)** | 0.96 | 0.96 | 0.9708 | Possible optimistic bias—future info leakage |
| **WALK-FORWARD** | 0.77 | 0.71 | 0.7596 | Realistic generalization on time-ordered data |

**1D CNN:**

The 1D CNN model was trained using L2 weight regularization to improve generalization and reduce overfitting. As shown in the **training vs. validation loss curve**, both losses decrease steadily while closely tracking each other, indicating stable learning and effective regularization. The absence of divergence confirms that overfitting was well controlled.




The model achieved an **AUC of 0.99**, demonstrating a very strong ability to distinguish between classes. However, such a high AUC suggests the need for further validation on unseen or cross-validated data to ensure true generalization. Overall, the model shows excellent fit quality with balanced loss behaviour and strong discriminative performance

## Challenges Encountered-

- OpenFace local dependency conflicts (DLL and CUDA)
- GPU unavailability in Docker environment
- Model (LSTM) bias towards high-confidence samples – Temporary fix – giving a higher threshold instead of 0.5

## Future Work-

- Expand dataset for balanced confidence levels
- Incorporate multimodal features (speech, tone, text)
- Dimension Reduction (PCA)
- Add attention mechanisms or transformer-based encoders
- Reintroduce GPU-compatible
- Docker containerization
- Develop an interactive feedback dashboard