DEEP LEARNING PROJECT

DeepFake Detection

Presented by:
Neha Kantheti CS22B1081
Sri Manaswini Velide CS22B2030
Bhargavi Antham ME22B2035
Under the guidance of Dr. Umarani Jayaraman
Dept. of CSE, IIITDM Kancheepuram

Overview

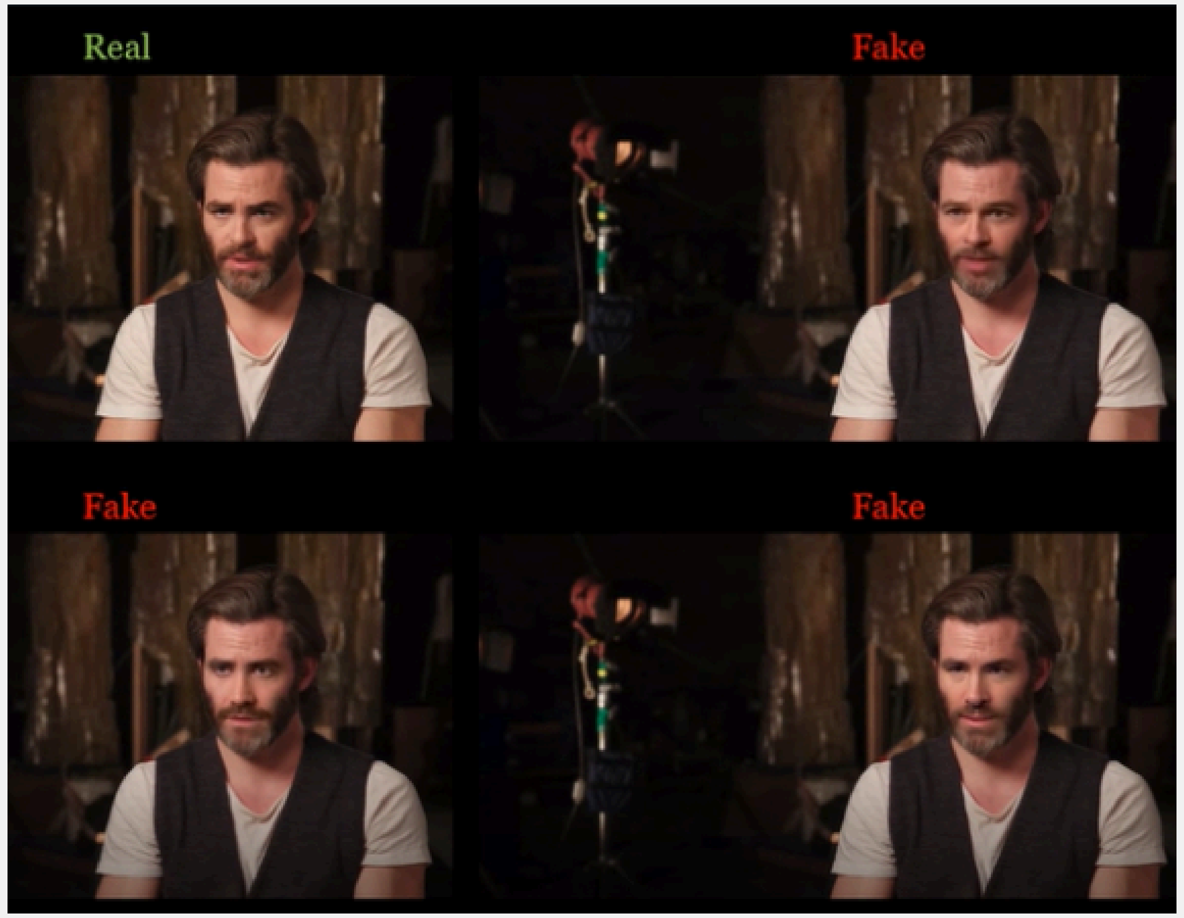
Objectives

- To develop deep learning models that accurately detect deepfake videos by noticing the subtle inconsistencies between frames.

Challenges

- Deepfake datasets consist of thousands of video frames with subtle manipulations, requiring high-quality preprocessing and frame selection.
- Training ViT, ResNet+LSTM, and Xception models on video data requires significant GPU resources and memory.

Dataset Introduction



- **Dataset Used – Celeb-DF v2**

Data Handling:

  - Celeb-DF v2 (Celeb-Deepfake Dataset)
  - Contains real and deepfake videos of celebrities
  - Real: ~590 videos + Youtube Real ~ 300 | Deepfake: ~5639 videos
  - Format: .mp4 videos with variable lengths
  - Purpose: Real-world challenging deepfake detection

Dataset Imbalance

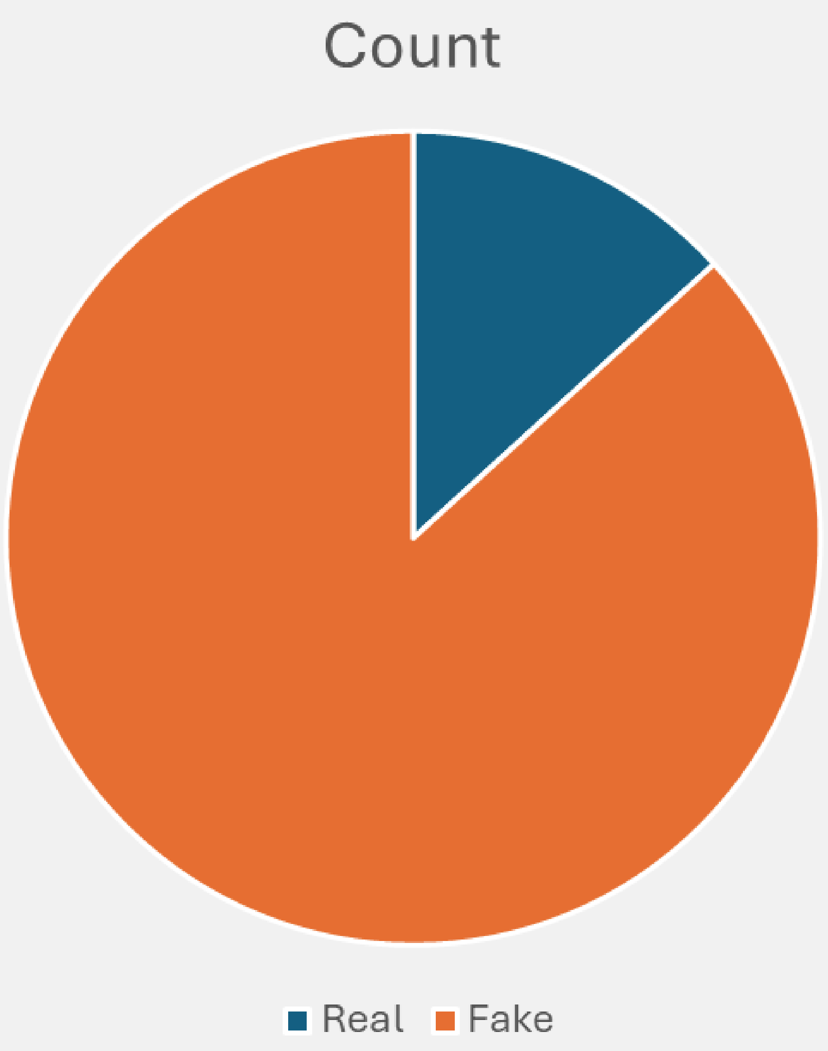
Observations:

- Real: ~890 videos (590 + 300)
- Fake: ~5639 videos
- Imbalance ratio: 1:~6.33
- Challenges: Biased learning, high false positives/negatives

Handling the Dataset

We've tried different data handling techniques

- Class Weights (1 : 6.33)
- UnderSampling



Data Preprocessing

- Loads sampled frames from videos.
- Applies transform (resize, center crop, normalization).
- Converts to tensor for fitting into the model.

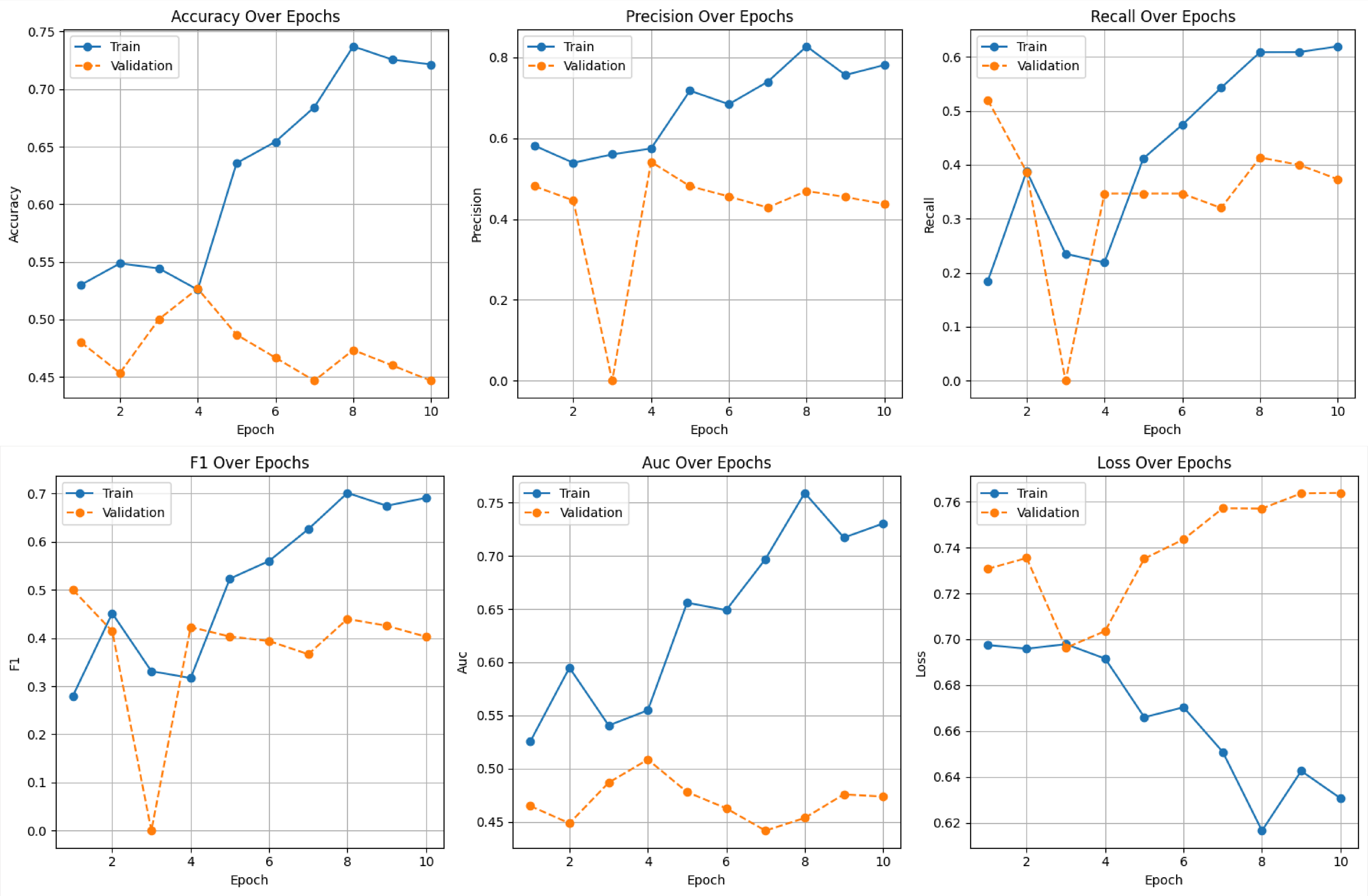
Models Used

- **Resnet + LSTM**
- **Vision Transformers**
- **XceptionNet**

Model Architecture - Model 1

1. **CNN (ResNet) for Feature Extraction**
   - A pre-trained ResNet18 model is used.
   - It extracts spatial features from individual video frames.
   - The final fully connected (fc) layer is removed to obtain only feature vectors.
2. **LSTM for Temporal Modeling**
   - An LSTM (Long Short-Term Memory) layer takes in the sequence of frame features.
   - Captures temporal dependencies across video frames.
   - The final output of LSTM is passed through a fully connected layer to classify the video.

Training and Testing (Resnet + LSTM)



Results(ResNet + LSTM)

```
Accuracy: 0.4867
              precision    recall  f1-score   support

        Real       0.49      0.56      0.52        75
        Fake       0.48      0.41      0.45        75

    accuracy                           0.49       150
   macro avg       0.49      0.49      0.48       150
weighted avg       0.49      0.49      0.48       150

Confusion Matrix:
[[42 33]
 [44 31]]
```

Model Architecture - Model 2

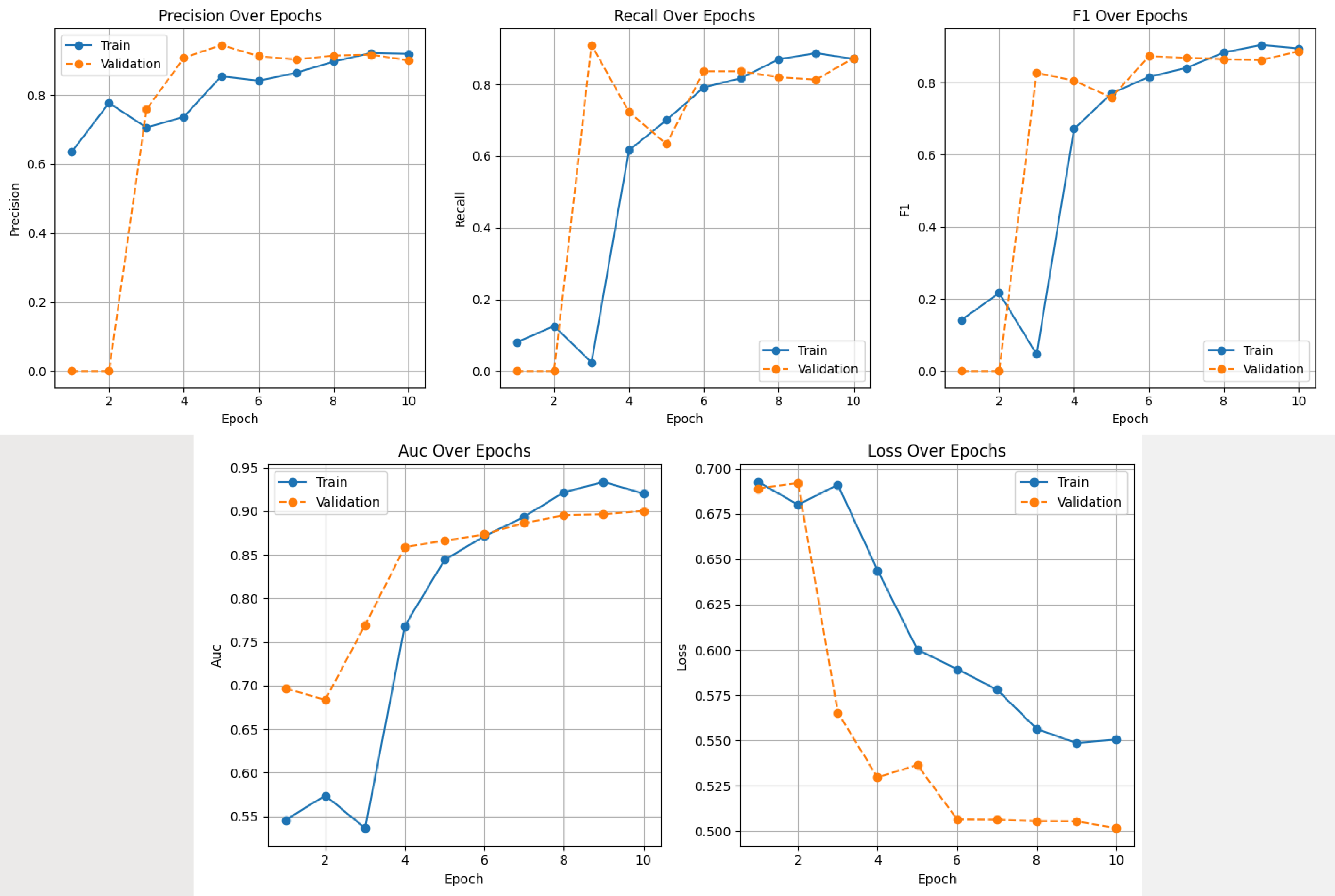
1. Vision Transformer (ViT) Backbone

- Uses a pretrained ViT model (like vit_base_patch16_224 from the timm library).
- The model takes in images (video frames) and treats them as a sequence of patches.
- Unlike CNNs, ViT models learn global dependencies across the image using self-attention mechanisms.

2. Classification Head

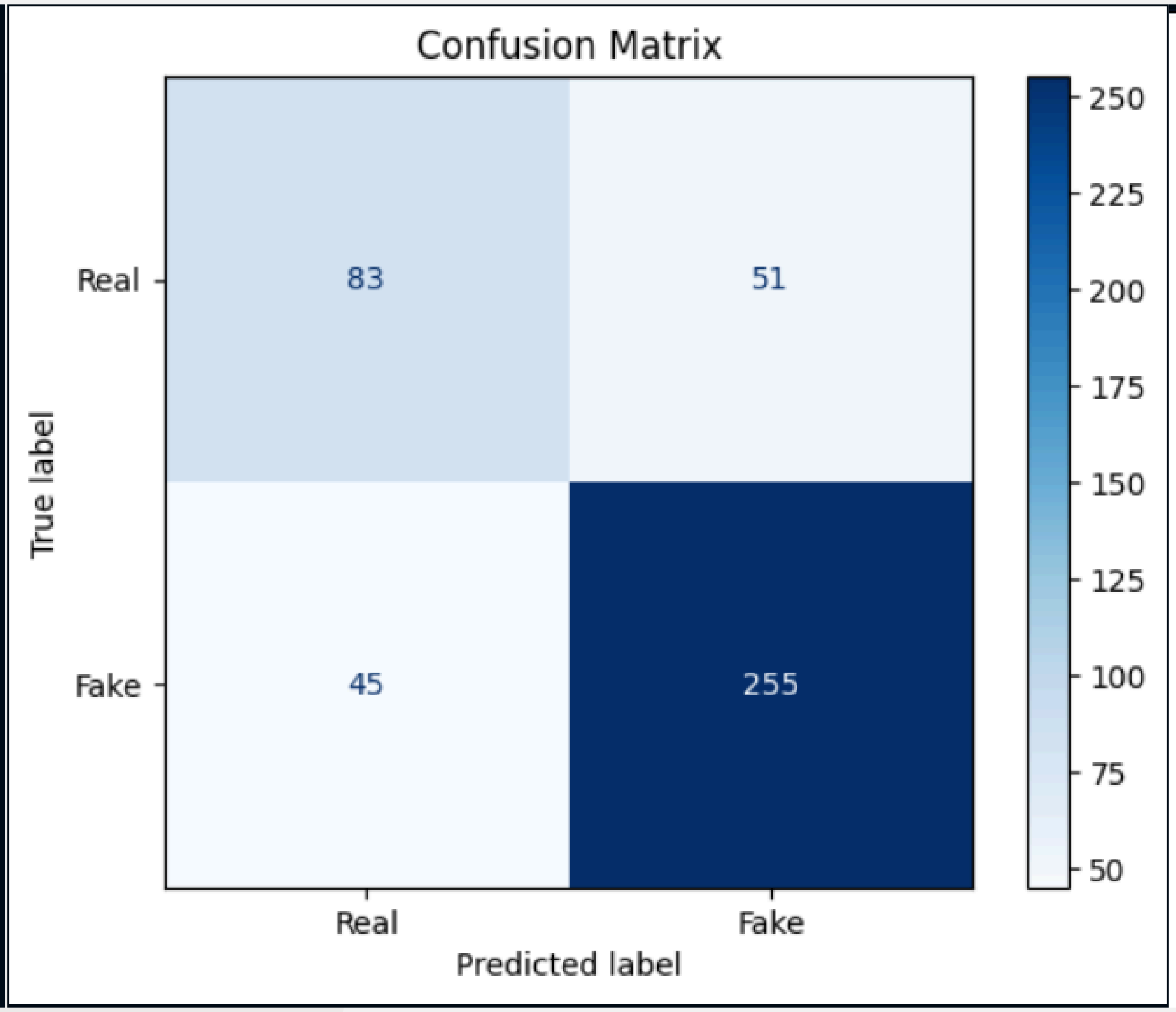
- The transformer outputs are pooled (usually from the [CLS] token).
- A linear classification head maps it to binary classes: Real or Fake.

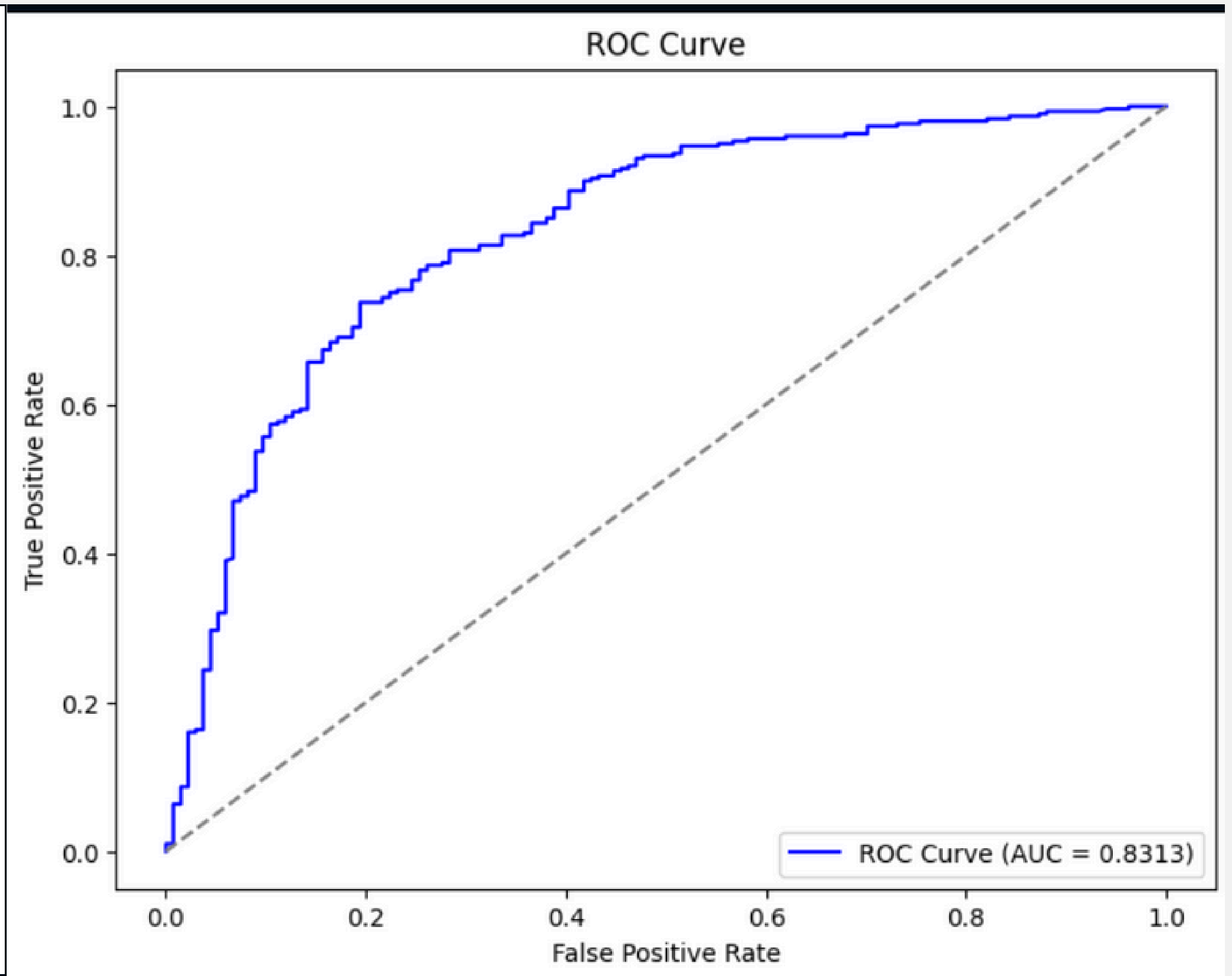
Training and Testing (ViT)



Results(Vision Transformer)

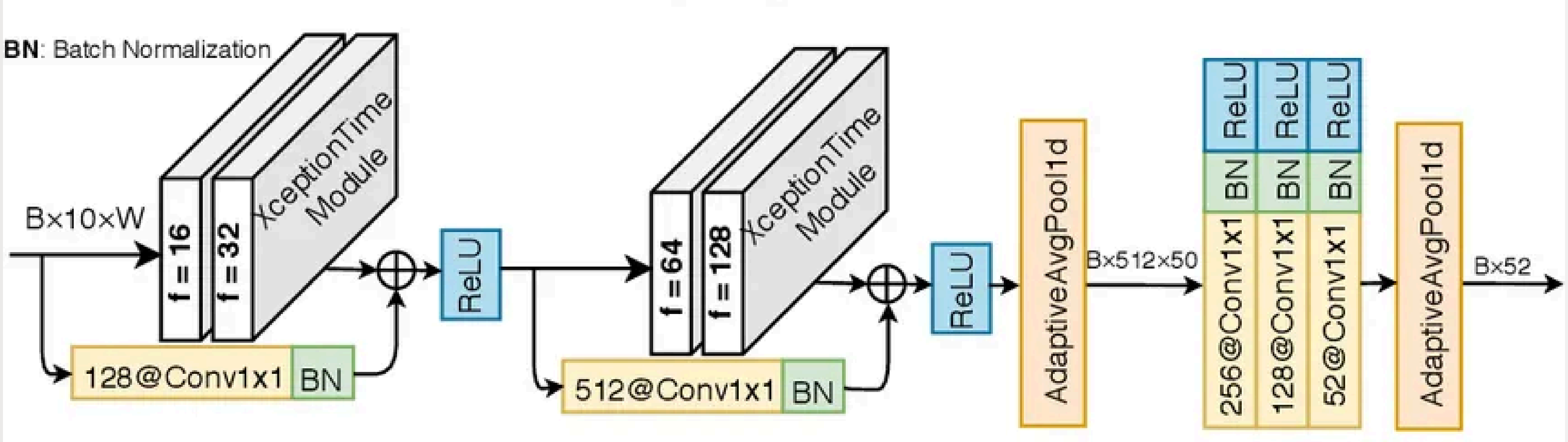
```
Evaluating Model...

Metrics:
Precision: 0.8333
Recall: 0.8500
F1: 0.8416
Auc: 0.8313
Accuracy: 0.7788
```





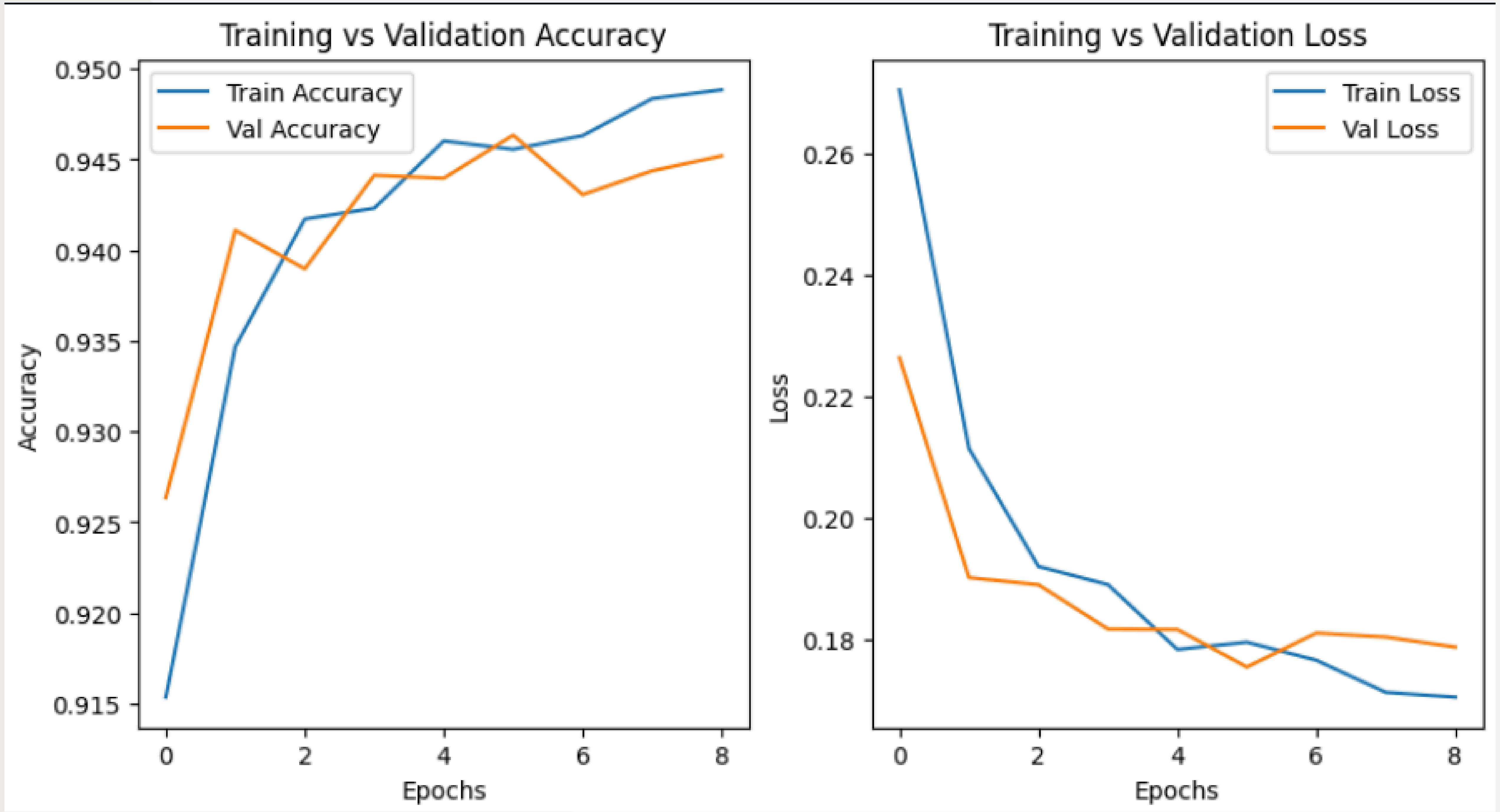
Model Architecture - Model 3

- Loads pretrained Xception on ImageNet.
- **num_classes=2** modifies the final layer to binary classification (Real vs Fake).
- Input size is 299x299, which matches transforms.Resize() in the dataset.



(b) XceptionTime Architecture

Training and Testing (XceptionNet)

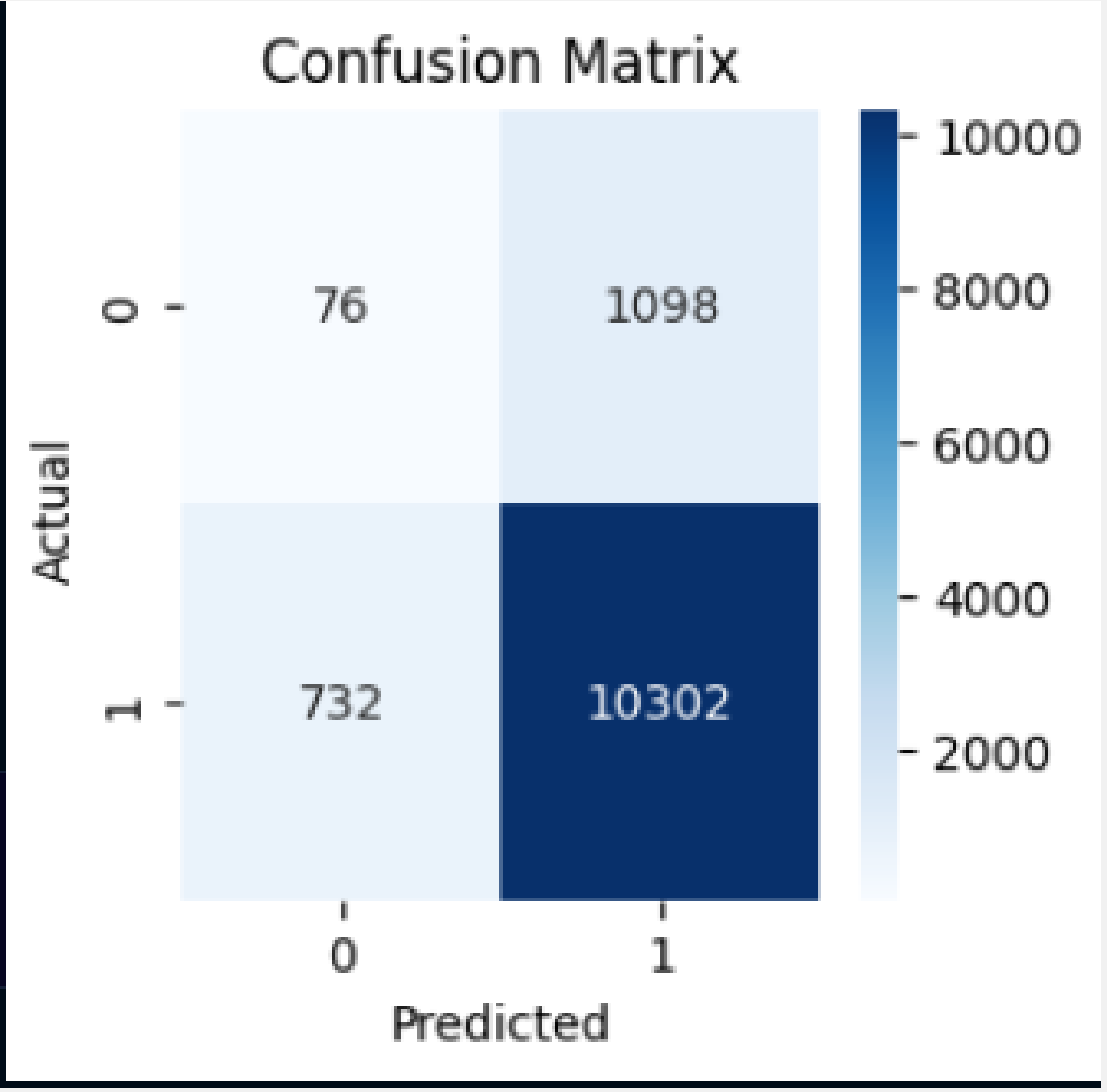


Results(XceptionNet)

```
382/382 ━━━━━━━━━━━━━━━━━━━━ 47s 117ms/step
              precision    recall  f1-score   support

        Real       0.96      0.40      0.56      1173
        Fake       0.94      1.00      0.97     11035

    accuracy                           0.94     12208
   macro avg       0.95      0.70      0.76     12208
weighted avg       0.94      0.94      0.93     12208

AUC-ROC: 0.8567588750202313
F1 Score: 0.9680112487916337
```

```
eer = compute_eer(y_true, y_pred)
print(f"EER Score : ", eer)
```

```
EER Score :  0.3026964888514457
```



Comparision

Metric	ResNet-18	Vision Transformer	XceptionNet
Accuracy	48.67%	77.88%	94.00%
Precision	48.44%	83.33%	94.00%
Recall	41.33%	85%	100.00%
F1 Score	44.60%	84.16%	97.00%
AUC	0.4933	0.8312	0.857
EER	N/A	N/A	0.3027

Conclusion

- This project evaluated three deep learning-based approaches for deepfake detection.
- The Xception model provided the most consistent results, followed by the Vision Transformer. ResNet18 with LSTM showed the challenges of temporal modeling with limited data.
- Future improvements could leverage advanced temporal networks and multimodal inputs to enhance performance further.

Thank you