Deepfake Detection System

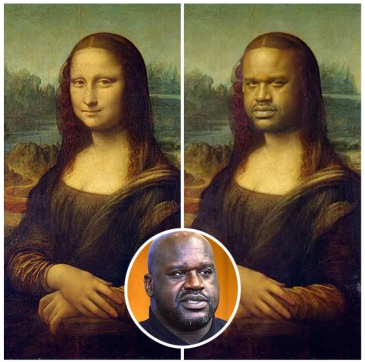
By: Harsh, Caleb, Ricky, Anand

Problem Statement

- The proliferation of deepfake technology has resulted in manipulated media that can be used to misinform the public, threaten privacy, and erode trust in digital content.

- Detecting such manipulations is a critical challenge in both technological and social contexts.

- This project aims to build a robust deepfake detection system that leverages Convolutional Neural Networks (CNN) for spatial feature extraction and Variational Autoencoders (VAE) for learning latent representations and anomaly detection, focusing on detecting video manipulations.



Success Metrics

- **Accuracy** provides overall model effectiveness, showing strong performance well above random baseline.
- **False Positive Rate** measures user trust impact
    - Incorrectly flagging real content as fake damages system credibility and user confidence.
- **False Negative Rate** is the **most critical metric** for deepfake detection, as missed fakes can spread misinformation, enable fraud, and cause social harm. Even small FNR represents thousands of undetected malicious videos at scale.
- Together, these metrics balance detection accuracy with real-world deployment considerations: minimizing harmful content spread while maintaining user trust."

Datasets

Combination of 4 datasets:

1. **FaceForensics++** (compressed c23 subset):
   ~1,000–1,500 real and fake videos for training and validation

2. **Celeb-DF v2:**
   For cross-dataset evaluation and robustness testing

3. **Deepfake Detection Challenge (DFDC, Kaggle):**
   Large-scale dataset with over 100,000 labeled clips, from which a balanced subset (~10,000–20,000 samples) will be curated

4. **WildDeepfake (Kaggle):**
   Real-world deepfake dataset containing diverse and in-the-wild manipulations for stress-testing generalization

Sampled ~50k from each dataset

Some Dataset EDA:

- **Video** - based statistics (file size, frame count, frame rate)
- **Image** - based statistics (Resolution, Brightness, sharpness, contrast)
- **Pixel** - based statistics (RGB mean and standard deviation)

📐 **Resolution Statistics:**

Original frames:
1280x720: 32 frames (26.7%)
1920x1080: 24 frames (20.0%)
640x480: 24 frames (20.0%)

Deepfake frames:
640x480: 32 frames (26.7%)
1920x1080: 24 frames (20.0%)
600x480: 24 frames (20.0%)

💡 **Brightness (0-255 scale):**
Original: 111.56 ± 39.56
Deepfake: 91.56 ± 40.73
Difference: 20.00

📊 **Contrast (std of pixel values):**
Original: 60.08 ± 7.60
Deepfake: 53.85 ± 12.94
Difference: 6.23

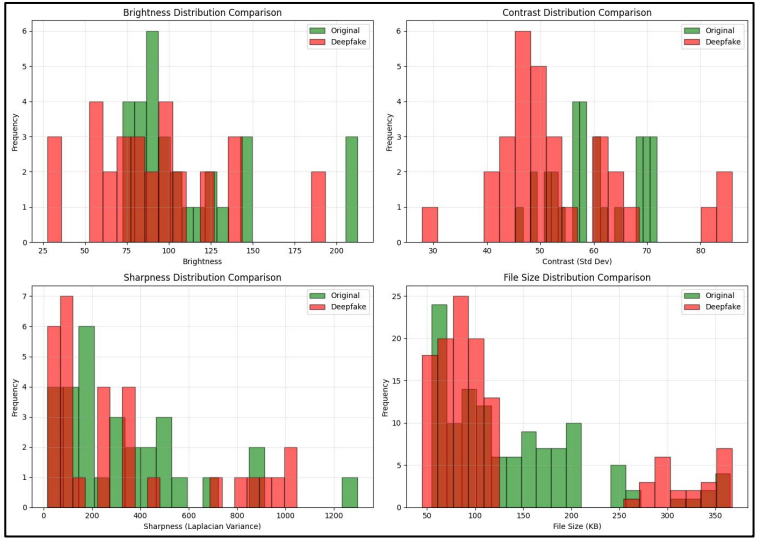
🔍 **Sharpness (Laplacian variance):**
Original: 339.77 ± 292.27
Deepfake: 338.32 ±

📈 **Per-Channel Statistics:**

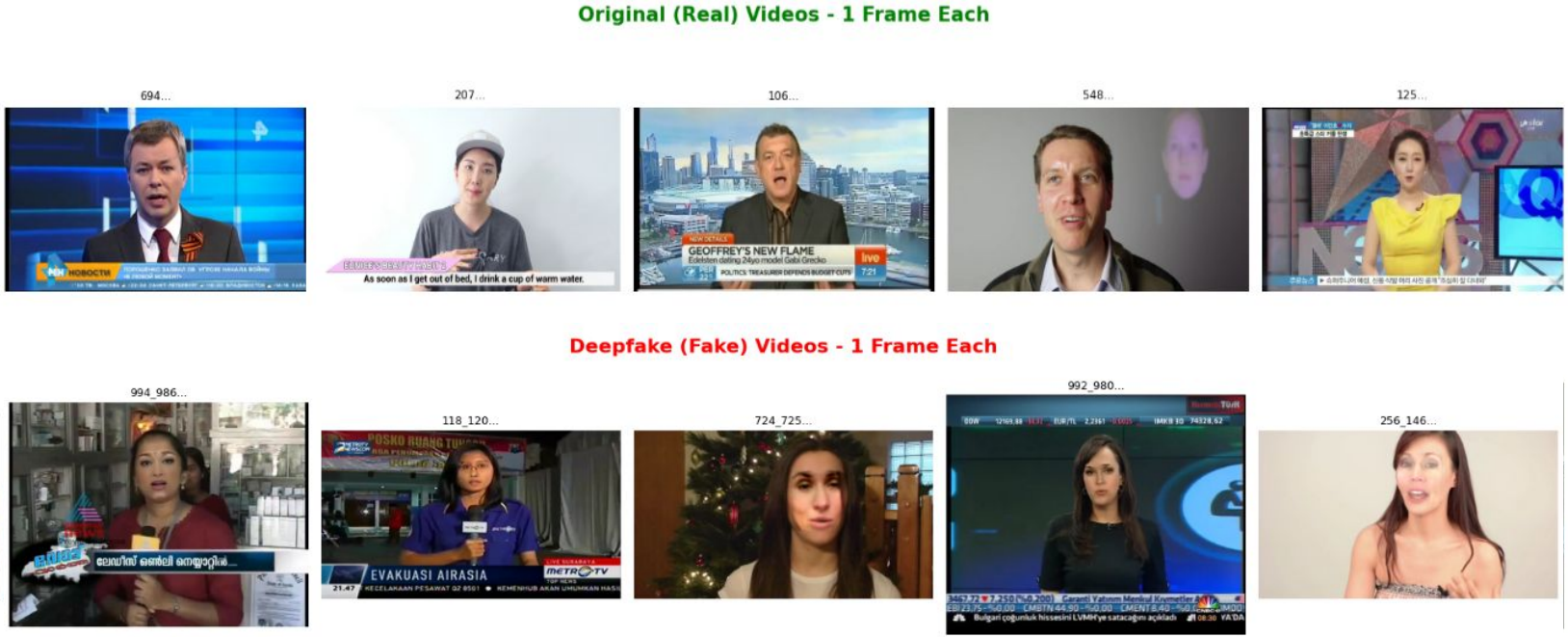
Red channel:
Mean: 0.411392
Std: 0.281409

Green channel:
Mean: 0.389298
Std: 0.264461

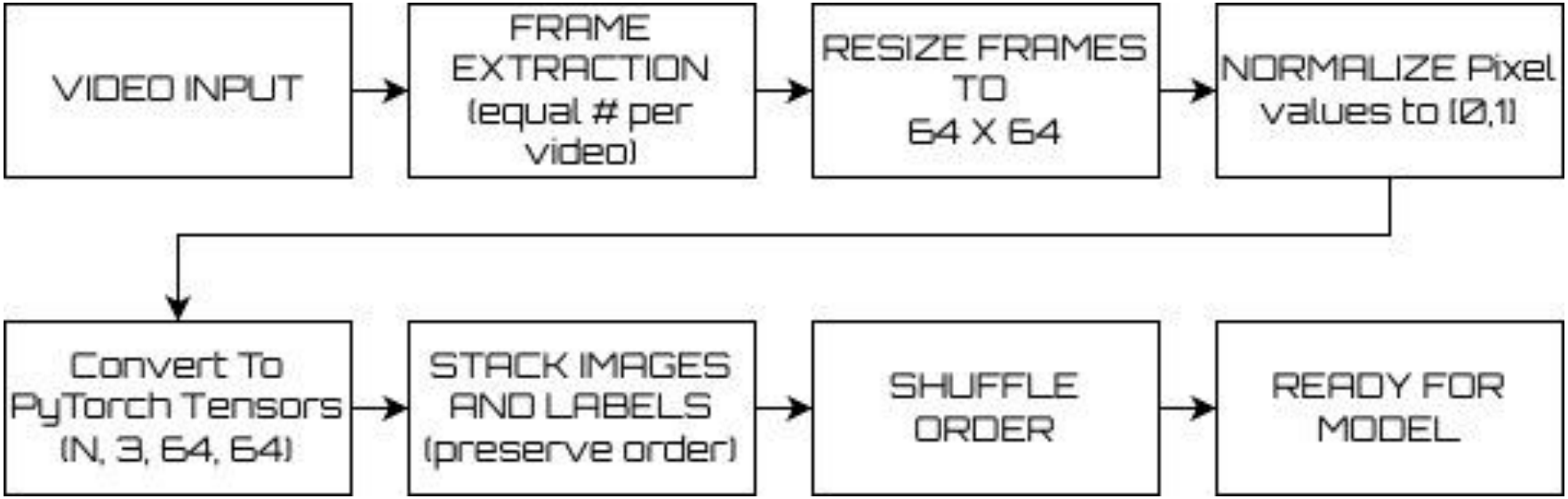
Blue channel:
Mean: 0.417100
Std: 0.275618



Some Dataset samples:



Pre-Processing Pipeline

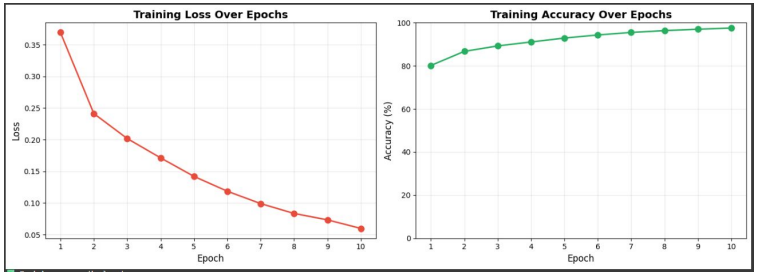


Base Model #1 - Vanilla CNN

Input (3, 64, 64)
↓
Conv1 + ReLU → (16, 64, 64)
↓
MaxPool → (16, 32, 32)
↓
Conv2 + ReLU → (32, 32, 32)
↓
MaxPool → (32, 16, 16)
↓
Flatten → (8192)
↓
FC1 + ReLU → (128)
↓
FC2 → (2)
↓
Output logits [score_real, score_fake]

Training Config:

Hyperparameter	Value
Epochs	10
Batch Size	64
Learning Rate	0.001
Optimizer	Adam
Loss Function	CrossEntropyLoss



Base Model #1 - Vanilla CNN

Metric	Training	Test
Accuracy	97.52%	**97.42%**
Precision	-	97.31%
Recall	-	97.49%
F1-Score	-	97.40%
ROC-AUC	-	**99.77%**
Loss	0.0599	-

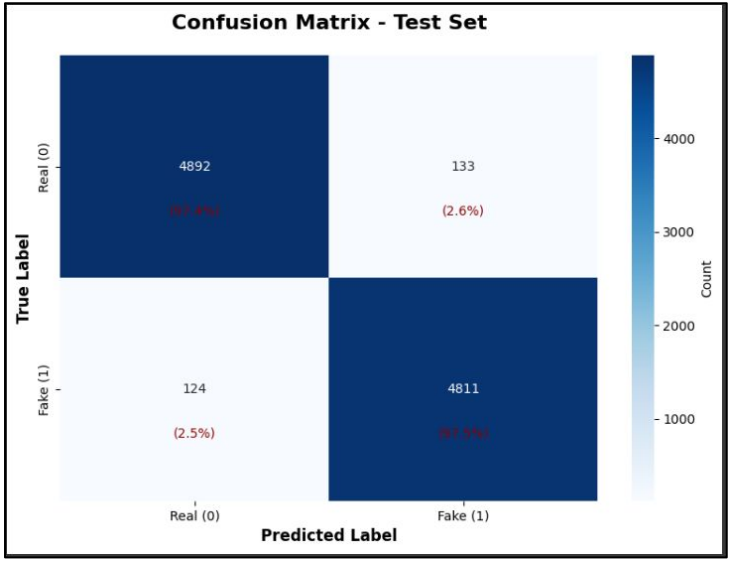
Train-Test Gap: 0.10% (excellent generalization)

Per-Class Performance (Test Set):

- Real videos: 97.35% accuracy (4,892/5,025 correct)
- Fake videos: 97.49% accuracy (4,811/4,935 correct)

Error Analysis:

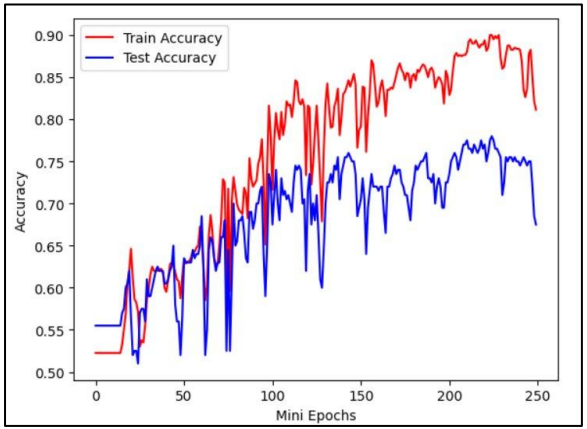
- False Positive Rate: 2.65% (133 real videos misclassified as fake)
- False Negative Rate: 2.51% (124 fake videos missed)

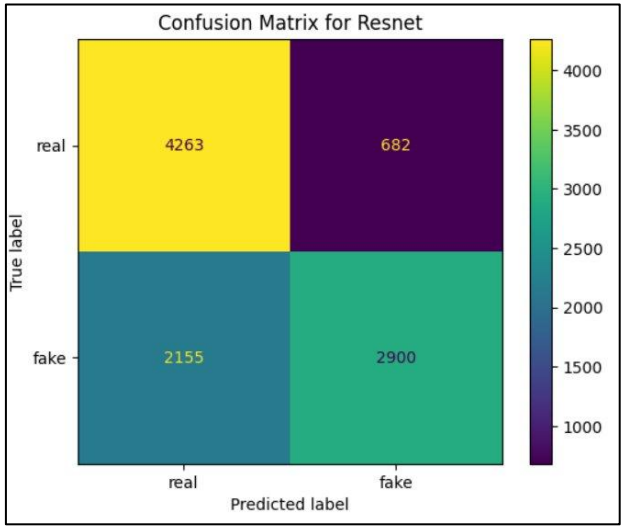


Base Model #1 - Vanilla CNN

Base Model #2 - ResNet18

- State-of-the-art CNN architecture (no pretrained weights)
- ~ 11.2M parameters
- Modified final layer for binary classification instead of 1000
- 1,000 training samples
- 10,000 test samples
- Accuracy: ~90% train, ~72% test





ResNet18 Evaluation

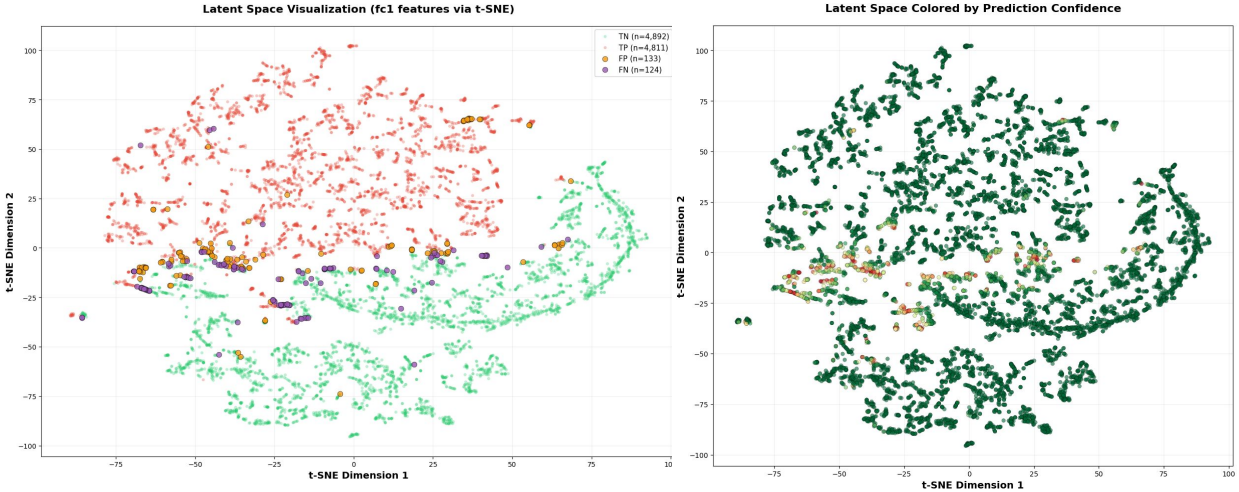
- **False positive rate of ~14%**
- **False negative rate of ~42%**
- CNN architecture is 3x more likely to be "fooled" by fake images
- Need to prioritize flagging fake images over not flagging real ones
- TOO EXPENSIVE (got kicked off Discovery Cluster for “disk quote exceeded”

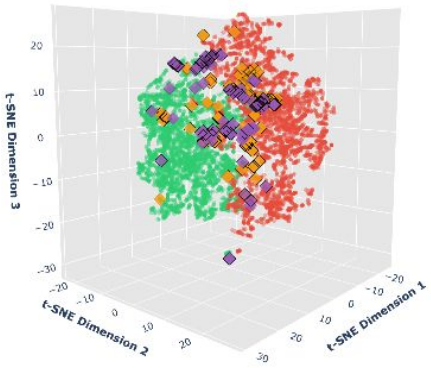
Real Images Predicted Fake

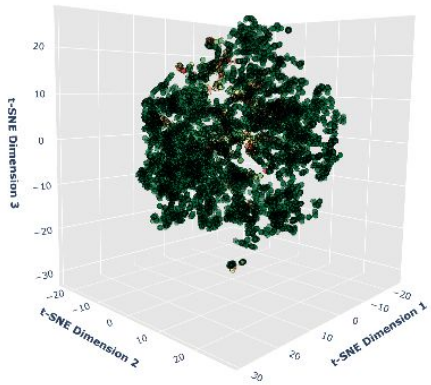
Fake Images Predicted Real

What went wrong with the Base models?

#1 Clustering



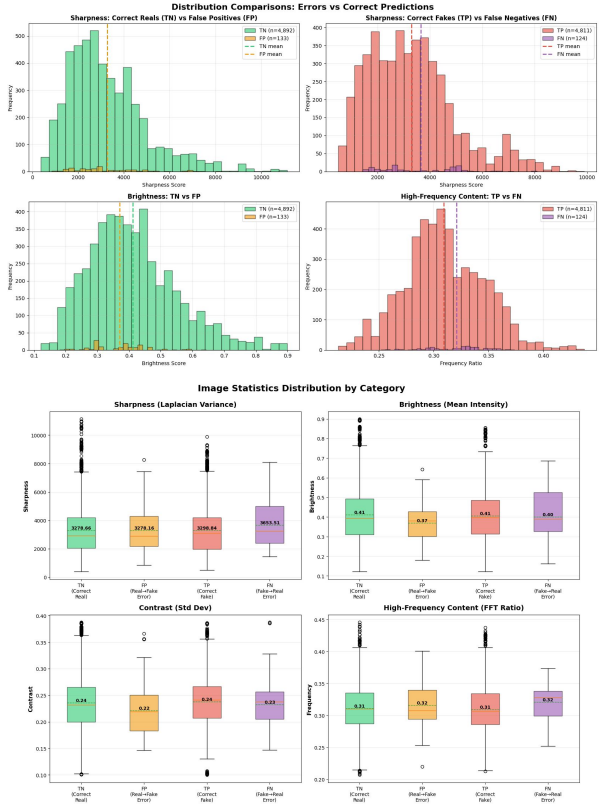




What went wrong with the Base models?

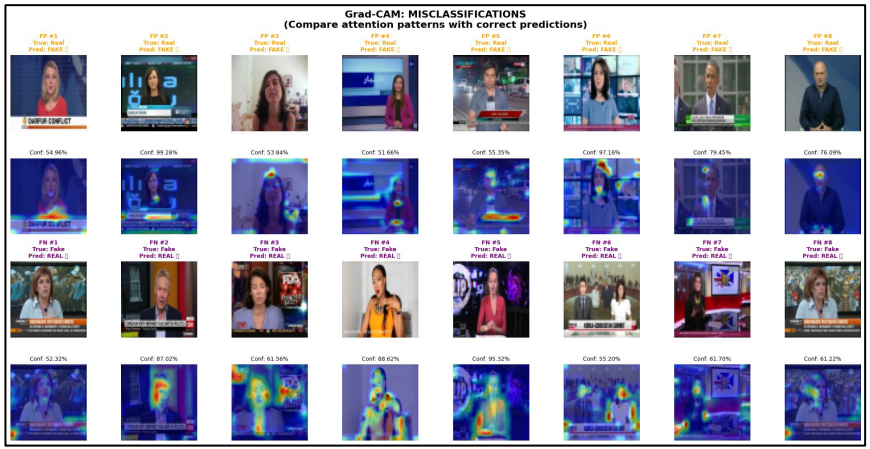
#2 Quality

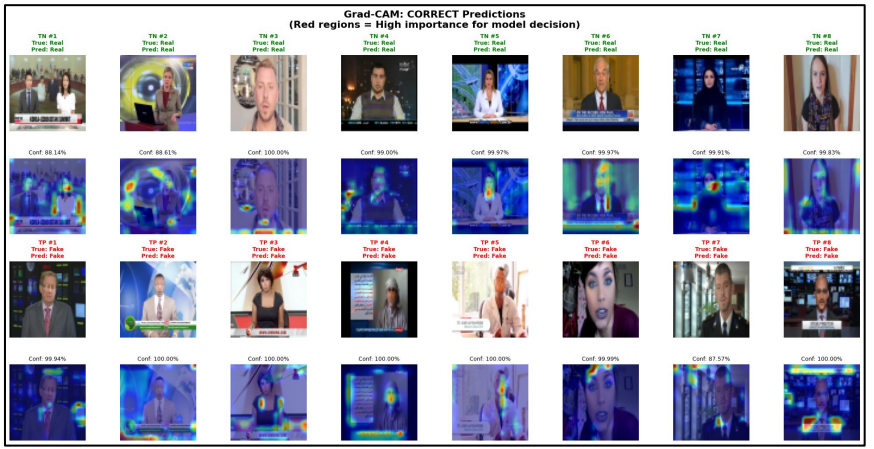
SHARPNESS ANALYSIS	BRIGHTNESS ANALYSIS	CONTRAST ANALYSIS
TN (Correct Reals): Mean: 3278.6577 Std: 1713.4177	TN (Correct Reals): Mean: 0.4121 Std: 0.1405	TN (Correct Reals): Mean: 0.2356 Std: 0.0514
TP (Correct Fakes): Mean: 3298.8435 Std: 1665.4311	TP (Correct Fakes): Mean: 0.4058 Std: 0.1233	TP (Correct Fakes): Mean: 0.2377 Std: 0.0458
FP (Real→Fake (Error)): Mean: 3278.1553 Std: 1573.1661	FP (Real→Fake (Error)): Mean: 0.3705 Std: 0.0883	FP (Real→Fake (Error)): Mean: 0.2210 Std: 0.0452
FN (Fake→Real (Error)): Mean: 3653.5122 Std: 1554.0906	FN (Fake→Real (Error)): Mean: 0.4015 Std: 0.1255	FN (Fake→Real (Error)): Mean: 0.2322 Std: 0.0440



What went wrong with the Base models?

#3 Model’s Attention



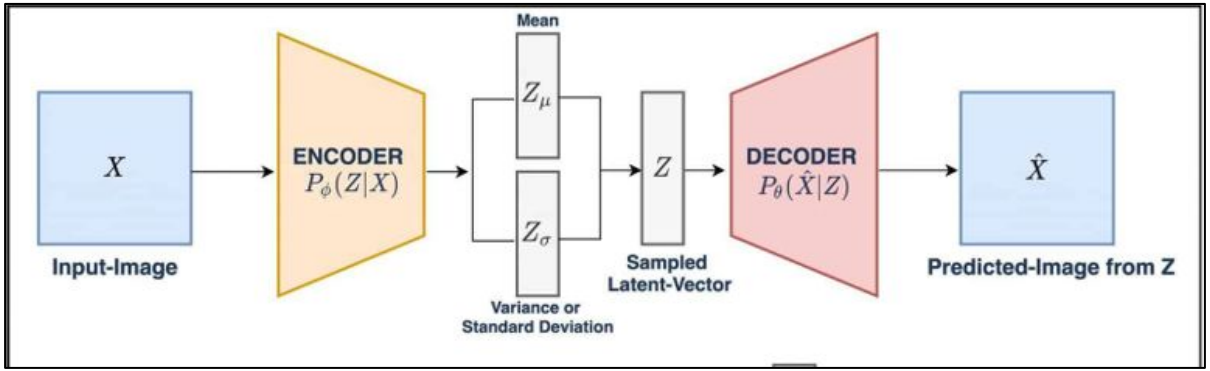


Training Modification for Vanilla CNN

- Combine CNN features with VAE reconstruction learning and image quality analysis
- Custom Loss Function that weights the BCE loss of each image by VAE reconstruction error and quality loss
- Inspired by Soft Margin SVM
- Lightweight (~1–2M params)
- Focus on flagging fakes, better generalization, maintain ≥90% accuracy

About the VAE…

- Convolutional Encoder and Decoder, input and output are both 3x64x64 tensor images
- Encoder used Conv2D for downsampling, decoder uses ConvTranspose2D for upsampling
- ReLU activations between layers
- Latent space dimension of 128
- Reparameterization trick for sampling latent space between encoder and decoder
- Trained on REAL images only



VAE Reconstruction Loss

$$L_{\text{VAE-Rec}}(X_i) = \frac{||X_i - \hat{X}_i||^2}{C \cdot H \cdot W}$$

- X_i = ith image of batch
- $\hat{X}_i$ = VAE prediction on input X_i
- C = image channels (3)
- H = image height (64)
- W = image width (64)
- Computes per pixel MSE between X_i and $\hat{X}_i$
- VAE trained on real images $\implies$ higher $L_{\text{VAE-Rec}}$ for fake images

Quality Loss

Quality degradation per metric:

$$\Delta S_i = \frac{|S(I_i) - S(\hat{I}_i)|}{S(I_i) + \epsilon}$$

$$\Delta B_i = |B(I_i) - B(\hat{I}_i)|$$

$$\Delta C_i = |C(I_i) - C(\hat{I}_i)|$$

Combined quality loss per sample:

$$\mathcal{L}_{quality}^{(i)} = \frac{\Delta S_i + \Delta B_i + \Delta C_i}{3}$$

For entire batch:

$$\mathcal{L}_{quality} = [\mathcal{L}_{quality}^{(1)}, \mathcal{L}_{quality}^{(2)}, ..., \mathcal{L}_{quality}^{(N)}] \in \mathbb{R}^N$$

Normalization:

$$\mathcal{L}_{quality}^{norm} = \frac{\mathcal{L}_{quality} - \min(\mathcal{L}_{quality})}{\max(\mathcal{L}_{quality}) - \min(\mathcal{L}_{quality}) + \epsilon}$$

Where $\epsilon = 10^{-8}$ prevents division by zero.

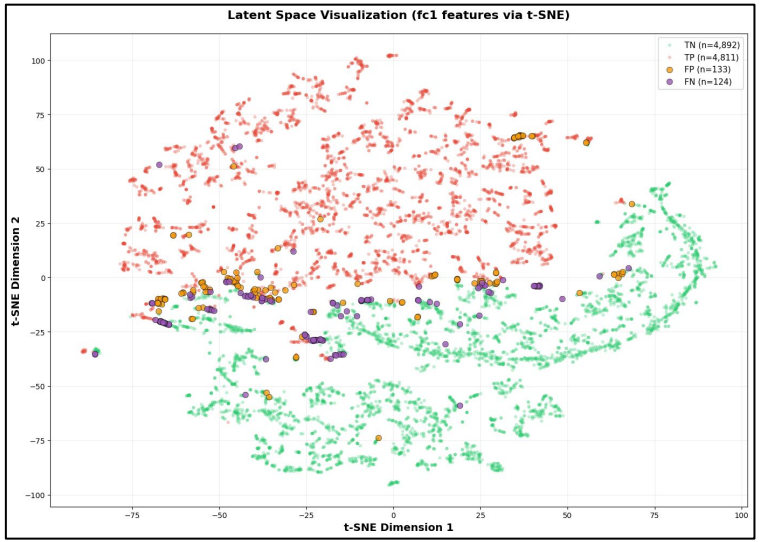
Custom Loss Function

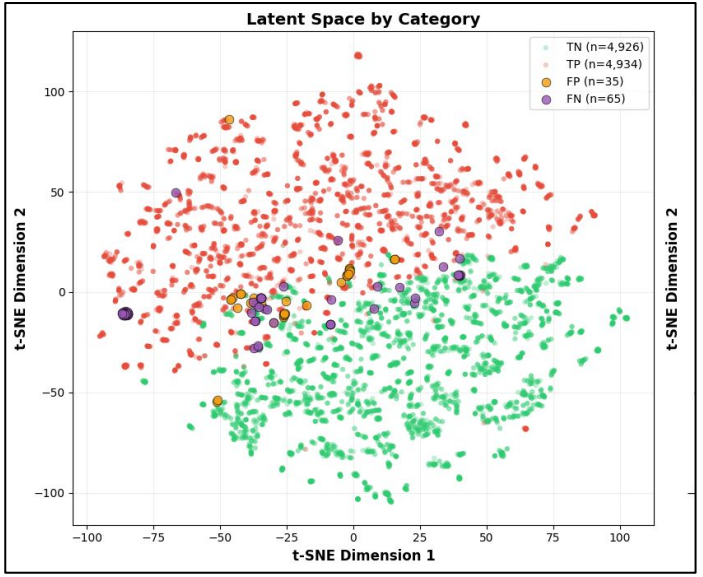
$$L_{Total}(X, y, M) = \sum_{i=0}^{n} [(L_{VAE\text{-}Rec}(X_i) + L_{Quality}(X_i)) \cdot L_{BCE}(M(X_i), y_i)]$$

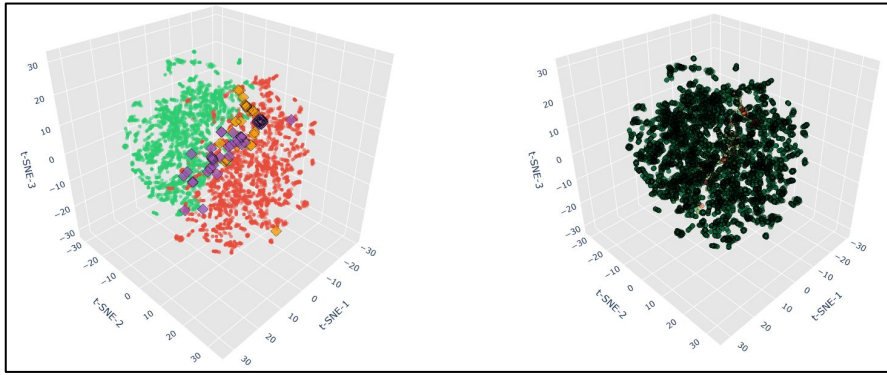
- X = batch of images, (n, 3, 64, 64) tensor
- y = batch labels, (n,) tensor
- M = Our CNN classifier
- Computes weighted binary cross entropy loss, using VAE reconstruction loss and quality loss per image
- Images that have **higher** VAE reconstruction loss or quality loss contribute **more** to the overall loss
- Inspired by Soft Margin SVM (each training sample weighted differently in loss function)

Final Result - Beating the Baseline…

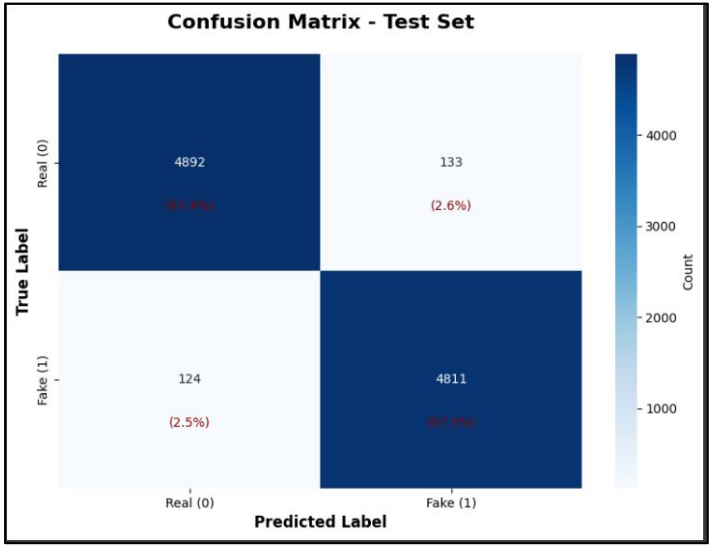
#1 Better Classification







#2 i.e. Success Metrics improved, ofc



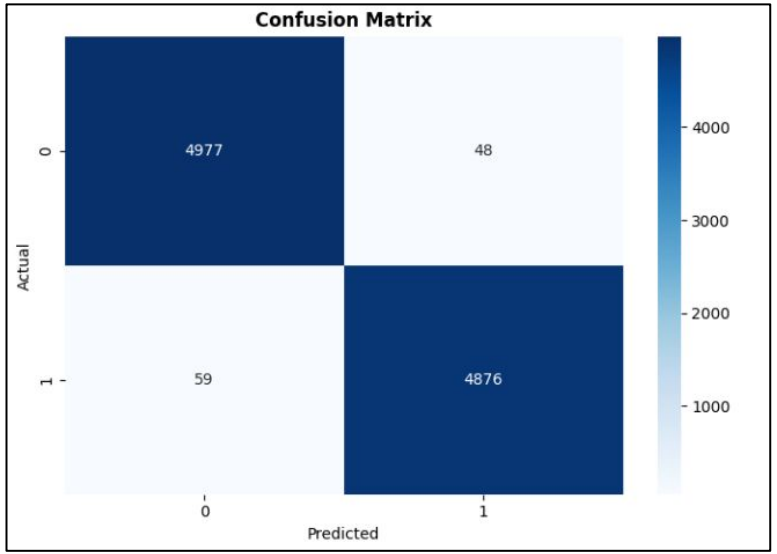
Per-Class Performance (Test Set):

- Real videos: 97.35% accuracy (4,892/5,025 correct)
- Fake videos: 97.49% accuracy (4,811/4,935 correct)

Error Analysis:

- False Positive Rate: 2.65% (133 real videos misclassified as fake)
- False Negative Rate: 2.51% (124 fake videos missed)

Accuracy = 97.42%

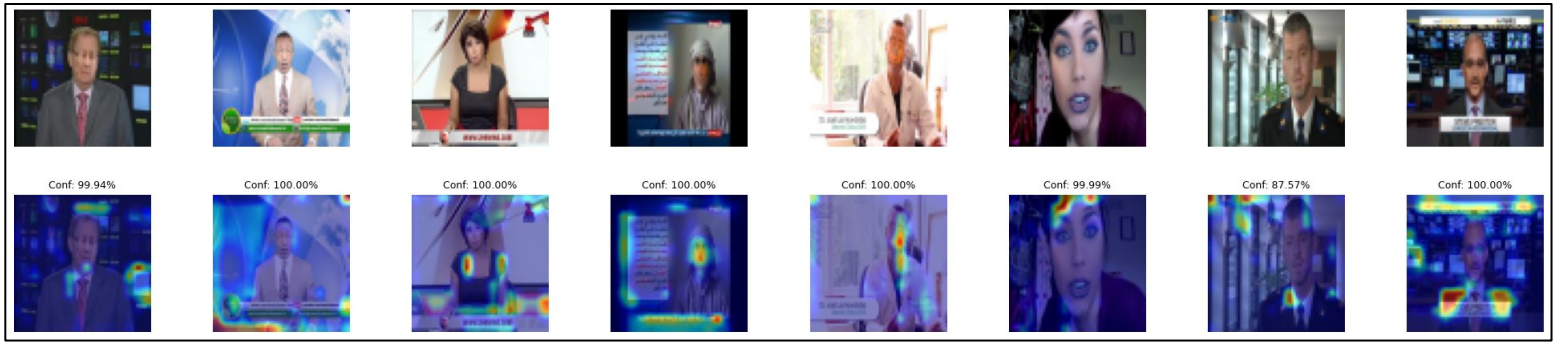


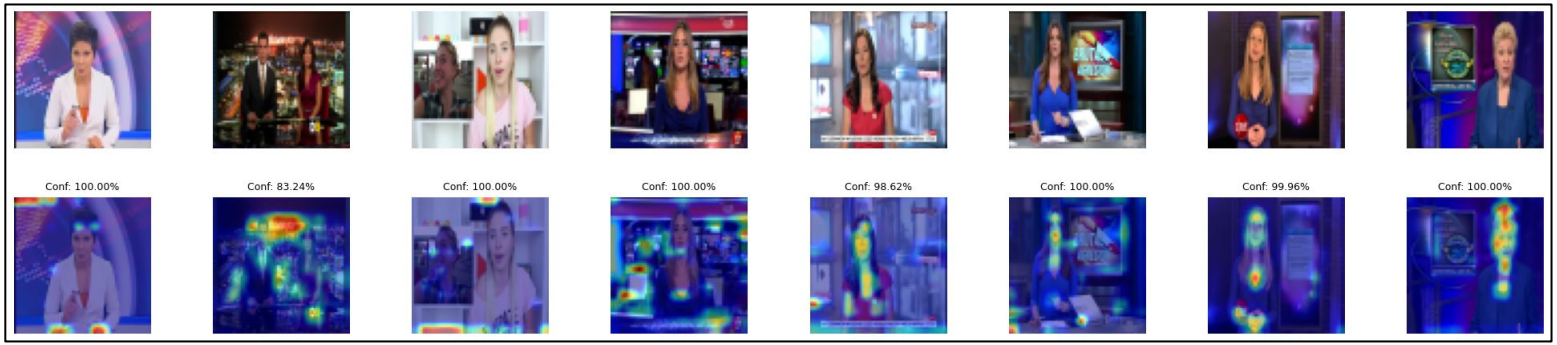
Per-Class Performance (Test Set):
• Real videos: **99.05**% accuracy (4,977/5,025 correct)
• Fake videos: **98.81**% accuracy (4,876/4,935 correct)

Error Analysis:
• False Positive Rate: **0.95**% (48 real videos misclassified as fake)
• False Negative Rate: **1.19**% (59 fake videos missed)

Accuracy = **98.93**%

#3 Attention to the ‘Right’ Details





It’s fun time, **DEMO**!!