TensorFuse-Net: Deepfake Detection via Multi-Domain Tensor Fusion

1 Approach

TensorFuse-Net implements a novel multi-domain tensor fusion approach for deepfake detection with the following key innovations:

- 3D Stacked Tensor Representation (DCT + FFT + Entropy)
- Tucker Decomposition for latent feature compression
- Self-Attention Gating to dynamically weight cross-domain anomalies
- Hadamard Product Binding to preserve high-frequency forensic traces
- ConvNet with Spectral-Augmented Convolutions

2 Architecture Block Diagram

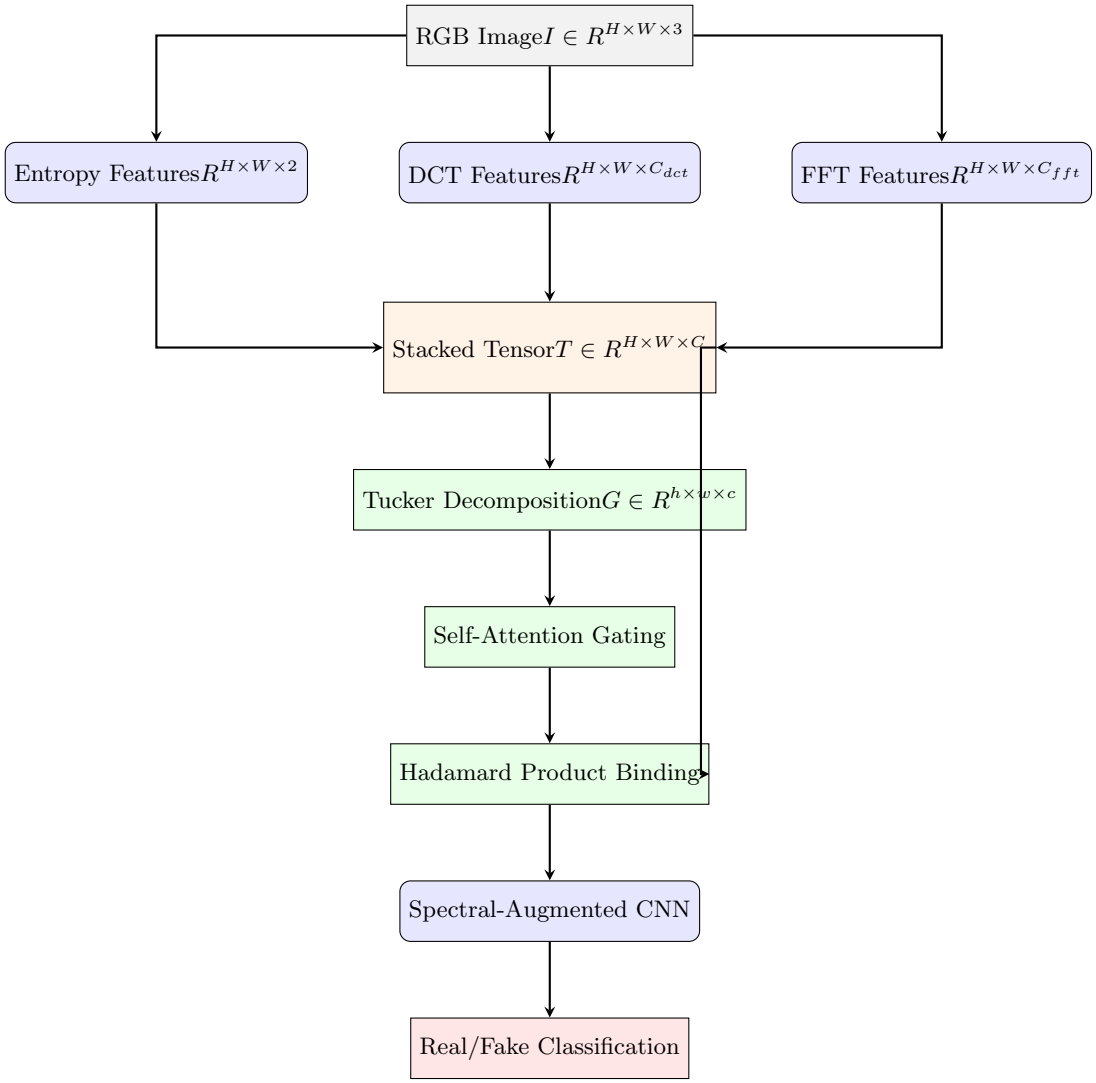


Figure 1: Block Diagram of TensorFuse-Net Architecture for Deepfake Detection

3 Step-by-Step Architecture

3.1 Multi-Domain Feature Extraction

- **Input:** RGB image $I \in R^{H \times W \times 3}$
- **Parallel Feature Extractors:**

- **DCT Features:** Compute block-wise DCT (e.g., 8×8 blocks) $\rightarrow$ DCT-F $\in R^{H \times W \times 64}$. Retain mid/high-frequency coefficients (truncate to C_{dct} channels).
- **FFT Features:** Compute log-magnitude of FFT $\rightarrow$ FFT-F $\in R^{H \times W \times C_{fft}}$. Focus on radial energy distribution (high-freq suppression in deepfakes).
- **Entropy Features:** Compute local Shannon entropy (sliding window) $\rightarrow$ Entropy-F $\in R^{H \times W \times 1}$. Stack with gradient magnitude (Sobel) $\rightarrow$ Entropy-F $\in R^{H \times W \times 2}$.

- **Stack Features:** Concatenate into tensor $T \in R^{H \times W \times C}$, where $C = C_{dct} + C_{fft} + 2$.

3.2 Tucker Decomposition (Latent Correlation Mining)

- Decompose T into core tensor G and factor matrices:

$$T \approx G \times_1 U_1 \times_2 U_2 \times_3 U_3 \tag{1}$$

- $G \in R^{h \times w \times c}$ (compressed core)
- U_1, U_2, U_3 project spatial/channel dims
- **Why?** Extracts cross-domain interactions (e.g., "DCT high-freq + FFT phase inconsistency")

3.3 Attention-Based Gating

- Flatten G to $G_{flattened} \in R^{hw \times c}$
- Compute self-attention:

$$Q = G_{flattened} \cdot W_q \tag{2}$$
$$K = G_{flattened} \cdot W_k \tag{3}$$
$$A = \text{softmax}(QK^T/\sqrt{d}) \tag{4}$$
$$G_{attended} = A \cdot G_{flattened} \tag{5}$$

- Reshape back to $G_{attended} \in R^{h \times w \times c}$

3.4 Hadamard Product Binding

- Reinforce high-frequency anomalies:

$$T_{fused} = T \odot \text{Upsample}(G_{attended}) \tag{6}$$

- Upsample$(\cdot)$ matches spatial dims via bilinear interpolation

4 Results

4.1 Classification Reports

Table 1: Classification Report (Training Set)

	precision	recall	f1-score	support
Real	0.93	0.99	0.96	185
Fake	0.99	0.93	0.96	181
accuracy			0.96	366
macro avg	0.96	0.96	0.96	366
weighted avg	0.96	0.96	0.96	366

Table 2: Classification Report (Test Set)

	precision	recall	f1-score	support
Real	0.92	1.00	0.96	44
Fake	1.00	0.92	0.96	48
accuracy			0.96	92
macro avg	0.96	0.96	0.96	92
weighted avg	0.96	0.96	0.96	92

4.2 Advanced Metrics

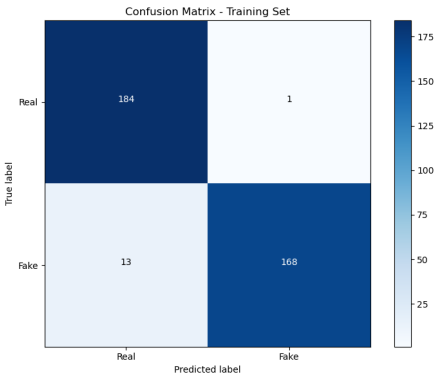
Table 3: Advanced Metrics (Training and Test Sets)

Metric	Training Set	Test Set
Accuracy	0.961749	0.956522
Balanced Accuracy	0.961386	0.958333
Precision	0.994083	1.000000
Recall	0.928177	0.916667
F1 Score	0.960000	0.956522
MCC	0.925428	0.916667
Cohen's Kappa	0.923433	0.913208
Log Loss	0.176850	0.225424
ROC AUC	0.997790	0.992424
PR AUC	0.997731	-

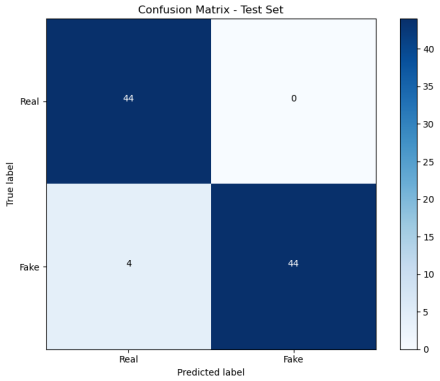
Table 4: Confusion Matrix Details (Test Set)

Class	Class Name	TP	FN	TN	FP	Precision	Recall	Specificity	F1 Score
0	Real	44	0	44	4	1.000000	0.916667	0.916667	0.916667
1	Fake	44	4	44	0	0.916667	1.000000	1.000000	1.000000

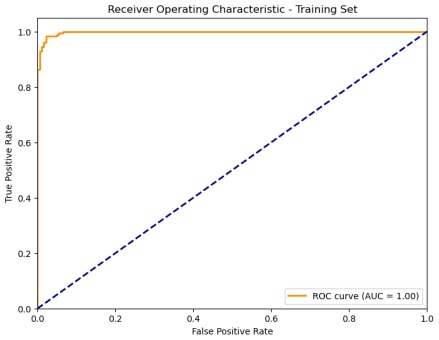
5 Visualizations



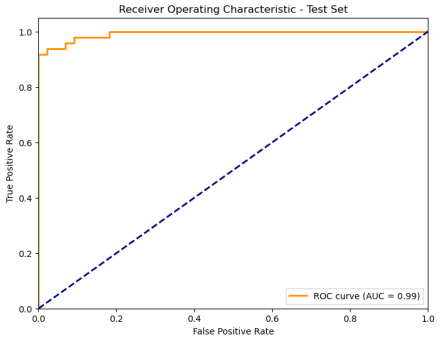
(a) Confusion Matrix (Training Set)



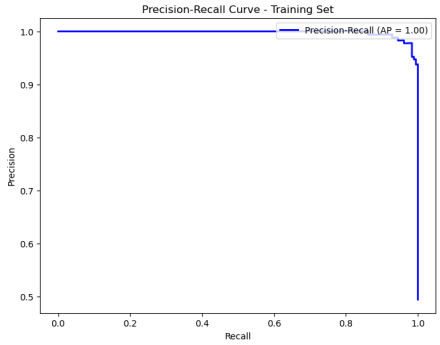
(b) Confusion Matrix (Test Set)



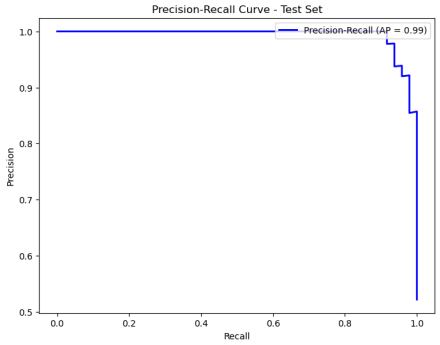
(a) ROC Curve (Training Set)



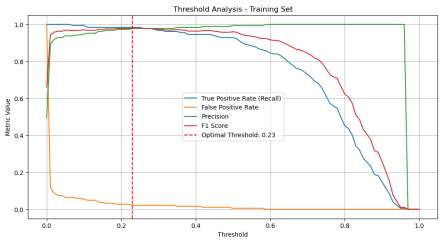
(b) ROC Curve (Test Set)



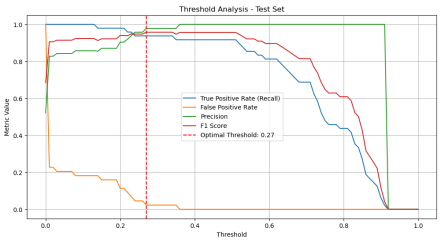
(a) Precision-Recall Curve (Training Set)



(b) Precision-Recall Curve (Test Set)

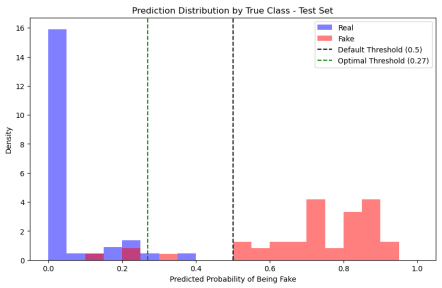


(a) Threshold Analysis (Training Set)



(b) Threshold Analysis (Test Set)

(a) Optimal Decision Threshold (Training Set)



(b) Optimal Decision Threshold (Test Set)

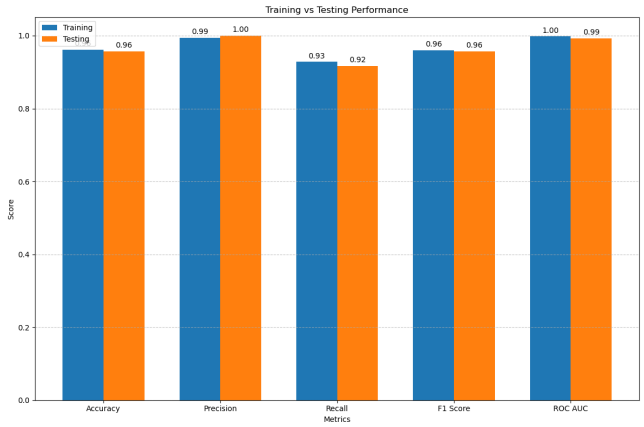


Figure 7: Training vs Test Performance