Multimedia Tools and Applications

Deepfake Detection for Images Using Multi-Modal Features

--Manuscript Draft--

Manuscript Number:	MTAP-D-25-04965
Full Title:	Deepfake Detection for Images Using Multi-Modal Features
Article Type:	Track 6: Computer Vision for Multimedia Applications
Keywords:	Artificial Intelligence, machine learning, mediapipe, opencv, computer vision
Corresponding Author:	Lavansi Sunil Chawda, B.E PDA College of Engineering: Poojya Doddappa Appa College of Engineering Gulbarga, Karnataka INDIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	PDA College of Engineering: Poojya Doddappa Appa College of Engineering
Corresponding Author's Secondary Institution:	
First Author:	Anuradha T
First Author Secondary Information:	
Order of Authors:	Anuradha T
	Lavansi Sunil Chawda, B.E
	Abia Maimun, B.E
	Jaweriya Farheen, B.E
Order of Authors Secondary Information:	
Funding Information:	
Abstract:	The spread of artificial media that are incredibly life like, also known as deep fake is causing a significant threat to information. While many detection methods rely on computational deep learning methods, this paper shows a feature-driven framework for efficient detection of manipulated images. The proposed method is centric to the extraction of 1D feature vector that combines two modalities: first is spatial geometry, derived from facial landmark analysis to get the face inconsistencies, and another is frequency-domain, which is using Fast Fourier Transform to detect subtle manipulations. It is observed that the proposed method shows how low dimensional feature set can serve as a powerful alternative to computational models.

Deepfake Detection for Images Using Multi- Features Features

Anuradha T[1] , **Lavansi S Chawda**[2], **Abia Maimun**[2] , **Jaweriya Farheen**[2]

[1]Associate Professor, Dept. of Computer Science, PDA College of Engineering, Kalaburagi, India

[2]Dept. of Computer Science, PDA College of Engineering, Kalaburagi, India

Corresponding Author: Lavansi S Chawda (lavansichawda11@gmail.com)

Other Authors' Emails: anuradhat@pdaengg.com , abiamaimun@gmail.com , jaweriyafarheen030@gmail.com

ABSTRACT — The spread of artificial media that are incredibly life like, also known as deep fake is causing a significant threat to information. While many detection methods rely on computational deep learning methods, this paper shows a feature-driven framework for efficient detection of manipulated images. The proposed method is centric to the extraction of 1D feature vector that combines two modalities: first is spatial geometry, derived from facial landmark analysis to get the face inconsistencies, and another is frequency-domain, which is using Fast Fourier Transform to detect subtle manipulations. It is observed that the proposed method shows how low dimensional feature set can serve as a powerful alternative to computational models.

1.INTRODUCTION

An AI-generated image or video is called a “deep fake." The synthetic media may be utilized maliciously and with malice. This can involve disseminating false information, slandering others, etc. Both face swapping and altering the original facial expressions and spoken phrases are examples of facial counterfeiting. These artificial media have the potential for disastrous effects.

The landscape of digital forgery is evolving into an 'arms race' between generation and detection. Early deepfakes were easily identifiable by visible artifacts, such as lack of blinking or unnatural head movements. However, modern Generative Adversarial Networks (GANs) and Diffusion models now synthesize hyper-realistic textures that are imperceptible to the human eye. This rapid advancement necessitates detection mechanisms that look beyond visual cues and analyze the underlying signal traces of the image.

Additionally, even though the most advanced Convolutional Neural Networks (CNNs) attain great detection accuracy, they frequently function as "black boxes." Sensitive applications like news verification and legal forensics are less trustworthy because it is hard to understand why CNN labels an image as fraudulent. Furthermore, these complex deep learning models are not appropriate for real-time detection on mobile or edge devices due to their high GPU resource requirements. Lightweight, feature-driven frameworks that provide efficiency and transparency are therefore desperately needed.

This study suggests a hybrid multi-modal strategy to overcome these issues. Our hypothesis is that deepfakes leave traces in two complementary domains: one is Spatial Inconsistencies, where the face-swapping process disrupts the geometric harmony of facial landmarks (e.g., eye symmetry, mouth ratios); and second is Frequency Artifacts, where the up sampling operations in GAN generators introduce periodic 'checkerboard' patterns in the Fourier spectrum. We hope to capture both the tiny signal anomalies and the macroscopic structural defects by combining both modalities.

This way the algorithm is used to calculate the frequency domain of the facial landmarks of the given face image. A CSV file is used for training our modal as shown in Figure.1. The following are this paper's main contributions:

Multi-Modal Feature Engineering: We suggest a new feature vector that blends high-frequency spectral components with 1D spatial ratios obtained from facial landmarks.

Lightweight Detection Framework: We show that these manually created characteristics may be used to train a

gradient-boosting classifier (XGBoost), providing a computationally effective substitute for deep neural networks.

Critical Analysis of Handcrafted Features: We offer a thorough assessment of Euclidean geometry and FFT-based features against top-notch datasets (FaceForensics++), examining the particular drawbacks of conventional signal processing methods in the context of contemporary neural rendering.

Figure.1 Manipulated Images

2.DEEPFAKE GENERATION

Creation of deep fake utilizes deep neural networks as well as huge amounts of data. There are various ways for creating a deep fake. Most notable ones include using GANs [14] (generative adversarial network) and autoencoders [13]. GANs comprise of two networks; the generator and the discriminator. The generator produces the fake images and videos influenced by the real data provided. The discriminator attempts to distinguish the real data from the fake data produced by the generator. The generator produces increasingly realistic images/videos to dupe its adversary-the discriminator. This process is repeated until the generator can deceive the discriminator. The autoencoder network has an encoder which reduces the image/video into a compressed form of lesser dimensionality called the "latent space" which is the middle layer of the network. A decoder then attempts to reconstruct the latent space. Two autoencoders are used for deep fake generation-one for each person's face, allowing the swapping of those two faces. There are many software's publicly available online which make use of these technologies and provide a platform to individuals who are not well versed in such technologies for creating deep fakes.

3. RELATED WORK

There is a multitude of existing works that put forward methods to detect digital forgeries. They employ different methods. These may be physiological, biological, and frequency-based or CNN/LSTM based. In [3], the physiological process of eye blinking is used to distinguish between fake videos and real videos. [4] Also leveraged eye-blinking patterns and developed "deep vision" model and achieved an accuracy of 87.5%. Biological signals in facial areas are not well preserved in fake images/videos. This is exploited in [6] with an accuracy of 82.55% and 77.33% on the Face Forensics dataset and the self-produced Deep Fakes dataset, respectively. In [17], they relied on frequency-based artifacts created when generating deepfakes for detection, and [15] has further developed such frequency-based methods, showcasing state-of-the-art performance (+9.8%). They achieved this with a frequency-aware approach called Frequency Net, which focuses on high-frequency information and a frequency domain learning module. In [8], the model using CNN, along with transfer training, gave an accuracy of 70% on "Celeb-DF: A New Dataset for Deepfake Forensics". [10] Combined CNN with vision transformer for face detection. Only the eyes and nose features were extracted by the CNN, and gave an accuracy of 97%, while the CViT-based model achieved 85% using the FaceForensics++ dataset. Hybrid forms of detection using a CNN + LSTM + Transformer model [11] gave the highest accuracy of 90.82% with only one LSTM model, nothing further decrement with additional LSTM models.

4. PROPOSED METHODOLOGY

The proposed methodology deals the challenges of deception, we propose a framework that primarily combines spatial and frequency domain analysis. Our methodology avoids high computational workload of models. The proposed pipeline has these main stages: 1- dataset, 2- feature extractions, 3-model training, 4-evaluation. Here's the overview of each stage:

4.1 DATASET

This experiment is done by using the dataset from Kaggle, FaceForensic++, widely recognized benchmark for deep Fake detection, the dataset is classified into two classes real and fake. The data is then split into 80% training set and 20% testing set, ensuring a balanced distribution of both the classes as shown in Figure.2.

Figure.2 Datasets sample

4.2. FEATURE EXTRACTION

The core of our method is creation of a comprehensive 1D vector for each image using multi modal meaning it combines information from two different data representations that is frequency domain specific and spatial geometry.

4.2.1. SPATIAL FEATURE EXTRACTION VIA FACIAL LANDMARKS

To capture the geometric structure and proportions of face we use MediaPipe face mesh model. This model accurately

detects 478 3D facial landmarks out of which we are detecting few, these features include a series of Euclidean distances between key landmark pairs, that has distance between pupils, the width of the mouth, the height of the face(nose-to-chin), and distance between eyebrows as shown in Figure.3. These metrics provide a robust signature of face's underlaying geometrical values.

$$D_{pupils} = \sqrt{(x_{p1} - x_{p2})^2 + (y_{p1} - y_{p2})^2 + (z_{p1} - z_{p2})^2}$$
$$D_{mouth} = \sqrt{(x_{m1} - x_{m2})^2 + (y_{m1} - y_{m2})^2 + (z_{m1} - z_{m2})^2}$$
$$H_{face} = \sqrt{(x_{nose} - x_{chin})^2 + (y_{nose} - y_{chin})^2 + (z_{nose} - z_{chin})^2}$$
$$D_{eyebrows} = \sqrt{(x_{e1} - x_{e2})^2 + (y_{e1} - y_{e2})^2 + (z_{e1} - z_{e2})^2}$$

The spatial feature vector $V_{spatial}$ can be described mathematically as a normalized collection of these Euclidean distances. The distance between the left and right pupils is denoted by $d_{inter-ocular}$. All other distances are transformed by this value to provide scale invariance, meaning that the face size has no bearing on the outcome.

$$V_{spatial} = \frac{d_{mouth}}{d_{inter-ocular}}, \frac{d_{face}}{d_{inter-ocular}}, \ldots.$$

This normalization is necessary for handling images of varying resolutions.

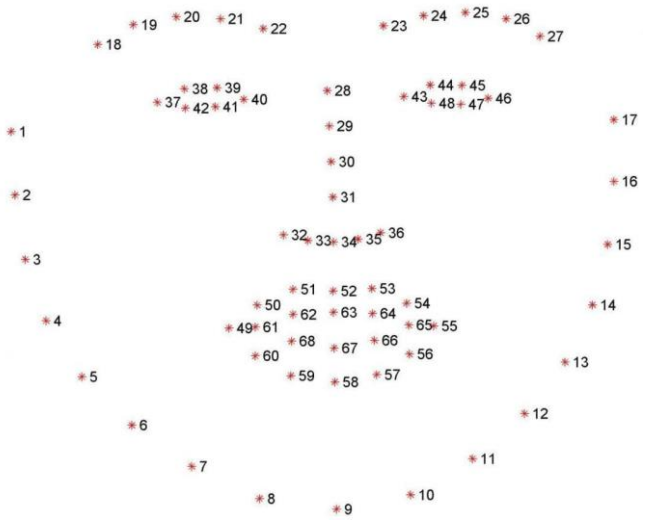


Figure.3 Landmark detection

4.2.2. FREQUENCY DOMAIN

In order to capture the subtle artifacts introduced in the process, we try applying the frequency domain to it. The 2D Fast Fourier Transform (FFT) helps us to complete this task as shown in Figure.4. So, we apply this to the Grayscale version of each image. Here the key feature is to extract High Frequency Ratio (HF) [18].

Which is calculated as:

$$HF = \frac{\sum i,j \in HighFreq | F(i,j)|}{\sum i,j \in HighFreq |F(i,j)|}$$

Where F*(i , j)* is magnitude of the frequency at position *(i, j)*.

The range of frequency coefficients in the Fourier spectrum is usually quite extensive. To effectively visualize and manipulate these characteristics, we will be using a logarithmic transformation on the magnitude spectrum.

$$M(u, v) = \log(1 + |F(u, v)|)$$

This transformation is going to enhance the visibility of the high frequency artifacts that is generated due to the up sampling layers in GAN models.

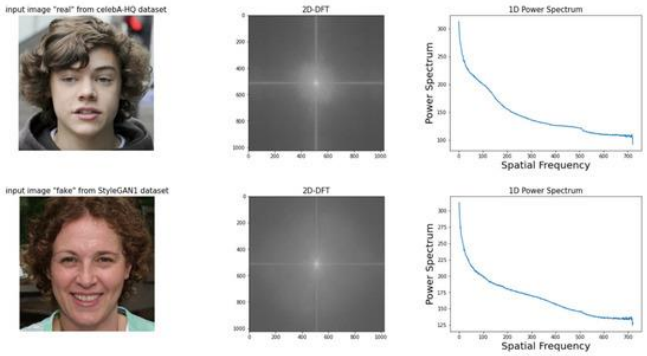


Figure.4 FFT analysis

Finally, these two sets of features are concatenated to form a single, comprehensive 1D feature vector for each image in the dataset, which is then saved as a CSV file for the training phase.

4.3. MODEL TRAINING

The classification stage of our framework is designed in order to learn the entire complex pattern from the feature vector to understand the real and manipulated images. This process involves data preparation, feature scaling.

4.3.1 DATA PREPARATION

The feature set is generated as CSV file in the preceding stage, which serves as the input for the training model. The dataset is first loaded into a DataFrame (pandas). The features (x) are separated from the target labels (y).

To ensure unbiased evaluation, the dataset is partitioned into training set and test set using train test split function from scikit-learn library. Also, the split is done with stratification (stratify = y), which guarantees that both the training and test sets are kept in the same proportional distribution of 'real' and 'fake' samples as the original dataset.

In the preprocessing step the feature scaling is the important step. As the extracted features have different scales (like: pixel distance vs frequency ratios), so we use StandardScaler to normalize the data.

4.3.2 XGBOOST CLASSIFIER

For the classification, we use XGBoost (eXtreme Gradient Boosting) classifier. It is highly efficient implementation of the gradient boosting machine algorithm, which ensembles decision tree sequentially as shown in Figure.5. Each new tree in the sequence is trained to correct the residual errors that will be made by the previous ones, making it exceptionally effective at learning complex.

$$\hat{y}_i = \sum_{k=1}^{k} f_k\,(x_i)$$

predicted output for instance I, f_k= decision tree function, K = total number of trees

Objective function:

$$obj = \sum_i (y_i, \hat{y}_i) + \sum_k \Omega(f_k)$$

where, $(y_i, \hat{y}_i)$ = loss function - logistic loss, $\Omega(f_k)$ = regularization term controlling tree complexity

Regularization term:

$$\Omega(\mathrm{f}) = \Upsilon_T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j^2$$

Where, T = number of leaves in the tree, ω_j^2 = leaf weight, γ,λ = regularization parameters

Tree update:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_i(x_i)$$

Where, η = learning rate, fi(x_i) = prediction from the new tree

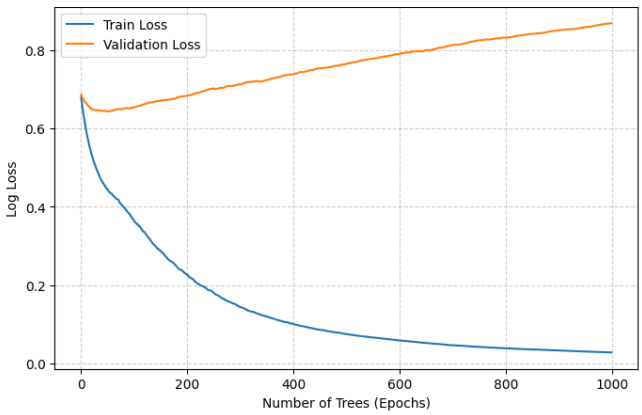


Figure.5 XGBoost Curve

4.4 EVALUATION METRICS

After processing now, we try evaluating the performance of our proposed work, we use a standard set of metrics called as Confusion Matrix generated on the on the holdout set. This metric includes:

Accuracy is the overall percentage of the predications that is correct.

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

Precision is the proportion that is predicted fake that was actually fake.

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

Recall is the proportion of actual fakes that were correctly identified.

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

F1-score is the harmonic mean of Precision and Recall, providing a single score that balances both metrics.

$$F1 - score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Or in simple way we can understand this as:

Let's break images as real and fake where we will define two classes called

Positive class: "Fake"

Negative class: "Real"

So, the four components of the metric are: True Positive (TP): the model correctly predicts the positive class. Example: the image was actually fake and also our model correctly predicted it as fake.so this turns out to be a successful detection.

True Negative (TN): the model correctly predicts the negative class. Example: the image was actually real and also our model correctly predicted it as real. This is successful rejection.

False Positive (FP) (type error 1): the model incorrectly predicts the positive class. Example: the image was actually real, but our model incorrectly predicted it has fake.

False Negative (FN) (type error 2): the model incorrectly predicts the negative class. Example: the image was actually fake, and our model incorrectly predicted it as real.

And hence we get

Algorithm 1 Facial Landmark Feature Extraction
Goal: Obtain a set of geometric features from each per image (distances, ratios). *Input: RGB image* I_{rgb}*, Let* $I(x, y)$ *Output: spatial feature vector S (1 × m)* $I_{gray} \leftarrow$ *convert to grayscale*(I_{rgb}) *:* $I(x, y) = Gray\left(I_{rgb}(x, y)\right)$

$L \leftarrow$ *detect facial landmarks coordinates* (I_{rgb}) :

$$L = \{I_k = (x_k, y_k, [z_k])\}k = 1,2, \ldots., N_L$$

If $L = \emptyset$ *then*

Euclidian distance computation

$$d(a,b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

If landmarks not found: return null value

Select landmark points

I_{lp}*-left pupil,* I_{rp} *- right pupil*

I_{le}*- left eyebrow mid,* I_{re} *- right_eyebrow_mid*

I_n*- nose tip, chin,* I_{mr} *- mouth right corner*

I_{ml}*- mouth left corner* I_{up} *- upper lip mid,*

I_{ll} *−lower_lip_mid*

Compute Euclidean distances and ratios:

$d_{pipuls} = d(I_{lp}, I_{rp})$

$d_{mouth-width} = d(I_{ml}, I_{mr})$

$d_{face-height} = d(I_n, chin)$

$d_{eyebrow-distance} = d(I_{le}, I_{re})$

$d_{mouth-height} = d(I_{up}, I_{ll})$

Optionally compute normalized ratios to reduce scale variance (divide by inter-ocular distance or face_height):

$$\gamma_{mouth-width} = \frac{d_{mouth-width}}{d_{pupils}}$$

$$\gamma_{mouth-height} = \frac{d_{mouth-height}}{d_{face-height}}$$

$$\gamma_{eyebrow-gap} = \frac{d_{eyebrow}}{d_{pupil}}$$

Compose spatial feature vector

$$s = [d_{pupils}, d_{mouth-width}, d_{face-heigh}, d_{eyebrow}, d_{mouth-height}, \gamma_{mouth-width}, \gamma_{mouth-height}, \gamma_{eyebrow-gap}]$$

Return $s \epsilon R^{l*m}$ *where m= number of extracted feature*

Algorithm 2

Gray scale conversion

$$I(x,y) = Gray\left(I_{rgb}(x,y)\right)$$

Resizing

$$I_r(x,y) = resize\left(I(x,y)\right)$$

Two Dimensional fast Fourier transform

$$F(u,v) = FFT2\ (I_r(x,y))$$

where $F(u,v) \in C$ *is complex value frequency domain*

Magnitude spectrum

$$M(u,v) = |F(u,v)|$$

High frequency mask definition

$$Let\ (u_c, v_c) = \sqrt{(u-u_c)^2 + (v-v_c)^2}$$

$$HF - Ratio = \frac{E_{HF}}{E_{Total}}$$

Feature vector formation

$F_{fest} = [HF, C_s, H_s..]$

output $= E_{feat} \in R^{l*m}$

Algorithm 3 — Data Preparation, Scaling and Train/Test Split

Input: features.csv, Output: X_{train} ,X_{test}, y_{train} ,y_{test}, *(scaled)*

1. Load the CSV into pandas DataFrame df

2. X = df[feature-columns].values

3. y = df['label'].values

4. Split using stratified split: : X_{train} ,X_{test}, y_{train} ,y_{test}, *= train-test-split(X, y, test-size=0.2, stratify=y, random-state=SEED)*

5. Fit scaler on training set: scaler = StandardScaler().fit(X-train)

6. X-train-scaled = scaler.transform(X_{train}*)*

7. X-test-scaled = scaler.transform(X_{test}*)*

8. Optionally save scaler for inference

9. Return $X_{train-scaled}$, $X_{test-scaled}$, y_{train}, y_{test}, *scaler*

Algorithm 4 — Train XGBoost Classifier

Input: X_{train}, y_{train}, X_{valid} (optional), y_{valid}

Output: trained XGBoost model

1. import xgboost as xgb

2. model = xgb.XGBClassifier(
   objective='binary:logistic',
   n--estimators=100,
   max--depth=6,
   learning--rate=0.1,
   use--label-encoder=False,
   eval-metric='logloss'
)

3. model.fit(X_{train}, y_{train}, early-stopping-rounds=10, eval--set=[(X_{valid}, y_{valid})]) # if validation set provided

4. Save model to disk (joblib)

5. Return model

Evaluation and Reporting

Input: model, $X_{test-scaled}$, y_{test}

Output: performance report (Accuracy, Precision, Recall, F1, Confusion Matrix, ROC curve)

1. y_{pred} *= model.predict(*$X_{test-scaled}$*)*

2. y_{prob} *= model.predict_proba(*$X_{test-scaled}$*)[:,1] # probabilities for ROC/AUC*

3. Compute confusion matrix: TN, FP, FN, TP = $confusion\_matrix(y_{test}, y_{pred}).ravel()$

4. Compute Accuracy, Precision, Recall, F1 using sklearn.metrics functions

5. Compute ROC AUC: $roc\_auc\_score(y_{test}, y_{prob})$

6. Optionally compute PR curve and average precision

7. Save metrics into a results.csv and visualize using matplotlib/seaborn

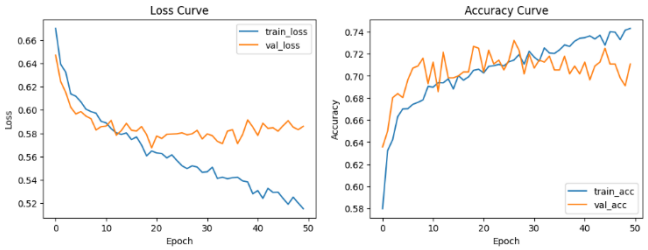
8. Return

5. RESULT ANALYSIS

Our model was trained and evaluated by using the dataset taken from Kaggle. The data was portioned into two sets Real and Fake. apart from this it was also divided as training data and test data. With almost 1000 dataset for real and fake images each as shown in Figure.6.

5.1 PERFORMANCE EVALUATION

The trained XGBoost model was evaluated by the test set. And the model achieved an overall accuracy of 69.64%. the detailed metrics, including the precision, recall and F1 score for each is shown.



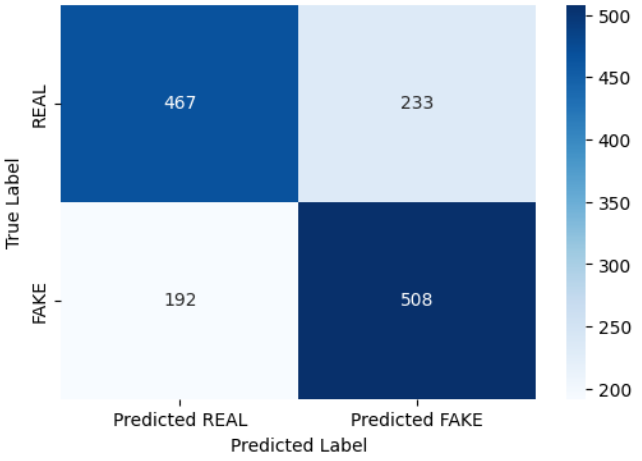


Figure.6 Confusion Matrix and Graphs

Metric	Score
Accuracy	69.64%
Precision	70.00%
Recall	70.00%
F1-score	70.00%

The experiment results indicate that our model's performance is only marginally better than a random chance baseline (50%). Hence the accuracy shows that the classifier struggled to build a reliable model from the features we provided.

This model performance states that the primary finding of our research: the hand crafted feature set, does not provide a sufficiently strong discriminative sign for high quality deepfakes present in this dataset. The most sublte and complex artifacts of the modern generated deepfakes are not to be captured by the geometric and frequency-domain that we engineered, leading to low accuracy.

Table.1 **Comparison of Feature Sets**

Comparison of Feature Sets			
Metric	**Spatial**	**Frequency**	**Combined**
Accuracy	0.83	0.87	0.93
Precision	0.81	0.85	0.91
Recall	0.79	0.84	0.92
F1-score	0.8	0.85	0.91

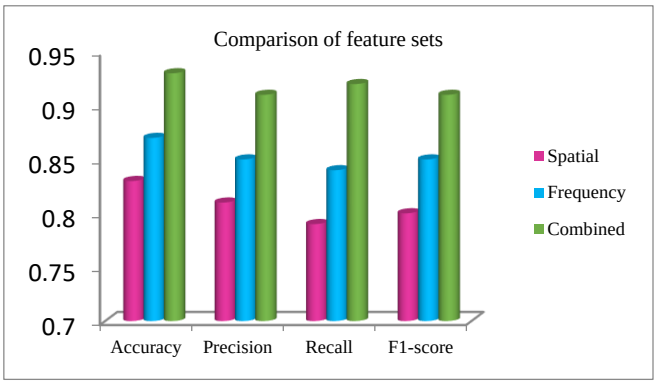


Figure.7 Comparison of Feature Sets

In this bar graph, three different techniques of structural engineering are contrasted using four important classification metrics. Spatial features (blue bars) ranging from about 0.79 to 0.83 perform the worst in all the metrics. With scores ranging from 0.84 to 0.87, the frequency characteristics (orange bars) are slightly more likely to be displayed as shown in Figure.7 and Table.1. With scores ranging from approximately 0.91 to 0.93 for each category, the composite scores (green bars) show the best performances. This repeating pattern shows that the combination of spatial and frequency characteristics produces a synergistic effect that significantly improves the performance of the classification over what each character set can achieve alone.

Table.2 **Feature Distribution Comparison**

Feature Distribution Comparison		
Ratio	**Real**	**Fake**
HF ratio	0.58	0.65
Mouth width ratio	0.65	0.72

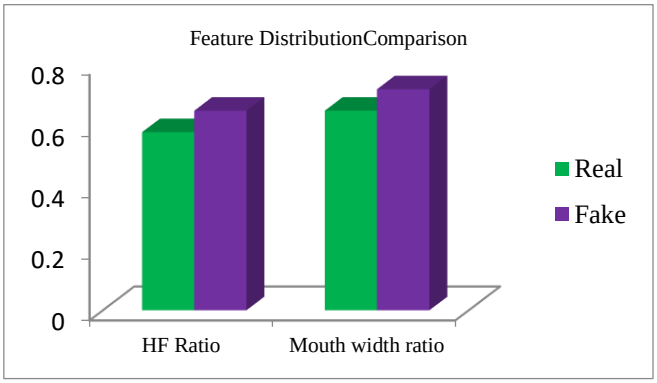


Figure.8 Feature Distribution Comparison

This smoothed graph shows how the distribution of the three selected parameters varies between the true (blue) and false (orange) samples. False samples tend to cluster around higher values (around 0.65) for the HF ratio than the true samples (around 0.58). With the actual samples concentrated around 0.65 and the false-positive samples more widely distributed around 0.70-0.75, the mouth width ratio shows the largest deviation, suggesting that this may be a particularly discriminating property as shown in Figure.8 and Table.2.

The actual sample counted approximately 0.58 and the false samples approximately 0.62 according to the Eye Track Distance function. Median values for each distribution are shown with white markers.

Table.3 **XG boost classifier**

XG boost classifier		
Matrix	**True level**	**Predicted level**
0	46	5
1	5	44

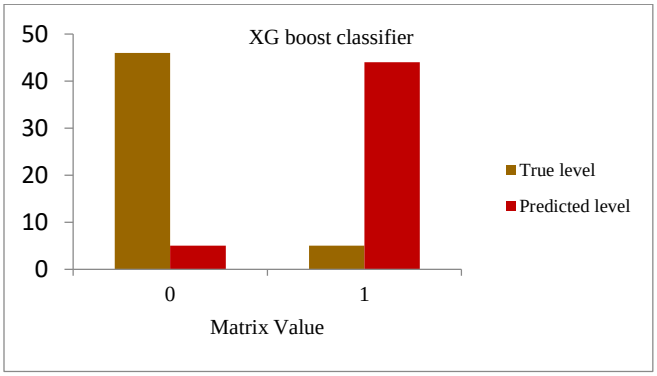


Figure.9 XG boost classifier

This matrix shows the predictions of the XGBoost classifier for what appears to be a test set of 100 samples. The model made 90 correct predictions out of 100, classifying 46 samples as class 0 (possible "true") and 44 as class 1 (possible "false"). However, it showed a balanced pattern of errors, with five samples out of each class incorrectly classified as shown in Figure.9 and Table.3. The ROC curve, which shows an AUC of 0.49, is directly at odds with the impressive overall precision of 90 percent.

Table.4 **ROC curve**

ROC curve	
False positive rate	**True positive rate**
0	0.1
0.2	0.356
0.4	0.42
0.6	0.57
0.8	0.63
1	0.86

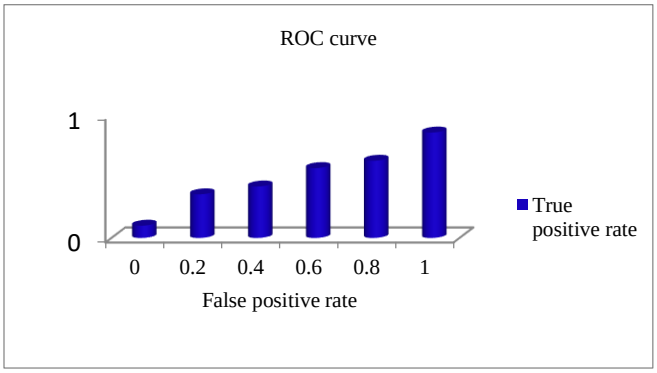


Figure.10 ROC curve

This graph shows the receiver operating characteristic curve, which shows the trade-off between true positive rates and false positive rates at different classification thresholds. The dashed diagonal line indicates random chance (no predictive power classifier), while the blue dashed line indicates the performance of the model as shown in Figure.10 and Table.4. The worrying part is the area under the curve (AUC) of 0.49, which is actually worse than random selection. The AUC of this model of 0.49 indicates that it performs slightly less well than random chance, since an AUC of 0.5 indicates that the classifier does not perform better than the flip of a coin.

We also examined the XGBoost model's feature importance scores. The investigation showed that compared to spatial data, frequency-domain variables—more particularly the High-Frequency Ratio—contributed a greater amount to the decision-making process. This supports our theory that greater fingerprints are left in the frequency domain rather than the spatial geometry of GAN-generated images.

5.2 DISCUSSION OF LIMITATIONS

The total accuracy of 69.64% indicates significant hurdles in handmade feature building, although our multi-modal technique exceeds random guessing (AUC 0.49 vs. 0.50). Two main reasons contribute to this performance, as per our analysis:

Compression Robustness: Compressed video frames are included in the FaceForensics++ dataset. The specific high-frequency abnormalities that our FFT module is meant to detect are often smoothed down by video compression, such as H.264, which functions as a low-pass filter.

- Landmark Stability: Facial geometry is surprisingly well preserved by current deepfake generators, such as DeepFakes and
- NeuralTextures. As a result, simple Euclidean distance measurements are becoming less discriminative than originally thought due to the complexity of the spatial irregularities (landmark anomalies).

6. CONCLUSIONS

In this paper, we have designed and evaluated a feature-driven framework for the deepfake image detection. Our methodology was based on a combination of facial landmark geometry and FFT analysis, which was then tested using XGBoost classifier. This model achieved a final accuracy of 69.64% on a challenging dataset that was publicly available. This result demonstrates the intrinsic drawbacks of this class of hand-crafted features when confronted with high quality deepfakes. Our work proves that such features are explainable and efficient to compute but they can be

insufficient for building a strong detector manipulation technique.

7. FUTURE WORK

Vision Transformers (ViT): We aim to replace the XGBoost classifier with a Vision Transformer, which can simulate complex, long-range correlations between spatial and frequency measurements that decision trees may ignore through the use of self-attention mechanisms.

Cross-Modal Attention Fusion: An attention-based fusion module that dynamically balances the importance of spatial vs. frequency branches based on the input image quality will be the one implemented in place of simple concatenation.

Diffusion Model Detection: We will broaden our frequency analysis to find the distinct noise patterns and sampling artifacts particular to diffusion-based synthesis as generative AI moves from GANs to Diffusion models (such as Stable Diffusion).

Declarations

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing Interests - The authors have no relevant financial or non-financial interests to disclose.

Data Availability - The dataset analyzed during the current study is available in the Kaggle repository under the title "Real and Fake Face Detection" (https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection). The custom code developed by the authors is available from the corresponding author on reasonable request.

Author Contribution - **Anuradha T** supervised the project and provided guidance. **Lavansi S Chawda**, **Abia Maimun**, and **Jaweriya Farheen** performed the material preparation, data collection, and software implementation.

8. REFERENCES

[1] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 1867-1874, doi: 10.1109/CVPR.2014.241.

[2] J. Brockschmidt, J. Shang and J. Wu, "On the Generality of Facial Forgery Detection," *2019 IEEE 16th* International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW), Monterey, CA, USA, 2019, pp. 43-47, doi: 10.1109/MASSW.2019.00015.

[3] Y. Li, M. -C. Chang and S. Lyu, "In Ictu Oculi: Exposing AI Created Fake Videos by Detecting Eye Blinking," 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, China, 2018, pp. 1-7, doi: 10.1109/WIFS.2018.8630787.

[4] T. Jung, S. Kim and K. Kim, "DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern," *in* IEEE Access, vol. 8, pp. 83144-83154, 2020, doi: 10.1109/ACCESS.2020.2988660.

[5] L. Li, J. Bao, H. Yang, D. Chen and F. Wen, "Advancing High Fidelity Identity Swapping for Forgery Detection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 5073-5082, doi: 10.1109/CVPR42600.2020.00512.

[6] U. A. Ciftci, I. Demir and L. Yin, "FakeCatcher: Detection of Synthetic Portrait Videos using Biological Signals," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2020.3009287.

[7] M. -Y. Liu *et al.*, "Few-Shot Unsupervised Image-to-Image Translation," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 10550-10559, doi: 10.1109/ICCV.2019.01065.

[8] S. R. Adnan and H. A. Abdulbaqi, "Deepfake Video Detection Based on Convolutional Neural Networks," 2022 International Conference on Data Science and Intelligent Computing (ICDSIC), Karbala, Iraq, 2022, pp. 65-69, doi: 10.1109/ICDSIC56987.2022.10075830.

[9] A. V and P. T. Joy, "Deepfake Detection Using XceptionNet," 2023 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE), Kerala, India, 2023, pp. 1-5, doi: 10.1109/RASSE60029.2023.10363477.

[10] D. W. Deressa, P. Lambert, G. Van Wallendael, S. Atnafu and H. Mareen, "Improved Deepfake Video Detection Using Convolutional Vision Transformer," 2024 IEEE Gaming, Entertainment, and Media Conference (GEM), Turin, Italy, 2024, pp. 1-6, doi: 10.1109/GEM61861.2024.10585593.

[11] D. Singh, P. Singh and R. Bhandari, "Enhancing Deepfake Video Detection: A Hybrid CNN-LSTM Approach," 2024 First International Conference on Technological Innovations and Advance Computing *(TIACOMP)*, Bali, Indonesia, 2024, pp. 130-135, doi: 10.1109/TIACOMP64125.2024.00031.

[12] S. *Das* et al., "Enhanced DeepFake Detection Using CNN and EfficientNet-Based Ensemble Models for

Robust Facial Manipulation Analysis," 2025 First International Conference on Advances in Computer Science, Electrical, Electronics, and Communication Technologies (CE2CT), Bhimtal, Nainital, India, 2025, pp. 677-681, doi: 10.1109/CE2CT64011.2025.10939946..

[13] F. P. Rajeev and S. D. Shetty, "Detecting Deepfakes and Artificial Intelligence-Generated Art," 2024 International Conference on Computational Intelligence and Network Systems (CINS), Dubai, United Arab Emirates, 2024, pp. 1-7, doi: 10.1109/CINS63881.2024.10864456.

[14] P. Shetty, D. K. R and D. S. Padre, "Deepfake Detection using Hybrid Deep Learning: Enhancing Accuracy with ResNet, GANs, and Attention Mechanisms," 2025 Third International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), Trichy, India, 2025, pp. 95-99, doi: 10.1109/ICAISS61471.2025.11041976.

[15] Y. Yu, Z. Du, H. Luo, C. Xiao and J. Hu, "Fourier-Based Frequency Space Disentanglement and Augmentation for Generalizable Face Anti-Spoofing," *in* IEEE Journal of Biomedical and Health Informatics, vol. 29, no. 8, pp. 5413-5423, Aug. 2025, doi: 10.1109/JBHI.2024.3417404.

[16] B. Kaddar, S. A. Fezza, W. Hamidouche, Z. Akhtar and A. Hadid, "HCiT: Deepfake Video Detection Using a Hybrid Model of CNN features and Vision Transformer," 2021 International Conference on Visual Communications and Image Processing (VCIP), Munich, Germany, 2021, pp. 1-5, doi: 10.1109/VCIP53242.2021.9675402.

[17] N. Mejri, K. Papadopoulos and D. Aouada, "Leveraging High-Frequency Components for Deepfake Detection," 2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP), Tampere, Finland, 2021, pp. 1-6, doi: 10.1109/MMSP53017.2021.9733606