A Novel Unified Approach to Deepfake Detection of Images

Lord Sen
Computer Science and Engineering
NIT Rourkela
Rourkela, Odisha
lordsen3008@gmail.com

Shyamapada Mukherjee
Computer Science and Engineering
NIT Rourkela
Rourkela, Odisha
mukherjees@nitrkl.ac.in

Abstract—The advancements in the field of AI is increasingly giving rise to various threats. One of the most prominent of them is the synthesis and misuse of Deep Fakes. To sustain trust in this digital age, detection and tagging of deepfakes is very necessary. In this paper, a novel architecture for Deepfake detection in images is presented. The architecture uses cross-attention between spatial and frequency domain features along with a blood detection module to classify an image as real or fake. This paper aims to develop a unified architecture and provide insights into each step. It is trained on two small datasets with 200 and 400 images respectively. On comparative analysis, our model was better than the other possibilities. Further, there was an increment in accuracy of 4.29% and 4.60% upon adding 200 images to the dataset. This shows that, if trained on a large dataset and hyper-parameters optimized, the performance will increase significantly.

Index Terms—DeepFake detection, Cross Attention, Fourier Transform etc.

I. INTRODUCTION

The Deepfake media is becoming an increasing threat in today's digital world, where trust in what we see and hear online is decreasing rapidly. These AI-generated fake videos, images, and audio can create realistic but entirely false depictions of people. This technology can be misused to spread lies about public figures, manipulate opinions, or even ruin someone's reputation. As they become easier to create and harder to detect, they challenge our ability to tell what's real from what's not. The rise of deepfakes highlights how important it is to develop tools to identify them and to raise awareness about their potential impact on trust and truth in the digital age.

With the development of Generative Adversarial Networks(GANs) and Auto-encoders the synthesis of hyper-realistic deepfakes have increased and become easy. Variants like CycleGAN and Pix2Pix are used for tasks like translating one image to another (such as changing facial expressions or backgrounds), while StyleGAN is known for generating high-quality, realistic images with precise control over features like facial attributes. StarGAN, on the other hand, enables multi-domain transformations, allowing for complex face-swapping tasks. Additionally, deep CNNs are often employed to manipulate facial features, while newer models using transformer-based attention mechanisms help capture long-range dependencies in images for even more realistic results. These techniques, combined with increasing computational power, make deepfake images harder to distinguish from real ones.

Through this paper a new and unified architecture for deepfake detection will be suggested. The primary focus of this paper is making a unified architecture that is strong and efficient, not increasing the accuracy. The model proposed in this paper is trained on 2 very small datasets one containing 200 images and the other containing 400 images from the FF++ dataset.

The section II of this paper briefly reviews the present methods for deepfake detection in images, videos and audio. In section III, the proposed architecture is discussed. The results obtained and the conclusions drawn with future plans are discussed in section IV and V.

II. LITERATURE REVIEW

In [2] a modified InceptionResNetV2 model based on the Constant-Q Transform (CQT) method to extract deep time-frequency features from audio has been suggested and they have combined it with XceptionNet model for video frames, for deepfake detection in videos. An architecture to detect deepfakes by CNN is presented in [3], [7], audio analysis in [8], cross domain analysis is done in [1], [10] to [16], multi-attentional approach in [6], vission transformer in [9], using Fourier transform in [5]. A live skin detection system in [4]. [1] also presents a trend of mean and corelation coefficient of real and fake images in frequency domain. In this paper, we have found out underlying trends of energy, entropy and power spectral density (PSD).

III. ARCHITECTURE

First, the image is sent to the input layer where it is resized and the pixel values normalized. Then, it is sent to the preprocessing layer where frequency domain analysis is done. Then, to spatial and frequency based encoders. Finally to a cross stream attention fusion layer and then Embedding and token refinement layers and finally classification probability is given as output. Parallely, the normalized image is sent for the detection of blood underlying the skin. These two outputs are combined and the final classification is done.

A. Input Layer

The proposed model takes an RGB image as input, which is first pre-processed to ensure that it is ready for analysis. The image is resized to a fixed size, such as 224×224 in our case, to maintain consistency between all inputs. The pixel values are then normalized, adjusting their range according to the mean and standard deviation values ($x_i \rightarrow \frac{x_i - \mu}{\sigma}$), helping the model learn more effectively, as the values are in range of -1 to 1. Finally, the processed image is converted to a tensor format, which is passed for the next steps in the model pipeline.

B. Preprocessing Layer

To efficiently detect subtle manipulations that are present in deepfake images, the model utilizes frequency domain analysis. In contrast to the spatial domain, which analyzes the pixel arrangements, the frequency domain reveals patterns in textures, edges, and inconsistencies that are often introduced during image manipulation i.e., deepfake image synthesis. The input image is transformed into the frequency domain using techniques such as the Discrete Fourier Transform (DFT). These methods break the image down into its frequency components, separating smooth, low-frequency regions (like backgrounds or skin tones) from sharp, high-frequency details (like edges and textures).

In the model first the image is converted to gray scale to focus more on the intensity, which mostly reflects changes due to manipulation, and reduce computational complexity.

$I_{\text{gray}}(x,y) = 0.2989 I_R(x,y) + 0.5870 I_G(x,y) + 0.1140 I_B(x,y)$

Then Discrete Fourier Transform (DFT) is applied, converting the spatial features to frequencies highlighting complex patterns.

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I_{\text{gray}}(x,y) \cdot e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}$$

where: $F(u,v)$ is the Fourier coefficient at frequency (u,v), $I_{\text{gray}}(x,y)$ is the grayscale intensity at pixel (x,y), M, N are the dimensions of the image.

Then the zero frequency component is shifted to the center for easy analysis.

$F_{\text{shifted}}(u,v) = F\left((u + M/2) \mod M, (v + N/2) \mod N\right)$

The magnitude and phase of frequency are then calculated, normalized and stacked. Magnitude:

$$|F(u,v)| = \sqrt{\text{Re}(F(u,v))^2 + \text{Im}(F(u,v))^2}$$

Phase:

$$\phi(u,v) = \arctan\left(\frac{\text{Im}(F(u,v))}{\text{Re}(F(u,v))}\right)$$

where: $\text{Re}(F(u,v))$ and $\text{Im}(F(u,v))$ are the real and imaginary parts of the Fourier coefficient $F(u,v)$.

Magnitude Normalization:

$$|F_{\text{norm}}(u,v)| = \frac{|F(u,v)| - \min(|F(u,v)|)}{\max(|F(u,v)|) - \min(|F(u,v)|)}$$

Stacking Magnitude and Phase:

$$\text{Features}(u,v) = \begin{bmatrix} |F_{\text{norm}}(u,v)| \\ \phi(u,v) \end{bmatrix}$$

The frequency spectrum is divided into multiple bands using band-pass filters, allowing the model to focus on specific ranges of frequencies. From these bands, important statistical features—such as energy distribution, entropy, and power spectral density (PSD)—are extracted to highlight critical patterns. Energy:

$$E = \sum_{f_1}^{f_2} |X(f)|^2$$

where $X(f)$ is the Fourier transform of the signal at frequency f and $|X(f)|$ is the magnitude of the frequency component. Entropy:

$$H = -\sum_f P(f) \log(P(f))$$

where $P(f) = \frac{|X(f)|^2}{\sum_f |X(f)|^2}$ is the normalized power spectral density, representing the probability distribution of power at frequency f. Power spectral density (PSD):

$$S(f) = \frac{|X(f)|^2}{T}$$

where: $|X(f)|^2$ is the squared magnitude of the Fourier transform $X(f)$ at frequency f and T is the total duration of the signal.
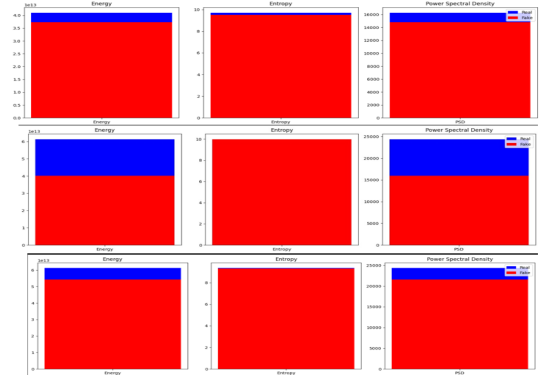

Fig. 1. The graphs of Energy, entropy and PSD of three different sets of real(Blue) and fake(Red) images.

These features are then restructured into feature maps that retain the image's spatial resolution but include multiple channels, each representing a different frequency band. For instance, an image resized to 224×224 might produce feature maps of size $224 \times 224 \times 8$, where the 8 channels correspond to different frequency ranges. After analyzing the plots of these three for various sets of real and fake images, a trend that the values of Energy, entropy and PSD were higher in real images than in fake ones was found (Fig. 1). These frequency feature maps are passed to the next layers for further processing. By focusing on these frequency patterns, the model gains a deeper understanding of the image's underlying characteristics, making it more reliable at identifying fake content.

C. Dual-Stream Feature Encoder

In order to efficiently capture both the spatial details and frequency-based features, the dual-stream feature encoder processes the input through two parallel paths: the Spatial Feature Encoder and the Frequency Feature Encoder. These streams independently extract complementary representations, which are later fused through cross attention to enhance the model's ability to detect deepfake-specific anomalies.

1) Spatial Feature Encoder: The spatial feature encoder takes in the original RGB image ($H \times W \times C$, e.g., $224 \times 224 \times 3$) using a lightweight convolutional neural network (CNN) backbone with the pretrained weights and last few layers removed, such as MobileNetV3, ResNet etc. We have used different models to check which one is giving more accuracy on a small dataset. We have also tried the same using vision transformer and swin transformer in place of these CNN models. These architectures are optimized for efficiency and accuracy in feature extraction. The backbone applies a series of convolutional layers with residual connections, that preserves low-level details while enabling deeper layers to focus on higher-level features.

Convolution Operation:

$$f_{i,j,k} = \sum_{m,n} \sum_{c} I_{i+m,j+n,c} \cdot K_{m,n,c,k}$$

where: $I_{i,j,c}$: Input pixel value at position (i,j) for channel c, $K_{m,n,c,k}$: Kernel weight at position (m,n) for channel c and filter k, $f_{i,j,k}$: Output feature map value at position (i,j) for filter k. Then, ReLU Activation is applied,

$$f'(x) = \max(0, x)$$

Down-sampling is performed via pooling layers, stride-based convolutions to reduce the spatial resolution and increase the depth of the feature map. Pooling Operation(e.g. max pooling):

$$P_{i,j,k} = \max_{m,n} f_{s \cdot i + m, s \cdot j + n, k}$$

where s is the stride. The Global Average Pooling layer is also used:

$$GAP_k = \frac{1}{H \cdot W} \sum_{i=1}^{H} \sum_{j=1}^{W} f_{i,j,k}$$

The output of this stream is a spatial feature map (S) of size $P \times P \times C_2$ where $P \times P$ represents the downsampled resolution, and C_2 is the number of feature channels. The output in our case of MobileNetV3Small with last few layers removed is $56 \times 56 \times 128$.

2) Frequency Feature Encoder: The frequency feature encoder processes the frequency-domain features obtained from preprocessing i.e., the output of preprocessing layer, which have dimensions $H \times W \times C$ (e.g., $224 \times 224 \times 8$ in our case). This stream uses a shallow transformer module, utilizing multi-head self-attention to capture global dependencies and relationships in the frequency data. Specifically, BERT and distilBERT, which applies a bidirectional processing, have been

used and their respective performances have been compared in Table 1.

The frequency feature map is divided into non-overlapping patches of size $k \times k$, which are flattened into tokens, forming a sequence of N tokens.

$$\text{Patch}_{i,j} = \{I_{m,n} \mid m \in [i \cdot k, (i+1) \cdot k), n \in [j \cdot k, (j+1) \cdot k)\}$$

$$t_p = W \cdot \text{Flatten}(\text{Patch}_{i,j}) + b$$

where: W is projection matrix, b is bias vector.

For an input of $H \times H$ and patch size $k \times k$, this results in $N = (H/k)^2$ tokens. Positional embeddings are added to the tokens to retain spatial information.

$$T_p = t_p + E_p$$

where E_p is the positional embedding.

The sequence is then processed through a shallow transformer, where multi-head self-attention captures long-range dependencies and global patterns in the frequency domain.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

$$\text{MHSA}(T_p) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h) W_O$$

where W_O is the output projection matrix.

The output of this stream is a frequency embedding (F) of size $P \times P \times C_3$ (e.g., $56 \times 56 \times 128$), where $P \times P$ represents the reduced spatial resolution, and C_3 is the number of embedding channels. A feed forward Layer is also used.

$$\text{FFN}(x) = \sigma(xW_1 + b_1)W_2 + b_2$$

where W_1, W_2, b_1, b_2 are learned parameters. The, final frequency embedding is:

$$F = \text{Reshape}(\text{Sequence Output})$$

where $F \in \mathbb{R}^{P \times P \times C_3}$.

The spatial encoder excels at identifying localized patterns such as edges, shapes, and textures, while the frequency encoder focuses on global patterns and artifacts, often associated with manipulation. The outputs from these encoders S and F are subsequently fused, enabling the model to leverage both spatial and frequency-based insights for robust deepfake detection. This dual-stream approach ensures that the model effectively analyzes both local and global inconsistencies introduced by image manipulation.

D. Detection of Blood underneath the skin

In this section, an approach for improving blood detection in images by combining features using cross-attention mechanisms has been introduced. Our method begins by extracting a variety of features from the image, such as histograms of the red channel, the a-channel from the Lab color space, the Cr-channel from YCbCr, and texture features obtained through Local Binary Patterns (LBP). These diverse features are then processed through a custom Cross-Attention Layer, which learns the relationships between different features by assigning attention weights. Specifically, we pair the red channel

histogram with the Cr channel histogram, and the a-channel histogram with the LBP histogram. This enables the model to focus on the most relevant parts of each feature, which contribute to detecting blood. The attention mechanism computes attended features for each pair, and these are combined into a single, comprehensive feature vector. The power of this approach is demonstrated by visualizing the attention maps, which highlight the areas in the image that most influence the model's decision. By using feature fusion and cross-attention, our model is able to capture complex patterns in the data, improving its ability to detect blood presence underlying the skin.

E. Cross-Stream Attention Fusion (CSAF)

The Cross-Stream Attention Fusion (CSAF) module integrates spatial and frequency feature maps to produce a unified representation, combining complementary insights from both streams. This fusion mechanism enhances the model's ability to detect deepfake-specific inconsistencies by aligning spatial details with frequency-domain artifacts.

The module receives two inputs: Spatial Feature Map (S), output by the spatial feature encoder, with dimensions $P \times P \times C_2$ (e.g., $56 \times 56 \times 128$) and Frequency Embedding (F), output by the frequency feature encoder, with dimensions $P \times P \times C_3$ (e.g., $56 \times 56 \times 128$).

The fusion process is driven by a cross-attention mechanism, where each stream's feature map is refined by using the complementary features from the other. Firstly, the spatial feature map S is treated as the query, and the frequency embedding F serves as the key-value pair, enabling spatial features to attend to relevant frequency patterns. Similarly, frequency features are updated by attending to spatial information interchanging the query and key-value pairs. The attention operation is mathematically represented as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V$$

where Q, K, and V are the query, key, and value matrices derived from S and F, and d_k is the scaling factor (dimensionality of K).

This bidirectional cross-attention ensures that spatial features guide the interpretation of frequency-based patterns and vice versa, leading to a robust, enriched representation.

The attention-enhanced feature maps from both streams are combined via summation. To refine the fused features, a feed-forward network is applied, ensuring the unified representation captures essential details from both spatial and frequency domains. Additionally, residual connections are incorporated to preserve the original feature information, with the final fused representation computed as:

$$H_1 = CSAF(S, F) + F \text{ and } H_2 = CSAF(F, S) + S$$

$$H_{\text{fused}} = H_1 + H_2$$

where $CSAF(S, F)$ denote that S is treated as query and F as key-value pair.

The CSAF module produces a fused feature map H_{fused} with dimensions $P \times P \times C_f$ (e.g., $56 \times 56 \times 128$), where $C_f = C_2 = C_3$. This unified feature map encodes spatial details and frequency-domain artifacts in a cohesive manner, effectively capturing both localized manipulations and global inconsistencies introduced by deepfake generation techniques.

Spatial features provide fine-grained details, such as texture and edges, while frequency features capture global anomalies and artifacts. The cross-attention mechanism ensures effective communication between these streams, creating a unified representation that is well-suited for downstream tasks like classification and anomaly detection. This design ensures robustness against diverse manipulation techniques and improves overall model performance.

F. Multiscale Patch Embedding

The Multiscale Patch Embedding module plays a vital role in capturing both fine details and broader patterns within the fused feature map (H_{fused}) produced by the Cross-Stream Attention Fusion (CSAF) module. This feature map, with dimensions $P \times P \times C_f$ (e.g., $56 \times 56 \times 128$), is divided into patches of various sizes, such as $k_1 \times k_1$, $k_2 \times k_2$, and so on. Larger patches are used to focus on global structures and long-range dependencies, while smaller patches target localized and intricate details. Each patch is flattened into a token, with the number of tokens (T) at a given scale calculated as $T = \left(\frac{P}{k}\right)^2$. To preserve the spatial context, learnable positional embeddings are added to the tokens. These multi-scale tokens are then processed in parallel using lightweight transformers, each specialized in extracting features at its respective resolution. Finally, the outputs from all scales are fused through weighted summation, producing a unified multi-scale representation. The output of this layer is (B, T_{min}, D), where B = batch size, T_{min} corresponds to the smallest number of tokens across scales and $D = C_f$ is the embedding dimension. This design ensures that the module captures the full spectrum of spatial variations, from subtle texture mismatches to broad structural inconsistencies. By combining local details with global context, the multi-scale patch embedding significantly enhances the model's ability to detect deepfake manipulations effectively and reliably.

G. Class Token Refinement Module (CTRM)

The Class Token Refinement Module (CTRM) is the final step in the architecture, responsible for condensing the rich, multi-scale features into a compact representation for classification. At the heart of this module is a learnable class token ($\mathbf{C} \in \mathbb{R}^{1 \times d_{\text{embedding}}}$), which acts as a global aggregator. This token is added to the sequence of multi-scale feature tokens ($\mathbf{F} \in \mathbb{R}^{T \times d_{\text{embedding}}}$, where T is the number of tokens and $d_{\text{embedding}}$ is the embedding dimension) produced by the previous module, which can be represented as

$$\mathbf{X}_0 = \text{concat}(\mathbf{C}, \mathbf{F}) \in \mathbb{R}^{(T+1) \times d_{\text{embedding}}}$$

This is then processed through a transformer encoder, inside which the class token interacts with all other tokens using
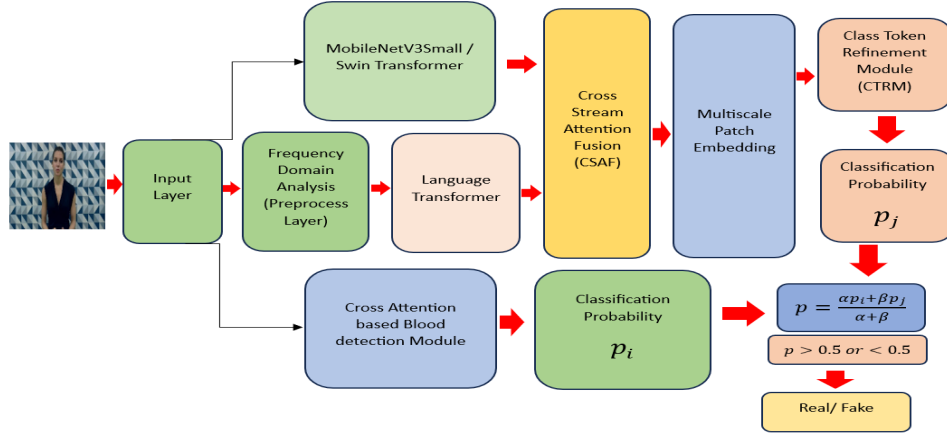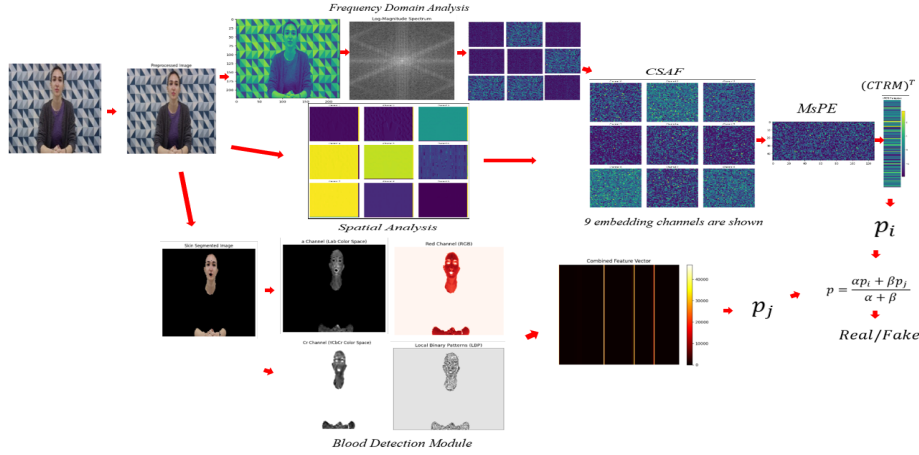
Fig. 2. Our Proposed Architecture-1 and 2

Fig. 3. The figure shows the outputs of various layers, when an image is passed to our proposed architecture.

multi-head self-attention, allowing it to gather both localized and global information critical for deepfake detection.

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{(T+1) \times d_k}$ are the query, key, and value matrices derived from $\mathbf{X}_l$ using learnable projection matrices, $d_k = \frac{d_{\text{embedding}}}{h}$ is the dimension per attention head (with h heads). Then, Multi-Head Aggregation is done:

$$\text{MultiHead}(\mathbf{X}_l) = \text{concat}(\text{head}_1, \ldots, \text{head}_h)\mathbf{W}_o$$

where $\text{head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i)$, $\mathbf{W}_o \in \mathbb{R}^{d_{\text{embedding}} \times d_{\text{embedding}}}$ is the output projection matrix.

Through this process, the class token refines its understanding of the input, distilling the most important features into a single representation. A feed-forward network (FFN) further enhances the class token's features, with residual connections and layer normalization ensuring stability and retaining essential information.

$$\text{FFN}(\mathbf{X}_l) = \text{ReLU}(\mathbf{X}_l\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_{\text{embedding}} \times d_{\text{hidden}}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{\text{hidden}} \times d_{\text{embedding}}}$, d_{hidden} is the intermediate dimension in the FFN. Residual Connections and Layer Normalization:

$$\mathbf{X}_{l+1} = \text{LayerNorm}(\mathbf{X}_l + \text{MultiHead}(\mathbf{X}_l))$$

$$\mathbf{X}_{l+1} = \text{LayerNorm}(\mathbf{X}_{l+1} + \text{FFN}(\mathbf{X}_{l+1}))$$

The result is a refined class token, with dimensions $1 \times d_{\text{embedding}}$, that encapsulates the key characteristics of the input image. This representation is then passed to the final classification layer to determine whether the image is real or a deepfake. By focusing on the most critical aspects of the extracted features, the CTRM ensures that the model achieves high reliability and accuracy in detecting subtle manipulations.

H. Final Classification Head

The Final Classification Head is the last step of the architecture, where the refined class token from the Class Token Refinement Module (CTRM) is transformed into a meaningful prediction. This class token, a compact representation of the input's critical features, is passed through a fully connected layer that maps it to a single value, called the logit, for binary

classification. A sigmoid activation function is then applied to the logit to generate a probability score between 0 and 1, indicating the likelihood of the input being a deepfake.

$$\hat{y} = \text{sigmoid}(\mathbf{C}_{\text{refined}}\mathbf{W}_c + b_c)$$

where: $\mathbf{W}_c \in \mathbb{R}^{d_{\text{embedding}} \times 1}$, $b_c \in \mathbb{R}$, $\hat{y}$ is the predicted probability that the input is a deepfake. Let it be p_i. Let, the classification probablity from the blood detection module be p_j. Then a weighted average of p_i and p_j is taken as: $p = \frac{\alpha p_i + \beta p_j}{\alpha + \beta}$, we are taking $\alpha = 0.6$ and $\beta = 0.4$ heuristically. To make the final decision, a threshold ($p > 0.5$) is applied, classifying the image as either real or fake. The probability score provides a confidence level for the prediction, making it particularly valuable in contexts like digital forensics and content verification, where interpretability and reliability are crucial.

IV. RESULTS

For, training the model two small datasets have been used: Dataset 1, containing 200 images and Dataset 2, containing 400 images which are taken from the FF++ dataset. For evaluation the metric accuracy have been used. The loss function that was reduced is binary crossentropy:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right]$$

Our Proposed Architecture-1 contains MobileNetV3Small and BERT along with the blood detection and Architecture-2 contains SwinTransformer and BERT with blood detection. If we do a comparative analysis on the results found on the two datasets then clearly there is an increase of accuracy and our proposed has the highest accuracy out of all. Moreover, there is a significant rise of 4.29% for Architecture-1 and 4.60% for Architecture-2, when 200 images were added to the dataset. The accuracy values will significantly increase when the hyperparameters are tuned, the models are trained on a large dataset and dropout and regularization techiniques are used in suitable places.

TABLE I
RESULTS

Table SL.No.	Results Table		
	Model	*Dataset 1*	*Dataset 2*
1	MobileNetV3Small	50.23	50.78
2	Swin Transformer	50.45	51.25
3	BERT	50.76	52.00
4	DistilBERT	50.46	51.32
5	MobileNetV3Small and BERT	52.34	53.23
6	MobileNetV3Small and DistilBERT	52.44	53.10
7	SwinTransformer and BERT	52.67	54.45
8	Our Proposed Architecture-1	55.47	59.76
8	Our Proposed Architecture-2	54.42	59.02

[a]Accuracy is written in terms of percentage.

V. CONCLUSION

Through this paper an unified novel approach to multi-modal deepfake detection was introduced. The approach integrates spatial and frequency domain features through cross attention, followed by multi-scale patch embedding and class token refinement module. It also employs a parallel segment for deepfake detection based on under skin blood detection of images. The classification probability obtained from both are combined via weighted average. Based on which the image is classified as real or fake. Despite of the small datasets and no hyper-parameter tuning, our proposed Architecture-1 achieved an accuracy of 59.76% and an increment of 4.29% and 59.02% with an increment of 4.60% for Architecture-2, when the dataset was increased by 200 images. In our future work, we will train and fine-tune the model on large datasets and tune the hyper-parameters to achieve much higher accuracy.

REFERENCES

[1] Amin MA, Hu Y, Li C, Liu B. Deepfake detection based on cross-domain local characteristic analysis with multi-domain transformer. Alex Eng J 2024;91: 592–609.

[2] Elpeltagy, Marwa, et al. "A novel smart deepfake video detection system." International Journal of Advanced Computer Science and Applications 14.1 (2023).

[3] Ahmed, Saadaldeen Rashid, et al. "Analysis survey on deepfake detection and recognition with convolutional neural networks." 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, 2022.

[4] Wang, S.-Y.; Yang, S.-H.; Chen, Y.-P.; Huang, J.-W. Face Liveness Detection Based on Skin Blood Flow Analysis. Symmetry 2017, 9, 305. https://doi.org/10.3390/sym9120305

[5] Dzanic, Tarik, Karan Shah, and Freddie Witherden. "Fourier spectrum discrepancies in deep network generated images." Advances in neural information processing systems 33 (2020): 3022-3032.

[6] Zhao, Hanqing, et al. "Multi-attentional deepfake detection." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[7] Guarnera, Luca, Oliver Giudice, and Sebastiano Battiato. "Deepfake detection by analyzing convolutional traces." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020.

[8] Almutairi, Zaynab, and Hebah Elgibreen. "A review of modern audio deepfake detection methods: challenges and future directions." Algorithms 15.5 (2022): 155.

[9] Wodajo, Deressa, and Solomon Atnafu. "Deepfake video detection using convolutional vision transformer." arXiv preprint arXiv:2102.11126 (2021).

[10] Agarwal, Shruti, et al. "Detecting deep-fake videos from appearance and behavior." 2020 IEEE international workshop on information forensics and security (WIFS). IEEE, 2020.

[11] Chen, Baoying, and Shunquan Tan. "FeatureTransfer: Unsupervised Domain Adaptation for Cross-Domain Deepfake Detection." Security and Communication Networks 2021.1 (2021): 9942754.

[12] Agarwal, Aayushi, et al. "MD-CSDNetwork: Multi-domain cross stitched network for deepfake detection." 2021 16th IEEE international conference on automatic face and gesture recognition (FG 2021). IEEE, 2021.

[13] Liu, Zihan, Hanyi Wang, and Shilin Wang. "Cross-domain local characteristic enhanced deepfake video detection." Proceedings of the Asian Conference on Computer Vision. 2022.

[14] Sun, Fang, et al. "Deepfake Detection Method Based on Cross-Domain Fusion." Security and Communication Networks 2021.1 (2021): 2482942.

[15] Song, Luchuan, et al. "Adaptive face forgery detection in cross domain." European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022.

[16] Amin, Muhammad Ahmad, et al. "Deepfake detection based on cross-domain local characteristic analysis with multi-domain transformer." Alexandria Engineering Journal 91 (2024): 592-609.