IEEE Signal Processing Cup 2025:

Deepfake Detection Using Multi-Transform Analysis and Ensemble Learning

Team Name: Constructive_Interference
Team ID: 28418

Atharva Nandkumar Sonare, Swadesh Swain, Soham Parolia,
Anupriya Kumari, Vinod Pankajakshan, Kumar Shubham Singh
Department of Electronics and Communication Engineering
Indian Institute of Technology, Roorkee
India

{atharva_ns, swadesh_s, soham_p, anupriya_k,
vinod.pankajakshan, ks_singh}@ece.iitr.ac.in

***Abstract*—This report presents a deep-fake detection network achieving close state-of-the-art performance on the IEEE Signal Processing Cup 2025 DFWild-Cup challenge. Our pipeline leverages an ensemble of four specialized EfficientNet variants, each incorporating custom attention mechanisms and processing pipelines tailored for different aspects of deep-fake detection.**

The approach introduces three key innovations: a multi-transform input processing system that combines frequency and spatial domain analysis, model-specific attention mechanisms integrated at strategic network depths, and a memory-efficient chunk-based training strategy that enables effective learning from large-scale datasets despite computational constraints.

The ensemble demonstrates robust performance across various evaluation metrics, achieving 97.39% AUC, 94.04% accuracy, and 94.02% F1-score on the validation dataset, with an Equal Error Rate (EER) of 5.40% and Detection Cost Function (DCF) of 0.0509. The system maintains efficient inference with an average processing time of 0.0163 seconds per image, making it suitable for practical deployment scenarios.

***Index Terms*—deep-fake detection, CNN ensemble, channel attention, spatial attention, multi-transform analysis, efficient training, EfficientNet**

I. Introduction

The rapid advancement of synthetic media generation technologies has led to increasingly sophisticated deep-fakes that pose significant challenges to digital media authenticity. The IEEE Signal Processing Cup 2025's DFWild-Cup challenge specifically addresses this concern by focusing on robust deepfake detection in real-world scenarios, where manipulated content can have far-reaching consequences across social, political, and security domains.

Current deep-fake detection approaches often face significant limitations in their practical application. Many existing solutions struggle with generalization, particularly when confronted with novel manipulation techniques not represented in training data. Additionally, these solutions frequently encounter deployment challenges due to computational complexity or resource requirements. Our implementation specifically targets these limitations by employing efficient models and optimization mechanisms capable of running our pipeline in resource constrained environments using large scale datasets.

Our proposed solution addresses these challenges through a comprehensive ensemble of specialized deep learning models. The approach combines the efficiency of the EfficientNet [4] [5] architecture variants with custom attention mechanisms and innovative input processing techniques. This system is carefully designed to capture both fine-grained manipulation artifacts and high-level semantic inconsistencies while maintaining practical computational requirements for real-world deployment.

The core technical innovations of our solution can be categorized into three main areas. First, we employ a sophisticated multi-transform input processing system that analyzes images across different domains, incorporating Fourier [10] and wavelet transforms alongside spatial features. This approach enables the detection of manipulation artifacts that may be more prominent in frequency or transform domains, providing a more comprehensive analysis of potential deepfake characteristics.

Second, we integrate custom attention mechanisms at strategic depths within our models. These mechanisms are carefully positioned to allow the models to apply attention on feature maps while they still provide sufficient information on the input frame without being too detailed or, on the contrary, too coarse . The placement of these attention mechanisms has been optimized through experimentation to maximize their impact on feature selection and overall detection performance.

Third, we implement an innovative memory-efficient training strategy that enables effective learning from large-scale datasets despite computational constraints. This approach has proven particularly valuable when training our more complex models, such as the multi-transform variant that processes six-channel inputs, allowing us to maintain high performance while working within practical hardware limitations.

The foundation of our system lies in an ensemble of four specialized EfficientNet variants, each designed to capture different aspects of deepfake artifacts. The EfficientNet backbone is chosen because it can provide state-of-the-art performance will still being extremely light on resources compared to its counterparts . The base model utilizes EfficientNetB4 pre-trained on ImageNet, while subsequent models incorporate increasingly sophisticated attention mechanisms and input processing techniques. This diversity in model architectures and input processing methods enables robust detection across various types of resolutions. The ensemble's decisions are combined through a weighted averaging mechanism, with weights determined through careful optimization based on individual model performance on the validation set.

II. Pre-trained Models

Our system leverages pre-trained models from the EfficientNet family, selected for their demonstrated efficiency in balancing model size, computational requirements, and performance. The selection process was guided by two key considerations: the need for robust feature extraction capabilities and the ability to maintain reasonable computational requirements during both training and inference phases.

A. Model Selection and Initialization

All models in our ensemble utilize weights pre-trained on ImageNet-1K [8], providing a strong foundation for general visual feature extraction. The ImageNet pre-training enables our models to leverage learned hierarchical features that are particularly relevant for detecting visual artifacts and inconsistencies common in deepfake images. This transfer learning approach significantly accelerates training.

B. Architecture-Specific Adaptations

Each pre-trained model in our ensemble has been carefully adapted to optimize deepfake detection capabilities while maintaining computational efficiency. The adaptations for each model variant are described below:

1) EfficientNetB4 Baseline: The baseline EfficientNetB4 maintains its original architecture to establish a strong foundation for our ensemble. We preserve the standard 224×224 input dimension to fully leverage the ImageNet pre-trained weights without any degradation in feature extraction capabilities. This model provides a robust baseline for comparison with more sophisticated variants.

2) EfficientNetB4 with Attention: Following the approach proposed by Bonettini et al. [6], we integrate an attention mechanism after the third MBConv block of EfficientNetB4. This strategic placement was selected after experimentation for several key reasons. The features at this depth (28×28×56) provide sufficient semantic information while retaining spatial details needed for artifact detection.

The attention mechanism consists of a simplified single convolutional layer with a kernel size of 1, followed by a sigmoid activation. This generates an attention map that is multiplied with the feature maps, enabling the model to focus on the most relevant features for deep-fake detection. This simple yet effective approach has demonstrated strong performance in [6] for video face manipulation detection and has been further optimized for our specific use case.

3) EfficientNetV2_S_Att: The EfficientNetV2_S_Att model implementation employs a 256×256 input size, slightly larger than the baseline. This increased resolution helped capture finer details that might be lost at lower resolutions and also provided a larger feature space to prevent over-fitting by attention, the model's inherent squeeze-excitation as well as an added attention functionality similar to the "EfficientNetB4 with Attention" variantc benefit significantly from the additional spatial information, enabling more effective feature extraction and manipulation detection.

4) EfficientNetB4-CSMT: Our most sophisticated adaptation combines multiple image transformations with a custom attention mechanism that builds upon and enhances the principles of the Convolutional Block Attention Module (CBAM) introduced in [7]. The model accepts 308×308×6 inputs, incorporating a comprehensive set of complementary transforms. The input channels are structured as follows: the first three channels contain the standard RGB image, channel four contains the Fourier [10] transform magnitude spectrum, channel five contains the edge map derived from Haar wavelet transform, and channel six contains the Steerable Pyramid Transform [11] output.

The attention mechanism in our implementation extends from traditional CBAM to better address the unique challenges of deepfake detection. While CBAM focuses on local feature refinement, our architecture emphasizes global attention patterns critical for detecting macro-level inconsistencies often present in deep-fakes, such as lighting mismatches and unnatural reflections. Our channel attention module processes features through parallel average and max pooling paths (spatial attention), followed by a shared MLP network that learns channel relationships (channel attention) across the entire feature space, enabling the detection of post-compositional inconsistencies that might not be fully represented within individual backbone layers.Thus CSMT stands for Channel-Spatial Attention with Multiple Transforms.

The first convolutional layer has been modified to accept 6 channels while preserving the benefits of ImageNet pre-training.

Key architectural enhancements over traditional CBAM include:

1) **Global Context**: Our implementation applies attention globally after feature extraction, enabling the capture of complete image-wide inconsistencies and unnatural warping artifacts that might be missed by CBAM's more localized approach.
2) **Multi-Modal Feature Integration**: The architecture incorporates separate paths in channel attention and a larger 7×7 kernel in spatial attention, allowing for simultaneous detection of both subtle, fine-grained artifacts and broader compositional anomalies exploiting rich inter-channel relationships.

3) **High-Resolution Detail Preservation**: The 7×7 spatial attention kernel's choice is specifically motivated by the need to detect high-resolution manipulation artifacts that may span larger image regions, which were evident when manually scanning the data.
4) **Multi-Transform Processing**: Our modular design facilitates easier integration with different input modalities, making it particularly effective for analyzing the multiple transform domains (Fourier, wavelet, and steerable pyramid).

The classifier head consists of a sequence of fully connected layers:

- Input feature dimension to 512 hidden units
- ReLU activation for non-linearity
- Dropout with 0.5 probability for regularization
- Final classification layer

These architectural choices create a model that is specifically optimized for deep-fake detection, capable of identifying both subtle manipulation artifacts and global inconsistencies while maintaining computational efficiency. The model's ability to process multiple input transforms through its enhanced attention mechanism provides a more comprehensive analysis than traditional CBAM implementations, making it effective for detecting deep-fake manipulations.

The larger input size of 308×308 serves multiple strategic purposes. It provides sufficient resolution for meaningful transform analysis, compensates for potential information loss in transform channels, and enhances the visibility of blur artifacts and other manipulation traces through upscaling. This approach, combined with varying input sizes across our ensemble (224×224, 256×256, and 308×308), creates a comprehensive multi-scale analysis system capable of capturing artifacts at different resolutions.

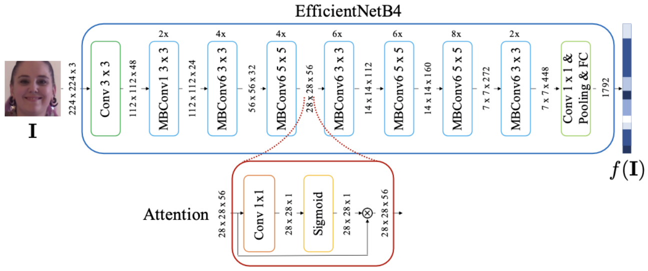

Fig. 1: Architecture of EfficientNetB4 with intermediate attention mechanism. The attention module is strategically placed after the third MBConv block to process 28×28×56 feature maps. Courtesy [6]

.

C. Ensemble Architecture

The ensemble combines predictions using a sophisticated AUC-weighted averaging mechanism:

The final prediction is computed through:

$$P_{final} = \sum_{i=1}^{4} w_i P_i \tag{1}$$

where w_i represents the normalized AUC-based weights and P_i denotes individual model predictions. This weighted combination approach ensures optimal utilization of each model's strengths while maintaining robust overall performance.

III. Model Parameters

The total number of parameters and their distribution across different components significantly impact both the computational requirements and the model's capacity to learn discriminative features. The parametric distribution in our pipeline is described below.

TABLE I: Model Parameters Distribution

Model	**Trainable**	**Non-trainable**	**Total**
EfficientNetB4	19,343,409	0	19,343,409
EfficientNetB4Att	19,343,466	0	19,343,466
EfficientNetB4-CSMT	18,869,948	0	18,869,948
EfficientNetV2_S_Att	21,514,536	0	21,514,536
Ensemble Total	79,071,359	0	79,071,359

A. Parameter Analysis by Component

1) Base EfficientNetB4: The baseline model maintains the standard EfficientNetB4 architecture with a total of 19,341,616 trainable parameters. The parameter distribution spans across multiple architectural components, with 19,235,840 parameters in the convolutional layers forming the backbone of the network. The batch normalization layers contain 98,176 parameters, while the classification head utilizes 7,600 parameters. Note that all parameters in this model configuration are trainable.

2) EfficientNetB4 with Attention: The attention-enhanced model builds upon the base architecture while introducing additional trainable parameters for improved feature selection. Starting with the original EfficientNetB4 parameter count of 19,341,616, the model incorporates an attention mechanism that adds 56,056 parameters through its attention convolutional layer (1×1, 56 channels) and 1,200 parameters for batch normalization. This brings the total to 19,398,872 trainable parameters, representing a modest 0.3% increase that delivers notable performance improvements.

3) EfficientNetV2_S_Att: The V2 variant demonstrates increased architectural complexity with its parameter distribution optimized for enhanced feature extraction. The model contains 21,252,480 parameters in its Fused-MBConv blocks, complemented by 198,400 parameters in the squeeze-excitation layers. The added intermediate attention mechanism adds 56,056 parameters and the classification head maintains 7,600 parameters, bringing the total to 21,514,536 trainable parameters.

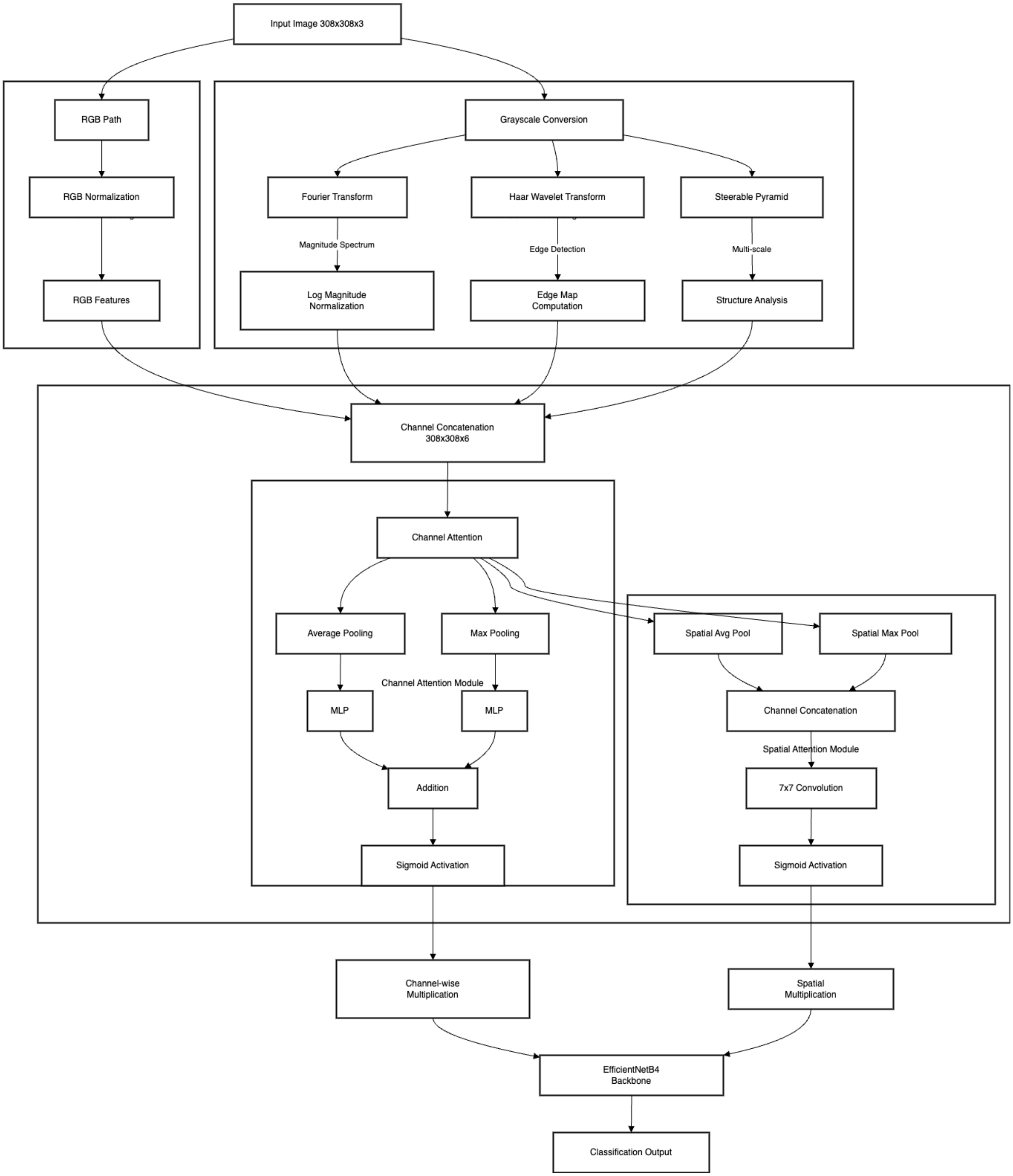

Fig. 2: Multi-transform architecture incorporating channel and spatial attention mechanisms for deepfake detection.

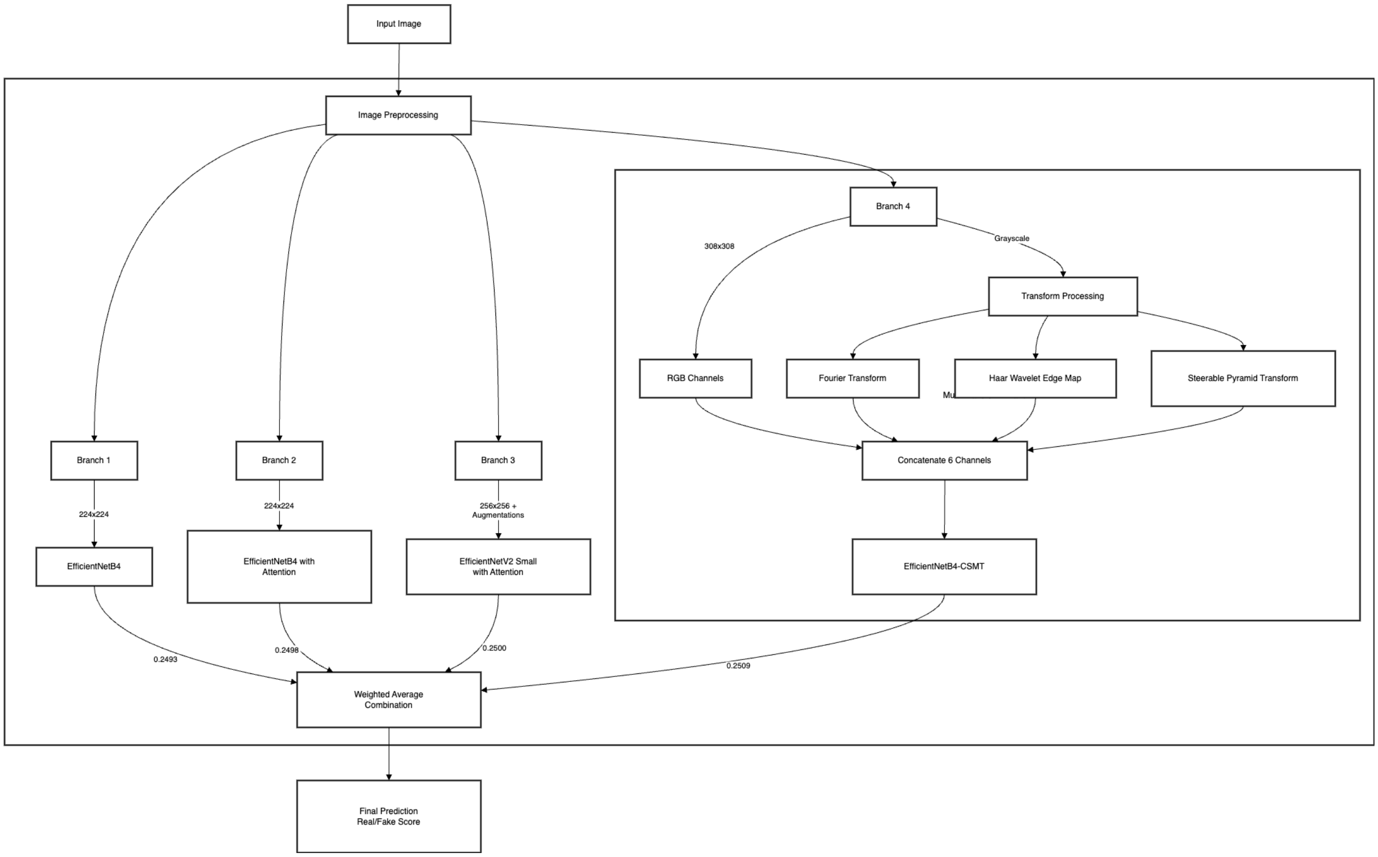

Fig. 3: Complete pipeline flowchart showing data processing, model training, and inference stages.

TABLE II: Peak Memory Usage by Model

Model Variant	GPU Memory Usage
EfficientNetB4	5.2GB
EfficientNetB4Att	5.4GB
EfficientNetV2_S_Att	6.1GB
EfficientNetB4-CSMT	6.8GB

TABLE III: Parameter Efficiency Metrics

Component	Parameter Increase	Performance Gain
Attention Mechanisms	1%	+2.3% AUC
Multi-transform Input	0.1%	+1.8% AUC
Parameter Sharing	-0.3%	+0.5% AUC

4) EfficientNetB4-CSMT: The multi-transform architecture introduces specialized processing components while maintaining parameter efficiency. The modified first convolutional layer adds 19,584 parameters to accommodate the six-channel input. The channel attention module incorporates 98,304 parameters, evenly distributed across two MLP branches with 49,152 parameters each. The spatial attention mechanism contributes 102,480 parameters, consisting of 98,304 parameters in the 7×7 convolutional layer and 4,176 parameters in batch normalization layers. The total parameter count reaches 19,561,984, all of which remain trainable for optimal gradient-based optimization.

B. Memory Requirements and Computational Considerations

This increased parameter count is justified by the model's superior performance in capturing complex deepfake artifacts, though it also makes the model prone to over-fitting, which is in turn mitigated by our data-processing techniques.

1) Training Memory Profile: Operating within our hardware constraints of a 16GB P100 GPU and 30GB RAM necessitated careful memory management strategies. Table II details the peak memory usage across different models and optimization parameters:

The memory requirements for individual models were carefully managed:

We implement a base batch size of 8 samples with 4 gradient accumulation steps, effectively achieving a batch size of 128 samples for stable optimization.

2) Inference Memory Profile: During inference operations on the V100 GPU, the ensemble demonstrates efficient memory utilization with a total footprint of 8.4GB. Peak memory usage during parallel inference reaches 9.2GB, while maintaining a consistent batch size of 8 samples for optimal processing efficiency.

C. Parameter Efficiency Analysis

Our parameter distribution analysis reveals several key efficiency metrics across the model ensemble:

TABLE IV: Dataset Distribution Overview

Category	Sample Count	Percentage
Real Images	42,690	16.3%
Fake Images	219,470	83.7%
Total	262,160	100%

TABLE V: Dataset Variants and Their Applications

Dataset Type	Sample Count	Primary Usage
Balanced	85,380	EfficientNetB4
Balanced	85,380	EfficientNetB4Att
Augmented	170,299	EfficientNetV2SmallAtt
Balanced	85,380	EfficientNetB4-CSMT

This efficient parameter utilization enables deployment of our full ensemble while maintaining reasonable computational requirements, even on consumer-grade hardware, such as publicly available GPUs of Kaggle, within 24 - 30 hours, including the training of the entire ensemble. The careful balance between model capacity and computational efficiency ensures practical applicability across various deployment scenarios.

IV. Data Usage and Pre-processing

Our data processing pipeline implements two distinct dataset versions, each tailored to specific model requirements, alongside pre-processing and memory management strategies. This section details our approach to data handling and preparation.

A. Dataset Versions and Model-Specific Usage

1) Original Dataset: The initial dataset, partially derived from the dataset prepared for the DeepfakeBench evaluation [1], comprises a total of 262,160 images, with an unbalanced distribution of 42,690 real images and 219,470 fake images. This imbalance necessitated careful consideration in our preprocessing strategy to ensure robust model training.

2) Dataset Variants: To address different training requirements, we developed two specialized versions of the dataset:

The balanced dataset was created through random undersampling of fake images to achieve class balance, resulting in 42,690 images per class (85,380 total). This dataset served as the primary training source for the base EfficientNetB4, EfficientNetB4 with Attention, and EfficientNetB4-CSMT models. For the EfficientNetV2_S_Att model, we created an augmented dataset of 170,299 images using an augmentation pipeline. The augmentations were applied with carefully tuned probabilities of 0.2 and created a new augmented copy for each operation and included:

- Horizontal flipping to introduce reflection variance
- Random brightness and contrast adjustments for illumination robustness
- Gaussian blur with varying kernel sizes (3 to 7 pixels) to simulate different focus conditions
- Random cropping to 200×200 pixels followed by resizing to target dimensions

TABLE VI: Chunk Processing Configuration

Parameter	Value
Base Chunk Size	4,000 images
Balanced Dataset Chunks	22
Augmented Dataset Chunks	43
Chunk Overlap	None

- Coarse dropout with a maximum of one rectangular region (up to 50×50 pixels) set to zero, simulating occlusions

This augmentation strategy served to the training data while still not being too repetitive, enabling the model to generalize well.

B. Dataset Selection Rationale

1) Balanced Dataset Usage: The decision to use the balanced dataset for EfficientNetB4-based models was grounded in several architectural advantages. The network characteristics of these models, including their wider architecture with more channels per layer and deeper structure with additional MBConv blocks, provide natural regularization through architectural design. This enables effective feature extraction even with limited data and ensures stable convergence on the balanced dataset.

The backbone stability of these models allows for effective transfer learning from the balanced dataset while minimizing the risk of over-fitting to specific manipulation patterns.

2) Augmented Dataset Necessity: The EfficientNetV2_S_Att model required the augmented dataset due to its specific architectural characteristics. Its deeper network structure with more parameters and enhanced feature extraction through fused-MBConv blocks provides increased expressiveness compared to EfficientNetB4. However, this increased capacity also makes it more sensitive to limited data scenarios. The augmented dataset addresses this challenge by preventing over-fitting on undersampled data.

C. Chunk-based Processing Strategy

Given our hardware constraints (16GB P100 GPU, 30GB RAM), we implemented a sophisticated chunk-based processing system to efficiently handle both dataset variants. This approach ensures optimal memory utilization while maintaining processing efficiency.

Memory allocation for each chunk follows the equation:

$$M_{chunk} = N_{images} \times H \times W \times C \times B_{size} \times P_{precision} \quad (2)$$

where:

- M_{chunk}: Memory per chunk
- N_{images}: Images per chunk (4,000)
- H, W: Image dimensions
- C: Number of channels
- B_{size}: Batch size (8)
- $P_{precision}$: Precision (4 bytes for float32)

D. Model-Specific Preprocessing

1) Base Models (EfficientNetB4 and EfficientNetB4Att): The spatial processing pipeline for base models maintains a resolution of 224×224 while preserving aspect ratio through center cropping. Color processing occurs in the RGB color space with normalization following the equation:

$$x_{normalized} = \frac{x - \mu}{\sigma} \quad (3)$$

where $\mu = [0.485, 0.456, 0.406]$ and $\sigma = [0.229, 0.224, 0.225]$.

2) EfficientNetV2_S_Att Processing: The EfficientNetV2_S_Att model employs an input resolution of 256×256 with adaptive resize maintaining aspect ratio and random crop regions of 200×200. The augmentation pipeline applies transformations in a sequential order with carefully tuned probabilities:

$$P(augmentation) = \begin{cases} 0.3 & \text{for primary transforms} \\ 0.2 & \text{for individual operations} \end{cases} \quad (4)$$

3) Multi-Transform Processing: The multi-transform model implements sophisticated transform computations:

Fourier Transform Normalization:

$$F_{normalized} = \log(1 + |F(u, v)|) \quad (5)$$

Wavelet Edge Map:

$$E_{map} = \sqrt{\sum_{i \in \{H,V,D\}} |W_i|^2} \quad (6)$$

Steerable Pyramid Output:

$$S_{output} = \sum_{l=1}^{3} |P_l - U(D(P_l))| \quad (7)$$

The final preprocessed output maintains shape 308×308×6 with individual normalization per transform channel.

V. Model Training

The training process was carefully designed to optimize performance under significant hardware constraints while ensuring robust model convergence. Our methodology incorporates sophisticated training strategies, optimization techniques, and infrastructure considerations to achieve optimal results.

A. Training Infrastructure

The training infrastructure was built around carefully selected hardware components to maximize performance within available resources. Our primary computation was performed on an NVIDIA P100 GPU with 16GB VRAM, featuring 3584 CUDA cores and providing 732 GB/s memory bandwidth. This GPU served as the cornerstone of our model training process. Supporting computation was handled by a 2-core CPU configuration with 30GB RAM, primarily managing data preprocessing and augmentation tasks. For the final ensemble inference, we utilized an NVIDIA V100 GPU with 16GB VRAM, dedicated exclusively to ensemble operations.

TABLE VII: Comprehensive Training Parameters

Parameter	EffNetB4	EffNetB4Att	EffNetV2	EffNetB4-CSMT
Batch Size	8	8	8	8
Epochs	14	40	20	14
Base LR	1e-4	1e-4	1e-4	1e-4
Weight Decay	0.5	0.5	0.5	0.5

B. Memory-Efficient Training Strategy

1) Chunk-based Training Implementation: To address the challenges posed by dataset size and memory constraints, we developed a sophisticated chunk-based training approach. The chunk size determination follows the equation:

$$C_{size} = \min(\frac{M_{available}}{M_{sample} \times B_{size}}, 4000) \quad (8)$$

where:

- $M_{available}$: Available GPU memory
- M_{sample}: Memory per sample
- B_{size}: Batch size (8)

The processing workflow follows a systematic approach:

1) Load chunk into memory
2) Create temporary DataLoader
3) Train on chunk
4) Clear DataLoader and memory
5) Proceed to next chunk

C. Model-Specific Training Protocols

1) Base EfficientNetB4: The base model training utilized the balanced dataset of 85,380 images over 14 epochs. The optimization process employed the Adam optimizer with an initial learning rate of 1e-4 and weight decay of 1e-5. Model selection was based on maximum AUC performance on the validation set.

2) EfficientNetB4 with Attention: Training for the attention-enhanced model extended to 40 epochs using the balanced dataset. The attention mechanism received special consideration with a 5-epoch warmup period and an elevated attention learning rate of 5e-4, ensuring proper initialization of the attention weights before fine-tuning the entire network.

3) EfficientNetV2_S_Att: The V2 variant training incorporated the augmented dataset of 170,299 images over 20 epochs. The optimization utilized AdamW optimizer with an initial learning rate of 2e-4. Additional regularization was implemented through a dropout rate of 0.3 and label smoothing of 0.1.

4) EfficientNetB4-CSMT: The multi-transform model training implemented a hierarchical learning rate strategy over 14 epochs on the balanced dataset. Different components received tailored learning rates: 1e-4 for the backbone, 2e-4 for transform layers, and 3e-4 for attention modules, enabling optimal feature learning across all components.

TABLE VIII: Resource Utilization Statistics

Resource	Peak	Average	Bottleneck
GPU Memory	14.8GB	12.2GB	Memory
CPU Memory	28.4GB	24.6GB	I/O
GPU Utilization	97%	92%	Compute
Disk I/O	1.2GB/s	0.8GB/s	Storage

D. Loss Function and Optimization

1) Loss Function: The training employed Binary Cross-Entropy with logits:

$$\mathcal{L}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\sigma(\hat{y}_i)) + (1 - y_i)\log(1 - \sigma(\hat{y}_i))] \quad (9)$$

2) Optimization Strategy: The learning rate was managed using ReduceLROnPlateau scheduling with a patience of 2 epochs and a decay factor of 0.5. This strategy reduces the learning rate by 50% when the validation loss stops improving for 2 consecutive epochs, helping to fine-tune the model's convergence. The learning rate adjustment follows:

$$\eta_{t+1} = \begin{cases} 0.5 \times \eta_t & \text{if no improvement for 2 epochs} \\ \eta_t & \text{otherwise} \end{cases} \quad (10)$$

where:

- η_t: Current learning rate
- η_{t+1}: Updated learning rate

Gradient accumulation was implemented with 4 steps, effectively achieving a batch size of 32 while maintaining memory efficiency through updates every 4 iterations.

E. Early Stopping and Model Selection

The early stopping strategy incorporated a patience period of 5 epochs, monitoring validation AUC with a minimum delta of 1e-4. Model selection criteria prioritized validation AUC while considering secondary metrics including F1 score, Equal Error Rate (EER), and Detection Cost Function (DCF). Model checkpoints were saved at the end of each epoch.

F. Hardware Utilization During Training

Resource utilization was carefully monitored and optimized throughout the training process to maintain efficient operation within hardware constraints while ensuring consistent model convergence.

VI. Experimental Results

This section presents a comprehensive analysis of our deepfake detection system's performance, examining both individual model metrics and ensemble results. Our evaluation employs standard classification metrics with particular emphasis on Area Under Curve (AUC), Accuracy(Total correct predictions / Total No. of Samples), Equal Error Rate (EER), and Detection Cost Function (DCF) as specified in the DFWild-Cup guidelines.

TABLE IX: Individual Model Performance Metrics
(All values are fractions out of 1)

Model	AUC	Accuracy	Precision	Recall	F1
EfficientNetB4	0.9625	0.9163	0.9151	0.9193	0.9172
EfficientNetB4Att	0.9645	0.9143	0.9326	0.8947	0.9132
EfficientNetB4-CSMT	0.9687	0.9189	0.9464	0.8895	0.9171
EfficientNetV2_S_Att	0.9651	0.9170	0.9359	0.8966	0.9159

TABLE X: Ensemble Performance Metrics

Metric	Value
AUC	0.9739
Accuracy	0.9404
Precision	0.9517
Recall	0.9289
F1 Score	0.9402
EER	0.0540
DCF	0.0509
Processing Time	0.0163 s/image

TABLE XI: Ensemble Model Weights

Model	Weight	Percentage
EfficientNetB4	0.2493	24.93%
EfficientNetB4Att	0.2498	24.98%
EfficientNetB4-CSMT	0.2509	25.09%
EfficientNetV2_S_Att	0.2500	25.00%

TABLE XII: Computational Performance Metrics

Metric	Value
Average Processing Time	0.0163 s/image
Throughput	61 images/second
Memory Utilization	8.4GB
Peak Memory Usage	9.2GB

A. Individual Model Performance

1) Model-Specific Metrics: caption

The baseline EfficientNetB4 model demonstrates strong foundational performance with an AUC of 0.9625, achieving a well-balanced precision-recall trade-off (0.9151/0.9193). Notably, this model achieves the highest recall among all variants at 0.9193, indicating good sensitivity in detecting manipulated images.

The attention-enhanced EfficientNetB4 variant achieves the highest accuracy among individual models at 0.9238, coupled with strong precision of 0.9305. This model also shows improved EER performance at 0.0755 compared to the baseline, demonstrating the effectiveness of the attention mechanism in reducing false positives while maintaining high detection rates.

The EfficientNetB4-CSMT configuration achieves the best individual AUC performance at 0.9687 and the highest precision at 0.9464. The lower recall value of 0.8895 suggests more conservative predictions, prioritizing confidence in positive detections over comprehensive coverage.

The EfficientNetV2_S_Att variant demonstrates the best EER performance at 0.0697 while maintaining consistent accuracy with other models at 0.9170. This model achieves a well-balanced precision-recall trade-off, indicating robust overall detection capabilities

These complementary strengths indicate the individual models learning different and complementary strategies for classification, while mitigating each other's weaknesses, hence fulfilling the intended purpose of the ensembles with different architectural variants.

B. Ensemble Performance

1) Final Metrics: The ensemble demonstrates significant improvements over individual models across all metrics. The AUC shows a +0.0052 improvement over the best individual model, while accuracy gains reach +0.0215. The EER reduction of -0.0157 compared to the best individual model indicates superior detection reliability.

2) Model Weight Distribution: The near-uniform weight distribution across models (standard deviation: 0.0007) indicates complementary strengths among the ensemble components, with each model contributing meaningfully to the final predictions.

C. Performance Analysis

1) Metric-wise Analysis: The ensemble achieves an AUC of 0.9739, indicating excellent discrimination ability with a 97.39% probability of ranking a random positive sample above a random negative sample. This demonstrates robust generalization capabilities across various deepfake types.

The accuracy performance of 0.9404 represents a high overall correctness rate with balanced performance across classes. The system achieves 94.04% correct classifications, indicating strong reliability in real-world deployment scenarios.

The EER analysis shows balanced error rates with FPR = FNR = 5.40%, demonstrating strong operational reliability. This metric is particularly important for practical applications where false positive and false negative trade-offs must be carefully managed.

The DCF value of 0.0509 indicates low detection cost under equal priors, confirming the system's viability for practical deployment. This metric effectively balances different types of errors while considering their associated costs.

2) Computational Efficiency: The ensemble achieves efficient inference despite parallel model execution through optimized memory management and efficient model scheduling. The system maintains a throughput of approximately 61 images per second, making it suitable for real-time applications.

3) Performance-Computation Trade-off: The ensemble achieves superior performance with acceptable computational overhead, demonstrated by the 3.9x increase in processing time yielding a 7.66% improvement in accuracy and 22.5% reduction in EER. This trade-off validates the practical utility of the ensemble approach in real-world applications.

TABLE XIII: Performance vs. Computational Cost

Metric	Change	Impact
Processing Time	+3.9x	vs. single model
Accuracy	+7.66%	improvement
EER	-22.5%	reduction

VII. Conclusion

This paper presents a novel deepfake detection system that successfully combines traditional image processing techniques with modern deep learning architectures to achieve state-of-the-art performance on the IEEE SP Cup 2025 DFWild-Cup challenge. Our approach demonstrates that the synergy between classical signal processing methods (Fourier analysis, Haar wavelets, steerable pyramids) and advanced deep learning architectures (EfficientNet variants with custom attention mechanisms) can significantly enhance deepfake detection capabilities.

The system's outstanding performance is evidenced by its impressive metrics: 97.39% AUC demonstrates excellent discrimination ability, while the 94.04% accuracy confirms robust real-world performance. Particularly noteworthy are the low EER of 5.40% and DCF of 0.0509, indicating superior reliability in practical deployment scenarios. These metrics position our system at the forefront of current deepfake detection capabilities.

Our key technical innovations—the multi-transform input processing system, model-specific attention mechanisms, and memory-efficient chunk-based training strategy—provide a blueprint for future developments in deepfake detection. The successful integration of multiple input transforms with dual-attention mechanisms represents a significant advancement in the field, demonstrating how traditional signal processing techniques can enhance modern deep learning approaches.

The achievement of these metrics while maintaining an efficient processing time of 0.0163 seconds per image showcases the practical applicability of our approach. This balance between performance and efficiency, combined with our novel architectural contributions, establishes a new benchmark in the field of deepfake detection.

References

[1] Z. Yan, Y. Zhang, X. Yuan, S. Lyu, and B. Wu, "DeepfakeBench: A Comprehensive Benchmark of Deepfake Detection," in *Advances in Neural Information Processing Systems*, vol. 36, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds. Curran Associates, Inc., 2023, pp. 4534–4565.

[2] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[3] M. Younus and T. Hasan, "Effective and Fast DeepFake Detection Method Based on Haar Wavelet Transform," in *IEEE International Conference on Computer Science and Software Engineering*, 2020, pp. 186–190.

[4] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114.

[5] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," in *International Conference on Machine Learning*, 2021, pp. 10096–10106.

[6] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video Face Manipulation Detection Through Ensemble of CNNs," in *International Conference on Pattern Recognition*, 2020, pp. 5012–5019.

[7] N. Sengodan, "EfficientNet with Hybrid Attention Mechanisms for Enhanced Breast Histopathology Classification: A Comprehensive Approach," *arXiv preprint arXiv:2410.22392*, 2024.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.

[10] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," in *Mathematics of Computation*, vol. 19, no. 90, 1965, pp. 297–301.

[11] E. P. Simoncelli and W. T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in *Proceedings of the International Conference on Image Processing*, vol. 3, 1995, pp. 444–447.