2025 IEEE Signal Processing Cup Report

[*]**Team UCSD FREQZ**: *University of California, San Diego*

Philip Pincencia
ECE Department
University of California, San Diego
San Diego, USA
ppincencia@ucsd.edu

Kendrick Nguyen
ECE Department
University of California, San Diego
San Diego, USA
kan006@ucsd.edu

Genaro Salazar
ECE Department
University of California, San Diego
San Diego, USA
gsalazarruiz@ucsd.edu

Benjia Zhang
ECE Department
University of California, San Diego
San Diego, USA
b9zhang@ucsd.edu

Jason Yang
ECE Department
University of California, San Diego
San Diego, USA
jay049@ucsd.edu

Brent Brewster
ECE Department
University of California, San Diego
San Diego, USA
bbrewster@ucsd.edu

Girish Krishnan
ECE Department
University of California, San Diego
San Diego, USA
gikrishnan@ucsd.edu

Truong Nguyen
ECE Department
University of California, San Diego
San Diego, USA
tqn001@ucsd.edu

.

***Abstract*—This report contains all the details of the models, training and testing methods, and analysis we used for the IEEE 2025 Signal Processing Cup.**

***Index Terms*—Deepfake detection, hyperparameter, machine learning, neural network**

I. Introduction

Building a robust pipeline for machine learning and DSP applications is never without its challenges, and our team's journey was no exception. From navigating unbalanced datasets and complex extraction processes to designing memory-efficient training workflows, we encountered numerous obstacles. However, each challenge became an opportunity to innovate, whether it was compressing large data chunks, optimizing resource allocations for scalable GPU-based processing, or refining our models. Our project focused on the critical task of detecting Deepfakes in images using convolutional neural networks, an issue of growing significance as manipulated media becomes more prevalent.

What began as an exploration of pre-trained models evolved into an intensive process of hyperparameter tuning and model refinement, ultimately pushing our detection accuracy to nearly 96%. This achievement not only highlighted the adaptability of these models, but also underscored the value of a collaborative and iterative approach.

This work contributes to the real-world need for high-accuracy scalable detection systems that can combat misinformation and protect the integrity of digital media. Our pipeline, with its emphasis on scalability, accuracy, and efficient workflows, stands as a testament to what can be achieved when real-world challenges are tackled head-on. It offers meaningful insights into ongoing research in Deepfake detection and the broader field of Digital Signal Processing.

II. Project Structure

To ensure efficiency and focus, the project was divided into two specialized teams: the **CNN Team** and the **Data Pipeline Team**. This division allowed each team to tackle key challenges in parallel while minimizing dependencies.

- **CNN Team**: Focused on selecting, optimizing, and implementing deep learning models for Deepfake detection. They worked with pre-trained models like XCeption, Inception-v3, and EfficientNet-B4, fine-tuning them for high accuracy. The main contributions included leveraging tools like Optuna for hyperparameter tuning and exploring ensemble learning techniques. Their work allowed for robust model training and high-performance detection.
- **Data Pipeline Team**: Managed the logistics of data set [1] preparation and integration, addressing issues such as data imbalance and large file handling. They optimized workflows for seamless data feeding into the training pipeline and configured the environment for GPU cluster training. Their efforts ensured smooth data handling, allowing the CNN Team to focus on model optimization.

This structure enabled focused contributions from each team while ensuring that the overall project advanced efficiently. The clear division of tasks allowed for parallel progress,

minimizing bottlenecks, and improving collaboration across the teams.

III. Methodology Overview

Wavelet-CLIP: Wavelet-CLIP [2] is the first model we experimented with. The algorithm is divided into two sections: encoding and classification. In the encoding step, we use the Vit-L/14 transformer to transform the image into its latent space, where its features can be used for analysis. The transformer is also pre-trained via CLIP fashion. Since Vit-L/14 was trained in a self-supervised way and not fine-tuned for a specific task like Deepfake detection, it is more general as it retains all general and robust features of the input images, which is useful for any task and purposes afterward. Furthermore, since it is not trained on specific data sets, the transformer generalizes better to unseen and diverse data sets (i.e. realistic Deepfakes).

Next, in the classification module, discrete wavelet transforms (DWTs) are used as the main sauce for classification. Compared to regular 2-D Fourier Transform, DWT can capture both frequency and location information about the image, as opposed to just frequency information in 2-D Fourier Transform. After the DWT is performed, an MLP layer is used. The first MLP layer in the methodology plays a critical role in refining the low-frequency features extracted through the DWT. Low-frequency features capture broad, nuanced patterns vital for identifying subtle inconsistencies in Deepfake images. However, these raw features may not be immediately suitable for classification. The MLP layer processes these low-frequency components to emphasize task-relevant patterns while suppressing noise or irrelevant details. This refinement step enhances the discriminative power of the features, enabling the model to focus on granular invariances critical for distinguishing between real and fake images. By transforming these features into a more expressive representation, the first MLP prepares them for recombination with the high-frequency features in the subsequent step. Then the image is reconstructed with the refined low frequency components using inverse discrete wavelet transform (IDWT). Then a second MLP layer is utilized to serve as the final classifier in the pipeline, tasked with making a binary decision: real or fake. Optimized for decision-making, this MLP translates the rich, multidimensional feature space into a classification result, such as a probability score or label.

Inceptionv3 + Xception: Our next model is a combination of the Inceptionv3 [3] and Xception [4] models. The InceptionV3 model is a deep learning approach that combines multiple ways of looking at images (like zooming in at different scales). It uses clever tricks to make computations faster, like breaking down large tasks (e.g., analyzing a 5x5 image patch) into smaller, easier tasks (two 3x3 patches). The model has several parts (Inception modules A to E), each designed to handle different kinds of patterns in images. It also includes helper layers (auxiliary classifiers) to make learning smoother and faster. InceptionV3 uses its modular structure to pull out patterns from images step by step. It reduces the image size as it goes deeper, focusing on the most important details. Helper layers also improve learning and act as backup systems. In the end, a fully connected layer makes the final decision about the image's category.

The Xception model is built on a simpler way of doing convolutions called depth wise separable convolutions. This breaks down the task into smaller, more efficient steps, focusing on spatial and channel-specific details separately. It's like an improved version of Inception modules. The model has 36 layers grouped into 14 sections, using shortcuts (residual connections) to keep things fast and accurate. Each layer is fine-tuned with techniques like normalization to make training stable. By focusing on separate details, the Xception model extracts better features for tasks like image recognition. These features are then passed through layers that decide if an image belongs to a certain category (real or fake, for example). It's designed to work well on new and large datasets without needing more memory or computation.

Combining these two models allows for better performance based on Inceptionv3's fast, modular computations and Xception's efficient convolutional layers.

Modified Inceptionv3: Our last model is a modified version InceptionV3 based on Sapphire0628 [5]'s existing comparisons between VGG16 [6], Resnet18 [7], and Inceptionv3 [3]. To optimize the performance of the models, we tuned the following hyperparameters: learning rate, regularization techniques, optimizer, gamma and step size. We chose the hyperparameters in Sapphire0628's final round of testing to train InceptionV3 on the given training and validation sets for the competition.

After splitting the data into training and validation sets, the transformer functions using the torchvision.transforms module. The training images had the following transformations: resize, center crop, random horizontal flip, random rotation, random grayscale, random Gaussian blur, conversion to PyTorch tensors, and pixel normalization. The validation set also underwent resizing and center cropping, as well as tensor conversion and normalization. These techniques help reduce noise and create a more robust dataset for learning.

IV. Challenges and Solutions

One of the first challenges we faced was managing large datasets. Because of the large volume of images in both the training and validation datasets, it is necessary to use our university's NRP Nautilus clusters to manage large datasets and GPU usage. Even so, to train three models with a large image dataset requires multiple nodes in parallel. Our team dedicated PCs to run these jobs in the GPU cluster throughout the competition.

Another challenge we faced was data imbalance. In the model training code, the imbalance in the dataset was addressed using **class weighting and a weighted random sampler**.

Class weights were computed based on the inverse frequency of each class. Let N be the total number of samples,

N_r the number of real images, and N_f the number of fake images.

$$W_r = 2N_r/N \tag{1}$$

$$W_f = 2N_f/N \tag{2}$$

These weights were used in the **CrossEntropyLoss** function to ensure that the loss contribution from each class was balanced during training.

Then, a **WeightedRandomSampler** was employed to sample training data such that the probability of selecting a sample from each class was proportional to its class weight. Specifically, the weight for each sample was assigned based on its label (w_r or w_f), ensuring balanced sampling during each training epoch.

V. Key Results

The evaluation of the Inception V3 model, identified as the best-performing model during training, demonstrates its strong ability to classify images as either real or fake. The model achieved an accuracy of 90.69%, indicating that it correctly identified the class of the images in the validation set in most cases.

The precision was measured at 85.99%, showing the model's ability to avoid false positives when predicting an image as real. The recall reached an impressive 97.05%, highlighting the model's effectiveness in capturing true positives—ensuring most real images were correctly classified. The balance between precision and recall is reflected in the F1 Score of 91.18%, which represents the harmonic mean of these two metrics, indicating an excellent overall classification performance.

Furthermore, the model achieved a ROC-AUC score of 88.68%, showing its capability to distinguish between real and fake images across various decision thresholds. The equal error rate (EER), a metric often used in classification problems to indicate the point where false positive and false negative rates are equal, was 8.66%, which is commendably low.

From a performance perspective, the model demonstrated efficient inference speed, with an average time of 0.0907 seconds per image, making it well-suited for real-time or near-real-time applications.

VI. Impact and Relevance

In this age of technology, misinformation becomes easier to disseminate as the accelerating development of AI creates more convincing and elusive Deepfakes. It is imperative to match or surpass this development with more robust methods of detection to be a step ahead of the curve. Digital signal processing concepts such as our experimentation with wavelets can be the competitive edge against adversarial Deepfakes. Multidisciplinary technologies like rPPG signal generation are being adapted to Deepfake detection algorithms with highly accurate results [8]. By finding more applications of these concepts, we can better understand not only the extent to which these concepts can be used but also make it harder for adversarial Deepfakes to evade detection.

Our work running multiple models rather than a single model in this competition proves advantageous outside this project. For Deepfakes in the wild, running images through parallel processing can safeguard against wrongly labeled images in one model if found in another model. With this, images in conflict between models can be put under better scrutiny. Biases in certain models can be eliminated with other models running in parallel as well. Running multiple models is the best way to find the highest accuracy model over several trials.

References

[1] Zhiyuan Yan, Yong Zhang, Xinhang Yuan, Siwei Lyu, and Baoyuan Wu. Deepfakebench: A comprehensive benchmark of deepfake detection. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 4534–4565. Curran Associates, Inc., 2023.

[2] Lalith Bharadwaj Baru, Rohit Boddeda, Shilhora Akshay Patel, and Sai Mohan Gajapaka. Wavelet-driven generalizable framework for deepfake face forgery detection, 2025.

[3] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015.

[4] François Chollet. Xception: Deep learning with depthwise separable convolutions, 2017.

[5] Sapphire0628. Deepfake detection. https://github.com/Sapphire0628/DeepFake-Detection, 2023.

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[8] Peng Xia Guang Stanley Yang Ruochen Tan Yin Ni, Wu Zeng. A deepfake detection algorithm based on fourier transform of biological signal. *Computers, Materials & Continua*, 79(3):5295–5312, 2024.