Development of Ensemble Strategies for Generalization in Deepfake Image Detection

by

Rohan M. Wagh

SB, Computer Science and Engineering and in Engineering as Recommended by the Department of Mechanical Engineering
MIT, 2025

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2025

Authored by: Rohan M. Wagh
Department of Electrical Engineering and Computer Science
August 15, 2025

Certified by: Amar Gupta
Research Scientist, Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Development of Ensemble Strategies for Generalization in Deepfake Image Detection

by

Rohan M. Wagh

Submitted to the Department of Electrical Engineering and Computer Science
on August 15, 2025 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

ABSTRACT

The growing accessibility of generative models has enabled the rapid proliferation of deepfake content, posing significant challenges in image-based biometric security and media authenticity. In this thesis, six diverse facial deepfake image datasets are assembled, and four modern detection models are evaluated in a cross-domain scenario. We observe that individual models fail to generalize to images generated by techniques outside the scope of their training data. This often hinders the applicability of a single model in real-world deepfake detection. This thesis proposes ensemble strategies as a means of addressing this lack of generalization. We find that the ensemble models outperform individual models in classifying deepfake images, particularly in terms of accuracy and recall. An exhaustive evaluation of combinations of models shows that ensembles of similar models provide limited benefit, whereas ensembles of complementary models lead to significant improvements in classification performance. Ensembling models based specifically on accuracy and recall metrics also produces models that lower the rate of more harmful false negative predictions. This work highlights the value of ensemble models in improving generalization across diverse image families and provides a framework for building robustness in real-world deepfake detection systems.

Thesis supervisor: Amar Gupta
Title: Research Scientist

Acknowledgments

I would like to begin by thanking MIT and the Computer Science and Artificial Intelligence Laboratory for providing an environment that has continually challenged and inspired me. The resources, people, and culture at this institution have played a defining role in shaping my academic and professional path.

I am deeply grateful to my thesis supervisor, Dr. Amar Gupta, for his guidance, encouragement, and trust throughout this project. I've learned a great deal about how to conduct independent research, think critically about machine learning systems, and communicate ideas effectively—skills that I will carry forward in my career.

Thank you to the members of my lab group and collaborators who contributed their time, feedback, and ideas along the way. I'd especially like to acknowledge my co-lead within the research group, Hilary Zen, and undergraduate research students, Megan Sun and Jeffrey Zhu, for discussions that pushed my thinking and experiments that shaped this work. I would also like to thank the members of Itau that supported our research, Miguel Wanderley, Gustavo Bicalho, Lucas Carvalho, and Guilherme Rinaldo; as well as Dr. Rafael Palacios for assistance in my research and feedback throughout the process. Their insights have helped me refine my thinking and approach research with clarity and focus.

Finally, I'm thankful to my family and friends for support and presence over these past years. From late-night problem sets to celebratory milestones, the memories we've shared during my time at MIT are ones I'll always hold close.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the past five years, the availability of AI tools has proliferated dramatically. While the act of making AI technology readily available online has delivered many advances in industrial, creative, and personal settings, it has also led to a rise in malicious and convincing synthetic media. One of the most common examples of such media comes from deepfakes, which are images or videos generated by AI to imitate real people, often without their knowledge or consent.

Advances in deep learning and computer vision have enabled the easy production of faces that are visually indistinguishable from those of real people. Just a few years ago, generating such content required custom model development and technical expertise, which served as a necessary barrier of entry. However, as the technology has rapidly improved, similarly high-quality forgeries are now accessible through easy-to-use web portals and open-source libraries [1]. As a result, deepfake content is increasingly present across entertainment and social media platforms, often maliciously used to promote political misinformation, identity theft, and targeted harassment.

As generative models become more prevalent and realistic, there is a growing need to develop more robust and reliable detection tools. Most deepfake content is difficult to detect visually, as artifacts are often subtle and manual review does not scale to the level of internet

traffic in a larger ecosystem [2]. This has highlighted the need to enhance the robustness of deepfake detection systems that can operate in environments with limited computing resources.

1.1 Motivations

Before discussing the specific motivations for this thesis, it is important to establish the context in which this work was developed. This project was conducted in collaboration with Banco Itaú, with the objective of combating the use of deepfake images in identity verification and account creation. This work focused on evaluating individual low-resolution RGB images, as end users may not have access to newer biometric systems on devices such as Apple's Face-ID and other infrared-mapped facial recognition systems. The goal of the collaborative was to improve real-time, compute-efficient deepfake detection pipelines capable of quickly verifying images passing into an identity verification system.

1.1.1 Special Concerns in Banking

The use of AI for malicious activity in a financial context is concerning, as unauthorized access to users' personal data can have severe consequences. Customers affected by identity theft often face long-term credit damage, legal disputes over inauthentic activity, and difficulty accessing their financial accounts. Banks that experience such breaches also face regulatory penalties, loss of user trust, and operational disruptions [3]. To protect sensitive data, banks often use biometric verification systems to authenticate users based on biological features such as fingerprints or facial recognition. Malicious actors may use deepfake images or videos to bypass facial verification systems and attempt to gain access to accounts and personal data.

Deepfake-supported attacks are not just theoretical. A 2023 report by Onfido notes that AI-supported spoofing attacks have overtaken legacy forms of security breaches across the

financial sector [4]. This trend is particularly evident in regions that have rapidly scaled their digital onboarding systems. The scale of these risks underscores the need to develop robust deepfake detection systems that serve as early warning flags for malicious activity.

In the banking context, such systems must meet stronger requirements, including robust performance across diverse generation methods, low false-positive rates, low verification latency, and non-discriminative performance [5]. In addition, these systems must be agile and easily adaptable to changing deepfake methods within the threat landscape.

1.1.2 Concerns with Generalization

As with many diverse classification problems, detecting deepfakes faces a central limitation in generalization. Generalization is the ability of a model to retain performance across subsets of the input space. Many existing deepfake detection systems fail to generalize well due to overfitting on the generation method used for their training data. For example, a detector trained on GAN-produced images (e.g., FaceSwap or StarGAN) often struggles with recognizing a forgery produced by a different method (e.g., DeepFakeFace or diffusion-based methods) [6]. Different generation methods produce unique artifacts or digital signatures in their output images. As models are trained, they learn to target and identify those specific phenomena, which may not transfer to other generation strategies. As digital manipulation techniques evolve rapidly, this results in an ecosystem where a successful detection model becomes outdated in a matter of months [7]. Improving the generalization and extensibility of model architectures would aid in designing a system that retains performance across generation methods and can be expanded to address newer technologies.

1.2 Key Contributions

This thesis explores two alternative ensemble architectures that aim to improve generalization and adaptability in deepfake image detection. The key contributions highlighted in this thesis are as follows.

1. Show that basic ensemble strategies can improve generalization of deepfake detection.
2. Establish guiding principles on constructing ensemble models, especially in selecting models to include in the ensemble.
3. Extend ensemble methods to include adjudicator models that dynamically weights predictions based on how reliable each sub-model is likely to be for a given input image.

Chapter 2

Related Works

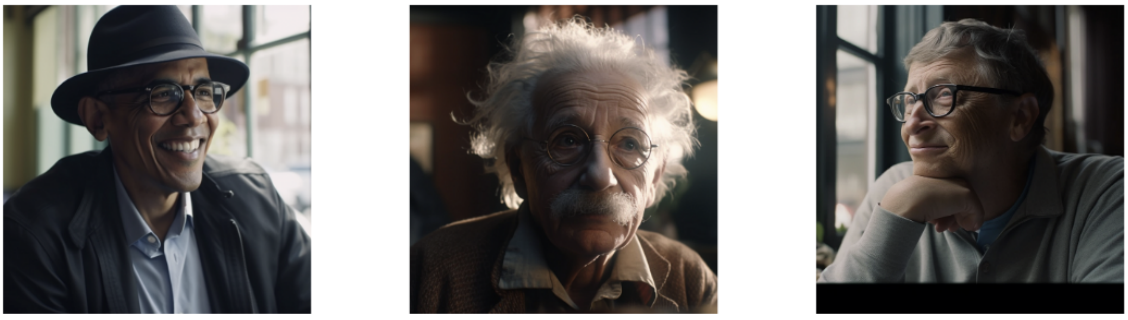
The following sections outline the relevant background information around the work in this thesis. We first outline the differences between types of altered media, then review common generation and detection methods, and finally cover existing ensemble strategies.

2.1 Altered Media

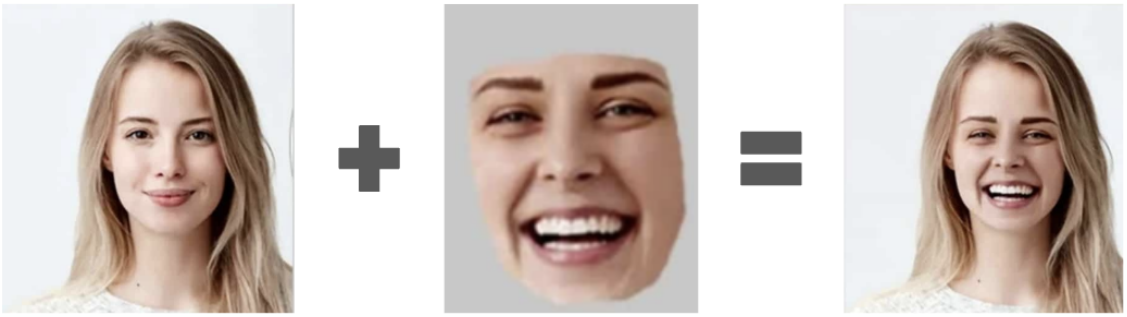
In this thesis, we use the term deepfake in a broad context, as often used colloquially, to refer to altered media that seeks to represent a fake identity. While this colloquial form of the term will continue to be used, it is important to highlight the differences between deepfake, face-swapped, and manually altered images.

The technical description of a deepfake specifically refers to media produced by a deep generative model with the intent of creating or altering an identity, as seen in Figure 2.1a. This typically refers to face renderings fully generated from noise or heavily edited to simulate an identity [2]. This contrasts with face-swaps, which involve transplanting a mesh of a target's face onto an otherwise minimally altered source media, as seen in Figure 2.1b. This is typically conducted using variational autoencoders or other transfer methods, which maintain the pose and expression of the target [8]. Each strategy of altering media results in unique artifacts, which are observable inconsistencies or anomalies that are produced during the

image generation process. Deepfakes are often identified due to texture or latent noise, while face-swaps are often identified by visual artifacts or breaks in the transition between the target and source [9].



(a) Images generated using generative models, Stable-Diffusion v5.



(b) Image generated using an online face-swap tool.

Figure 2.1: Examples of images generated using different media alteration strategies.

There are other forms of altering media that are also encountered, including morphing, warping, and other conventional photo-editing methods. While simpler in origin, these methods are still quite effective in passing visual verification. However, as they are not within the scope of AI-produced synthetic media, these types of images are not included within this thesis. Understanding the distinctions between types of altered media is critical for designing successful detection systems. The identification of the types of artifacts present in a diverse set of synthetic images allows for the scope of detection systems to be broad enough to catch the most commonly used manipulation pipelines.

2.2 Deepfake Generation

This thesis focuses on three families of synthetic image generation: Generative Adversarial Networks [GANs], Variational Autoencoders [VAEs], and Diffusion Models. Each family represents a different strategy for creating synthetic content and produces unique visual artifacts during generation. As a result, each family is successfully classified by a different set of detection models. By understanding what artifacts are produced by these image generation families, a more robust detection system can be developed.

2.2.1 Generative Adversarial Networks [GANs]

The first family of generators are Generative Adversarial Networks [GANs]. This framework uses two competing networks called the generator and the discriminator. The generator has a goal of producing a realistic image, while the discriminator has a goal of differentiating between real and synthetic images. They contrastingly work together during training to teach the generator how to fool the discriminator by improving the quality of its output image [10].

Within the family of GAN models, there are many variants that seek to improve different aspects of the output image. ProGAN introduced a training paradigm where the generator and discriminator grow from low to high resolution. This approach improved the synthesis of fine-grained detail and produced more photorealistic faces [11]. To improve multi-domain image-to-image translation, StarGAN introduced domain classification loss and reconstruction loss [12]. This allows StarGAN to flexibly transform images across age, hair color, or emotion. Recently, more advanced GAN-based models have improved the realism of output images and allowed for control of parameters such as the pose, expression, and lighting conditions for the output image [13]. StyleGAN introduced a style-based generator architecture where style-based modifications are added at each convolutional layer [14]. The separately learned latent space leads to more controllable behavior in the image generation process.

Images that are generated using GAN models are known to exhibit some common artifacts. These include boxed textures, bands of spikes in their frequency space, and reflections that do not match real-world physics [15]. While these artifacts are often targeted by detection models, they are starting to become less frequent in newer versions of GAN generators. This supports the need to update detectors for newer GAN models, as a detector trained only on early GAN outputs would exhibit poor performance on new architectures.

2.2.2 Variational Autoencoders (VAEs)

Variational Autoencoders constitute another popular architecture and are especially used in face swaps. VAEs use a probabilistic encoder-decoder setup to modify or create images. The encoder is responsible for generating a latent distribution, typically Gaussian, from the input image. The decoder, on the other hand, is responsible for reconstructing the image using samples it takes from the produced distribution. During training, the loss function of the model is determined as a combination of loss through reconstruction as well as a KL-divergence regularization term that monitors the behavior of the encoder [8,9].

Using this structure, VAEs are able to support a high level of image consistency. This allows them to be used successfully in face-swapping. In this application, two VAEs are often trained in tandem. One of the models is used for the source face, and the other is used for the target face. The source identity is then projected onto the context for the target's pose and lighting. While this does create successful face-swaps, it also tends to leave behind artifacts at the boundary of the face and environment, typically presenting itself in lighting mismatch or visually apparent seams [16].

2.2.3 Diffusion Models

Perhaps the most common deepfake generation models are diffusion-based. These models operate by learning how to reverse a noise process, called de-noising [17]. Diffusion models work in two processes, a forward and a reverse diffusion process. The forward process takes an

initial data sample and progressively adds noise to the image until it is completely corrupted. This is typically mathematically represented as a series of conditional distributions often modeled as a Markov process [18]. In the reverse process, the model looks to recover the original data from the noise by learning how to reverse the steps of the chain. This is completed using a learned function that predicts the noise added in each step. By learning how to predict the reversal of noise in each step, the model becomes strong at developing high-quality data and high-fidelity images [19].

In the context of deepfakes, diffusion models are commonly used to alter facial expressions, construct a new identity into a masked region, or to create highly realistic new content [20]. That being said, diffusion models often preserve artifacts in frequency domains as well as in image textures that are inconsistent with real-world images.

2.3 Deepfake Detection

As rapidly advancing deepfake generation methods become readily available, the need for strong detection strategies has rapidly advanced as well. These systems span classic convolutional models, patch by patch classifiers, transformer architectures, and even repurposing generation models such as GANs. This section outlines the key families of detection models and highlights representative work for each of them.

2.3.1 Convolutional Neural Networks (CNNs)

CNN models have traditionally served as the backbone for early deepfake detection pipelines. These types of models excel at identifying pixel-level artifacts, such as blurring, texture repetition, and lower-level patterns in noise [21]. Extensions on the CNN architecture, such as CNN-F, are also commonly used to target CNN-based generators, which allow CNNs to generalize across different classes of images that share a generator architecture [22].

CNN-F showed that a CNN trained from images developed on a single generator can retain performance to other architectures when exposed to well-constructed post-processing and augmentation techniques [22]. Other CNN-based models, like GramNet, extend beyond patch-level inconsistencies and instead target global artifacts present in style and texture patterns [23]. GramNet introduces a "Gram Block" to the CNN pipeline, which computes a style-based representation of the image's texture [23]. This improves robustness under common editing techniques - such as compression, blur, and noise - that can be used to cover up small inconsistencies. CNN-based systems are foundational in deepfake detection due to their scalability and flexibility.

2.3.2 Transformer-Based Detectors

Recent work has also explored the use of Vision Transformers (ViTs) for deepfake detection. These models leverage self-attention mechanisms to identify cross-region relationships and global inconsistencies. Transformers are able to correlate regions of an image that are distant in their location, which helps to detect manipulations that are spatially distributed [24].

Recently proposed ViT-based models can compete with CNNs while also demonstrating better generalization to unseen deepfake types [24]. These models focus on targeting artifacts like warped geometry or inconsistent lighting that are present across the facial regions. Attention-based reasoning allows ViT models to better adapt to a wider range of manipulation techniques and maintain performance across a diverse set of images.

2.3.3 Patch-Based Classifiers

Patch-based approaches divide an image into smaller regions, or patches, and analyze each independently to identify localized forgery traces. These models excel at catching minute spatial artifacts, such as abrupt texture transitions, pixel-level misalignment, and blending seams that occur during face recompositing.

One prominent example is Patch-Forensics, which processes non-overlapping patches through a truncated CNN and then aggregates per-patch scores to arrive at a final classification. Patch-based methods are especially valuable for generalization, as they avoid overfitting to global image properties [25].

Extensions of these models aim to tackle the challenge posed by diffusion-generated images, which often exhibit near-photorealistic quality. Dolos, a recent weakly supervised framework, modifies Patch-Forensics using Xception as the backbone to not only detect but also localize manipulated regions within an image [26]. Among three proposed pipelines in their study, the patch-based variant shows state-of-the-art performance on unseen diffusion datasets, outperforming both traditional and transformer-based models in terms of detection accuracy and generalization.

2.3.4 Repurposed GAN Architectures

As GANs are often used in deepfake generation, it is intuitive to think that they could be repurposed to detect GAN-produced images instead. As GAN models contain a discriminator network that already seeks to solve this task, many researchers have looked at retraining the discriminators to catch their generator counterpart's output images. Since the discriminator is explicitly trained to differentiate between fake and real samples during adversarial training, it naturally learns to recognize subtle inconsistencies [27]. A retrained GAN discriminator can achieve 100% detection accuracy on its own generator within few epochs [13]. However, this performance does degrade when tested on images produced by other generators, as the discriminator often over-fits to the artifacts produced by its own generator.

2.4 Ensemble Models

Since this thesis focuses on ensemble strategies, it is pertinent to provide a brief overview of such methods. The underlying intuition behind ensembles is that classifiers trained on different datasets or architectures may capture distinct visual artifacts, generation cues, or biases [28]. Ensemble methods can be implemented in a multitude of ways, often related to the nature of the inputs they seek to classify. Regarding classifying images, there tend to be three representative strategies, which are outlined below.

2.4.1 Majority Voting

The most basic ensemble method is *majority voting*, where each classifier independently outputs a binary decision (real or fake), and the final label is determined by simple aggregation. Let $M_i(x) \in \{0, 1\}$ represent the output of sub-model i on input image x, with 1 indicating a prediction of an *altered* or *fake* image. The overall ensemble decision in a majority voting scheme $E(x)$ is given by:

$$E(x) = \begin{cases} 1, & \text{if } \frac{1}{n} \sum_{i=1}^{n} M_i(x) \geq \tau \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

where τ is the voting threshold, which is typically set to 0.5 for a simple majority but can be altered depending on the sensitivity of the ensemble model to fake labels. This formulation is transparent and easy to implement, but it assumes that all models are equally reliable across all inputs—a problematic assumption when some models generalize poorly across domains. For instance, Sabir et al. [29] evaluated ensemble methods on frame-wise predictions for deepfake videos and found that majority voting schemes failed to generalize across videos with varied compression or codec levels. Majority voting schemes can suffer when faced with distributional shifts in the input space, making them unreliable in cross-domain scenarios and real-world applications [29].

2.4.2 Fully Connected Decider

A second approach uses the outputs of sub-models not as final decisions, but as inputs to a learned decision network. In this case, each model produces a scalar confidence called a logit $z_i \in \Re$, which can be scaled to any range and combined to form an input vector:

$$\mathbf{z}(x) = [z_1(x), z_2(x), \ldots, z_n(x)]^\top \tag{2.2}$$

This vector is then passed through a small multilayer perceptron (MLP), typically composed of 2–3 fully connected layers, to produce the final prediction. This concept is introduced by Wolpert [30], where component model outputs are passed as inputs to a higher-level learner, allowing the system to produce more context-aware prediction weighting. This produces an output as seen in Equation 2.2, adapted from Wolpert et al. [30], where σ is the sigmoid activation for binary classification.

$$E(x) = \sigma\left(W_2 \cdot \mathrm{ReLU}(W_1 \cdot \mathbf{z}(x) + \mathbf{b}_1) + \mathbf{b}_2\right) \tag{2.3}$$

This architecture allows the ensemble to learn nonlinear interactions between sub-model outputs and adjust weights accordingly during training [31]. Wang et al. [22] demonstrated that a fully connected fusion network using CNN component models improved generalization to unseen GAN images. Similarly, Ricker et al. [6] applied a fusion network to patch-based predictions in the Dolos framework, showing improved robustness to different diffusion-generated regions.

2.4.3 Adaptive Reweighting via Boosting Techniques

A third ensemble strategy is to iteratively train new sub-models or meta-classifiers in a way that emphasizes examples where previous models performed poorly. Boosting frameworks such as AdaBoost or Gradient Boosted Decision Trees (GBDTs) operate by assigning sample-specific weights during training, focusing model capacity on hard-to-classify inputs [32].

In this setting, base classifiers M_i are trained sequentially, and each is given a weight α_i based on its individual performance. The final prediction is then a more accurate weighted sum:

$$E(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i \cdot h_i(x)\right) \tag{2.4}$$

where $h_i(x) \in \{-1, +1\}$ denotes the new binary prediction of model i. This formulation allows the ensemble to compensate for weak classifiers by down-weighing them and emphasizing those that reduce the classification error on difficult or ambiguous samples [32]. Boosting-style ensembles can also be applied to decision fusion layers, where the feature space includes model logits, image quality scores, and other handcrafted descriptors. For instance, Ghita et al. [24] implemented gradient boosting on an ensemble of vision transformer models, showing improved detection over a diverse in-the-wild deepfake dataset [33].

Chapter 3

Methods

Prior work in deepfake detection has shown that no single detection model performs consistently well across all generative methods. Models trained on specific generation techniques often capture artifacts or anomalies unique to that method, which limits their generalization. This work extends upon prior research by presenting an ensemble model capable of handling diverse deepfake types. We first collect a set of representative state-of-the-art detection models, each emphasizing a different feature extraction strategy ranging from spatial artifacts and texture patterns to frequency domain analysis and regional inconsistencies. We then evaluate the individual models over a diverse set of image classes collected across six datasets. This chapter highlights the process of selecting both the datasets and models, as well as dataset pruning and model fine-tuning as needed. It also covers individual model evaluation.

3.1 Dataset Selection

The first step in setting up the inquiry is selecting a diverse set of images. Six benchmarking datasets are selected: DeepFakeFace [DFF] [34], FaceForensics++ [FF++] [35], Individualized Deepfake Detection Dataset [IDDD] [36], ProGAN [PG] [22], StarGAN [SG] [22], and WhichFace [WF] [33]. Together, these datasets cover three categories of generated images:

diffusion models, VAE models, and GAN models.

Each of these datasets contains face images cropped with minimal backgrounds. This allows the images across datasets to display similar content as well as reflect the types of user-provided images that Itaú takes during identity verification. For each dataset, the sample images are either downloaded from open-source repositories or generated according to publicly available documentation. Each of the datasets is cleaned and size-balanced so as not to bias the model prediction evaluation toward the larger datasets. Table 3.1 shows a breakdown of the datasets selected as well as metadata on their size and year of production.

Table 3.1: Overview of datasets in this study.

Dataset	**Generation Methods**	**Year**	**Real**	**Fake**
DeepFakeFace [34]	Stable Diffusion v1.5 Stable Diffusion Inpainting InsightFace	2023	1500	1500
FaceForensics++ [35]	Deepfakes Face2Face FaceSwap NeuralTextures	2018	2707	2698
Individual [36]	Faceswap-GAN	2024	1500	1500
ProGAN [11]	ProgressiveGAN	2017	200	200
StarGAN [12]	StarGAN	2018	2000	2000
WhichFace [33]	StyleGAN	2019	1000	1000

Each of the datasets subscribes to one of the families of deepfake generation techniques: Diffusion images from DeepFakeFace; VAE images from FaceForensics++; and GAN images from Individual, ProGAN, StarGAN, and WhichFace.

3.1.1 Diffusion Set

To represent diffusion-generated images, the publicly available DeepFakeFace dataset has been sourced. This dataset contains generated images of celebrities using a range of diffusion models. These include Stable Diffusion v1.5, Stable Diffusion Inpainting, and InsightFace, [34]. The dataset consists of 120,000 different images, made up of 30,000 real images from the IMDB Wiki dataset, and 30,000 fake images generated using the three techniques described

above. All images in this dataset have a resolution of 512 by 512. DeepFakeFace is loaded from Hugging Face and re-partitioned in testing sets with 1,500 real and 1,500 fake images [7].

3.1.2 VAE Set

The face swap images are sourced from the Face Forensics++ dataset, which is partitioned from the collection of images compiled for the Face Forensics public image detection challenge [35]. This dataset is composed of images generated from 1000 original video sequences. The videos are generated using four different face manipulation methods: Deepfakes [37], Face2Face [38], FaceSwap [35], and NeuralTextures [39]. The images are all cropped to contain a frontal view of the face without occlusions.

3.1.3 GAN Set

To include a variety of GAN models, several datasets are selected. The Individualized Deepfake Detection Dataset [36] uses Faceswap-GAN [40] to produce images of 45 specific individuals, with a total of 23,000 authentic images and 22,000 deepfake images. These images are divided into a training dataset, collected from the CelebDFv2 dataset [23], and a testing dataset, composed of images originally sourced from the CACD dataset [41].

The ProGAN and StarGAN datasets are generated by running the pretrained ProGAN [11] and StarGAN [12] models on images from CelebA [42]. The real faces are drawn from CelebA, and the fake image pairs are generated by each of the GAN models. These images are cropped on the long edge (center crop length is exactly the length of the short edge) and then resized to 256×256.

The last data source is WhichFace [33]. This dataset includes 1k real face and 1k fake face images scraped from whichfaceisreal.com [33]. This website holds both real images and Style-GAN generated faces. These images are downloaded, compressed in JPEG, and resized from 1024 x 1024 to 256 x 256.

3.2 Model Selection

As mentioned earlier in this chapter, chosen models target subclasses of the images. This allows the ensemble strategies to be evaluated based on how the combined models perform against the different image classes targeted by the component models. Each of the chosen component models, outlined below, is also individually benchmarked to facilitate a comparison to the ensemble architecture. Table 3.2 below shows the chosen models as well as the method of generation they target. It also provides the names of the models used to generate the set of image data used during their pre-training stage.

Table 3.2: Overview of models used in this study.

Model Name	**Year**	**Deepfake Generation**	**PreTrained Image Sources**
CNN-F [22]	2020	GAN	ProGAN
MesoNet [43]	2018	VAE	Deepfake Face2Face
DCT [6]	2022	GAN	ProGAN, StyleGAN, ProjectedGAN
		Diffusion	DDPM, IDDPM, ADM, PNDM, LDM
Dolos [26]	2024	Diffusion	Repaint–P2, Repaint–LDM

3.2.1 CNN-Fingerprints [CNN-F]

CNN-fingerprints is a specialized CNN that identifies artifacts common across GAN-generated images. This model specifically looks to extract finer artifacts typically created by generators. CNN-F shows better performance in capturing subtle irregularities that are present in synthesized images and pass through the pooling layer of GAN generators [22].

Prior deepfake detection models often fail to capture low-level artifacts that are consistent across varied GAN architectures, limiting their generalization. In the context of an ensemble model, CNN-F provides insight into the smaller pixel-level artifacts that are not caught by models that focus on overall textural elements or contextual discrepancies [22]. CNN-F itself

also shows promise for decent generalization, as the artifacts it targets are present across GAN-based deepfake generation methods.

3.2.2 MesoNet

Face-swapping techniques often introduce subtle inconsistencies in facial textures — artifacts that are too localized for high-level semantic detectors and too diffuse for low-level pixel analyzers. To build an ensemble that is successful against face-swap, the second model introduced into the study is MesoNet, proposed as a compact CNN architecture specifically designed to detect mid-level forgeries [43]. The goal of MesoNet is to target facial forgeries by focusing on discrepancies present in the facial textures. It was originally developed to analyze video inputs, but since the model analyzes each frame of the video individually, it can still be used on single images. The model uses specially selected kernel and pooling sizes to capture facial-specific digital manipulation [43].

Prior work on ensemble models for deepfake detection often underperforms on face-swaps as the component models don't target subtle facial inconsistencies. Within our ensemble model, MesoNet efficiently processes and extracts relevant features from the facial regions in the input image, which can aid in improving the overall robustness and accuracy of the ensemble model.

3.2.3 Discrete Cosine Transforms

The DCT-based approach [6] looks for identifying artifacts created by diffusion-based image generation. These artifacts are typically present in frequency transforms of the image. The method begins by applying a Discrete Cosine Transform (DCT) to the input image.

This transformation exposes artifacts in the frequency domain that are developed during the denoising process of diffusion models. In this work, a 2D Discrete Cosine Transform, adapted from Ricker et al.[6], is applied on Image I to produce an output T, where α represents normalization factors and M,N are the height and width of the image patch,

respectively:

$$T[u,v] = \alpha_u \alpha_v \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I[i,j] \cos\left[\frac{\pi}{M}\left(i+\frac{1}{2}\right)u\right] \cos\left[\frac{\pi}{N}\left(j+\frac{1}{2}\right)v\right] \tag{3.1}$$

$$\alpha_u = \begin{cases} \sqrt{\frac{1}{M}}, & \text{if } u = 0 \\ \sqrt{\frac{2}{M}}, & \text{if } 1 \leq u \leq M-1 \end{cases} \tag{3.2}$$

$$\alpha_v = \begin{cases} \sqrt{\frac{1}{N}}, & \text{if } v = 0 \\ \sqrt{\frac{2}{N}}, & \text{if } 1 \leq v \leq N-1 \end{cases} \tag{3.3}$$

By isolating frequency discrepancies, the DCT classification approach obtains a stronger performance against diffusion-based image generation. Two formats of the DCT model were reproduced for this work, originally sourced through GitHub from Ricker et al.[6], with varied compression factors. These are referred to as DCT(0.1) and DCT(0.5) in this thesis. Within our ensemble framework for deep fake detection, the DCT-based module enhances our ensemble's robustness by adding a frequency-focused perspective to the detection process.

3.2.4 Dolos

The last model included in the ensemble is the Dolos model, which is based on Patch-Forensics. Patch-Forensics truncates two other models, Resnet and Xception, to produce a binary deepfake prediction on "patches", which are small regions of the image. Each of these binary patch classifications is then combined to produce a classification for the image as a whole [25]. Dolos is a version of the patch-forensics model that is specifically retrained on diffusion-generated images [26]. The authors of Dolos provide three setups for the model, depending on the amount of information evaluated between patches. This work chooses Setup B, which targets partially manipulated images without localization details. This setup fits the context of a biometric verification system, where it is likely that there is manipulation around

only some regions of the image, with minimal manipulation in environmental contexts.

The inclusion of Dolos in our ensemble model introduces a new deepfake detection strategy that takes advantage of differences between deepfake manipulated regions and the background, which is often not leveraged.

3.3 Individual Model Evaluation

The models are all sourced from publicly available codebases, and their pretrained weights are downloaded as per the researcher's documentation. Since this thesis focuses on improving classification for unseen or untrained datasets, the loading of pre-trained models better simulates a real-world application of the ensemble in a verification system setting.

The models are modified to output a prediction ranging from 0 to 1, with 1 indicating the label for a deepfake image and 0 the label for a real image. It is possible that an arbitrary threshold of 0.5 for each model might not produce the highest fidelity model. The weighting process of the ensemble model provides an opportunity for the individual model thresholds to be rebalanced; this means that the model's performance in the ensemble is likely higher than against a 0.5 threshold. To address this, the ROC curves for each model and precision score are further examined to understand how the models behave and evaluate the quality of the votes that the models provide. In the ROC curves, a point is placed at the spot on the curve corresponding to the threshold with the maximum J-Score, defined in [44]. The threshold for each model on each dataset is defined as such:

$$t^* = \arg\max_t \left[\mathrm{TPR}(t) - \mathrm{FPR}(t)\right] \tag{3.4}$$

The performance of each model is evaluated using an updated threshold associated with the highest J-Score. Table 3.3 below shows the accuracy, precision, and recall of each model at the best threshold.

Table 3.3: Individual Model results for various datasets with a 0.5 threshold. Metrics include accuracy (Acc) split into total accuracy, deepfake accuracy, and real accuracy; precision (Prec); recall (Rec); and area under the ROC curve (AUC). Bold indicates the highest score on each dataset for some metrics.

Model	Dataset	Accuracy | Metrics			
		Total | Fake | Real	**Prec**	**Rec**	**AUC**
CNN-F	**DeepFakeFace**	52.9 | 70.8 | 35.1	52.2	70.8	53.1
	FaceForensics	56.7 | 60.9 | 52.6	56.2	60.9	58.9
	Individual	51.8 | 69.7 | 34.0	51.2	**69.7**	51.7
	ProGAN	55.3 | 52.0 | 58.5	55.6	52.0	53.7
	StarGAN	50.0 | 100.0 | 0.0	50.0	**100.0**	43.0
	WhichFace	50.4 | 2.9 | 97.9	58.0	2.9	45.9
	Average	51.5 | 51.7 | 51.2	51.4	51.7	51.5
MesoNet	**DeepFakeFace**	53.2 | 27.8 | 78.5	56.4	27.8	54.8
	FaceForensics	**68.7** | 74.8 | 62.5	66.6	**74.8**	74.6
	Individual	61.1 | 64.7 | 57.4	60.2	64.7	64.7
	ProGAN	53.5 | 21.5 | 85.5	59.7	21.5	49.7
	StarGAN	51.8 | 67.0 | 36.7	51.4	67.0	50.6
	WhichFace	51.4 | 68.3 | 34.5	51.0	68.3	50.7
	Average	57.3 | 39.8 | 74.7	61.1	39.8	57.3
DCT(0.1)	**DeepFakeFace**	54.3 | 10.9 | 97.7	82.8	10.9	**77.8**
	FaceForensics	67.5 | 42.4 | 92.5	**88.9**	44.4	**85.2**
	Individual	**65.6** | 43.4 | 81.7	70.2	43.4	**69.3**
	ProGAN*	**100.0** | 100.0 | 100.0	**100.0**	**100.0**	**100.0**
	StarGAN	**95.2** | 93.2 | 97.2	**97.1**	93.2	**98.8**
	WhichFace	98.8 | 98.5 | 99.0	99.0	98.5	99.9
	Average	**72.2** | 58.3 | 86.1	80.7	**58.3**	**77.8**
DCT(0.5)	**DeepFakeFace**	51.0 | 3.1 | 99.5	**83.0**	2.6	77.1
	FaceForensics	54.6 | 14.8 | 94.4	72.3	14.8	64.4
	Individual	56.9 | 23.3 | 90.3	**70.6**	23.3	65.8
	ProGAN*	**100.0** | 100.0 | 100.0	**100.0**	**100.0**	**100.0**
	StarGAN	91.7 | 89.4 | 94.0	93.7	89.4	97.1
	WhichFace	**99.3** | 99.2 | 99.4	**99.4**	**99.2**	**100.0**
	Average	68.6 | 42.8 | 94.3	**88.2**	42.8	76.9
Dolos	**DeepFakeFace**	**68.1** | 86.7 | 49.4	63.2	**86.7**	71.3
	FaceForensics	50.1 | 0.0 | 99.9	0.5	0.0	19.8
	Individual	50.8 | 35.1 | 66.4	50.9	35.1	50.5
	ProGAN	50.0 | 0.0 | 100.0	50.0	100.0	27.4
	StarGAN	50.0 | 0.0 | 100.0	50.0	100.0	0.9
	WhichFace	50.0 | 100.0 | 0.0	0.0	0.0	34.1
	Average	52.9 | 14.9 | 90.7	61.6	14.9	23.9

As expected, there is varied performance of the models across different generation methods. As seen in Table 3.3, the DCT models outperform other component models over the entire collection of images, scoring the highest overall accuracies at 72.2% and 68.6% for DCT(0.1) and DCT(0.5), respectively. The DCT models are particularly strong on GAN-based images, with near-perfect classification on ProGAN and StarGAN datasets. This is likely a result of the DCT model holding GAN-based images in its pretraining data, as seen in Table 3.2. However, the DCT models do not generalize well to diffusion images. On diffusion-generated images in the DeepFakeFace dataset, the DCT models overclassified images as real, resulting in poor recall scores at 10.9% and 2.6%.

On the other hand, the Dolos model successfully classifies these diffusion-generated images, with a total accuracy of 68.1% over the DeepFakeFace dataset. However, it does not generalize this performance to the other image generation families, with near-random classification over the GAN and VAE images present in the other datasets. MesoNet also shows a similar lack of generalization. It successfully classifies the images from FaceForensics++, representing VAE-generated images, but cannot retain this performance to images in the diffusion or GAN-based image families.

It is clear from the results in Table 3.3 that the individual models fail to generalize well to unseen image families. When presented with families of images outside of its training scope, the models all show a drop in accuracy, precision, recall, and AUC. These results further support the motivation for constructing ensembles to improve model generalization in deepfake detection. However, the value of an ensemble depends on understanding the specific strengths and weaknesses of its components. Different deepfake generation methods produce visual and statistical artifacts: diffusion-based techniques often leave subtle regional inconsistencies, VAEs can introduce compression-like texture patterns, and GANs tend to embed low-level convolutional traces. Evaluating each model within these separate families allows us to identify where a given detector excels or fails, informing how models should be combined in an ensemble.

To gain context on how to ensemble these component models, the ROC curves and performance of component models are evaluated independently for each family of images included in this study. This section further explores the results of the individual models across three families of images: diffusion, VAEs, and GANs.

3.3.1 Diffusion

The first family to address is the diffusion-generated images. These images are present in the DFF dataset, using Stable Diffusion v1.5 and Inpainting to generate samples. In this family of images, the only model that shows promise in classification is the Dolos model. Dolos achieves a higher than random accuracy of 68% across the diffusion-generated images. In comparison, the next highest performing model, DCT(0.1), holds a near-random accuracy of only 54.3% over DFF. However, it is noted that Dolos tends to over-classify images as positive. This results in a high false-positive rate, which presents itself in a low real-image accuracy at 49.4% and a lower precision at 63.2%.

In a biometric verification context, having a higher false-positive rate is not of immediate concern. Positive tags coming from a model like Dolos can be used as an initial screen to tag images that would require further manual inspection or more documentation from the user. Including an *over-eager* model like Dolos also allows us to explore how ensemble models can combine Dolos predictions with other models to soften the false positive rate of the combined system.

When observing the ROC curves of each model on the diffusion images, Figure 3.1, it is clear that the Dolos model is not the only model with a positive performance. The DCT models show potential to classify diffusion images, with strong ROC curves; however, they do so at extreme thresholds. Based on the ROC curve, it is likely that the DCT models can achieve a high precision when their threshold is set very low, or in other words, if they are made to be highly sensitive. The DCT model does, however, under-classify images as deepfake, with a low recall metric on the DFF dataset. However, when observing the ROC

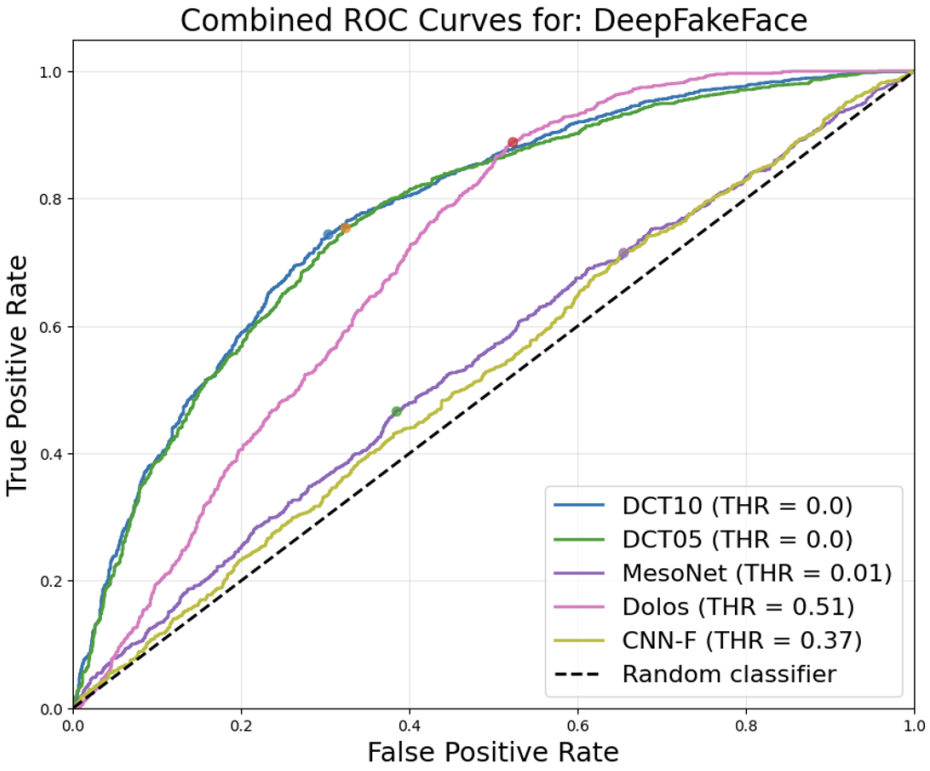


Figure 3.1: ROC curves of each model over diffusion generated images in DeepFakeFace

curve and high precision metric at around 83%, it is clear that the diffusion generated images DCT does flag as deepfakes are not likely to be false positives. Intuitively, high precision should suggest that a model is a good candidate for the ensemble strategy, as it is unlikely to provide a *vote* for an image that is not deepfake.

3.3.2 VAE

In the results obtained on the VAE-based Dataset, FaceForensics++, we see a tradeoff between choosing recall and precision. MesoNet and DCT(0.1) both show comparable accuracy over these images at 68.7% and 67.5%, respectively. When prioritizing recall, MesoNet emerges as the stronger classifier, successfully flagging 74.8% of the deepfake images present in the dataset. In comparison, DCT(0.1) only flags 44.4% of the deepfaked images. When looking at the quality of the entire set of images, each model that flags as deepfake, scored based on precision, DCT(0.1) emerges as the stronger classifier. DCT(0.1) scores a

precision of 88.9% while MesoNet scores only 66.6%. This suggests that MesoNet acts as an *over-eager* classifier, capturing a larger collection of deepfakes, while DCT(0.1) acts as a *selective* classifier, capturing a small but precise collection of deepfakes.

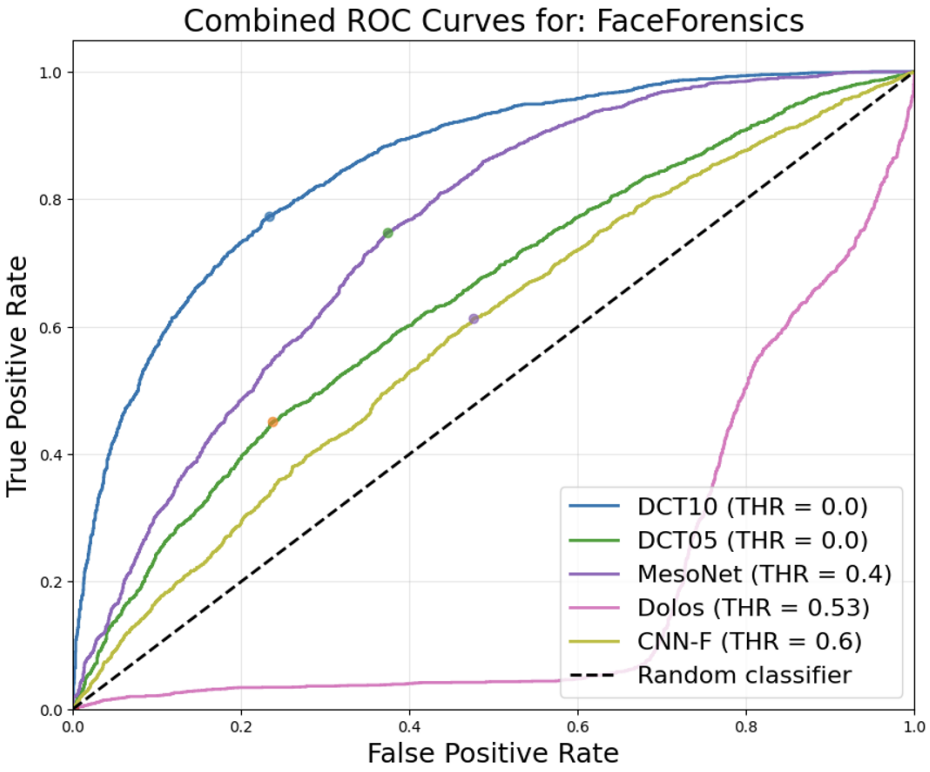
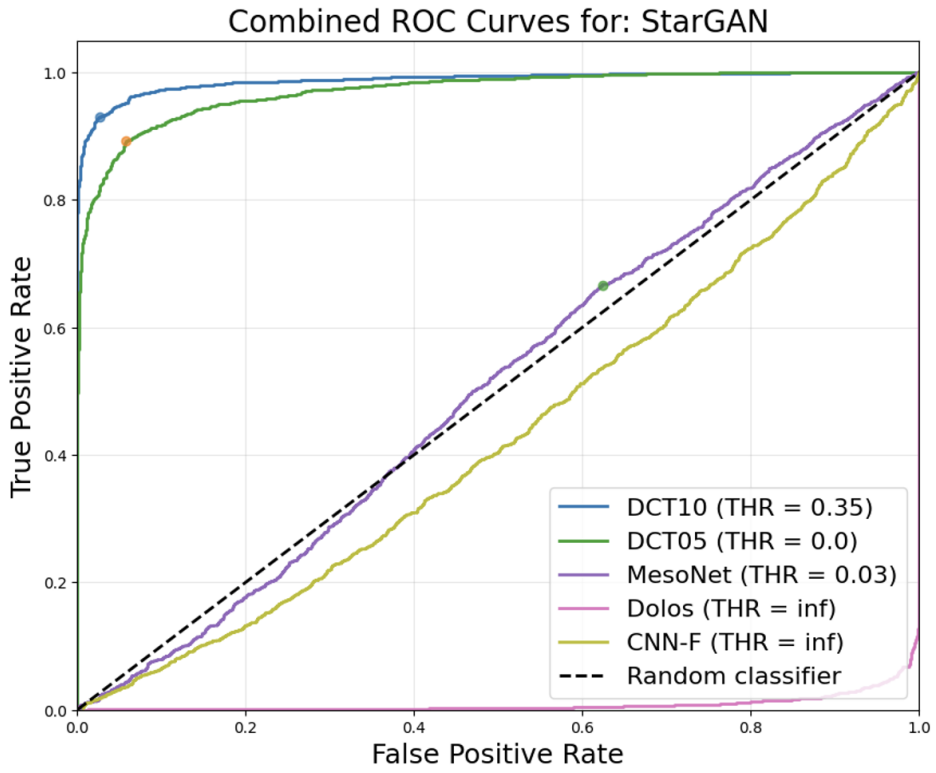


Figure 3.2: ROC curves of each model over face swapped images in FaceForensics++

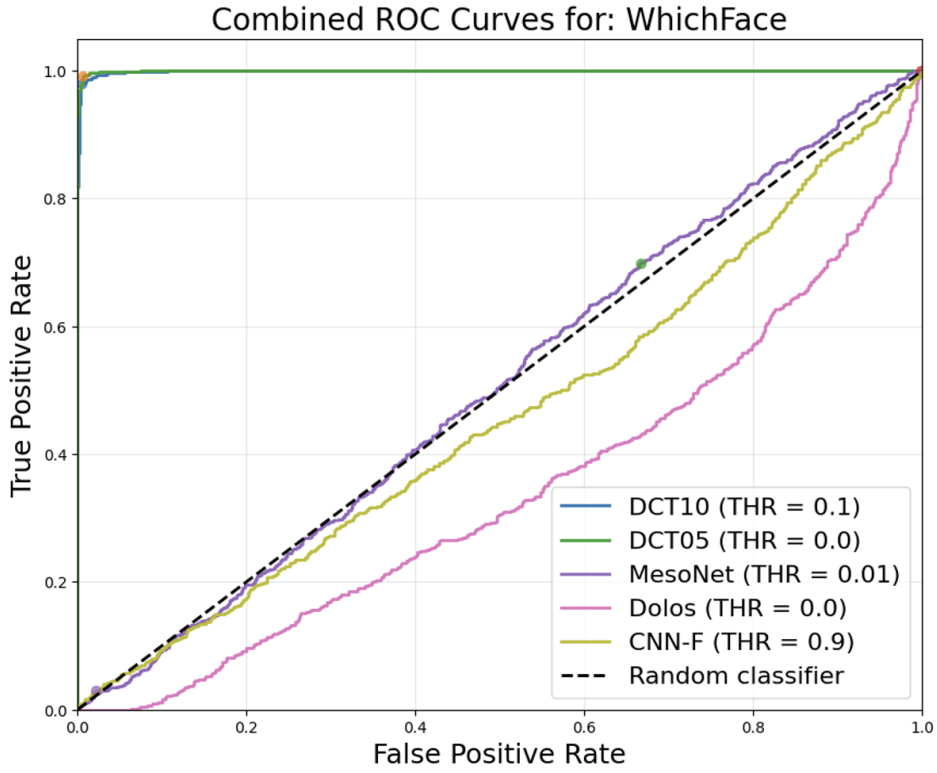
When observing the ROC curves, Figure 3.2, we see the AUC for DCT(0.1) was the largest for both datasets, at 85.2% and 69.3% respectively. As AUC is linked to a more precise classifier, it suggests that DCT could prove to be a more successful member of the ensemble than MesoNet, due to the lack of tendency to vote for real images.

3.3.3 GANs

When partitioning to GAN-based images, the DCT models were by far the highest performers. The ROC curves for the GAN datasets, seen on StarGAN and WhichFace in Figure 3.3, support using DCT models as the primary components of an ensemble.



(a) ROC curve on StarGAN



(b) ROC curve on WhichFace

Figure 3.3: ROC curves of each model over GAN-generated images

In both datasets, the DCT models achieve near-perfect classification, with AUC scores approaching 1. One interesting note about the GAN-based images is the tendency for Dolos to exhibit flipped performance, also observed on ROC curves for FaceForensics++ in Figure 3.2. This suggests that Dolos is a precise classifier on GAN-based images, but achieves this strong classification with flipped predictions. Intuitively, one might ask why the predictions of Dolos can't be flipped to achieve strong performance. This makes sense in the context of the GAN-based images alone, but flipping the labels for Dolos would then negate the predictions on the diffusion-based images. There is potential for Dolos to be dynamically flipped based on the artifacts present in the image. For example, if the image seems to be produced by a GAN generator, then flip the Dolos prediction; otherwise, leave it the same. This observation is the genesis of the attempt at expanding the ensemble model covered in Chapter 5. Based on the performance of the DCT models on GAN-based images, the ensemble should be constructed with DCT models as a base, with additional models included to improve robustness to VAE and diffusion-generated input samples.

Chapter 4

Blind Ensemble

As mentioned in Chapter 1, this thesis seeks to answer how ensembles of model outputs could improve generalization in deepfake image classification. In this section, exhaustive combinations of blind ensembles are formed to understand how the combination of model outputs affects the performance of the detection system. This type of ensemble is referred to as *blind*, as the final layer does not take in image information. Instead, the decision layer of the ensemble only observes the binary decision made by each component model. This structure allows the ensemble system to remain simple to construct and flexible enough to quickly add additional sub components without expensive retraining. By comparing the results of the different ensemble combinations to the performance of their individual models, we extend existing literature on ensemble models by answering guiding questions about ensemble construction.

1. What metrics suggest strong ensembles (accuracy, precision, etc)?

2. Does combining models with similar individual performance result in constructive or destructive ensemble performance?

3. Can ensemble strategies improve performance on unknown or non-target image classes (classes that both models were weak at classifying)?

4.1 Ensemble Architecture

The blind ensemble begins by passing the input image into each of the component models. Once the predictions from each of the components is computed, the ensemble combines those predictions to form a single binary classification [45]. A diagram of this structure can be seen in Figure 4.1 below.

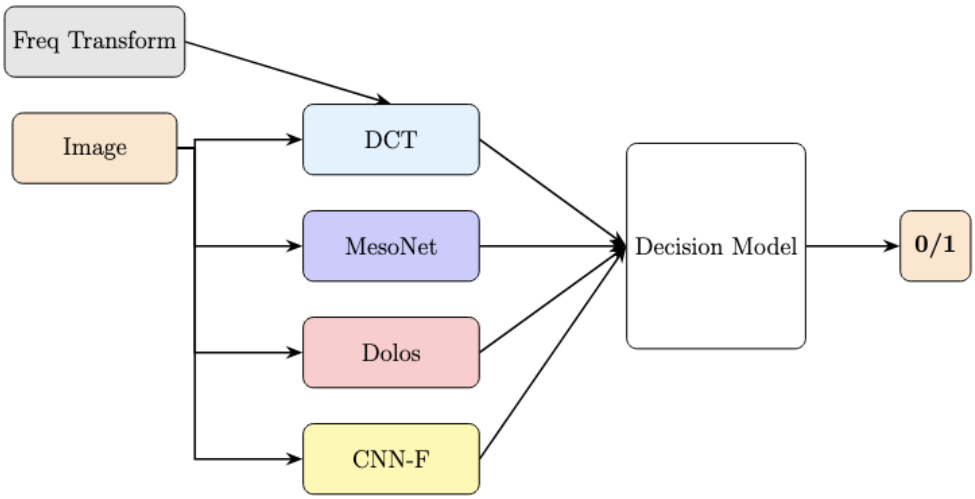


Figure 4.1: Architecture of blind ensemble-based deepfake detection.

The decision model chosen for this ensemble architecture is a random forest classifier [46]. This decision model is trained using an equal split from each family of the training data involved in benchmark evaluation. Across each of the equally sized dataset families, 80% of the samples are included in training and the remaining 20% are left for testing the ensemble model. These splits are completed with 5-fold cross validation. For each of the ensembles tested, both the average and standard deviation of each metric score are reported. Within the random forest model, each tree represents a partition of the training data and looks to define splits within nodes based on a minimization of Gini Impurity of its children nodes. This is defined by the following equation, where the Gini Impurity G is defined based on the proportion of deepfake and real images present, d and r respectively:

$$G = 1 - \left(d^2 + r^2\right) \tag{4.1}$$

The node splits are finalized when the child node's Gini Impurities reach 0, indicating that the samples are successfully split as deepfake and real between the child nodes. The random forest classifier used in this work has nine decision trees with a max depth of 3 layers.

4.2 Evaluation

The first step in evaluating the different ensemble models involves testing each permutation of the ensemble to understand how combining the different component models alter system performance. One of the models, CNN-F, does not show better than random performance for any of the image classes and was not introduced to the ensemble.

In Table 4.1 the average accuracy, precision, recall, and area under ROC curve (AUC) are listed for each of the ensembles model combinations.

Table 4.1: Performance comparison of different ensemble models.

Ensemble Model Combination	**Accuracy**	**Precision**	**Recall**	**AUC**
DCT(0.1) & DCT(0.5)	72.1 ± 0.8	83.1 ± 2.5	61.2 ± 3.5	80.1 ± 1.1
DCT(0.1) & Dolos	75.5 ± 0.8	79.5 ± 4.9	71.5 ± 3.5	85.3 ± 0.9
DCT(0.1) & Mesonet	76.3 ± 0.2	84.7 ± 0.8	68.0 ± 1.0	82.9 ± 1.1
DCT(0.5) & Dolos	74.0 ± 0.4	**87.7 ± 2.4**	60.1 ± 2.4	82.8 ± 1.3
DCT(0.5) & Mesonet	72.7 ± 0.8	78.2 ± 3.3	67.2 ± 2.6	80.0 ± 0.7
Mesonet & Dolos	72.6 ± 0.8	77.0 ± 3.4	68.2 ± 4.7	80.4 ± 1.2
DCT(0.1) & DCT(0.5) & Dolos	74.4 ± 1.8	84.2 ± 1.8	64.6 ± 4.4	84.2 ± 0.7
DCT(0.1) & DCT(0.5) & Mesonet	75.4 ± 0.8	85.5 ± 0.7	65.3 ± 2.2	82.3 ± 0.9
DCT(0.1) & Dolos & Mesonet	78.0 ± 1.9	86.9 ± 3.0	69.2 ± 3.2	**87.8 ± 1.3**
DCT(0.5) & Dolos & Mesonet	77.1 ± 1.7	84.0 ± 3.9	70.1 ± 5.6	85.2 ± 1.3
DCT(0.1) & DCT(0.5) & Dolos & Mesonet	**79.2 ± 1.5**	83.4 ± 2.1	**73.0 ± 2.0**	87.4 ± 0.9

As is expected, the combination of all four models shows improvement in the accuracy of the system. When both DCT models, Dolos, and MesoNet are introduced in the ensemble framework, the overall accuracy jumps to 79.2 ± 1.5%. It is important to note that due to the standard deviation across splits, this is not statistically much larger than some of the smaller combinations of models, namely DCT(0.1) & Dolos & Mesonet. The all-four ensemble shows a similar performance to ensembles constructed with swapped DCT models, suggesting that

combining similar models does not drastically improve the ensemble system. It is important to note that adding in similar models, DCT(0.1) and DCT(0.5), does not detract from the performance of the ensemble, which will be explored further in Section 4.2.1.

From Table 4.1, we also note that the ensemble of all four models does outperform the individual model performance across most of the families of data. This is particularly true within the image classes that did not have very strong individual performance, such as DeepFakeFace and FaceForensics++. This can be observed in the ROC curves of the all-four ensemble as compared to the individual models in Figure 4.2. For both DeepFakeFace and FaceForensics, the accuracy and AUC of the ensemble show an improvement over the individual models.

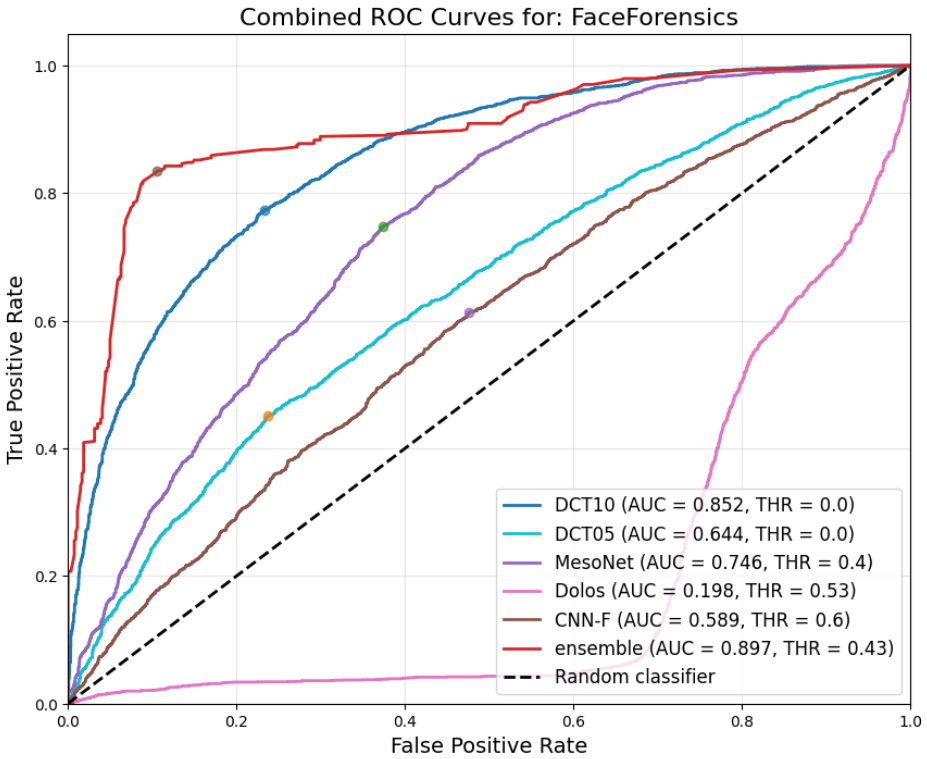


Figure 4.2: ROC curves for individual models and all-four ensemble on FaceForensics++.

The ROC curve of the all-four ensemble on the FaceForensics dataset shows that the ensemble could be pulling some performance from the Dolos model, as the rate of change of True positives over False positives follows near the inverse of the Dolos rate. It is possible

that the ensemble is able to implicitly flip the Dolos predictions when other models show cohesive responses, though it is hard to unpack exactly how the ensemble is addressing the combination of predictions.

While the all-four ensemble does perform strongly on the diffusion and VAE datasets, there are some exceptions to its improvement as compared to component models. In some of the GAN datasets, especially StarGAN and WhichFace, the all-four ensemble does not outperform some of the component models. This is not unexpected, as adding in votes from poor performing models on samples of images that have a perfect component component classifier would result in a combination that preserves some predictions from the poor classifiers.

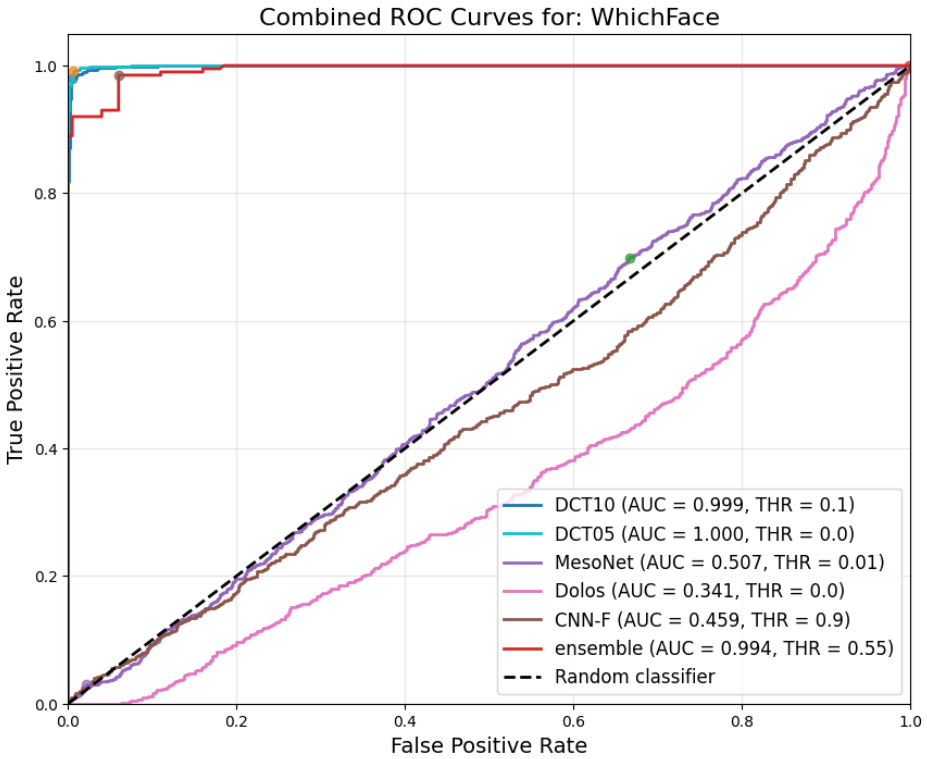


Figure 4.3: ROC curves for individual models and all-four ensemble on WhichFace

This is visible in WhichFace dataset, as seen in Figure **??**. While this presents in the examples where some components are near perfect, it does not present itself in examples with good but not ideal classifiers, such as StarGAN. In images from the StarGAN dataset,

the ensemble seems to combine the constructively combine the predictions from both DCT models and produce a more successful system, seen in Figure 4.4.

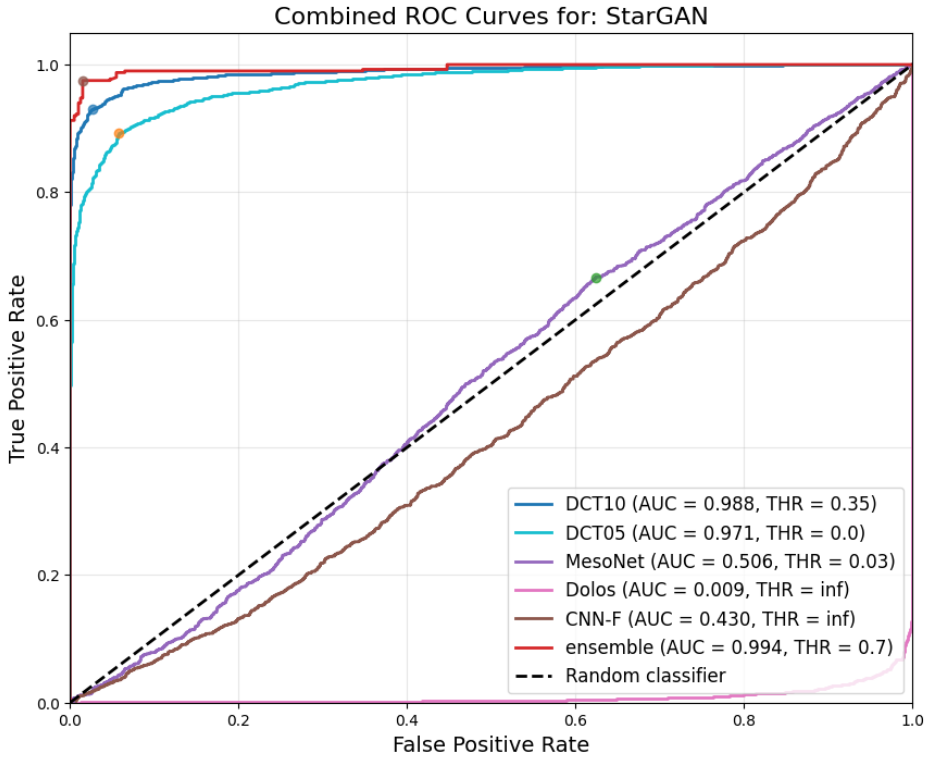


Figure 4.4: ROC curves for individual models and all-four ensemble on StarGAN

Using the results from each combination of ensemble models, the efficacy of certain techniques for ensemble model construction can be evaluated. This should provide some insight into guiding principles when chooses models to include in a naive ensemble structure. The following sections highlight the performance from on constructing ensembles based on certain metrics of the component models: similar/different performance, precision, and recall.

4.2.1 Combining Similar Models

The first behavior to observe is what happens when two similarly performing models are combined into an ensemble. It is not expected that combining similar models will result in drastic improvements to the ensemble [32]; instead, this inquiry looks to see if the combination of similar models reduces the ensemble performance. Let us say, for example, that model A

and model B tend to contain some false positives. If the ensemble model learns to listen to both models in a manner that reflects *OR* logic, then the ensemble model may lose precision by collecting the false positive predictions from both models. Conversely, it would also be concerning to observe the ensemble lose recall by only flagging cases where both model A *AND* model B report a deepfake. Now, as the inputs to the ensemble model are not binary decisions but rather a float reflecting the confidence in the prediction, there is more nuance to how the ensemble will handle such cases. We explore the combination of the two versions of the DCT model, DCT(0.1) and DCT(0.5), as an example of whether two models that detect similar types of deepfakes can be combined into an improved ensemble.

Table 4.2: Comparison of performance between DCT(0.1), DCT(0.5), and an ensemble of the two.

Model	**Dataset**	**Metrics**		
		Acc	**Prec**	**Rec**
DCT(0.1)	**DeepFakeFace**	54.3	82.8	10.9
	FaceForensics	67.5	**88.9**	44.4
	Individual	**65.6**	70.2	43.4
	ProGAN	**100.0**	**100.0**	**100.0**
	StarGAN	**95.2**	**97.1**	93.2
	WhichFace	98.8	99.0	99.0
	Average	**72.2**	80.7	58.3
DCT(0.5)	**DeepFakeFace**	51.0	**83.0**	2.6
	FaceForensics	54.6	72.3	14.8
	Individual	56.9	**70.6**	23.3
	ProGAN	**100.0**	**100.0**	**100.0**
	StarGAN	91.7	93.7	89.4
	WhichFace	**99.3**	**99.4**	99.2
	Average	68.6	**88.2**	42.8
DCT(0.1) & DCT(0.5)	**DeepFakeFace**	**54.9 ± 0.2**	81.7 ± 4.6	**12.7 ± 1.1**
	FaceForensics	**69.3 ± 3.6**	81.6 ± 2.9	**49.7 ± 8.5**
	Individual	63.2 ± 2.1	68.3 ± 2.0	**48.8 ± 5.3**
	ProGAN	99.3 ± 1.1	98.6 ± 2.1	**100.0 ± 0.0**
	StarGAN	83.4 ± 3.0	75.6 ± 3.2	**98.9 ± 0.4**
	WhichFace	91.1 ± 1.6	84.9 ± 2.2	**100.0 ± 0.0**
	Average	72.1 ± 0.8	83.1 ± 2.5	**61.2 ± 3.5**

Table 4.2 presents the accuracy, precision, and recall of the DCT models and their combination over all six data sets. The combination of DCT models only shows an increase in accuracy on the datasets that the component models do not perform well on: DeepFakeFace and FaceForensics. For the other families of images, the combination of similar models resulted in a lower accuracy. This is especially apparent in the WhichFace and StarGAN datasets, where the combination of the models performed 7.7% and 8.1% lower than the component models, respectively.

Stepping aside from accuracy, a more interesting observation arises when looking at the precision and recall of the ensemble. In most families of images, the ensemble observed a lower precision and higher recall than either component model. Referring back to the simplified binary example from earlier in this section, these results imply that the ensemble model is likely to hold more of an *OR* logic for the positive flags (deepfakes) and more of an *AND* logic for the negative flags (real). The ensemble thus becomes more sensitive to classifying images as real and broadens the collection of images classified as deepfake, explaining the boost in recall and subsequent loss of precision.

In a real-world setting, an increase in recall could support a stronger first-barrier verification system. Fewer false negative results mean that fewer malicious user inputs will pass through the detector as a real image. If the verification system has methods to manually double-check images that the detector flags as deepfakes, then the false positives can be cleared in a later stage of verification. This suggests that in the setting for this project, the boost in recall could justify combining similar models to improve the recall for the ensemble system.

4.2.2 Combining Different Models

The next inquiry surrounds combining models with different architectures. By combining two models that pick up on different artifacts present in images, the system could better generalize to a wider set of input images. However, it would not be surprising to see a slight drop in performance on families of images where a component model exhibits near-perfect

classification. To understand how two different models combine, we explore the ensemble of DCT(0.1) and Dolos. DCT shows positive performance on GAN-generated images and face-swaps, while Dolos shows positive performance on only the diffusion-generated models.

Table 4.3: Comparison of performance between DCT(0.1), Dolos, and an ensemble of the two.

Model	Dataset	Metrics		
		Acc	Prec	Rec
DCT(0.1)	**DeepFakeFace**	54.3	**82.8**	10.9
	FaceForensics	67.5	88.9	44.4
	Individual	**65.6**	**70.2**	43.4
	ProGAN	**100.0**	**100.0**	100.0
	StarGAN	**95.2**	**97.1**	93.2
	WhichFace	**98.8**	**99.0**	99.0
	Average	**72.2**	**80.7**	58.3
Dolos	**DeepFakeFace**	**68.1**	63.2	**86.7**
	FaceForensics	50.1	50.0	4.0
	Individual	50.8	50.9	35.1
	ProGAN	50.0	50.0	100.0
	StarGAN	50.0	50.0	100.0
	WhichFace	50.0	0.0	0.0
	Overall	52.9	61.6	14.9
DCT(0.1) & Dolos	**DeepFakeFace**	**71.0 ± 4.3**	78.2 ± 9.0	62.3 ± 17.4
	FaceForensics	**74.2 ± 2.2**	**90.7 ± 3.8**	**54.0 ± 5.5**
	Individual	64.2 ± 3.0	68.4 ± 4.8	**52.9 ± 3.1**
	ProGAN	98.3 ± 1.9	96.7 ± 3.4	**100.0 ± 0.0**
	StarGAN	81.2 ± 1.9	72.9 ± 2.0	**99.3 ± 0.7**
	WhichFace	87.0 ± 3.8	79.8 ± 4.5	**99.5 ± 1.1**
	Overall	**75.5 ± 0.8**	79.5 ± 4.9	**71.5 ± 3.5**

We show accuracy, precision, and recall for each individual model as well as their ensemble in Table 4.3. We see that the DCT(0.1) & Dolos ensemble boosts performance on weakly classified families of images, specifically, DeepFakeFace and FaceForensics. The ensemble of DCT(0.1) & Dolos achieves an accuracy of 71.0 +4.3% on DeepFakeFace and 74.2 +2.2% on FaceForensics, both of which are statistically significantly larger than both of the component models. However, there is a drop in ensemble performance on datasets where Dolos exhibits weak classification. For the GAN-based family of images, the DCT(0.1) & Dolos ensemble

under-performs compared to DCT(0.1). On StarGAN and WhichFace images specifically, the ensemble has a smaller accuracy of 81.2% and 87.0%, respectively, which is more than a 10% drop compared to the component models.

This behavior is also noted on the all-four ensemble ROC curve in Figure 4.3. Based on these results, combining different models seems to 'smooth' out performance between the component models, which reduces accuracy in strongly classified families of data. This seems to be a necessary tradeoff that should be evaluated prior to constructing an ensemble. Ensemble model developers should look to see if there is justification to sacrifice system accuracy in some classes of inputs for better generalization over the entire input space.

The ensemble of DCT(0.1) & Dolos has an equal or higher recall when compared to the component models over each dataset. The improvement in recall is also noted in the earlier ensemble of DCT(0.1) & DCT(0.5) mentioned in Section 4.2.1. This recall boost suggests that, once again, the ensemble airs on the side of caution by becoming sensitive to images predicted as real and producing positive predictions when the component models disagree.

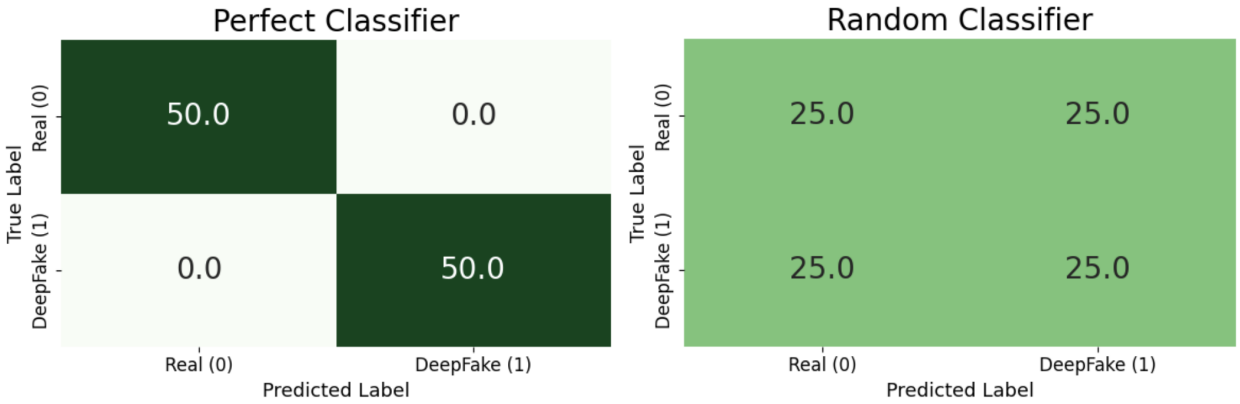
Over a normalized average of the families of images, the ensemble improves performance compared to the component models, with improvement in both overall accuracy and recall. This suggest that we can combine two models with different strengths to produce a more robust defense system, even if the individual models generalize poorly.

4.2.3 Ensembles based on Accuracy, Precision, and Recall

The last inquiry surrounds which metrics to listen to when constructing an ensemble. Based on benchmark testing displayed in Table 3.3, ensembles can be constructed by combining the highest performing models from each dataset based on accuracy, precision, or recall. In different real-world applications, different metrics of a classification system are prioritized. For example, some systems might prioritize having strong recall to avoid false negatives. Other systems might instead prioritize precision or accuracy to target false positives or true predictions. It is important to observe how prioritizing different metrics in the selection of

component models impacts the performance of the resulting ensemble. For ensemble based on Accuracy, the MesoNet & DCT(0.1) & Dolos ensemble is selected. For Precision, DCT(0.1) & DCT(0.5) is selected. Finally, for recall, the ensemble of all five models is considered.

In this section of the study, the confusion matrices of the models and their ensembles are studied. The confusion matrix is a strong tool to compare the performance of classification models by displaying the percentage of images that are: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). For the models in this study, positive and negative relate to the predicted label of deepfake or real, respectively. True and false reflect cases where the real label of the image matches or does not match the predicted label, respectively. Figure 4.5 below is an example of the confusion matrix for both a perfect and random classifier for reference.



(a) Perfect Confusion Matrix (b) Random Confusion Matrix

Figure 4.5: Reference examples of confusion matrices for a perfect and random classifier

Accuracy

As mentioned in the section above, we construct an ensemble based on accuracy involving MesoNet & DCT(0.1) & Dolos. MesoNet achieves the highest total accuracy over the FaceSwap family of images in the FaceForensics dataset. DCT(0.1) holds the highest accuracy over the GAN-based datasets. It does have a lower accuracy on the WhichFace dataset, at 98.8%, compared to the DCT(0.5) model, at 99.3%. Since both models are extremely close

to one another in performance on GAN images, and since combining the two near-perfect classifiers is not constructive, as seen in Table 4.2, DCT(0.5) is omitted from this ensemble. Making these nuanced decisions is part of constructing an ensemble, and two models have similar accuracy on high-performing families of images; it is advisable to choose one of the models to include.

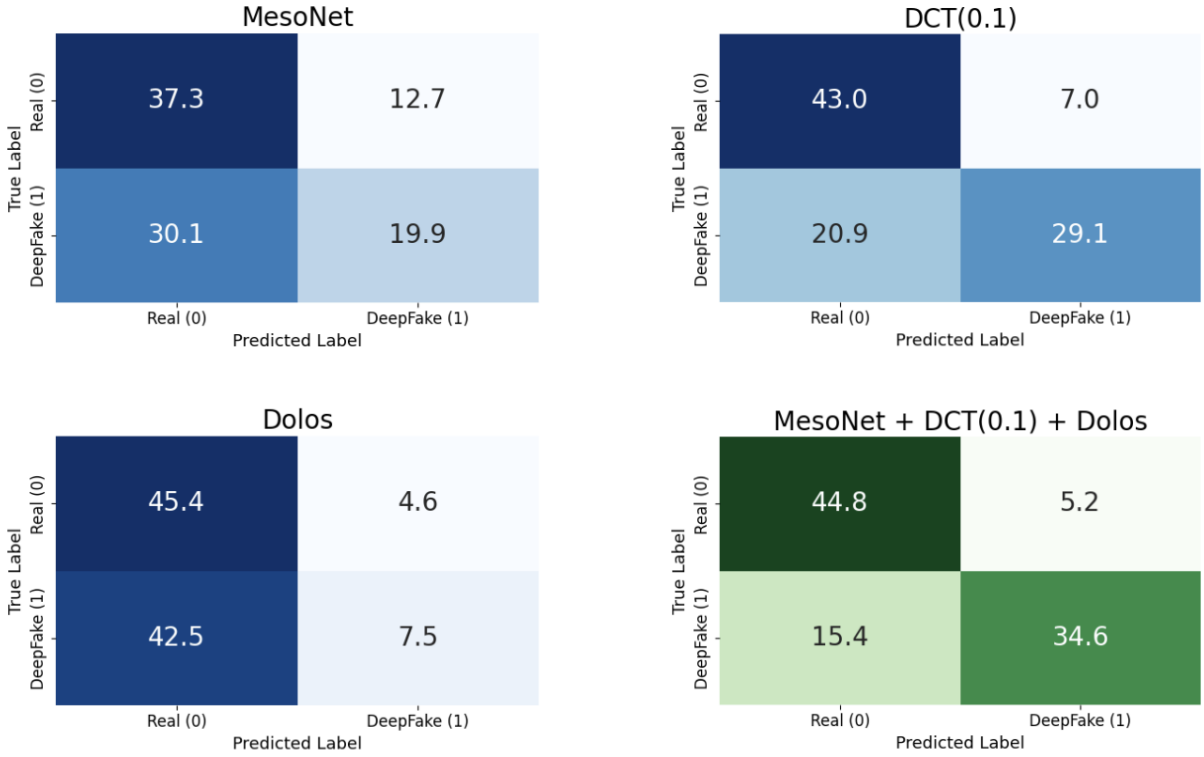


Figure 4.6: Comparison of confusion matrices for accuracy based ensembling.

The confusion matrices for each of the models and the ensemble are computed based on performance over the entirety of the testing image set, containing a third of the images from each family of deepfakes: Diffusion, VAE, and GANs. The confusion matrix of the MesoNet & DCT(0.1) & Dolos in Figure 4.5 shows an improvement in the accuracy of the ensemble, with only 20.6% of the image set incorrectly classified. This shows that the ensemble can increase the true positive rate without decreasing the TN rate when combined based on accuracy. However, the model is not strictly better in each category compared to the individual models. The Dolos model has fewer false positives when compared to the

ensemble, at 4.6% and 5.2%, respectively. Similarly, the Dolos model also has more TN when compared to the ensemble, at 45.4% and 44.8%, respectively. Nonetheless, the ensemble still shows better overall classification, with much stronger true positive and false negative rates. In addition, the ensemble maintains similar performance to the Dolos model and is not statistically significantly out of range of the performance of Dolos in the false positive and TN rates. This suggests that combining models based on accuracy across diverse image sets is a strong method to improve the generalization of the ensemble to a diverse set of images.

Precision

We can conduct a similar experiment on the performance of a model built based on the precision of the individual models. When looking at which models hold the highest precision for each dataset, we identify DCT(0.1) and DCT(0.5) as candidate models. This ensemble was already explored in Section 4.2.1, so we are already aware that the precision of the combined model did not improve when combining the similar models as compared to the performance of the individual DCT models. However, it is still insightful to observe the confusion matrices of the ensemble as compared to the individual to see if the resulting ensemble could be a better classifier.

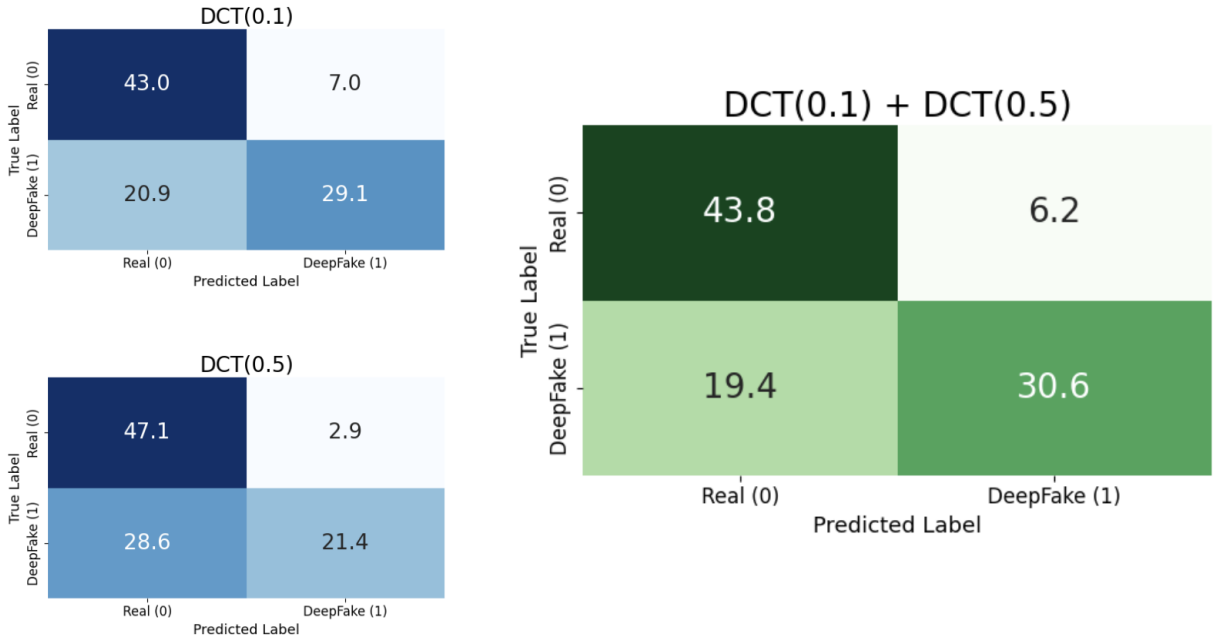


Figure 4.7: Comparison of confusion matrices for precision based ensembling.

Once again, we see that the resulting classifier, though lower in precision than the components, does show characteristics of being a more robust classifier when looking at the resulting confusion matrix in Figure 4.7. This is especially apparent in an increase in the true positive rate, which relates to deepfake predicted labels that are truly deepfake. We also see a reduction in false negatives, which relate to deepfake images that the model fails to classify correctly. These false negatives, which are represented in the lower left quadrant of the confusion matrix, are particularly concerning as they represent synthetic images that would pass through the system. While the ensemble based on precision did improve the false negative rate, it was not as successful as the ensemble based on accuracy. When prioritizing precision, the collection of component models tends to be more selective in the images they label as fake. This restricts how many images can pass into the decision model and caps the performance of the ensemble system.

Recall

The final combination of models is selected from the component models with the highest recall across the datasets. This constitutes all 5 individual models pulled for the study, whose results are in Table 4.1. For the Diffusion family of images, DeepFakeFace, Dolos exhibits the highest Recall at 86.7%. Following along, MesoNet holds the best recall for FaceSwap Images in FaceForensics, at 74.8%. For the images generated specifically by FaceSwapGAN, CNN-F has the highest recall at 69.7%. The remaining GAN models are split between DCT(0.1) and DCT(0.5). All five component models are combined in an ensemble to see if the resulting system retains performance from each of the individual models. If the ensemble produces a higher recall than the best performing individual model across the entire equally partitioned set of images, which is DCT(0.1) at 58.3%, then we can say that the ensemble based on recall performs successfully.

Figure 4.8 below shows that the ensemble combination exhibits stronger classifier performance, with an improvement in the accuracy of predictions. Interestingly however, the ensemble does not have a strictly better rate of false positives than the components. Both DCT models as well as the Dolos each predict less false positives than the ensemble model. However, the ensemble does have a larger number of total deepfake predictions as well as a better rate of true positives, which offsets the increase in false positive results. In the context of verification of images for security, a higher false positive rate is not as concerning as the false negative rates, as false positives can be rectified in further layers of security. The ensemble model based on recall does show fewest false negatives compared the other ensembles and individual models, meaning that fewer deepfake images can pass through the system undetected. This suggests that the recall-based ensemble model would be more successful than the individual models over the entire collection of image inputs.

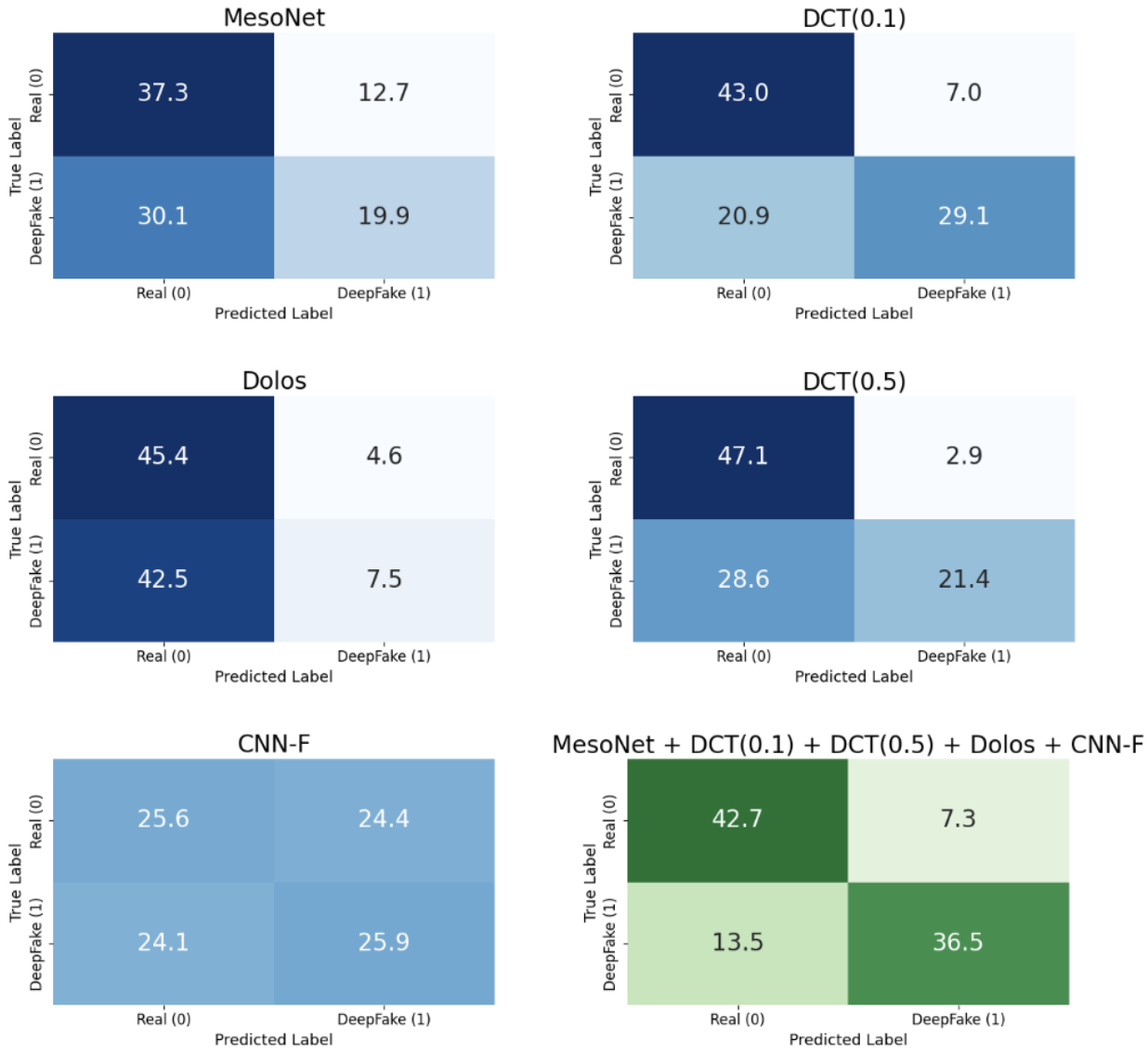


Figure 4.8: Comparison of confusion matrices for recall based ensembling.

We can see this improvement in the recall when observing Figure 4.9. The improvement in the ensemble recall is likely due to the reduction in false negatives that the ensemble model predicts. In the context of a verification system, these false negatives are particularly concerning. Seeing a reduction in the false negatives and improvement in recall suggests that constructing ensemble models creates stronger verification systems, even if the performance on a subset of images is reduced as seen in Table 4.2 and Table 4.3. In a real-world application, it is not known which type of synthetic images is being provided, and thus the improvement

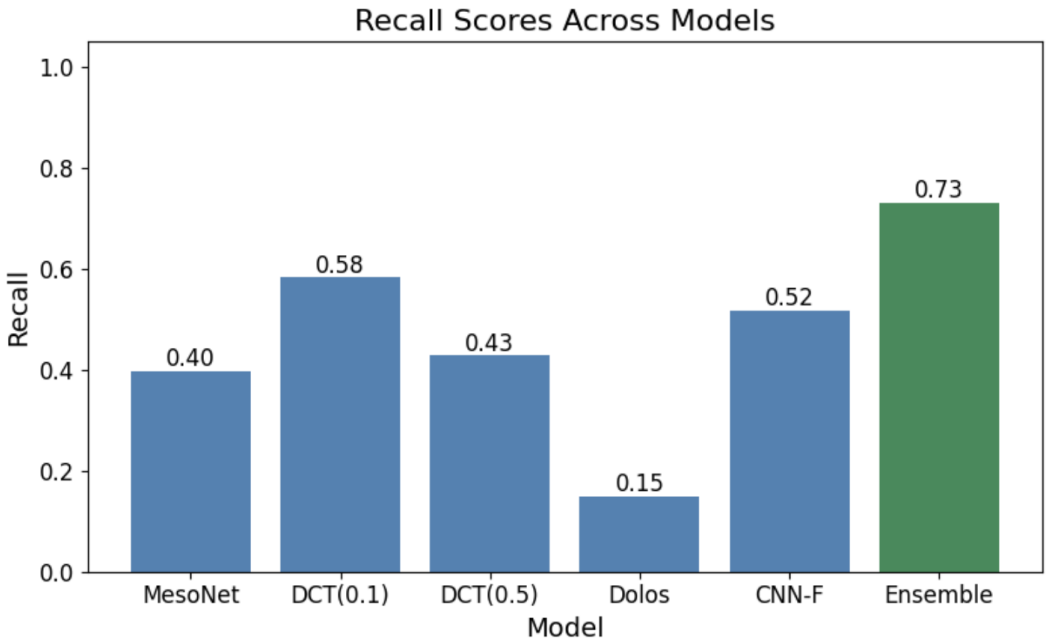


Figure 4.9: Comparison of recall of individual models vs ensemble.

over a larger diverse set of image types can justify small reductions in performance compared to select component models in specific families of inputs.

4.3 Discussion on Blind Ensembles

From the evaluation of the blind ensemble methods, we can see that the ensemble models show a consistent improvement in the classification of a diverse image set. In this section, some important notes on the application of blind ensembles will be discussed, namely: ensemble construction, important tradeoffs, and concerns for overfitting to training data.

Classifier Evaluation and Metric Prioritization

In this work, the individual models were assessed based on their accuracy, precision, and recall. While these are conventionally the main metrics used to assess binary classifiers, they do not fit every use case. In some applications, a single quadrant of a confusion matrix might be of particular concern, for example, false negatives. While a metric like recall would

account for false negatives, it does so in conjunction with the true positive rates, which in some use cases might not be of particular concern, for example, if the ensemble is part of a cascade of detecting systems that can catch false positive results later.

In the application for this thesis, identity verification, the false negative results are particularly harmful. This is the reasoning behind the focus on recall in later sections of the evaluation of the ensemble models in this work. For an application in identity verification systems, the ensemble models show a promising reduction in false negatives. Each ensemble configuration tested in Table 4.1 outperforms individual models on recall over the entirety of the input space. For example, the all-four ensemble achieves the highest recall on DeepFakeFace and FaceForensics as compared to any of the single sub-models, as seen in Table 4.1.

Another important note is on the difference in harmful impact attributed to each of the families of input data. In many high-risk applications, for example, certain selected error types could also be more impactful than any collective metric [5]. For example, a security developer might specifically care about a subset of their input space being correctly evaluated, even if overall accuracy drops. In our application, an example of this concern might present itself in the classification of Diffusion models as compared to GAN or FaceSwap images. As well-constructed GAN models hold artifacts that are at a smaller scale, they are harder to identify visually as compared to global texture artifacts that might be present in a diffusion-generated image. This could result in a desire to tune the ensemble to improve recall specifically on one class of the images rather than an overall recall metric.

As mentioned earlier, this work provides examples of ensemble strategies and some guidelines for constructing said ensembles, but they are by no means a global standard. It is important that the construction of the ensemble model is done to target its specific use case and is built based on a well-defined metric that fits the context surrounding its application.

Tradeoffs in Ensemble Composition

One observation from the combination of ensemble models is that adding more models does not always improve performance across all families of synthetic images. It was observed that some combinations that raise global accuracy tend to weaken performance on datasets that are classified near perfectly by one of the component models. For example, in Table 4.3, we see the ensemble of DCT(0.1) and Dolos shows an improvement in overall accuracy, but a reduction in the performance on StarGAN and WhichFace sourced images. Poor performance on specific methods of generation, in this case, GAN and diffusion-based fake images, can arise from mismatches in the specialization of the component models within the ensemble. Dolos is observed to perform extremely poorly in its positive predictions on GAN models, seen with the extremely low recall values in Table 4.1. It is likely that adding the almost reversed votes from Dolos resulted in a drop in performance of that ensemble on GAN-sourced images.

To avoid situations like this, designers of ensemble models should consider what class of images is most concerning and tailor the composition of the models to target the more common synthetic inputs. For example, a developer particularly concerned about GAN-based attacks could consider omitting Dolos if diffusion-generated images are uncommon. In such scenarios, a smaller ensemble or even a well-targeted individual model could be a more practical solution for the problem.

From a developer point of view, it would be strategically beneficial to prioritize the ensemble performance to target against the distribution of attack types they observe [47]. This could be done by weighting the overall classes of images to reflect this distribution of observed attacks. One of the key strengths of the blind ensemble arises in the ability to dynamically adjust to changes in the distribution of attacks. Models can be reconfigured, replaced, or extended without expensive retraining of the final decision model. This allows the developers to easily modify the detection system to match the range inputs they experience.

Overfitting Risks and Mitigation in Ensemble Design

One concern around ensemble construction is overfitting to one component model at the expense of performance on a class of inputs. This would likely occur if there is a large number of input subspaces or if the distribution, as mentioned above, heavily weights towards one of the classes. In those cases, it is likely possible that the ensemble's decision model could ignore the component models that are selected to target the underrepresented classes of inputs. This could be addressed by better tuning parameters of the decision model, either the number of trees or the minimum samples for a split in order to balance the weighting of component models.

In general, there is no ideal configuration for all use cases. The configuration of the decision model and selection of component models should be tuned to how the system will be applied and the type of fake data encountered. However, in all cases, careful control over the parameters chosen and sampling of data during training is required to ensure that the ensemble is well balanced and successfully generalized over the input space.

Chapter 5

Attempt at Informed Ensemble

As has been mentioned in earlier chapters, a blind ensemble model cannot dynamically adjust weights to address the characteristics of the input image. This inability to dynamically adjust the weighting of the component models sometimes results in the ensemble model reducing performance when a component model flips predictions for positive outputs. This behavior can be seen in the performance of the Dolos model on some of the GAN and VAE Datasets, as seen for FaceForensics in Figure **??**. These observations promote the introduction of input image information into the ensemble model.

The trial application of this technique yields overall results of far poorer ensemble models that do not classify better than their components, and in many cases are worse than the results from a random classifier. The reasoning behind the poor performance is explained further in this section. While the attempts at creating an informed ensemble method are unsuccessful, the core intuition behind building such a model is still sound. As will be described, the limitation of this approach lies in the accuracy of the adjudicator models that provide insight into the input image. This chapter serves to provide reasoning behind poor informed ensemble performance. The following sections cover two methods of informing the ensemble model: direct dynamic weighting and informed decision models.

5.1 Ensemble Architecture

Two informed ensemble strategies are designed to attempt to improve the performance of a blind ensemble method in deepfake image detection: one based on direct vote adjustment and another based on introducing input information to the decision model. Both approaches begin with the same base structure in which multiple classifiers are either imported off the shelf (COTS) or trained independently on deepfake datasets corresponding to different generation methods, as done with the blind ensemble models.

Both of the informed ensemble strategies use the same model architecture, with different decision models depending on the strategy. They both build on traditional ensemble classification by introducing a learned *adjudicator*, which learns to assess the reliability of each sub-model prediction on a per-image basis. Conceptually, the adjudicator sits one level above the ensemble: it does not produce its own real/fake classification, but rather serves as a dynamic controller to provide information on how much to trust the output of each sub-model for a given input.

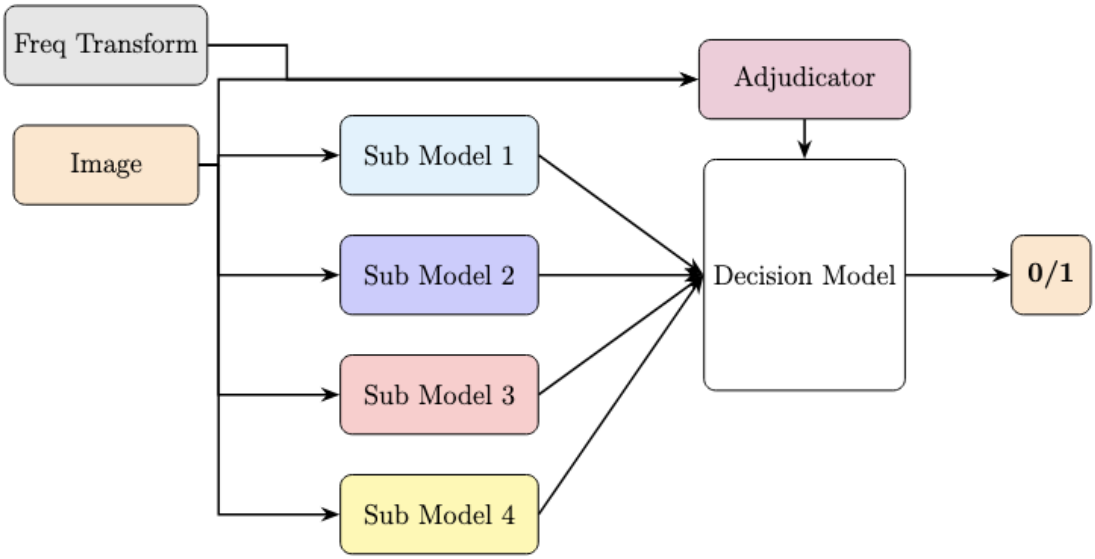


Figure 5.1: Architecture of ensemble model with adjudicator-guided decision model.

This adjudicator can be effectively conceptualized as a model that looks internally to

understand which generation technique produced the image and then learn which classifiers can be trusted on which types of images. For example, if an image contains artifacts commonly seen in StarGAN-based images, the adjudicator may learn to down-weight models trained on other generation methods. This enables the ensemble to respond more intelligently to image-specific generation cues, rather than relying on hardcoded voting rules.

The adjudicator receives the same input image as the base classifiers, and optionally a set of transformed versions such as DCT, FFT, or other frequency and compression domain representations. These transformations are included to expose artifacts that may not be obvious in pixel space but are often indicative of specific generation techniques [6,22].

The adjudicator's structure is typically a compact convolutional neural network (CNN), chosen for its ability to extract local and global features from visual input. It may resemble a reduced version of a deepfake detector, but its output is not a single prediction; instead, it produces a vector of confidence scores $\mathbf{w}(x) = [w_1(x), w_2(x), \ldots, w_n(x)]$, one for each sub-model. To produce $\mathbf{w}(x)$, the adjudicator is trained with a supervision target as a binary vector of correctness indicators, defined as:

$$\mathbf{w}_{target}(x) = [\{M_1(x) = y(x)\}, \ldots, \{M_n(x) = y(x)\}] \tag{5.1}$$

The adjudicator learns to map the input image (and any auxiliary transforms) to this output, effectively modeling which classifiers tend to succeed on a given deepfake image. The remainder of this section describes the two strategies in which the adjudicator's outputs can be used to inform the ensemble's final prediction.

5.1.1 Dynamic Weighting

When passed an input image, the adjudicator outputs confidence scores $w_i(x)$, which are used to smooth the output predictions from the component models. Each individual prediction is softened based on confidence:

$$\hat{M}_i(x) = w_i(x) \cdot M_i(x) + (1 - w_i(x)) \cdot (1 - M_i(x)) \tag{5.2}$$

This expression adjusts the prediction when the adjudicator expresses uncertainty and preserves model predictions when confidence is high. The ensemble's final output is then the normalized sum of these adjusted predictions:

$$E(x) = \frac{1}{n} \sum_{i=1}^{n} \hat{M}_i(x) \tag{5.3}$$

This method accounts for cases where some classifiers may systematically misclassify specific types of images. For instance, this formulation helps address situations where models like Dolos may flip their predictions for unfamiliar manipulations. If the Dolos model outputs a true prediction for an image from a GAN dataset, the shifted weight of the prediction would allow the prediction from Dolos to be shunted to 0. This would allow the model to retain the high recall of other component models without being impacted by weak Dolos predictions.

5.1.2 Informed Random Forest

The alternative strategy incorporates the adjudicator's output into the same random forest decision model from the blind ensemble structure. Here, the adjudicator is trained using the same expected accuracy supervision as in the previous strategy. At inference, its confidence vector $\mathbf{w}(x) = [w_1(x), w_2(x), \ldots, w_n(x)]$ is concatenated with the base model prediction vector $\mathbf{M}(x) = [M_1(x), M_2(x), \ldots, M_n(x)]$, to form $\mathbf{z}(x) = \text{concat}(\mathbf{M}(x), \mathbf{w}(x))$, and the resulting combined input is passed into the ensemble decision model. The random forest decision model is expanded to double the number of trees to reflect the increase in input vector size.

This strategy allows the decision model to learn complex interactions between vote patterns and their associated confidences, potentially capturing nuanced decision logic that is not easily expressed through the dynamic weighting strategy. It brings the flexibility of

training the ensemble model while still preserving the structure of the adjudicator module used in the first strategy.

5.2 Adjudicator Training

The adjudicator module is kept the same for both of the strategies for informed ensembles. The input provided to the module is the RGB Image as well as the same 2D discrete cosine transform used for the DCT model [6]. Thus, for some image I, the input into the adjudicator, I' can be defined as such:

$$T[u,v] = \alpha_u \alpha_v \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I[i,j] \cos\left[\frac{\pi}{M}\left(i+\frac{1}{2}\right)u\right] \cos\left[\frac{\pi}{N}\left(j+\frac{1}{2}\right)v\right] \tag{5.4}$$

$$\alpha_u = \begin{cases} \sqrt{\frac{1}{M}}, & \text{if } u = 0 \\ \sqrt{\frac{2}{M}}, & \text{if } 1 \leq u \leq M-1 \end{cases} \tag{5.5}$$

$$\alpha_v = \begin{cases} \sqrt{\frac{1}{N}}, & \text{if } v = 0 \\ \sqrt{\frac{2}{N}}, & \text{if } 1 \leq v \leq N-1 \end{cases} \tag{5.6}$$

$$I' = \text{concat}(T, I) \tag{5.7}$$

The architecture of the adjudicator model is a CNN trained with outputs as a vector of component model correctness over the balanced input space described in Section 4.1. This decision was made after observing that CNN structures can be used to identify synthetic image generation sources in prior research [48–50]. This style of adjudicator would also remain lightweight while still being able to identify spatial and frequency-domain artifacts. The adjudicator model consists of two convolutional blocks, each containing a Conv2D layer (3x3 kernel, 32 and 64 filters respectively) followed by a 2x2 Max Pool, and one 256-unit fully connected layer with dropout (p = 0.25). The final output layer has a sigmoid activation,

producing one score per component model. This score represents the percent confidence in the model's accuracy. The adjudicator is trained on 80% of the available data using binary correctness labels for each of the four component classifiers: MesoNet, DCT(0.1), DCT(0.5), and Dolos. A label of 1 indicated that the corresponding model correctly classified the input image, and 0 otherwise. The adjudicator is trained using the Adam optimizer with a learning rate of 0.001, and binary cross-entropy loss is chosen across all four outputs. As with the blind ensemble model, the remaining 20% of the data is reserved for testing, with 5-fold cross-validation.

5.3 Evaluation of Informed Ensemble

To evaluate the performance of the informed ensemble strategies, an ensemble of MesoNet & DCT(0.1) & DCT(0.5) & Dolos is constructed and the trained adjudicator is attached to the system. This produces three ensemble models to be compared, which are named in the following sections as:

1. Informed Dynamic Weighting [IDW]
2. Informed Random Forest [IRF]
3. Blind Ensemble [BE]

The same partitions for training and test data are used to train the decision models in all three of the ensembles, as described in Section 4.1 for BE and Section 5.2 for IDW and IRF. The remaining 20% of data reserved for testing is used to evaluate the models, and the same 5 cross-fold validation sets are used to ensure the models are tested on the same sub-partitions of data.

As hinted in the preamble of this chapter, the informed ensemble methods do not perform well at all. As can be seen in Table 5.1, neither of the informed ensemble methods is stronger than the blind ensemble model. While neither showed positive results, IRF is closer to

Table 5.1: Performance comparison of different ensemble models.

Ensemble Model Combination	Accuracy	Precision	Recall	AUC
IDW	47.1 ± 8.1	45.3 ± 3.6	42.7 ± 2.0	44.0 ± 1.7
IRF	67.5 ± 2.2	74.0 ± 1.9	58.0 ± 1.7	72.5 ± 1.2
BE	**79.2 ± 1.5**	**83.4 ± 2.1**	**73.0 ± 2.0**	**87.4 ± 0.9**

the performance of the blind ensemble model. This does make sense, as the IRF ensemble preserves more of the structure of the blind ensemble model. It is likely that the IRF decision model learns to devalue the outputs of the adjudicator model where possible and leans more heavily on classifying an output in a similar manner to the BE model itself. On the other hand, the IDW model, which simply weights the outputs from the component models, experiences performance worse than a random classifier.

At first glance, this result is quite surprising. The core intuition behind the IDW model should support an improvement in the performance of the ensemble system. However, this ignores one major assumption around the accuracy of the adjudicator module. If the adjudicator module is not accurate at predicting the *trustworthiness* of the individual models, then the weighting it produces will skew the results of the ensemble to be less accurate than the learned performance from a blind ensemble method. To further explore this potential issue, the accuracy of the adjudicator's predictions is evaluated for each pair of dataset and model.

Table 5.2: Accuracy of adjudicator at predicting the correctness of component models.

Dataset	MesoNet	DCT(0.1)	DCT(0.5)	Dolos
DeepFakeFace	34.1	22.6	25.0	47.3
FaceForensics	45.8	20.9	22.3	23.2
Individual	23.1	62.4	63.3	26.8
ProGAN	25.6	58.7	59.2	29.3
StarGAN	26.0	70.3	69.1	25.5
WhichFace	32.7	63.5	62.4	24.3

As can be seen in Table 5.2, the adjudicator is not very successful at predicting the correctness of the individual models. For most pairs of models and datasets, the accuracy

is under 30%, explaining the reduction in effectiveness of the IDW ensemble strategy. The adjudicator can somewhat successfully predict the performance of the DCT models on the GAN datasets, though this could be skewed by the DCT models' generally high performance throughout the data. As the DCT model performs incredibly well across the GAN and faceswap datasets, the adjudicator likely tends to predict the DCT model as correct for most input images to minimize loss, resulting in around 70% correct predictions over those successful datasets.

These results explain the reduction in performance for both of the informed models. If the adjudicator provides low-quality information, then the output of the combined system is likely to be low quality as well.

5.4 Discussion on Informed Ensembles

The low performance of IDW and IRF does not mean that informed ensembles are not potentially viable strategies to explore, rather that a more nuanced adjudicator system would need to be designed. There are some inherent limitations with the method chosen to inform the ensemble, which result from difficulty in identifying artifacts that can be attributed to certain generation methods.

Part of the intuition behind the adjudicator is that it can identify what generation method was used on the image and inform which models to trust as a result. The central issue with this approach is that in order to identify the artifacts that each component models, the adjudicator would need to replicate aspects of the discriminators of the component models. This poses a challenge in the design of the adjudicator itself: how can one model architecture pull out the artifacts that multiple components look to identify?

Another challenge with this approach is in handling real images. Based on how the component models are trained, they will be able to identify different subsets of the real image set. We see this when observing the slight loss in performance on true negative rates from

Section 4.8. It is difficult for the adjudicator to identify how to rate the correctness of the component models in cases where there are no artifacts present. It is also possible for the adjudicator to learn patterns among the real images based on poor model performance that don't reflect the nature of the images. It may begin to then flip predictions for real images based on inaccurate predictions on real images as well.

As seen in Table 5.2, many of the adjudicators' accuracies hover around 20-30%. This seems especially low at first, but makes sense when considering what the adjudicator is trying to predict. The adjudicator is not evaluated on its accuracy in identifying the class of the image, but rather on trying to predict if the component model would accurately label that image in that class of image. For classes of samples where the component model is already poor, at around 50-60%, the adjudicator likely won't be able to learn how to differentiate between images within the class it identifies. Even if the adjudicator has a 75% accuracy at figuring out that the image contains artifacts corresponding to some class, it still must predict if the image lies in the 50% of that class that the component model correctly labels. In this case, we would expect a 75% accurate adjudicator to only have a 37% accuracy at also identifying if the component model would label the image as correct.

5.4.1 Future Work for Informed Ensembles

Future work on adjudicator architecture in informed ensemble models can lead to more insightful ensemble systems. If the adjudicator can become highly accurate at predicting the class of the image, the decision model could function as a switch that leans only on the predictions from the best decision model of that class.

Another future avenue for work is to dig into the component models themselves and identify ways to bisect the model. By finding layers of the components that target the presence of artifacts and exposing those vectors, the decision model could use that presence layer to decide whether to toggle on or off the component.

Inclusion of Output Embedding

A more promising avenue to inform the ensemble model would be to eliminate the adjudicator altogether. Rather than aggregating discrete votes, this strategy could combine learned embeddings from each classifier into a unified feature space and train a downstream classifier to make the final decision. This strategy has been explored for the purpose of developing GAN-specific deepfake detection [51,52]. This could allow the system to capture more nuanced, latent patterns of the input that each component model identifies. Each component model in the ensemble could be modified to expose an intermediate embedding vector, typically from the last layer of the network. For each image, the embeddings from all n classifiers are extracted and concatenated into a single feature vector:

$$\mathbf{f}(x) = \text{concat}(\mathbf{e}_1(x), \mathbf{e}_2(x), \ldots, \mathbf{e}_n(x)) \tag{5.8}$$

This joint vector $\mathbf{f}(\mathbf{x})$ would then serve as an input to the final decision model, which would train to classify the image as real or fake [51]. This model would face some challenges, especially surrounding the size of the input vector to the decision model. This could be mitigated through preprocessing steps to compress the model embeddings into the most relevant nodes.

One downside of the embedding-based strategy would be an increased difficulty in scaling or adapting the ensemble when adding new models. Since the final classifier is trained on a fixed-size concatenation of embeddings, adding a new model changes the input structure and requires retraining the full decision layer [53]. This makes the system less modular—each time a model is added, the ensemble has to be rebuilt. Nevertheless, it still serves as a strong potential improvement to the informed ensemble model strategies.

Chapter 6

Conclusion

This thesis presents ensemble-based approaches to improve the generalization of deepfake detection models for particular use in identity verification. The motivation for this ensemble approach arises from observations of poor cross-dataset performance of individual deepfake detection models, which is especially present when facing generation methods outside the scope of the model's training data. We select a collection of images from six facial deepfake datasets, covering a diverse range of generation techniques: GANs, VAEs, and diffusion models. In this study, four detection models are sourced: CNN-F, MesoNet, DCT-based, and Dolos. These detectors are combined into an exhaustive collection of ensemble models to test efficacy and identify important guidelines for the construction of ensemble models. We observe that blind ensemble strategies statistically significantly improve system performance as compared to their component models across the range of image classes. We especially note that ensembles reduce false negatives and improve generalization across different methods of synthetic image generation. Based on the results of this study, we suggest constructing ensembles based on recall or accuracy and leaning towards models with less overlapping strong performance between image classes.

This thesis also explores building *informed* ensemble strategies with the inclusion of an adjudicator module. While promising in theory, the specific approaches in this work do not

yield successful results. Future avenues for work in informed ensembles are identified and can further improve how combination models tackle generalization issues.

The ensemble strategies tested in this thesis do provide a strong baseline for improving model generalization in deepfake detection. Based on the results of this thesis, it can be said that modular ensemble methods offer a scalable path forward for improving the robustness of real-world biometric systems.

References

[1] A. Naitali, M. Ridouani, F. Salahdine, and N. Kaabouch. "Deepfake Attacks: Generation, Detection, Datasets, Challenges, and Research Directions." *Computers*, **12**(10), 2023. ISSN: 2073-431X. DOI: 10.3390/computers12100216. URL: https://www.mdpi.com/2073-431X/12/10/216.

[2] A. Diel, T. Lalgi, I. C. Schröter, K. F. MacDorman, M. Teufel, and A. Bäuerle. "Human performance in detecting deepfakes: A systematic review and meta-analysis of 56 papers." *Computers in Human Behavior Reports*, **16**(), 2024, p. 100538. ISSN: 2451-9588. DOI: 10.1016/j.chbr.2024.100538. URL: https://www.sciencedirect.com/science/article/pii/S2451958824001714.

[3] P. Korshunov and S. Marcel. "Vulnerability assessment and detection of deepfake videos." In: *2019 International Conference on Biometrics (ICB)*. IEEE. 2019, pp. 1–6. DOI: 10.1109/ICB45273.2019.8987375.

[4] Onfido. *Identity Fraud Report 2023*. Accessed: 2025-07-23. Onfido, 2023. URL: https://onfido.com/resources/reports/identity-fraud-report-2023.

[5] A. Jain, A. Ross, and K. Nandakumar. *Introduction to Biometrics*. SpringerLink : Bücher. Springer US, 2011. ISBN: 9780387773261. DOI: 10.1007/978-0-387-77326-1. URL: https://link.springer.com/book/10.1007/978-0-387-77326-1.

[6] J. Ricker, S. Damm, T. Holz, and A. Fischer. "Towards the detection of diffusion model deepfakes." *arXiv preprint arXiv:2210.14571*, (), 2022. DOI: 10.48550/arXiv.2210.14571.

[7] H. Song, S. Huang, Y. Dong, and W.-W. Tu. *Robustness and Generalizability of Deepfake Detection: A Study with Diffusion Models*. 2023. arXiv: 2309.02218 [cs.CV].

[8] D. P. Kingma, M. Welling, et al. "An introduction to variational autoencoders." *Foundations and Trends® in Machine Learning*, **12**(4), 2019, pp. 307–392. DOI: 10.1561/2200000056.

[9] A. Dehghani and H. Saberi. "Generating and Detecting Various Types of Fake Image and Audio Content: A Review of Modern Deep Learning Technologies and Tools." *arXiv preprint arXiv:2501.06227*, (), 2025. DOI: 10.48550/arXiv.2501.06227.

[10] K. Remya Revi, K. R. Vidya, and M. Wilscy. "Detection of Deepfake Images Created Using Generative Adversarial Networks: A Review." In: *Second International Conference on Networks and Advances in Computational Technologies*. Ed. by M. Palesi, L. Trajkovic, J. Jayakumari, and J. Jose. Springer International Publishing, 2021, pp. 25–35. ISBN: 978-3-030-49500-8. DOI: 10.1007/978-3-030-49500-8_3.

[11] T. Karras, T. Aila, S. Laine, and J. Lehtinen. "Progressive Growing of GANs for Improved Quality, Stability, and Variation." In: *International Conference on Learning Representations (ICLR)*. Accessed: 2025-07-23. 2018. URL: https://arxiv.org/abs/1710.10196.

[12] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[13] Preeti, M. Kumar, and H. K. Sharma. "A GAN-Based Model of Deepfake Detection in Social Media." *Procedia Computer Science*, **218**(), 2023. International Conference on Machine Learning and Data Engineering, pp. 2153–2162. ISSN: 1877-0509. DOI: 10.1016/j.procs.2023.01.191. URL: https://www.sciencedirect.com/science/article/pii/S1877050923001916.

[14] T. Karras, S. Laine, and T. Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks." In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[15] D. Yadav and S. Salmani. "Deepfake: A Survey on Facial Forgery Technique Using Generative Adversarial Network." In: *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. 2019, pp. 852–857. DOI: 10.1109/ICCS45141.2019.9065881.

[16] H. Khalid and S. S. Woo. "Oc-fakedect: Classifying deepfakes using one-class variational autoencoder." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, pp. 656–657. DOI: 10.1109/CVPRW50498.2020.00336.

[17] J. Ho, A. Jain, and P. Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems*, **33**(), 2020, pp. 6840–6851. DOI: 10.5555/3495724.3496298.

[18] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. "Deep unsupervised learning using nonequilibrium thermodynamics." In: *International conference on machine learning*. PMLR. 2015, pp. 2256–2265. DOI: 10.48550/arXiv.1503.03585.

[19] J. Song, C. Meng, and S. Ermon. "Denoising diffusion implicit models." *arXiv preprint arXiv:2010.02502*, (), 2020. DOI: 10.48550/arXiv.2010.02502.

[20] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. "Score-based generative modeling through stochastic differential equations." *arXiv preprint arXiv:2011.13456*, (), 2020. DOI: 10.48550/arXiv.2011.13456.

[21] B. Li, J. Sun, and C. M. Poskitt. "How generalizable are deepfake detectors? An empirical study." *arXiv preprint arXiv:2308.04177*, (), 2023. DOI: 10.48550/arXiv.2308.04177.

[22] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros. "CNN-generated images are surprisingly easy to spot... for now." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8695–8704. DOI: 10.1109/CVPR42600.2020.00872.

[23] Z. Liu, X. Qi, and P. H. Torr. "Global texture enhancement for fake face detection in the wild." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 8060–8069. DOI: 10.1109/CVPR42600.2020.00808.

[24] B. Ghita, I. Kuzminykh, A. Usama, T. Bakhshi, and J. Marchang. "Deepfake Image Detection Using Vision Transformer Models." In: *2024 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. 2024, pp. 332–335. DOI: 10.1109/BlackSeaCom61746.2024.10646310.

[25] L. Chai, D. Bau, S.-N. Lim, and P. Isola. "What makes fake images detectable? understanding properties that generalize." In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*. Springer. 2020, pp. 103–120. DOI: 10.1007/978-3-030-58574-7_7.

[26] D.-C. Țânțaru, E. Oneață, and D. Oneață. "Weakly-supervised deepfake localization in diffusion-generated images." In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 6258–6268. DOI: 10.48550/arXiv.2311.04584.

[27] J. Asan, I. Ekaputri, C. Natalie, and K. Purwandari. "Exploring Generative Adversarial Networks (GANs) for Deepfake Detection: A Systematic Literature Review." In: *2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP)*. 2023, pp. 189–194. DOI: 10.1109/IWAIIP58158.2023.10462832.

[28] L. Hansen and P. Salamon. "Neural network ensembles." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(10), 1990, pp. 993–1001. DOI: 10.1109/34.58871.

[29] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan. "Recurrent Convolutional Strategies for Face Manipulation Detection in Videos." In: *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2019, pp. 2181–2189. URL: https://arxiv.org/abs/1905.00582.

[30] D. H. Wolpert. "Stacked generalization." *Neural Networks*, **5**(2), 1992, pp. 241–259. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-6080(05)80023-1. URL: https://www.sciencedirect.com/science/article/pii/S0893608005800231.

[31] D. Salvi, H. Liu, S. Mandelli, P. Bestagini, W. Zhou, W. Zhang, and S. Tubaro. "A Robust Approach to Multimodal Deepfake Detection." *Journal of Imaging*, **9**(6), 2023. ISSN: 2313-433X. DOI: 10.3390/jimaging9060122. URL: https://www.mdpi.com/2313-433X/9/6/122.

[32] O. Oriola. "A stacked generalization ensemble approach for improved intrusion detection." *International Journal of Computer Science and Information Security (IJCSIS)*, **18**(5), 2020.

[33] M. Groh, R. Chang, D. L. Rosen, and H. Farid. *Which Face is Real?* https://www.whichfaceisreal.com/. Accessed: 2025-07-23. 2022.

[34] H. Song, S. Huang, Y. Dong, and W.-W. Tu. "Robustness and generalizability of deepfake detection: A study with diffusion models." *arXiv preprint arXiv:2309.02218*, (), 2023. DOI: 10.48550/arXiv.2309.02218.

[35] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. "FaceForensics++: Learning to Detect Manipulated Facial Images." In: *International Conference on Computer Vision (ICCV)*. 2019.

[36] M. Rahman. *Individualized Deepfake Detection Dataset*. 2024. DOI: 10.21227/w7ma-fp34.

[37] Y. Li and S. Lyu. "Exposing DeepFake Videos By Detecting Face Warping Artifacts." In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pp. 46–52. URL: https://arxiv.org/abs/1811.00656.

[38] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. "Face2Face: Real-time Face Capture and Reenactment of RGB Videos." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2387–2395. DOI: 10.1109/CVPR.2016.262. URL: https://niessnerlab.org/projects/thies2016face.html.

[39] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. "Deferred Neural Rendering: Image Synthesis Using Neural Textures." In: *ACM Transactions on Graphics (TOG)*. Vol. 38. 4. 2019, pp. 1–12. DOI: 10.1145/3306346.3323020. URL: https://niessnerlab.org/projects/thies2019neural.html.

[40] H. Yu. *FaceSwap-GAN: A Modern Deepfakes Pipeline*. https://github.com/shaoanlu/faceswap-GAN. Accessed: 2025-07-23. 2019.

[41] C. C. Bingyu Chen and W. H. Hsu. "Cross-Age Reference Coding for Age-Invariant Face Recognition and Retrieval." In: *European Conference on Computer Vision (ECCV)*. Accessed: 2025-07-23. Springer, 2014, pp. 768–783. URL: https://bcsiriuschen.github.io/CACD/.

[42] H. Khalid, S. Tariq, M. Kim, and S. S. Woo. "FakeAVCeleb: A novel audio-video multimodal deepfake dataset." *arXiv preprint arXiv:2108.05080*, (), 2021. DOI: 10.48550/arXiv.2108.05080.

[43] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. "Mesonet: a compact facial video forgery detection network." In: *2018 IEEE international workshop on information forensics and security (WIFS)*. IEEE. 2018, pp. 1–7. DOI: 10.1109/WIFS.2018.8630761.

[44] W. J. Youden. "Index for rating diagnostic tests." *Cancer*, **3**(1), 1950, pp. 32–35. DOI: 10.1002/1097-0142(1950)3:1<32::AID-CNCR2820030106>3.0.CO;2-3.

[45] R. Palacios, A. Gupta, and P. S. Wang. "Feedback-based architecture for reading courtesy amounts on checks." *Journal of Electronic Imaging*, **12**(1), 2003, pp. 194–202. DOI: 10.1117/1.1526105.

[46] L. I. Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014.

[47] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. "Adaptive Mixtures of Local Experts." *Neural Computation*, **3**(1), 1991, pp. 79–87. DOI: 10.1162/neco.1991.3.1.79.

[48] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros. "CNN-Generated Images Are Surprisingly Easy to Spot... for Now." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Accessed: 2025-07-23. 2020, pp. 8695–8704. URL: https://openaccess.thecvf.com/content_CVPR_2020/html/Wang_CNN-Generated_Images_Are_Surprisingly_Easy_To_Spot..._For_Now_CVPR_2020_paper.html.

[49] N. Yu, L. Davis, and M. Fritz. "Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints." In: *IEEE International Conference on Computer Vision (ICCV)*. Accessed: 2025-07-23. 2019, pp. 7556–7566. URL: https://openaccess.thecvf.com/content_ICCV_2019/html/Yu_Attributing_Fake_Images_to_GANs_Learning_and_Analyzing_GAN_Fingerprints_ICCV_2019_paper.html.

[50] P. Zhang, A. Nair, and V. M. Patel. "Detecting GAN-Synthesized Faces Using Landmark Locations." *Signal Processing: Image Communication*, **89**(), 2020. Accessed: 2025-07-23, p. 106918. DOI: 10.1016/j.image.2020.106918. URL: https://arxiv.org/abs/1906.00035.

[51] F. L. Hung Dang and X. Liu. "On the Detection of Digital Face Manipulation." *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, **44**(11), 2022. Used high-level CNN embeddings fed to a decision-level classifier, pp. 8075–8089. DOI: 10.1109/TPAMI.2021.3054827. URL: https://arxiv.org/abs/2002.11645.

[52] I. Masi, Y. Wu, T. Hassner, and P. Natarajan. "Two-Stream Networks for Deepfake Detection." In: *IEEE CVPR Workshops (CVPRW)*. One stream captures deep CNN embeddings, the other focuses on residuals; outputs are fused via a final classifier. 2020. URL: https://openaccess.thecvf.com/content_CVPRW_2020/html/w31/Masi_Two-Stream_Networks_for_Deepfake_Detection_CVPRW_2020_paper.html.

[53] L. Verdoliva, D. Cozzolino, and G. Poggi. "A Feature-Based Ensemble for GAN Source Attribution." In: *IEEE International Workshop on Information Forensics and Security (WIFS)*. Embeddings from multiple CNNs are concatenated and fed into an ensemble classifier for source attribution. 2020. URL: https://arxiv.org/abs/2002.01198.

[54] H. Khalid, M. Kim, S. Tariq, and S. S. Woo. "Evaluation of an Audio-Video Multimodal Deepfake Dataset using Unimodal and Multimodal Detectors." In: *Proceedings of the 1st Workshop on Synthetic Multimedia - Audiovisual Deepfake Generation and Detection*. ADGD '21. Virtual Event, China: Association for Computing Machinery, 2021, pp. 7–15. ISBN: 9781450386821. DOI: 10.1145/3476099.3484315. URL: https://doi.org/10.1145/3476099.3484315.

[55] Y. Mirsky and W. Lee. "The creation and detection of deepfakes: A survey." *ACM computing surveys (CSUR)*, **54**(1), 2021, pp. 1–41. DOI: 10.1145/3425780.

[56] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia. "Deepfakes and beyond: A survey of face manipulation and fake detection." *Information Fusion*, **64**(), 2020, pp. 131–148. DOI: 10.1016/j.inffus.2020.06.014.

[57] G. Pei, J. Zhang, M. Hu, Z. Zhang, C. Wang, Y. Wu, G. Zhai, J. Yang, C. Shen, and D. Tao. "Deepfake generation and detection: A benchmark and survey." *arXiv preprint arXiv:2403.17881*, (), 2024. DOI: 10.48550/arXiv.2403.17881.

[58] CSA Top Threats Working Group. *Top Threats to Cloud Computing 2024*. Tech. rep. Cloud Security Alliance, 2024. URL: https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-2024.

[59] J. M. Latorre, S. Cerisola, A. Ramos, and R. Palacios. "Analysis of stochastic problem decomposition algorithms in computational grids." *Annals of Operations Research*, **166**(), 2009, pp. 355–373. DOI: 10.1007/s10479-008-0476-1.

[60] Y. Wang, X. Chen, X. Ma, S. Zhou, Z. Huang, Y. Wang, C. Yang, Y. He, J. Yu, P. Yang, et al. "Lavie: High-quality video generation with cascaded latent diffusion models."

International Journal of Computer Vision, (), 2024, pp. 1–20. DOI: 10.1007/s11263-024-02295-1.

[61] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu. "Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation." *Advances in Neural Information Processing Systems*, **36**(), 2024. DOI: 10.5555/3666122.3666490.

[62] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. "High-resolution image synthesis with latent diffusion models." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695. DOI: 10.1109/CVPR52688.2022.01042.

[63] A. Sauer, D. Lorenz, A. Blattmann, and R. Rombach. "Adversarial diffusion distillation." In: *European Conference on Computer Vision*. Springer. 2024, pp. 87–103. DOI: 10.1007/978-3-031-73016-0_6.

[64] Y. Nirkin, L. Wolf, Y. Keller, and T. Hassner. "Deepfake detection based on the discrepancy between the face and its context." *arXiv preprint arXiv:2008.12262*, (), 2020. DOI: 10.48550/arXiv.2008.12262.

[65] Z. Huang, J. Hu, X. Li, Y. He, X. Zhao, B. Peng, B. Wu, X. Huang, and G. Cheng. "SIDA: Social Media Image Deepfake Detection, Localization and Explanation with Large Multimodal Model." *arXiv preprint arXiv:2412.04292*, (), 2024. DOI: 10.48550/arXiv.2412.04292.

[66] C. Koutlis and S. Papadopoulos. "DiMoDif: Discourse Modality-information Differentiation for Audio-visual Deepfake Detection and Localization." *arXiv preprint arXiv:2411.10193*, (), 2024. DOI: 10.48550/arXiv.2411.10193.

[67] Y. Zhang, C. Miao, M. Luo, J. Li, W. Deng, W. Yao, Z. Li, B. Hu, W. Feng, T. Gong, et al. "MFMS: Learning Modality-Fused and Modality-Specific Features for Deepfake Detection and Localization Tasks." In: *Proceedings of the 32nd ACM International Conference on Multimedia*. 2024, pp. 11365–11369. DOI: 10.1145/3664647.3688984.

[68] J. Dong, J. Chen, X. Xie, J. Lai, and H. Chen. "Survey on Adversarial Attack and Defense for Medical Image Analysis: Methods and Challenges." *ACM Computing Surveys*, **57**(3), 2024, pp. 1–38.

[69] K. D. Apostolidis and G. A. Papakostas. "A survey on adversarial deep learning robustness in medical image analysis." *Electronics*, **10**(17), 2021, p. 2132.

[70] J. C. Costa, T. Roxo, H. Proença, and P. R. M. Inácio. "How Deep Learning Sees the World: A Survey on Adversarial Attacks Defenses." *IEEE Access*, **12**(), 2024, pp. 61113–61136. DOI: 10.1109/ACCESS.2024.3395118.

[71] J. Jheelan and S. Pudaruth. "Using Deep Learning to Identify Deepfakes Created Using Generative Adversarial Networks." *Computers*, **14**(2), 2025. ISSN: 2073-431X. DOI: 10.3390/computers14020060. URL: https://www.mdpi.com/2073-431X/14/2/60.