DEEPFAKE GENERATION & DETECTION

Ahn Yebin, Kim Jeonghyun, Jung Jaehee
2024-CSE4007-12143 Artificial Intelligence
Department of Data Science
Hanyang University
Seoul, Korea

ABSTRACT

With the activation of generative models, the issue of Deepfake has been continuously raised. Our attention has been directed to both the generative models that create Deepfakes and the detection models that identify them. Initially, we noted problems with generative models due to the characteristic of "human faces"—realistic human facial images are challenging to generate with simple prompts and a limited variety of images produced. Due to resource constraints on modifying the model itself, we employed captioning and the gram matrix to develop a diverse and realistic set of images to some extent. We selected the ViT(Vision Transformer) model for the detection models, which demonstrates high performance through patch-based attention. To meet the given conditions (abundant data, Colab runnable), we applied the model lightening technique, LoRA, to reduce the trainable parameters. Additionally, we chose the Single center loss(SCL), a loss function specialized for Deepfake detection, to improve the generalizability, which was a core issue in existing detection models.

1 Introduction

As generative AI becomes more common, many people easily create new creations. Generative AI is producing results that surpass humans in fields where it was expected that it would not be able to replace humans, such as images, video, and music. While many people can easily create new art as quickly as they want, it also brings big problems. For example, deepfakes are a very ethical issue and have recently become a hot topic. This is because it can be a means of threatening someone with false information or distorted pornography, or it can cause problems such as fraud. In particular, false information can lead to DeepFaith and significantly impact society as a whole. If we detect these 'fakes,' we can prevent some confusion. However, the real problem is that it is now easier for humans to distinguish between this generated information. New-generation technologies such as GAN, Diffusion, LDM, and ADM are becoming more realistic and sophisticated, making them difficult to recognize as deepfakes. Therefore, creating models that can detect generative creations and deepfakes that can no longer be distinguished by humans is a problem that must be addressed not only academically but also for the social good.

To this end, we present a simple idea to make generative models more realistic and generate different images. Additionally, we also worked on a project to improve our understanding of deepfakes and design detection pipelines for deepfake detection.

2 Related Works

2.1 Generation

2.1.1 Latent Diffusion Model, Stable Diffusion (Rombach et al. 2022)

Stable diffusion is a high-resolution image synthesis technique based on latent diffusion models (LDMs). In the autoencoder of LDM, the original image is taken as input and compressed into latent vectors. Moreover, the autoencoder is learned by combining perceptual loss and adversarial goals to ensure high-quality reconstruction without relying

on aggressive spatial compression. Model learns the diffusion process using the time-conditional U-Net architecture and the latent vector, which generates high-resolution images by repeatedly removing noisy latent vectors. Moreover, the cross-attention mechanism allows the user to put in various types of desired input conditions, and the model can produce high-quality conditional images accordingly.

2.1.2 DeepFace (Taigman et al. 2014)

This system aims to improve face recognition performance close to the human level. Generally, face recognition follows four steps: 'detect →, align →, represent →, classify.' In this study, the alignment and representation steps were improved. 3D face modeling was introduced to align the face precisely, and facial expressions were created using a neural network with a 9-layer depth. It uses 3D model-based alignment to create robust facial representations across various angles and twists. They also use convolution and max pooling layers to extract simple edges and textures in the initial stage. Afterward, local statistics of different facial areas are learned through locally connected layers. The top layer is a fully connected layer, which captures the overall features of the face. The ReLU activation function, Dropout in the first fully connected layer, Cross-Entropy loss function, and SGD were also used. This model was trained on a large dataset containing 40 million images and over 4,000 identities. Such a large dataset made face recognition possible in various environments and showed results approaching human-level performance in datasets such as Labeled Faces in the Wild (LFW).

2.1.3 Vision Transformer (Dosovitskiy et al. 2021)

Since the advent of Transformer, transformer has been the mainstream in the NLP (Natural Language Processing) field. However, CNN, not in the field of computer vision, was still widely used, and transformers were not used for computer vision. Therefore, this study showed that image classification can be performed without CNN by cutting the image to make it a sequence and using it as an input of a transformer. Because the image is too large when cut in pixels, it is divided into patches of a specific size, and the order of these patches enters the transformer as input. Since this is similar to how NLP uses word sequences as input, text is transformed into a vector like the existing transformer, added to the positional embedding, and goes through the transformer architecture.

2.1.4 GPT-2 (Radford et al., n.d.)

It is usually supervised learning in NLP. However, supervised learning lacks generalization performance, such as being sensitive to even slight changes in the data distribution. In previous GPT-1, we used the Transformer's Decoder part and some fine-tuning, leaving room for SOTA and zero-shot. In this study, using WebText, a new dataset consisting of millions of web pages, we wanted to create a model that can perform downstream tasks with zero shots using only unsupervised pre-training without fine-tuning. Moreover, in this study, Byte Pair Encoding (BPE), which is a trade-off between word level and byte level, was used as the input. Transformer-based models are also compared to traditional GPT models, and additional layer normalization was introduced after the final self-attention block.

2.1.5 Gram Matrix (Li et al. 2017)

In this paper, we explain why the gram matrix exhibits style. To this end, the concept of domain adaptation problem emerges, which refers to a kind of transfer learning aimed at applying source and learning models to unlabeled target domains. To summarize, we address the neural style transfer (NST) principle to solve the domain adaptation problem. They consider matching of gram matrices, which is the same as minimizing the maximum mean discrepancy (MMD), which measures the difference in distribution between two sets of samples between feature maps. This shows that the characteristics of the NST equalize the feature distribution between the style image and the generated image. The experiments conducted in this study tested various distribution alignment methods obtained using VGG-19, and new style transfer results were obtained using a combination with various methods such as linear, polynomial, Gaussian, and batch normalization statistical matching.

2.2 Detection

2.2.1 Fundamental(Rössler et al. 2018; "DeepfakeBench," n.d.; Yan, Zhang, Yuan, et al. 2023)

Faceforensic++ detected forgeries using xception image features with a CNN backbone and provides datasets which are face swap, deepfake, Face2face, and Neural Textures. They are generally used for pre-training deep face detection.

And DeepfakeBench is the first paper to stand up the benchmarks of deep face detection. Needless to say, the content of this paper is excellent and clean. The deep fake detection model cannot be overlooked in that it compares performance under the code implementing most papers, pre-trained weights, and thorough control variables.

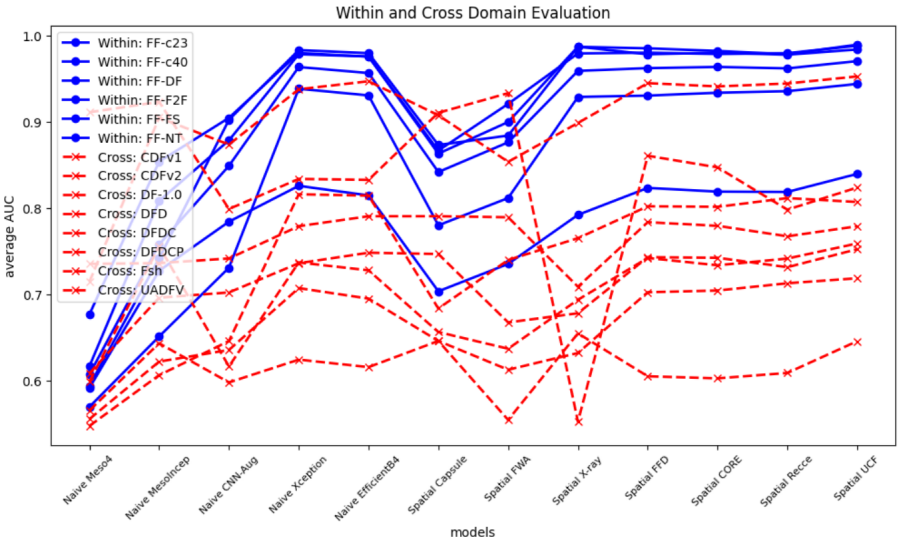

Figure 1: Within and cross domain evaluation

We will introduce the insights gained when this model was compared with famous models in this field, including the Sota model, under the same conditions. First, when compression or blur augmentation was performed in cross-domain evaluation, a significant performance decrease was observed in the dfdcp dataset provided by group 28. This phenomenon can be observed mainly when using the xception backbone. Second, when comparing xception, EffcientNet, and ResNet, primarily used as backbone networks, a significant performance decrease occurred in ResNet. The difference between the two is depthwise separable convolution and model scale. In particular, since xception showed a significant performance improvement in dfdcp, xception was considered as the backbone as a priority.

2.2.2 Recent issue(Yan, Zhang, Yuan, et al. 2023)

A recent issue is that detection models overfit the dataset. This can be seen by comparing within-domain evaluation and cross-domain evaluation. Within-domain evaluation refers to the case where the training and testing datasets are the same, and cross-domain evaluation relates to cases where the datasets are not the same. In other words, even if the performance is good within domain evaluation, it can be considered overfitting to the complete train data set if the performance is terrible in cross-domain evaluation. The graph shown in [Figure 1] shows that the x-axis represents various detection model types, and the y-axis represents the AUC score: blue means within-domain evaluation, and red means cross-domain evaluation. As you can see, there is blue on top of red. This means that overfitting is a critical problem that occurs overall.

2.2.3 Solve issue

There are several ways to encounter this issue. There were two paths when I read several of the latest papers from 2023 and 2024. The first is to use LLM(Chang et al. 2023; Jia et al. 2024; Keita et al. 2024; Santosh et al. 2024), which has excellent performance, and the second is to design an algorithm that follows the existing traditional pipeline but can be generalized(Yan, Zhang, Fan, et al. 2023; Yan et al. 2024; Ni et al. 2022; Haliassos et al. 2022). We decided that the latter would be more advantageous for developing new ideas. So, I will introduce the relevant previous studies in more detail.

Disentanglement learning You can better understand and manipulate your data by learning representations that separate the various explanatory elements. At this time, each disentangled dimension or subset of dimensions corresponds to a specific feature in the data. According to Uncovering Common Features for Generalizable Deepfake(UCF)(Yan, Zhang, Fan, et al. 2023), prior research applying the existing disentangle methodology extracted only content and forgery for features extracted through the CNN backbone. However, at this time, it shows poor cross-domain evaluation because it relies on the method-specific pattern of forgery used in the training dataset. Therefore, we proposed a multi-task disentanglement framework to find specific and standard forgery features. It performed pretty well when compared to other models in Deepfakebench(Yan, Zhang, Yuan, et al. 2023). Preserving Fairness Generalization in Deepfake Detection(Lin et al. 2024; Ju et al. 2023) is a method that borrows UCF's methodology(Yan, Zhang, Fan,

et al. 2023) to extract additional demographic features and adds fair learning with fairness loss. Although there is no significant difference methodologically, the significance of this paper lies in its argument that fairness should be pursued. Although it is less hot than the deepfake detection generalization issue, issues regarding the fairness of detection have been consistently raised. For example, according to "An examination of fairness of AI models for deepfake detection" by L. Trinh and Y. Liu(Trinh and Liu 2021), meaningful error differences were found between subgroups in deepfake datasets and detection models. As deepfake detection started from an ethical issue, we thought fairness was an issue that runs through the topic, and we proceeded with the project considering this.

Latent space augmentation(Yan et al. 2024) To generalize forgery-specific and overfitted features, train in the direction of widening the margin at the decision boundary that divides the forgery and the actual image. Of course, it is possible with entanglement learning, but the pipeline divided into three branches - reconstruction, multiclass cls, and binary cls - is quite long. So, the method chosen in this paper is to leave the real image features to a model with excellent performance; only the forgotten image features go through the CNN backbone and then augment them with various handcrafted methods to learn unseen features by a much lighter student model. What is most emphasized is that it contributed to building a more general decision boundary by augmenting the latent vector rather than performing augmentation before entering the backbone.

Lightening the model through distillation inspired us greatly. Training a model in Colab is a task that only very light models can handle. Perhaps that's why I recently researched more lightweight models. As a result, I ended up using a detection model that applied LoRA.

2.3 PEFT

Parameter-Efficient Fine-Tuning(PEFT) is an optimization function library provided by HuggingFace. In large-scale language models, fixing most parameters while adjusting only a few parameters is required.This approach deals with the weaknesses (high training and storage costs) of conventional fine-tuning methods. By reducing the costs, PEFT helps to avoid the "catastrophic forgetting" problem that can occur during full fine-tuning.These techniques have been proven to improve generalizability, especially in data-limited conditions. It is also greatly adaptable across various domains, especially in computer vision fields.

2.3.1 LoRA

Low-Rank Adaptation (LoRA) significantly reduces the number of trainable parameters for downstream tasks by freezing the weights of a pre-trained model and injecting trainable rank decomposition matrices into each layer of the transformer architecture. LoRA offers the advantage of parameter efficiency and, importantly, allows the merging of trainable matrices with the original weights without additional inference latency when deployed. Unlike existing methods, LoRA also provides the flexibility and can be combined with various fine-tuning approaches, enhancing its versatility and scalability.(Hu et al. 2021)

3 Proposed methods

3.1 Generation

Since Stable Diffusion is a model trained with a large amount of data, a large GPU, and a long time, we decided that it would not be easy to change the model itself in a situation where the project had to be carried out with a laptop and Colab. Therefore, we decided to create an image with the goal of "creating an image similar to the style of the original image with a pre-trained model."

3.1.1 Captioning

Initially, we planned to create an image by giving a simple prompt for text to image stable diffusion. However, the simple prompt did not look like a real person like the image in the train data. Additionally, by using the same prompt, we were able to confirm that images with a similar feel were created (figure 3).

Therefore, we tried to make image data unique for validation by using it as a prompt after performing a capturing operation to extract the unique characteristics of each image from training data. First, we detected human faces and extracted information on emotion, gender, race, and age through a DeepFace pre-trained model("DeepFace," n.d.). With this extracted information, a prompt was created: 'The {gender} is {age} years old {race} and {emotion}.' This prompt and initial image was given to text-to-image stable diffusion to create the image("Stable Diffusion Pipelines," n.d.). At

Figure 2: The example of train real data. People wear glasses, hats, sunglasses, etc. in various ways.

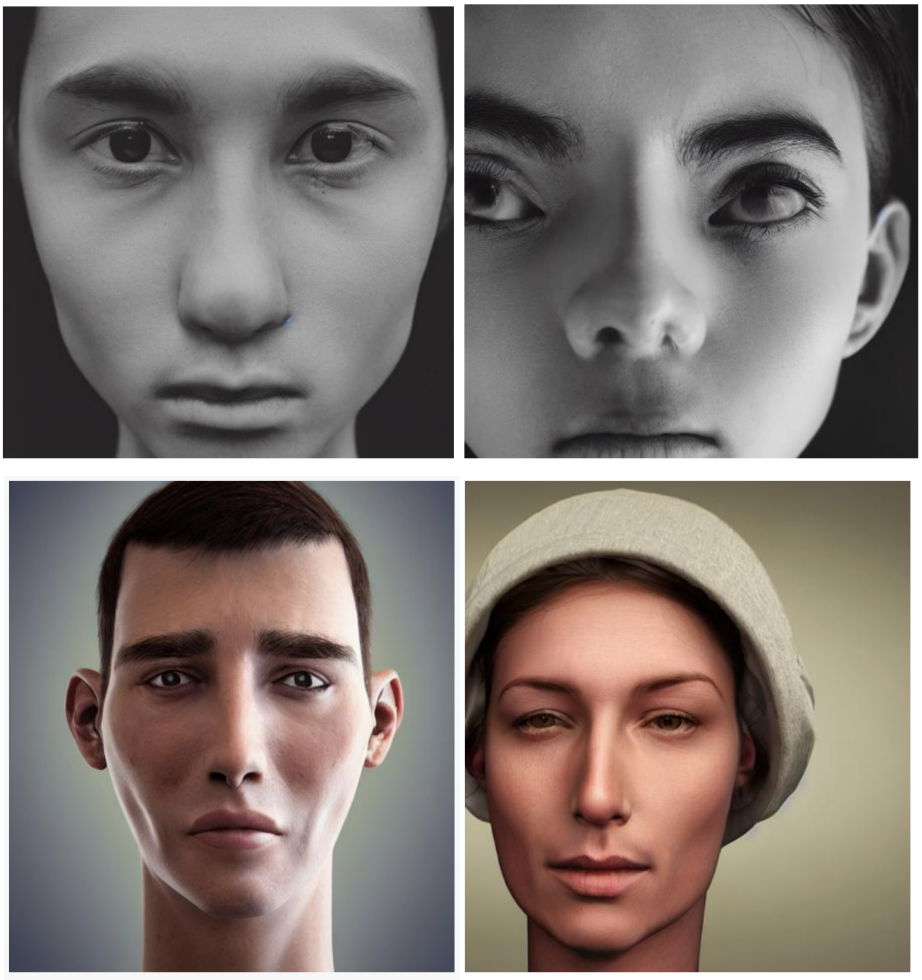

Figure 3: Upper two images were generated with the prompt of ‘A photo of human face’ and Lower two images were generated with the prompt of ‘A realistic photo of human face’.

Figure 4: 'The woman is 21 years old black and neutral' was given to prompt for text-to-image stable diffusion. The output generated is much more realistic compared to Figure 3.

first, we tried to use image-to-image stable diffusion. Still, the quality of the generated image changed dramatically depending on the seed, making stable generation impossible, so we decided to use text-to-image stable diffusion.

However, it was still challenging to create various images like train data. Therefore, we decided to utilize ViT additionally to create a richer prompt("Vision Transformer (ViT)," n.d.). While DeepFace was only able to extract information about facial expressions, age, gender, and race, ViT was also able to extract the tools the person had. DeepFace is based on CNN. Thus, local features are well extracted by stacking multiple layers to create a feature map. However, ViT is based on the Transformer. Therefore, using a self-attention mechanism, widespread features can be learned well, and global features can be well captured. Therefore, by combining these two captioning models with different approaches, they could complement each other and create detailed captions. Combining these two captions, 'A face image of {caption from ViT}. The final prompt was 'And the {gender} is {age} years old {race} and {emotion}.'

3.1.2 Gram Matrix

Our goal was to "Create an image with a similar style to the original image," but just as important as having a similar style was creating an image that was realistic enough to fool the detection model. As a result of applying the style transfer as is, the image appeared to have more noise than when style transfer was not performed. Accordingly, it was decided not to perform style transfer, but to use only the gram matrix on which it is based.

The Gram matrix is a dot product matrix of feature maps. This is an indicator of how similar the feature maps of a given layer are to each other. The feature map represents various features extracted from the input image, and each layer includes features at different levels. Assuming that feature map F has N filters, activation at M positions, i and j represent filter index, and k represents space position, Gram Matrix G can be expressed as

$$Gij = kFikFjk$$

Insert the train real image and the image created with the above prompt into pre-trained VGG-19 in PyTorch, and extract the feature map. Using the feature map extracted in this way, the gram matrix of each image is calculated. In the NST introduced earlier, the distance was calculated using L2 loss, but because we wanted to determine whether the overall feeling was similar with less influence from outliers, we calculated the distance using L1 loss.

Since we will not be using this distance for style transfer, no optimization process has been performed. However, using the distance and threshold obtained in this way, if distance > threshold, 'This face image should be similar to the initial image' was added to the prompt so that the image was created again.

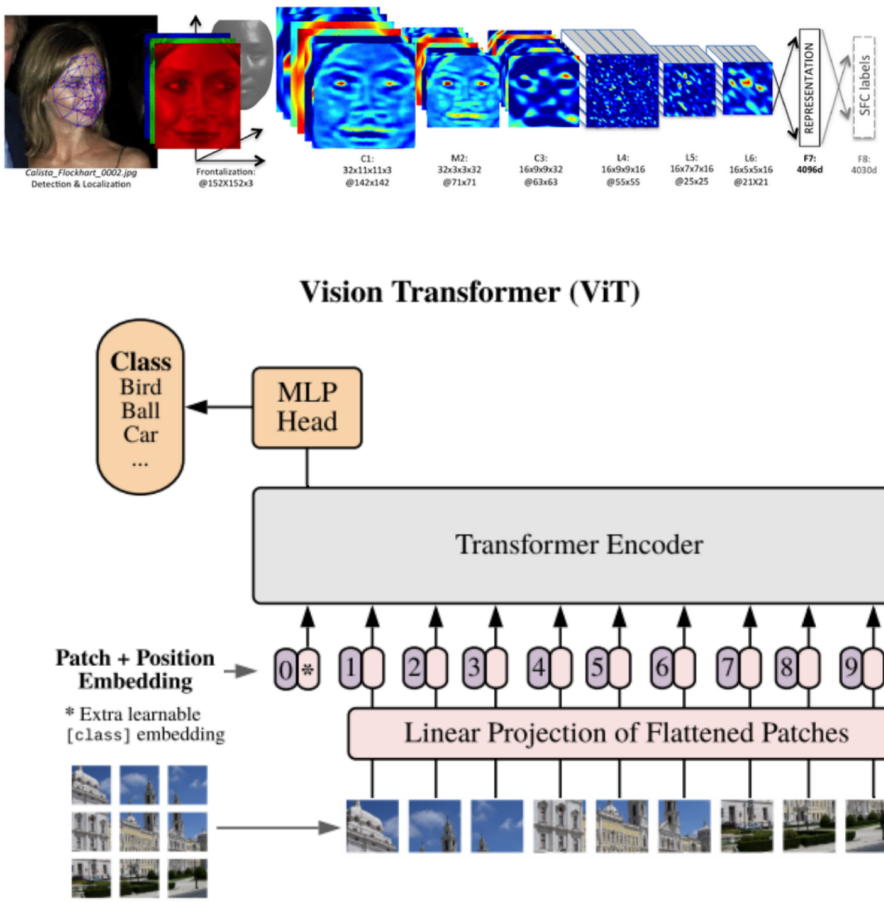

Figure 5: DeepFace(Upper) & ViT(Lower)
DeepFace is a CNN based model (Sun and Redei, 2022). The DeepFace model uses a nine-layer deep neural network to learn approximately 120 million parameters and 40,000 features. And ViT is a Transformer based model. The most significant difference is only that it uses image patches as input.

3.2 Detection

3.2.1 Considering Fairness

Scheme To incorporate fairness, we first followed the pipeline of Lin et al., "Preserving Fairness Generalization in Deepfake Detection(Lin et al. 2024)," which shows good performance and is one of the most recent and classic deep fake detection models.

Data Collection First, we would like to clarify that there was no room for selection in data because the program had to proceed under the assumption that the conditions provided by group 28 must be followed.

As a training set, they provided a Kaggle link that determines whether it is accurate/fake, and the images are provided as CSV. I quote the explanation provided in the link.

"This dataset consists of all 70k REAL faces from the Flickr dataset collected by Nvidia, as well as 70k fake faces sampled from the 1 Million FAKE faces (generated by StyleGAN) that Bojan provided. I conveniently combined both datasets in this dataset, resized all the images into 256px, and split the data into train, validation, and test sets. I also included some CSV files for convenience.

For more details, check out the threads:

Thread for real faces dataset: https://www.kaggle.com/c/deepfake-detection-challenge/discussion/122786

$$L_{cls} = C(\widetilde{h}(f_i^g), Y_i) + \ \rho_2 M(\widehat{h}(d_i), D_i)$$

$$L_{con} = [b + \|f_{\text{anchor}} - f_+\|_2 - \|f_{\text{anchor}} - f_-\|_2]_+$$

$$\mathcal{L}_{dis} = \frac{1}{n}\sum_i [L_{cls} + \rho_3 L_{con} + \rho_4 L_{rec}],$$

Figure 6: Classification, Contrastive, Distillation Loss

$$\mathcal{L}_{fair} = \min_{\eta \in \mathbb{R}} \eta + \frac{1}{\alpha|\mathcal{J}|}\sum_{j=1}^{|\mathcal{J}|}[L_j - \eta]_+, \tag{2a}$$

$$\text{s.t. } L_j = \min_{\eta_j \in \mathbb{R}} \eta_j + \frac{1}{\alpha'|\mathcal{J}_j|}\sum_{i:D_i=\mathcal{J}_j}[C(h(I_i), Y_i) - \eta_j]_+. \tag{2b}$$

Figure 7: Bi-Level Fairness Loss

1 Million Fake Faces: https://www.kaggle.com/c/deepfake-detection-challenge/discussion/121173"

We want to say through this quote that the dataset corresponds to dfdcp styleGAN, so the train was done based on dfdcp(Dolhansky et al. 2020). Afterward, the train proceeded through the 500 sets given later.

Model pipeline First, we used Xception as the backbone. The feature vector extracted this way is subjected to disentanglement learning with four different features. At this time, extracting context features and forgery standard features was helped by referring to "Uncovering common features for generalizable deepfake detection(Yan, Zhang, Fan, et al. 2023)" a previous study that boasted excellent performance. There is no feature-specific entanglement because the given dataset only contains stylGAN fake data sets. Each makes predictions through the head (Figure 6).

The added part uses fair prediction through AdaINFusion for fair learning. If you approach the model naively, there may be a way to give a fairness penalty. When this method was applied in previous research, it only increased a specific fairness score and degraded model performance, and it is questionable whether actual fairness was guaranteed. Forcing prediction ratios to different population combinations can lead to significant overfitting.

Therefore, we propose a bi-level fairness loss, which consists of two levels, as the name suggests. First, you must understand the fairness risk measure presented by Williamson & Menon(Williamson and Menon 2019) and the large-scale methods for distributionally robust optimization presented by Levy et al(Levy et al. 2020). To briefly explain, the former introduces an objective function that minimizes CVaR and uses it to learn a fair classifier, and the latter performs robust optimization on convex loss using conditional value at risk (CVaR) and chi-square divergence uncertainty set. Bi-level fairness, proposed by combining the two, provides direction to ensure that predictions proceed fairly while avoiding overfitting (Figure 7).

Model Training We followed the prior initial setting with some changes to our differences. Batch size 8, epochs 10, use SGD optimizer with learning rate $\beta = 5 \times 10{-}4$. We ran it in the given setting to decide how to initialize parameters, but 2 hours away, the first epoch didn't end. So, we accepted reality and reduced the data size, batch size, and epoch count. Also, there was only StyleGAN forgery in the given train dataset, but I thought the generalization implemented module would overcome it.

Input of trains are the model, optimization program, learning rate scheduler, total number of epochs, and start epochs and to repeat the learning and evaluation, go through loop epoch. After calculating the loss, we perform backpropagation via the optimizer to update the model's weight. Uniquely, we created an efficient way to go into evaluation mode after one loop and proceed with optimization faster. Then pass through the sensing pipeline and require other functions. Then, we must ensure the GPU is available, set the test dataset path, and define the CustomImageFolder class to load the images and labels. We define the data transformation and load the dataset. Once you define the model and import the model weights from the checkpoint file, you can insert the trained weights. Unfortunately, however, the results were very bad. Binary classifier had an ROC of 0.5, which could not be exceeded.

Even after a long run by buying a 70,000 won premium colab for a GPU, it showed no signs of improving. AUC = 0.152, TPR = 0.38, FPR = 0.93, ACC = 0.23 is our score when we trained on the largest setting. The dataset contains

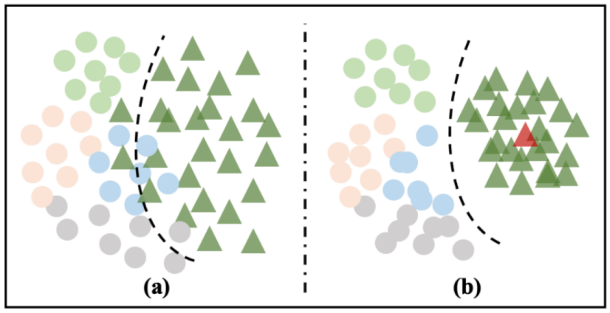

Figure 8: Illustration of feature distribution (a). without single-center loss; (b). with single-center loss.

only styleGAN. Meeting different datasets that do not result in overfitting would have shown SOTA performance by exploiting the learning characteristics of these models to exploit latent vector augmentation in latent space.

Failure analysis However, we went in the opposite direction and overfitted styleGAN, but we have no choice if this will work well. So the model structure selects forgery-specific features for disentanglement, and the method is to select the corresponding forgery among various generation methods or to learn from multi-class cls classification loss to select the actual deepfake.

Therefore, the conclusion is that disentanglement was an unsuitable learning strategy, given the conditions that forced styleGAN to be used as a forgery in the first place. Through this, we realized, felt, and reflected on many things.

3.2.2 Vit-Single Center Loss

Single Center Loss Fig. 6 illustrates the of feature distribution w/o and w/ SCL, where circles with different colors indicate different manipulation methods while triangles represent real samples. SCL is designed to make the feature distribution of real faces more compact and, at the same time, move fake features away from the center of real features (red triangle in Fig. 6). Eqn.

(1) shows the single-center loss function:

$$LSCL = dreal + max(drealdfake + margin, 0)$$

where d is the average distance between the real center and each feature, as shown in Eqn. (2):

$$d = 1Ni = 1N = fiC^2$$

where C represents the real center. We pick the features after the second last fully-connected layer to calculate LSCL. By using such loss, the features of real and fake faces become more discriminative and separable, thus leading to a more general face forgery detection performance.(Kong, Li, and Wang 2023)

Model training To build models efficiently, the transfer learning method was adopted. In the PyTorch-based image classification model library, 'timm', vision transformer(Vit) with an image size of 224x224 and a patch size of 16x16 was selected as a pre-trained model. LoRA was applied to the weights of query and key, in each attention layer so as to reduce the trainable parameter. This corresponds to the 'qkv' layer in the structure of our pre-trained model. Lora attention dimension (the rank) was set to 16, the scaling value (lora_alpha) was set to 32, and the dropout probability (lora_dropout) for Lora layers was set to 0.1. Through this, it was possible to reduce the number of trainable parameters from the total of 87157480 to 589824. This guarantees the performance of the model with 68% parameter adjustment. Adam was used for optimization with a learning rate of 0.001. Due to the issue of the provided dataset being too large to upload, the training data was randomly sampled to 3,000 entries and the validation data to 300 entries, and the model was trained over 10 epochs. And SCL(Single Center Loss) is used as a loss function.

4 Results

4.1 Generation

A caption was automatically created from the initial image. By creating a prompt with this caption and using the gram matrix, we were able to create much more diverse and realistic human images compared to the initial work (figure 7).

As a result, it was possible to overcome to some extent the limitations of stable diffusion, which limits diversity. In addition, it was discovered in the early work that stable diffusion resulted in character-like images that did not look like real people, but this problem was also solved through this process.

What can be seen here is that the captions generated with ViT were relatively accurate, while the captions generated with DeepFace were inaccurate. As 3rd and 4th figures show, DeepFace's caption generates 'Man' even if the real is 'Woman'. However, in most cases, it was possible to create images similar to the original based on the preceding words (ViT-based generation), and I would like to analyze this phenomenon in the future.

4.2 Detection

4.2.1 Vit-LoRA

We conducted a total of two training sessions, one with the SCL (Single-Center Loss) and one without it(naive one), and compared their loss values and accuracy. Additionally, we performed validation on the images we generated by diffusion model using the model trained with SCL to check its generalizability.

Naive Version Inference Accuracy: Correct predictions: 420 out of 500 Accuracy: 84.00%

Accuracy from generated images(DM): (Generated) Validation Loss: 0.7447 (Generated) Validation Accuracy: 75.00%

With SCL

Inference Accuracy:
Correct predictions: 421 out of 500
(SCL) Accuracy: 84.20%

Accuracy from generated images(DM):
(Generated / SCL) Validation Loss: 0.6208
(Generated / SCL) Validation Accuracy: 88.00%

Compare the results When trained with a given dataset, the accuracy was 84.00% and 84.20%, respectively, making little difference. However, when we validated the image generated by the diffusion model earlier, the accuracy was greatly different. Although the Naive model was only 75%, the w/SCL model showed a 4% higher performance compared to the given dataset trained by random sampling with an accuracy of 88%.

5 Discussions

We created complex captions by taking advantage of the fact that CNN and ViT emphasize different parts of the same image. By using this caption as a prompt and at the same time utilizing the distance of the Gram matrix, we overcome the diversity problem and the problem of creating completely different images in Stable Diffusion.

However, we found two problems with this process. First, as mentioned in the Results part, CNN-based DeepFace generated relatively inaccurate captions compared to ViT-based ones. However, what was surprising was that even with the contradictory captions (ViT: woman, DeepFace: man), the image was created well in line with the initial image. As a result, we became interested in the contradictory prompt and initial image process, and thought we would like to address this in future research.

Additionally, our project is capable of generating diverse and unbiased images only when an initial image is given. Therefore, in order to create a completely diverse and bias-free generative model, the problem of stable diffusion itself must be solved. Since Stable Diffusion is currently unable to completely perform mode separation in the latent space, future challenges may include clearly separating different modes in the latent space and learning to adjust each mode to exhibit different characteristics.

The results for the ViT-based detection model were quite startling. All the train, validation, and test data provided to us consisted of real images and fake images generated from 'GAN'. Noting the issue of conventional models overfitting the dataset, we applied SCL as the loss function to prevent this and improve generalizability. When applied to a different dataset generated from 'DM', the result of w/SCL was 13% higher. This demonstrates the benefits of using a suitable loss function for deepfake detection, enhancing its generalizability clearly.

Figure 9: The four examples above show how the caption below is created from the real train image on the left and how the image is created when the caption is used as a prompt for stable diffusion.

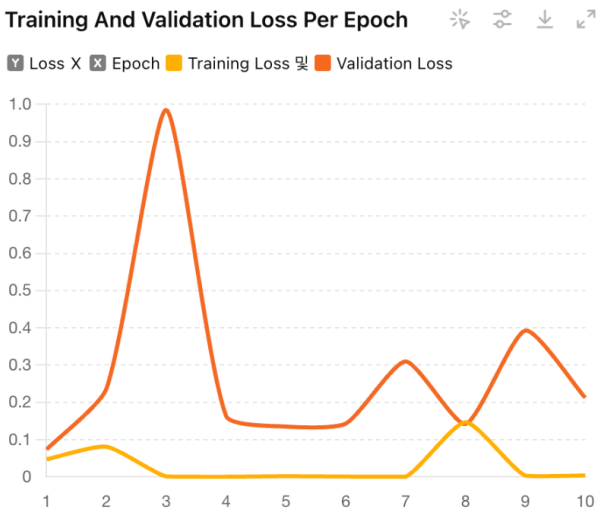

Figure 10: Training and validation loss per epoch for naive model

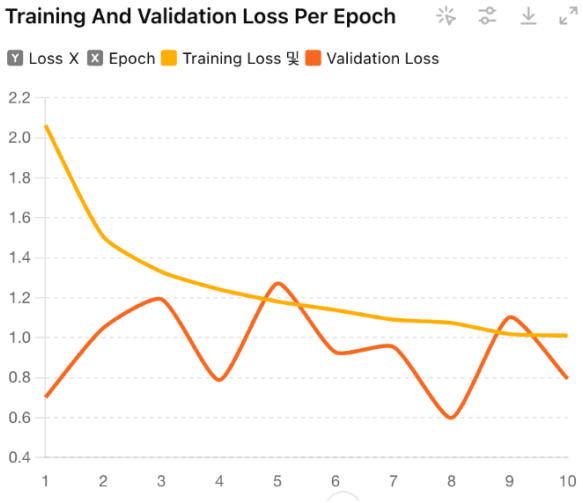

Figure 11: Training and validation loss per epoch for SCL applied model

Significantly, a new attempt was made that broke away from the existing paradigm of the forgetting detector by using scl loss as the backbone of it. In particular, for a person who uses Colab, the runtime cannot be guaranteed. When processing a large amount of data is complex, an efficient method must be chosen above all else. For this purpose, Lora was applied to overcome the situation. In the future, I suggest creating a new loss function that is fast and efficient with similar resources in less time.

Forgery detection has been performed using CNN-based feature extraction. A lot of research is underway to detect using LLM, such as anti fake prompt and robust clip-based detector, but the backbone is often used as a CNN architecture in the primary pipeline. However, according to the Deepfakebench paper, traditional CNN-based backbones such as Xception and EfficientNet are still mainly used. We reveal that we took on a new challenge in using a CNN-based backbone with vit and scl loss, which was not commonly used before.

If feature extraction is performed naively CNN encoder, the forgery common feature for the image and the forgery forgery-specifice will appear. In addition, there are anthropometric features and context features to particular patterns. Going further, there may be features that the model could not extract but play a critical role, such as punctum. In the future, we can extract the punctum by considering whether it is possible to extract such unpredictable but existing features.

References

\begin{thebibliography} Chang, You-Ming, Chen Yeh, Wei-Chen Chiu, and Ning Yu. 2023. "AntifakePrompt: Prompt-Tuned Vision-Language Models Are Fake Image Detectors." arXiv. http://arxiv.org/abs/2310.17419.

"DeepFace." n.d. Python. https://github.com/serengil/deepface/tree/master.

"DeepfakeBench." n.d. https://github.com/SCLBD/DeepfakeBench.

Dolhansky, Brian, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. 2020. "The DeepFake Detection Challenge (DFDC) Dataset." arXiv. http://arxiv.org/abs/2006.07397.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, et al. 2021. "An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale." arXiv. http://arxiv.org/abs/2010.11929.

Haliassos, Alexandros, Rodrigo Mira, Stavros Petridis, and Maja Pantic. 2022. "Leveraging Real Talking Faces via Self-Supervision for Robust Forgery Detection." arXiv. http://arxiv.org/abs/2201.07131.

"Hugging Face:PEFT." n.d. https://huggingface.co/docs/peft/index.

Jia, Shan, Reilin Lyu, Kangran Zhao, Yize Chen, Zhiyuan Yan, Yan Ju, Chuanbo Hu, Xin Li, Baoyuan Wu, and Siwei Lyu. 2024. "Can ChatGPT Detect DeepFakes? A Study of Using Multimodal Large Language Models for Media Forensics." arXiv. http://arxiv.org/abs/2403.14077.

Ju, Yan, Shu Hu, Shan Jia, George H. Chen, and Siwei Lyu. 2023. "Improving Fairness in Deepfake Detection." arXiv. http://arxiv.org/abs/2306.16635.

Keita, Mamadou, Wassim Hamidouche, Hessen Bougueffa Eutamene, Abdenour Hadid, and Abdelmalik Taleb-Ahmed. 2024. "Bi-LORA: A Vision-Language Approach for Synthetic Image Detection." arXiv. http://arxiv.org/abs/2404.01959.

Kong, Chenqi, Haoliang Li, and Shiqi Wang. 2023. "Enhancing General Face Forgery Detection via Vision Transformer with Low-Rank Adaptation." arXiv. http://arxiv.org/abs/2303.00917.

Levy, Daniel, Yair Carmon, John C. Duchi, and Aaron Sidford. 2020. "Large-Scale Methods for Distributionally Robust Optimization." arXiv. http://arxiv.org/abs/2010.05893.

Li, Yanghao, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. 2017. "Demystifying Neural Style Transfer." arXiv. http://arxiv.org/abs/1701.01036.

Lin, Li, Xinan He, Yan Ju, Xin Wang, Feng Ding, and Shu Hu. 2024. "Preserving Fairness Generalization in Deepfake Detection." arXiv. http://arxiv.org/abs/2402.17229.

Ni, Yunsheng, Depu Meng, Changqian Yu, Chengbin Quan, Dongchun Ren, and Youjian Zhao. 2022. "CORE: Consistent Representation Learning for Face Forgery Detection." arXiv. http://arxiv.org/abs/2206.02749.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. n.d. "Language Models Are Unsupervised Multitask Learners."

Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. "High-Resolution Image Synthesis with Latent Diffusion Models." arXiv. http://arxiv.org/abs/2112.10752.

Rössler, Andreas, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. 2018. "FaceForensics: A Large-Scale Video Dataset for Forgery Detection in Human Faces." arXiv. http://arxiv.org/abs/1803.09179.

Santosh, Li Lin, Irene Amerini, Xin Wang, and Shu Hu. 2024. "Robust CLIP-Based Detector for Exposing Diffusion Model-Generated Images." arXiv. http://arxiv.org/abs/2404.12908.

"Stable Diffusion Pipelines." n.d. Python. Hugging Face. https://huggingface.co/docs/diffusers/api/pipelines/stable_diffusion/overview.

Sun, Qiyu, and Alexander Redei. 2022. "Knock Knock, Who's There: Facial Recognition Using CNN-Based Classifiers." *International Journal of Advanced Computer Science and Applications* 13 (1). https://doi.org/10.14569/IJACSA.2022.0130102.

Taigman, Yaniv, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. 2014. "DeepFace: Closing the Gap to Human-Level Performance in Face Verification." In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1701–8. Columbus, OH, USA: IEEE. https://doi.org/10.1109/CVPR.2014.220.

Trinh, Loc, and Yan Liu. 2021. "An Examination of Fairness of AI Models for Deepfake Detection." arXiv. http://arxiv.org/abs/2105.00558.

"Vision Transformer (ViT)." n.d. Python. Hugging Face. https://huggingface.co/docs/transformers/model_doc/vit.

Williamson, Robert C., and Aditya Krishna Menon. 2019. "Fairness Risk Measures." arXiv. http://arxiv.org/abs/1901.08665.

Yan, Zhiyuan, Yuhao Luo, Siwei Lyu, Qingshan Liu, and Baoyuan Wu. 2024. "Transcending Forgery Specificity with Latent Space Augmentation for Generalizable Deepfake Detection." arXiv. http://arxiv.org/abs/2311.11278.

Yan, Zhiyuan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. 2023. "UCF: Uncovering Common Features for Generalizable Deepfake Detection." In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 22355–66. Paris, France: IEEE. https://doi.org/10.1109/ICCV51070.2023.02048.

Yan, Zhiyuan, Yong Zhang, Xinhang Yuan, Siwei Lyu, and Baoyuan Wu. 2023. "DeepfakeBench: A Comprehensive Benchmark of Deepfake Detection." arXiv. http://arxiv.org/abs/2307.01426.

\end{thebibliography}