Fine-tuning a Vision Transformer (Swin-Tiny) for Detection and Classification of AI-generated Images

Alejandro Castellano <ac2578>, Kim Sha <ks966>, Abhinav Girish <ag2242> , Vikranth Kanumuru <vk326>

Abstract

The goals of this research focus primarily on fine-tuning a pre-trained vision transformer (Swin-Tiny) to extend a binary classification problem: identifying whether an image is created by generative AI. That baseline objective mirrors a current Hugging Face competition: Aiornot. The project will expand the scope of this baseline into a multiclass classification problem: identifying whether an image is authentic (human-generated) or generated by one of a series of text-to-image AI generators (i.e., Stable Diffusion, Midjourney, and DALL-E).

1. Introduction

The wide-spread availability of text-to-image generators such as Midjourney, Stable Diffusion and DALL-E raises a host of ethical concerns that makes it increasingly important to differentiate between AI-generated and authentic content. For artists and photographers, these tools represent threats against their livelihood, particularly if there is no attribution afforded to them for images created in their style. Furthermore, the ease and speed of these tools make them accessible to bad actors that may use them to propagate disinformation through visual content in social networks and news settings. As a result, it is becoming increasingly imperative to develop models capable of discriminating between authentic or artificial images generated by one of a predefined set of text-to-image generators. The end users would include, but are not limited to: teachers and educators who want to detect artistic plagiarism, social media platforms that want to moderate or ban AI-generated media used in the spread of disinformation, and art auction houses and collectors that must guarantee the authenticity of human-generated art their dealings.

1.1. Related Work

Prior research in this area has focused on images generated by previous state of the art generative models (in particular, GANs). One of the primary findings of this research is that detection methods used to identify AI-generated images produced by GANs (augmentation by gaussian blurring; learning fourier domain features) are not reliable because of issues related to image compression, resizing, and low-resolutions. These are all problems that are frequently encountered in the real world (i.e. all over the web). Furthermore, models only trained on GANs work poorly on images generated by Diffusion Models that are widely proliferating today.

This is why this project focuses on tackling this problem using the same underlying models that power current products such as Stable Diffusion and DALL-E: transformers.

Vision Transformer (ViT) is the first transformer encoder successfully trained on ImageNet to attain results that rival state-of-the-art convolutional neural networks (CNNs) while requiring comparatively fewer computation resources. Transformer models have superseded the widespread use of recurrent neural networks (RNNs) in sequential model applications too. While the effectiveness of RNNs drop as input sequences increase, transformers use positional embeddings to bypass constraints around the 'window size' of sequences that can be considered by the model. These input positional embeddings are used to create a score matrix that determines how much "attention" should be afforded to each combination of subsequences. In language applications, these subsequences are tokenized words of a sentence. In vision applications, these subsequences are 16x16 pixel patches of the input image.

Swin Transformer (Swin-T) is a Transformer-based model designed for computer vision tasks, introduced by Liu et al. in the paper "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows" (2021)[22]. It adapts the Transformer architecture, originally developed for NLP, for image processing by employing local self-attention with shifted windows. Swin-T leverages hierarchical representations by progressively merging local windows into larger ones, enabling efficient processing of high-resolution images. This approach significantly improves performance on a variety of vision tasks, including image classification and object detection.

In practical terms, there currently exists an implementation of the multiclass classification model that this project aims to create by Hive. That model first classifies an image as authentic or AI-generated, before determining the source of the image (if AI-generated). While there exists a demo tool for users to play with, the model itself is not open-source and is proprietary to the company.

2. Methodology

2.1. Data Discovery and Sourcing

Data discovery is a crucial step in building the vision transformer model for detecting whether an image is AI generated or not. The quality of the data used significantly impacted the accuracy and effectiveness of the model. In this project, data discovery involved identifying suitable datasets and data sources and preprocessing the data.

Fig. 1. DALL-E image scraped from instagram (@openaidalle)

Data was gathered to represent four classes: images generated by Stable Diffusion, DALL-E (Fig. 1), and MidJourney, as well as a 'control' dataset consisting of non-AI-generated images.

The control dataset used in the baseline objective will be sampled from the labeled training set provided by the Hugging Face "aiornot" competition. It provides 18,619 jpegs (512x512 pixels), 45% of which are non-AI-generated (8,289 images).

Images generated through Stable Diffusion [20] and MidJourney [11] were obtained from publicly hosted datasets on HuggingFace.

The relative scarcity and somewhat exclusive nature of images produced by OpenAI's DALL-E made it less accessible through open-source platforms (other than a couple datasets consisting entirely of cats [14] or dogs [13]). In addition, users of the service that aren't paying subscribers have usage caps that prevent us from generating thousands of images ourselves. As a result, all DALL-E images used in this project were programmatically scraped from OpenAI's instagram profile, "openaidalle" [15], and dalle2.gallery [16]. We could be confident these images were generated by OpenAI due to the sources they were drawn from and the presence of the characteristic DALL-E watermark in the bottom-right corner of each sample.

In order to avoid pitfalls associated with imbalanced classification problems, we deliberately constructed balanced datasets throughout the project that had an equal number of samples representing each class. The first set of experiments were run with 4000 images (1000 of each class). In the improvements section, we examine the impact of transfer learning with 8000 images (2000 of each class).

2.2. Data Cleaning and Management

Before using the dataset for training, the data was pre-processed to ensure compatibility and consistency across sources. In other words, this involves combining the different sources together into a single dataset with consistent format and then reshaping the images to a size required for the Swin-Tiny and Resnet101 models.

Major risks come from mixing different datasets that contain real images and AI-generated images from MidJourney, Stable Diffusion and Dall-E. This combination will require the need to standardize image inputs as the images will have different sizes. For the images in each of the four categories, we had utilized an Auto Image Processor from the transformer module from HuggingFace to preprocess the data before splitting into a train and test set. The feature extractor resized the images to 224x224 pixels, applied a bilinear interpolation resampling filter (the new pixel values are calculated based on the weighted average of the surrounding four nearest pixels), and normalized the inputs [21].

In the case of the DALL-E dataset, we had an additional preprocessing step to crop 2 pixels off the bottom of each sample in order to remove the watermark. This would prevent any models we train from learning to rely entirely on the watermark to identify DALL-E images.

2.3. Model Identification and Implementation

The primary model leveraged throughout this project is a pre-trained vision transformer hosted on Hugging Face: `swin-tiny-patch4-window7-224` [17].

The '`tiny`' variant refers to a smaller and more computationally efficient version of the model designed to strike a balance between model size and performance, making it suitable for resource-constrained environments. '`patch4`' in the name indicates that the input image is divided into 4x4 pixel patches, the size of which determines the granularity at which the image is processed and allows the model to capture fine details and local information. '`window7`' refers to the sliding

window mechanism used to aggregate information across different patches. Here, each patch is grouped together with its neighboring patches within a window of size 7x7, and the model attends to these patches collectively to capture global context. '`224`' denotes the input image resolution, which is set to 224x224 pixels.

The goal was to tackle the multiclass classification problem using three separate approaches to transfer learning, so we initially performed three corresponding experiments with Swin-Tiny. Examining the resulting test accuracies allowed us to see the impact of progressively fine-tuning more of the model, as well as which classes were more difficult to classify.

The first experiment used the model as a **feature extractor**. Extracted outputs were passed to a logistic regressor implemented in Scikit-learn (`LogisticRegressionCV`) to classify the images.

The second experiment was **fine-tuning with frozen layers**. It involved us freezing all of the parameters up until the final linear layer, and then adding our own linear layer that transformed the output dimensions and handed off to a softmax for the classification.

The third experiment was **selective fine-tuning**: a natural extension to experiment 2 where we froze every layer except the last one (specifically Stage 3, Block 1), which would remain unfrozen and trainable. As with the previous experiment, we added a trainable linear layer with a softmax for classification.

While the project focuses primarily on comparing and visualizing the three methods of transfer learning described above, we also did a brief model comparison by using a ResNet CNN as a feature extractor: `microsoft/resnet-101` [18].

These experiments are all examples of transfer learning, where we take a model that has been pre-trained for a separate image classification task and adapt it for a new task: to predict whether something is human-generated or which generative model was used to create the image. The main idea behind this approach is that AI-generated images often have certain patterns or artifacts that are not present in real images, and that these patterns could be learned by Swin-Tiny or ResNet.

3. Experiments & Observations

The following experiments impose an 80:20 train-test split on the input image data. In the case of our 4,000 image dataset, this means 3,200 images were used for training, while 800 images were held out for testing.

3.1. Feature Extractor

The mean-per-class test accuracy achieved by using the ResNet 101 model as a feature extractor with a logistic regression as the final layer was 81.4%. Table 1 shows the precision, recall and F1-score for this implementation:

Table 1
Classification Report: ResNet as a Feature Extractor

Class	Precision	Recall	F1-score
Non-AI (0)	0.90	0.89	0.89
Dall-E (1)	0.76	0.79	0.77
Midjourney (2)	0.75	0.78	0.76
Stable Diffusion (3)	0.87	0.80	0.84

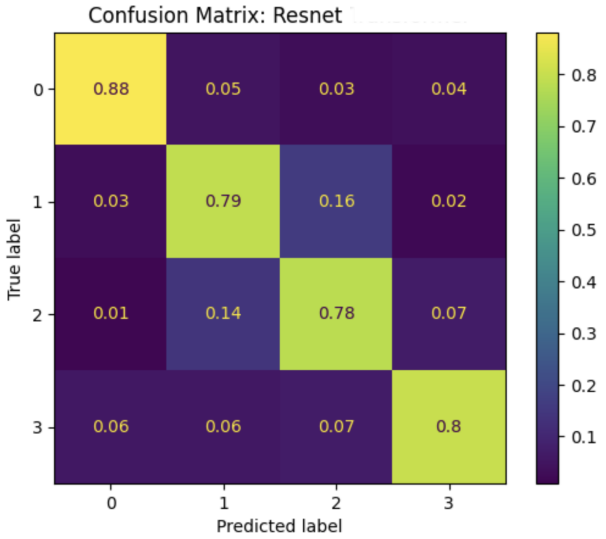

Fig. 2. Confusion matrix: ResNet as a feature extractor

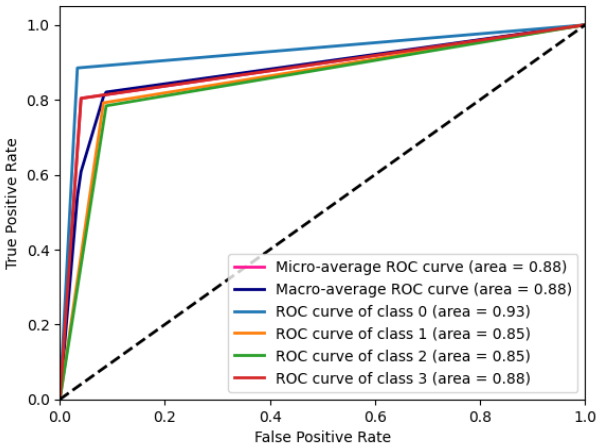

Fig. 3. ROC curves: ResNet as a feature extractor

On the other hand, the Swin transformer model as a

feature extractor with the logistic regression achieved a mean-per-class test accuracy of 86.25%. Table 2 shows the precision, recall and F1-score for this implementation:

Table 2
Classification Report: Swin-Tiny as a Feature Extractor

Class	Precision	Recall	F1-score
Non-AI (0)	0.94	0.90	0.92
Dall-E (1)	0.81	0.81	0.81
Midjourney (2)	0.82	0.88	0.84
Stable Diffusion (3)	0.88	0.86	0.87

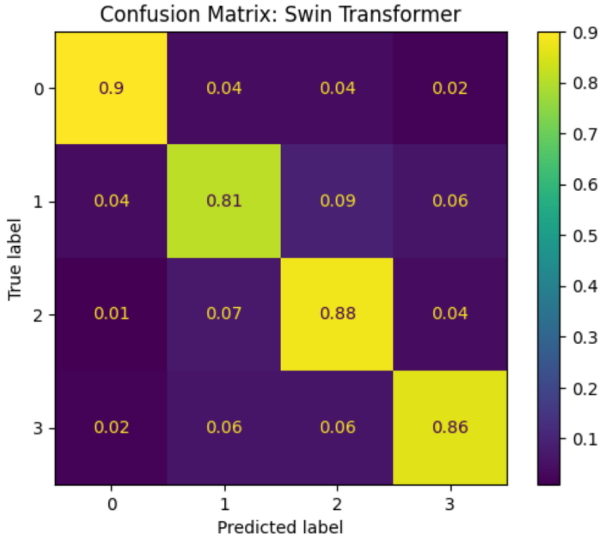

Fig. 4. Confusion matrix: Swin-Tiny as feature extractor

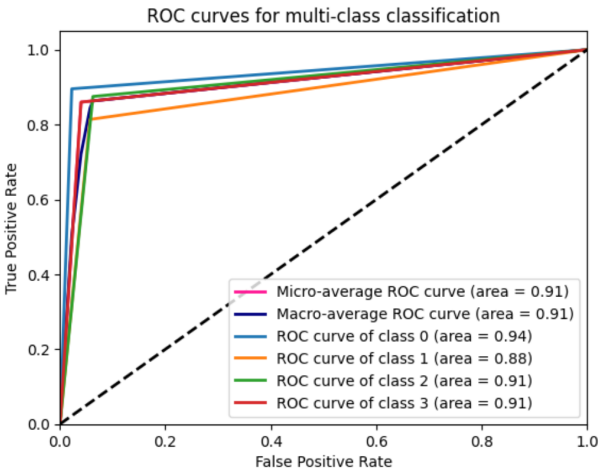

Fig. 5. ROC curves: Swin-Tiny as a feature extractor

As the Swin transformer model got a higher test accuracy, we further improved this model by fine-tuning.

3.2. Fine-Tuning with Frozen Layers

Freezing all of the parameters up until the final linear layer and adding our own classification head yields a more complex model that reached a higher test accuracy of 88.25%. Table 3 shows the precision, recall and F1-score for this implementation:

Table 3
Classification Report: Fine-tuning Swin with frozen layers

Class	Precision	Recall	F1-score
Non-AI (0)	0.98	0.87	0.92
Dall-E (1)	0.82	0.88	0.85
Midjourney (2)	0.84	0.88	0.86
Stable Diffusion (3)	0.89	0.91	0.90

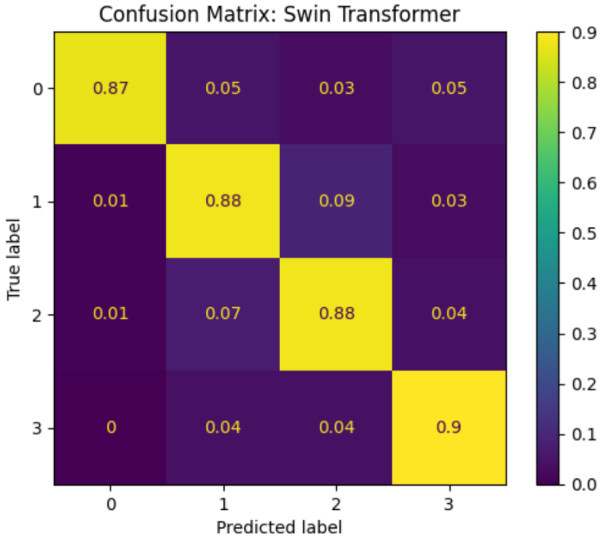

Fig. 6. Confusion matrix: Fine-tuning Swin with frozen layers

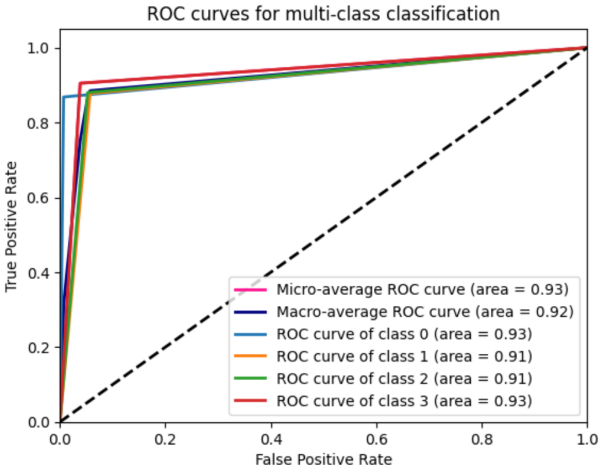

Fig. 7. ROC curves: Fine-tuning Swin with frozen layers

3.3. Selective Fine-Tuning

Removing the restrictions we've placed on fine-tuning so far and unfreezing part of the network further increases the predictive power of the resulting model, with a test accuracy of 91.75%. Table 3 shows the precision, recall and F1-score for this implementation:

Table 3
Classification Report: Selectively Fine-Tuning Swin-Tiny

Class	Precision	Recall	F1-score
Non-AI (0)	0.97	0.94	0.95
Dall-E (1)	0.91	0.90	0.91
Midjourney (2)	0.91	0.92	0.91
Stable Diffusion (3)	0.88	0.91	0.89

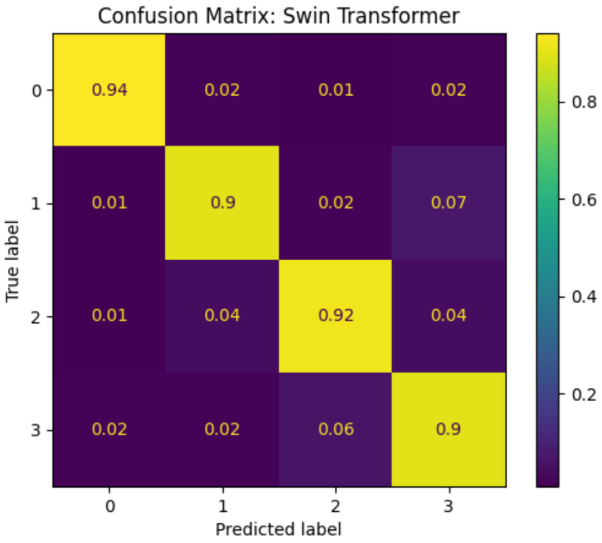

Fig. 8. Confusion matrix: selectively fine-tuning Swin-Tiny

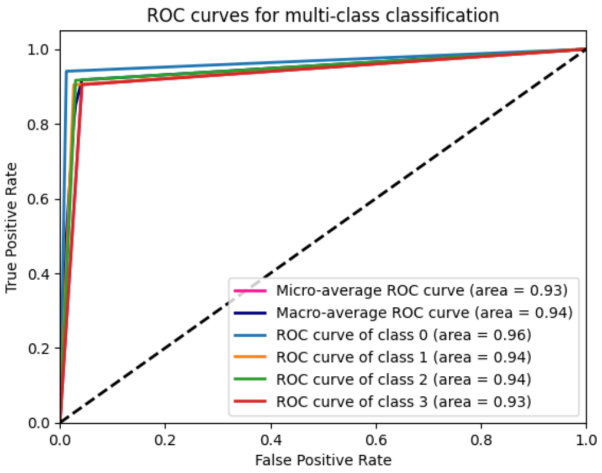

Fig. 9. ROC curves: selectively fine-tuning Swin-Tiny

3.4. Evaluation

Although the previous results were good, we wanted to further improve our classification models. To understand what methods of improvement we should use, we evaluated the two models by doing a bias and variance diagnosis, plotting their confusion matrix and their Receiver Operating Characteristic (ROC) curve.

1) **Bias and Variance Diagnosis** : We started with a bias and variance diagnosis to see if the models were underfitting or overfitting. Table 4 shows the training and test accuracy for both models.

Table 4
Training vs Test Error

Model	Type	Train accuracy	Test accuracy
ResNet	Feature Extractor	93.5%	81.4%
Swin Tiny	Feature Extractor	100%	86.25%
	Frozen Fine-Tuning	97.81%	88.25%
	Selective Fine-Tuning	98.38%	91.75%

As the difference between the training error and the validation error for both models is high, it means we are overfitting.

2) **Confusion Matrix** : We plotted a confusion matrix for predicting the 4 classes. See Fig. 4, Fig. 6, and Fig. 8 above. Table 5 shows a summary of the sensitivity values (true positive rate) for both models used as feature extractors:

Table 5
Sensitivity Summary

Class	Sensitivity (ResNet)	Sensitivity (Swin)
Non-AI (0)	0.86	0.88
DALL-E (1)	0.76	0.80
MidJourney (2)	0.77	0.80
Stable Diffusion (3)	0.86	0.86

The prediction patterns between the two implementations are similar as both models perform the best at correctly classifying non-AI images and images generated by Stable Diffusion. Moreover, both models make the most mistakes by misclassifying images generated by Dall-E and Midjourney.

3) **Receiver Operating Characteristic Curve** : Furthermore, we graphed the ROC curve for both models. As this is a multiclass classification problem, we need to compute it as a set of binary classification in which one class is considered positive and the remaining classes are considered negative. This is done for each class at the time.

In Fig. 5, Fig. 7, and Fig. 9, we can see that the greatest area under the curve comes from the binary classification between class 0 (Non-AI images) and the other 3 classes (AI generated images). This is a really good result as it tackles one of our main objectives, which is to avoid misclassifying human art as AI-generated.

3.5. Improvements

The bias and variance diagnosis described above shows a consistent drop in performance when moving from the train to test set. This suggests we are overfitting the training data and generalizing poorly (i.e., low bias, high variance). To counter overfitting, we ran a second set of experiments for transfer learning on Swin-Tiny, this time with twice the amount of training data. We considered doing some data Augmentation of existing images (e.g. random cropping, rotations, affine transformations etc.), but since additional data was readily available, this was deemed non-critical.

Repeating each of the three experiments with more data consistently results in a better mean-per-class accuracy. In table 6 we can see the impact on accuracy (applied on the same original 800 image test set) after training the models with twice the amount of data:

Table 6
Accuracy Summary.

Training Size (images)	3200	6400
Feature Extraction (mean-per-class accuracy)	0.86	0.88
Transfer Learning (mean-per-class accuracy)	0.88	0.90
Fine-Tuning (mean-per-class accuracy)	0.92	0.93

While this may not represent major gains, even a small improvement of accuracy could represent a lot of images at scale. That being said, there's clearly diminishing returns as we continue to pile on more data and accrue higher costs associated with computational complexity, memory constraints, and training time.

The following sections report specific findings for each of the original experiments repeated on the larger training dataset. One thing of note is that much of the performance gains are coming in predicting classes 0-2 (non-AI, DALL-E, and MidJourney). Although the feature extraction experiment performed particularly poorly on class 3 (Stable Diffusion), the accuracy was comparable with other classes by the time we conducted selective fine-tuning on the larger dataset. This initial poor performance could be due to higher variability across Stable Diffusion images in terms of themes, styles, or content. This would make the class more heterogenous and harder to train a classifier to detect

consistent underlying patterns in the images.

3.5.1 Feature Extractor

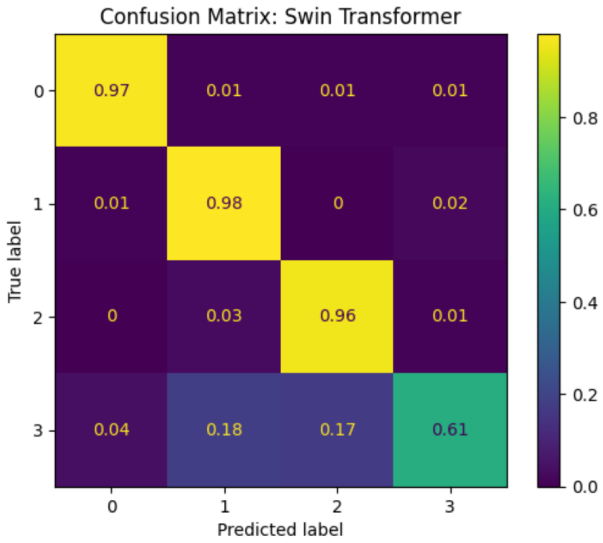

Fig. 10. Confusion matrix: Swin-Tiny as feature extractor trained with twice amount of data

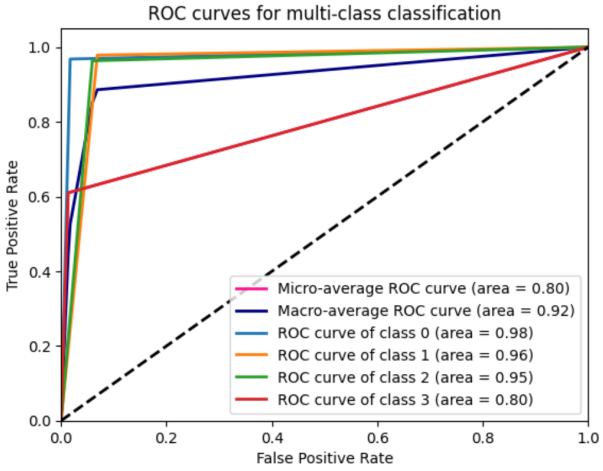

Fig. 11. ROC curves: Swin-Tiny as a feature extractor trained with twice amount of data

3.5.2 Fine-Tuning with Frozen Layers

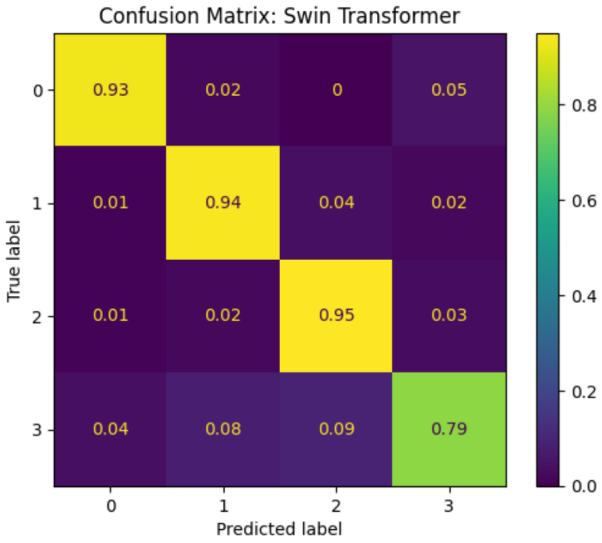

Fig. 12. Confusion matrix: Fine-tuning Swin with frozen layers trained with twice amount of data

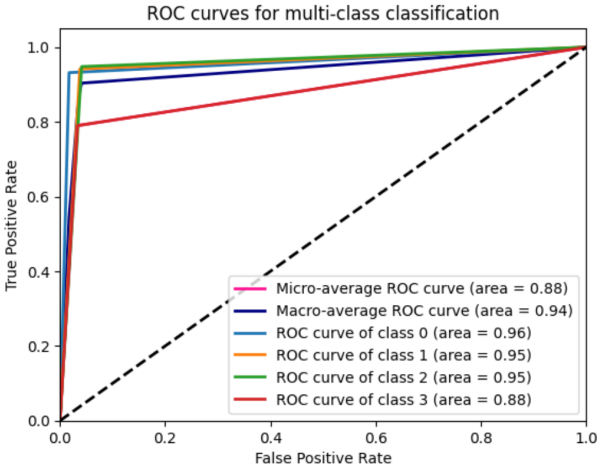

Fig. 13. ROC curves: Fine-tuning Swin with frozen layers trained with twice amount of data

3.5.3 Selective Fine-Tuning

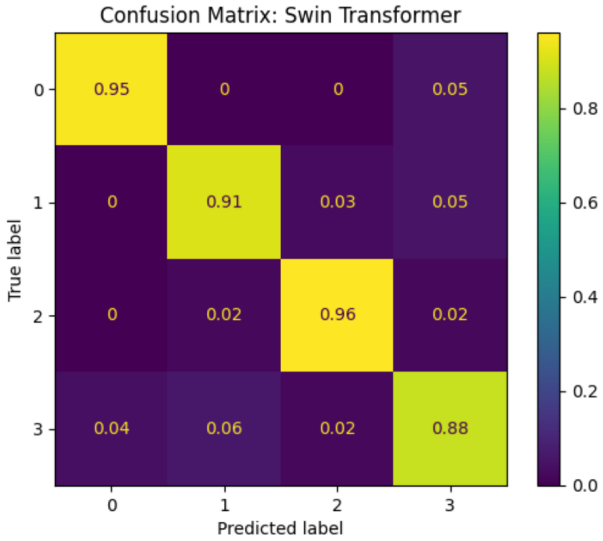

Fig. 14. Confusion matrix: selectively fine-tuning Swin-Tiny trained with twice amount of data

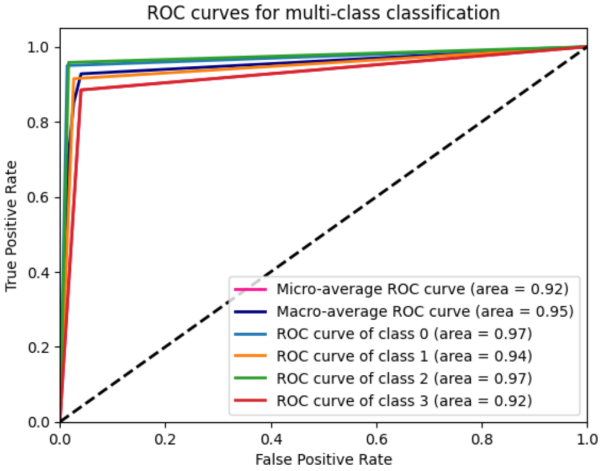

Fig. 15. ROC curves: selectively fine-tuning Swin-Tiny trained with twice amount of data

4. Ethical Considerations

Even if the model built is very accurate, the consequences of even a minor number of false positives can be highly undesirable. In that scenario, artists get their authentic work discredited as a result of their style resembling AI-generated artwork. There will likely be a tradeoff with the reverse scenario, which would contribute to misinformation if AI-generated images are classified as human-generated.

There also exists a risk that models created for this purpose can be used to improve AI image generators down the line e.g., as adversarial discriminators in a GAN. Ironically, this would only make it harder to identify AI-generated images advertised as authentic down the line.

5. Conclusion and Future Work

In the course of this project, we've successfully built models using ResNet and Transformers to identify AI-generated images. These deep learning architectures effectively discerned subtle features in the images, with Transformers showing superior performance.

This project has not only achieved its goals but also provided valuable insights into the capabilities of different architectures. As AI continues to evolve, so too will our models need to adapt, promising a future of exciting research and development in this field.

5.1 Future Work

As we move forward, we believe there are several areas of improvement possible:

5.1.1. Implementation in Real-time Scenarios

Our current model could be further streamlined for optimized performance and real-time execution. The ultimate goal is to deploy a tool that users can utilize in real time to differentiate between AI-generated and authentic images, enhancing the real-world applicability of our project, similar to the features Google is adding to its image search [23].

5.1.2. Dynamic Model Selection

We have chosen the Swin Transformer due to its efficacy for our specific use case. However, given the rapid evolution of AI, it's possible that newer models might emerge that are better suited for this task. We therefore must remain adaptable and open to integrating these advancements into our system.

5.1.3. More Data

To improve our model's performance, we could employ adversarial training techniques. This involves creating a pipeline for periodically scraping images generated by various AI models and implementing batch wise training. This continuous enrichment of our dataset will ensure our model is up-to-date and equipped to handle the ever-evolving landscape of AI-generated images.

Moreover, we can address concerns in the ethical considerations by intentionally creating a class imbalance. Introducing a greater amount of non-AI generated images will encourage the model to reduce false-positives in these cases, which we want to prioritize.

6. Appendix: References

[1] Wang, S.-Y., Wang, O., Zhang, R., Owens, A., & Efros, A. A. (n.d.). CNN-generated images are surprisingly easy to spot... for now. Retrieved March 8, 2023, from https://www.motherjones.com/politics/2019/03/

[2] Gragnaniello, D., Cozzolino, D., Marra, F., Poggi, G., & Verdoliva, L. (2021). Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. http://arxiv.org/abs/2104.02617

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 2017-December, 5999–6009. https://doi.org/10.48550/arxiv.1706.03762

[4] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. http://arxiv.org/abs/2010.11929

[5] Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., & Verdoliva, L. (2022). On the detection of synthetic images generated by diffusion models. http://arxiv.org/abs/2211.00680

[6] We found a way to check if an image is generated by an AI | by Morelli.ai | morelli.ai | Jan, 2023 | Medium. (n.d.). Retrieved March 8, 2023, from https://medium.com/morelli-ai/we-found-a-way-to-check-if-an-image-is-generated-by-an-ai-f97b2c84d8d0

[7] Can an AI learn to identify "AI art"? | by Matthew Maybe | Medium. (n.d.). Retrieved March 8, 2023, from https://medium.com/@matthewmaybe/can-an-ai-learn-to-identify-ai-art-545d9d6af226

[8] Aiornot - a Hugging Face Space by competitions. (n.d.). Retrieved March 8, 2023, from https://huggingface.co/spaces/competitions/aiornot

[9] AI-Generated Content Detection | Hive. (n.d.). Retrieved March 8, 2023, from https://hivemoderation.com/ai-generated-content-detection

[10] mini, D.A.L.L.E. Dalle-mini/yfcc100m_openai_subset · datasets at hugging face, dalle-mini/YFCC100M_OpenAI_subset · Datasets at Hugging Face. Available at: https://huggingface.co/datasets/dalle-mini/YFCC100M_OpenAI_subset (Accessed: March 9, 2023).

[11] Shemesh, D. Danielshemesh/midjourney · Datasets at hugging face, danielshemesh/midjourney · Datasets at Hugging Face. Available at: https://huggingface.co/datasets/danielshemesh/midjourney (Accessed: March 9, 2023).

[12] Raw, N. Nateraw/Midjourney-texttoimage-new · datasets at hugging face, nateraw/midjourney-texttoimage-new · Datasets at Hugging Face. Available at: https://huggingface.co/datasets/nateraw/midjourney-texttoimage-new (Accessed: March 9, 2023).

[13] Big Innovation Research and Development Laboratories. BIRDL/dall-E-dogs · datasets at hugging face, E. Available at: https://huggingface.co/datasets/BirdL/DALL-E-Dogs (Accessed: March 9, 2023).

[14] Big Innovation Research and Development Laboratories. BIRDL/dall-E-cats · datasets at hugging face, E. Available at: https://huggingface.co/datasets/BirdL/DALL-E-Cats (Accessed: March 9, 2023).

[15] *DALL·E by OpenAI (@openaidalle) • Instagram photos and videos*. (n.d.). Retrieved May 14, 2023, from https://www.instagram.com/openaidalle/?hl=en

[16] *DALL·E 2 Largest image database*. (n.d.). Retrieved May 14, 2023, from https://dalle2.gallery/#search

[17] *microsoft/swin-tiny-patch4-window7-224 · Hugging Face*. (n.d.). Retrieved May 14, 2023, from https://huggingface.co/microsoft/swin-tiny-patch4-window7-224

[18] *microsoft/resnet-101 · Hugging Face*. (n.d.). Retrieved May 14, 2023, from https://huggingface.co/microsoft/resnet-101

[19] Wang, Z. J., Montoya, E., Munechika, D., Yang, H., Hoover, B., & Chau, D. H. (2022). *DiffusionDB: A Large-scale Prompt Gallery Dataset for Text-to-Image Generative Models*. https://huggingface.co/datasets/poloclub/diffusiondb

[20] *poloclub/diffusiondb · Datasets at Hugging Face*. (n.d.). Retrieved May 14, 2023, from https://huggingface.co/datasets/poloclub/diffusiondb

[21] *Vision Transformer (ViT)*. (n.d.). Retrieved May 14, 2023, from https://huggingface.co/docs/transformers/v4.17.0/en/model_doc/vit#transformers.ViTFeatureExtractor

[22] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., Huang, Y., & Dai, J. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv preprint arXiv:2103.14030.

[23] *Google introduces new features to help identify AI images in Search and elsewhere | TechCrunch*. (n.d.). Retrieved May 14, 2023, from https://techcrunch.com/2023/05/10/google-introduces-new-features-to-help-identify-ai-images-in-search-and-elsewhere/?utm_medium=TCnewsletter&tpcc=TCdailynewsletter&guccounter=1