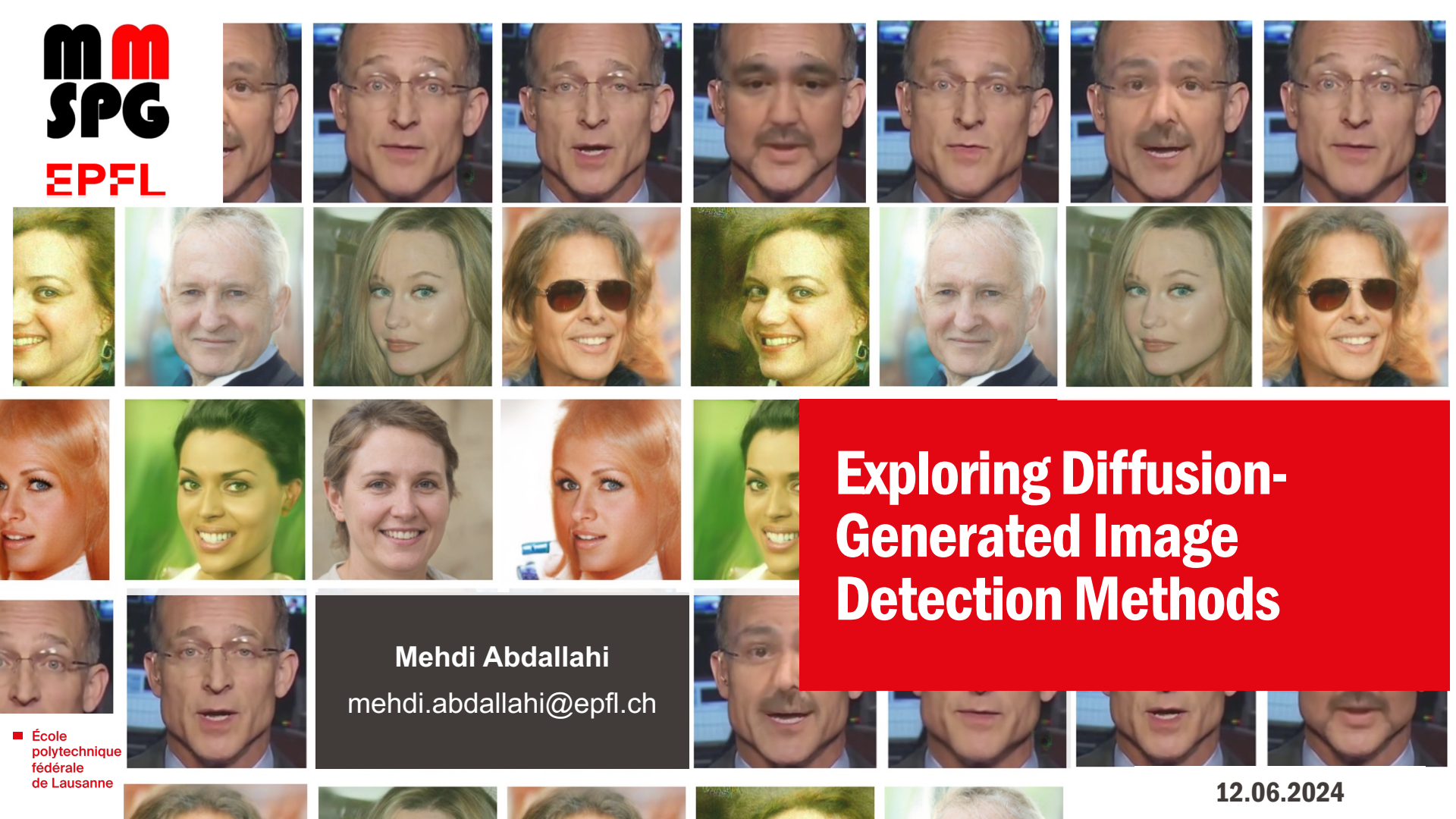


Exploring Diffusion-Generated Image Detection Methods

Mehdi Abdallahi

mehdi.abdallahi@epfl.ch

École polytechnique fédérale de Lausanne

12.06.2024

Motivation

Misinformation and Social Harm | **Security Threats** | **Ethical and Legal Concerns**

Vague d'indignation suite à la publication de deepfakes pornographiques de Taylor Swift

POLITICS | MARCH 15, 2019

The Bizarre and Terrifying Case of the "Deepfake" Video that Helped Bring an African Nation to the Brink

A controversial appearance by the president of Gabon portends a future where you ca believe your eyes.

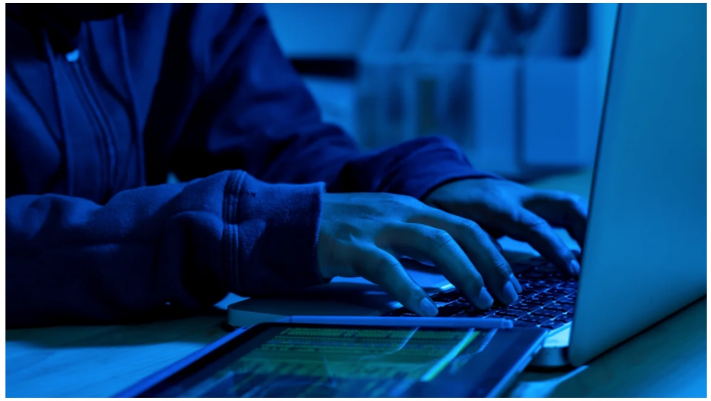
ALI BRELAND
Reporter
Bio

Government of Gabon

Finance worker pays out $25 million after video call with deepfake 'chief financial officer'

By Heather Chen and Kathleen Magramo, CNN

2 minute read · Published 2:31 AM EST, Sun February 4, 2024



Authorities are increasingly concerned at the damaging potential posed by artificial intelligence technology. boonchai wedmakawand/Moment RF/Getty Images

By Marianna Spring
BBC disinformation and social media correspondent

London Mayor Sadiq Khan says deepfake audio of him supposedly making inflammatory remarks before Armistice Day almost caused "serious disorder".

ié jeudi à 12:55

Amandine Le Pen, ce e au service de l'extrême

tendue "nièce" de Marine Le Pen dépasse les 32'000 abonnés. - [DR]

Avec ses plus de 30'000 abonnés, Amandine Le Pen (le compte a été modifié puis banni après la publication de cet article, ndlr) a construit une solide communauté sur TikTok. Elle se présente comme une "fière Française" qui adore "sa tata", Marine Le Pen, et parsème chacune de ses publications d'un drapeau français. Seul hic: elle n'est pas réelle. D'autres comptes du même type pullulent sur l'application chinoise.

Final Presen

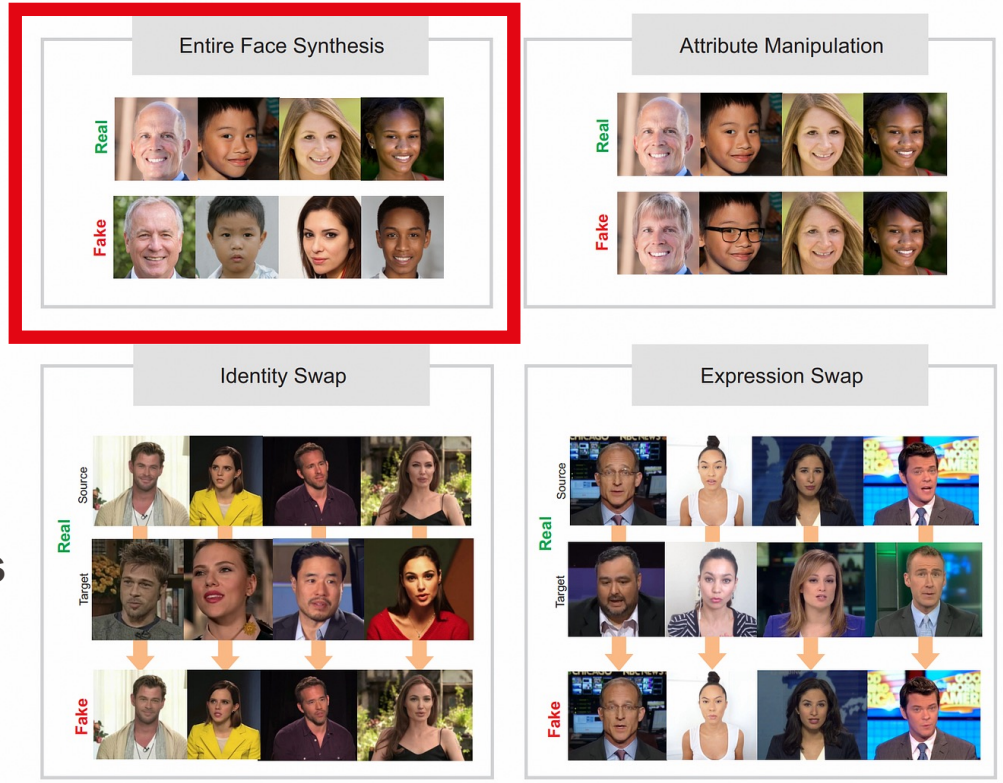
https://www.motherjones.com/politics/2019/03/deepfake-gabon-ali-bongo/

Introduction

Different types of face manipulation techniques:

1. Attribute Manipulation
2. Identity Swap
3. Expression Swap
4. Entire Face Synthesis

→ **Focus on Entire Face Synthesis (static images).**



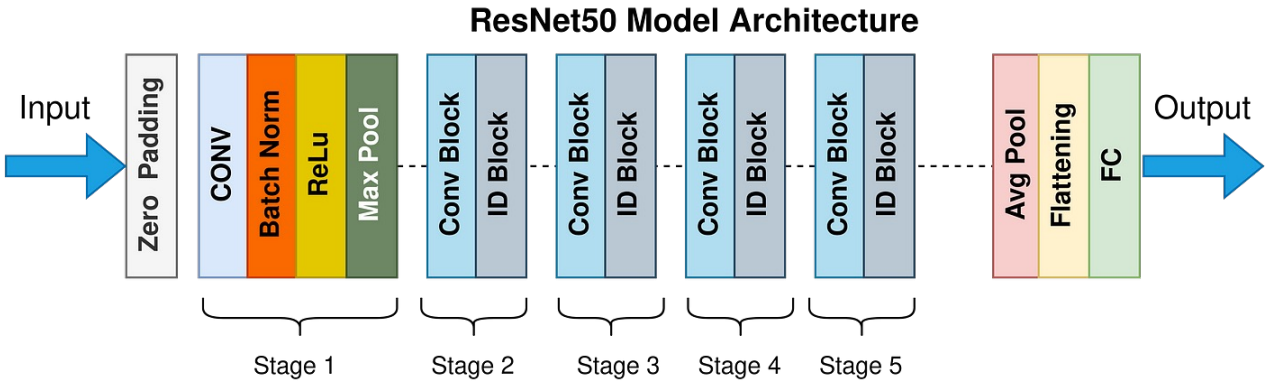
[2]

Wang et al. 2020 [27]

CNN-generated images are surprisingly easy to spot... for now

- Model used: **ResNet50** [6] (pre-trained weights on ImageNet v1 [22])
- Approach: create a detector to identify synthesized images, regardless of the method used to obtain them

→ **Obtained good generalization on all synthesized images.**

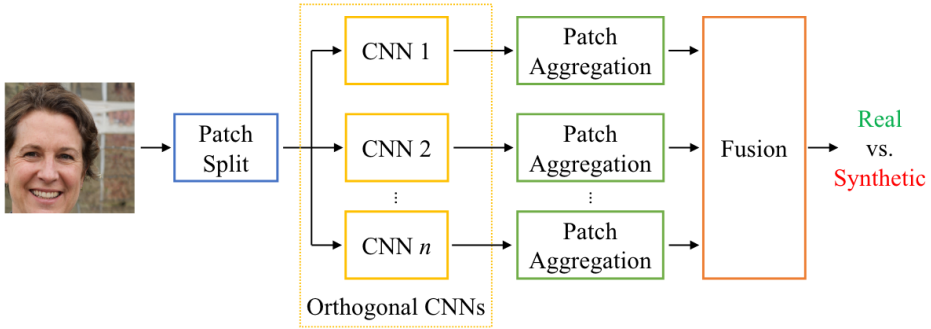


https://medium.com/@nayanchaure601/variants-of-resnet-a-comparative-analysis-63fdc1573b34

Mandelli et al. 2022 [16]

Detecting GAN-generated images by orthogonal training of multiple CNNs

- Ranked 1st in competition organised by NVIDIA
- **Train a batch of models "orthogonally"**
- Each model classifies 1000 patches of the image, the highest/lowest score is retained. Final score is the average of all the CNNs

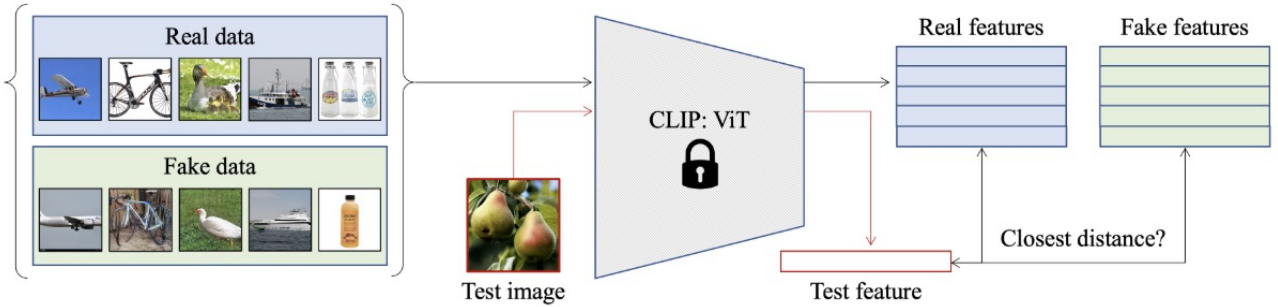


→ **Very accurate predictions for entire face synthesis images.**

Ojha et al. 2023 [17]

Towards Universal Fake Image Detectors that Generalize Across Generative Models

- Adressed issues in classification: the **real class acts as a 'sink' class**
- Method: use a **feature space** that hasn't been trained for this task (unbiased) and apply **kNN or Linear Classifier** to determine the class



→ **Improved generalization of the detection of synthesized images.**

Lu and Ebrahimi 2024 [17]

Towards the detection of AI-synthesized human face images

- **Benchmark** to comprehensibely assess the generalization ability of detectors against synthesized human face images, composed of **images generated by GANs and diffusion models**.
- Showed detectors using frequency representations showed significant improvements across various generative models.

→ **Presents a robust benchmark for evaluating the effectiveness of AI-synthesized human face detectors.**

Our approach

- Build a pipeline to train/validate/test the models.
- Test a lot of different models to find the best **architecture** for generalization.
- Find the best **pre-processing/augmentations** to train the models (blurring, compression…)
- Find the best method for **generalization**.

Datasets

Family	Dataset	Class	train/val/test	Size	Format
GANs [15]	ProGAN [9]	fake	38k / 1k / 1k	256x256	PNG
	StyleGAN2 [8]	fake	38k / 1k / 1k	256x256	PNG
	VQGAN [5]	fake	38k / 1k / 1k	256x256	PNG
Diffusion Models [15]	DDIM [24]	fake	38k / 1k / 1k	256x256	PNG
	PNDM [12]	fake	38k / 1k / 1k	256x256	PNG
	DDPM [7]	fake	38k / 1k / 1k	256x256	PNG
	LDM [20]	fake	38k / 1k / 1k	256x256	PNG
Celebrity HQ [11]	CelebA-HQ-img	real	28k / 1k / 1k	1024x1024	JPEG
FaceForensics++ [21]	Original	real	64k / 1.4k / 1.4k	256x256	PNG
	Deepfakes	fake	64k / 1.4k / 1.4k	256x256	PNG
	Face2Face	fake	64k / 1.4k / 1.4k	256x256	PNG
	FaceSwap	fake	64k / 1.4k / 1.4k	256x256	PNG
	NeuralTextures	fake	64k / 1.4k / 1.4k	256x256	PNG

Datasets

Fa	ormat
G	PNG
	PNG
	PNG
	PNG
Diffu	PNG
	PNG
	PNG
Cel	JPEG
	PNG
FaceFo	PNG
	PNG
	PNG

Figure 2: Images generated by GAN. Examples from: ProGAN, StyleGAN, VQGAN, in order.

Figure 3: Images generated by diffusion. Examples from: DDIM, DDPM, PNDM, LDM, in order.

Figure 1: Images from FaceForensics++ dataset. Examples of: real image, Deepfake, Face2Face, FaceSwap, NeuralTextures, in order.

Models Explored

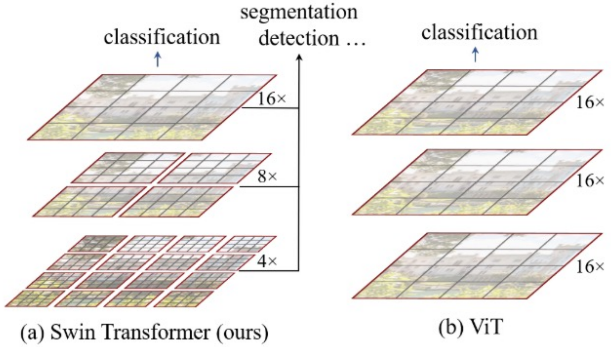
- VGG16 [23]
- Resnet50 [6]
- ResNext [28]
- Efficient Nets (b0 and b4) [25]
- BiT [10]
- Swin Transformers [13]
- ConvNext [14]
- ViT [29]
- DeiT [26]
- BeiT [1]
- RegNet [18]
- CoAtNet [4]

→ Study the levels of generalization for the different architectures.

Models Explored

- BiT Base (CNN, 2020):

Transfer Learning, adapting well to new tasks with minimal re-training.



- Swin Transformer (2021):

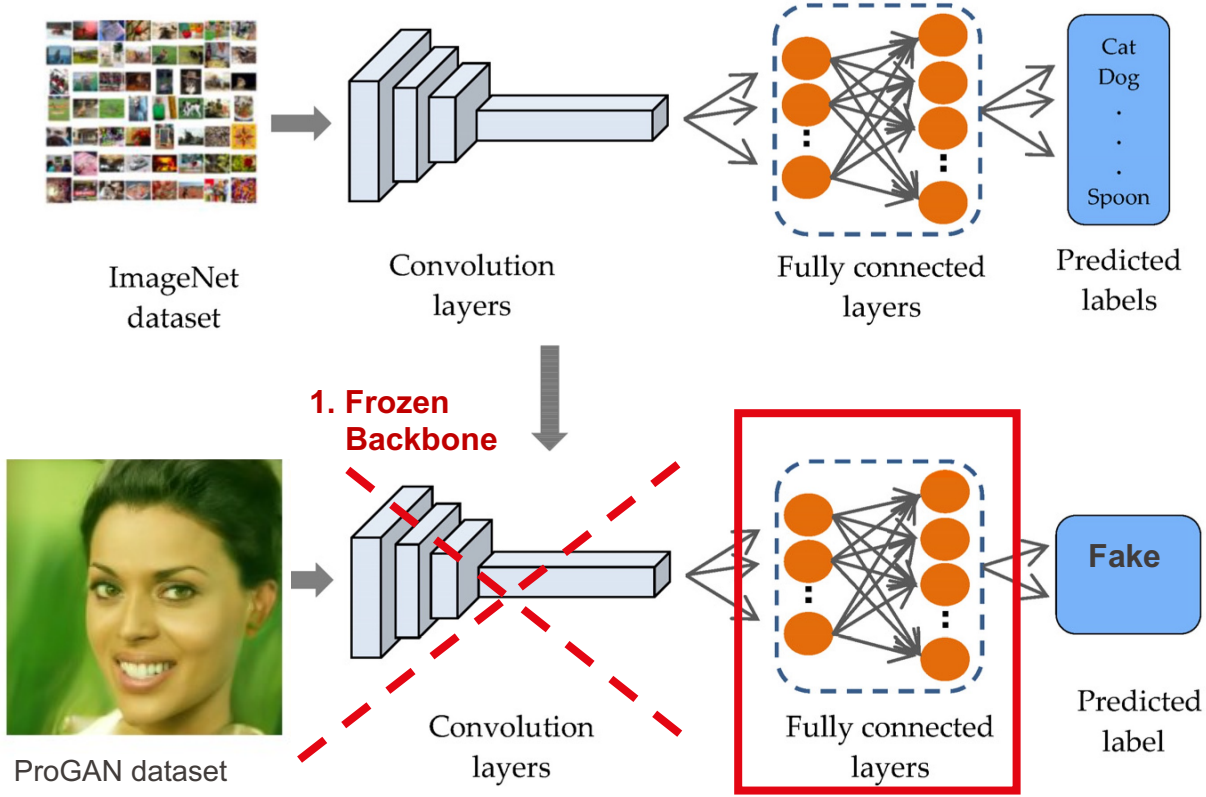
Window-based self-attention to **reduce computational complexity** and **improve spatial relationship capture**. This allows **hierarchical feature learning** (at different levels of abstraction).

- ConvNext (2022):

Combines **CNNs and transformer-inspired architectures**, enabling it to effectively **learn general features** and achieve state-of-the-art performance in diverse vision tasks.

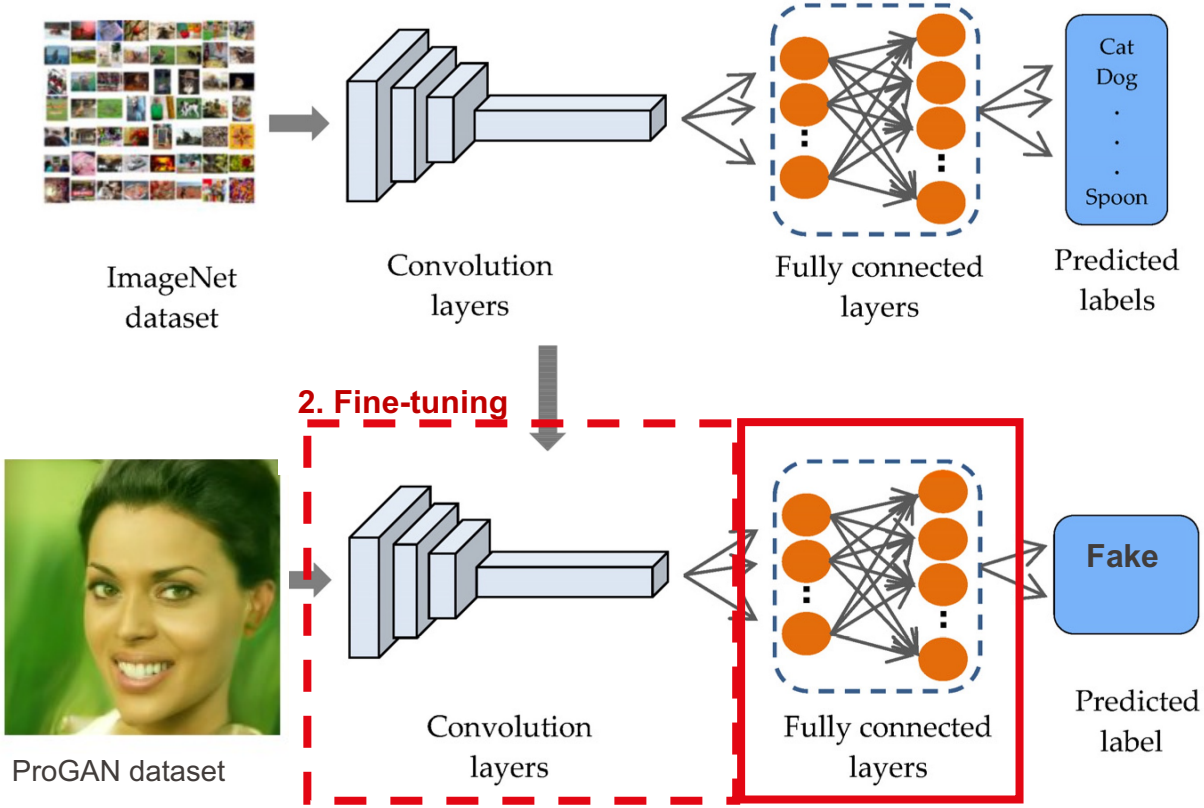
→ **Adapted to our goal of generalizing the detection of synthesized images.**

Types of training



https://www.mdpi.com/1424-8220/23/2/570

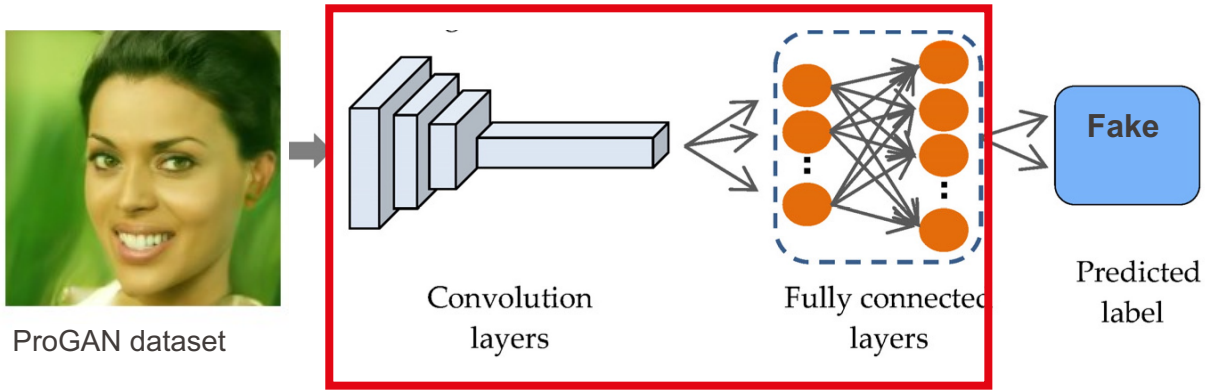
Types of training



https://www.mdpi.com/1424-8220/23/2/570

Types of training

3. Training from scratch (newly initialized weights)



https://www.mdpi.com/1424-8220/23/2/570

First Results and Choice of Training Data

Model	StyleGAN	VQGAN	PNDM	LDM	DDPM	Average
ResNet50	78.49	100.00	100.00	85.93	100.00	92.28
VGG16	44.71	100.00	99.99	85.01	100.00	85.94
EfficientNet b0	72.11	100.00	99.99	53.87	100.00	85.99
EfficientNet b4	98.71	100.00	99.99	97.13	100.00	99.57
Swin Tiny	**99.91**	100.00	100.00	99.87	100.00	**99.96**
Swin Base	99.30	100.00	100.00	**99.99**	100.00	99.86
Swin Large	40.44	100.00	100.00	99.96	100.00	88.88
BiT	96.00	100.00	99.99	98.79	100.00	98.96
ResNext	50.04	100.00	99.99	72.66	100.00	84.54
CoAtNet	47.86	100.00	100.00	42.83	100.00	78.14

Table 3: AP score (in %) for different models fine-tuned (training data: DDIM, ProGAN, Celeb-HQ), tested on images generated by GANs and DMs.

Make it more challenging by using a training set generated by only ProGAN and generalizing to diffusion-generated images

Same Family of Generators (GANs)

Model		GAN Dataset			Average
		ProGAN	StyleGAN	VQGAN	
ResNet50	Freeze Layers	99.79	90.08	99.79	96.55
	Fine Tuning	100.00	81.88	100.00	93.96
	From Scratch	99.97	46.08	99.97	81.34
VGG16	Freeze Layers	91.21	49.16	91.21	77.86
	Fine Tuning	100.00	60.08	100.00	86.69
	From Scratch	99.98	65.20	99.98	88.72
EfficientNet b0	Freeze Layers	98.37	61.29	98.37	86.68
	Fine Tuning	100.00	74.34	100.00	91.45
	From Scratch	99.99	52.63	99.99	84.87
EfficientNet b4	Freeze Layers	98.16	78.88	98.16	91.73
	Fine Tuning	100.00	98.86	100.00	99.62
	From Scratch	99.95	49.81	99.95	83.91
BiT	Freeze Layers	98.28	76.23	98.28	90.93
	Fine Tuning	100.00	99.39	100.00	99.80
	From Scratch	100.00	42.45	100.00	80.82
Swin tiny	Freeze Layers	99.66	85.13	99.66	94.82
	Fine Tuning	100.00	99.99	100.00	99.99
	From Scratch	100.00	45.05	100.00	81.68
ConvNext	Freeze Layers	99.92	91.27	99.92	97.04
	Fine Tuning	100.00	88.07	100.00	96.02
	From Scratch	100.00	40.17	100.00	80.06

Table 6: AP (in %) for different models and training methods, tested on ProGAN, StyleGAN, and VQGAN datasets (training set: ProGAN and CelebA-HQ-img).

Generalization to Diffusion Models

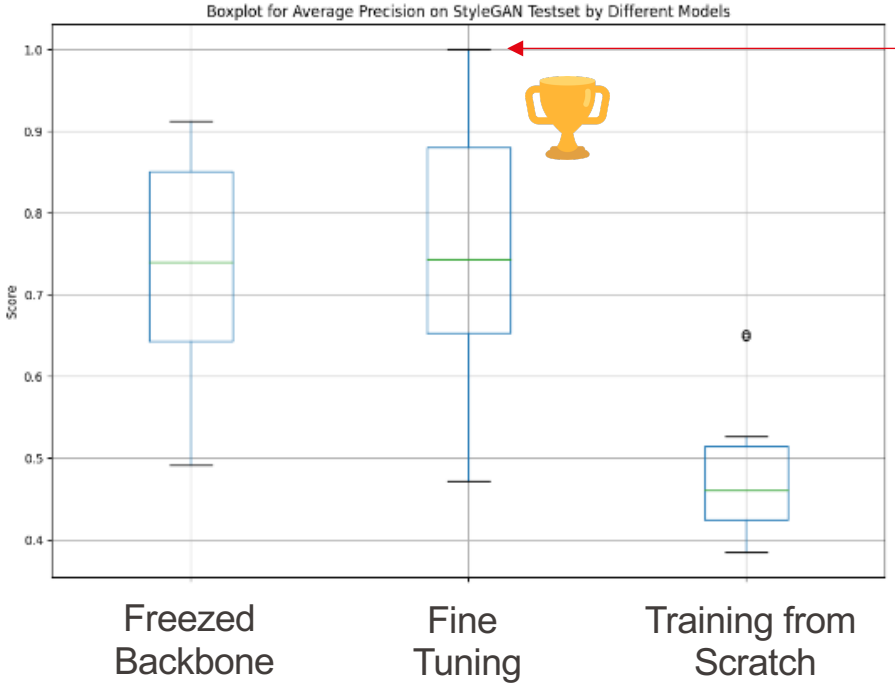
Model		Diffusion Models Dataset				Average
		DDIM	DDPM	PNDM	LDM	
ResNet50	Freeze Layers	57.79	54.48	66.86	82.93	65.65
	Fine Tuning	62.28	53.89	68.90	84.73	67.45
	From Scratch	76.66	66.30	76.21	66.82	71.45
VGG16	Freeze Layers	83.42	66.50	79.19	75.79	76.22
	Fine Tuning	94.08	88.70	94.08	87.77	91.16
	From Scratch	86.05	75.81	86.41	69.72	79.50
EfficientNet b0	Freeze Layers	72.27	57.48	78.40	53.84	65.45
	Fine Tuning	49.20	52.45	60.33	52.23	53.50
	From Scratch	63.27	56.43	61.94	63.57	61.30
EfficientNet b4	Freeze Layers	60.66	51.77	69.33	67.98	62.48
	Fine Tuning	71.82	61.14	94.20	92.74	79.98
	From Scratch	75.69	68.68	75.38	60.59	70.03
BiT	Freeze Layers	74.70	61.24	84.75	86.15	76.76
	Fine Tuning	69.31	71.39	96.27	98.71	83.92
	From Scratch	62.58	47.14	61.44	59.30	57.62
Swin Tiny	Freeze Layers	53.66	41.08	65.68	95.56	63.99
	Fine Tuning	90.21	91.29	99.85	99.88	95.36
	From Scratch	69.94	51.58	72.71	57.17	62.80
ConvNext	Freeze Layers	72.21	65.71	84.59	95.52	79.56
	Fine Tuning	100.00	100.00	100.00	100.00	100.00
	From Scratch	68.58	53.33	70.05	69.57	65.33

Table 8: AP (in %) for different models and training methods, tested on DDIM, DDPM, PNDM, and LDM datasets (training set: ProGAN and CelebA-HQ-img).

Summary of intermediate results

- All the models achieve good performance on GANs, but **not when testing on diffusion-generated images**.
- BiT didn't perform as good as expected, effective augmentations could improve the results.
- **Swin** and **ConvNeXt** achieved **best performance**.
- Fine-tuning appears to be the best method of training.

Summary of intermediate results



Only models that are fine-tuned achieve perfect predictions for StyleGAN

Figure 5: Boxplot for AP on StyleGAN Testset by Different Models for different Training Modes (training data: ProGAN, Celeb-HQ)

Pre-processing

- **Compression** (quality of 75)
- **Blurring** with a kernel of (5,5)

Figure 6: Comparison between original (left) and blurred image (right) from the LDM dataset.

→ Testing for different proportion of the training set.

Pre-processing: Results on Same Family of Generators (GANs)

Percentage transformed		Testing Dataset			Average
Compression (%)	Blurring (%)	ProGAN	StyleGAN	VQGAN	
-	-	100.00	96.37	100.00	98.79
20%	-	100.00	51.41	100.00	83.80
40%	-	100.00	55.97	100.00	85.32
60%	-	100.00	46.36	100.00	82.12
80%	-	100.00	51.63	100.00	83.88
100%	-	100.00	47.39	100.00	82.46
10%	10%	100.00	68.51	100.00	89.50
-	20%	100.00	98.97	100.00	99.66
-	40%	100.00	98.60	100.00	99.53
-	60%	100.00	89.24	100.00	96.41
-	80%	100.00	94.35	100.00	98.12
-	100%	100.00	92.41	100.00	97.47

Table 4: AP (in %) for for Swin Tiny (batch size 128), tested on ProGAN, StyleGAN, and VQGAN datasets (training set: ProGAN and CelebA-HQ-img).

Pre-processing: Results on Generalization to Diffusion Models

Percentage transformed		Testing Dataset				Average
Compression (%)	Blurring (%)	DDIM	DDPM	PNDM	LDM	
-	-	71.92	69.31	89.57	96.50	81.33
20%	-	81.84	78.46	84.40	84.13	82.71
40%	-	86.87	86.77	87.15	78.02	84.20
60%	-	83.13	79.75	85.46	70.84	79.80
80%	-	82.49	81.19	83.90	71.63	79.80
100%	-	71.59	67.99	70.73	72.97	70.82
10%	10%	89.31	82.44	91.22	79.52	85.62
-	20%	90.27	84.35	98.02	99.16	92.95
-	40%	90.94	79.12	97.27	99.05	91.10
-	60%	80.67	76.43	94.19	94.27	86.39
-	80%	59.82	50.74	81.11	93.74	71.35
-	100%	51.73	49.84	68.77	93.28	65.91

Table 5: AP (in %) for for Swin Tiny (batch size 128), tested on DDIM, DDPM, PNDM, and LDM datasets (training set: ProGAN and CelebA-HQ-img).

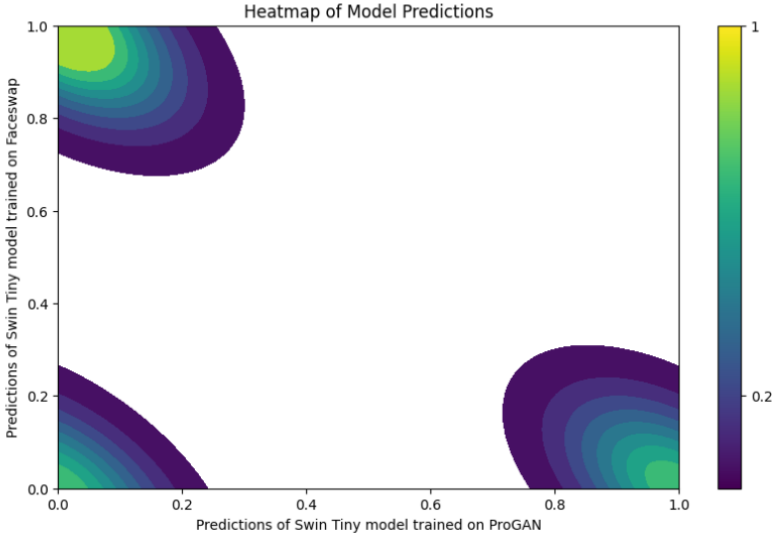
Summary of pre-processing results

- **Impact on GANs (StyleGAN2)**: Compressing images for training reduces model performance on the StyleGAN2 test set, while **moderate blurring improves classification**.

- **Impact on Diffusion Models**: Moderate image compression (around 40%) generally improves classification performance on diffusion models, **moderate blurring also improves classification results**.

Voting Ensembles

- Inspired by Mandelli et al.
- Ability to pick *best* prediction of both models.
- Tried to use a **meta model** trained on the predictions of the training set (Linear Regression).

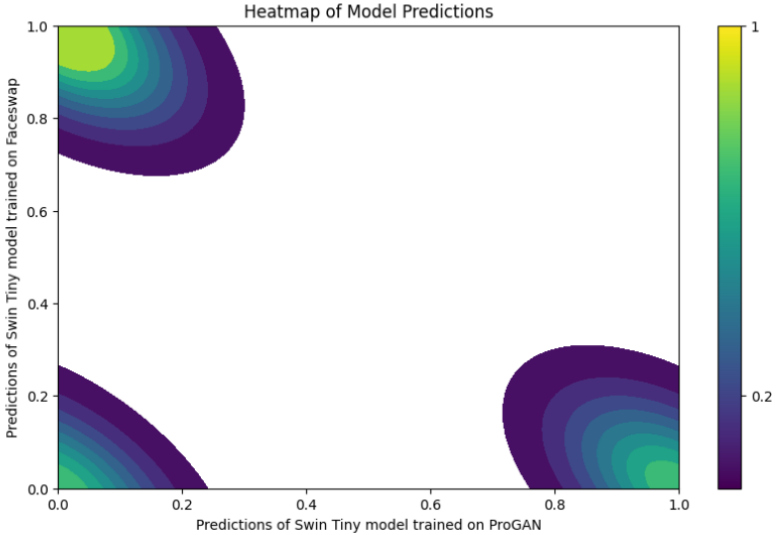


Type of Detector	GANs and DMs							Average
	ProGAN	StyleGAN	VQGAN	DDIM	DDPM	PNDM	LDM	
Swin tiny FaceSwap	48.37	47.14	48.37	72.36	64.79	69.05	63.49	59.37
Swin tiny ProGAN	100.00	99.99	100.00	90.21	91.29	99.85	99.88	97.60
Averaging Predictions	100.00	98.41	100.00	76.73	69.44	96.65	98.52	91.39
Meta Model	100.00	98.44	100.00	76.78	69.49	96.69	98.54	91.42
Swin tiny ProGAN+Faceswap	100.00	97.70	100.00	93.42	90.53	98.13	99.89	97.24

→ Not so interesting here.

Voting Ensembles

- Inspired by Mandelli et al.
- Ability to pick *best* prediction of both models.
- Tried to use a **meta model** trained on the predictions of the training set (Linear Regression).



Type of Detector	FaceForensics++				Average
	Deepfakes	Face2Face	FaceSwap	NeuralTextures	
Swin tiny FaceSwap	63.06	75.21	99.84	51.36	72.37
Swin tiny ProGAN	56.11	62.59	55.86	65.78	60.59
Averaging Predictions	58.25	65.34	99.43	65.91	72.23
Meta Model	57.96	64.87	95.12	65.85	70.95
Swin tiny ProGAN+Faceswap	69.24	73.97	99.69	51.00	73.47

→ Higher precision on NeuralTextures.

Exploration of FaceForensics++

Training Dataset	Testing Dataset							Average
	ProGAN	StyleGAN	VQGAN	DDIM	DDPM	PNDM	LDM	
Deepfakes	100.00	96.99	100.00	87.53	83.21	94.60	98.65	94.14
Face2Face	65.85	64.68	65.85	52.43	55.34	50.51	55.65	58.04
FaceSwap	48.37	47.14	48.37	72.36	64.79	69.05	63.49	59.37
NeuralTextures	59.15	51.45	59.15	44.62	41.74	47.63	55.17	51.13

Table 11: AP (in %) for swin tiny trained trained on different datasets and tested on various GAN and DM datasets

Training Dataset	Testing Dataset				Average
	Deepfakes	Face2Face	FaceSwap	NeuralTextures	
Deepfakes	99.79	71.24	41.57	74.95	71.39
Face2Face	78.86	99.70	50.11	62.91	72.40
FaceSwap	63.06	75.21	99.84	51.36	72.37
NeuralTextures	81.82	63.99	44.55	98.39	72.19

Table 12: AP (in %) for swin tiny trained on different datasets and tested on FaceForensics++ datasets

Conclusion and Summary of Results

- **Fine-tuning** appears to work best **for generalization** over different families of image generators.
- Managed to get **perfect separation** between real and diffusion-generated images when training on images generated by ProGAN.
- Hard to generalize to other types of deepfakes (FaceForensics++).
- Voting ensembles are interesting for specific applications.
- Best models: **Swin Transformers, ConvNeXt.**

Future Explorations

- Try Voting Ensembles with models trained **using different techniques**: entire image (ConvNeXt, Swin), and frequency transformations (FFT or others [15] [27]).

- Use **kNN or a Linear Classifier** to learn using the **feature space** of a set of orthogonally trained models.

- Observe the impact of the **quality of the training sets:** experiment with different training sets (FID score) to observe the impact on generalization.

References

- [1] Hangbo Bao, Li Dong, and Furu Wei. “**BEiT: BERT Pre-Training of Image Transformers**”. In: CoRR abs/2106.08254 (2021). arXiv: 2106.08254. url: https://arxiv.org/abs/2106.08254.
- [2] Ruben Tolosana et al. “**DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection**”. In: arXiv preprint arXiv:2001.00179 (2020)
- [3] Jiaxuan Chen, Jieteng Yao, and Li Niu. **A Single Simple Patch is All You Need for AI-generated Image Detection**. 2024. arXiv: 2402.01123 [cs.CV].
- [4] Zihang Dai et al. “**CoAtNet: Marrying Convolution and Attention for All Data Sizes**”. In:arXiv preprint arXiv:2106.04803 (2021).
- [5] Patrick Esser, Robin Rombach, and Bjorn Ommer. “**Taming transformers for high-resolution image synthesis**”. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, pp. 12873–12883.
- [6] Kaiming He et al. “**Deep Residual Learning for Image Recognition**”. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.
- [7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “**Denoising diffusion probabilistic models**”. In: Advances in neural information processing systems 33 (2020), pp. 6840–6851.
- [8] Tero Karras et al. “**Analyzing and improving the image quality of StyleGAN**”. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
- [9] Tero Karras et al. “**Progressive growing of gans for improved quality, stability, and variation**”. In: arXiv preprint arXiv:1710.10196 (2017).
- [10] Alexander Kolesnikov et al. **Big Transfer (BiT): General Visual Representation Learning**. 2019. doi: 10.48550/ARXIV.1912.11370. url: https://arxiv.org/abs/1912.11370.
- [11] Yuezun Li et al. “**Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics**”. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.22
- [12] Luping Liu et al. “**Pseudo numerical methods for diffusion models on manifolds**”. In: arXiv preprint arXiv:2202.09778 (2022).
- [13] Ze Liu et al. “**Swin Transformer: Hierarchical Vision Transformer using Shifted Windows**”. In: Proc. IEEE/CVF International Conference on Computer Vision (ICCV).2021, pp. 10012–10022. doi: 10.1109/ICCV48922.2021.00987.
- [14] Zhuang Liu et al. “**A ConvNet for the 2020s**”. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022, pp. 11976–11986. doi: 10 . 1109/CVPR52688.2022.01169.
- [15] Yuhang Lu and Touradj Ebrahimi. “**Towards the Detection of AI-Synthesized Human Face Images**”. In: arXiv preprint arXiv:2402.08750 (2024). Available at https://arxiv.org/abs/2402.08750.

References

- [16] Stefano Mandelli et al. “**Detecting GAN-Generated Images by Orthogonal Training of Multiple CNNs**”. In: arXiv preprint arXiv:2203.02246 (2022). Available at https://arxiv.org/abs/2203.02246.
- [17] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. “**Towards Universal Fake Image Detectors that Generalize Across Generative Models**”. In: arXiv preprint arXiv:2302.10174 (2023). Available at https://arxiv.org/abs/2302.10174.
- [18] Ilija Radosavovic et al. “**Designing Network Design Spaces**”. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020, pp. 10428–10436. doi: 10.1109/CVPR42600.2020.01044.
- [19] Benjamin Recht et al. **Do ImageNet Classifiers Generalize to ImageNet?** 2019. arXiv:1902.10811 [cs.CV].
- [20] Robin Rombach et al. “**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.
- [21] Andreas Rössler et al. “**FaceForensics++: Learning to Detect Manipulated Facial Images**”. In: Proc. IEEE/CVF International Conference on Computer Vision. 2019.
- [22] Olga Russakovsky et al. **ImageNet Large Scale Visual Recognition Challenge**. 2015. arXiv:1409.0575 [cs.CV].
- [23] Karen Simonyan and Andrew Zisserman. “**Very Deep Convolutional Networks for Large-Scale Image Recognition**”. In: arXiv preprint arXiv:1409.1556 (2015).
- [24] Jiaming Song, Chenlin Meng, and Stefano Ermon. “**Denoising diffusion implicit models**”.In: arXiv preprint arXiv:2010.02502 (2020).
- [25] Mingxing Tan and Quoc V. Le. “**EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks**”. In: Proc. International Conference on Machine Learning (ICML). arXiv preprint arXiv:1905.11946. 2019, pp. 6105–6114.
- [26] Hugo Touvron et al. **Training data-efficient image transformers distillation through attention**. 2021. arXiv: 2012.12877 [cs.CV].
- [27] Sheng-Yu Wang et al. “**CNN-generated images are surprisingly easy to spot... for now**”.In: arXiv preprint arXiv:1912.11035v2 (2019). Available at https://arxiv.org/abs/1912.11035v2.
- [28] Saining Xie et al. “**Aggregated Residual Transformations for Deep Neural Networks**”. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017,pp. 1492–1500. doi: 10.1109/CVPR.2017.634.23
- [29] Alexey Dosovitskiy et al. **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale**. 2021. arXiv: 2010.11929 [cs.CV]

Thank you!

Multimedia Signal Processing Group

EPFL

https://mmspg.epfl.ch/

Mehdi Abdallahi

mehdi.abdallahi@epfl.ch

Appendix: Same Family of Generators (GANs)

Model		GAN Dataset			Average
		ProGAN	StyleGAN	VQGAN	
ViT	Freeze Layers	98.27 / 98.27	62.41 / 65.48	98.27 / 98.27	86.32 / 87.34
	Fine Tuning	100.00 / 100.00	74.10 / 77.23	100.00 / 100.00	91.37 / 92.41
	From Scratch	100.00 / 100.00	39.22 / 34.21	100.00 / 100.00	79.74 / 78.07
DeiT	Freeze Layers	99.80 / 99.83	73.82 / 76.57	99.80 / 99.83	91.81 / 92.08
	Fine Tuning	100.00 / 100.00	72.94 / 73.64	100.00 / 100.00	90.98 / 91.21
	From Scratch	99.99 / 99.99	38.51 / 31.79	99.99 / 99.99	79.50 / 77.26
CoAtNet	Freeze Layers	99.19 / 99.26	85.11 / 86.15	99.19 / 99.26	94.50 / 94.89
	Fine Tuning	100.00 / 100.00	65.35 / 74.54	100.00 / 100.00	88.45 / 91.51
	From Scratch	100.00 / 100.00	64.91 / 60.57	100.00 / 100.00	88.30 / 86.86
BeiT	Freeze Layers	90.87 / 90.88	64.30 / 66.38	90.87 / 90.88	82.01 / 82.71
	Fine Tuning	100.00 / 100.00	47.14 / 50.96	100.00 / 100.00	82.38 / 83.65
	From Scratch	100.00 / 100.00	45.31 / 45.94	100.00 / 100.00	81.77 / 81.98
ResNext	Freeze Layers	99.47 / 99.47	73.95 / 79.32	99.47 / 99.47	90.96 / 92.08
	Fine Tuning	100.00 / 100.00	60.26 / 62.95	100.00 / 100.00	86.75 / 87.65
	From Scratch	99.65 / 99.64	51.42 / 54.18	99.65 / 99.64	83.57 / 84.82
RegNet	Freeze Layers	97.70 / 97.57	72.02 / 73.70	97.70 / 97.57	89.81 / 89.61
	Fine Tuning	100.00 / 100.00	85.19 / 84.45	100.00 / 100.00	95.06 / 94.82
	From Scratch	99.83 / 99.82	47.16 / 46.96	99.83 / 99.82	82.60 / 82.23

Table 6: AP / AUC scores (in %) for different models and training methods, tested on ProGAN, StyleGAN, and VQGAN datasets (training set: ProGAN and CelebA-HQ-img).

Appendix: Generalization to Diffusion Models

Model		Diffusion Models Dataset				Averaged
		DDIM	DDPM	PNDM	LDM	
ViT	Freeze Layers	84.29 / 83.44	77.83 / 78.02	81.58 / 80.45	83.14 / 83.07	81.71 / 81.29
	Fine Tuning	77.52 / 75.85	72.34 / 71.93	80.36 / 78.42	74.00 / 71.09	76.06 / 74.32
	From Scratch	74.92 / 73.43	63.42 / 64.37	78.31 / 76.37	62.79 / 61.23	69.86 / 68.87
DeiT	Freeze Layers	73.45 / 73.88	57.19 / 57.67	78.28 / 78.19	89.57 / 89.24	74.62 / 74.80
	Fine Tuning	60.17 / 56.30	56.73 / 55.63	63.46 / 60.45	74.10 / 71.39	63.61 / 60.99
	From Scratch	66.40 / 62.16	53.33 / 51.72	68.40 / 64.89	55.86 / 53.14	60.99 / 57.93
CoAtNet	Freeze Layers	78.10 / 79.23	66.67 / 69.54	79.90 / 81.97	80.32 / 81.13	76.25 / 77.91
	Fine Tuning	82.21 / 81.15	70.76 / 68.12	86.68 / 84.92	66.97 / 68.57	76.66 / 75.64
	From Scratch	89.53 / 87.02	84.85 / 82.65	90.75 / 88.71	70.62 / 67.52	83.99 / 81.42
BeiT	Freeze Layers	79.95 / 78.31	71.97 / 72.01	81.30 / 79.83	64.13 / 62.73	74.34 / 73.22
	Fine Tuning	79.44 / 79.57	73.70 / 76.27	83.25 / 82.80	69.86 / 70.88	76.56 / 77.33
	From Scratch	76.48 / 76.66	69.58 / 72.29	80.72 / 80.79	71.47 / 72.63	74.51 / 75.60
ResNext	Freeze Layers	70.69 / 71.48	60.01 / 62.20	73.08 / 73.25	73.65 / 74.00	69.36 / 70.28
	Fine Tuning	49.09 / 50.97	51.51 / 52.94	57.67 / 59.19	58.73 / 56.65	54.20 / 54.88
	From Scratch	87.20 / 90.22	78.52 / 82.15	86.60 / 90.29	68.30 / 72.07	80.16 / 83.68
RegNet	Freeze Layers	71.56 / 71.87	55.44 / 55.79	83.28 / 82.26	65.31 / 64.30	68.90 / 68.50
	Fine Tuning	48.61 / 47.74	48.87 / 49.99	69.39 / 60.86	56.19 / 52.02	55.71 / 52.65
	From Scratch	80.87 / 80.76	68.83 / 69.18	82.82 / 82.72	66.06 / 67.15	74.70 / 74.95

Table 7: AP / AUC scores (in %) for different models and training methods, tested on DDIM, DDPM, PNDM, and LDM datasets (training set: ProGAN and CelebA-HQ-img).

Appendix: Boxplots

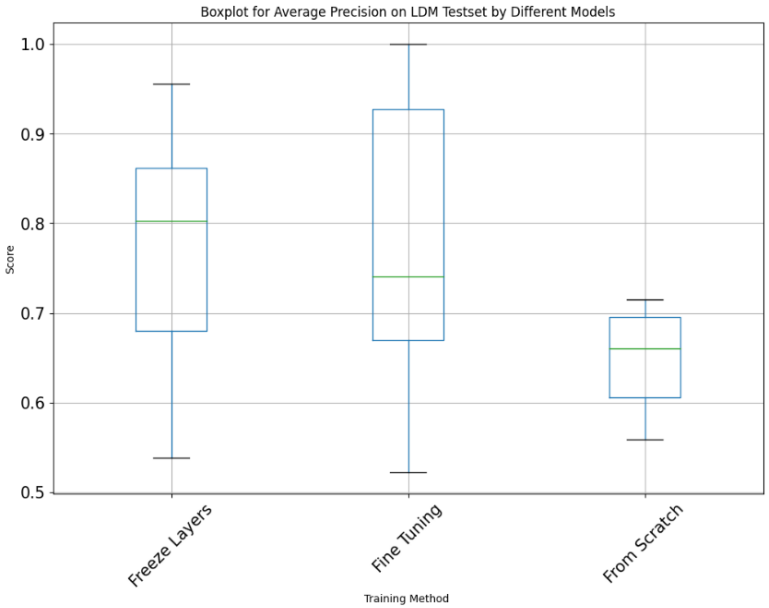


Figure 9: Boxplot for AP (y-axis) on LDM testset obtained by different models for different training modes (x-axis) (training data: ProGAN, Celeb-HQ)

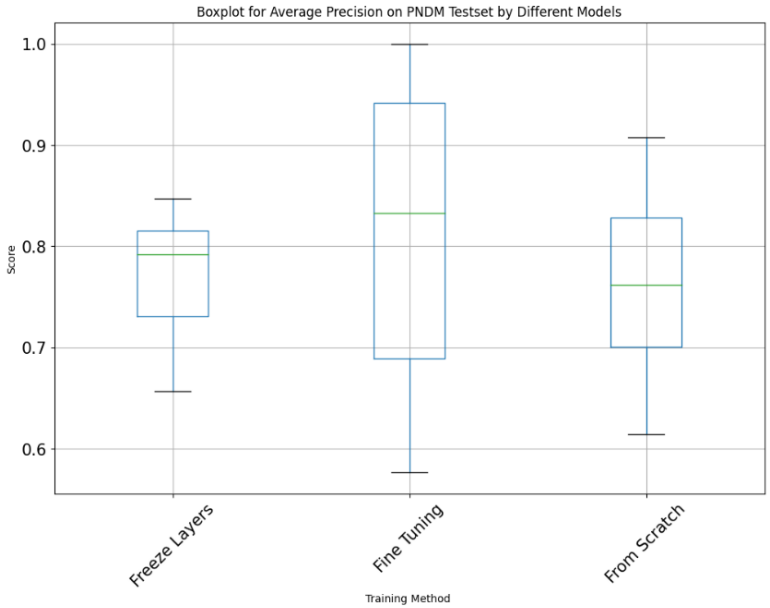


Figure 10: Boxplot for AP (y-axis) on PNDM testset obtained by different models for different training modes (x-axis) (training data: ProGAN, Celeb-HQ)

Appendix: Boxplots

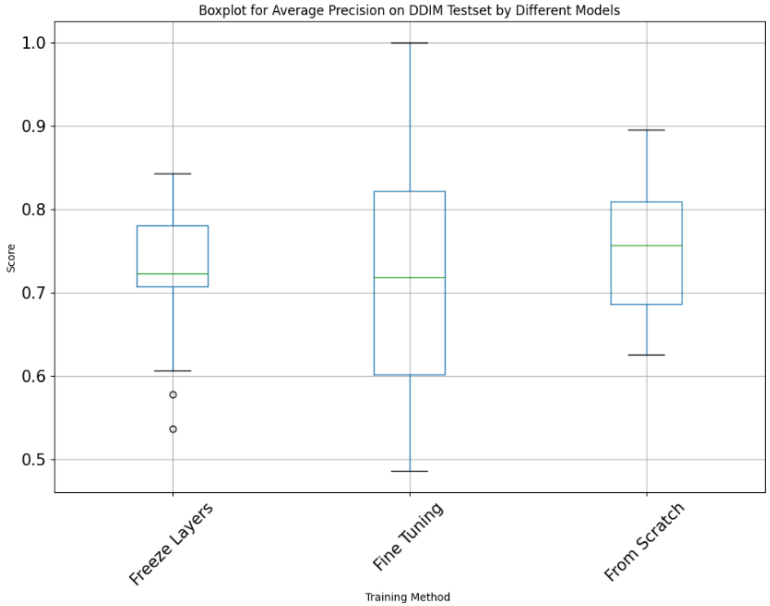


Figure 11: Boxplot for AP (y-axis) on DDIM testset obtained by different models for different training modes (x-axis) (training data: ProGAN, Celeb-HQ)

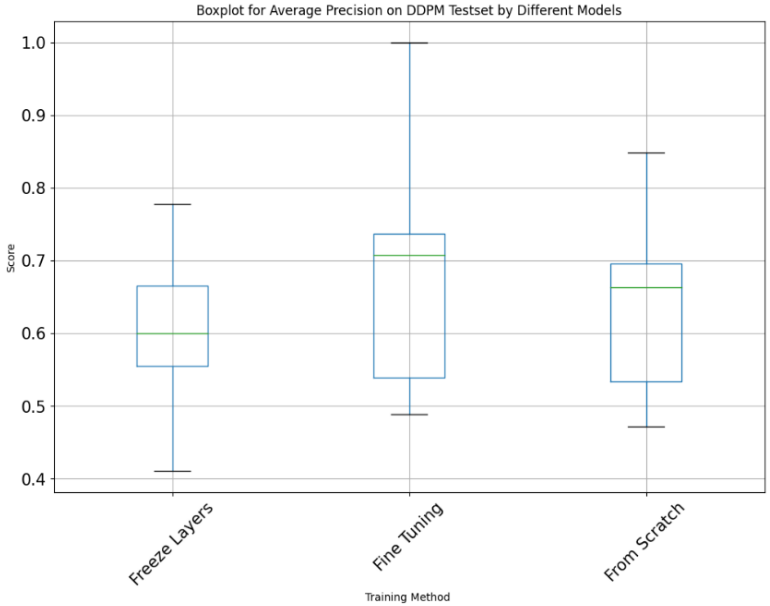


Figure 12: Boxplot for AP (y-axis) on DDPM testset obtained by different models for different training modes (x-axis) (training data: ProGAN, Celeb-HQ)

Appendix: Compression

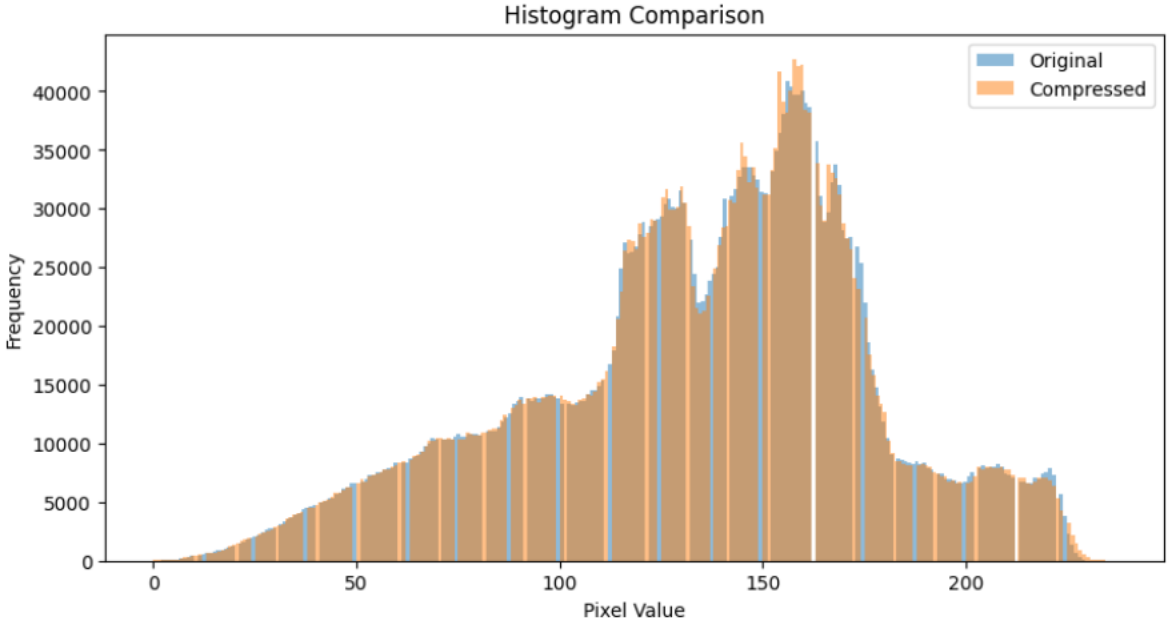


Figure 6: Histogram that shows the absence of certain values of pixels when compressing a synthesized image from StyleGAN2.

Appendix: Block Designs

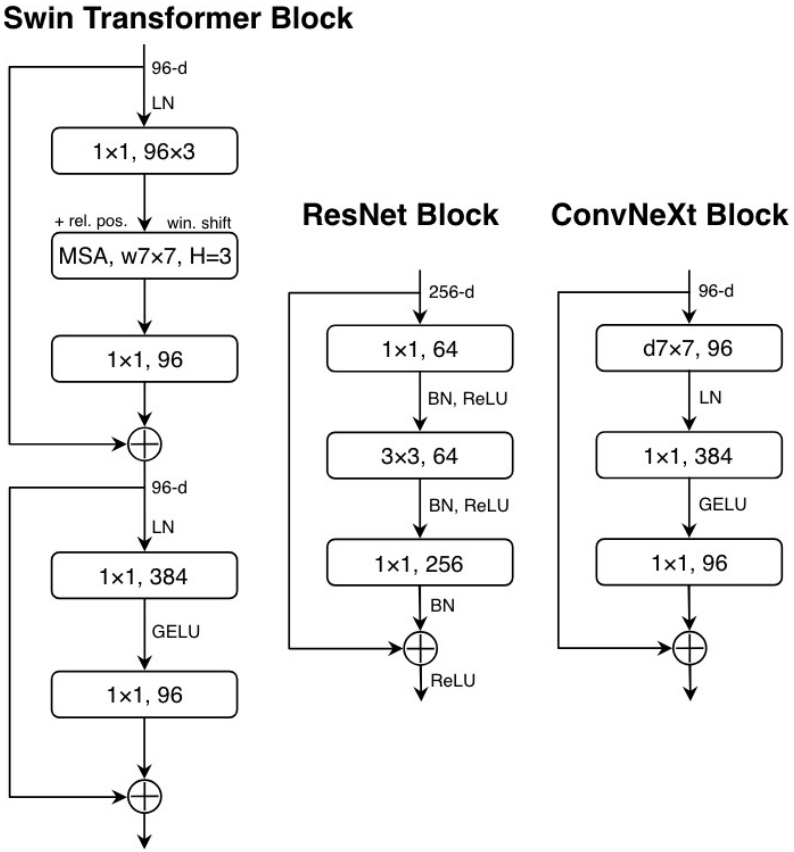
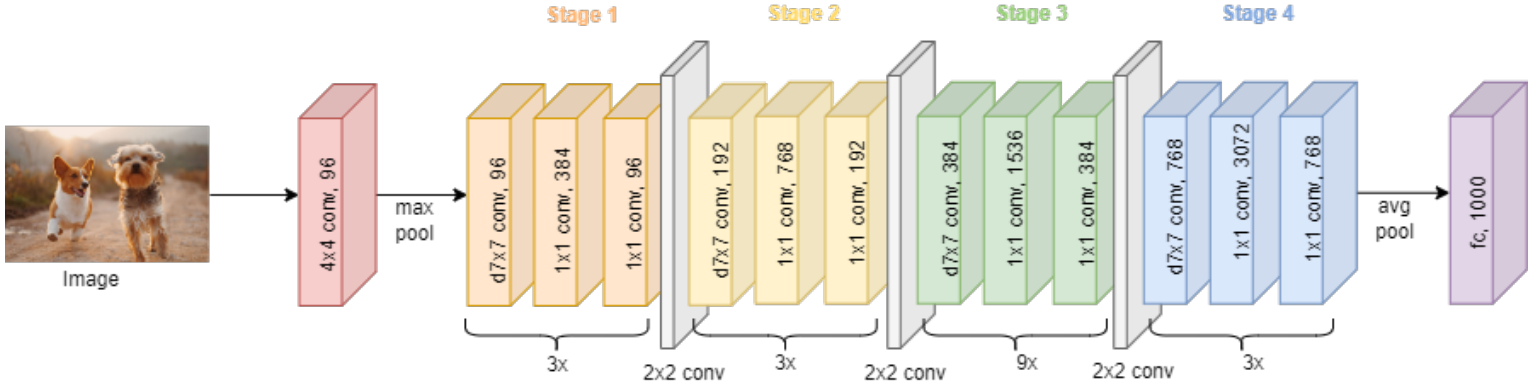
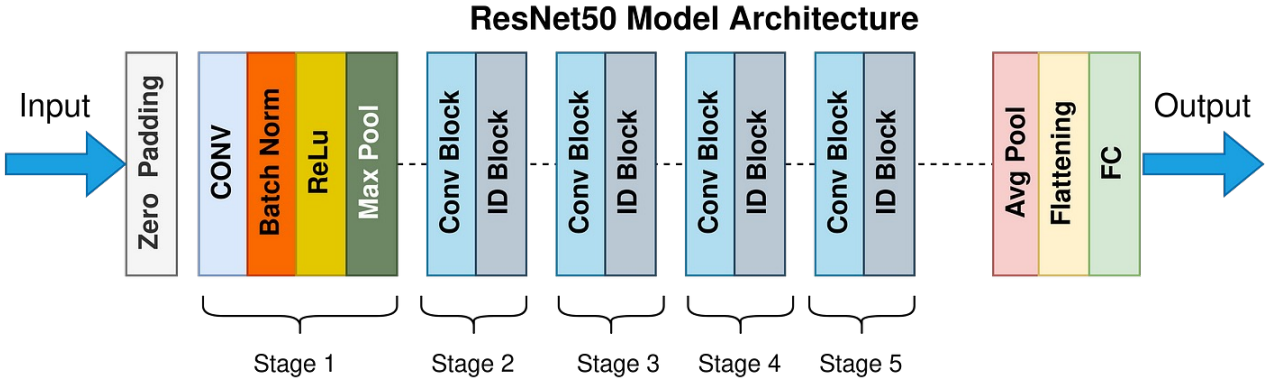


Figure 4. **Block designs** for a ResNet, a Swin Transformer, and a ConvNeXt. Swin Transformer's block is more sophisticated due to the presence of multiple specialized modules and two residual connections. For simplicity, we note the linear layers in Transformer MLP blocks also as "1×1 convs" since they are equivalent.

Taken from [14]

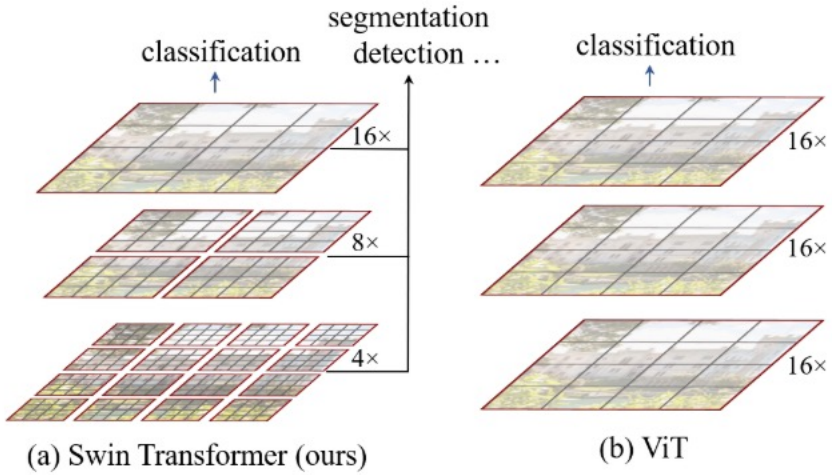
Appendix: ConvNeXt vs ResNet50





Appendix: Swin Transformer

Taken from [13]



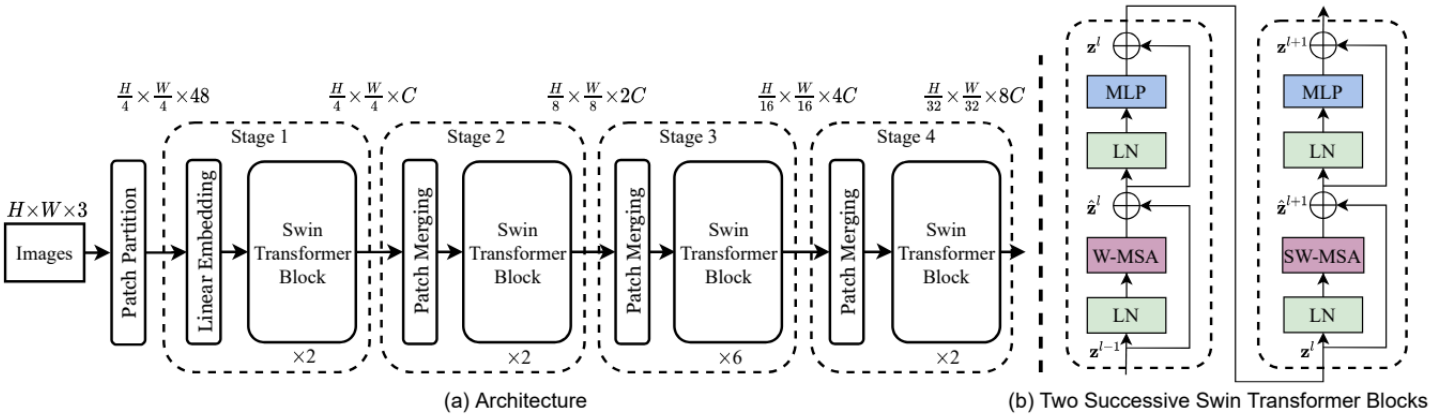


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Appendix

Training Dataset	Testing Dataset							Average
	ProGAN	StyleGAN	VQGAN	DDIM	DDPM	PNDM	LDM	
Deepfakes	100.00	96.99	100.00	87.53	83.21	94.60	98.65	94.14
Face2Face	65.85	64.68	65.85	52.43	55.34	50.51	55.65	58.04
FaceSwap	48.37	47.14	48.37	72.36	64.79	69.05	63.49	59.37
NeuralTextures	59.15	51.45	59.15	44.62	41.74	47.63	55.17	51.13
FFpp1, FFpp2, FFpp3	49.71	44.28	49.71	76.50	74.94	75.08	59.45	61.38
Voting	59.61	58.60	59.61	68.35	62.60	64.41	60.99	62.02

Table 11: AP (in %) for swin tiny trained trained on different datasets and tested on various GAN and DM datasets

Training Dataset	Testing Dataset				Average
	Deepfakes	Face2Face	FaceSwap	NeuralTextures	
Deepfakes	99.79	71.24	41.57	74.95	71.39
Face2Face	78.86	99.70	50.11	62.91	72.40
FaceSwap	63.06	75.21	99.84	51.36	72.37
NeuralTextures	81.82	63.99	44.55	98.39	72.19
FFpp1, FFpp2, FFpp3	99.68	99.73	99.48	69.39	92.07
Voting	99.40	99.21	99.00	66.37	91.00

Table 12: AP (in %) for swin tiny trained on different datasets and tested on FaceForensics++ datasets