AI Generated Image Detection: A Deep Learning Approach

Wenqi Yu, Alden Hegel Tanuwijaya, Xingyu Liu, Zhi Liu

The Chinese University of Hong Kong, Shenzhen

DDA4210 Advanced Machine Learning

April 23, 2024

Contents

Introduction

Among Us: Who is the "Imposter" ?

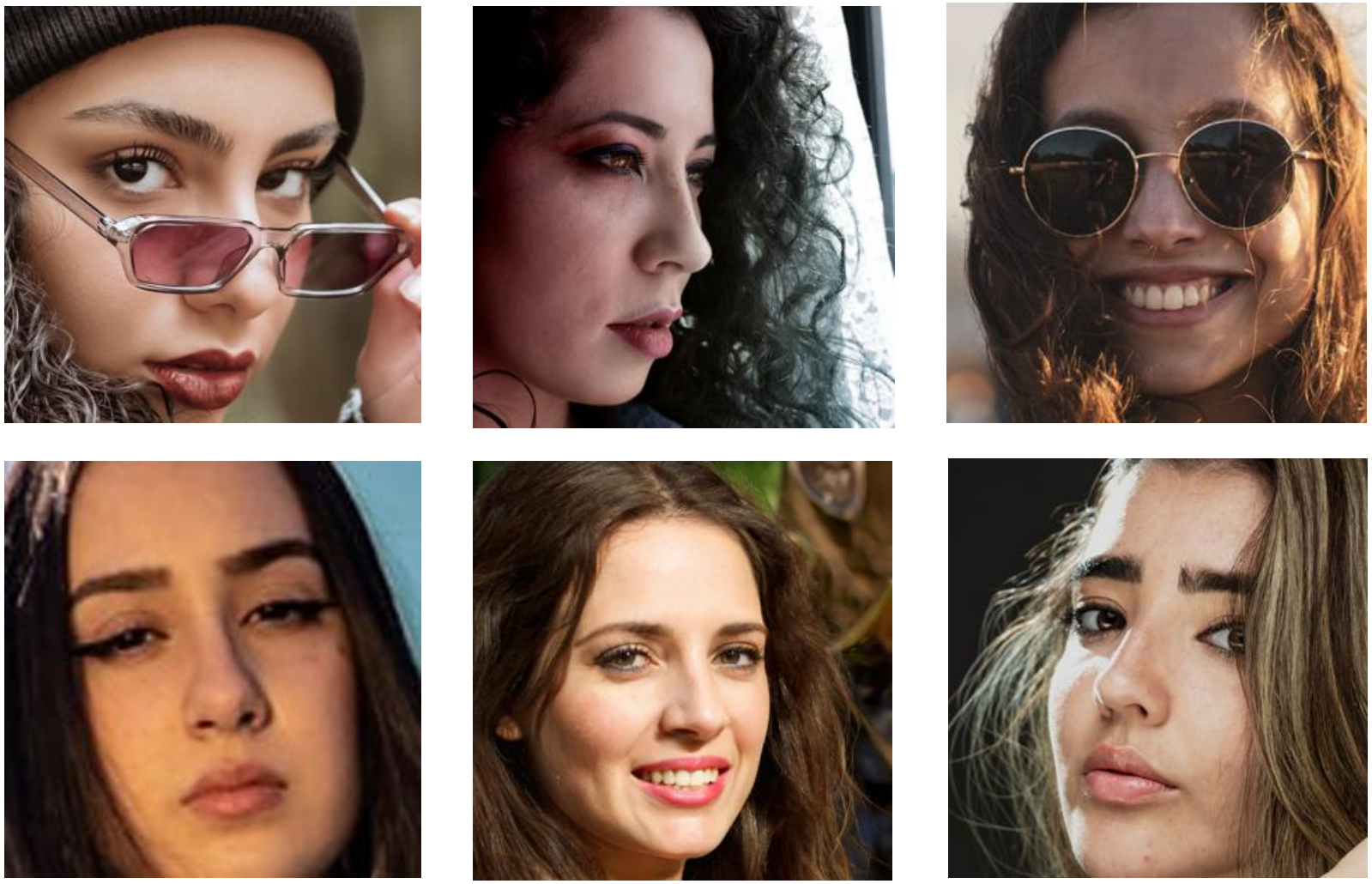


Figure: real or fake ?

Motivation & Significance

Significance:

- **Image authenticity detection** holds significant importance in contemporary society. Media manipulation and dissemination of misinformation have become serious issues.
- **Trustworthy** visual content is crucial for information dissemination and decision-making across various industries, especially in journalism, advertising, and medical imaging.

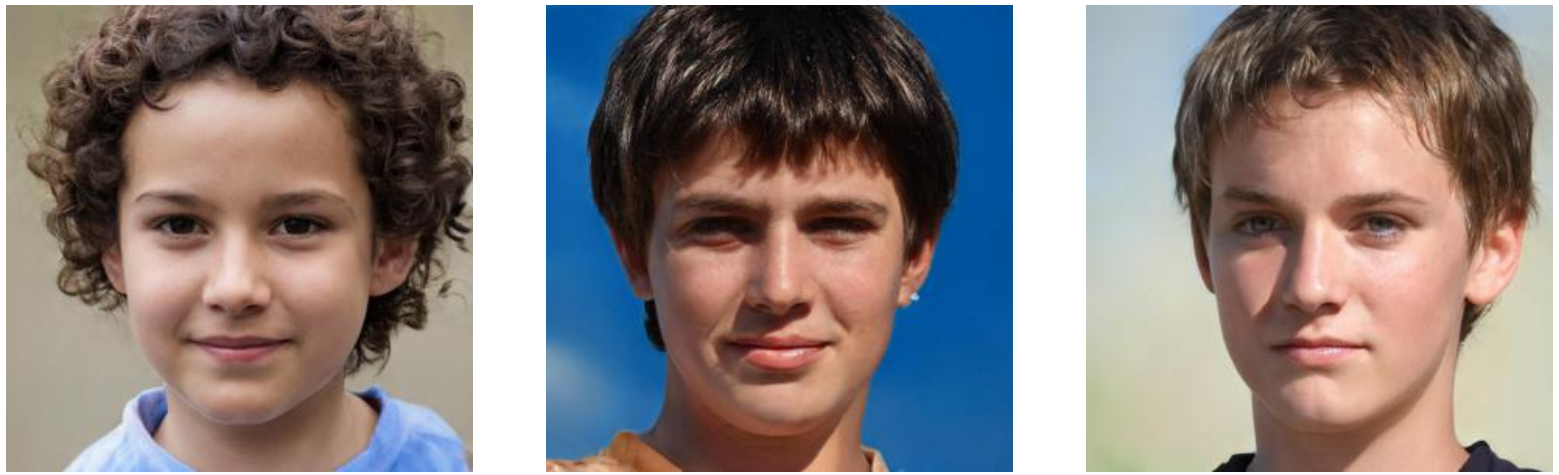


Figure: Sample of AI generated images

Motivation & Significance

Misuse of AI generated images

- Threat of misinformation
- Privacy infringement
- Ethical issues

What we aim to explore

- Variation in image generators
- Potential ways to improve detection performance
- Explainable AI: let our model tells you everything

Figure: Sample of AI generated images

AI Image Generation

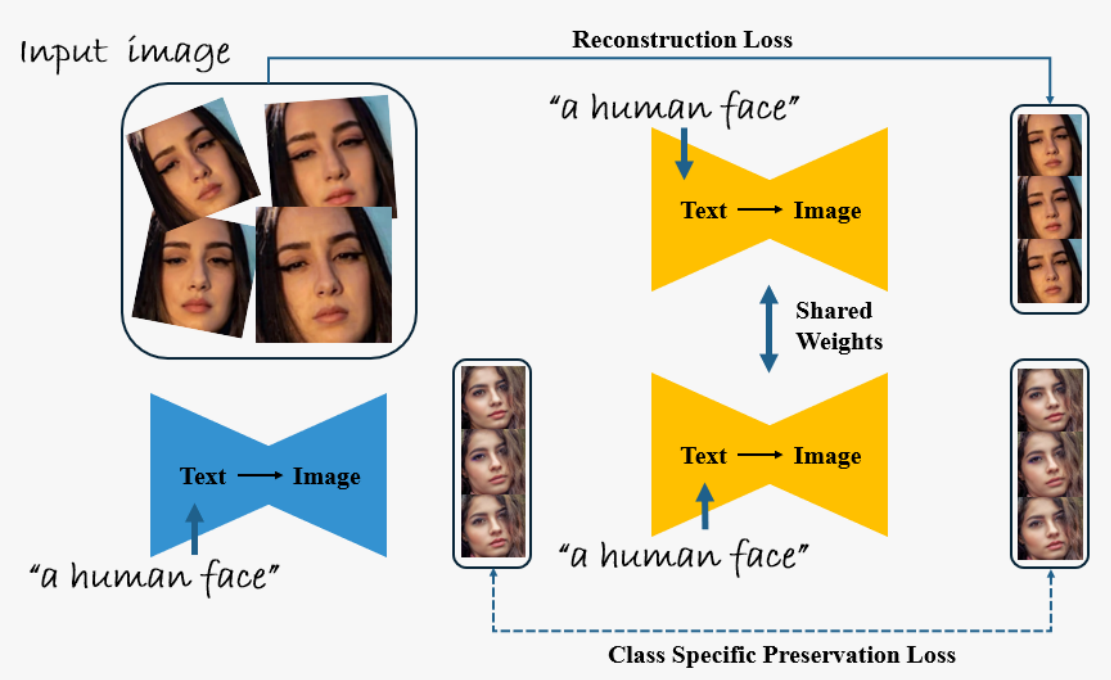
DreamBooth

- GAN based image generator
- Interchangeably minimizing loss of generator and discriminator

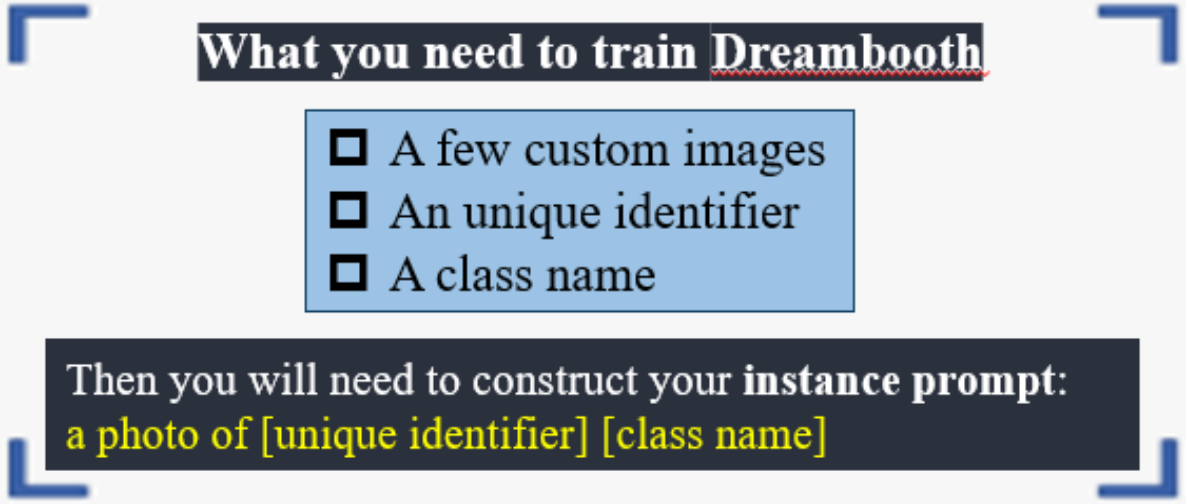
$$L_G = -\mathbb{E}_{z\sim p(z)}[\log D(G(z))]$$

$$L_D = -\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] - \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))]$$

- Features:
  - Custom model training capabilities
  - Powerful AI text prompts
  - Guidance scales
  - Inference steps



DreamBooth



Our instance_prompt:

- A photo of NewFace humanface

Our class_prompt:

- A photo of a humanface

Train steps:

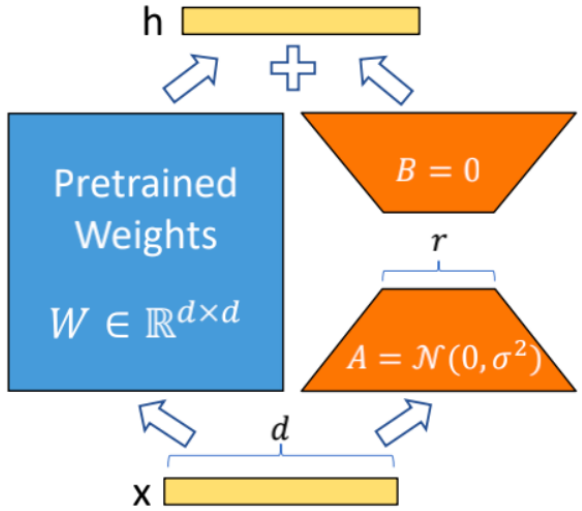
- 800

How we use DreamBooth

Step 1: *Prepare training images*
Step 2: *Resize images to 512×512*
Step 3: *Utilize LoRA*
Step 4: *Introduce negative prompt*
Step 5: *Train the model*

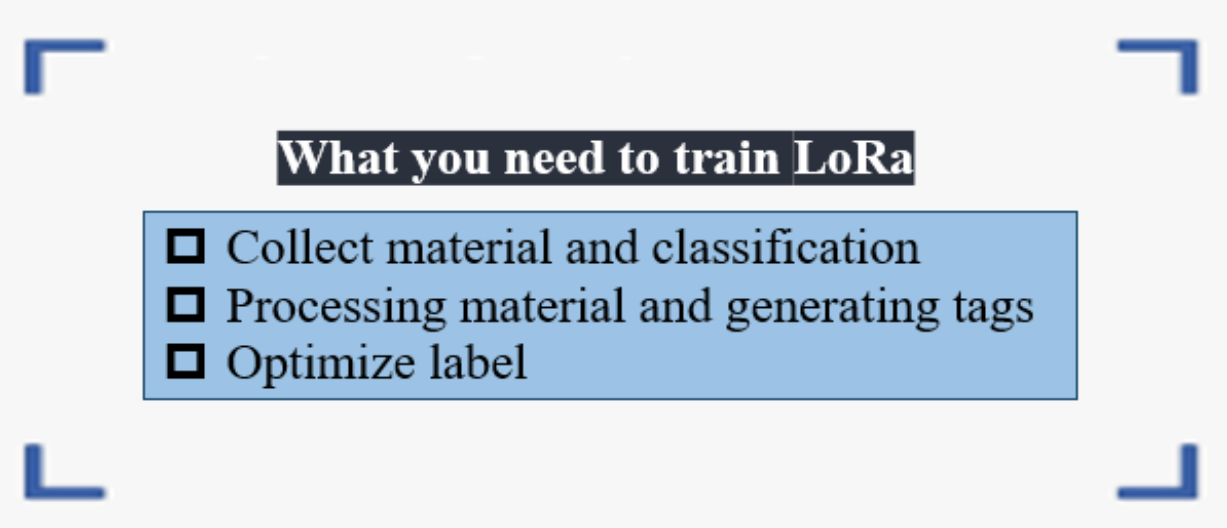
LoRA

- Minor-dimensional fine-tuning method
- More amount of pictures required
- Latent overwrite
  - explore a broader space of latent representations during training
  - better coverage of the data distribution
- Regularization Adversarial Training
  - e.g. spectral normalization & weight clipping



1.Model Enhancement: *Auxiliary structures A and B next to the pre-trained model*
2.Dimension Matching: *A and B match the model's input and output dimensions but are smaller, embodying a low-rank structure that reduces the number of trainable parameters.*
3.Training and Inference: *Only A and B are updated during training, with the original parameters fixed. During inference, reparametrization combines A and B with the original weights to avoid extra computations.*
4.Adaptability: *Quick adaptation to various tasks is achieved by retraining only A and B, speeding up the training process for large models.*

LoRA



- Repeat: 10
- Epoch: 100
- Train steps: 800

How we use LoRa

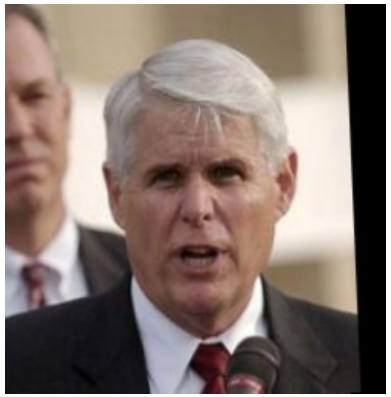
Step 1: *Collect pictures and determine categories*
Step 2: *Material processing and generating labels*
Step 3: *Optimized label*
Step 4: *Adjust the parameters and start training*
Step 4: *Save parameters, test feedback*

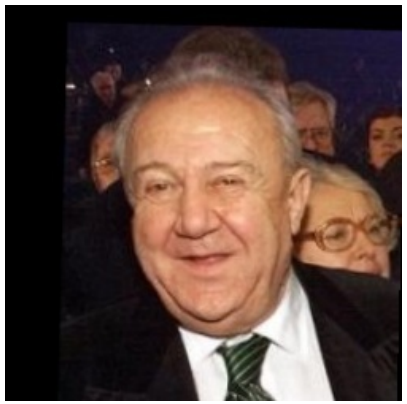
Dataset Overview

- Manually Annotated Human Face Dataset from CelebA (500 images)




- LFW-Funneled Dataset (352 images)



Data Processing

Figure: dataset

Dataset collection

I. Real face dataset is obtained.
II. Fake face images are generated using DreamBooth and LoRA models..

Preprocessing Images

I. Reading and Resizing : Utilize OpenCV to read and resize images to desired dimensions. Uniform dimensions ensure compatibility with neural network input.
II. Normalization : Normalize pixel values to a range between 0 and 1. Stabilizes and accelerates training process by maintaining consistent data range.
III. Function Implementation : Develop a function for image preprocessing tasks.

Dataset Splitting

I. Purpose : Prevents overfitting by evaluating model on unseen data.
II. Split Ratio : Divide dataset into training (80%), testing (10%), and validation (10%) sets.
III. Integrity Maintenance : Shuffle samples and maintain class ratio during splitting. Use random state for reproducibility.
IV. Stratification : Ensure distribution of target classes in training and testing sets accurately represents overall population.

Model Construction

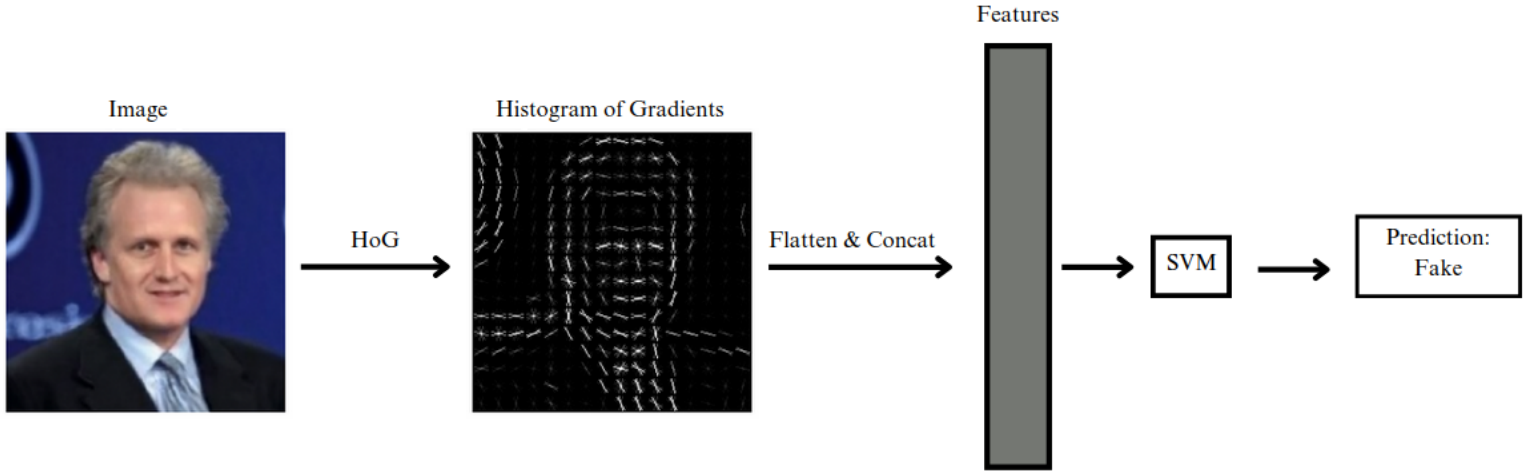
HoG + SVM

HoG

- Capture the local gradient information of the image
- Divide image into small cells
- Compute gradient magnitude and orientation for each cell
- Construct a histogram and concatenate it into the final vector

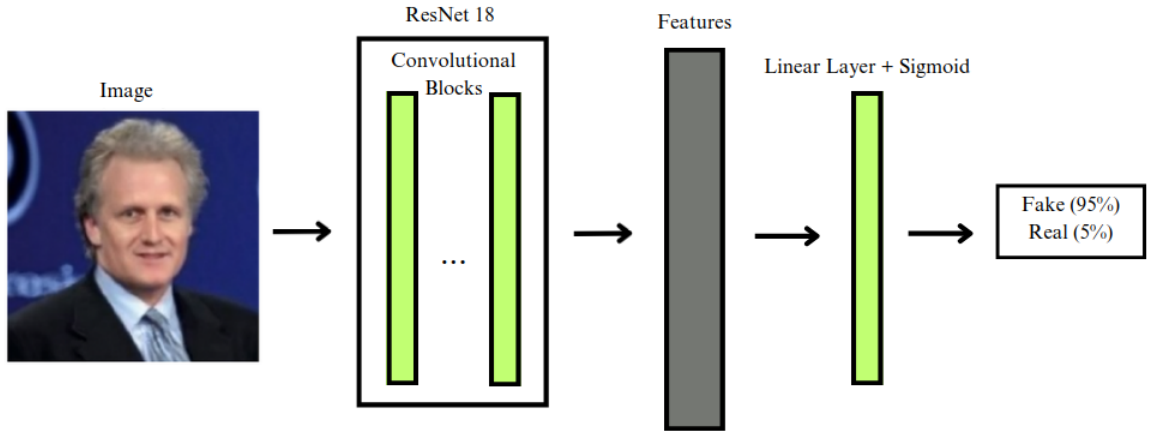
SVM

- Grayscale 128 x 128 pixels
- Linear kernel, classification

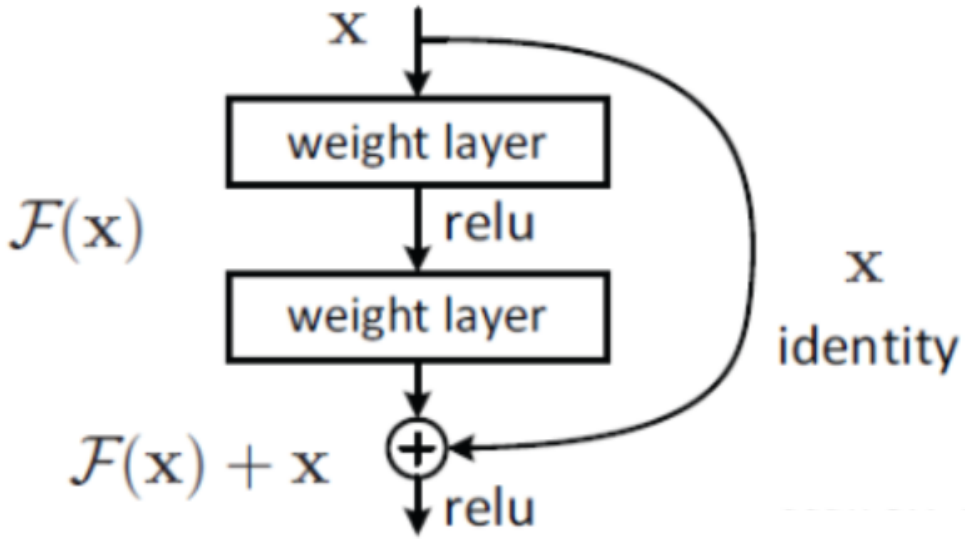


ResNet-18

- Effective in handling vanishing gradient and exploding gradient
- Epoch = 15
- Batch size = 32
- Learning rate = 0.0001



$$H(x) = F(x) + x$$

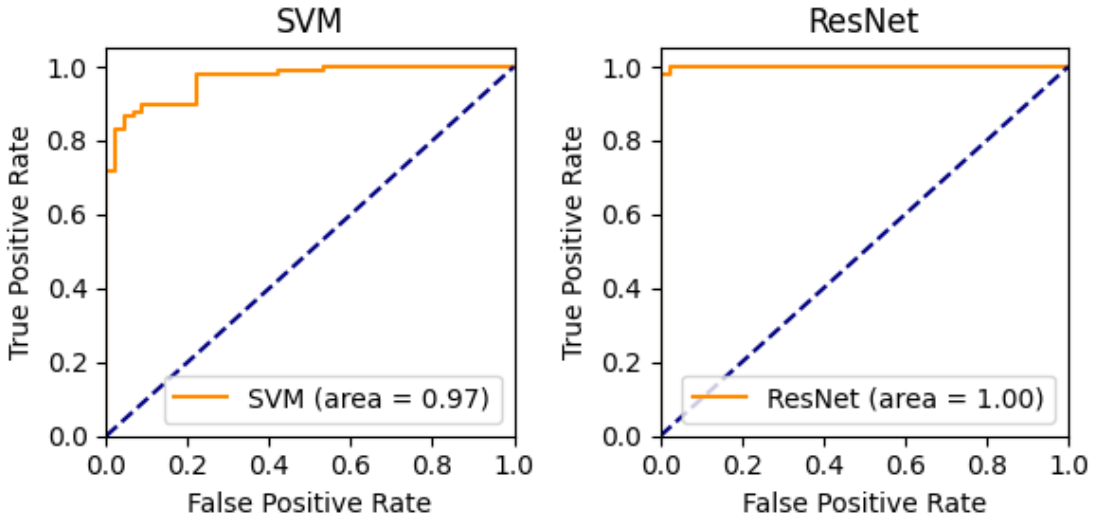


Baseline Models

- **Advantage of DL Models:** ResNet-18 outperforms HoG+SVM in terms of accuracy, precision, recall and F1-score
- **Generalization Concern:**
  - Suffer from overfitting
  - Trained on small and specific datasets
  - Poor generalization of images from other generators

Model Setting	Accuracy	Precision	Recall	F1
HoG+SVM	0.88	0.93	0.90	0.91
ResNet18	0.99	1.00	0.98	0.99

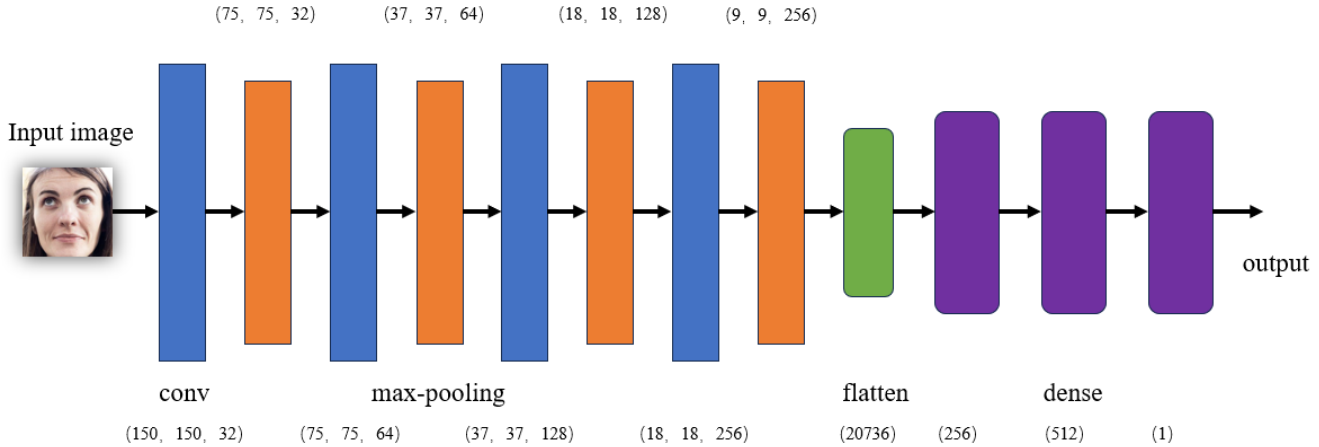
Figure: Comparative result of two models



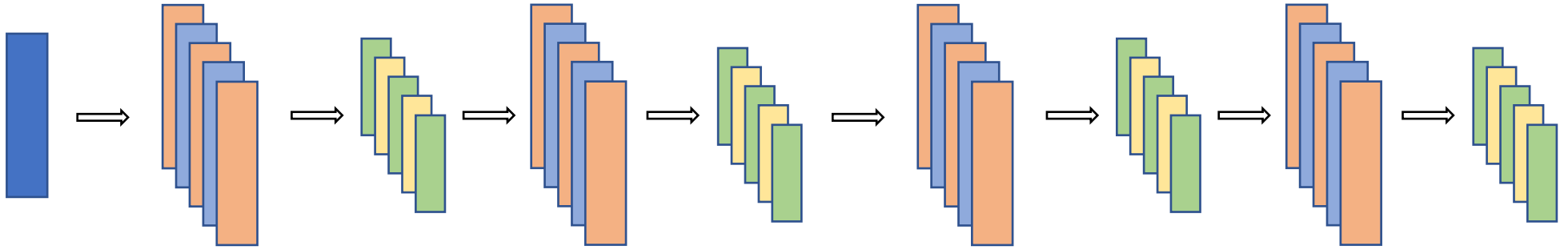
Modified CNN

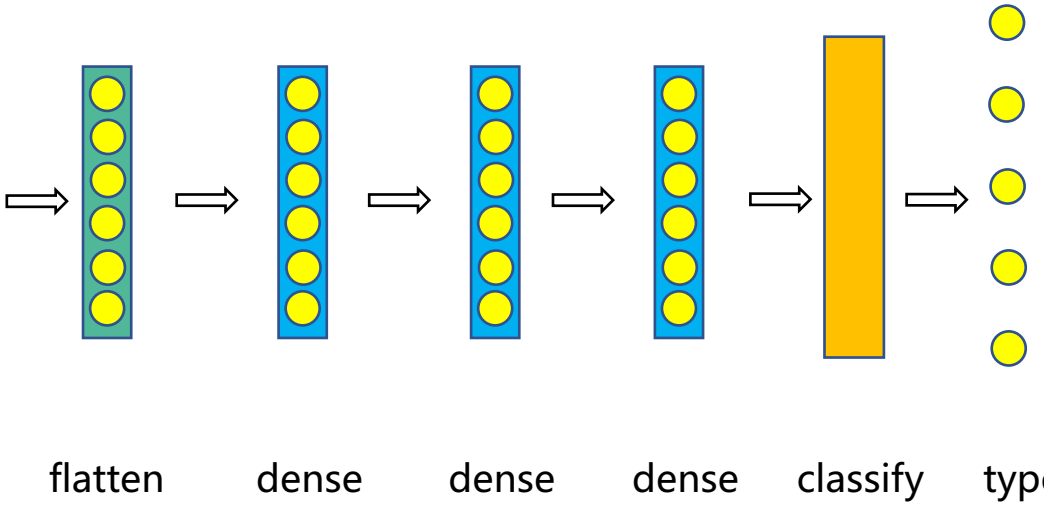
Model Architecture

- **Convolutional Layers:** 5 layers in total. The first layer has 32 filters with a kernel size of 2 by 2. Subsequent layers double the number of filters, and the kernel size is incremented by one
- **Max-Pooling Layers:** After each convolutional layer, reduce overfitting and computational costs
- **Dense Layers:** The output is flattened and passed to 4 dense layers. The first dense layer has 256 neurons, and the number of neurons is halved in the next two dense layers
- **Dropout Layers:** Randomly ignore some neurons and prevent overfitting
- **Output Layer:** Two neurons with softmax activation, one for each class (real or fake)



Modified CNN





Model Parameters

- Number of Convolutional Layers: 4
- Number of Filters in First Convolutional Layer: 32
- Kernel Size in First Convolutional Layer: (2,2)
- Number of Filters in First Dense Layer: 256

Modified CNN

Mathematical Derivation: $$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log P(y=1|x_i) + (1-y_i)\log(1-P(y=1|x_i))]$$

Our idea for performance improvement:

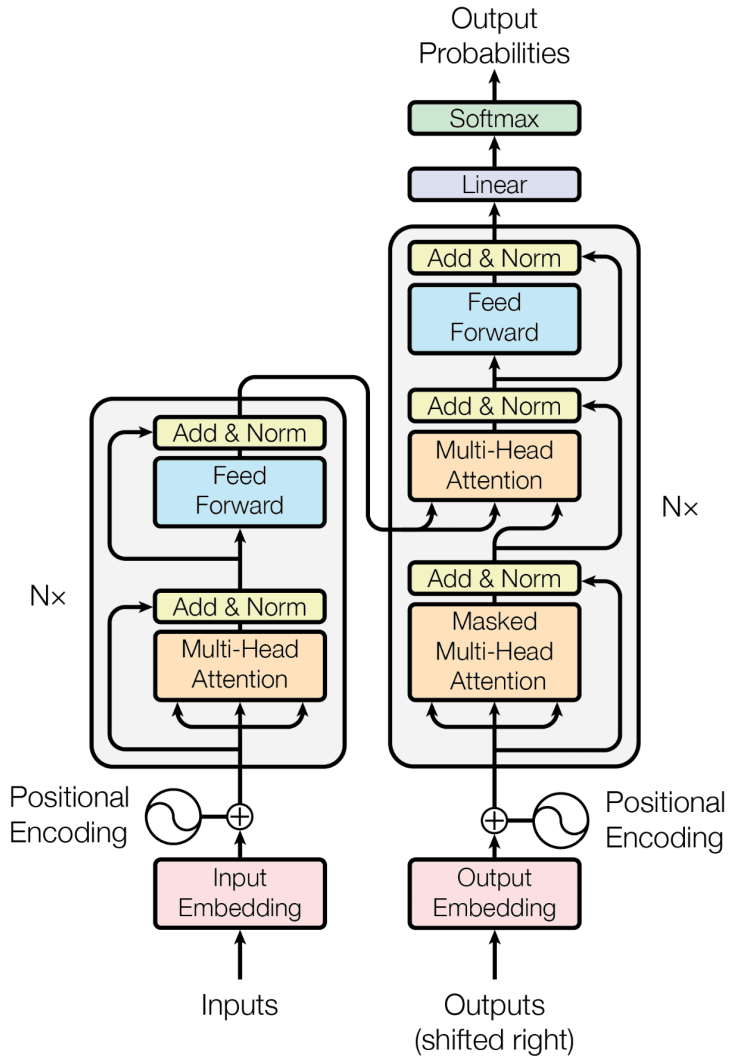
- Dropout Layers:
  - Randomly discarding part of the output
  - Reduce the dependence of weight update
- Data Augmentation
  - Introduces variations in training data (e.g., rotation, scaling, flipping)
  - enhance model robustness and generalization
- Early Stopping
  - Monitors model performance on a separate validation set and stops training when performance plateaus
  - prevent overfitting and save computational resources

Figure: fake images

Figure: real images

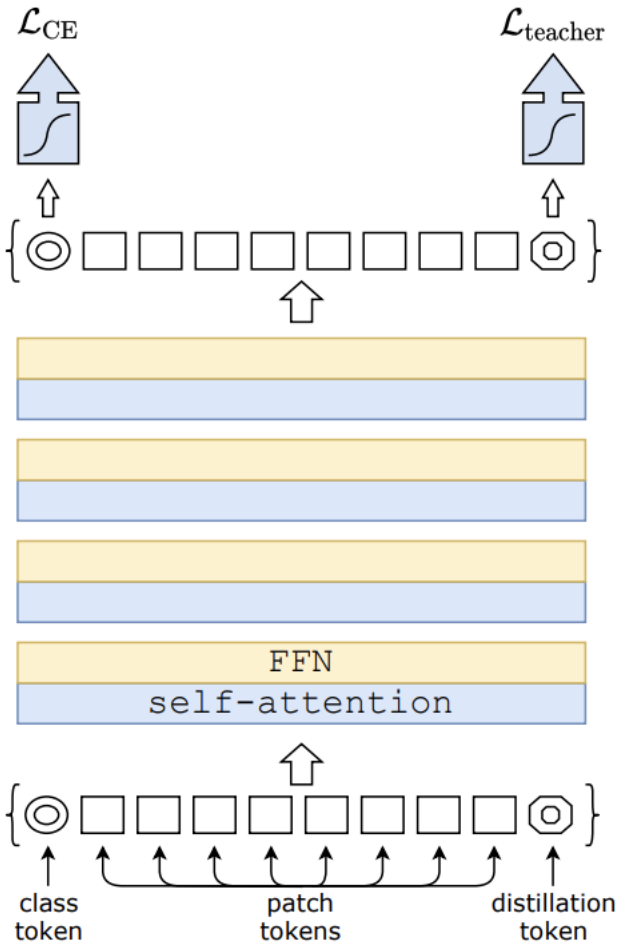
Transformer-based

- Initially used for NLP:
  - Key features:
    - Self-attention mechanism
    - Allowed the model to focus on different parts of a sequence simultaneously.
    - Multi-Head Attention allows multiple attention computations in parallel, providing a broader perspective.
    - Highly efficient due to parallelism.



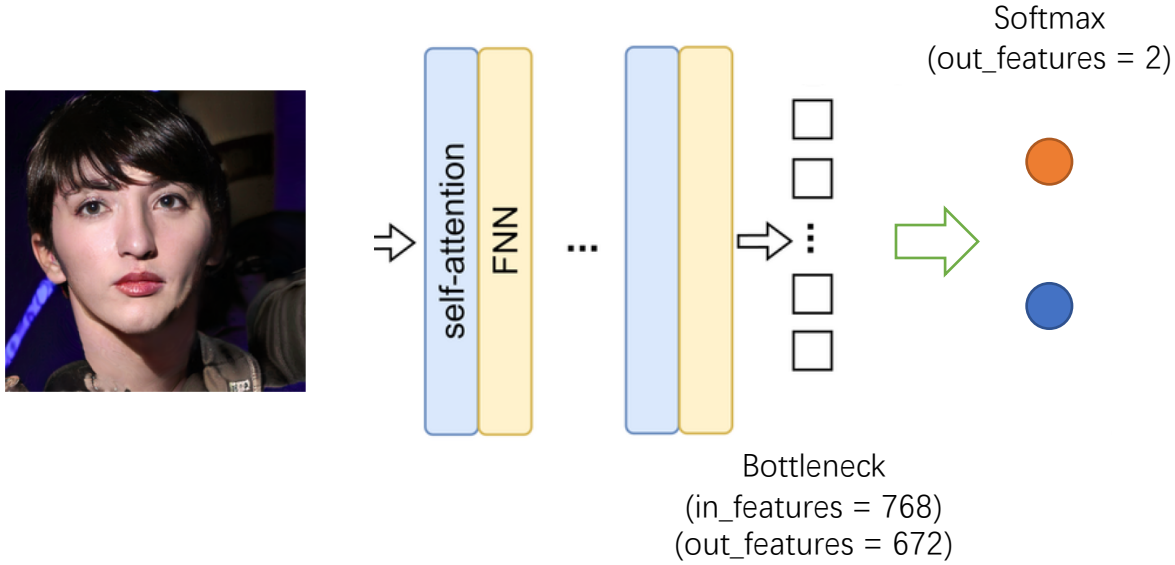
ViT and DeiT

- (ViT) Dosovitskiy et al. in 2020
  - First implementation in computer vision
  - Competitive performance
  - Resource-heavy
- (DeiT) Facebook AI Research 2021
  - Teacher-student strategy
  - Data and resource efficient



Cross-generator Generalization

- Modified DeiT_224
  - Eliminating top classification layer
  - Adding " bottleneck" layer
  - Final layer for binary classification
- Training
  - Resize image into 224 x 224
  - Batch size = 32
  - Learning rate = 0.001
  - Early stopping (max epochs = 50)



Cross-generator Generalization

- Training Data
  - StyleGAN and StyleGAN2
  - Adversarial generation
- Test Data
  - DreamBooth and LoRA
  - Prompt-based generation

StyleGAN

StyleGAN2

DreamBooth

LoRA

Results & Analysis

Analysis & Comparison

Type	precision	recall	F1-score	support
DreamBooth	0.87	0.89	0.88	800
LoRA	0.89	0.87	0.88	800

Result 1: According to the model established above, identify the test set, please refer to the data right figure, It can be seen that the detection effect of this model is still good.

Result 2: Make predictions on the testing data, evaluate its performance, and compute its accuracy, precision, recall, and F1-score using the confusion matrix, It can be seen that the detection model performs similar in Dreambooth and LORA's picture generation algorithm, and the effect is not bad.

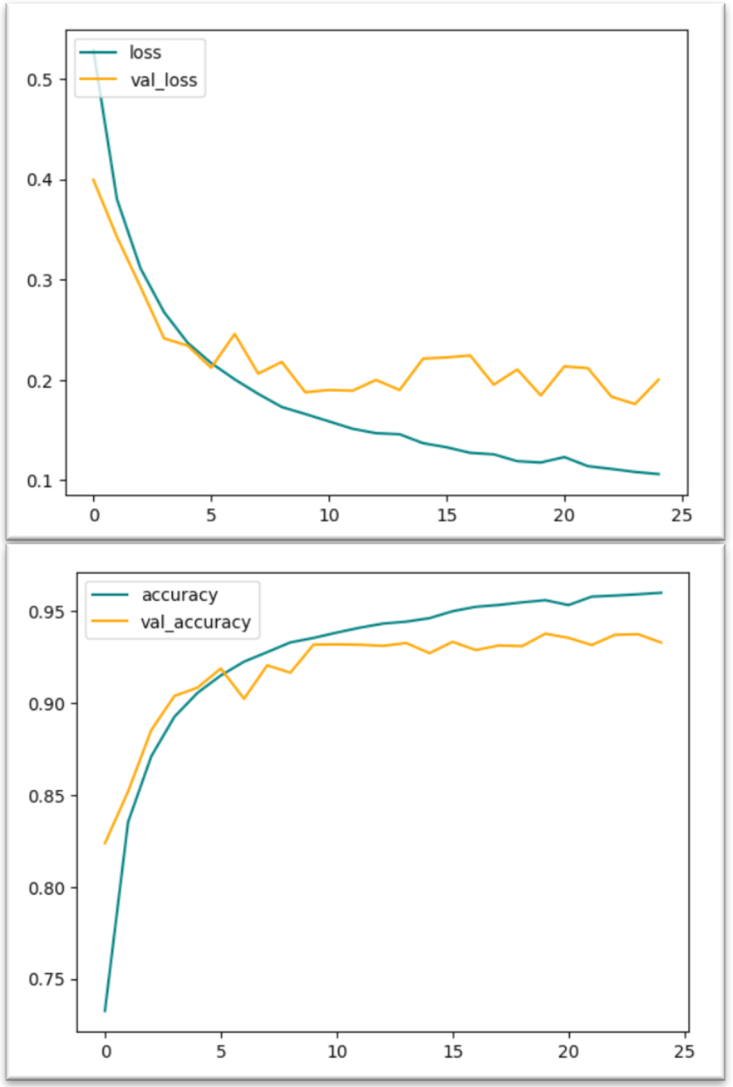


Figure: loss & accuracy of modified CNN model

Modified DeiT

Type	Accuracy	Precision	Recall	F1
Foreign	0.92	0.83	0.99	0.90
Self	0.97	0.99	0.96	0.97

Promising generalization capability: There's a drop in accuracy, but we still manage a score of > 0.90 for cross-generator.

Caveat: Lower precision can be attributed to imbalanced dataset (500 real vs 800 fake, more room for false positives). Nevertheless, there's room for improvement in precision metric.

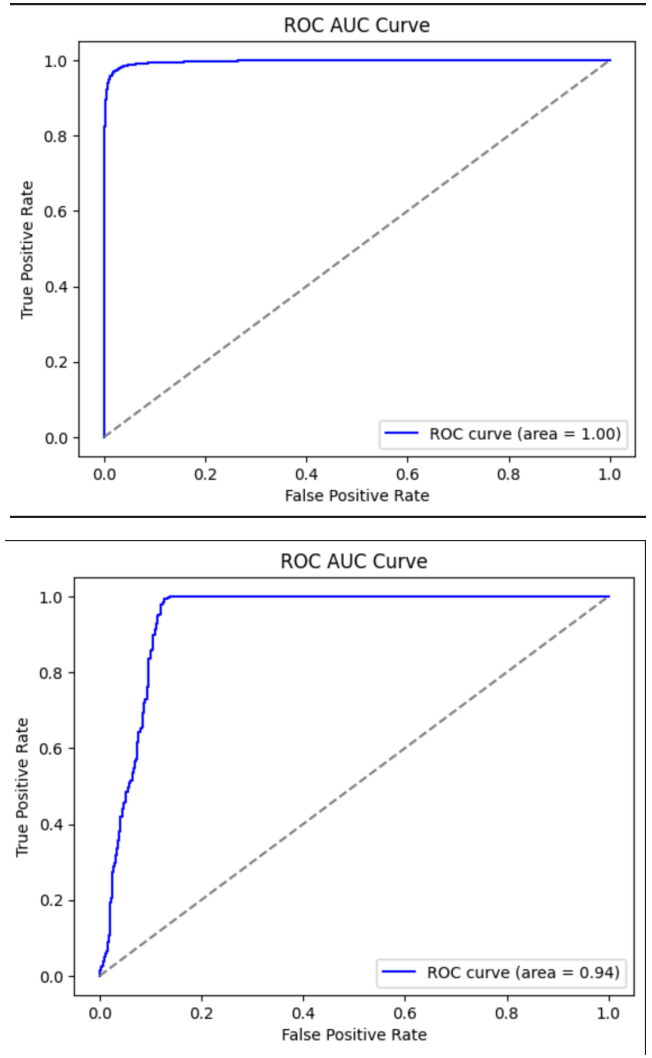


Figure: ROC AUC self and foreign respectively

Gradient Analysis

- Forward image X through network f and produce logit output $\bar{y} = f(X, \theta)$
- Compute the objective function $L = BCE(\bar{y}, y)$
- Compute the gradient $\frac{\partial L}{\partial x}$ of the binary cross entropy loss with respect to input images

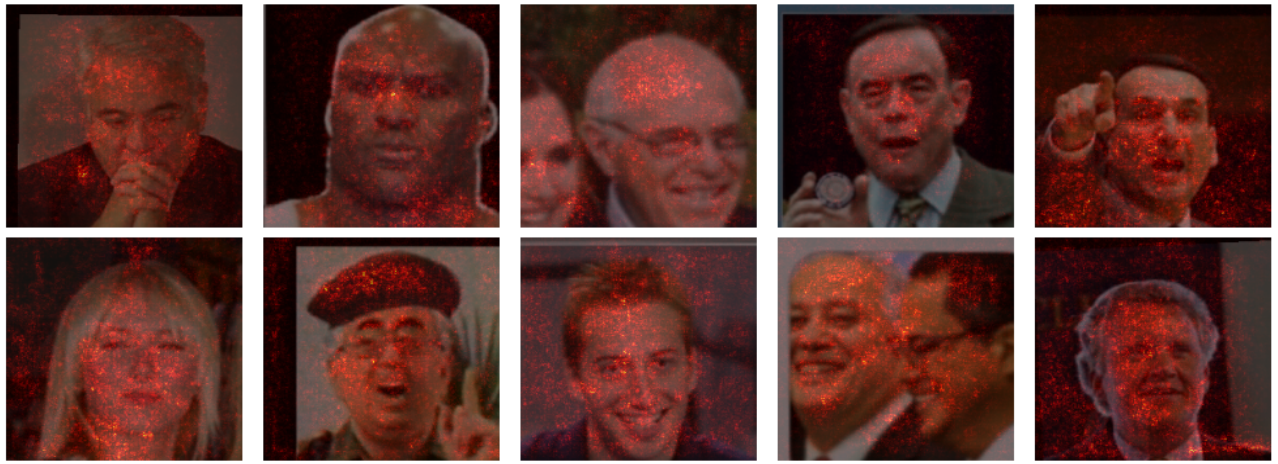


Figure: Gradient distribution of input image

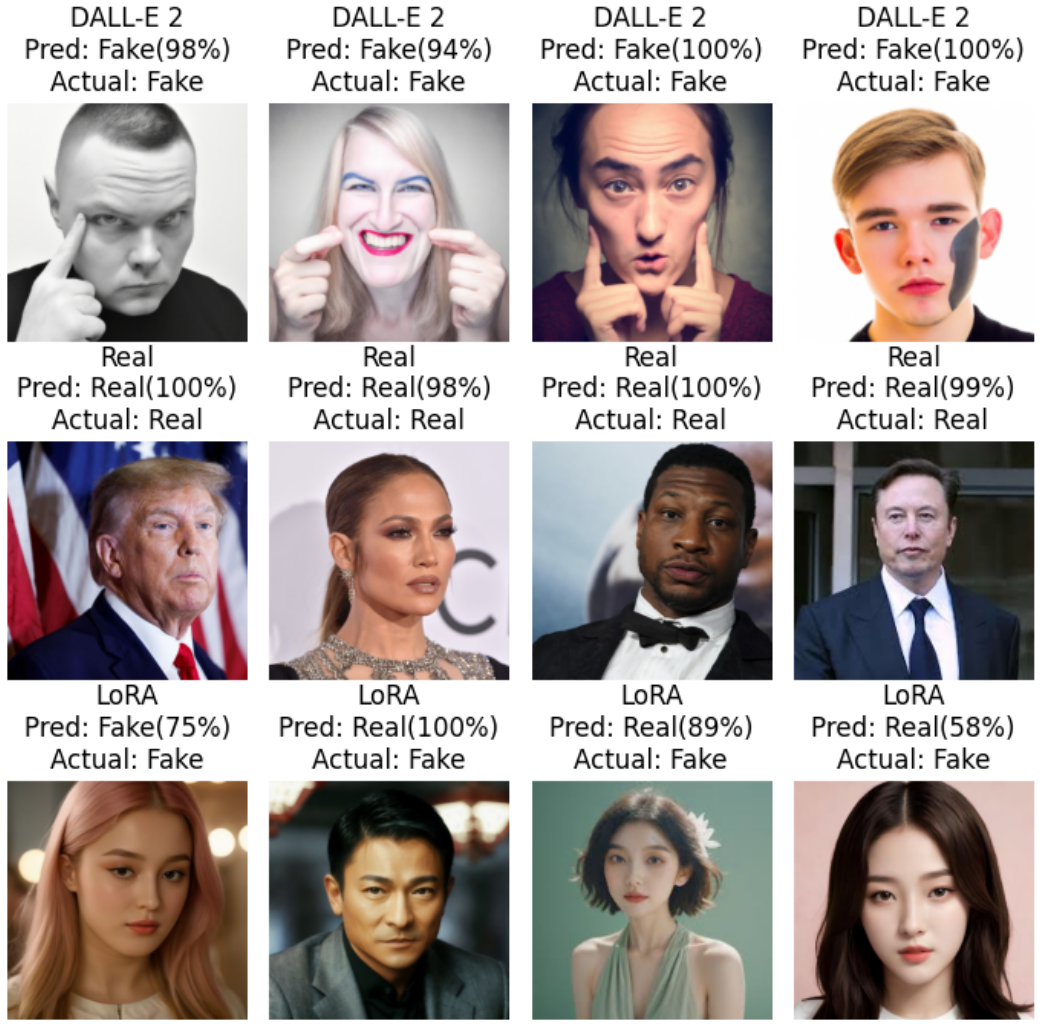
Gradient Analysis

- The model makes decisions mostly based on the skin texture of the faces
- AI generated skin (over-smooth) vs. Real skin (large variation in pixel values)

Figure: generated face vs. real face

Generalization Concern

- Incorporate images generated by DALL-E
- Add images generated by LoRA, but not included in the training data
- Nice generalization ability



Conclusion

Innovation

- Create the fake image dataset using three different generators
- Modify the CNN architecture to improve detection performance
- Non-convolutional model (modified DeiT)
- Cross-generator
- Conduct gradient analysis to improve explainability

Findings

- Learn image features from different generators
- Data matters more than models !
- Potential generalization for newer/unseen generator(s)

Figure: DreamBooth fake

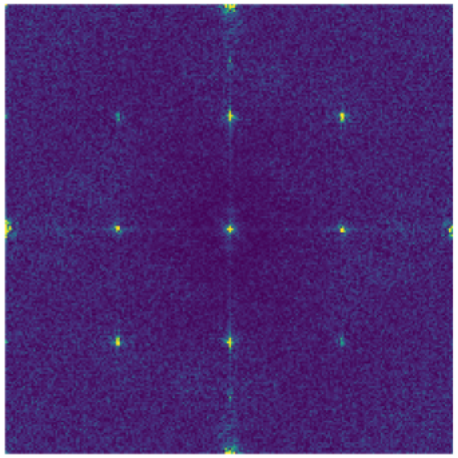
Figure: LoRA fake

Figure: DALL-E fake

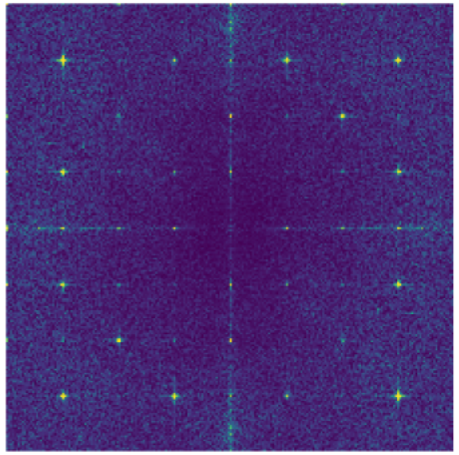
Future Ideas

- With robust XAI (gradient analysis, deep feature factorization, etc.), analyze generator " fingerprint "

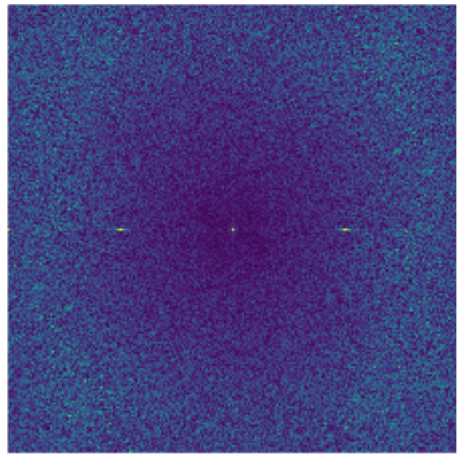
StyleGAN2



Stable Diffusion



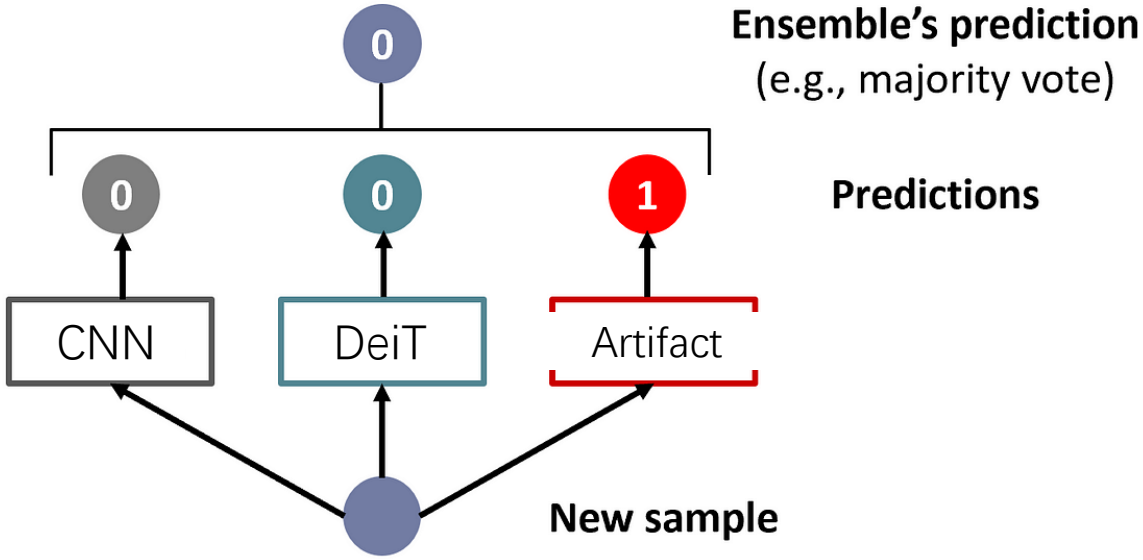
DALL·E 2



Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, Luisa Verdoliva (2022)

Future Ideas

- Ensemble models for robustness (as illustrated in CNN and DeiT dynamic), mitigating individual model weakness and improving overall performance.



Thanks for listening !