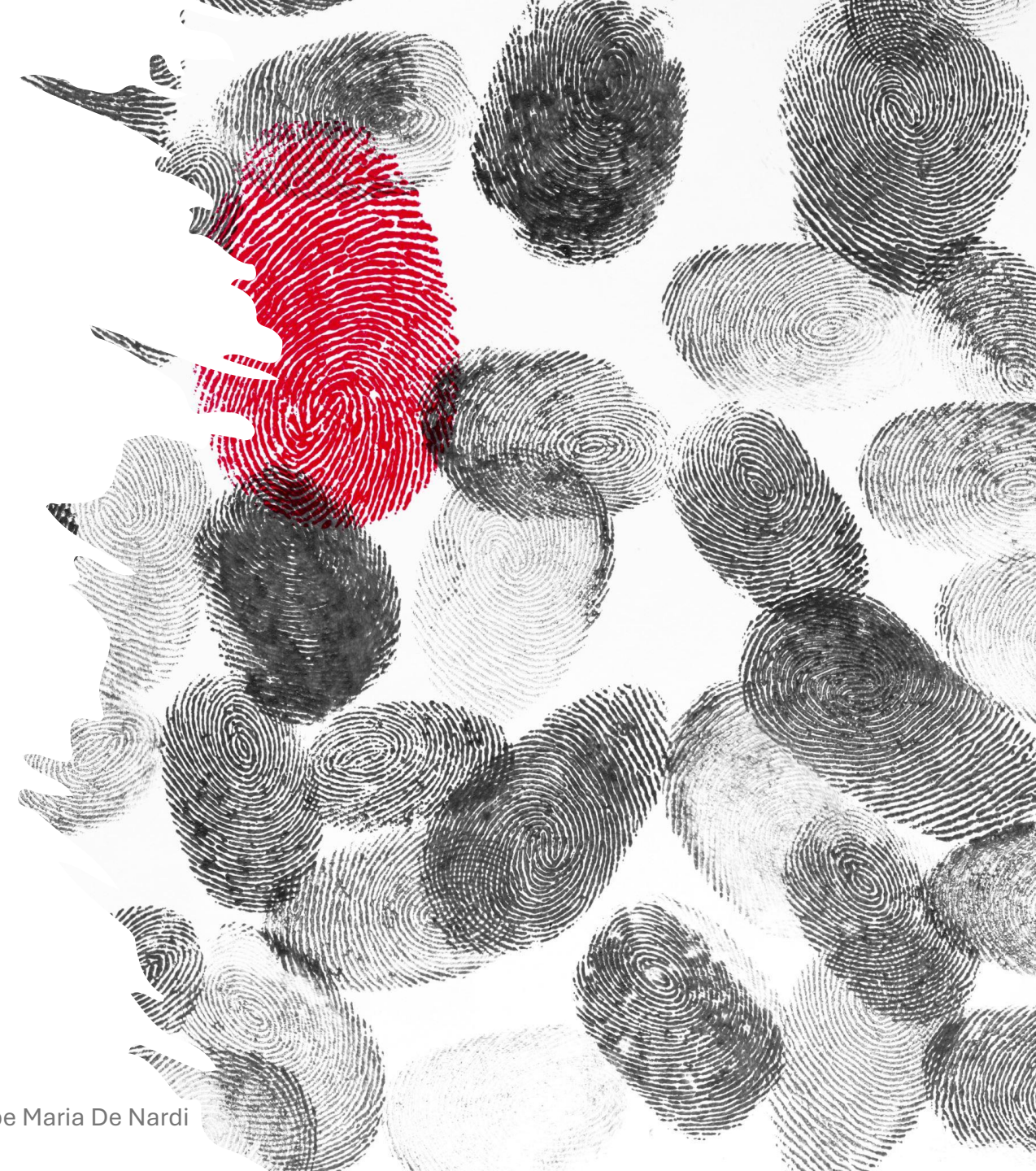
AI Detector

Gianluca Giuseppe Maria De Nardi

Method idea

To identify patterns in AI-generated images, this method focuses on extracting the image's **fingerprint** based on the difference in entropy between low-texture and high-texture regions.



FingerPrint model

Gianluca Giuseppe Maria De Nardi

High-pass filter

The high-pass filter has characteristics that leave the components of the high-frequency signal unchanged and modify the low-frequency ones. The masks implemented and applied to the image are **30** filters with a shape of **5x5** with **1** channel (gray). The rules for a high-pass filter are

- All filter coefficients can be positive or negative.
- The sum of all coefficients must give 0.

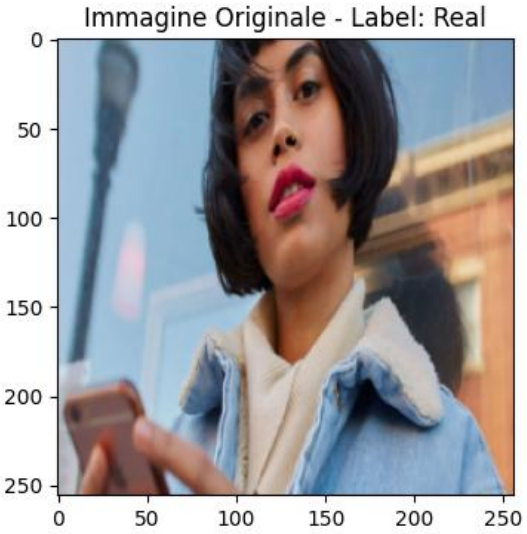
*To get the fingerprint of the image, the **high-pass** filters are **convoluted** to the image made of rich and poor patches.*

The idea is to increase the difference between them and intensify the contrast of the colors that an image generates by AI, emphasizing features such as uneven colors and black patches.

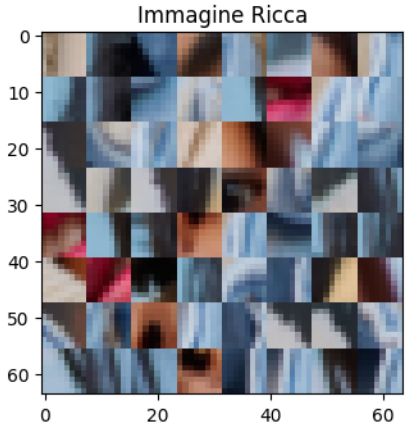
Texture Diversity

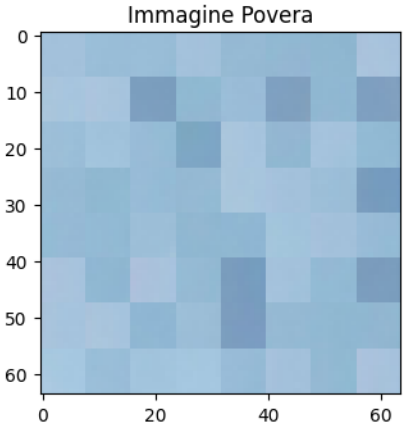
```python
def texture_diversity(patch):
    grayscale_patch = torch.mean(patch, dim=0)
    gradients = [
        torch.abs(grayscale_patch[:-1, :-1] - grayscale_patch[1:, 1:]),
        torch.abs(grayscale_patch[:-1, 1:] - grayscale_patch[1:, :-1]),
        torch.abs(grayscale_patch[:, :-1] - grayscale_patch[:, 1:]),
        torch.abs(grayscale_patch[:-1, :] - grayscale_patch[1:, :])
    ]
    diversity = sum(torch.sum(grad) for grad in gradients)
    return diversity.item()
```

FingerPrint Process

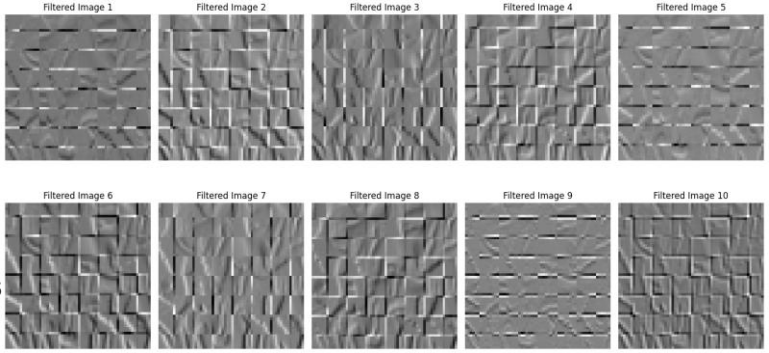


Smashed





High-Pass Filters



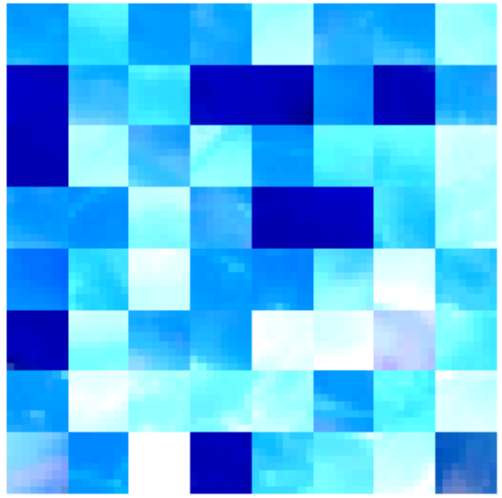
FingerPrint extraction

High-Frequency image

High Frequency image

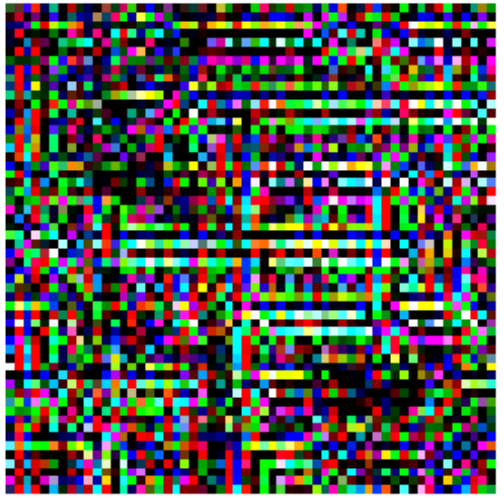
Original Image

Poor texture image



Rich texture image

Fingerprint Image

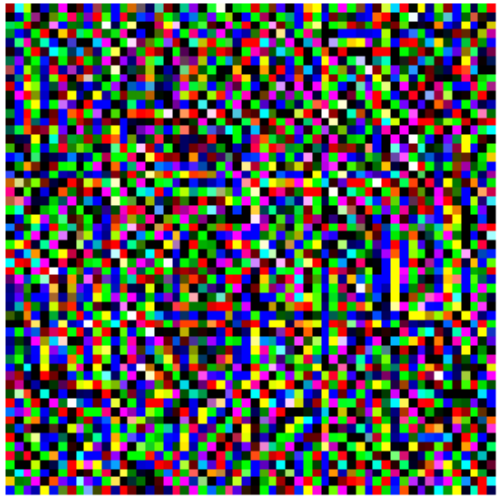


Original Image

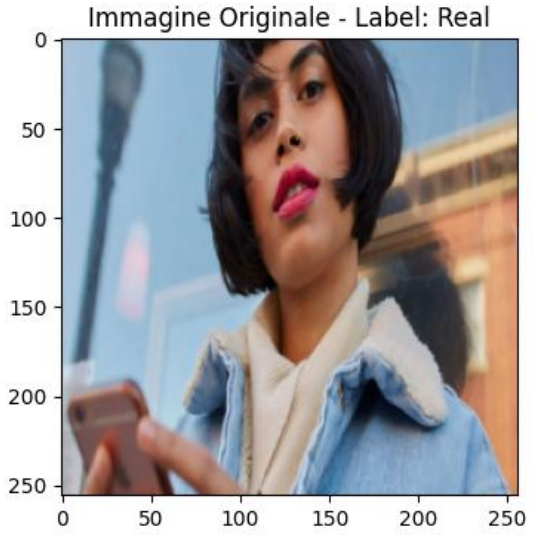
Poor texture image

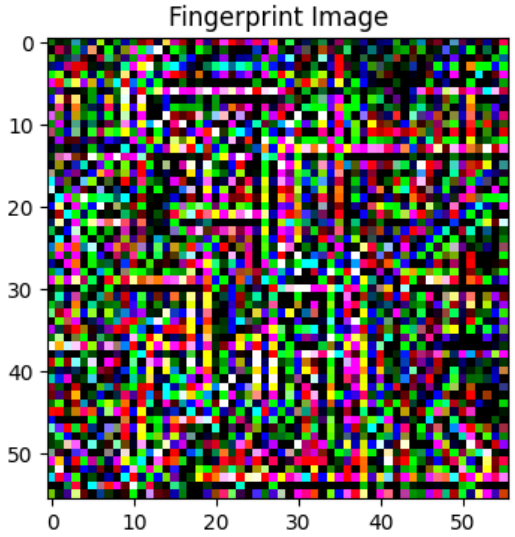
Rich texture image

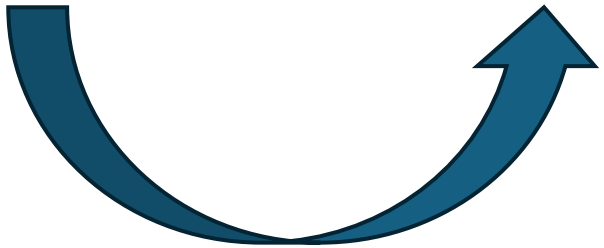
Fingerprint Image



FingerPrint

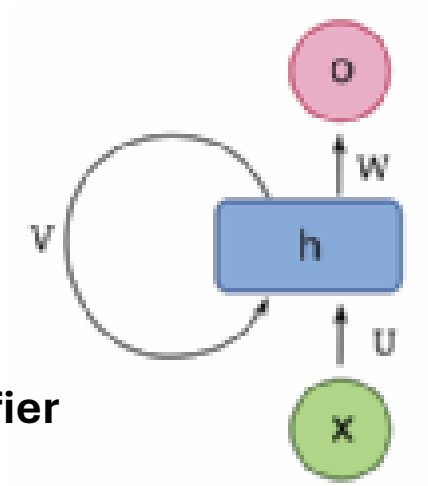






Fed to the Classifier

ResNet

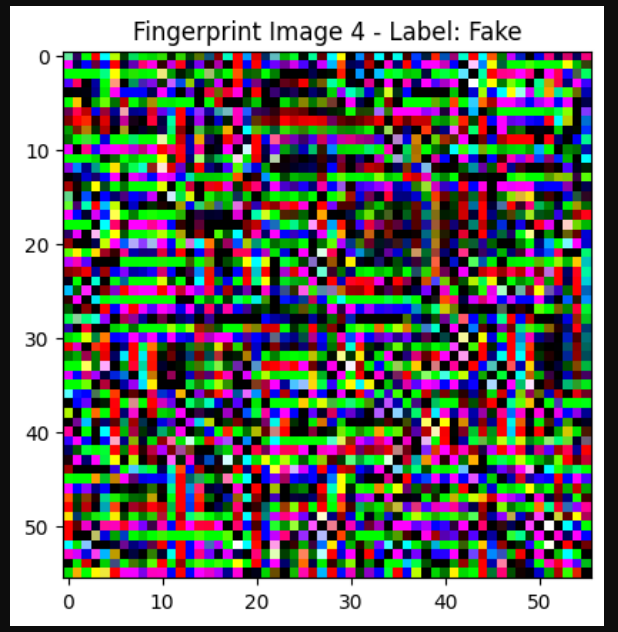


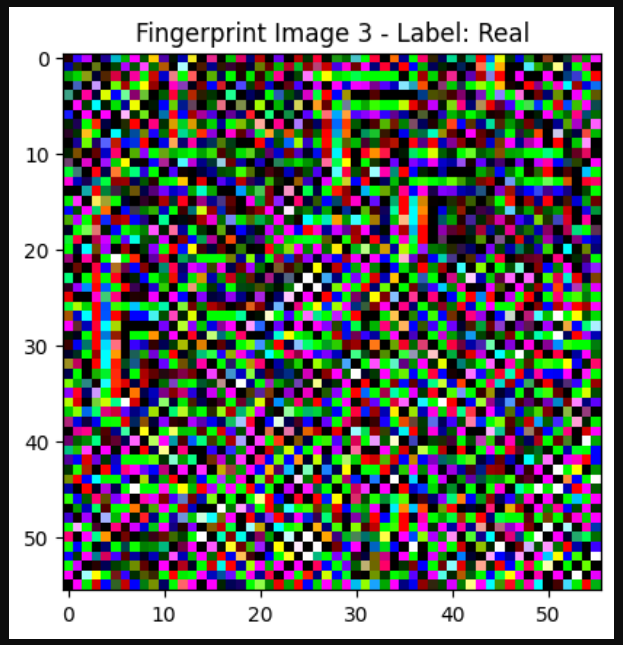
Real

Fake

What pattern?

Presence of **regular patterns** and **repetitive structures**.





Classifiers used

- **CNN**
  (4 convolutional layers)
- **ResNet50**
- **ResNet101**
- **ResNet151**

Datasets

Gianluca Giuseppe Maria De Nardi

Camera Photos vs Ai generated Photos Classifier:

This dataset contains two different categories: camera-clicked photos and AI-generated photos. Each category contains more than 300 images.

Gianluca Giuseppe Maria De Nardi

AI Generated Images vs Real Images:

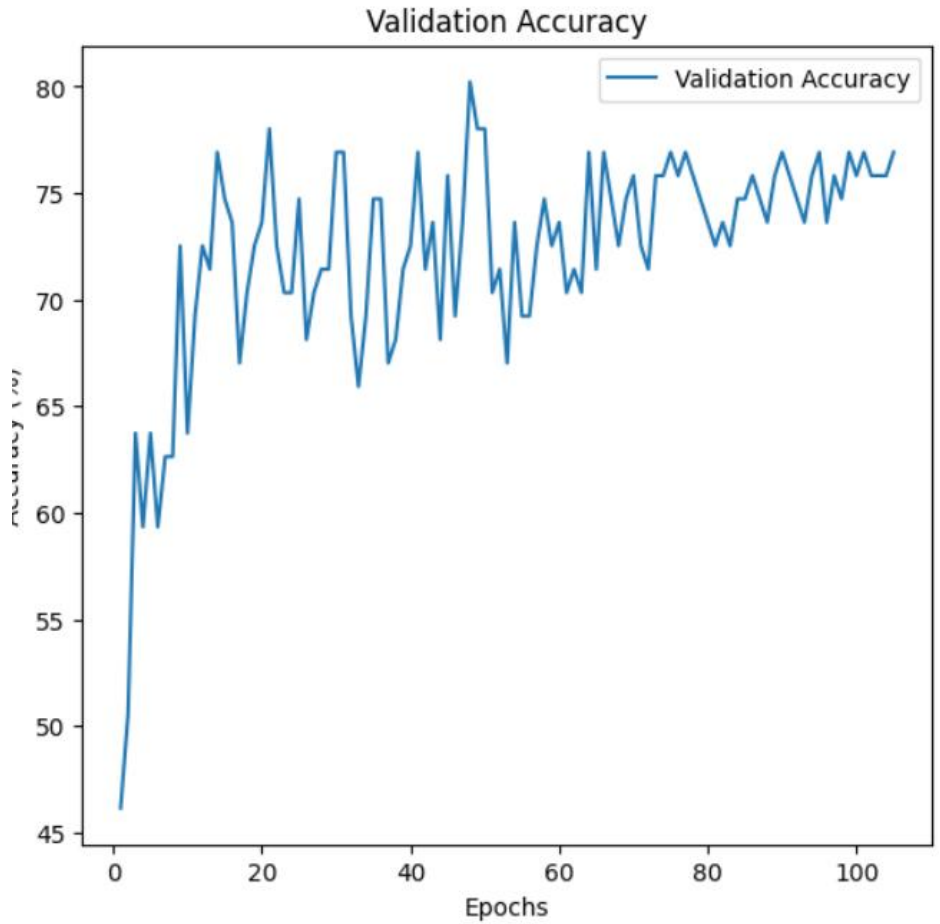
The dataset is a captivating ensemble of images sourced from two distinct channels: web scraping and AI-generated content. The content covers many subjects; however, special emphasis is placed on these topics: people, animals, portraits, scenery, and psychedelics.

Gianluca Giuseppe Maria De Nardi

Table

Classifier	*Accuracy*	*Loss*	*Epochs*
CNN	*59.28%*	*0.53*	***600***
ResNet50	*70.33%*	*0.65*	*70*
ResNet101	***79%***	***0.52***	*100*
ResNet151	*73.63%*	*0.56*	*100*

ResNet101 on Camera Dataset



Conclusion

The proposed method is effective, especially with ResNet101, but relies on complex information extraction models and substantial amounts of data.

SwinTransformer?

Incorporating a SwinTransformer could significantly increase accuracy and predictions. The **sliding window** function would add attention to patches before applying **high-pass filters**, emphasizing high-frequency differences and ensuring efficient model performance even with limited datasets.

Limitation?

GPU power.