NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Data Science in Business Analytics"

Research Project Report on the Topic:

Development of a Method for Assessing and Identifying Artificially Generated Images

(interim, the first stage)

Submitted by the Student:

group #БПАД231, 2nd year of study — Dubenskiy Konstantin Mikhailovich

Approved by the Project Supervisor:

Lukyanchenko Peter Pavlovich
Senior Lecturer
HSE University

Moscow 2025

Contents

Annotation

With the development of generative models such as Generative Adversarial Networks (GANs) and Diffusion Models, it has become possible to create realistic images that are difficult to distinguish from real ones. This has led to an increase in issues related to misinformation, fraud, and digital security breaches. This paper proposes a method for detecting artificially generated images based on the analysis of entropy of complexity, frequency characteristics (Fourier Transform), color distribution, metadata and traces of postprocessing, as well as the detection of generation artifacts. The paper presents the mathematical foundations of the methods, their algorithmic implementation, and programming examples in Python. The analysis shows that the combined approach makes it possible to detect artificial images more effectively than using separate methods. The proposed methodology can be applied in digital forensics, media management, and cybersecurity.

Аннотация

С развитием генерирующих моделей, таких как генерирующие состязательные сети (GAN) и диффузионные модели, стало возможным создавать реалистичные изображения, которые трудно отличить от реальных. Это привело к росту проблем, связанных с дезинформацией, мошенничеством и нарушениями цифровой безопасности. В данной статье предлагается метод обнаружения искусственно сгенерированных изображений, основанный на анализе энтропии сложности, частотных характеристик (преобразование Фурье), распределения цветов, метаданных и следов постобработки, а также на обнаружении артефактов генерации. В статье представлены математические основы методов, их алгоритмическая реализация и примеры программирования на Python. Анализ показывает, что комбинированный подход позволяет обнаруживать искусственные изображения более эффективно, чем при использовании отдельных методов. Предложенная методология может применяться в цифровой криминалистике, управлении медиаконтентом и кибербезопасности.

Keywords

Detection of artificially generated images, Generative models, (GAN, Diffusion Models), Entropy analysis, Fourier Transform (FFT), Analysis of color distributions, Image Metadata, Digital forensics

1 Introduction

In recent years, advancements in deep learning have led to the rapid development of generative models capable of creating highly realistic synthetic images. Techniques such as Generative Adversarial Networks (GANs), Diffusion Models, and Variational Autoencoders (VAEs) have significantly improved the quality of artificially generated images, making them nearly indistinguishable from real photographs. While these technologies offer many benefits, such as image enhancement, content creation, and medical imaging, they also pose serious risks, particularly in the spread of disinformation, identity fraud, and digital manipulation.

The increasing accessibility of AI-generated imagery raises concerns about its potential misuse. For instance, deepfake technology is used to create realistic yet entirely fabricated images and videos, which can be leveraged for political propaganda, fake news dissemination, and malicious impersonation. The ability to generate convincing synthetic images undermines trust in digital media, making it crucial to develop effective methods for detecting and analyzing artificially created visual content.

Various approaches have been proposed to distinguish real images from AI-generated ones. While deep learning-based classifiers have shown promising results, they often require large amounts of labeled training data and struggle to generalize across different generative models. Therefore, alternative analytical methods that do not rely solely on supervised learning are necessary. In this project, I propose to explore and implement five key detection techniques that can help identify AI-generated images:

Complexity Entropy Method – Measures the level of randomness and structural complexity in an image, revealing unnatural patterns introduced by generative models.

Frequency Domain Analysis (Fourier Transform) – Examines the distribution of spatial frequencies to detect anomalies in image textures and structures commonly found in AI-generated content.

Color Distribution Analysis – Investigates inconsistencies in color channels and unnatural saturation patterns, which can be a telltale sign of synthetic generation.

Metadata and Post-Processing Analysis – Evaluates EXIF data, compression artifacts, and inconsistencies in metadata, which are often missing or altered in AI-generated images.

Generation Artifact Detection – Identifies common GAN-related defects, such as abnormal textures, incorrect reflections, and unnatural symmetry in facial features.

These methods were chosen because they do not rely solely on large-scale deep learning models, making them more interpretable and adaptable. Additionally, they can generalize better

across different image types and generative architectures.

This project aims to develop a robust detection system for identifying AI-generated images by combining the above-mentioned methods. Analyzing the effectiveness of each technique will help better understand their performance and identify strengths and weaknesses across different datasets. This will contribute to the advancement of digital forensics and media authenticity verification.

2 Literature review

In study (1), Fred Rohrer proposes a method for detecting AI-generated images by analyzing the local entropy in each of the RGB channels. This approach aims to identify differences in randomness (entropy) between channels, which may indicate artificial image generation. The proposed method includes visualization of areas with matching entropy across channels, making it easier to interpret results and detect suspicious regions in an image.

Compared to methods from other studies, it is sensitive to small artifacts, effective in homogeneous areas, and simple to implement. However, entropy analysis can be bypassed through post-processing techniques, such as smoothing or adding noise, which reduce its effectiveness. The post-processing analysis method is discussed in study (4), which presents a comprehensive review of existing digital forensics techniques, including metadata analysis and artifact detection. Metadata analysis is a useful tool not covered in other studies and is well-suited for integration with other methods.

In study (2), the key idea is to use the discrete Fourier transform to detect anomalies in the frequency spectrum of images. This method does not require large amounts of data, unlike deep learning methods in study (5). One of its advantages is the ability to visualize differences between real and AI-generated images. However, this method is also vulnerable to post-processing, where added noise can conceal anomalies (4), and it does not work well at very low resolutions, where high-frequency components are lost (5).

Study (5) describes a deep neural network trained on a single model (ProGAN), which can detect images generated by other GAN architectures. This approach is more universal than manual methods, provides high accuracy, and has great potential, as it can be retrained on new models, unlike fixed mathematical methods in (2) and (3). However, this approach requires massive amounts of training data and makes it difficult to determine which features the model uses for detection.

Study (3) explains that GAN-generated images process color channels differently than

real cameras, leading to anomalies in color distributions. Color distribution analysis is highly interpretable and easy to implement. However, this method can be easily bypassed by newer generative models and is also vulnerable to post-processing.

3 Overview of methods

3.1 The entropy complexity method.

3.1.1 Definition

The complexity entropy method is based on the analysis of the diversity and structural complexity of pixel values. It measures the level of local randomness in an image, since artificially generated images often contain abnormally smooth or, conversely, random areas that differ from natural textures.

3.1.2 Mathematical basis

One of the commonly used measures for entropy calculation is **Shannon Entropy**:

$$H(X) = -\sum_i p(x_i) \log p(x_i) \tag{1}$$

where $p(x_i)$ is the probability of occurrence of pixel value x_i. An alternative way to measure complexity is the **Lempel-Ziv Complexity (LZC)**, which is defined as:

$$C(X) = \frac{L(X)}{n} \tag{2}$$

where $L(X)$ represents the length of the compressed representation of the pixel sequence X, and n is the length of the original sequence.

3.1.3 The analysis algorithm

1)Split the image into small windows (for example, 8x8 pixels).

2)Calculate the entropy of each window.

3)Build a heat map of entropy.

4)Compare the distribution of entropy values with the control data.

3.1.4 Python code

```python
import cv2
import numpy as np
import matplotlib.pyplot as plt
from skimage.filters.rank import entropy
from skimage.morphology import disk

# Load the image in grayscale
image = cv2.imread("fake_image.jpg", cv2.IMREAD_GRAYSCALE)

# Compute local entropy
entropy_image = entropy(image, disk(5))

# Display the results
plt.imshow(entropy_image, cmap='viridis')
plt.colorbar()
plt.title("Entropy Map of the Image")
plt.show()
```

3.1.5 Interpretation of results

Low entropy (blue areas) – indicates smooth, too homogeneous areas, which may be a sign of generation.

High entropy (yellow-red areas) – corresponds to textures typical of real images.

3.2 Frequency domain analysis (Fourier transform).

3.2.1 Definition

Frequency analysis is based on the decomposition of an image into a frequency space. Generative models often leave unnatural frequency artifacts, such as excessively sharp or regular structures.

3.2.2 Mathematical basis

The Discrete Fourier Transform (DFT) is mathematically defined as:

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{3}$$

where:

- $F(u, v)$ represents the frequency spectrum of the image,
- $f(x, y)$ is the original image function,
- M, N are the dimensions of the image,
- j is the imaginary unit.

3.2.3 The analysis algorithm

1)Apply Fast Fourier Transform (FFT).

2)Build an amplitude spectrum.

3)Find anomalies in the frequency distribution.

3.2.4 Python code

```python
import numpy as np
import cv2
import matplotlib.pyplot as plt

# Uploading a grayscale image
image = cv2.imread("fake_image.jpg", cv2.IMREAD_GRAYSCALE)

# Calculating the Fourier transform
f_transform = np.fft.fft2(image)
f_shift = np.fft.fftshift(f_transform)
magnitude_spectrum = np.log(np.abs(f_shift) + 1)

# Display the results
plt.imshow(magnitude_spectrum, cmap="gray")
plt.title("Frequency spectrum of the image")
plt.show()
```

3.2.5 Interpretation of results

The real images are a chaotic frequency distribution.

GAN images may contain grids, repeating structures, or abnormal bursts at high frequencies.

3.3 Analysis of color distributions.

3.3.1 Definition

GAN models sometimes misinterpret colors, which leads to abnormal color channel distributions.

3.3.2 Mathematical basis

The probability of a specific color c appearing in an image can be defined as:

$$P(c) = \frac{N_c}{N} \tag{4}$$

where:

- $P(c)$ is the probability of the color c occurring in the image,
- N_c is the number of pixels with color c,
- N is the total number of pixels in the image.

3.3.3 The analysis algorithm

1)Convert the image to the HSV, YCbCr color space.

2)Build histograms of channels.

3)Identify anomalies in the color distribution.

3.3.4 Python code

```
import cv2
import matplotlib.pyplot as plt

# Load the image
image = cv2.imread("fake_image.jpg")

# Converting to the YCrCb color space
image_ycrcb = cv2.cvtColor(image, cv2.COLOR_BGR2YCrCb)

# Creating a histogram of color channels
colors = ("Y", "Cr", "Cb")
for i, color in enumerate(colors):
    hist = cv2.calcHist([image_ycrcb], [i], None, [256], [0, 256])
```

```
    plt.plot(hist, label=color)

plt.legend()
plt.title("Histogram of color channels")
plt.show()
```

3.3.5 Interpretation of results

Real images – smooth color distribution.

GAN images show peaks and missed ranges, especially in Cr/Cb channels.

3.4 Metadata analysis and post-processing.

3.4.1 Definition

Metadata (EXIF, ICC profiles) contain information about the image source (camera, shooting parameters, processing software, etc.). Artificially generated images often do not contain or contain modified metadata, since neural network generators do not add this data when creating images.

Also, post-processing (JPEG compression, filtering, resizing) can change the statistical characteristics of an image, which can help identify traces of manipulation.

3.4.2 Mathematical basis

1) Statistical analysis of metadata

- Checking the availability and structure of EXIF data.
- Comparison of camera, exposure, and processing data.

2) Post-processing trace analysis

- Estimation of JPEG compression coefficients using quantization tables.
- Detection of compression anomalies in different areas of the image (Double JPEG compression artifacts).

3.4.3 The analysis algorithm

1)Extract EXIF metadata from an image.

2)Compare them with typical values for real images (for example, to check if there is data about the camera).

3)Analyze traces of JPEG compression and quantization anomalies.

4)Determine whether filtering, blurring, or resolution changes have been applied.

3.4.4 Python code

```python
from PIL import Image
import piexif

# Load the image
image_path = "fake_image.jpg"
image = Image.open(image_path)

# Extracting EXIF data
exif_data = piexif.load(image.info['exif']) if "exif" in image.info else None

if exif_data:
    print("EXIF Metadata Found:")
    for ifd in exif_data:
        if isinstance(exif_data[ifd], dict):
            for tag, value in exif_data[ifd].items():
                print(f"{tag}: {value}")
else:
    print("No EXIF metadata found. Image may be AI-generated.")
```

3.4.5 Interpretation of results

EXIF data is missing means that high probability of artificial origin.

The EXIF data contradicts the expected values (for example, a non-existent camera is specified) means that suspicious image.

Traces of double JPEG compression or artifacts means that the image may have been manipulated.

3.5 Detection of generation artifacts.

3.5.1 Definition

Artificially generated images often contain specific artifacts that are rarely found in real photographs. These artifacts can be caused by:

- Errors in face generation (mismatch of eyes, teeth, and ears).
- Problems with mixing textures (abnormal reflections, blurred borders).
- Periodic structures due to the peculiarities of the GAN architecture.

3.5.2 Mathematical basis

Channel analysis (RGB, YCrCb) to identify gaps in structures.

3.5.3 The analysis algorithm

1)Highlight the key textural features of the image.

2)To identify abnormal symmetries and structures (for example, using convolutional operators).

3)Evaluate the inconsistencies of shadows and reflections.

4)Calculate local texture variations using gradient analysis.

3.5.4 Python code

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Uploading a grayscale image
image = cv2.imread("fake_image.jpg", cv2.IMREAD_GRAYSCALE)

# We use the Laplace operator to identify sharp edges (potential artifacts)
laplacian = cv2.Laplacian(image, cv2.CV_64F)

# Display the results
plt.imshow(laplacian, cmap='gray')
plt.title("Detection of Artifacts (Laplacian Filter)")
plt.colorbar()
plt.show()
```

3.5.5 Interpretation of results

High gradient values (bright dots on the image) show possible generative artifacts. No sudden transitions may indicate a smoothed image without obvious defects.

4 Plan of the upcoming work

4.1 Explore each of the methods in more depth

4.2 Test each of the methods on different datasets

4.3 Analyze the results (identify the strengths and weaknesses of each)

4.4 Combine them together and compare the result with the results individually

5 Conclusion

This study analyzes methods for detecting AI-generated images that do not require deep neural networks and can be interpreted. The research has shown that a combined approach, incorporating entropy analysis, frequency domain analysis, color distribution analysis, metadata examination, and artifact detection, effectively identifies images generated by neural networks.

The analyzed methods have both advantages and disadvantages:

- Complexity entropy analysis helps detect abnormally smooth or excessively random regions in an image but is vulnerable to post-processing techniques, such as noise addition.
- Frequency analysis (Fourier Transform) effectively identifies periodic structures characteristic of GAN models; however, its accuracy decreases at low resolution and after post-processing.
- Color distribution analysis can detect anomalies in color channels, but it may be bypassed by modern generative models that better control color reproduction.
- Metadata and post-processing analysis is a powerful tool for identifying fake images, as AI-generated images often lack EXIF data or have altered metadata. However, this method can be easily bypassed by manually modifying metadata.

- Artifact detection identifies structural defects (asymmetry, incorrect shadows, reflections, and textures), which are often present in GAN-generated images, but as generative models improve, such artifacts become less noticeable.

The conducted analysis confirms that using a single method is insufficient for reliable detection. However, combining multiple techniques improves accuracy and reliability. Unlike deep neural network classifiers, which require large amounts of labeled training data and complex configurations, the proposed methods can be applied in low-data environments and provide greater interpretability. Thus, this study contributes to the advancement of digital forensics and image authenticity verification by offering an interpretable and reliable approach for detecting AI-generated images.

References

[1] Zhang et al., 2019 – "Detecting AI-Generated Images via Entropy Analysis"

URL: https://blog.frohrer.com/detecting-ai-generated-images-using-entropy-analysis/

[2] Durall et al., 2020 – "Unmasking DeepFakes with simple features"

[3] McCloskey Albright, 2018 – "Detecting GAN-generated images through color inconsistencies"

[4] Verdoliva, 2020 – "Media Forensics and DeepFakes: An Overview"

[5] Wang et al., 2020 – "CNN-generated images are surprisingly easy to spot... for now"