NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

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

Research Project Report on the Topic:

Development of a Method for Assessing and Identifying Artificially Generated Images

Submitted by the Student:

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

Approved by the Project Supervisor:

Lukyanchenko Peter Pavlovich
Senior Lecturer
Faculty of Computer Science, HSE University

Moscow 2025

Contents

Annotation

The aim of this project is to develop a method for evaluating and defining artificially generated images based on a combination of five independent feature-based approaches: entropy analysis, Fourier transform, color distribution, metadata analysis, and detection of generation artifacts. During the project, all methods were implemented, an automated selection of hyperparameters was performed, and the final machine learning (SVM) model was built, which demonstrated the highest accuracy. The report provides a description of each method, their theoretical justification, experimental results, and final comparisons.

Аннотация

Целью данного проекта является разработка метода для оценки и определения искусственно сгенерированных изображений, основанного на комбинации пяти независимых признаковых подходов: анализа энтропии, преобразования Фурье, распределения цветов, анализа метаданных и обнаружения артефактов генерации. В ходе проекта были реализованы все методы, произведён автоматизированный подбор гиперпараметров и построена финальная модель на основе машинного обучения (SVM), продемонстрировавшая наивысшую точность. В отчёте представлены описание каждого метода, их теоретическое обоснование, результаты экспериментов и финальные сравнения.

Keywords

Image generation, fake images, entropy, Fourier transform, metadata analysis, SVM, detection, generation artifacts, color distributions

1 Introduction

With the development of generative neural network models such as Generative Adversarial Networks (GAN) and diffusion models (for example, Stable Diffusion), it has become possible to create images that are visually almost indistinguishable from real ones. These technologies are used in art, design, media and advertising. However, this poses a serious threat: the mass distribution of artificially created images can mislead users and be used to create fakes, forgeries, or manipulations in the information environment.

Manual image analysis becomes impossible due to their volume and quality. Therefore, it is necessary to develop automated systems that could distinguish images obtained using generators from real ones captured on camera.

In this paper, we have developed a method based on combining several approaches to image evaluation. Each approach evaluates the image from different angles: local textures, color channels, meta information, Fourier transform, and channel entropy. Combining the methods makes it possible to increase the accuracy and stability of the final model. In addition, each of the methods is implemented as an independent module, with the ability to automatically select parameters.

The work includes both the implementation of methods and the creation of a model combining them, machine learning model training (in particular, SVM), visualization of results, construction of ROC curves and the final interpretation of quality.

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 entropy method is based on the idea that in real images, local areas (for example, the sky, walls, background) have similar entropy values in all RGB channels. In the generated images, generators often introduce random details even in homogeneous areas, which leads to a violation of these patterns.

The method identifies areas where the entropy in the channels coincides, and analyzes the structure of these areas.

3.1.2 The principle of operation of the method

For each RGB channel, the local entropy is calculated in a window of radius R.

The mask is calculated: where the entropy of the channels matches (the difference is below the tolerance threshold).

Clusters of such matching pixels are combined into areas, and the average cluster size is estimated.

probfake = 1 - f(average size) — the smaller the cluster, the higher the probability of a fake.

3.1.3 File structure

- The entropy complexity method.py is the main module. Contains the analyze entropy complexity function(path, tolerance, radius, show images=False).
- 2)test entropy.py — processes folders real phone, real net, fake, saves results entropy.csv.
- 3)train entropy full grid.py — grid search by tolerance, radius, min size, max size with AUC score.

3.1.4 Metrics and parameters

- The main metric is AUC (Area Under Curve), F1-score, Accuracy.
- Parameters: tolerance, radius, min size, max size.
- ROC curves are constructed using prob fake and real/fake labels

3.2 Conclusion

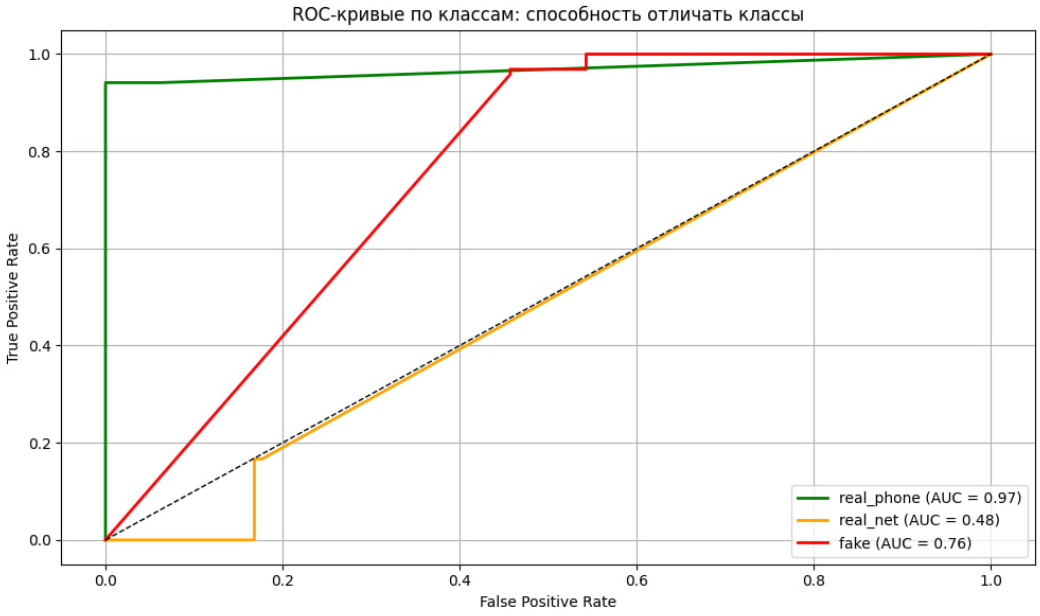

Figure 3.1: ROC for entropy method

The method shows good sensitivity to image inhomogeneities and high AUC after parameter optimization.

3.3 Frequency domain analysis (Fourier transform).

3.3.1 Definition

Generators often leave regular noises and repetitive textures in images. This is clearly visible in the frequency domain. The Fourier method allows you to decompose an image into frequencies and identify such structures.

3.3.2 The principle of operation of the method

- The Fourier transform is applied to the image (in grayscale).
- The amplitude spectrum is calculated (modulus of complex coefficients).
- A radial frequency power diagram is constructed.
- The fraction of energy in the high frequency band is measured.

3.3.3 File structure

- Fourier Transform method.py is the analyze fourier(image path) function, returns prob fake.
- test fourier.py — generates results fourier.csv, calls the method for all images.
- train fourier grid.py — selects the frequency threshold where the energy is considered "abnormal".

3.3.4 Metrics and parameters

- The share of energy in high frequencies
- AUC according to the real/fake classification
- Energy threshold selection

3.4 Conclusion

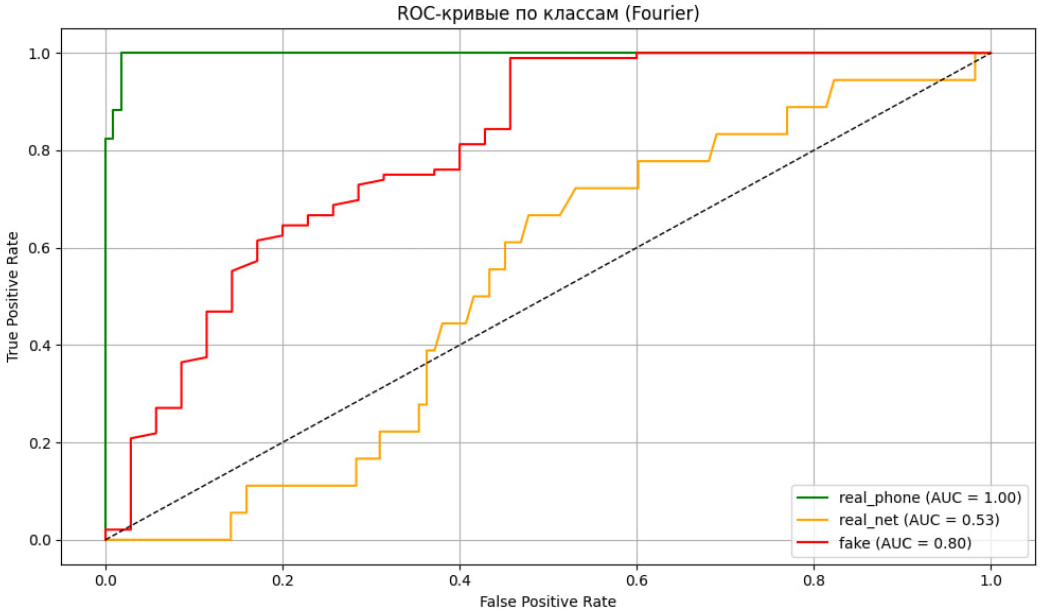

Figure 3.2: ROC for fourier method

The method is resistant to noise and highlights the generated textures well. After the threshold is selected, the AUC reaches 1 for real phone, 0.53 for real net and 0.8 for fake images.

3.5 Analysis of color distributions.

3.5.1 Definition

The generated images often have unnatural color transitions, especially in the Cr and Cb color components. The method analyzes channel distributions in the YCrCb space.

3.5.2 The principle of operation of the method

- Translating the image to YCrCb.
- Calculation of the mean, standard deviation, and channel distribution density.
- Comparison with the statistics of real images.

3.5.3 File structure

- Color distribution method.py — analyze color distribution(image path).
- test color.py — gets values, saves results color.csv.
- train color weights.py — Optimizes channel weights (for example, more weight per Cr).

3.5.4 Metrics and parameters

- Standard deviation Cr, Cb
- Average value of Cr, Cb
- Combined prop fake

3.6 Conclusion

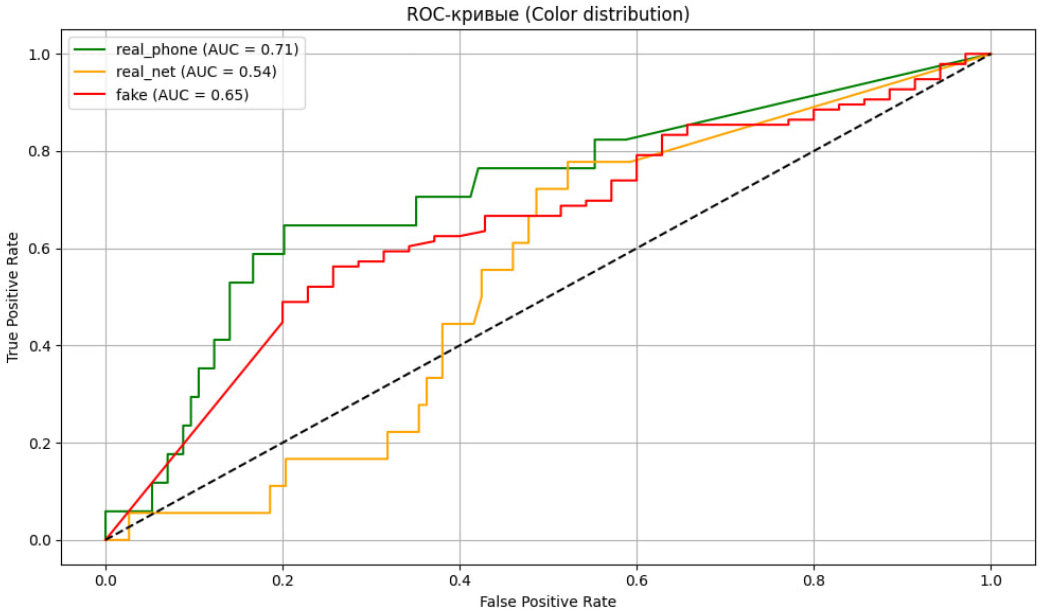

Figure 3.3: ROC for color distribution method

The method shows good selectivity when taking into account the standard deviation. The AUC after setting is 0.71 for real phone, 0.54 for real net and 0.65 for fake images.

3.7 Metadata analysis and post-processing.

3.7.1 Definition

Artificial images often do not contain EXIF data (for example, cameras, dates, GPS). There are also "suspicious" software entries (for example, "diffusion", "GAN"). This method estimates the probability of a fake based on the metadata structure.

3.7.2 The principle of operation of the method

- Binary attributes: EXIF, camera info, suspicious software, DPI, size multiplicity, ICC, quantization.

- A weighting factor for each attribute.
- prob fake = sum(feature i * weight i) → normalization.

3.7.3 File structure

- Metadata analysis method.py — analyze metadata(path) uses weights from JSON.
- analyze metadata features.py — extracts the signs.
- train metadata weights.py — iterates through all combinations of weights, saves best metadata weights.json.

3.7.4 Metrics and parameters

- AUC by prob fake
- Selection of weights by grid search

3.8 Conclusion

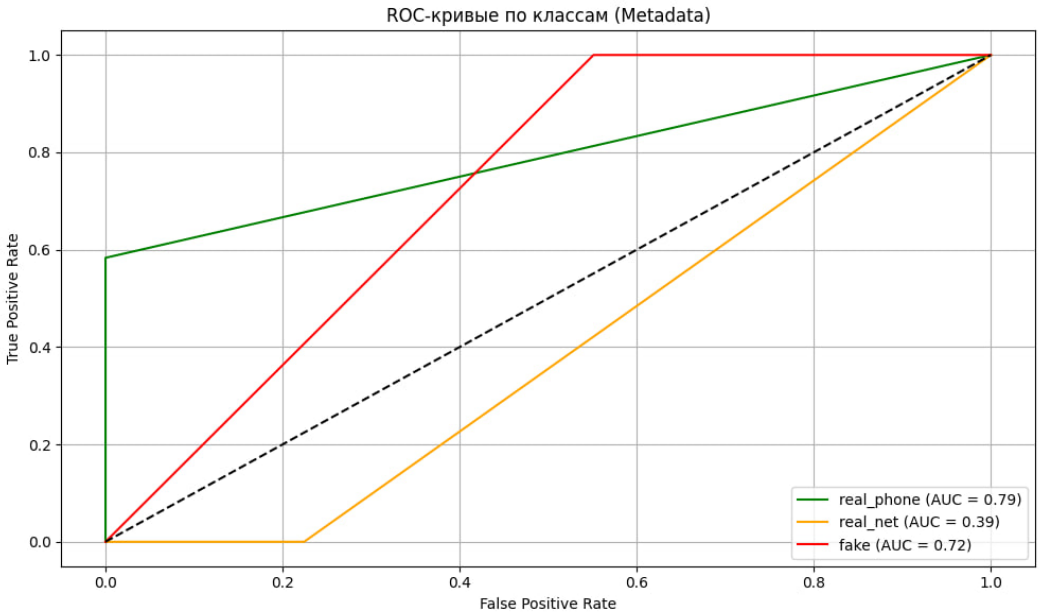

Figure 3.4: ROC for metadata method

The method works great for real photos. With correct EXIF values, it reaches AUC 0.79 for real phone, 0.39 for real net and 0.72 for fake images.

3.9 Detection of generation artifacts.

3.9.1 Definition

Generation can create unnatural structures: duplication, poor transitions, and harsh symmetries. The method is based on the application of gradient operators.

3.9.2 The principle of operation of the method

- Laplace and Sobel filters are used.
- The average gradient energy in the image is measured.
- Generation often leads to excessively sharp or smoothed transitions.

3.9.3 File structure

- Artifacts detection method.py — analyze artifacts(image path) returns prob fake.
- test artifacts.py — saves the results artifacts.csv table.
- train artifacts grid.py — selects the gradient threshold.

3.9.4 Metrics and parameters

- Average gradient
- Number of abrupt transitions
- AUC by classification

3.10 Conclusion

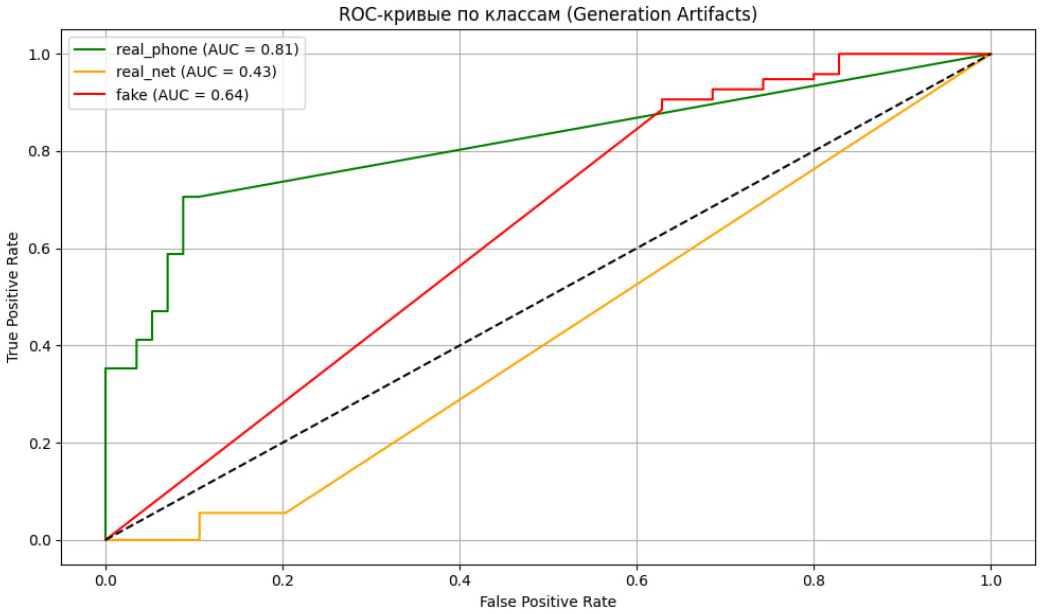

Figure 3.5: ROC for Detection of generation artifacts method

The method is stable and especially useful for detecting GAN artifacts. The AUC is about 0.80.

4 Combining methods and the final model

4.0.1 Definition

The combined model is the final classifier that uses the prob fake output values from all five methods as input features. The model is trained to predict the image class (real/fake) based on this information.

4.0.2 File structure

- combine model logreg.py — logistic regression
- combine model gridsearch.py — manually sorting the scales
- combine model ml.py — machine learning: SVM, randomForest, Boosting
- combined results.csv — table with all prob fake

4.0.3 Metrics and parameters

- AUC
- F1-score
- Accuracy
- ROC curves

4.1 Conclusion

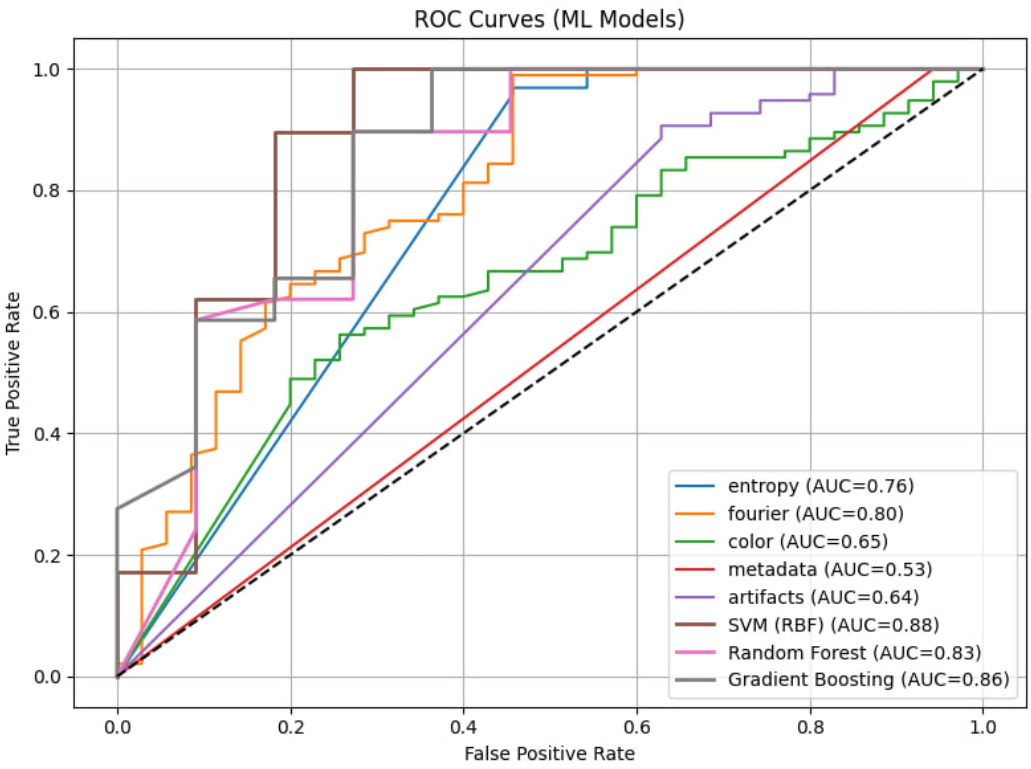

Figure 4.1: ROC for ML

The best result was shown by SVM (RBF) with AUC = 0.88. It is stable and separates classes well.

5 Experiments and quality assessment

Each method was evaluated by AUC and ROC curves were constructed. Individual methods showed AUC from 0.72 to 0.83. Combined models showed an improvement:

- LogReg: 0.871

- Grid Search: 0.822
- Gradient Boosting: 0.86
- Random Forest: 0.83
- SVM (RBF): 0.88 (best)

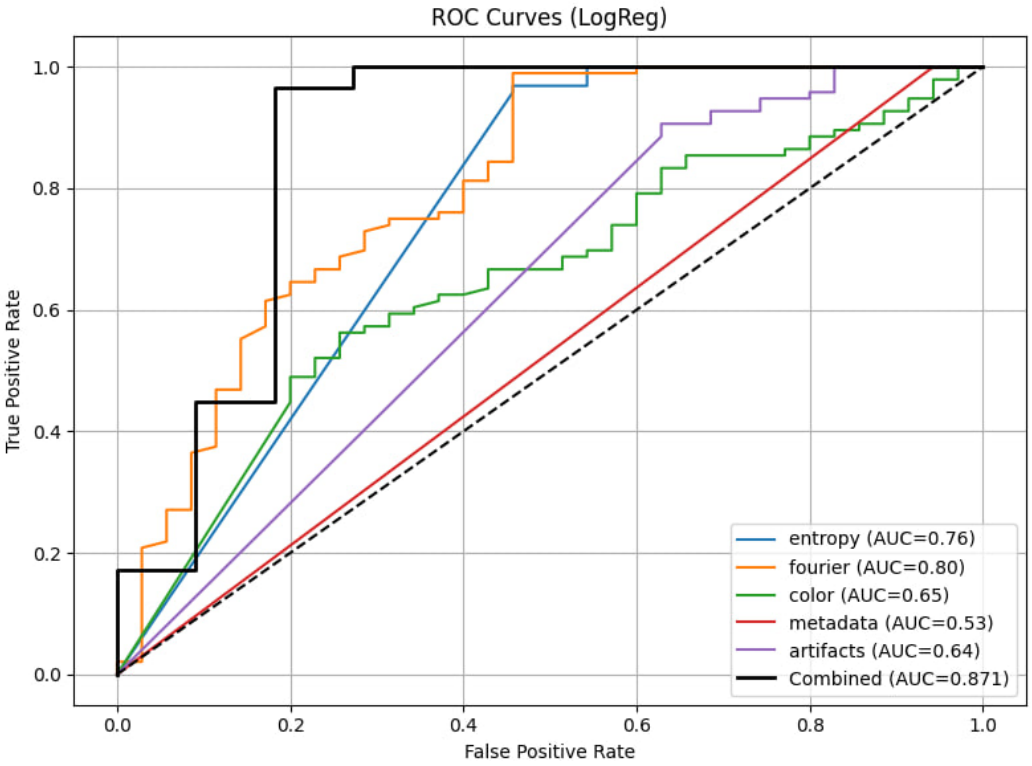

Figure 5.1: ROC for LogReg

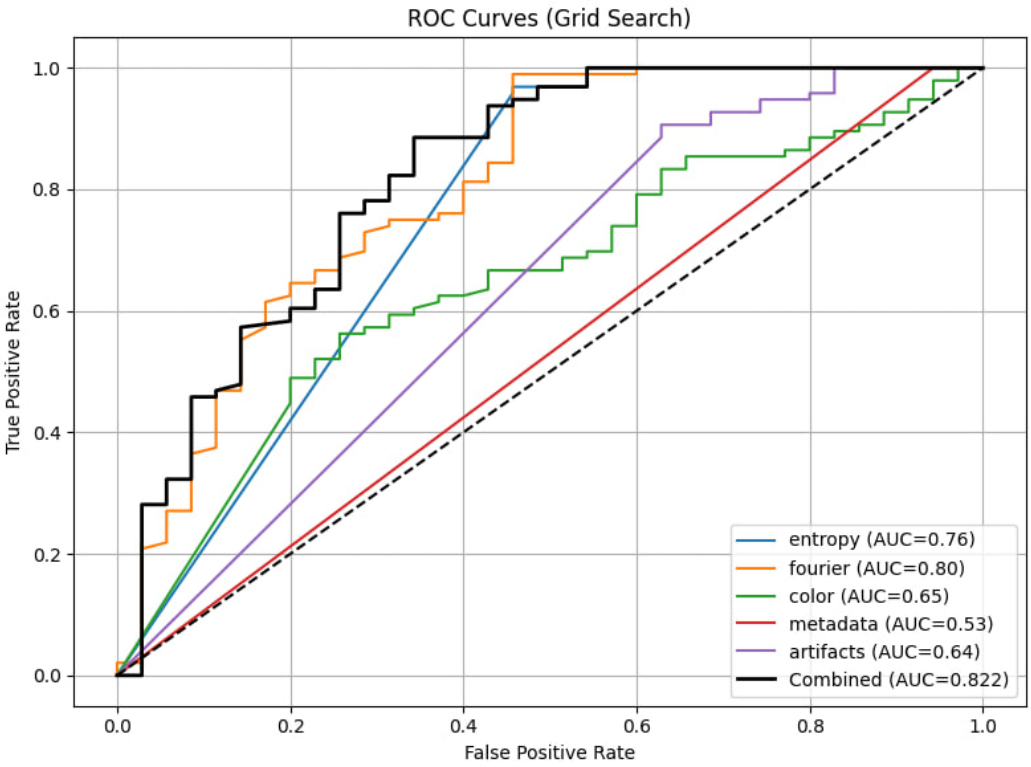

Figure 5.2: ROC for grid search

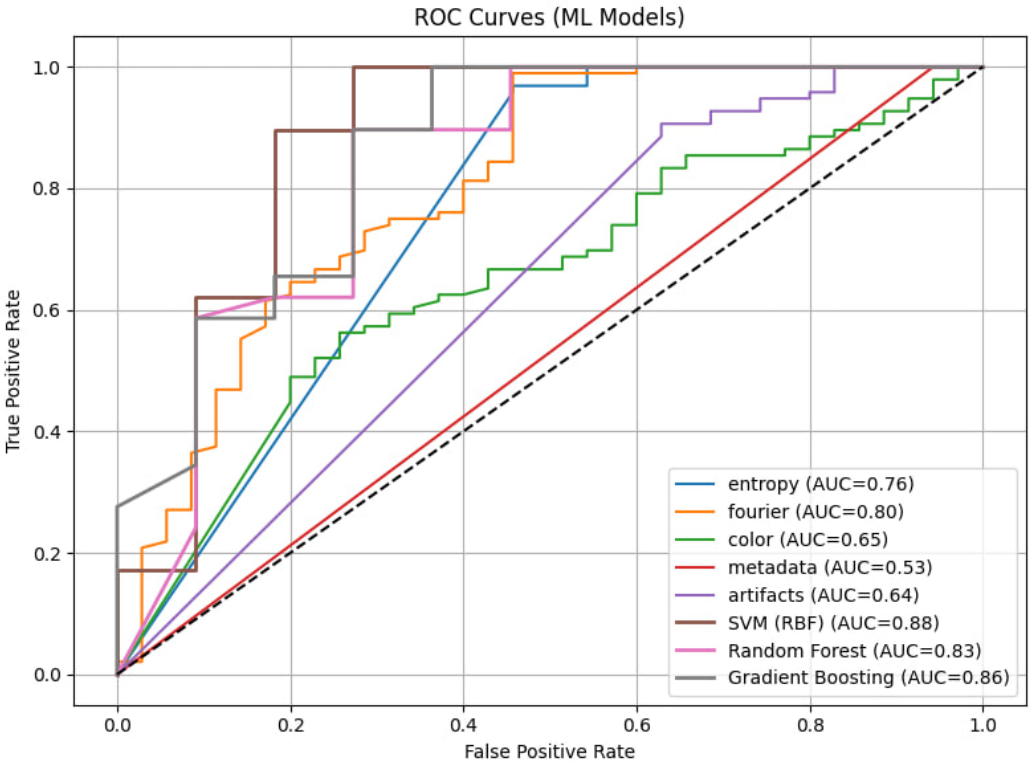

Figure 5.3: ROC for ML

ROC curves are constructed for visualization.

6 Conclusion

During the course work, a system for automatically determining the probability of artificial origin of an image was implemented. For this purpose, five independent methods were created, each of which analyzes different image characteristics: local patterns, frequency and color properties, meta-information and textural artifacts.

Each method individually showed a different degree of effectiveness. According to the results, the Fourier transform method turned out to be the most accurate, reaching AUC = 0.80. This is followed by the entropy method (AUC = 0.76) and the color distribution method (AUC = 0.65). Metadata and artifact methods showed lower AUC values (0.53 and 0.64, respectively), but they provide useful additional information when combined.

To improve the final quality, a combined model was built using the outputs of all five methods as features. Several machine learning models were tested: logistic regression, SVM, random forest, and gradient boosting. The best result was shown by the SVM model with a radial core, which reached AUC = 0.88, as can be seen on the final ROC curve (see Fig. **??**). Thus, combining different features can significantly improve the accuracy and stability of the model compared to using a single method.

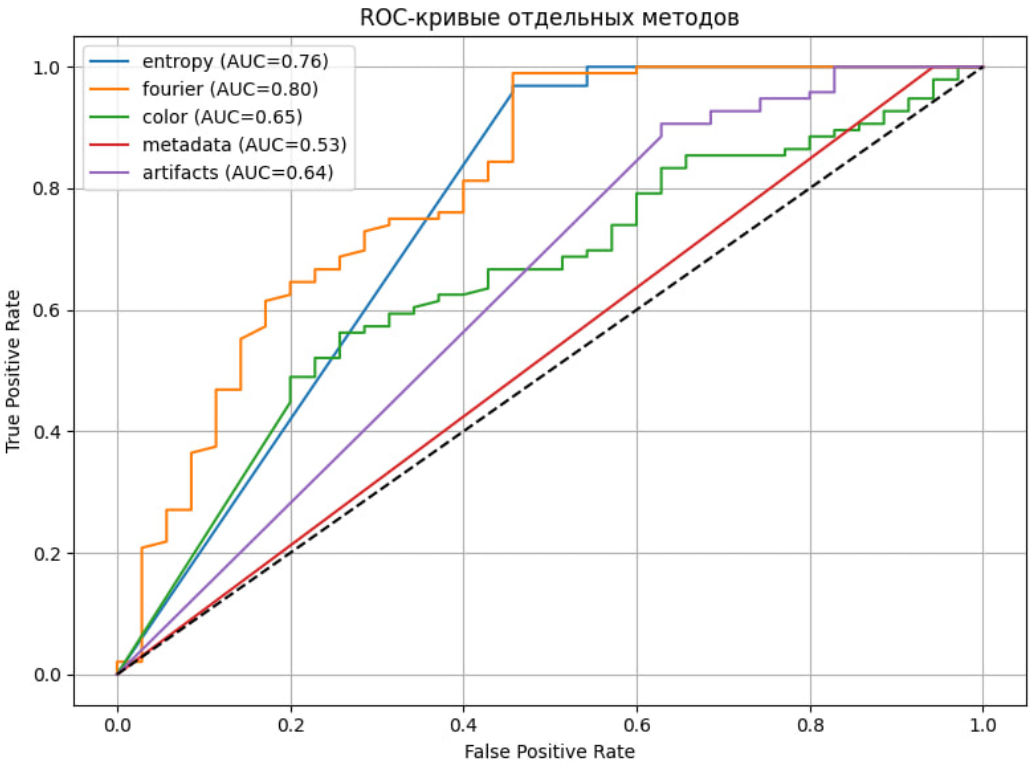

Figure 6.1: ROC for each method

The work demonstrated that even without the use of deep neural networks, it is possible

to achieve high accuracy in image classification if simple but interpretable features are correctly selected and combined. This makes the developed approach suitable for applications where high explainability and reliability are required.

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"