Handwriting features definition methods and algorithms

I Ye Voronina and S V Zinoviev
Department of Applied Mathematics, Informatics and Mechanics, Voronezh State University, 1, University square, 1, Voronezh, 394018, Russia
E-mail: irina.voronina@gmail.com, servzin@gmail.com

Abstract. The research relevance is conditioned upon the lack of assistance tooling for forensics experts and graphologists that allow to justify examination and analysis results. It is a non-trivial task to develop such toolset. The goal of the research is to formalize handwriting features extraction process in order to validate an authorship of a given Russian handwritten text. Existing approaches analysis and a mathematical model of handwriting features extraction from handwritten documents are provided along with the choice of implementation approach. Software toolset that allows the model’s practical application is developed.

1. Introduction
Handwriting computer analysis problem remains being challenging, involving psychologists, forensics, medical, and legal experts. In psychology, handwriting analysis is used both in personal and professional cases, when hiring for a job or during staff assessments. The results acquired by such process allow to detect human’s personal traits, their aptitude for the particular duties, their strengths and weaknesses. In forensics, handwritten objects analysis is widely used for crime combating. Forensics examination is eagerly sought due to the progress of the technical tools which allow modifying or forging handwritten documents. Aside from that, handwriting analysis allows to conduct research of particular historical personalities’ personal traits.

Along with that, there is a lack of additional instruments aiding forensics experts and graphologists to represent their examination and analysis results in an objective manner. Thus, there is a problem of the acquired results’ subjectivity.

It is not uncommon in legal practice when a need in an external independent expert’s involvement is raised during different steps of court proceedings or official investigations. An expert with rich experience and a corresponding qualification uses existing regulations and other important instruments from the judicial standpoint, allowing them to come by some conclusion and decision and to answer the defined questions. However, it is also not uncommon that one of the involved parties disagrees with the testimony results’ quality and objectiveness, and then another examination is assigned by a court to some other expert. According to the research, given the fact that graphologists use quite fuzzy and subjective self-developed and self-ranked criteria when it comes to identifying an author’s identity, it is difficult to qualify the examination results as objective, and thus, it is hard for a court to choose the most trusted results. A judge has to select one of the provided testimonies to become a base for their following decree or maybe even for the sentence. At the same time the judge doesn’t have to obtain professional knowledge in the area of examination, and because of that they assign an independent expert, obliged by the Criminal Code of Russian Federation to be objective within the conducted assessment. Further
hard choice could be avoided if the data would be as objective as possible, and any of the parties involved would be able to assess it.

Forensic documents examination, as one of the methods for expert testimony in court, is an effective tool for crime combating. According to [1], it is the most widespread technique in use of forensics institutions, taking about a half of all conducted forensic investigations. It has a major importance when investigating especially dangerous crimes, graft cases against state or society, documents tampering cases, etc. Expert examination works with various sorts of documents as its subject.

A document in wide forensics meaning is a written act that serves as a proof or evidence of something. This means that documents forensics examination, as a part of criminal investigation technique, one of the forensic science subsections, also deals with documents as proofs.

Computer and peripheral devices are widely used for traditional, paper documents tampering along with other, more traditional ways of forgery. Experts and specialists serve a key role in crime solving and investigation, but law enforcement departments infrequently have their own forensics specialists as their staff.

In 2007 Fedotov [2] paid attention to the fact that important computer forensic sources were mostly published in the US, while there were few publications in Russian that lacked proper legal or technical base by that time. He also mentioned evidence gathering and other forensic tasks software series production as one of the computer forensics metrics along with the existence of social or interdepartmental groups of computer forensics and court experts. Unfortunately, according to his research such software was neither produced nor purchased in Russian Federation at that time; social organizations did not exist even as branches of foreign ones, which serves as a witness to the small number of appropriate specialists.

All this leads to the conclusion that forensic document examination is performed by manual approach. Writing is still one of the main communication channels, and lots of written documents contain handwritten symbols, no matter if it is a script or a signature marking the document’s authenticity. As a result, it is important for graphologists to determine if the document is genuine or tampered in any way. One of the authenticity evaluation criteria is to map handwriting with its author.

Science community conducted a lot of research activities related to handwriting [3]. For instance, [4] perform handwriting analysis from a writing hand’s movement standpoint. It allows to differentiate left hand writing from the right hand one with the help of horizontal and leaning strokes direction. Chernov [4, 5] observes handwriting psychological analysis problem from the human resources management perspective. At the same time, he also takes note of other areas which may use such analysis for their own purposes: medical field may use it for health monitoring, since changes may affect the handwriting; forensics specialists may use it for anti-crime activities. [6] provides results of their research in handwritten records imitation built with the help of a computer and a plotter with a pen mounted to the latter. [7] explore graphologists aid software, which includes such steps as scanning handwritten example document, prerendering the uploaded image, extracting handwriting features, and their further analysis. [5, 8, 9] are dedicated to differentiating a genuine example of handwriting and signature from their imitation.

This work represents a part of the research in graphology examination aimed at the development of a mathematical model of handwriting features extraction from sources in Russian.

2. Materials and methods

2.1. Goal of the research
The goal of the research is to formalize the process of handwriting features extraction in order to determine authorship on a Russian language-written script.

In order to achieve the goal, the following steps were defined:

- conduct a comprehensive research on existing approaches and their in-depth analysis;
- develop a mathematical model of handwriting features extraction process based on script documents;
choose an approach to the developed mathematical model’s implementation;

2. Develop software solution that would implement the mathematical model and allow its application in practice, making it possible to extract handwriting features from example text scripts;

3. Explore and research real-world application of the developed software in forensics and graphology fields.

Due to the lack of any automated toolsets or mathematical models by the time of the research, the goal of the research was set to develop such automated toolset. Within the task several subtasks were identified:

1. Transform handwritten text into a computer-comprehensive representation, a mathematical model, or a digital object.

2. Determine and extract features that allow to categorize and differentiate the transformed objects.

3. Conduct series of experiments targeted on the developed model elaboration and adjustment.

In order to solve the stated problem, it is important to take into account such valuably accounted in forensics features as drawn lines pattern, quantitative amount of force applied to the writing surface, strokes angle. In the meantime, these features may vary their level of intensity even for a single person depending on their moral, physical condition, and on the environmental conditions while creating the handwritten example.

In practice these features are determined in a subjective manner, not being expressed algorithmically or mathematically.

2.2. Mathematical methods and algorithms

Based on the existing solutions review and analysis, we propose a computer analysis solution. The idea is to search for the metrics that would allow to automate identification and grouping tasks methods, along with diagnostic tasks applied to handwritten texts in Russian. To achieve that, it is necessary to extract the most important global and local handwriting features and to build a statistical model based on these features.

Before starting an attempt to create a descriptive model of handwriting in Russian, series of experiments must be conducted. In case of positive results of the research an automated handwriting analysis could be conducted. The obtained metrics could be used for more complex tasks, such as personal traits extraction based on a handwritten text in Russian, author of a signature or its replica identification, and many others.

For the start we made a decision to use a two-dimensional representation of handwriting sources due to the following reasons:

1. Two-dimensional representation allows to use the sources after they were created and doesn’t apply additional restrictions to an input device;

2. In cases when two-dimensional model generated data is not satisfiable, or per other reasons, it may be extended to the third dimension, bringing in such attributes for analysis as the amount or level of pressure applied by the writing instrument.

The problem of handwriting features extraction is complexified by the fact that the importance level of certain features’ applicability and significance vary from one person to another; they may be not even present at all. That being said, there is no such object as a strictly defined set of features: they may gradually drift from one form to another, or they could be unable to be identified depending on psychophysiological state of a single person, not to mention that it is even worse when exploring a subset of people available for identity distinction or of the handwritten documents’ examples.

Because of the concerns mentioned, a decision was made to investigate the generative adversarial networks (GANs) usage for the resolution of the features extraction and categorization problem.

Generative adversarial networks are neural networks which have at least two artificial neural networks in a base structure. These networks compete with each other. One of them that creates objects in data space is called generator; the other one, discriminator, learns to differentiate the objects created by the generator from real life examples, both provided by a learning dataset. Generator and
discriminator have strictly opposite goals, and as the learning process progresses, such competition improves quality of the generated objects.

The general flow diagram of such networks in image creation field is presented in figure 1.

**Figure 1.** General representation of GAN flow.

Discriminant represents a binary classifier that returns one of the two values: 1 for real images and 0 for forged ones. The model learns by end-to-end backpropagation method.

Competitive modelling architecture is best used when each of the two models represent a multi-layered perceptron. In order to get generator distribution \( p_g \) above image data \( x \), prior distribution of noise input \( p_z(z) \), which maps into an image data space \( G(z; \theta_g) \), where \( G \) is a differentiable function represented by multi-layer perceptron with parameters \( \theta_g \). Therefore, images are generated from input noise which shape may be configured and fine-tuned. As a result, generator approaches to the maximization of the value:

\[
\mathbb{E}_{x \sim p_{data}(x)} \left[ \log D(x) \right] + \mathbb{E}_{z \sim p_z(z)} \left[ \log (1 - D(x)) \right]
\]  

(1)

Also, a second perceptron \( D(x; \theta_d) \) is introduced, which passes a single scalar to the output. \( D(x) \) represents probability that \( x \) has been received from input data and not from \( p_g \). As a result, discriminator is learning to maximize the probability of a correct mark assignment both to example samples and to the samples created by the generator. Along with that generator is learning to minimize the value of \( \log (1 - D(G(z))) \).

Generative adversarial networks are based on the maximum likelihood minimization principle. It means that such models parameters are calculated, so that they maximize likelihood of the dataset consisting of \( m \) samples, which were selected independently and evenly distributed from the general population. In a general way, likelihood can be represented as:

\[
\prod_{i=1}^{m} p_{model}(x, \theta)
\]  

(2)

Images from the real images database \( p_{data} \) are sampled, i.e. acceptance sampling from the initial dataset methods are applied to \( p_{data} \) given the fact that the initial goal is present. They allow to select such data that enable performing structural-parametric identification of the best statistical model for a random process. That being said, sampling is performed in one step in case of using generative adversarial networks, unlike chaining that is used in alternative approaches. That allows not to additionally bring in Markov chains for assistance. Images from the real images database \( p_{data} \) are marked with the value of 1 as real ones, and after that they are fed to the discriminator. As a result, probability of belonging to real images set is calculated.
Usually for the sake of computational efficiency likelihood logarithm is maximized, which is represented in (3):

$$\theta^* = \arg\max_{\theta} \log \prod_{i=1}^{m} p_{\text{model}}(x_i, \theta) = \arg\max_{\theta} \sum_{i=1}^{m} \log p_{\text{model}}(x_i, \theta)$$  

(3)

It is worth noting separately that likelihood maximization is equivalent to Kullback—Leibler divergence minimization [10] between the real data distribution and modelled one, which is indicated in (4):

$$\theta^* = \arg\min_{\theta} D_{KL}(p_{\text{data}}(x) \| p_{\text{model}}(x, \theta))$$  

(4)

Discriminator likelihood logarithm maximization is performed on the dataset constructed from real and generated images. This causes the density shift to the real images’ side, which is represented in (5):

$$\max_{\theta} \mathbb{E}_{z \sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z \sim p_{\text{gen}}}[\log (1 - D(x))]$$

(5)

where \(D(x, \theta_g) \in [0, 1]\) is the discriminator neural network and \(\theta_g\) represents its parameters.

Then the functional \(\min_G \mathbb{E}_{z \sim p_{\text{data}}}[1 - D(G(z))]\) is minimized, aiming to the state so that the discriminator would differentiate real images from the forged ones as bad as possible. It is possible to update generator weights by gradient descend algorithm because \(D \circ G\) is a differentiable function. When summing up everything from above in a single optimization problem, a minimax game for two players is defined. It may be represented as:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z \sim p_{\text{gen}}}[\log (1 - D(G(z)))]$$

(6)

Forward network pass serves as a channel for information transfer from the generator to the discriminator, and reverse pass serves as a way to transfer the information from the discriminator to the generator. However, despite the resemblance to the initial conditions of supervised learning, whole model leaning with the help of coming from the discriminator gradients inverse is not suitable for generative adversarial networks. During the first learning iterations discriminator would not have any trouble differentiating forged data from authentic one, which brings to the conclusion that \(D(G(z)) \approx 0\), which, at the same time, means that \(\log (1 - D(G(z))) \approx 0\). As a result, gradients will fade, meaning that the generator wouldn’t get any new information from the discriminator, and generator will lose. In case of an assumption that discriminator would always operate next to the optimum, substitution of \(\min_G \log (1 - D(G(z)))\) to \(\max_G \log D(G(z))\) will help to solve the fading problem, but at the same time will lead to the fact that the solution of the minimax game and of the new Maximax one would be the same.

In order to guarantee that discriminator would always operate next to optimum values, another approach is suggested in [11]. On each learning iteration first the discriminator is learning on sample batches constructed from real and fake images with the fixed generator. Then one step of generator learning is performed for its learning process on the fake images batch with the fixed discriminator. Such approach would allow to always keep the discriminator around optimum and gradually modify the generator. As a result, tug of war game is achieved, resulting into an unstable learning.

At the same time, it is noted that minimax game equation from (6) may fail to provide a suitable gradient for the generator that would provide training on a decent level. During the early stages of learning when the generator is not trained enough yet, the discriminator is able to reject the samples with the high assurance because they are obviously different from the training data. In this case
log\(1 - D(G(x))\) is saturated. Instead of training the generator to minimize the value of 
log\(1 - D(G(x))\), the generator may be trained to maximize log \(D(G(z))\). Such target function
returns the same fixed point of the generator’s and discriminator’s dynamic, but at the same time
provides a significantly stronger gradient on early learning stages.

The same article provides a proof that the optimal discriminator is represented with an equation (7) for any fixed generator:

\[
D^*_G(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}
\]  

(7)

This equation allows to rephrase the minimax game equation as:

\[
C(G) = \max_D V(G,D) = \min_G \max_D V(D,G) =
\[
= \mathbb{E}_{x \sim p_{data}} \left[ \log D^*_G(x) \right] + \mathbb{E}_{z \sim p_z} \left[ \log \left( 1 - D^*_G(G(z)) \right) \right] =
\[
= \mathbb{E}_{x \sim p_{data}} \left[ \log \left( \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right) \right] + \mathbb{E}_{x \sim p_z} \left[ \log \left( \frac{p_g(x)}{p_{data}(x) + p_g(x)} \right) \right]
\]  

(8)

Also, the equation from formula (7) enables to reveal functional’s \(C(G)\) global minimum existence
exclusively at \(p_g = p_{data}\), and its value will equal \(-\log 4\). Indeed, if \(p_g = p_{data}\), then \(D^*_G(x) = \frac{1}{2}\),
and because of that \(C(G) = \log \frac{1}{2} + \log \frac{1}{2} = -\log 4\). To make sure that it is the best possible value
of \(C(G)\) which may be obtained only at \(p_g = p_{data}\), let’s observe the equation:

\[
\mathbb{E}_{x \sim p_{data}} \left[ -\log 2 \right] + \mathbb{E}_{z \sim p_z} \left[ -\log 2 \right] = -\log 4
\]  

(9)

When subtracting (9) from \(C(G) = V(D^*_G, G)\), following equation is concluded:

\[
C(G) = -\log 4 + KL \left( \frac{p_{data} p_{data} + p_g}{2} \right) + KL \left( \frac{p_g p_{data} + p_g}{2} \right),
\]  

(10)

where KL — Kullback—Leibler divergence.

Generative adversarial networks topology is presented in figure 2.

Also, the authors of [11] prove that the consecutive process of generator and discriminator training
converges, but it is true only in cases when any possible functions are explored as candidates to generator
and discriminator. However, in real life optimization problem cannot be solved in the space of all
possible functions. Instead, it must be set within the neural networks’ family restrictions, or in other
words, parametrically represented functions. Thus, when optimizing the functional \(C(G)\) in the space
of neural networks (i. e. in high dimensional space of real numbers), that particular global optimum
cannot be found.
Figure 2. GAN topology graphical representation. Figure 3 represents idealized visualization of convergence process and its result.

Figure 3. Idealized virtualization of convergence process and its result in classical GANs. The black-dotted line represents data distribution $P_{\text{data}}$; the green line shows generator distribution $p_g$ that approaches data distribution during the consecutive training process; the blue line is a division surface of the discriminator which cannot differentiate created by generator forgeries from genuine data by the end of training process. The black arrows in the figure represent the function $x = G(z)$ that maps each value for the data distribution to a point from the prior distribution.

As a result, it is possible to highlight following differentiative properties of generative adversarial networks:

- generative adversarial networks do not require apparent condensation. Nonetheless, there is a possibility to organize learning with the goal of some particular form of distribution, which makes a somewhat connection between generative adversarial networks and variational methods;
- generative adversarial networks don’t require obtaining the lower bound for likelihood logarithm computational complexity;
• sampling in generative adversarial networks is performed within a single step, unlike the consecutive chain of methods application and transformations used in such algorithms as Markov chain Monte Carlo (MCMC), which also take a lot of time to process. At the same time, all learning process still happens during an extended period of time with an unstable behavior and without any guarantee within parameterized functions space as opposed to the same asymptotically convergent MCMC;
• the generator learns to create close-to-real images without actually facing with one by just receiving an encoded signal within gradients from the discriminator about the criteria the fake element was detected;
• the model is absolutely useless for features extraction from an image, which is the final goal of the research.

Due to the latest a decision was made to refer to modified GAN which could allow to solve the initial problem. **Bidirectional GAN (BiGAN)** flavor was chosen as a base. The main difference between simple generative adversarial networks and bidirectional ones is that in BiGAN discriminator model is trained to detect a joint probability \( P(x, z) = \text{real} \), where \( x \) is a sample and \( z \) is a generated distribution. At the same time, it means that generator model is trained to encode real sample into its generative distribution. When bidirectional generative adversarial networks finish their training cycle, both the real image into generative distribution encoder and activations of the first layer of convolution could be used for data representation [12].

The basic diagram of BiGAN is presented in figure 4.

In addition to the generator \( G \) from the standard generative adversarial networks, BiGAN also includes an encoder \( E \) into its structure, which maps data \( x \) onto latent representations \( z \). At the same time, the discriminator \( D \) rejects not only the data from the data space (\( x \) versus \( G(z) \)), but also jointly in data and latent space (tuples \( (x, E(x)) \) versus \( (G(z), z) \)), where the latent encoder is either an output from the encoder \( E(x) \) or an input for the generator \( z \).

![Diagram](image)

**Figure 4.** General BiGAN topology diagram.

It may not be obvious from the description that BiGAN encoder \( E \) must learn to invert the generator \( G \). The two modules cannot directly interact with each other because the encoder never receives a generator’s output (i.e. \( E(G(z)) \) is never calculated), and vice versa. However later the authors point out to the fact that both encoder and generator must learn to invert each other in order to fool BiGAN’s discriminator.

Thus, the encoder of BiGAN learns to predict features \( z \) from the received data \( x \). Along with that, other articles on generative adversarial networks indicated that these features capture data semantic attributes. As a summary, a supposition was made that a trained BiGAN encoder may serve as a useful mapping of according semantic problems in the same way as fully trained for semantics prediction by a supervisor visual models, which mark (place tags) onto provided images. Such approach may be used
for other similar visual problems. In the given context a latent representation \( z \) may be visioned as mark
for data \( x \) acquired without any need in supervised learning.

An alternative approach to learning an inverted correlation from data to latent representation is to
directly model \( p(z|G(z)) \), which predicts a generator input \( z \) by the provided generated data
\( G(z) \). Such alternative was named a latent regressor. Later it was argued that BiGan encoder may be
preferable in feature learning context.

Despite all these features, BiGAN is a very generalized and solid approach for non-supervised
learning while theoretically not assuming data structure or type of its application.

From the mathematical standpoint, bidirectional generative adversarial networks train both the
generator and the encoder \( E : \Omega_x \to \Omega_z \). The encoder populates a distribution
\( p_E(z|x) = \delta(z - E(x)) \) which maps data points \( x \) into the latent space of generative model features.
The discriminator is also modified in order to accept input data from the latent space by forecasting
\( p_D(Y|x,z), \) where \( Y = 1 \) if \( x \) is genuine (i. e. taken from the real data distribution \( p_x \)), and \( Y = 0 \) if
\( x \) was generated (obtained from the generator \( G(z), z \sim p_z \).

The target function of BiGan is defined as a minimax goal:

\[
\min_{G,E} \max_D V(D, E, G)
\]

where:

\[
V(D, E, G) = \mathbb{E}_{x \sim p_x} \left[ \mathbb{E}_{z \sim p_z(x)} \left[ \log D(x, z) \right] \right] + \mathbb{E}_{z \sim p_z} \left[ \mathbb{E}_{x \sim p_G(z)} \left[ \log (1 - D(x, z)) \right] \right] = \mathbb{E}_{x \sim p_x} \left[ \log D(x, E(x)) \right] + \mathbb{E}_{z \sim p_z} \left[ \log (1 - D(G(z), z)) \right]
\]

This function is optimized by using the same optimizations based on variable gradient as for standard
GAN.

BiGan possesses a lot of theoretical features that are applicable to the standard generative
adversarial networks. Yet it additionally guarantees that at global optimum generator \( G \) and encoder
\( E \) are reciprocals. BiGan also has a close relationship with autoencoders with a loss function \( \ell_0 \).

2.3. Practical implementation of the approach to handwriting feature extraction algorithmization

As a result of potential methods analysis of the handwriting features algorithmic identification
implementation, it was decided to look into one of the generative adversarial networks variations. Due
to the problem stated, modified BiGan was chosen as the most suitable for the research architecture. It
allows not only to review the result of generator’s and discriminator’s competitive work, but also to
extract data features. The idea is the following: instead of training to encode real sample into its
generative distribution, the model is trained to encode features received as a result from teaching
discriminative model (activation of the last layer of convolution) into a generative distribution.
Activation of the last layer of convolution and its encoding concatenation into a generative distribution
may be used as a new data representation. When adding L2 regularization to the model losses and using
Adam (a gradient-based method of efficient stochastic optimization) as an optimizer [13], such
architecture revealed the best results on a training dataset comparing to regular BiGan. Parameter
calculation by Adam algorithm is defined by formulas (13), (14), (15), (16) and (17):
\begin{equation}
m_{w}^{(t+1)} \leftarrow \beta_{1} m_{w}^{(t)} + (1 - \beta_{1}) \nabla_{w} L^{(t)}
\end{equation}

\begin{equation}
v_{w}^{(t+1)} \leftarrow \beta_{2} v_{w}^{(t)} + (1 - \beta_{2}) \left( \nabla_{w} L^{(t)} \right)^{2}
\end{equation}

\begin{equation}
\hat{m}_{w} = \frac{m_{w}^{(t+1)}}{1 - \beta_{1}^{t+1}}
\end{equation}

\begin{equation}
\hat{v}_{w} = \frac{v_{w}^{(t+1)}}{1 - \beta_{2}^{t+1}}
\end{equation}

\begin{equation}
w^{(t+1)} \leftarrow w^{(t)} - \eta \frac{\hat{m}_{w}}{\left( \hat{v}_{w} \right)^{1/2} + \epsilon},
\end{equation}

where \( w^{(t)} \) is a given parameter, \( L^{(t)} \) is a loss function, \( t \) is a current iteration index, \( \epsilon \) — small addition used to prevent division by zero, and \( \beta_{1} \) and \( \beta_{2} \) are decay rates for gradients and for the second moment gradients correspondingly.

3. Results and discussion

To build a model of modified BiGAN Python 2.7 programming language was chosen. TensorFlow was used as a machine learning framework.

First, a class implementing BiGAN architecture support was developed; its structure is represented in table 1.

**Table 1. BiGAN class.**

| Method name      | Arguments         | Description                                      |
|------------------|-------------------|--------------------------------------------------|
| __init__         | Number of features| Initializer method of BiGAN class.               |
|                  | Number of historical days | Receives input data and sets a necessary topology |
|                  | Input data size   |                                                  |
| discriminator_conv | Received data x  | Method that implements discriminator convolution |
| discriminator    | Received data x   | Method that implements discriminator structure based on discriminator convolution |
|                  | Generated features z |                                                  |

A separate class was implemented on top of TensorFlow framework for model training method. To start with, we used MNIST dataset consisting of handwritten digits.

Input data was represented by digits images of \( 28 \times 28 \) pixels resolution that should be used in the format of 784-D vector (i.e. unstructured vector which features are evaluated by the greyscale intensity). In this case such condition was satisfied by the fact that each module was built as a multi-layer perceptron which didn’t have any information about underlying space structure (for instance, in contrast with convnet [14]). Latent distribution was chosen as \( p_{z} = \left[ U(-1,1) \right]^{50} \) — continuous uniform distribution. Number of epochs was set to 200, number of iterations for deployment steps execution was equal 1000, maximum number of samples for representation was set to 400, and calculation and mapping of the test results were performed every 25 epochs. Adam was chosen as an optimization method and was implemented by built-in software instruments and libraries.

BiGAN target function was set by the following parameter string:
OBJECTIVE="--encode_gen_weight 1 --encode_weight 0 --discrim_weight 0 --joint_discrim_weight 1"

As the result of the launched tests on One Nearest Neighbors (1NN) classification on the permutational-invariant MNIST dataset, the accuracy achieved 97.39%, which is comparable to other architectures on generative adversarial networks. Overall, it is an expected result since the initial dataset is rather simple and MNIST dataset is quite narrow. Qualitative results are represented in figure 5, figure 6, and figure 7.

Figure 5. Qualitative test results on 1NN for permutational-invariant MNIST dataset, including generator samples $G(z)$.

Figure 6. Qualitative test results on 1NN for permutational-invariant MNIST dataset, including genuine data $x$.

Figure 7. Qualitative test results on 1NN for permutational-invariant MNIST dataset, including corresponding reconstructions $G(E(x))$.

Acquired results allow to continue the research on more complex datasets with more global goals. Before proceeding with the further tests, a separate problem would be to collect a high-quality dataset. It is not easy to solve due to the following concerns:

- currently there is no such publicly available prepared dataset with samples of handwritten symbols used in Russian writing;
- similar data collection would result in legal intricacy that needs to be complied in accordance with the laws of Russian Federation, including but not limited to personal data collection and handling according to the federal law “On personal data” from June 26, 2006 № 152-ФЗ;
- if two-dimensional images would not be sufficient, special software and hardware for three-dimensional samples collection would be necessary to be used, which may substantially complicate testing preparation phase;
- obtained raw data should be prepared accordingly in order to be transformed into generative adversarial networks input format. This format should enable getting the best results for artificial neural networks training. Within the conducted research the simple method was used which theoretically could provide not the best results with the initial research goal taken into account.

Later it would be important to receive some structured feedback as the calculation results from artificial neural networks during the experiments and evaluations. It would be also valuable to provide the obtained results to forensics experts and graphologists for their analysis in order to identify weak spots and fine-tune the software to improve the produced results. Ideally the developed system would be trained to the point where a graphologist won’t be assured in his opinion on the author identity of a handwritten document laying in front of them; only in this case the chosen approach would be considered as successful and suitable to solve the problem.

4. Conclusion
The developed system prototype allows to show vitality of the chosen mathematical model for the goal of algorithmic toolset for handwriting features extraction development that would help to identify the authority of a script. The next steps would be to develop a transparent aiding solution for graphology examination and court proceedings.
References

[1] Vinberg L A and Shvankova M V 1977 A Graphology Expertise ed R S Belkin (Volgograd: Higher Investigations School of USSR Internal Ministry) pp 14–23 (in Russian)

[2] Fedotov N N 2007 Computer Forensics (Moscow: Judicial World) pp 26–27 (in Russian)

[3] Zinoviev S V, Voronina I Ye and Meshcheryakov V A 2018 Handwriting computer analysis problem Computer science: problems, methodology, technologies: Proc. of the Int. Conf. vol 1 pp 45–9 (in Russian)

[4] Chernov Yu G 2012 Handwriting Analysis in Human Resources (Saint-Petersburg: BHV-Petersburg) p 180 (in Russian)

[5] Chernov Yu G 2011 Psychological Handwriting Analysis: System Approach and Computer Implementation in Psychology, Forensics and Legal Examination (Moscow: Genesis) pp 86–90 (in Russian)

[6] Panova T O, Milovidova O Yu and Karpukhina Ye S 2008 Complex research on handwriting imitation Forensic Exam. Th. Prac. 11 118–21 (in Russian)

[7] Sheikhoaslami G, Srihari S N and Govindaraju V 1997 Progress in handwriting recognition World Sci. Publishers ed A C Downton and S Impedovo (World Scientific) pp 503–9

[8] Mann M, Shukla S K and Gupta S 2015 J Forensic Res 6 293–8

[9] Brugnatelli F 2013 The variability measurement of shape independent features to establish a new method to differentiate genuine signatures from simulated ones J Forensic Res. 4 206–19

[10] Goodfellow I, Bengio Yo and Courville A 2016 Deep Learning (MIT Press) p 631

[11] Goodfellow I J, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair Sh and Bengio Yo 2014 Generative adversarial networks arXiv 1406.2661

[12] Donahue J, Darrell T and Krähenbühl P 2017 Adversarial feature learning arXiv 1605.09782

[13] Kingma D and Ba J 2014 Adam: a method for stochastic optimization arXiv 1412.6980

[14] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Zh, Karpathy A, Khosla A, Bernstein M et al 2015 ImageNet large scale visual recognition challenge arXiv 1409.0575