# Intelligent systems for recognizing artistic styles: a Deep Learning approach

Nataliya Boyko[1†], Yaroslav Borys[1]

[1] Lviv Polytechnic National University, the Department of Artificial Intelligence Systems, Lviv, 79013, Ukraine

**Abstract**

The study presents a machine learning-based approach for artistic style recognition in images, examining its practical value, feasibility, and potential applications. Existing research on the topic is analyzed, comparing different approaches and highlighting their strengths and limitations. The proposed method utilizes convolutional neural networks (CNNs) for style classification, trained on the WikiArt dataset containing over 100,000 high-quality images. The study details the data preparation process for training, provides a general overview of neural networks, and offers an in-depth analysis of the proposed CNN architecture. Finally, the experimental results are reviewed, identifying the model's limitations and discussing possible enhancements to improve accuracy and overall performance.

**Keywords**

machine learning, image classification, artistic style recognition, convolutional neural networks (CNNs), data preprocessing, WikiArt dataset

## 1. Introduction

In the modern era of digital transformation, the automation of visual content analysis processes is becoming increasingly important in various fields, from art history to information technology. In particular, the recognition of artistic styles based on images has become one of the promising research areas, uniting the interests of researchers in the field of artificial intelligence, cultural heritage and computer vision. Traditionally, the identification of artistic style required in-depth expert assessment by art historians, which made the process subjective and small-scale. However, the development of machine learning, in particular convolutional neural networks (CNN), allows us to automate this process and increase its objectivity and efficiency [1, 5].

To date, there are a number of studies devoted to the automatic classification of artistic styles, in particular using pre-trained models such as AlexNet, ResNet, VGG and Inception. At the same time, most of them focus on a limited number of styles or demonstrate reduced accuracy in recognizing similar stylistic directions. In addition, some of the solutions require significant computational resources or large amounts of training data, which limits their practical use.

Among the unresolved problems, it is worth highlighting the low classification accuracy when detecting styles with similar visual features, such as abstract expressionism and color field, as well as the difficulty of scaling models without losing the quality of the result. There is also a contradiction between the accuracy of models and their ability to generalize, which is especially noticeable when using models on new, unfamiliar data [3].

The goal of this research is to develop an effective model for automatically identifying artistic styles of images using convolutional neural networks, which provides high classification accuracy at moderate computational costs. The task is to create our own CNN architecture, train it on a

*MoDaST 2025: Modern Data Science Technologies Doctoral Consortium, June, 15, 2025, Lviv, Ukraine

[1*] Corresponding author.

[†] These authors contributed equally.

nataliya.i.boyko@lpnu.ua (N. Boyko); yaroslav.borys.kn.2020@lpnu.ua (Y. Borys)

0000-0002-6962-9363 (N. Boyko); 0009-0006-6830-7211 (Y. Borys)

subset of the large WikiArt set, implement preprocessing and data augmentation methods, as well as analyze the results and potential areas for improvement.

Thus, the research is relevant and aimed at overcoming the limitations of existing approaches, using the advantages of modern information technologies in the field of machine learning to solve interdisciplinary problems.

## 2. Problem statement

Classification of artistic style is a non-trivial task since styles often have overlapping visual characteristics, which complicates their automatic identification. Moreover, stylistic features can manifest themselves in compositional details, color palette, brushstrokes, or general aesthetics, which are not always amenable to clear formalization. Thus, there is a need to build a model that can effectively generalize complex visual patterns and distinguish even close painting styles [4].

From a mathematical point of view, the problem of classifying artistic styles is formalized as a multi-class classification problem:

Given a set of images $X=\{x_1,x_2,\ldots,x_n\}$, where each $x_i \in R^{h\times w\times c}$ is a three-dimensional tensor representing an image of size $h\times w$ with $c$ color channels (usually $c=3$ for RGB images).

Let, $Y=\{y_1,y_2,\ldots,y_n\}$, where each $y_i \in \{1,2,\ldots,K\}$ is a class label corresponding to one of $K$ art styles.

The goal is to find the function $f: R^{h\times w\times c} \rightarrow \{1,2,\ldots,K\}$, which approximates the correspondence between the input image $x_i$ and its style $y_i$ with maximum accuracy: $\hat{y}_i=f(x_i;\theta)$, where $\theta$ – model parameters.

To construct such a function, a convolutional neural network (CNN) is used, which consists of a composition of nonlinear functions and parameters $\theta=\{W_1,b_1,\ldots,W_l,b_l\}$, which are learned during the optimization of the objective function (1):

$$L(\theta)=\frac{1}{n}\sum_{i=1}^{n} L_{CE}\left(y_i,f(x_i;\theta)\right), \tag{1}$$

where $L_{CE}$ is the crossentropy function for multiclass classification, which measures the difference between the predicted distribution and the true class.

Thus, the solution to the problem is to construct a model $f$ that minimizes $L(\theta)$ on the training sample, while ensuring good generalization ability on new images.

## 3. Analysis of the latest research and publications

In scientific research on the automatic identification of artistic styles using machine learning, various approaches are considered that allow classifying paintings by stylistic features. One such method is UFLK (Unsupervised Feature Learning), proposed by Eren Gultepe and other authors. This approach consists of using unsupervised learning to highlight stylistic characteristics of paintings, after which they are classified according to stylistic similarity. The results of the study show that this method can be useful for classification and clustering, but its accuracy does not always correspond to the level of more complex approaches [2].

Another approach using the support vector method (SVM) is described in the work of Alexander Blessing. In his study, works of art are classified by artists, and the model trained on a sample of 750 paintings achieved an accuracy of about 78.53%. However, when trying to use hidden features for classification, the accuracy of the model decreased, indicating the problem of overtraining when processing complex features [6].

A study by Adrian Lecoutra and colleagues examines the use of deep neural networks, including the AlexNet and ResNet models, to automatically recognize artistic style based on 25 categories. The authors note that the accuracy of the model increased when adding additional layers to the pre-trained networks, although the overall accuracy remained at 62%. Further improvements in the results are possible by using the bootstrap aggregation method [11].

Saqid Imran and his colleagues propose an interesting two-stage approach to classifying painting styles. The first stage involves dividing the image into five parts, each of which is classified by a separate convolutional neural network. The second stage processes and combines the probability vectors obtained from the first stage. This approach allows for a significant improvement in classification accuracy, reaching 90.7% accuracy, and with properly tuned hyperparameters even up to 96.5%. However, this method requires a large amount of data for training and pre-tuning the models.

A study by Maftuhah Rum and Arda Priscilla on the classification of naturalism and realism styles showed that the use of pre-trained MobileNetV3 models provides high classification accuracy, reaching 95%. This result demonstrates the effectiveness of using lightweight pre-trained models for specific tasks, although for a wider range of styles such models may have limitations [7, 15].

A study by Jacqueline Valencia and Gerradina Pineda provides an overview of trends in the use of machine learning for predicting artistic styles. They highlight that most existing research focuses on historical styles, while contemporary art remains understudied. This opens up significant opportunities for further developments in this area.

Finally, an analysis of existing tools for classifying painting styles, such as Art Style Identifier AI and Analyzer-Art Style Identification, shows that most of them require a subscription to access full functionality. This indicates a lack of accessible and effective tools for widespread use in research and education, which highlights the importance of developing alternative solutions [8, 9].

A brief comparison of existing publications, outlining their advantages and disadvantages, is presented in Table 1. This comparison helps identify the strengths and limitations of current research, providing insight into areas where further improvements and developments are needed.

**Table 1**
Analysis of the main publications on the researched topic

| Article | Methods | Advantages | Disadvantages |
|---|---|---|---|
| Predicting and grouping digitized paintings by style using unsupervised feature learning [1] | Unsupervised feature learning using the K-means algorithm (UFLK). | Finds stylistic features without prior information about their nature and whether the painting belongs to a particular style. | Not being a finished product, the study only emphasizes the possibility of further use of the approach. |
| Using Machine Learning for Identification of Art Paintings [2] | SVM | The model training accuracy on a small sample reaches 78.53%. | When determining the prized traits, the accuracy decreases and there is a significant risk of overtraining. |
| Recognizing Art Style Automatically in Painting with Deep Learning [3] | Use of trained AlexNet and ResNet models. | The model was trained on a dataset of 25 different styles, and we are considering improving the results by bootstrap aggregation. | It had an accuracy of only 62%. |
| Artistic Style Recognition: Combining Deep and Shallow | A two-stage system, the first stage of which involves the | Using pre-trained models, the accuracy is 90.7%, which was increased to | For accurate classification, the model requires a large data set, |

| Article | Methods | Advantages | Disadvantages |
|---|---|---|---|
| Neural Networks for Painting Classification [6] | use of CNNs to process images divided into 5 parts, and the second stage involves the use of SNNs. | 96.5% by adjusting hyperparameters. | and the accuracy drops sharply when studying hidden features. |
| Using Machine Learning to Classify Art Style in Naturalism and Realism [12] | Use of trained LeNet models, Pretrained ResNet-50 and Pretrained MobileNetV3. | An accuracy of about 95% was achieved. | The model can only distinguish between two artistic styles: realism and naturalism. |

Thus, existing research indicates progress in the application of machine learning to classify artistic styles, but also reveals a number of problems, such as limited classification accuracy when working with similar styles and the need for larger datasets for the models to work effectively.

## 4. Materials and Methods

The main method for recognizing artistic styles is the use of convolutional neural networks (CNNs), which can automatically extract important features from images and detect complex visual patterns. CNNs consist of several types of layers [11, 13]:

- Convolutional layer - the convolutional layer, the main layer of the network. It contains a set of filters (kernels) whose parameters will be learned during training. There are several layers of this type, and each subsequent layer usually learns a larger number of filters (it is common to use a power of two to determine the number of filters, e.g. 32, 64, 128). In most cases, the size of the filters is smaller than the size of the image; each filter creates an activation map when collapsed with the image.
- Pooling layer - an aggregation layer that divides the input data (activation maps) into small regions over which aggregation operations (e.g., average, maximum, or minimum) are performed. This operation allows for compression of activation maps without significant cost; often 2x2 regions are used.
- Fully connected (dense) layer - a layer where every input “neuron” is connected to every output “neuron”.

For this study, a proprietary CNN architecture was developed, consisting of 10 layers, including:

- 4 convolutional layers for image feature extraction.
- 5 subsampling layers (Max Pooling) to reduce data size and reduce the number of parameters.
- 1 fully connected layer for classification of results.

The ReLU (Rectified Linear Unit) function was used to activate neurons, which helps to avoid the problem of "dying" neurons and speeds up the learning process.

The model is trained using the Backpropagation algorithm, which allows optimizing network weights by minimizing the loss function. For this task, cross-entropy was used as the loss function for multi-class classification (Equation 1).

The Adam algorithm was used as an optimizer for training, which allows for effective tuning of network parameters.

Now let's discuss main arguments of Keras layers.Conv2D class [14], that was used for our research:

- Kernel size - specifies the size of the convolution window
- Filters - number of filters in the convolution, which affects the dimension of output space
- Stride - specifies stride length of convolution (the number of pixels the kernel moves through), if > 1 may conflict with other arguments
- Padding - when value is ‘same’ creates padding evenly to the sides of the input
- Activation function - determines whether a neuron should be activated based on its input.

A few more words on activation function. It introduces non-linearity, allowing the network to learn complex patterns beyond simple linear relationships. Each activation function is best suited for its scenario, as described below in Table 2 [10, 14].

**Table 2**
Description of most used activation functions

| Activation function | Formula | Description |
|---|---|---|
| ReLU | $f(x)=max(0,x)$ | Simple and prevents vanishing gradients. |
| Leaky ReLU | $f(x)=x\, if\, x>0,$ <br> $else\, 0.01x$ | Solves “dying ReLU” problem. |
| Sigmoid | $f(x)=\frac{1}{1+e^{-x}}$ | Used in binary classification, suffers from vanishing gradient. |
| Tanh | $f(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$ | Similar to sigmoid but centred at 0, reducing bias. |
| Softmax | $f(x_i)=\frac{e^{x_i}}{\Sigma e^{x_j}}$ | Converts outputs into probability distributions, best suited for multi-class classification. |

In this research, the training will be done on the WikiArt dataset [12]. The WikiArt [12] dataset is a large and diverse dataset containing images of works of art collected from the eponymous WikiArt.org, an online resource for studying artworks.

In total, it contains more than 100,000 high-quality images of various artistic styles and authors: 27 styles are available for research, including Baroque, Renaissance, Impressionism, Realism, Pop Art, etc.; the authors include Claude Monet, Paul Cezanne, Gustav Klimt, Leonardo da Vinci, etc (See Fig. 1).

The dataset is structured in three main categories:

- by artistic style;
- by author;
- by subject.

This organization makes it ideal for research in the following areas:

- classification of artistic styles;
- analysis of image authorship;
- training models to generate new images based on the provided ones.

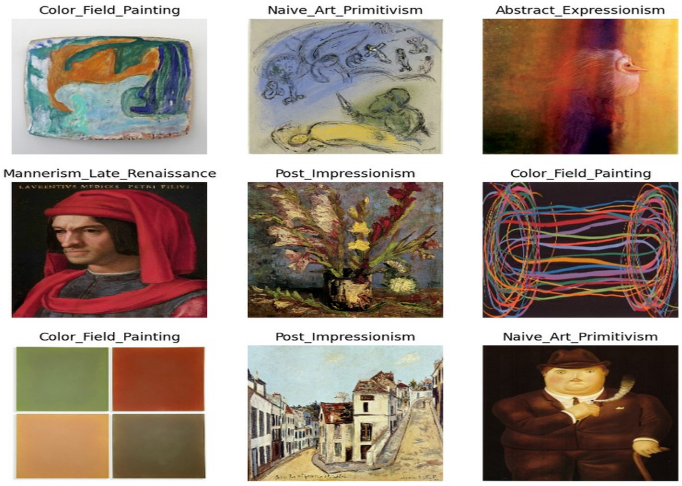


**Figure 1:** Example of WikiArt dataset batch [12].

An analysis of the distribution of paintings by artistic styles shows that the largest number of works belongs to impressionism - more than 13,000 works, while the smallest number, only 98, is represented in the style of action painting.

A similar analysis was performed to classify images by resolution. Most images are between 500 and 2000 pixels in size, which requires additional adjustments before feeding them to the model.

In this research only 5000 images in total will be used in training on 5 classes (styles):

- Abstract Expressionism.
- Color Field Painting.
- Mannerism Late Renaissance.
- Naive Art Primitivism.
- Post Impressionism.

Let's discuss how data is read, transformed and presented as input to the model. The data is stored in folders with the names, corresponding to specific classes. Then using keras image_dataset_from_directory [12] function images are read and stored alongside an alphabetically sorted array of classes. During this operation images are resized to specified size, namely 265 by 265 pixels.

To allow the model to learn faster data normalization is used. Each pixel in image channels (R, G, B) is divided by its maximum value, 255. This also improves generalization and prevents bias towards bright or dark images.

To improve model robustness and reduce potential overfitting data augmentation was introduced. Several techniques were used, namely

- Random flip.
- Random rotation.
- Random contrast.
- Random brightness.

The algorithm for image processing and analysis using convolutional neural networks (CNN) consists of several stages that can be described mathematically. The main goal of this algorithm is to transform an image into a set of features that allow for classification by artistic styles.

We have an input image $I$, which is represented as a matrix of size $h \times I_n \times c$, also:

- $h$ – mage height (number of pixels vertically),
- $I_n$ – image width (number of pixels horizontally),
- $c$ – number of color channels (usually $c=3$for RGB images, where each pixel contains three values – red, green and blue).

Where $I_{i,j,k}$ the value of the pixel at position $(i, j)$ for color channel $k$.

The first step is image normalization, which scales pixel values to a range from 0 to 1. This is done by dividing each pixel by its maximum value (255 for 8-bit images) (2):

$$I'_{i,j,k}=\frac{I_{i,j,k}}{255}. \tag{2}$$

Now all pixel values $I_{i,j,k}$ are within the range [0, 1].

In addition, to improve the generalization ability of the model, data augmentation can be applied, which includes operations such as random rotations, reflections, changes in contrast or brightness. This helps to avoid overfitting and increase the diversity of the data.

A neural network processes an image using convolutional operations, which are applied to each layer of the CNN. Convolution is an operation in which a filter (kernel) $K$ of size $f \times f \times c$ is slid over the image and creates a new feature matrix (activation).

This can be written as (3):

$$A_{i,j,k}=\sum_{p=1}^{f}\sum_{q=1}^{f}\sum_{r=1}^{c} I'_{i+p,j+q,r}\cdot K_{p,q,r}, \tag{3}$$

where $A$ is the activation for pixel $(i, j)$ on $k$-th channel, $K$ is the filter, $f$ is the filter size, and $c$ is the number of channels in the image.

This process allows us to extract local features such as edges, textures, and colors, which will then be used for classification.

After each convolution, pooling is usually applied to reduce the size of the activations and the number of parameters. Typically, Max Pooling is used, where the maximum value is selected for each small region. Mathematically, this looks like this (4):

$$P_{i,j,k}=\max_{p,q} A_{i+p,j+g,k}, \tag{4}$$

where $P_{i,j,k}$are the values after subsampling for pixel $(i, j)$ on the $k$-th channel, and $p$ and $q$ are the sizes of the subsampling window.

This process allows you to reduce the dimensionality of the image and preserve important features.

After several convolutional and subsampling layers, the image is passed to the fully connected layers, which perform the classification. These are a set of neurons, each of which is connected to all the outputs of the previous layers [13].

Let $x$ be the vector containing the activations after the last convolutional and subsampling layer. These activations are now fed to the input of the fully connected layers, where each neuron of the $j$-th layer has an output (5):

$$y_j=\sigma\left(\sum_i W_{ij} x_i + b_j\right), \tag{5}$$

where $y_j$ is the output of $j$-th neuron, $W_{ij}$ is the weight between $i$-th input and $j$-th output, $b_j$ is the offset for $j$-th neuron, and $\sigma$ is the activation function (usually ReLU or Softmax for multiclass classification).

At the output of the last fully connected layer, we get a probability vector for each class (art style). The probability vector $\hat{y}$ can be calculated using the Softmax function (6):

$$\hat{y}_k = \frac{\exp(y_k)}{\sum_{i=1}^{K} \exp(y_i)}, \tag{6}$$

where $\hat{y}_k$is the probability that the image belongs to class $k$, and $K$ is the number of classes (styles).

The loss function for multiclass classification is usually calculated using crossentropy (7):

$$L = -\sum_{k=1}^{K} y_k \log(\hat{y}_k) \tag{7}$$

where $y_k$ is the true class label, and $\hat{y}_k$ is the predicted probability for class $k$.

To optimize the weights and biases of the network, methods such as the backpropagation algorithm and the Adam optimizer are used, which minimizes the loss function by updating the model parameters.

## 5. Experiments

Several experiments were conducted to evaluate the performance of the designed convolutional neural network for image style recognition. The final version of the network consists of 10 hidden layers, including 4 convolutional layers, 5 pooling layers, and 1 dense layer. The full network structure is illustrated in Fig. 2. During the experiments, various hyperparameters, data augmentation techniques, and optimization strategies were tested to enhance model performance and ensure robust classification.

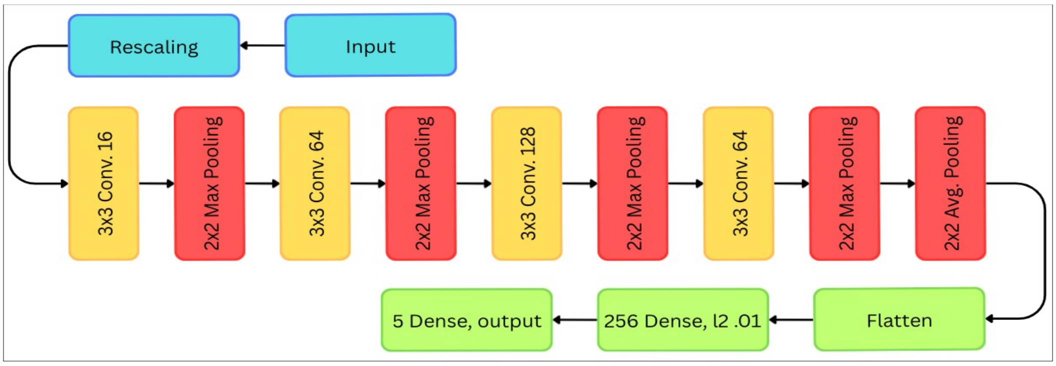


**Figure 2:** Our own CNN architecture.

To optimize computational efficiency early stopping was implemented. This algorithm continuously monitors a specified model metric – validation loss in this research - and halts the training process when the metric shows consistent increase. This helps the model avoid potential overfitting and also reduces training time.

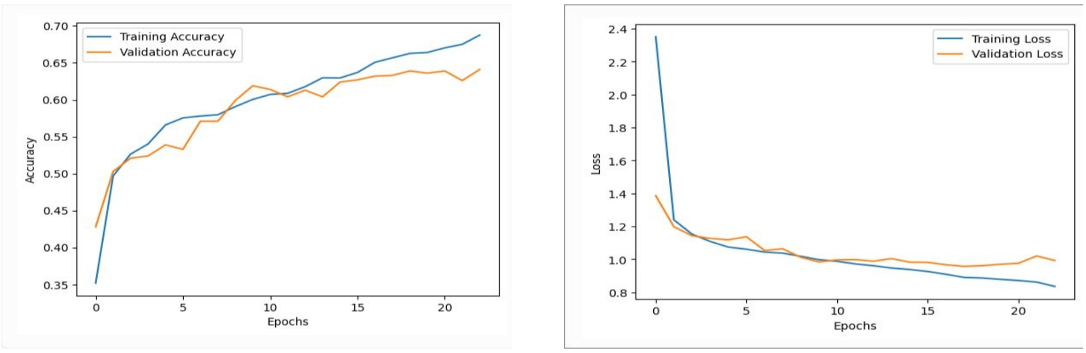
**Table 3**
The values of the parameter used in training

| Parameter | Value |
|---|---|
| Number of epochs | 100 |
| Batch size | 32 |
| L2 regularization | $10^{-2}$ |
| Learning rate | $10^{-2}$ |

The training results are presented in Fig. 3 and Fig. 4. The highest accuracy achieved was 0.7 for the training data and 0.63 for the validation data. Analyzing the graphs, it can be concluded that the model lacked sufficient complexity for the given task. This suggests that a deeper architecture, additional feature extraction mechanisms, or further hyperparameter tuning may be required to improve performance.

Currently two possible solutions are being considered to improve model performance:

- Reducing the number of classes, thus artificially increasing model complexity by allowing it to focus on fewer categories. This also opens the possibility of using ensembles methods to detect a larger number of classes.
- Using pre-trained models such as VGG16 or Xception for transfer learning, which can significantly enhance feature extraction.



a) b)

**Figure 3:** a) Graph of model accuracy; b) Graph of model losses.

To further analyze the model's performance in style detection, it was tested on a subset of 250 images spanning five different styles. As shown in Fig. 4, the model demonstrates high accuracy in detecting the Color Field Painting style (labeled as 1), while achieving moderate accuracy for Mannerism - Late Renaissance (labeled as 2) and Post-Impressionism (labeled as 4).

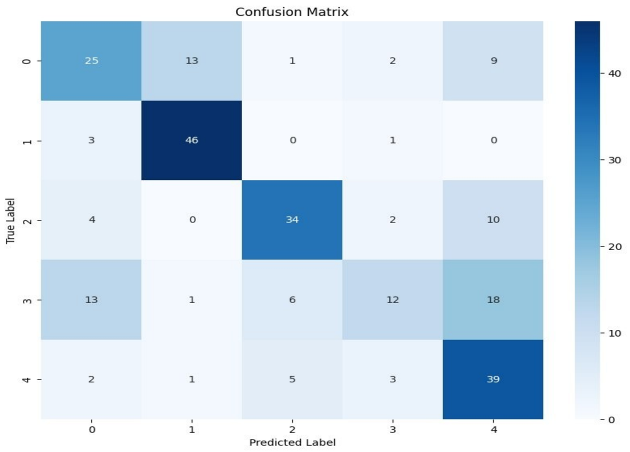


**Figure 4:** Confusion matrix.

However, the model struggles to correctly classify Abstract Expressionism (labeled as 0) and Naive Art / Primitivism (labeled as 3).

The model also exhibits confusion between similar styles, particularly:

- Abstract Expressionism (0) and Color Field Painting (1) – Since both are substyles of Abstract Art, they share overlapping visual characteristics, making it challenging the model to distinguish between them accurately.
- Mannerism – Late Renaissance (2) and Post-Impressionism (4) – These styles also show some degree of misclassification, likely due to similarities in color palettes, brushwork, or composition techniques.

To reduce misclassifications, a more refined feature extraction process is necessary, as was already discussed earlier.

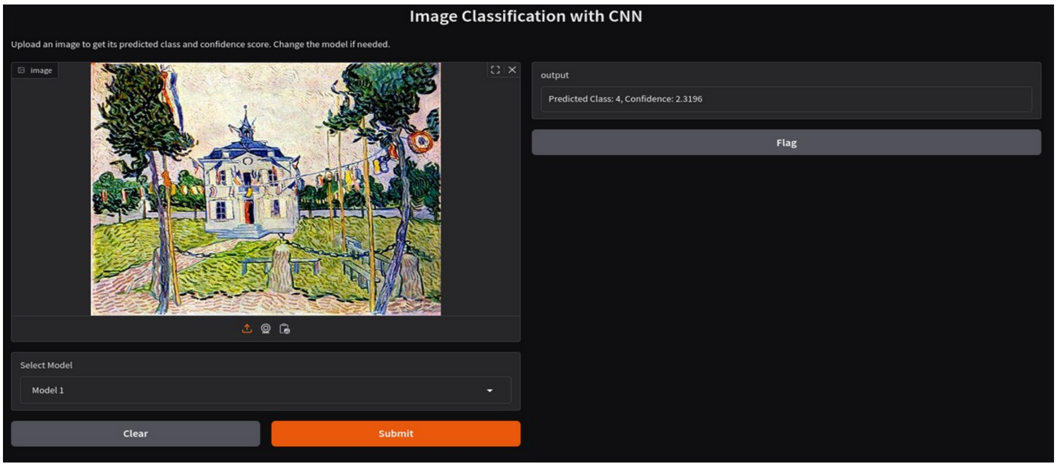


Image Classification with CNN

Upload an image to get its predicted class and confidence score. Change the model if needed.

output: Predicted Class: 4, Confidence: 2.3196

Select Model: Model 1

**Figure 5:** Example of the working solution.

The result, shown in Fig. 5, is a prototype interface demonstrating the practical application of the solution. Built using Gradio [10], the interface consists of three main elements:

- Input field – allows users to select an image for analysis
- Model selection field – enables choosing between different trained models
- Output field – displays the predicted artistic style based on the selected image and model.

## 6. Conclusions

Image style recognition remains an underexplored yet highly promising field. Most existing research focuses on theoretical aspects, with only a few practical solutions available – many of which can classify only 2 to 3 styles or require large datasets for training.

One of the key stages is image preprocessing, which includes pixel normalization and data augmentation to improve the generalization ability of the model. These measures reduce the probability of overtraining and ensure better adaptation of the model to new, unknown data.

An important element is the use of convolutional neural networks, which automatically highlight important features of images, in particular, different textures, edges, colors and other stylistic characteristics, which is necessary for the classification of artistic styles. The use of multiple layers of convolution and subsampling allows you to effectively reduce the dimensionality of the input data and increase the recognition accuracy.

Of particular note is the role of fully connected layers, which perform the final classification based on activations from previous layers, allowing for accurate image style identification. To improve the results, the ReLU activation function and the Adam optimization method were used, which provide fast learning and accurate predictions.

The study presented a working method for recognizing the artistic style of images across 5 styles, achieving an accuracy of 0.63. An in-depth analysis of the model's architecture was conducted, and various approaches to pre-training the data were discussed to enhance the model's performance

Additionally, the training and validation results were thoroughly discussed, highlighting the model's performance on both training and validation datasets. The study also displayed the practical application of the solution, demonstrating how the model can be used in real-world scenarios for artistic style recognition.

While the proposed solution is not perfect, it represents a functional approach that can be further optimized for specific applications. Additionally, this paper outlines several strategies to enhance model accuracy. Future work can build on these foundations to develop more robust and efficient style recognition systems.

## Acknowledgements

The study was created within research topic "Methods and means of artificial intelligence to prevent the spread of tuberculosis in war-time"(№0124U000660), which is carried out at the Department of Artificial Intelligence Systems of the Institute of Computer Sciences and Information of technologies of the National University "Lviv Polytechnic".

## Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and DeepL in order to: Grammar and spelling check, translation. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

## References

[1] C. Li, M. Wand, Precomputed real-time texture synthesis with markovian generative adversarial networks,in: European conference on computer vision ECCV 2016, 2016. doi: 10.1007 / 978-3-319-46487-9_43

[2] G.Team, Gradio, 2025. URL: https://www.gradio.app/. Last visited 02/04/2025.

[3] L. A. Gatys, A. S. Ecker, M. Bethg, A Neural Algorithm of Artistic Style, Journal of Vision Vision Sciences Society Annual Meeting Abstract , Vol.16, p. 326, 2016. doi: https://doi.org/10.1167/16.12.326

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, Image to-image translation with conditional adversarial networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. doi: 10.1109/CVPR.2017.632

[5] N. Boyko, A. Bronetskyi, N. Shakhovska, Application of Artificial Intelligence Algorithms for Image Processing, in: Workshop Proceedings of the 8th International Conference on “Mathematics. Information Technologies. Education”, MoMLeT&DS-2019, Vol-2386, 2019, pp. 194-211,.

[6] N. Boyko, B. Mandych, Technologies of Object Recognition in Space for Visually Impaired People”, The 3rd International Conference on Informatics & Data-Driven Medicine (IDDM 2020), Växjö, Sweden, November 19-21, CEUR, 2020, pp. 338-347

[7] N. Boyko, N. Tkachuk, Processing of Medical Different Types of Data Using Hadoop and Java MapReduce, in: The 3rd International Conference on Informatics & Data-Driven Medicine (IDDM 2020), 2020, pp. 405-414.

[8] S. Ioffe, Ch. Szegedy, Batch Normalization: Accelerating Deep Network Training by reducing Internal Covariate Shift, in: Proceedings of the 32 nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP, 2015, Vol. 37, pp. 448-456

[9] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778

[10] J. Johnson, A. Alahi, F.F. Li, Perceptual Losses for Real-Time Style Transfer and Super Resolution, in: European Conference on Computer Vision ECCV 2016, 2016, pp. 694-711. doi: 10.1007/978-3-319-46475-6_43

[11] D. P. Kingma, J. Ba, Adam. A Method for Stochastic Optimization, in: 3rd International Conference for Learning Representations, San Diego, 2015. URL: https://arxiv.org/abs/1412.6980

[12] Open WikiArt.org - Visual Art Encyclopedia, 2025. URL: https://www.wikiart.org/. Last visited 02/04/2025

[13] I. J.Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative Adversarial Nets, in: NeurIPS Proceedings Advances in Neural Information Processing Systems 27, 2014. URL: arXiv: 1406.2661

[14] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, Image to-image translation with conditional adversarial networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. doi: 10.1109/CVPR.2017.632

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, Image to-image translation with conditional adversarial networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. doi: 10.1109/CVPR.2017.632