

APPLICATION OF THE VGG-16 CONVOLUTIONAL NEURAL NETWORK FOR THE INTELLIGENT ANALYSIS OF FOREST FIRE IMAGERY

*T. H. Nguyen*¹, nguen_t@mirea.ru,

*T. L. Nguyen*¹, nguen@mirea.ru,

*A. G. Shmeleva*¹, shmeleva_a@mirea.ru,

¹MIREA – Russian Technological University, Moscow, Russian Federation

Deep learning application projects face numerous challenges, with many problems requiring optimal solutions to improve system performance. We can focus on a specific step or the entire process. The goal of this work is to demonstrate a novel approach using convolutional neural networks for intelligent image processing. Due to the impact of climate change, natural disasters are becoming more complex and have severe consequences. In particular, the phenomenon of forest fires is influenced by factors such as humidity, temperature, vegetation characteristics, etc., making their detection and prediction especially challenging. In this study, we focus on using image processing techniques and visual algorithms for data preprocessing and noise filtering to create a comprehensive database for model training and testing. Natural elements are also combined with object features (forest fire images) to build feature vectors. This research will contribute to technological advancement and create new opportunities for subsequent applications.

Keywords: deep learning; forest fire; image processing; VGG-16; convolutional neural network; artificial intelligence.

Introduction

In the current period, marked by rapid technological advancement and the automation of nearly every human process, it is unsurprising that computer vision once the realm of science fiction writers has evolved to a point where it is no longer a future concept, but a tangible reality. The swift development of image processing theories has given rise to sophisticated computer vision technologies. In essence, computer vision is a technology that enables equipment to recognize industrial processes, analyze data, and make informed decisions all within a fraction of a second [1, 2].

Machine learning-based approaches are utilized in fire detection applications. In this study [3], the use of a color model for flame detection is proposed. The authors demonstrated the importance and strengths of this approach compared to other methods. A fire pixel can be identified based on color pixels that fall within a predefined intensity value range. Typically, there are pixels similar to fire-color pixels, such as orange, sunset, sunlight, etc. There is a need to differentiate these pixels into fire and non-fire pixels. Color models will assist in selecting among them to propose a new method.

As AUV technology advances, scientists are transitioning to their use for forest fire monitoring [4]. The cost-effectiveness and suitability of this technology for natural environmental factors, as well as topography, have led to its growing application and popularity. Fractal codes have been proposed [5] for use in smoke detection systems. By defining the properties of the smoke zone, they delineate the boundary between areas. This task is similar to the process of classifying objects in an image. The primary objective

is to identify smoke characteristics to improve alert levels and predict fire direction. Kruskal's rule set is used to construct a minimum spanning tree and adjust the criteria accordingly. The variability between adjacent superpixels is applied to the development method, ensuring certain global properties are maintained despite locally made decisions. However, controlling the desired number of superpixels with this method is challenging [6].

Traditional forest fire detection using temperature sensors is susceptible to environmental factors such as spatial altitude and width, ventilation speed, dust, humidity, etc., resulting in a high false alarm rate. Image-based fire detection features non-contact sensing and has emerged as a new research direction in fire detection technology. Existing image-based fire detection primarily relies on algorithms such as Support Vector Machines (SVM), which utilize the principle of structural risk minimization and balance training errors with generalization capability [7]. This allows them to address practical challenges of small sample sizes, nonlinearity, high dimensionality, and local minima [8], making them a prominent area in machine learning [9, 10]. Several deep learning techniques have been implemented for forest fire detection [11, 12]. However, neural network algorithms carry risks of overfitting, slow convergence, and a tendency to fall into local minima [13], which can reduce the sensitivity and stability of fire detection. Therefore, we propose an optimization method to overcome these shortcomings. This method involves enhancing image segmentation in the preprocessing stage to improve the efficiency, accuracy, and speed of forest fire detection. The work in [14] describes using a neural feature extractor to generate a feature set for a neural network-based character and digit recognition system. The results are then compared with those obtained previously using a similar application and a PCA-based feature extraction method. In [15], six machine learning algorithms for classifying different pedestrian events are applied, based on signals from an inertial measurement unit and a global positioning system.

1. Methodology

1.1. Dataset and Data Preparation Process

The dataset comprises satellite imagery, ground-based observation images, combined with natural elements to construct feature vectors. Data is divided into two classes—fire images (showing flames, smoke, and burning zones) and non-fire images (normal forest scenes under various conditions).

The preprocessing pipeline begins with size normalization, where all input images are resized to $224 \times 224 \times 3$ pixels to ensure compatibility with the VGG-16 architecture. Following this, noise filtering techniques are applied to remove unwanted artifacts while simultaneously integrating environmental factors such as humidity, temperature, and vegetation characteristics with object features to enhance the quality of the input data. To further enrich the dataset and mitigate the risk of overfitting, data augmentation methods are employed, including random rotation, horizontal flipping, brightness and contrast adjustment, and the addition of Gaussian noise. Once preprocessing is complete, the dataset is systematically partitioned into three subsets: 70% for training, 15% for validation, and 15% for testing.

1.2. VGG-16 Architecture for Forest Fire Detection

A convolutional neural network is a specialized form of feedforward neural network, where the term “feedforward” refers to the organization of neurons within the network into distinct groups called layers. When such a multi-layered neural network processes data, the activation of each layer—the values of these neuron variables – is computed sequentially: first the activation values for the first layer, followed by the second layer, and so on until the final layer. The activation results of the last layer serve as the final output of the convolutional neural network.

The forest fire detection process using CNNs involves three key stages. First, the VGG-16 network is configured with appropriate parameters, loss function, and optimizer. Second, image data is collected from satellites and UAVs, then preprocessed through normalization, noise removal, and augmentation. Finally, the model is trained on labeled data using backpropagation to recognize fire characteristics and evaluated on test data to assess performance.

The VGG-16 convolutional neural network is a deep architecture consisting of 13 convolutional layers, 5 pooling layers, and 3 fully connected layers. A key feature of this architecture is the use of small 3×3 filters in all convolutional layers, which reduces the number of parameters while maintaining the depth of the network.

$$y = \sigma(Wx + b), \quad (1)$$

where x – input image, W – weight filter, b – shift, σ – activation function ReLU.

Theorem 1. *The feature extraction capability of VGG-16 improves with increasing network depth, as it combines simple features into complex hierarchical representations.*

Proof. The proof is based on an analysis of the loss function and the backpropagation process. During network training, gradients propagate through all layers, enabling the adjustment of weights for optimal extraction of features at different levels of abstraction.

□

Lemma 1. *The use of small 3×3 filters in successive convolutional layers is equivalent to applying a single large receptive field with fewer parameters.*

1.3. Forest Fire Image Processing

For the task of forest fire detection, input images are pre-processed and normalized to a size of $224 \times 224 \times 3$. The processing pipeline includes the following stages:

$$E_{total} = \frac{1}{2}(target - output)^2, \quad (2)$$

$$W = W_i - \eta \frac{dE_{total}}{dW}, \quad (3)$$

where $target$ – labeled, $output$ – network prediction, η – learning rate.

The backpropagation method consists of four distinct stages. It begins with forward propagation, where input data is passed through the network to generate an output. Next, the loss function computes the error between the predicted output and the actual target value. This is followed by backward propagation, which calculates the gradient of the loss

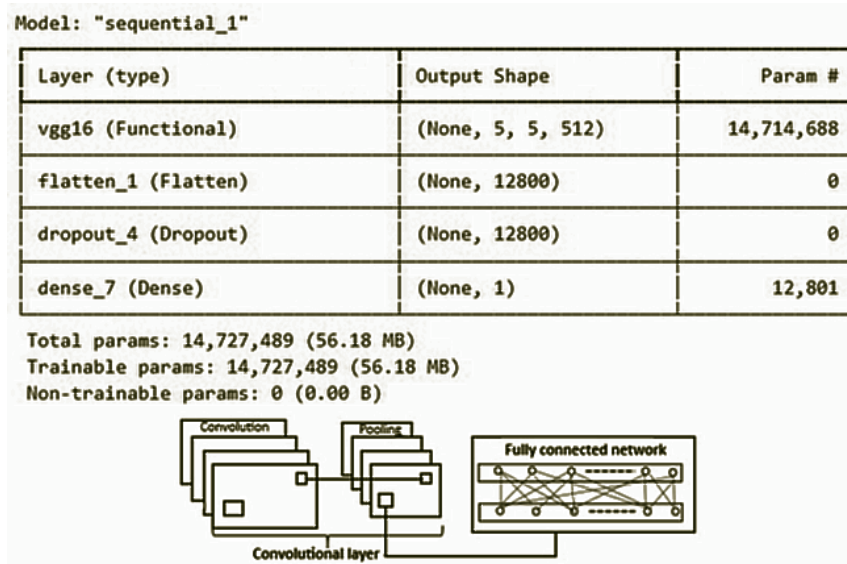


Fig. 1. VGG-16 Architecture

function with respect to each weight in the network. Finally, the weight update stage adjusts the network parameters using these gradients to minimize the loss, completing one iteration of the learning process.

The neural network training process begins with the forward propagation stage, where a training image is fed into the network. As we recall, this image is represented as a numerical matrix with dimensions $32 \times 32 \times 3$, where 32×32 defines the spatial resolution, and 3 corresponds to the number of color channels (red, green, blue). At the very beginning of training, when all weight coefficients and filter parameters have been initialized randomly, the network's output is uninformative. Instead of producing a meaningful prediction from the output layer—which, in a digit classification task, typically consists of ten neurons—the network generates a vector with roughly uniform values. Such an even output, where each digit is assigned a similar probability, demonstrates that the network in its current state cannot favor any particular class, as its internal parameters have not yet been adjusted to recognize any patterns in the data.

It is at this point that the concept of the loss function comes into play. For the network to learn anything, we use labeled training data where each image is paired with a corresponding reference label—the ground truth. Suppose our first training image contains the digit "3". In this case, its label would be a one-dimensional vector where all elements are zero except for the one at the position corresponding to the digit "3". This format, known as one-hot encoding, unambiguously indicates the true class of the object.

By comparing the network's ambiguous output with this reference label, the loss function calculates a numerical measure of the error, quantifying how far the network's prediction is from the truth. While the loss function can be expressed mathematically in various ways, one of the fundamental and widely used methods is the Root Mean Square Error (RMSE) (1). This function aggregates all discrepancies between the prediction and the target into a single metric of the network's total error, often denoted as E_{total} . This computed error forms the basis for the subsequent backpropagation stage, during which all network weights are adjusted to minimize the loss in subsequent iterations.

Backpropagation through the network determines which weights had the greatest influence on the loss and finds ways to adjust them to reduce the loss. All filters are updated so that the weights change in the direction of the error-reducing gradient, according to formula (2).

Essentially, we are minimizing the loss (error) generated by the convolutional neural network's processing. From a calculus perspective, we minimize the loss by adjusting the independent variables – the weights.

The VGG-16 model, a classical Convolutional Neural Network (CNN), is effectively applied to forest fire detection by analyzing images. Its primary advantage is its deep and uniform architecture, which enables the sequential extraction of visual features, from simple to complex.

The image processing pipeline for fire detection in VGG-16 is a feature extraction pipeline. First, the input image (typically sized 224x224x3) passes through a series of convolutional blocks. The initial layers, using 3x3 filters, extract basic features: edges, colors, and textures. In the context of fire, this could correspond to areas of orange/red color, smoke, or changes in the forest canopy structure. After each convolutional block, a max-pooling layer follows to reduce the data dimensionality. This decreases computational load and makes the model more robust to minor object shifts within the frame.

As the network deepens, subsequent convolutional layers combine simple features into more complex and abstract patterns: flame shapes, the structure of the burning area, or combinations of smoke and fire. After 13 convolutional and 5 pooling layers, the original image is transformed into a feature map. This map is then flattened into a one-dimensional vector and passed through three fully connected layers. These layers act as a classifier, aggregating all extracted features for the final prediction. The output generates a binary probability, for example, "FIRE" or "NO FIRE," allowing the system to automatically issue warnings.

```
def build_model_conv2():
    model = models.Sequential([
        layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
        layers.MaxPooling2D(2, 2),
        layers.Conv2D(64, (3, 3), activation='relu'),
        layers.MaxPooling2D(2, 2),
        layers.Conv2D(128, (3, 3), activation='relu'),
        layers.MaxPooling2D(2, 2),
        layers.Flatten(),
        layers.Dense(512, activation='relu'),
        layers.Dropout(0.5),
        layers.Dense(1, activation='sigmoid')
    ])
    return model
model_conv2 = build_model_conv2()
model_conv2.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Fig. 2. Detailed description of VGG-16 layers in the image processing system for forest fire detection

Convolutional layers are the core component of VGG-16, responsible for extracting visual features from input images for forest fire detection. Each layer uses a set of small 3x3 filters that scan the entire image. In the initial layers, these filters detect basic visual features: edges, colors, and textures. Specifically, for the fire detection task, they identify areas of orange and red color characteristic of flames, or gray-white smoke plumes. As the network deepens, subsequent convolutional layers combine these basic features into more complex patterns, such as the shape of active flames or the spreading structure of a

fire. After each convolution operation, a ReLU activation function is applied, allowing the model to learn non-linear relationships and significantly enhancing its ability to recognize complex fire patterns under real-world conditions.

In the VGG-16 architecture (Fig. 2) applied to forest fire detection, the Max-Pooling layer plays a crucial role in feature generalization. After every two or three convolutional layers, a Max-Pooling layer with a 2x2 window and a stride of 2 follows. This operation reduces the spatial dimensions of the feature map by half, retaining only the maximum value in each 2x2 region. In the context of fire detection, this means the most pronounced activations are preserved—for instance, the brightest flame regions or the densest smoke areas. This not only significantly reduces computational load and prevents overfitting but also increases the model’s robustness. By ignoring minor shifts in flame position or smoke shape, Max-Pooling forces the network to focus on the presence of critical features rather than their exact location, thereby improving the model’s generalization capability and its accuracy in real-world scenarios.

After the image passes through a series of 13 convolutional and 5 pooling layers, the data is transformed into a set of abstract high-level features. The purpose of the fully connected layers (typically three in VGG-16) is to perform classification based on these features. First, the three-dimensional feature map is flattened into a one-dimensional vector. This dense vector is fed into the first fully connected layer, where each neuron is connected to all inputs. Here, the network aggregates all learned patterns—from the color and texture of fire and smoke to complex fire shapes—to form a comprehensive representation. The two subsequent fully connected layers continue to refine and transform this information, learning complex non-linear relationships between the features. Finally, the output of the last fully connected layer (typically using a Softmax activation function) produces a probabilistic prediction, such as the probability of “forest fire present” or “no fire”, which forms the final output of the system.

Finally, with an excellent method for measuring the similarity of the composite image to the original, we need to update it so that the resulting image has a smaller difference—in other words, a lower loss function value—in the next iteration. Step by step, we will update the composite image so that it increasingly resembles both the content image and the style images.

We update it by subtracting the derivative (gradient) of the loss function from its current values. The derivative itself simply indicates the direction in which to adjust the image. This process is called cost minimization. In practice, we use nothing other than stochastic gradient descent, which aims to approach zero loss. For this, we will use Adam (Adaptive Moment Estimation)—a method that calculates adaptive learning rates for each parameter.

2. Performance Evaluation Results of the Method Proposed

The key parameters of the forest fire detection algorithm include: processing speed and accuracy in identifying fire images. Processing speed is measured by the number of frames that can be processed per unit time, as the execution time per frame depends on the computational capacity of the computer system. To obtain an overview of processing time, we will measure the average execution time for each input frame. Accuracy, on the other hand, is a more complex characteristic. On one hand, it depends on expert evaluation

of the fire detection results. On the other hand, the algorithm through each training iteration optimizes the weights to more accurately recognize fire characteristics according to the established parameters. Therefore, for the algorithm, accuracy is measured through the loss function. To avoid relying on subjective qualitative assessments, we will use this parameter to evaluate model quality.

This process is repeated over multiple epochs, enabling the network to gradually learn the characteristic features of fire such as flame color, smoke patterns, and burned area structure.

During training, the models are optimized based on two main metrics: accuracy and loss value. Accuracy reflects the proportion of correct classifications between the two classes (fire or no fire). After approximately 50 epochs, the values on the validation set begin to stabilize. To ensure training quality, a total of 100 epochs was selected for the entire process. No overfitting occurred, which would typically be manifested through a decline in accuracy on the validation set.

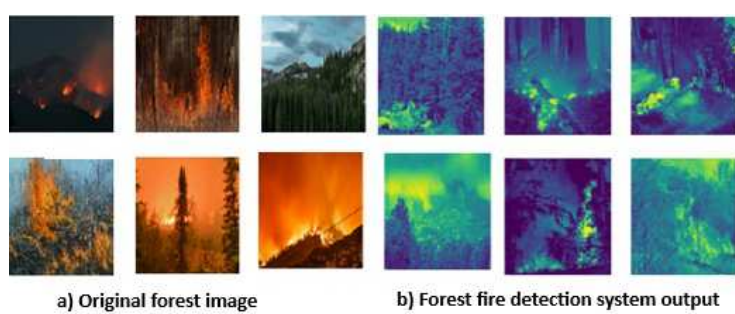


Fig. 3. Original forest image and forest fire detection system output

Evaluation results demonstrate that the model achieves high performance in forest fire detection, with stable accuracy on both the training set and test set. The decreasing loss value over epochs confirms that the model is learning effectively and converging toward an optimal set of weights.

To evaluate the performance of the developed algorithm, a series of tests must be conducted to verify the application's performance under realistic conditions.

During model training, the models were optimized using accuracy and loss metrics. Accuracy essentially describes the average proportion of the two groups (fire or no fire) that were correctly classified. The loss value is derived from the neural network's objective function and represents the error to be minimized. After approximately 50 epochs, the validation set values stabilized. However, to ensure the success of training, a total of 100 epochs were chosen for the entire training process. No overfitting was observed, which would have been indicated by a decline in accuracy. To illustrate this study, the following figure shows the training and validation accuracy, as well as the loss (Fig. 3), plotted against the epoch level for an example model of the original dataset.

To evaluate the algorithm's performance according to the required criteria, a test was conducted using different pairs of content and style images (see Fig. 5). During each test, the following metrics were recorded: execution time (Time spent), total loss (Total Loss), and the generated image was saved at every 50th iteration.

During the third test, 5000 iterations were completed, generating 101 images. The execution time per iteration was approximately 12 seconds. This is also clearly evident in

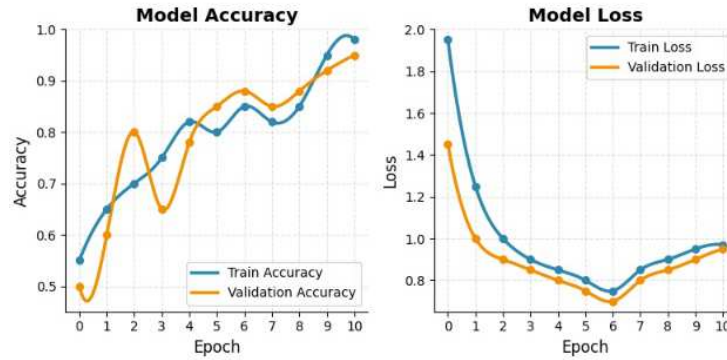


Fig. 4. Training and validation accuracy, as well as the loss of the proposed method

```

epochs=10
history = model_conv2.fit(
    train_ds,
    validation_data=val_ds,
    epochs=epochs
)

Epoch 1/10
38/38 [=====] - 130s 3.5s/step - loss: 0.3406 - accuracy: 0.8627

Epoch 2/10
38/38 [=====] - 54s 1s/step - loss: 0.1333 - accuracy: 0.9490 -

Epoch 3/10
38/38 [=====] - 55s 1s/step - loss: 0.1104 - accuracy: 0.9597 -

Epoch 4/10
38/38 [=====] - 61s 2s/step - loss: 0.0999 - accuracy: 0.9696 -

Epoch 5/10
38/38 [=====] - 53s 1s/step - loss: 0.0745 - accuracy: 0.9729 -

Epoch 6/10
38/38 [=====] - 56s 1s/step - loss: 0.0692 - accuracy: 0.9753 -
    
```

Fig. 5. VGG-16 performance evaluation results: accuracy and execution time

the test results presented in Fig. 6. Before drawing any conclusions about the algorithm’s performance during testing, it is essential to plot the total loss values for each iteration.

3. Conclusion

This research focuses on studying convolutional neural networks and developing novel methods for their application. The objective of the work defined its primary research direction—the development of a forest fire detection algorithm using convolutional neural networks to analyze and identify fire characteristics from input images. To achieve this goal, various current applications of convolutional neural networks were reviewed and analyzed. During this analysis, an area where such networks are underutilized was identified—specifically the problem of image-based forest fire detection—leading to the formulation of a novel approach and solutions for practical implementation.

The developed system is capable of analyzing input images and accurately predicting the presence of fire based on visual characteristics such as flame color, smoke patterns, and burned area structure. The system is computationally intensive, as it performs numerous complex operations during feature extraction and classification. Additionally, the system

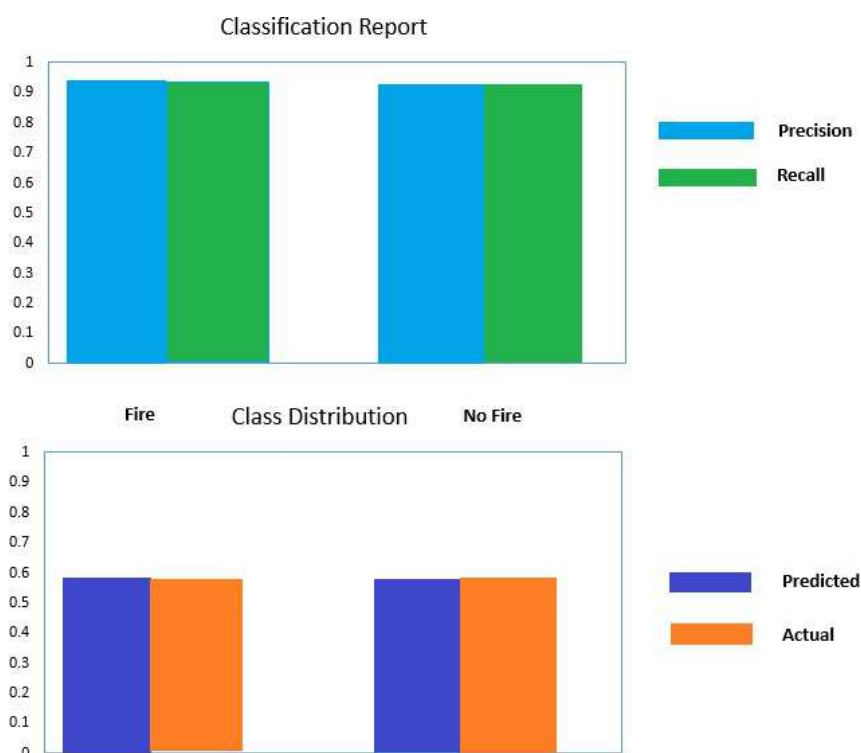


Fig. 6. Graph of total loss across test iterations

requires sufficient computer memory to load the convolutional neural network architecture along with its trained parameters. The results of this work also highlight potential directions for future development, such as reducing processing time and improving accuracy in early-stage fire detection.

References

1. Kulikov A.A. the Model Is a Reprint of an Object in the Image. *Russian Technological Journal*, 2020, vol. 8, no. 3, pp. 7–13. DOI: 10.32362/2500-316X-2020-8-3-7-13
2. Kulikov A.A. Application of Biometric Systems in Face Identification Technologies. *Russian Technological Journal*, 2021, vol. 9, no. 1, pp. 7–14. DOI: 10.32362/2500-316x-2021-9-3-7-14
3. Alves J., Soares C., Torres J.M., Sobral P., Moreira R.S. Automatic Forest Fire Detection Based on a Machine Learning and Image Analysis Pipeline. In: Rocha Á., Adeli H., Reis L.P., Costanzo S. (eds.) *New Knowledge in Information Systems and Technologies*. WorldCIST 2019. Advances in Intelligent Systems and Computing, Springer, Cham, 2019, vol. 931, pp. 240–251. DOI: 10.1007/978-3-030-16184-2_24
4. Arridge S., Hauptmann A. Networks for Nonlinear Diffusion Problems in Imaging. *Journal of Mathematical Imaging and Vision*, 2020, vol. 62, no. 3, pp. 471–487. DOI: 10.1007/s10851-019-00901-3

5. Bangare S., Dubal A., Bangare P., Patil S. Reviewing OTSU Method for Image Thresholding. *International Journal of Applied Engineering Research*, 2015, vol. 10, no. 9, pp. 21777–21783.
6. Bhanumathi V., Sangeetha R. CNN Based Training and Classification of MRI Brain Images. In: *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, Coimbatore, India, 2019, pp. 129–133. DOI: 10.1109/ICACCS.2019.8728453
7. Browne M., Ghidary S.S. Convolutional Neural Networks for Image Processing: An Application in Robot Vision. In: *Advances in Artificial Intelligence. AI 2020. Lecture Notes in Computer Science*, vol. 12568, pp. 641–652. Springer, Cham (2020).
8. Chen Q., Xu J., Koltun V. Fast Image Processing With Fully-Convolutional Networks. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 235–250. DOI: 10.1109/ICCV.2017.273
9. Nguyen T.H., Nguyen T.L., Sidorov D.N., Dreglea A.I. Machine Learning Algorithms Application to Road Defects Classification. *Intelligent Decision Technologies*, 2018, vol. 12, no. 1, pp. 1–8. DOI: 10.3233/IDT-170323.
10. Wang L. *Support Vector Machines: Theory and Applications*. Springer Verlag, Berlin, Heidelberg, 2005. DOI: 10.1007/b95439
11. Fujiwara N., Terada K. Extraction of a Smoke Region Using Fractal Coding. *IEEE International Symposium on Communications and Information Technology*, 2004. vol. 2, pp. 659–662. DOI: 10.1109/ISCIT.2004.1413797
12. Huang X., Shan J., Vaidya V. Lung Nodule Detection in CT Using 3D Convolutional Neural Networks. In: *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, Melbourne, Australia, 2017, pp. 379–383. DOI: 10.1109/ISBI.2017.7950542
13. Hui T.H.A., Xin L., Jie C., Hongyong Y. Multi-Type Flame Detection Combined With Faster R-CNN. *Journal of Image and Graphics*, 2019, vol. 24, no. 1, pp. 73–83.
14. Zhao J., Zhong Z., Han Z., Lu Y. SVM Based Forest Fire Detection Using Static and Dynamic Features. *Computer Science and Information Systems*, 2011, vol. 8, no. 3, pp. 821–841. DOI: 10.2298/CSIS101231030Z
15. Zhao W., Zhang H., Fu Y., Zheng J., Zhang X., He L. A Semantic Segmentation Algorithm Using FCN with Combination of BSLIC. *Applied Sciences*, 2018, vol. 8, no. 4, pp. 501–525. DOI: 10.3390/app8040501

*Huong Thu Nguyen, PhD (Tech), Associate Professor, Department of Informatics, MIREA – Russian Technological University (Moscow, Russian Federation),
nguen_t@mirea.ru*

*Long The Nguyen, PhD (Tech), Associate Professor, Department of Informatics, MIREA – Russian Technological University (Moscow, Russian Federation),
nguen@mirea.ru*

Anna G. Shmeleva, PhD (Tech), Associate Professor, Department of Informatics, MIREA – Russian Technological University (Moscow, Russian Federation), shmeleva_a@mirea.ru

Received December 22, 2025

УДК 519.25

DOI: 10.14529/jcem260104

ПРИМЕНЕНИЕ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ VGG-16 ДЛЯ ИНТЕЛЛЕКТУАЛЬНОГО АНАЛИЗА ИЗОБРАЖЕНИЙ ЛЕСНЫХ ПОЖАРОВ

Т. Х. Нгуен¹, Т. Л. Нгуен¹, А. Г. Шмелева¹

¹МИРЭА – Российский технологический университет, г. Москва, Российская Федерация

Существует множество вызовов для проекта применения глубокого обучения. Многие проблемы необходимо решать оптимальным образом для улучшения производительности системы. Мы можем сосредоточиться на конкретном шаге или на всем процессе. Цель работы – показать новый способ использования сверточных нейронных сетей для интеллектуальной обработки изображений. Из-за влияния изменения климата природные катастрофы становятся более сложными и имеют серьезные последствия. В частности, явление лесных пожаров подвержено влиянию факторов, таких как влажность, температура и характеристики растительности и т.д., поэтому обнаружение и прогнозирование лесных пожаров сталкиваются с многими вызовами. В этом исследовании мы сосредотачиваемся на использовании техник обработки изображений и визуальных алгоритмов для предварительной обработки данных и фильтрации шума, чтобы создать полную базу данных для обучения и тестирования модели. Естественные элементы также используются в сочетании с особенностями объектов (изображения лесных пожаров) для построения векторов признаков. Это исследование позволит развить технологию и открыть новые возможности для ее последующего применения.

Ключевые слова: глубокое обучение; лесной пожар; обработка изображений; VGG-16; сверточная нейронная сеть; искусственный интеллект

Нгуен Тху Хыонг, кандидат технических наук, доцент, кафедра информатики, МИРЭА – Российский технологический университет (г. Москва, Российская Федерация), nguен_t@mirea.ru.

Нгуен Тхе Лонг, кандидат технических наук, доцент, кафедра информатики, МИРЭА – Российский технологический университет (г. Москва, Российская Федерация), nguен@mirea.ru.

Шмелева Анна Геннадьевна, кандидат технических наук, доцент, кафедра информатики, МИРЭА – Российский технологический университет (г. Москва, Российская Федерация), shmeleva_a@mirea.ru.

Поступила в редакцию 22 декабря 2025 г.