This paper considers a model of object recognition in images using convolutional neural networks; the efficiency of the model-based process involving the training of deep layers in convolutional neural networks has been studied. There are objective difficulties associated with determining the optimal characteristics of neural networks, so there is an issue related to retraining a neural network. Eliminating the retraining by determining only the optimal number of epochs is insufficient since it does not provide high accuracy.

The requirements for the set of images for model training and verification have been defined. These requirements are better met by the INRIA image set (France).

GoogLeNet (USA) has been established to be a trained model that can perform object recognition on images but the object recognition reliability is insufficient. Therefore, it becomes necessary to improve the effectiveness of object recognition in images. It is advisable to use the GoogLeNet architecture to build a specialized model that, by changing the parameters and retraining some layers, could allow for better recognition of objects in images.

Ten models were trained using the following parameters: learning speed, the number of epochs, an optimization algorithm, the type of learning speed change, a gamma or power coefficient, a pre-trained model.

A convolutional neural network has been developed to improve the precision and efficiency of object recognition in images. The optimal neural network training parameters were determined: training speed, 0.000025; the number of epochs, 100; a power coefficient, 0.25, etc. A 3 % increase in precision was obtained, which makes it possible to assert the proper choice of the architecture for the developed network and the selection of its parameters. That allows this network to be used for practical tasks of object recognition in images

Keywords: image processing, object recognition, convolutional neural networks, unmanned aerial vehicle

UDC 004.932

DOI: 10.15587/1729-4061.2021.233786

IMPROVING A MODEL OF OBJECT RECOGNITION IN IMAGES BASED ON A CONVOLUTIONAL NEURAL NETWORK

Bogdan Knysh
Corresponding author
PhD, Associate Professor
Department of Electronics and Nanosystems*
E-mail: tutmos-3@i.ua
Yaroslav Kulyk
PhD, Associate Professor
Department of Automation and
Intelligent Information Technologies*
*Vinnytsia National Technical University
Khmelnytsky highway, 95,
Vinnytsia, Ukraine, 21000

Received date: 15.03.2021
Accepted date: 25.05.2021
Published date: 30.06.2021

***How to Cite**: Knysh, B., Kulyk, Y. (2021). Improving a model of object recognition in images based on a convolutional neural network. Eastern-European Journal of Enterprise Technologies, 3 (9 (111)), 40–50.doi: https://doi.org/10.15587/1729-4061.2021.233786*

1. Introduction

Image processing is extremely important in modern science and practice, so it is constantly evolving and improving. Image processing can be used in many industries, namely precision farming (agricultural monitoring), safety systems, quality control, etc. The given areas employ vision systems, robotic complexes, unmanned aerial vehicles (UAVs), video surveillance systems, web services, and mobile applications for identification and search.

One type of image processing is the recognition of objects in images, which is widely used in the industry, art, medicine, space technology, process management, automation, and many other fields [1]. Recognition of objects in images involves class attrition of the source data to a certain class by highlighting significant features. These attributes characterize the initial data from the general array of non-essential information.

There are many methods for recognizing objects in images, among which Random Forests techniques, boosting methods, as well as neural network procedures, specifically convolutional [2–6], are the most common.

Certain requirements are put forward for object recognition methods, namely:

– correspondence of the recognized object to the real object;
– high performance;
– resistance to errors;
– high accuracy.

Therefore, it becomes necessary to analyze the methods of object recognition in images and to choose the optimal according to the above requirements, specifically high accuracy. It is also worth considering the parameters that characterize these methods, changing which directly affects the precision, performance, and overall efficiency of the process of object recognition.

A modern relevant industrial area is the development of precision agriculture, which is based on the results from agricultural monitoring. These data, acquired from UAV video cameras, make it possible to assess the harvested crop, control the routes of movement of agricultural machinery, predict yields, etc. In this case, an important criterion is the UAV's ability to avoid collisions with close objects, determine the position in space, direction, and trajectory of the flight by receiving input data on the recognized objects.

The effectiveness of these systems is determined by the precision of object recognition whose evaluation requires experimental research.

2. Literature review and problem statement

A Random Forests method for recognizing many classes of objects is considered in paper [2]; it is characterized by high accuracy, resistance to retraining, and is easily accelerated when using parallel computations. However, unresolved issues

remain related to the lack of visual interpretation of the process and the complexity of explanations for their decisions, as well as high sensitivity to noise in images. That causes difficulties associated with high requirements for the absence of noise in the images and the inability to get an explanation of the result. Works [3, 4] show the results of object recognition in images using boosting methods, specifically Adaboost. High speed and efficiency of work, as well as adaptability to a specific application, are shown. However, there are difficulties associated with retraining in the presence of noise in the input data, a large number of image features, as well as the need for a significant amount of data for the training sample. This makes research costly and limits the use of these methods when working with low-quality images. Work [5] reports the results of object recognition in images using neural network methods. The ability to train the system to highlight key characteristics of objects from training sampling is shown. However, these methods require the use of an ensemble of neural networks, auxiliary methods for selecting the plot part of the image, as well as their architectures are extremely sensitive to external influences. The reason for this may be difficulties associated with the computational complexity and quality of preprocessing the initial and working data. That makes the use of these methods for certain tasks not effective. Work [6] provides the results of real problems of object recognition in images using neural network methods. It is shown that input data can be presented in any order, which does not affect the purpose of learning. However, these methods require taking into consideration a large number of parameters since images in real recognition tasks have large dimensionality. The reason for this may be difficulties caused by the requirements of a larger training sample. That increases the time and computational complexity of the learning process, which limits the application of this method.

An option to overcome the above difficulties associated with insufficient accuracy, efficiency, and performance may be the use of methods for recognizing objects in images based on convolutional neural networks [7, 8]. This is the approach used in work [9], which employs multispectral data acquired from a satellite while UAV video cameras provide multispectral data. In addition, a similar principle is implemented in work [10], where training parameters are analyzed and recommendations for changing the neural network architecture are provided; however, these recommendations are general in nature without analyzing specific applications, specifically for recognition tasks.

All this gives reason to assert that it is advisable to conduct a study into improving the effectiveness of training a neural network, which could significantly improve the precision of object recognition in images.

3. The aim and objectives of the study

The purpose of this work is to improve the model of a convolutional neural network in order to recognize objects in images and to select learning parameters for this network. That would make it possible to obtain a new neural network with increased precision for recognizing objects in images that could be used as a pre-trained neural network for other tasks.

To accomplish the aim, the following tasks have been set:

– to investigate neural network models based on the INRIA image set;

— to evaluate the Inria-9 model.

4. The study materials and methods

We studied the recognition of objects in images by using appropriate methods based on convolutional neural networks, taking into consideration the parameters of neural network learning. To test the effectiveness of these methods, the INRIA set was employed, which contains a large number of images with marked groups of pixels and defined classes of objects. INRIA contains images acquired from video cameras attached to UAV while being shot from a height of several hundred meters [11]. The study was carried out using the DIGITS programming environment involving the Caffe environment designed to deeply train a neural network taking into consideration the speed and modularity in the development of the model. The combination of these environments makes it possible to quickly train neural networks with deep layers and is used for the tasks of classification, segmentation of images [12], and recognition of objects on them. DIGITS contains a pre-trained GoogLeNet model, which is characterized by adapted parameters for recognizing objects in images (Tables 1, 2) and has a flexible architecture (Fig. 1).

The GoogLeNet architecture consists of 22 layers (27 layers when taking into consideration the merge layers) and part of these layers consists of 9 initial modules. Moreover, their parameters may change in the learning process. An image with an RGB palette of 224×224 is sent to the input. The filter size of the first layer is 7×7. The kernel size of 1×1×256 is used. The output activation function is Softmax, and in layers – ReLU, which makes it possible to increase performance by 6 times. Compared to similar models [13, 14], GoogLeNet contains 12 times less parameters, the network depth is increased to 22 layers without additional involvement of computing resources [15].

Thus, the GoogLeNet architecture was used as the base one to build a specialized FCN-GoogLeNet model by adding a fully linked convolution layer by making the following changes to DIGITS:

– we added a layer of data that receives training images and labels, and a conversion layer that applies real-time data magnification;

– we added a layer of normalization of data;

– we added a fully connected convolutional network (FCN), which removes the characteristics and forecasts object classes and field boundaries to a grid square;

– we added a layer of error, which simultaneously measures two values of forecasting;

– after determining the size of the input image, a random number is set, which determines how much the input image should be reduced;

– we added parameters to complement the data, which determine to what extent random conversions (pixel shifts, image flipping, etc.) should be applied to input images;

– we added a layer that uses a linear combination of two separate loss functions to calculate the total loss function for optimization;

– we deleted the layers of input and output data and a pooling layer [16].

The choice of the FCN-GoogLeNet model optimization algorithm is determined by the features of object recognition in images, for which it is necessary to have a good convergence of the algorithm, and for practical use – high performance.

Table 1

GoogLeNet model parameters

Type	Patch size/ pitch	Output size	Depth	#1×1	#3×3 before convolution	#3×3	#5×5 before convolution	#5×5	Filters 1×1	Weight	Mathematical operations
convolution	7×7/2	112×112×64	1	–	–	–	–	–	–	2.7K	34M
max pool	3×3/2	56×56×64	0	–	–	–	–	–	–	–	–
convolution	3×3/1	56×56×192	2	–	64	192	–	–	–	112K	360M
max pool	3×3/2	28×28×192	0	–	–	–	–	–	–	–	–
inception (3a)	–	28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)	–	28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0	–	–	–	–	–	–	–	–
inception (4a)	–	14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)	–	14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)	–	14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)	–	14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)	–	14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0	–	–	–	–	–	–	–	–
inception (5a)	–	7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)	–	7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0	–	–	–	–	–	–	–	–
dropout (40 %)	–	1×1×1024	0	–	–	–	–	–	–	–	–
linear	–	1×1×1000	1	–	–	–	–	–	–	1000K	1M
softmax	–	1×1×1000	0	–	–	–	–	–	–	–	–

Table 2

GoogLeNet model layer parameters

Type	Number of neurons	Total number of neuron connections in a layer	Number of links to the next layer
convolution	20M	402M	3K
max pool	1M	125M	3K
convolution	1.2M	944M	784
max pool	401K	78M	784
inception (3a)	200K	78M	784
inception (3b)	376K	295M	196
max pool	200K	18M	196
inception (4a)	100K	9M	196
inception (4b)	100K	19M	196
inception (4c)	100K	19M	196
inception (4d)	103K	20M	196
inception (4e)	163K	31M	49
max pool	81K	998K	49
inception (5a)	40K	1.9M	1
inception (5b)	1024	1,024	1
avg pool	25K	40	1
dropout (40 %)	1,024	1,024	1
linear	1,000	1,000	1
softmax	1,000	1,000	–

The comparison of algorithms [17] reveals that for the task of recognizing objects in images, Adam shows the best performance results (an increase of 10–50 %). That algorithm also demonstrates good convergence.

Fig. 1. GoogLeNet neural network model architecture: *a* – first part; *b* – second part

The main indicators of neural network training effectiveness, which were determined during our study, were chosen the following characteristics [18]:

– precision – the ratio of correctly recognized objects to the total number of predictable or true objects:

$$Precision_{val} = \sum_{k=1}^{N_{val}} \frac{N_{TP_k}}{N_{TP_k} + N_{FP_k}}, \quad (1)$$

where N_{TP} is defined as the number of correctly recognized objects in the image; N_{FP} is defined as the number of errone-

ously recognized objects; N_{val} is the number of images in the verification sample; k is the current image;

– recall – the ratio of correctly recognized objects to the total number of objects in the images:

$$Recall_{val} = \sum_{k=1}^{N_{val}} \frac{N_{TP_k}}{N_{TP_k} + N_{FN_k}}, \quad (2)$$

where N_{FN} is defined as the number of erroneously unrecognized objects;

– mean average precision – the simplified assessment of mathematical expectation based on the product of precision and recall, which shows how sensitive the network is to the right objects and resistant to errors:

$$mAP = Precision_{val} \times Recall_{val}. \quad (3)$$

To assess the effectiveness of neural network training, optimal neural network parameters are determined. These parameters are the duration of training (the number of epochs), the optimization algorithm (adaptive instant evaluation (Adam)), the type of change in the speed of learning, the coefficient gamma or power, the speed of learning (learning step), the pre-trained model. The combinations of parameters in the process of training six models are summarized in Table 3.

The Adam algorithm shows good optimization results, particularly in the duration of training, but does not always demonstrate satisfactory convergence [19]. Therefore, different values of the learning step were used to train the model with a balance between good convergence and duration of training (Table 3). With a good convergence of the model, the values of characteristics (1) to (3) are stable. Otherwise, the risk of retraining increases, and the values of characteristics (1) to (3) change dramatically in the learning process, which complicates the practical use of the model. Therefore, models with frequent and sharp drops in characteristic values (1) to (3) are not to be used as pre-trained models. For additional verification of the selected learning step values to ensure satisfactory convergence and lack of retraining, the model is to be tested on another set of images that were not used for the test sampling. The values of characteristics (1) to (3) on the new set should not differ significantly from the values obtained for the verification set of images, which also indicates the adequacy of the resulting model.

Table 3

Combinations of parameters for the training process

Trained model	Training duration (the number of epochs)	Optimization algorithm	The type of change in learning speed, gamma coefficient	Learning duration (learning step)	Based on
Inria-1	30	Adam	Exponential, 0.99	0.0001	GoogLeNet
Inria-2	30	Adam	Exponential, 0.99	0.000075	GoogLeNet
Inria-3	30	Adam	Exponential, 0.99	0.00005	GoogLeNet
Inria-4	30	Adam	Exponential, 0.99	0.000025	GoogLeNet
Inria-5	100	Adam	Exponential, 0.99	0.00001	GoogLeNet
Inria-6	100	Adam	Exponential, 0.99	0.000075	GoogLeNet

Our study was conducted on a test sample, which is a set of marked INRIA images. Features: 2 classes of objects; images in the form of color images with a resolution of 0.3 m with a total coverage of 810 km^2, of which 405 km^2 for training and 405 km^2 for verification.

The values of precision, recall, and mean accuracy estimates on the test sample should gradually increase. These parameters, and especially the assessment of mean accuracy, which includes precision and recall, characterize the adequacy of the model, that is, the correctness of neural network training and the lack of retraining. Validation of learning outcomes could be defined as a gradual increase in precision, recall, and assessment of mean accuracy on the test sample. The number of epochs of training is selected from the condition of obtaining the highest precision, recall, and assessment of mean accuracy on the test sample in the absence of significant fluctuations in numerical values. The expediency criterion for increasing the epochs of learning is a gradual increase in precision, recall, and assessment of mean accuracy on the test sample. The beginning of the drop in precision, recall, and assessment of mean accuracy on the test sample is a criterion for retraining, the absence of which is a condition for validating the model.

5. Results of studying the recognition of objects in images using convolutional neural networks

5. 1. Investigating neural network models for the recognition of objects in images from the INRIA set

Our study was conducted on pixelated images from the INRIA set. Since the dimensions of the images in the set are different, if one needs to test the model in a new image, one must mark up the existing objects. In addition, precision calculation is carried out in soft real time, which requires a high performance from a neural network with limited memory to ensure high accuracy. The neural network input image comes with an RGB (256 color palette) no larger than 5,000×5,000 pixels at a resolution of 30 cm. That corresponds to the surface with an area of up to 1,500×1,500 m. The output image is formed in TIFF or GeoTIFF format. The batch size is 32 with the number of threads equal to 4. Models are imported in prototxt or protobuf format. The recognition time should not exceed 50 ms for a Full HD image.

The model performance check is illustrated by charts that were constructed automatically in the DIGITS programming environment based on the specified parameters given in Table 3. Caffe environment was used for hardware acceleration of training. The number of values of precision, recall, and assessment of mean accuracy is equal to the number of epochs of learning.

Fig. 2 shows the Inria-1 performance test chart.

Fig. 2 shows that the values of precision, recall, and mean accuracy estimate gradually increase and acquire maximum value on learning epoch 16 learning. The values of precision, recall, and mean accuracy estimate are 69.91 %, 51.01 %, and 37.79 %, respectively.

Fig. 3 shows that the precision, recall, and mean accuracy estimate values increase and acquire their maximum values during learning epoch 23 for precision and learning epoch 22 for recall and mean accuracy estimate. The precision, recall, and mean accuracy estimate values for epoch 22 are 79.65 %, 70.90 %, and 57.80 %, respectively.

Fig. 3 shows the Inria-2 performance test chart.

Fig. 4 shows the Inria-3 performance test chart.

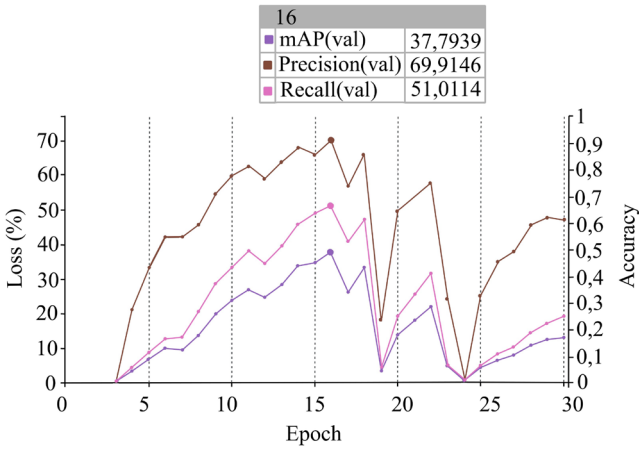

Fig. 2. Charts of change on the test sample depending on the epoch for the Inria-1 model:
– precision; – recall; – mean accuracy estimate

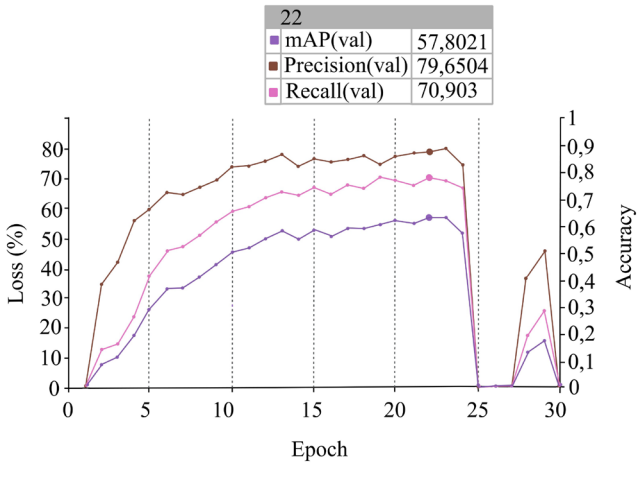

Fig. 3. Charts of change on the test sample depending on the epoch for the Inria-2 model:
– precision; – recall; – mean accuracy estimate

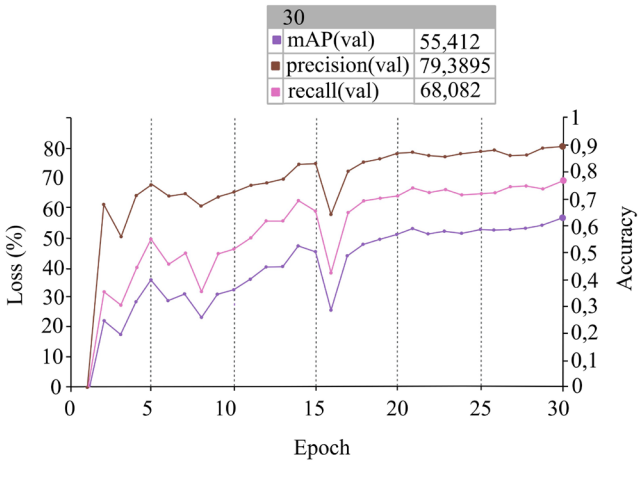

Fig. 4. Charts of change on the test sample depending on the epoch for the Inria-3 model:
– precision; – recall; – mean accuracy estimate

Fig. 4 shows that the values of precision, recall, and mean accuracy estimate gradually increase and acquire their maximum values during learning epoch 30. The precision, recall, and mean accuracy estimate values are 79.38 %, 68.08 %, and 55.41 %, respectively.

Fig. 5 shows the Inria-4 performance test chart.

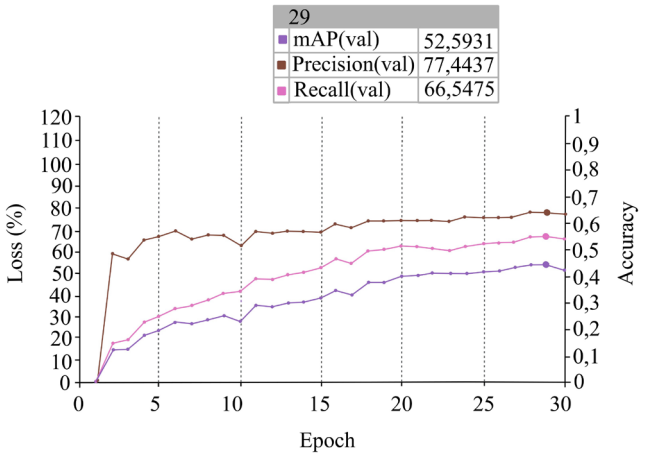

Fig. 5. Charts of change on the test sample depending on the epoch for the Inria-4 model:
– precision; – recall; – mean accuracy estimate

Fig. 5 shows that the values of precision, recall, and mean accuracy estimate gradually increase and acquire their maximum values during learning epoch 29. The precision, recall, and average accuracy estimate values are 77.44 %, 66.54 %, and 52.59 %, respectively.

Fig. 6 shows the Inria-5 model performance test chart.

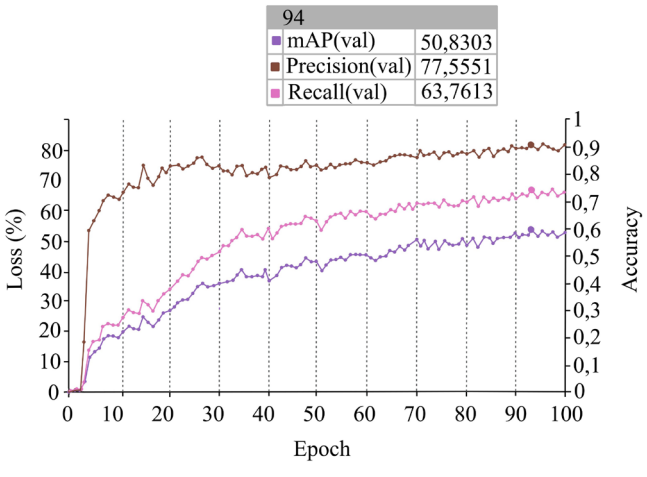

Fig. 6. Charts of change on the test sample depending on the epoch for the Inria-5 model:
– precision; – recall; – mean accuracy estimate

Fig. 6 shows that the values of precision, recall, and mean accuracy estimate gradually increase and acquire their maximum values during learning epoch 94. The precision, recall, and mean accuracy estimate values are 77.55 %, 63.76 %, and 50.83 %, respectively.

Fig. 7 shows the Inria-6 performance test chart.

Fig. 7 shows that the precision, recall, and mean accuracy estimate values increase and acquire their maximum

values during learning epoch 23 for precision, and learning epoch 22 for recall and mean accuracy estimate. The precision, recall, and average accuracy values for learning epoch 22 are 79.65 %, 70.90 %, and 57.80 %, respectively.

The research results showing our findings regarding the effectiveness of the six models are given in Table 4.

Table 4 shows that the highest mean accuracy estimate in the absence of sharp jumps in the indicators is demonstrated by the Inria-3 model, 55.41 %, over 30 learning epochs at a learning speed of 0.00005.

Thus, Inria-3 was used as the basis for training the new Inria-7 model over 30 epochs with an exponential change in the learning speed, which is 0.000025, the gamma coefficient of 0.99, and the Adam optimization type.

Fig. 8 shows the Inria-7 performance test chart.

Fig. 8 shows that the values of precision, recall, and mean accuracy estimate gradually increase and acquire their maximum values during learning epoch 23. The precision, recall, and mean accuracy estimate values are 82.12 %, 72.69 %, and 60.77 %, respectively.

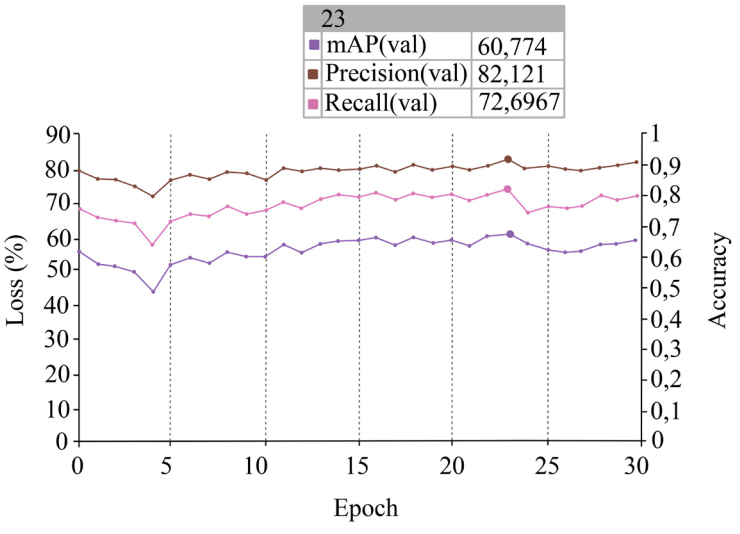

Fig. 8. Charts of change on the test sample depending on the epoch for the Inria-7 model: — precision; — recall; — mean accuracy estimate

Table 4

Results of exploring the effectiveness of models with different parameters

Trained model	Epoch with the best result/the number of epochs	Mean accuracy estimate, %	Precision, %	Recall, %
Inria-1	16/30	37.79	69.91	51.01
Inria-2	22/30	57.80	79.65	70.90
Inria-3	30/30	55.41	79.38	68.08
Inria-4	29/30	52.59	77.44	66.54
Inria-5	94/100	50.83	77.55	63.76
Inria-6	22/100	57.80	79.65	70.90

Thus, the mean accuracy estimate increased from 55.41 % to 60.77 %.

This model demonstrates good growth rates of the mean accuracy estimate and the stability of results, so it was used to train Inria-8 and Inria-9 training (a polynomial change in training speed) while learning duration increased to 100 (Table 5).

Table 5

Parameters that changed during the learning process

Trained model	Duration of learning (the number of epochs)	Optimization algorithm	Type of change in learning speed, gamma/power coefficient (for polynomial)	Learning speed	Based on
Inria-8	100	Adam	Exponential, 0.99	0.00001	Inria-7
Inria-9	100	Adam	Polynomial, 3	0.00005	Inria-7

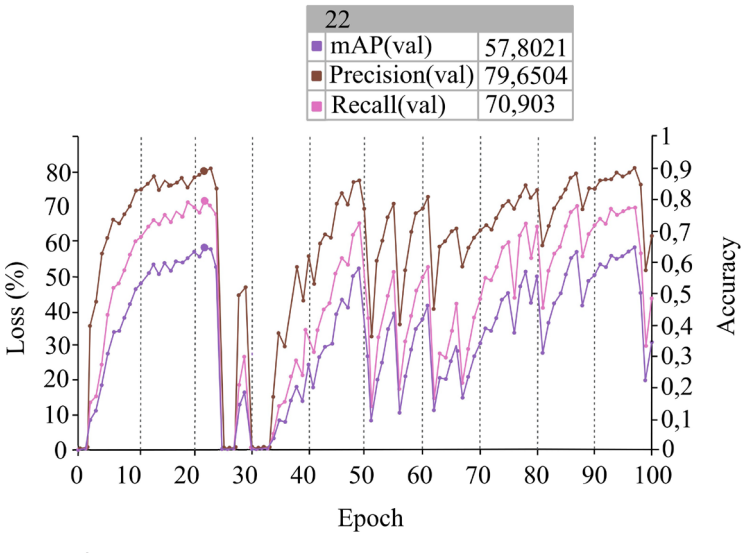

Fig. 7. Charts of change on the test sample depending on the epoch for the Inria-6 model: — precision; — recall; — mean accuracy estimate

Fig. 9 shows the Inria-8 performance test chart.

Fig. 9 shows that the values of precision, recall, and mean accuracy estimate gradually increase and acquire their maximum values during learning epoch 97. The precision, recall, and mean accuracy estimate values are 84.15 %, 74.00 %, and 63.22 %, respectively.

Fig. 10 shows the Inria-9 performance test chart.

Fig. 10 shows that the precision, recall, and mean accuracy estimate values increase and acquire their maximum values during learning epoch 24 for recall, and during learning epoch 45 for precision and mean accuracy estimate. The precision, recall, and mean accuracy estimate values for epoch 45 are 85.68 %, 75.59 %, and 65.70 %, respectively.

The research findings showing the results of verifying the effectiveness of the three models are given in Table 6.

Table 6 shows that the Inria-9 model demonstrates the highest mean accuracy estimate in the

absence of sharp jumps in the indicators. This is observed over 100 epochs with a polynomial change in the speed of learning, which is 0.00005, the power factor of 3, and the Adam type of optimization.

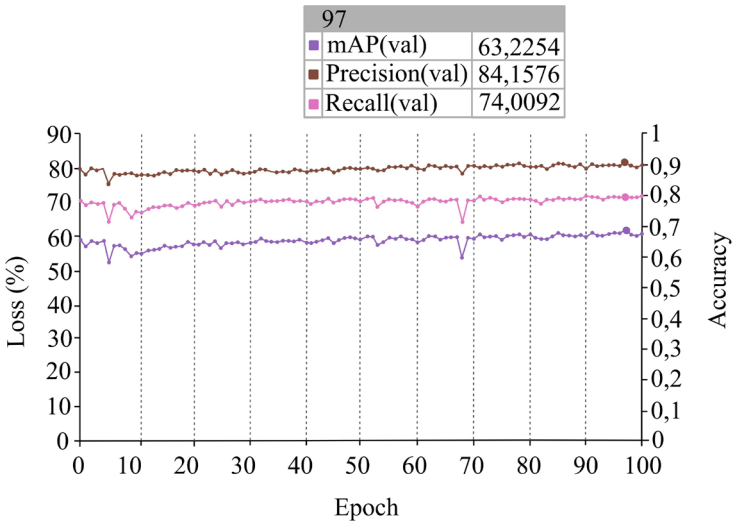

Fig. 9. Charts of change on the test sample depending on the epoch for the Inria-8 model: — precision; — recall; — mean accuracy estimate

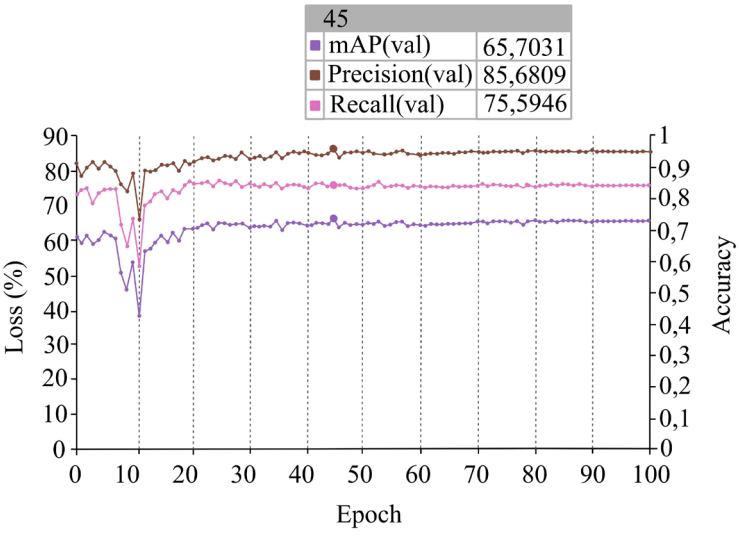

Fig. 10. Charts of change on the test sample depending on the epoch for the Inria-9 model: — precision; — recall; — mean accuracy estimate

Table 6

Results of exploring the effectiveness of models with different parameters

Trained model	Epoch with the best result/the number of epochs	Mean accuracy estimate, %	Precision, %	Recall, %
Inria-7	23/30	60.77	82.12	72.69
Inria-8	97/100	63.22	84.15	74.00
Inria-9	45/100	65.70	85.68	75.59

For this model, we managed to increase the mean accuracy estimate from 60.77 % to 65.70 %.

5. 2. Assessing the Inria-9 trained model for object recognition in images

In practice, object recognition in images is part of environmental monitoring with UAV that requires high accuracy in terms of control and orientation in space. Therefore, the model with the highest mean accuracy estimate, Inria-9, should be additionally trained using the values of parameters defined as optimal based on our study (Table 3):

– learning speed, 0.000025;

– the duration of learning (the number of epochs), 100;

– optimization algorithm, Adam;

– the type of change in the speed of learning, polynomial;

– power factor, 0.25;

– pre-trained model, Inria-7.

Thus, the Inria-10 model was built, the results of testing of which are shown in Fig. 11.

Fig. 11 shows that the values of precision, recall, and mean accuracy estimate gradually increase and acquire their maximum values during learning epoch 97. The precision, recall, and mean accuracy estimate values are 85.95 %, 79.26 %, and 68.78 %, respectively.

Our findings showing the results of the effectiveness test of all ten models are given in Table 7.

Table 7 shows that among the ten trained models, Inria-10 demonstrated the highest mean accuracy estimate. For this model, it was possible to increase the mean accuracy estimate from 55.41 % (Inria-3) to 60.77 % (Inria-7), then to 65.70 % (Inria-9) and, finally, to 68.78 %. A further increase could be achieved through experiments to change the neural network architecture, more diligent selection of images from the set, and a combination of training cycles on different data sets; that, however, requires significant computing resources.

Fig. 12 shows the recognition of buildings in images from the UAV camcorder for the Inria-10 model in the DIGITS programming environment.

The example (Fig. 12) allows us to conclude that the network recognizes almost all buildings. Structures such as sheds, greenhouses, unfinished buildings, as well as buildings that were partially present in the photo, were partially covered with trees or, due to their close location, were recognized as one building, remained unrecognized. The number of unrecognized buildings confirms the experimental accuracy of about 70 %. At the same time, there was no mistaken attrition of objects that are not buildings to the "building" class.

Thus, the operational quality of the Inria-10 model depends significantly on the visual dimensions of the desired object, lighting, shooting angle, the presence of objects that interfere with the inspection. However, with close-ups at good lighting, the buildings are almost guaranteed to be recognized. Therefore, a given model could be used to control farms, build orthophotoplans, draw up field maps, monitor territories, solve tasks related to cadaster and land management, etc.

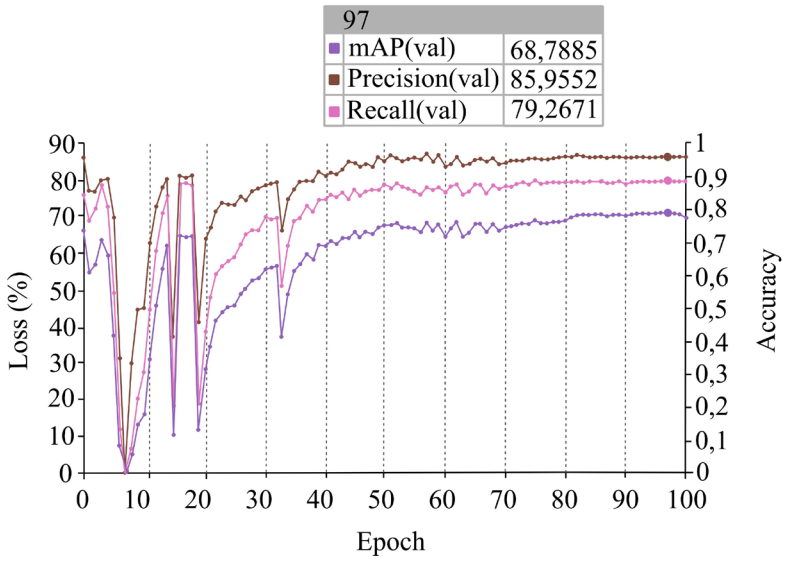

Fig. 11. Charts of change on the test sample depending on the epoch for the Inria-10 model: — precision; — recall; — mean accuracy estimate

Fig. 12. Example of recognizing a building by the Inria-10 model

Table 7

Results of exploring the effectiveness of models with different parameters

Trained model	Epoch with the best result/the number of epochs	Type of change in learning speed	Learning speed	Based on	Mean accuracy estimate, %
Inria-1	16/30	Exponential	0.0001	GoogLeNet	37.79
Inria-2	22/30	Exponential	0.000075	GoogLeNet	57.80
Inria-3	30/30	Exponential	0.00005	GoogLeNet	55.41
Inria-4	29/30	Exponential	0.00025	GoogLeNet	52.59
Inria-5	94/100	Exponential	0.00001	GoogLeNet	50.83
Inria-6	22/100	Exponential	0.000075	GoogLeNet	57.80
Inria-7	23/30	Exponential	0.000025	Inria-3	60.77
Inria-8	97/100	Exponential	0.00001	Inria-7	63.22
Inria-9	45/100	Polynomial	0.00005	Inria-7	65.70
Inria-10	97/100	Polynomial	0.000025	Inria-9	68.78

The achieved performance values make it possible to compare the Inria-10 model with others [2–10], but it makes sense to compare with models close in architecture that are obtained during training on a similar base. Therefore, the comparison was carried out according to the criterion for assessing mean accuracy estimate with some well-known GoogLeNet-based models, developed according to similar parameters, trained on the basis of images acquired from UAV cameras. GoogLeNet-like (Switzerland), InceptionResNetV2 (Turkey), U-Net InceptionResNetV2 (Turkey) were chosen as such models [20, 21]. The results of assessing the mean accuracy of the models are given in Table 8.

Table 8 shows that the highest mean accuracy estimate, namely 75 %, is demonstrated by the Inria-10 model, as others were trained based on images acquired from UAV cameras that were not part of the INRIA set.

Table 8

Models' precision assessment results

Model name	Mean accuracy estimate, %
GoogLeNet-like	70
InceptionResNetV2	72
U-Net InceptionResNetV2	73
Inria-10	75

We estimated the adequacy, reliability, and convergence between the Inria-10 model and others [20, 21]. To this end, the recognition of 100 manually marked images from the set NVidia Aerial Drone Dataset (USA) [22] was performed. This set is selected because the images in this set are acquired under different shooting conditions than in the INRIA set. The calculations were carried out in the Jupiter Notebook environment in the Python language. Averaged results are summarized in Table 9.

Table 9 shows that the highest mean accuracy estimate, namely 67 %, is demonstrated by the developed Inria-10 model. This indicates the high reliability of the Inria-10 model.

Table 9

Results of testing the reliability, adequacy, and convergence of models

Model name	Precision, %	Recall, %	Mean accuracy estimate (mAP), %
GoogLeNet-like	81	76	63
InceptionResNetV2	82	79	64
U-Net InceptionResNetV2	84	79	65
Inria-10	87	81	67

Objects in the images for verification were marked in certain classes, according to which the model recognizes the specified objects (buildings) in the image. In the experimental verification of models, the results were incorrectly positive (the presence of a certain class in the image in its absence) and falsely negative (the absence of a certain class in the image in its presence) when

designating classes. The share of images with incorrectly marked classes is 15 %, and the share of images with correctly marked classes is 85 %. These results make it possible, by using formulas (1) to (3), to calculate precision, recall, and mean accuracy estimate. The precision value is 87 %, which indicates the convergence of models. The resulting recall value, which is 81 %, indicates the reliability of the model. The mean accuracy estimate value is 67 %, which indicates the adequacy of the model.

6. Discussion of results of studying the recognition of objects in images using convolutional neural networks

The results of our study show that the Inria-10 trained model demonstrates the high accuracy of object recognition in images (Fig. 11). This is due to the choice of optimal parameters for the neural network, as well as the introduction of a convolutional layer into the standard neural network architecture. Inria-10 is based on Inria-9. This model has demonstrated the best mean accuracy estimate values (Table 6) in the learning process based on the INRIA set (Fig. 10). That is explained by the choice of a polynomial change in the speed of learning (Table 5). Therefore, it was Inria-9 that was chosen for additional training with optimal neural network parameters. The Inria-10 model built in this way could be used to recognize objects in real images (Fig. 12), the high accuracy of which determines the effectiveness of UAV control system. The results of the comparison of the mean accuracy estimate of object recognition in images for Inria-10 and other similar models are given in Table 8. Inria-10, compared to others, demonstrates high values of the mean accuracy estimate of object recognition in images, which indicates the adequacy of this model and no need for retraining it.

The accuracy and performance of the developed Inria-10 neural network model have higher values than similar ones reported in [20], by 2–4 % and 20–50 %, respectively. In this case, the recognition process does not require significant computing resources at the stage of using the model. Compared to [21], this model has a 3 % higher precision of object recognition in images. That was achieved by adding data layers, converting, normalizing data, error, calculating the mean error and parameters to complement the data, as well as FCN, and deleting layers of input/output data, and layer pooling. The reliability, adequacy, and convergence of the developed Inria-10 neural network model is comparable (Table 9) to other models [20, 21], and is not inferior to them. The value of precision is greater by 3–6 %, which indicates the convergence of the model. The recall value is greater by 2–5 %, which indicates the reliability of the model. The mean accuracy estimate value is higher by 2–4 %, which indicates the adequacy of the model.

Since the neural network model was trained for images from the INRIA set, high recognition precision values are typical of the images obtained from a drone's camera, usually due to the high contrast of pixel groups. For other types of images, precision probably won't be as high. That requires additional research.

The disadvantages include the cost of time and computing resources at the stage of training the neural network. This disadvantage could be overcome by using parallel graphic computing using CUDA technology and employing a more compact neural network as a pre-trained neural network, for example, MobileNet.

The advancement of a given model may be to further increase the precision, performance, as well as a decrease in computing resources. That would require sophisticated mathematical modeling, taking into consideration the subject area of application, and the development of software modules for a particular system.

7. Conclusions

1. We have investigated the models of Inria-1, Inria-2, Inria-3, Inria-4, Inria-5, Inria-6, Inria-7, Inria-8, Inria-9 neural networks based on the INRIA set. It was found that the largest mean accuracy estimate is demonstrated by the Inria-9 model, 68.78 %, at a training speed of 0.000025 based on Adam at a polynomial change in learning speed with a power coefficient of 0.25. The lowest mean accuracy estimate is 37.79 % for the Inria-1 model, which uses an exponential change in learning speed. That means that the greater learning accuracy is provided by a polynomial change in the speed of learning.

2. A mean accuracy estimate value has been obtained for the Inria-10 model, built on the basis of the pre-trained Inria-9 model for the recognition of objects in images from the INRIA set with the parameters defined during our study. This value is quite high as it gradually increases and acquires its maximum value during learning epoch 97. The precision, recall, and mean accuracy estimate values are 85.95 %, 79.26 %, and 68.78 %, respectively. The resulting values make it possible to assert the correctness of the choice of network architecture and the selection of parameters. That allows this model to be used for practical tasks of recognizing objects in images, for example, in autopilots, in collision avoidance systems with other UAVs, for machine vision, analysis of agricultural infrastructure, etc.

References

1. Bilinskiy, Y. Y., Knysh, B. P., Kulyk, Y. A. (2017). Quality estimation methodology of filter performance for suppression noise in the mathcad package. Herald of Khmelnytskyi national university, 3, 125–130. Available at: http://ir.lib.vntu.edu.ua/bitstream/handle/123456789/23238/47857.pdf?sequence=2&isAllowed=y
2. Gall, J., Razavi, N., Van Gool, L. (2012). An Introduction to Random Forests for Multi-class Object Detection. Outdoor and Large-Scale Real-World Scene Analysis, 243–263. doi: https://doi.org/10.1007/978-3-642-34091-8_11
3. Viola, P., Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. doi: https://doi.org/10.1109/cvpr.2001.990517
4. Weiming Hu, Wei Hu, Maybank, S. (2008). AdaBoost-Based Algorithm for Network Intrusion Detection. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 38 (2), 577–583. doi: https://doi.org/10.1109/tsmcb.2007.914695

5. Shang, W., Sohn, K., Almeida, D., Honglak, L. (2016). Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units. Proceedings of The 33rd International Conference on Machine Learning, 48, 2217–2225. Available at: http://proceedings.mlr.press/v48/shang16.html
6. Simonyan, K., Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR. Available at: https://arxiv.org/pdf/1409.1556.pdf
7. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: https://doi.org/10.1109/cvpr.2016.91
8. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D. et. al. (2015). Going deeper with convolutions. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: https://doi.org/10.1109/cvpr.2015.7298594
9. Prathap, G., Afanasyev, I. (2018). Deep Learning Approach for Building Detection in Satellite Multispectral Imagery. 2018 International Conference on Intelligent Systems (IS). doi: https://doi.org/10.1109/is.2018.8710471
10. Wu, K., Chen, Z., Li, W. (2018). A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks. IEEE Access, 6, 50850–50859. doi: https://doi.org/10.1109/access.2018.2868993
11. Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P. (2017). Can semantic labeling methods generalize to any city? The inria aerial image labeling benchmark. 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). doi: https://doi.org/10.1109/igarss.2017.8127684
12. Knysh, B., Kulyk, Y. (2021). Development of an image segmentation model based on a convolutional neural network. Eastern-European Journal of Enterprise Technologies, 2 (2 (110)), 6–15. doi: https://doi.org/10.15587/1729-4061.2021.228644
13. Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems, 1097–1105. Available at: https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
14. Zeiler, M. D., Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. Lecture Notes in Computer Science, 818–833. doi: https://doi.org/10.1007/978-3-319-10590-1_53
15. Deep Learning: GoogLeNet Explained. Towards Data Science. Available at: https://towardsdatascience.com/deep-learning-googlenet-explained-de8861c82765
16. Tao, A., Barker, J., Sarathy, S. (2016). DetectNet: Deep Neural Network for Object Detection in DIGITS. NVidia developer blog. Available at: https://developer.nvidia.com/blog/detectnet-deep-neural-network-object-detection-digits
17. Kingma, D. P., Ba, J. (2015). Adam: a method for stochastic optimization. ICLR 2015. Available at: https://arxiv.org/pdf/1412.6980.pdf
18. Kvetny, R. N., Masliy, R. V., Kyrylenko, O. M. (2020). Detection and classification of traffic objects using the environment digits. Optoelectronic Information-Power Technologies, 1 (39), 14–20. doi: https://doi.org/10.31649/1681-7893-2020-39-1-14-20
19. Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., Recht, B. (2017). The marginal value of adaptive gradient methods in machine learning. 31st Conference on Neural Information Processing Systems (NIPS 2017). Available at: https://arxiv.org/pdf/1705.08292v2.pdf
20. Guo, Z., Chen, Q., Wu, G., Xu, Y., Shibasaki, R., Shao, X. (2017). Village Building Identification Based on Ensemble Convolutional Neural Networks. Sensors, 17 (11), 2487. doi: https://doi.org/10.3390/s17112487
21. Erdem, F., Avdan, U. (2020). Comparison of Different U-Net Models for Building Extraction from High-Resolution Aerial Imagery. International Journal of Environment and Geoinformatics, 7 (3), 221–227. doi: https://doi.org/10.30897/ijegeo.684951
22. Nvidia Aerial Drone Dataset. Available at: https://nvidia.box.com/shared/static/ft9cc5yjvrbhkh07wcivu5ji9zola6i1.gz