Date of publication SEP-30, 2021, date of current version AUG-11, 2021.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.20.3.2288

Comparison of Semantic Convolution Neural Networks on the Example of Crack Segmentation in Asphalt Images

Svetlana Mustafina[1], Andrey Akimov[2], Sofia Mustafina[3], Alexandra Plotnikova[4]

[1]Bashkir State University, Boulder, Republic of Bashkortostan, Ufa, 453103, Russia (e-mail: mustafina_sa@mail.ru)
[2]Bashkir State University, Republic of Bashkortostan, 453128, Russia (e-mail: andakm@yandex.ru)
[3]Bashkir State University, Republic of Bashkortostan, Ufa, 450076, Russia (e-mail: sofjamustafina@gmail.com)
[4]Southwest State University, Kursk, Russia e-mail: (e-mail: aleksanklyu@yandex.ru)

Corresponding author: Andrey Akimov (e-mail: andakm@yandexu.ru).

This study was performed in the framework of state assignment FZWU-2020-0027.

ABSTRACT The article is devoted to a comparative analysis of the effectiveness of convolutional neural networks for semantic segmentation of road surface damage marking. Currently, photo and video surveillance methods are used to control the condition of the road surface. Assessing and analyzing new manual data can take too long. Thus, a completely different procedure is required to inspect and assess the state of controlled objects using technical vision. The authors compared 3 neural networks (Unet, Linknet, PSPNet) used in semantic segmentation using the example of the Crack500 dataset. The proposed architectures have been implemented in the Keras and TensorFlow frameworks. The compared models of neural convolutional networks effectively solve the assigned tasks even with a limited amount of training data. High accuracy is observed. The considered models can be used in various segmentation tasks. The results obtained can be used in the process of modeling, monitoring, and predicting the wear of the road surface.

KEYWORDS cracks; defect pavement; Dice; image classification; IoU; LinkNet; PSPNet; road pavement; segmentation; U-Net.

I. INTRODUCTION

Continuous video monitoring of the road surface can be an extremely tedious task for humans, but a straightforward task for automated computer vision (CV) systems. As noted in [1], transport infrastructure is the basis of the national economy, which needs to be systematically improved.

Many researches are devoted to improving algorithms for detecting road defects. Depending on the method of road surface monitoring, defects can be detected both on two-dimensional images (2D) and on three-dimensional (3D) images [2] obtained by laser scanning in the form of a point cloud. Compared to two-dimensional (2D) pavement images, three-dimensional pavement data is less vulnerable to lighting conditions and provides more useful information. In addition, 2D methods cannot detect some defects due to the lack of depth information.

The existing algorithms for visual detection of defects on the road can be conditionally divided into two branches: traditional methods [3] for detecting defects and methods of artificial intelligence [4].

Until recently, mainly manual monitoring techniques were used to solve these problems, such as [5, 6]:

- image thresholding [5],
- morphological operations [6],
- analysis of geometric features [7],
- application of Gabor filters [8],
- wavelet transforms [9],
- building histograms-oriented gradients (HOG) [10],
- texture analysis.

These methods are usually based on photometric and geometric hypotheses about the properties of fracture images. The most distinctive photometric property is that the pixels belonging to the crack are the darker pixels in the image. Based on this, a global or local threshold for pixel

brightness for fracture segmentation can be determined. However, these approaches are very sensitive to noise as they are executed on individual pixels. To solve this problem, the geometric characteristics of the damage are considered. For example, the property of fracture continuity is considered to reduce the likelihood of false detection. And based on the local orientation of a pixel, a local binary operator can be constructed to determine whether a specified pixel belongs to a crack [11]. Another method, using wavelet transform, is used to separate regions. In this case, for crack detection, frequency bands are used to separate the cracked and non-cracked regions, and high and low amplitudes are identified as cracks and noise, respectively.

The latest methods that have been proposed to achieve accuracy in detecting cracks are methods based on the minimum path problem [12]: the minimum path problem is to find the shortest path between the nodes of the graph. Several methods have been proposed based on the minimum path principle. In particular, an algorithm has been developed that can find a curve without knowing the endpoints or topology of the curve. All of these methods use the degree of image brightness gradation and the notion of graph connectivity to detect cracks.

All of these considered methods are effective for detecting cracks, but they are not effective enough for detecting all existing cracks in the image. More importantly, these traditional algorithms tend to only detect one type of defect on the road surface (usually a cracks).

In high resolution 3D road surface data (depth resolution $\leq$ 0.5 mm, lateral resolution $\leq$ 1 mm), cracks include microscopic local defects, while other deformations include rut, potholes, subsidence, pavement bulging are macroscopic defects in profiles. The search for these defects is associated with the geometric features of these road anomalies. For example, a crack often appears as some kind of linear structure, it usually has a width greater than 1 mm, shows a greater depth than a coating without cracks. The track mainly arises as a result of frequent traffic loads on the road surface, has a certain width, depth and continuous length. Potholes and subsidence in the pavement are often characterized by a larger area with greater depth and deformation, and the ridge has a certain height that exceeds the normal surface.

These defects can be more efficiently detected using the information contained in the 3D data [13]. The 3D pavement data acquisition system is used to measure the elevation of the road surface, preprocess and store the profile data. After transforming the coordinates of the image into the coordinates of the object, that is, calibration, data of the elevation difference of the road surface, that is, the profile of the road surface, can be obtained. Further, for each profile, the developed algorithm was used to build the control and standard profiles. In this case, macroscopic deformations are analyzed by comparing the standard contour and the control one, and microscopic defects (cracks) are analyzed according to the so-called residual profile obtained as the difference between the control profile and the profile of the scanned road surface.

The above methods for detecting road defects have a number of significant disadvantages: the extremely narrow focus of the developed algorithms, which are able to detect only individual defects in the road surface, as well as low accuracy in detecting defects. This also results in a low level of automation in detecting deformations of the road surface.

However, these tools are now used less and less. It is being supplanted by the neural network technologies. Since CNN [14] can generalize features from raw data well. So CNN can examine the structure of a defect in an image to find the entire defect at the pixel level without preprocessing. For solution the problem of classification by several labels with unbalanced samples, a strategy is proposed with a change in the ratio of positive and negative training samples. After training on a pre-labeled database of various defects, CNN will be able to detect and classify road surface defects with a high degree of accuracy. One of the approaches for preliminary marking of road surface defects can be preliminary clustering of all present defects. Cracks in the road surface can be thought of as a temporal sequence of pixels that forms a graph on the surface of the road surface. There are many studies on the detection and localization of cracks in 2D images, including using recurrent neural networks [15] (RNN, LSTM).

Asphalt and concrete have a diversity of surface structures, so random debris, drawings (lines or figures drawn on the roads) make it difficult to detect anomalies, namely cracks. While severely damaged surfaces are fairly easy to spot, defects that begin to form are almost invisible. Various approaches can be found to detect pavement defects using different techniques. One of the modern and promising methods is the use of an autoencoder, since the architecture of such a network makes it possible to more accurately segment the damage to the asphalt. In the image segmentation process, [16] proposes U-net to perform semantic image segmentation based on the encoder-decoder architecture to improve accuracy. Various variants of encoder-decoders were also proposed in [17].

Baur et al. [18] propose a framework for defect segmentation using autoencoding architectures and a perpixel error metric based on the `distance.Other approaches take into account the structure of the latent space of variational autoencoders (VAEs) [19] in order to define measures for outlier detection.

All the a works that use autoencoders for unsupervised defect segmentation have shown that autoencoders reliably reconstruct non-defective images while visually altering defective regions to keep the reconstruction close to the learned manifold of the training data.

The authors of this paper compare various technological solutions in the field of machine learning. In paper was presented on the three most famous architectures of neural networks with 6 backbones for each, i.e. a total of 18 different neural network architectures to identify the most

optimal option for using crack detection in wild. Their implementation allows to automate the process of assessing the quality of the road surface. For this purpose, convolutional neural networks of various architectures and conceptual concepts are trained on manually labeled data.

II. MATERIAL AND METHOD

In this work, we studied the segmentation of road surface cracks based on semantic neural networks and studied the effectiveness of the use of machine learning methods and neural network technologies for detecting and classifying damage to asphalt concrete pavements in comparison with traditional methods. Semantic image segmentation is the division of an image into groups of pixels corresponding to one class of an object while simultaneously determining the type of an object in each area. The semantic segmentation task is a high-level image processing related to the group of tasks of the so-called weak artificial intelligence. It is more complicated than the problem of image classification and object detection, since it is necessary not only to define the objects, but also to correctly identify their boundaries in the image. At the same time, the task of semantic segmentation differs markedly from ordinary economic activity, when the regions operate according to the principle of color or texture similarity. Objects can have elements that differ in photometric characteristics and have a significant scatter in the indicators of objects within one class.

Based on the results of the analysis of known architectures, the following architectures were selected for detecting defects: the classic U-Net autoencoder architecture [20], LinkNet autoencoder architecture [21] and convolution network PSPNet [22].

All these neural networks are an autoencoder type neural network. The autoencoder is a neural network that copies input data to output. Autoencoders attempt to reconstruct an input image through a bottleneck, effectively projecting the input image into a lower-dimensional space, called latent space. The goal is to get the response that is closest to the input on the output layer.

A distinctive feature of autoencoders is that the number of neurons at the input and output is the same. An autoencoder consists of two parts:

Encoder: Responsible for compressing the input to latent-space. Introduced by the encoding function $h = f(x)$.

Decoder: designed to recover input from latent-space. Introduced by decoding function $g = f^{-1}(x)$.

Thus, an autoencoder is described by the function $g(f(x)) = r$, where r coincides with the original x at the input.

Semantic neural networks papers describe their network architectures with excellent graphs and simple descriptions, following are the figures copy from the papers.

In this paper, we will compare the advantages and disadvantages of various classical basic classification networks as backbones through experiments [23].

One of the most famous networks used in segmentation is the U-Net neural network. On Fig. 1 is shown the common framework of this network.

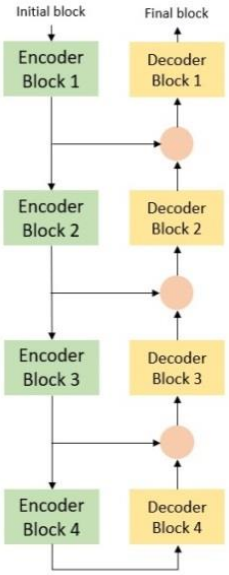

Figure 1. Common architectural framework

The U-net was originally invented and pioneered for biomedical image segmentation. Its architecture can be broadly thought of as an encoder network followed by a decoder network. The goal is to semantically project the distinctive features (lower resolution) learned by the encoder onto pixel space (higher resolution) in order to obtain a dense classification. The encoder is the first half, which is a typical convolutional neural network architecture. On Fig. 2 is shown the structure of each encoder and decoder block.

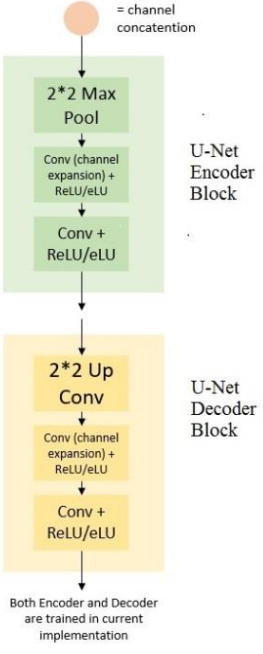

Figure 2. U-Net blocks

It consists of reapplying two 3 × 3 convolutions, followed by a ReLU and a maximum combining (2 × 2 power of 2) operation to downsample. Property channels are doubled at each downsampling step. The decoder is the second half of the architecture. The decoder consists of upsampling and concatenation followed by regular convolution operations. Each step in the decoder consists of a feature map upsampling operation, followed by:

-convolution 2 × 2, which reduces the number of features channels;

-combining with an appropriately cropped feature map from the collapsing path;

-two 3 × 3 convolutions followed by ReLU.

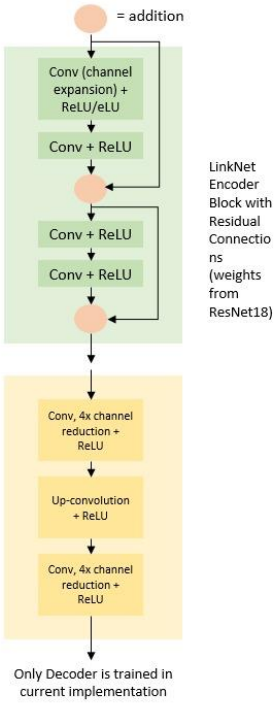

Figure 3. LinkNet blocks

The LinkNet architecture is a complete convolutional network based on an encoder-decoder structure designed for semantic segmentation as shown on Fig. 1. The encoder block consists of four convolutional layers with residual connections, as shown in Fig. 3. LinkNet performance or come from adding the output of encoder to the decoder, this help the decoder easier to recover the information. The current implementation used pre-trained encoder weights from various backbones, with training limited by decoder parameters. Using pretrained weights when simulating a nonlinear segmentation function provides more flexibility and access to a richer set of imaging functions. LinkNet was implemented in the Keras framework. This network is 10 times faster than SegNet.

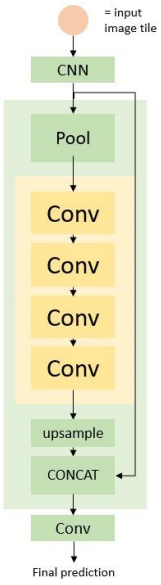

Figure 4. PSPNet architecture

PSPNet is another semantic segmentation model, along with U-Net, which is an autoencoder that takes into account the global image context to predict predictions locally, therefore providing better performance on test datasets. Pyramid Scene Parsing Network (PSPNet) performs a merge operation (via the max or average function) using kernels of different sizes and with different steps applied to the mappings of the output functions from the convolutional neural network, see Fig. 4. The pyramid merging module is a core part of this model as it helps the model capture the global context of the image, which helps it classify pixels based on the global information presented in the image. Then, using bilinear interpolation, the size of all outputs from the merge layer is recalculated and the output characteristics are mapped from the CNN; the model then merges all new outputs along the channel axis. To generate the forecast, the final convolution is performed for the combined output.

In order to experiment with different architectures, build processes for loading images, augmentation of data, calculating metrics, visualizing results and solving other related tasks, researchers create their own frameworks that include all the functionality. The implementation of the above models is present inside the Segmentation models framework.

The Crack500 dataset is used to train the constructed model [24]. The size of each image in the dataset is large enough, so there is a problem with the size of the input data of the neural network due to the limited amount of computer resources. Thus, the existing images were cut into fragments of 320x320 pixels.

The Crack500 dataset was presented in [24], and it contains images captured with mobile phones around the main campus of Temple University. It consists of images with pixel annotations of about 2000 × 1500 pixels (different sizes). It has 250 training, 200 test and 50

verification samples. According to the authors of [24], this is the largest dataset of road surface defects with pixel annotations. A sample scan data is shown in Figure 5.

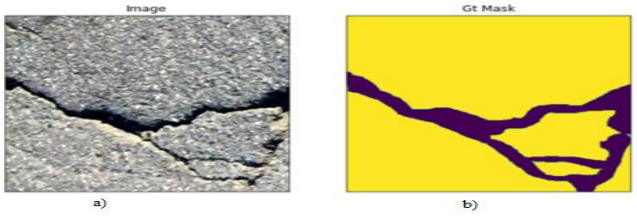

Figure 5. Crack500 datasample. a) Image; b) groundtruth

The data in our problem is presented as a classic CV – in the form of color RGB images. Each photo contains at least one defect. This takes into account defects that occupy at least 6% of the image area. All data is divided into three parts: training (contains 2270 images), validation (164 images) and test (759 images).

Augmentation of data (artificial increase in the data set) is carried out by changing the brightness, scaling, displaying, adding Gaussian noise. The models were trained on a mixed dataset for 5 epochs at a learning rate of 0.001. Only values from the neural network output with a higher or equal to 50% confidence rate were taken into consideration. The best performing solution (according to IoU score) from every training were evaluated [25] with recall, precision, F1, and intersection over union (IoU) measures [26]:

$$Recall = \frac{TP}{TP + FN}, \quad (1)$$

$$Precision = \frac{TP + TN}{TP + FP}, \quad (2)$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}, \quad (3)$$

$$IoU = \frac{GroundTruth \cap Prediction}{GroundTruth \cup Prediction}, \quad (4)$$

where TP is the true positive (correct detection of pixels belonging to labeled defect area); TN isthe true negative (nondefective background pixels correctly recognized by detector); FP is the false positive (wrongly detected defect pixels); FN is the false negative (defect pixels undetected by detector);$GroundTruth$ is the labeled image pixels. $Precision$is the proportion of false alarms; $Recall$ is theproportion of undetected defect pixels; and $F1$is harmonic mean of the precision and recall.

The $F1$ score has been improved with additional transformations such as applying a Gaussian filter, adjusting brightness, shifting. These transformations were applied using the Python albumentations library.

At the final stage, 5 best backbones were selected and training was carried out on 15 epochs with a optimizer Adadelta, loss function – Focal Loss, batch size – 8. The obtained results were entered into a summary table.

When training as an optimizer, Adadelta, loss function – Focal Loss, batch size – 8. To initialize the parameters, we used pretrained on the ImageNet data collection [27] encoder weights. Training time ranged from 4 to 6 hours on the Tesla K80 video card. The training graphs show that for all models, the value 0.93 for F1-score is the limit value. Since among the models there are different kinds of architecture as lighter (MobileNetV2, EfficientNetB0) and heavier (InceptionV3, VGG16), and they all remember the training set in the same way, then we can assume that this border is due to the quality of the markup training sample.

III. RESULTS AND DISCUSSION

After training the neural network, it is validated on test data. Each fragment of the image is fed to the input of the network, and the output is a generated map of the probability of the presence of a defect. The results of predictions of semantic fracture extraction are shown for various Backbones in the case of the U-Net neural network in Table

A. U-Net neural network

The results of predictions of semantic fracture isolation are shown for various Backbones in the case of the U-Net neural network in Table I. (the best values are highlighted in the table):

Table 1. U-Net

U-Net					
Backbones	**IoU**	**F1**	**Precision**	**Recall**	**Loss**
vgg16	0.92849	0.96849	0.95445	0.97204	0.057567
resnet18	0.93047	0.96342	0.95826	0.96993	0.070784
seresnet18	0.92528	0.9607	0.96971	0.9533	0.06774
resnext50	0,89011	0,94127	0,89276	0,99675	0,090145
seresnext50	0,88684	0,93924	0,88961	0,99662	0,092067
senet154	0,89633	0,94473	0,89995	0,99572	0,11845
densenet121	0,89936	0,94624	0,90418	0,99431	0,087808
inceptionv3	0,9012	0,94742	0,90432	0,99568	0,1121
inceptionresnetv2	0,90855	0,95155	0,91619	0,99122	0,09905
mobilenet2	0,86527	0,92709	0,86614	0,99886	0,093656
efficientnetb0	0,90485	0,94949	0,90955	0,99446	0,10463

In Figures 6-7 show the results of the trained U-Net network and their comparison with the true values from the test sample.

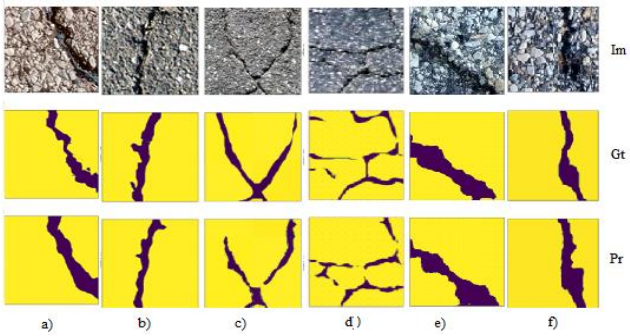

Figure 6. The results of the work of the trained neural network U-Net with various backbones a) VGG16; b) Resnet18; c) Seresnet18; d) Resnet50; e) Seresnext50; f) Senet154.

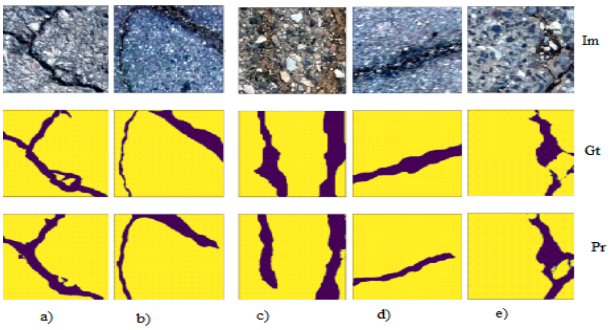

Figure 7. The results of the work of the trained neural network U-Net with various backbones a) Densenet121; b) Inceptionv3; c) Inceptionresnetv2; d) Mobilnet; e) Seresnext50; f) Efficientnetb0.

B. LinkNet neural network

The results of predictions of semantic fracture isolation are shown for various Backbones in the case of the LinkNet neural network in Table II. (the best values are highlighted in the table):

Table 2. LinkNet

LinkNet					
Backbones	**IOU**	**F1**	**Precision**	**Recall**	**Loss**
vgg16	0,85607	0,92094	0,86063	0,99342	0,19355
resnet18	0,8524	0,91883	0,85683	0,99382	0,1979
seresnet18	0,8925	0,9423	0,89967	0,99151	0,094954
resnext50	0.88724	0.93935	0.89209	0.99406	0.095869
seresnext50	0.85353	0.92014	0.85622	0.99638	0.29034
senet154	0.85579	0.92076	0.8643	0.98841	0.31713
densenet121	0,85394	0,92044	0,85603	0,99723	0,31401
inceptionv3	0,85548	0,92144	0,85576	0,99962	0,41015
inceptionresnetv2	0,85548	0,92144	0,85576	0,99962	0,41015
mobilenet2	0,85548	0,92144	0,85576	0,99962	0,41015
efficientnetb0	0,85063	0,91801	0,85553	0,99325	0,36172

In Figures 8-9 show the results of the trained LinkNet network and their comparison with the true values from the test sample.

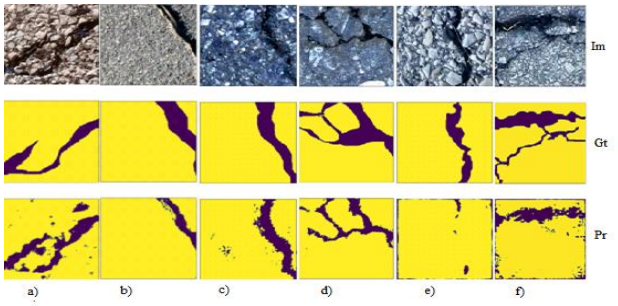

Figure 8. The results of the work of the trained neural network LinkNet with various backbones a) VGG16; b) Resnet18; c) Seresnet18; d) Resnet50; e) Seresnext50; f) Senet154.

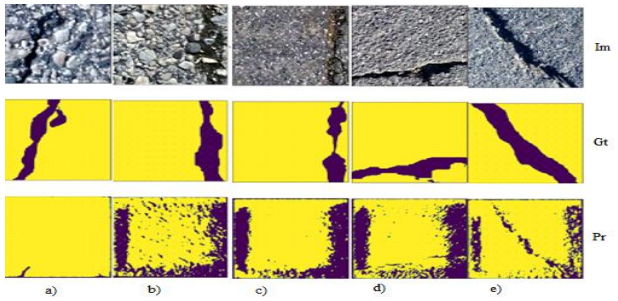

Figure 9. The results of the work of the trained neural network LinkNet with various backbones a) Densenet121; b) Inceptionv3; c) Inceptionresnetv2; d) Mobilnet; e) Efficientnetb0.

C. PSPNet neural network

The results of predictions of semantic fracture isolation are shown for various Backbones in the case of the PSPNet neural network in Table III. (the best values are highlighted in the table):

Table 3. PSPNet

.					
Backbones	**IOU**	**F1**	**Precision**	**Recall**	**Loss**
vgg16	0.89694	0.94525	0.89706	0.99986	0.098065
resnet18	0.89361	0.94341	0.89363	0.99998	0.110510
seresnet18	0.89871	0.94586	0.84651	0.99850	0.105894
resnext50	0.89589	0.94468	0.89596	0.99992	0.081242
seresnext50	0.89282	0.94296	0.89282	1.00000	0.279520
senet154	0.89282	0.94296	0.89282	1.00000	0.279520
densenet121	0.91830	0.95699	0.92089	0.99710	0.062210
inceptionv3	0.89494	0.94413	0.89507	0.99985	0.134600
inceptionresnetv2	0.90363	0.94891	0.90417	0.99938	0.090984
mobilenetv2	0.87964	0.93379	0.89539	0.98000	0.187550
efficientnetb0	0.90055	0.94725	0.9016	0.99866	0.130300

In Figures 10-11 show the results of the trained PSPNet network and their comparison with the true values from the test sample.

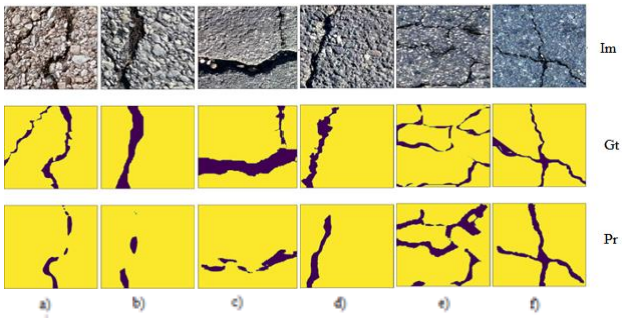

Figure 10. The results of the work of the trained neural network PSPNet with various backbones a) VGG16; b) Resnet18; c) Seresnet18; d) Resnet50; e) Seresnext50; f) Senet154.

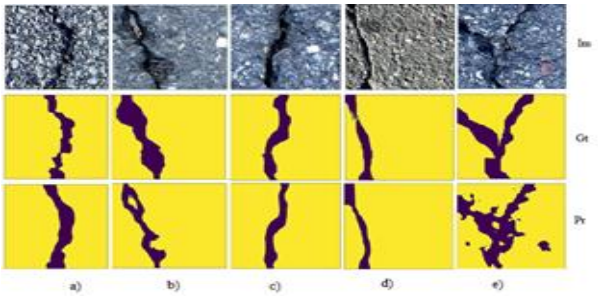

Figure 11. The results of the work of the trained neural network PSPNet with various backbones a) Densenet121; b) Inceptionv3; c) Inceptionresnetv2; d) Mobilnet2; e) Efficientnetb0.

Training results after 15 epochs on 5 backbones for U-Net (the best values are highlighted in the table):

Table 4. U-Net

UNet					
Backbones	**IOU**	**F1**	**Precision**	**Recall**	**Loss**
vgg16	0.91037	0.95258	0.91719	0.99209	0.060295
Resnet18	0.91678	0.95617	0.92072	0.99545	0.045474
inceptionv3	0.92418	0.96024	0.92778	0.99593	0.051915
mobilenetv2	0.92622	0.96131	0.93097	0.9946	0.05407
Efficientnetb0	0.9136	0.95427	0.91738	0.99558	0.063473

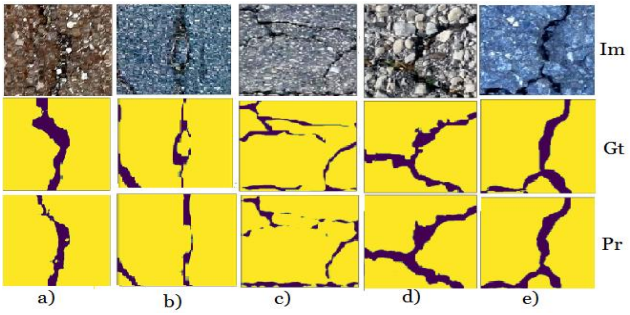

Figure 12. The results of the work of the trained neural network PSPNet with various backbones (15 epochs) a) VGG16; b) Resnet18; c) Inceptionv3; d) Mobilnet2; e) Efficientnetb0.

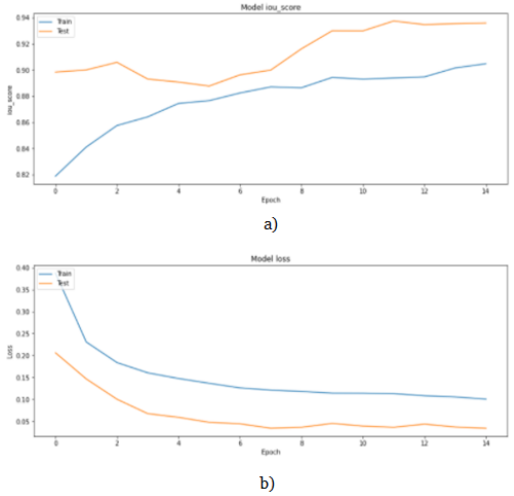

Figure 13. Sample U-Net (Mobilenet2) a) Metric IoU on test and train; b) Train and Test Loss Function.

Training results after 15 epochs on 5 backbones for U-Net (the best values are highlighted in the table):

Table 5. LinkNet

LinkNet					
Backbones	**IOU**	**F1**	**Precision**	**Recall**	**Loss**
vgg16	0.84022	0.90842	0.95	0.99209	0.25843
Resnet18	0.91248	0.94716	0.91072	0.99545	0.087314
inceptionv3	0.92248	0.95933	0.92539	0.99672	0.067774
mobilenetv2	0.91552	0.95516	0.92701	0.9868	0.070107
Efficientnetb0	0.92322	0.95964	0.92729	0.9954	0.050769

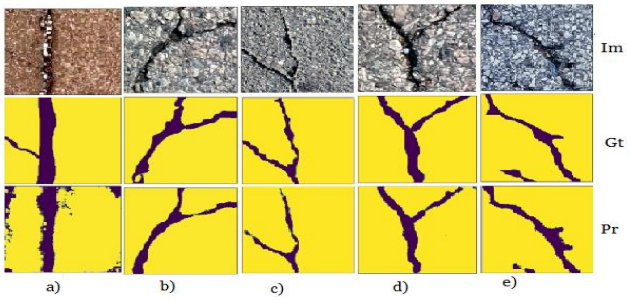

Figure 14. The results of the work of the trained neural network LinkNet with various backbones (15 epochs) a) VGG16; b) Resnet18; c) Inceptionv3; d) Mobilnet2; e) Efficientnetb0.

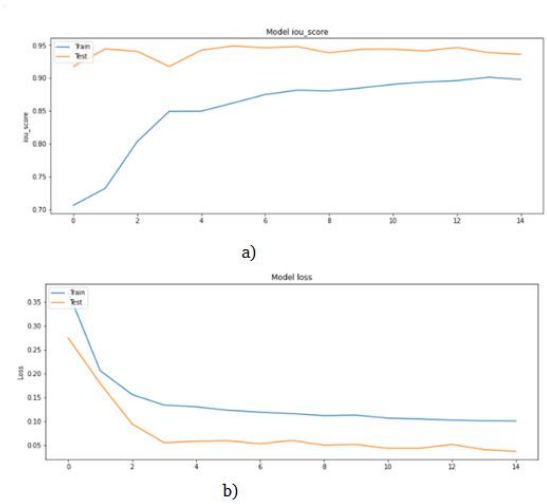

Figure 15. Sample LinkNet (Efficientnetb0) a) Metric IoU on test and train; b) Train and Test Loss Function.

Training results after 15 epochs on 5 backbones for U-Net (the best values are highlighted in the table):

Table 6. PSPNet

PSPNet					
Backbones	**IOU**	**F1**	**Precision**	**Recall**	**Loss**
vgg16	0.87207	0.92834	0.89758	0.96798	0.15995
Resnet34	0.91255	0.95385	0.95385	0.99796	0.091124
inceptionv3	0.92217	0.95909	0.92668	0.995	0.16507
mobilenetv2	0.89913	0.94643	0.90114	0.99759	0.1788
Efficientnetb0	0.9134	0.95433	0.91499	0.99818	0.10465

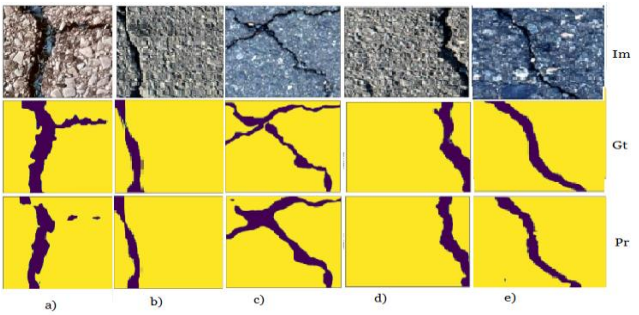

Figure 16. The results of the work of the trained neural network PSPNet with various backbones (15 epochs) a) VGG16; b) Resnet18; c) Inceptionv3; d) Mobilnet2; e) Efficientnetb0.

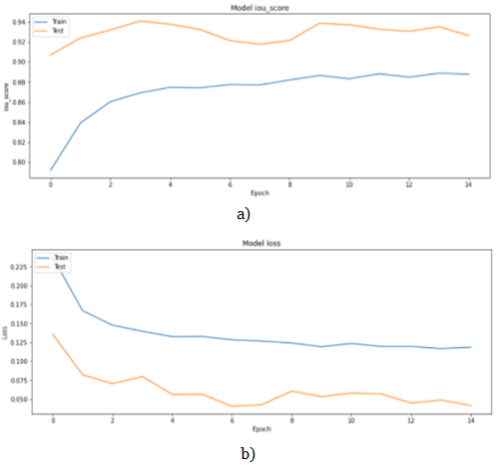

Figure 17. Sample PSPNet(Inceptionv3 a) Metric IoU on test and train; b) Train and Test Loss Function.

IV. CONCLUSIONS

As it can be seen from the data obtained, none of the three convolutional networks showed fundamentally better results on any backbone. Also, there are no significant improvements in results when moving from 5 epochs to 15 training epochs. Almost all metric values were round 0.9. We can assume that this border is due to the quality of the markup training sample. On the other hand, based on the results obtained, it can be concluded that of the three presented architectures of convolutional neural networks, the U -Net convolutional network based on VGG16, Resnet18 demonstrated the highest accuracy, F1 score – 0.96849, IoU score – 0.93047 on 5 epochs. In future work, we plan to revise the annotations and introduce even more different data for the problem of detecting damage to the road surface. Since collecting and labeling data samples takes time and precision, synthetic data can also be introduced into the model training process. While traditional image processing techniques such as rotation, brightness correction, and noise addition may be limited in complex cases, techniques such as generative adversarial networks (GANs) or variational autoencoders (VAEs) can be employed to address specific problems. This has shown promising results in recent studies and could be a possible solution to the problem under analysis.

References

[1] K. Gopalakrishnan, "Deep learning in data-driven pavement image analysis and automated distress detection: A review," *Data*, vol. 3, issue 3, article 28, 2018. https://doi.org/10.3390/data3030028.

[2] W. Li, H. Ju, S. Tighe, Q. Ren, Z. Sun, "Three-dimensional pavement crack detection algorithm based on two-dimensional empirical mode decomposition," *J. Transp. Eng.*, Part B: Pavements, vol. 143, issue 2, 04017005, 2017. https://doi.org/10.1061/JPEODX.0000006.

[3] D. Hwang, D.E. Kim, "Special features on intelligent imaging and analysis," *Appl. Sci*, vol. 9, article 4804, 2019. https://doi.org/10.3390/app9224804.

[4] R. Augustauskas, A. Lipnickas, "Improved pixel-level pavement-defect segmentation using a deep autoencoder," *Sensors*, vol. 20, issue 9, article 2557, 2020. https://doi.org/10.3390/s20092557.

[5] H. Oliveira, P.L. Correia, "Automatic road crack segmentation using entropy and image dynamic thresholding," In *Proceedings of the European Signal Processing Conference*, Glasgow, UK, 2009, pp. 622–626.

[6] M.-R. Jahanshahi, et al., "An innovative methodology for detection and quantification of cracks through incorporation of depth perception," *Machine Vision and Applications*, vol. 24, pp. 227–241, 2013. https://doi.org/10.1007/s00138-011-0394-0.

[7] H. Oliveira, P. L. Correia, "CrackIT – An image processing toolbox for crack detection and characterization," in *Proceedings of the IEEE International Conference on Image Processing*, Paris, 2014, pp. 798–802, 2014. https://doi.org/10.1109/ICIP.2014.7025160.

[8] R. J. Medina, et al., "Enhanced automatic detection of road surface cracks by combining 2D/3D image processing techniques," in *Proceedings of the IEEE International Conference on Image Processing* (ICIP-2014), Paris, pp. 778–782, 2014. https://doi.org/10.1109/ICIP.2014.7025156.

[9] S. Chanda, et al., "Automatic bridge crack detection – A texture analysis-based approach," *Artificial Neural Networks in Pattern Recognition (ANNPR), Lecture Notes in Computer Science*, Springer, vol. 8774, 2014, pp. 193–203. https://doi.org/10.1007/978-3-319-11656-3_18.

[10] E. Salari, G. Bao, "Automated pavement distress inspection based on 2D and 3D information," In *Proceedings of the 2011 IEEE International Conference on Electro/Information Technology*, Mankato, MN, USA, 2011, pp. 2–5. https://doi.org/10.1109/EIT.2011.5978575.

[11] Y. Hu and C.-X. Zhao, "A local binary pattern based methods for pavement crack detection," *Journal of Pattern Recognition Research*, vol. 1, no. 2010, pp. 140–147, 2010. https://doi.org/10.13176/11.167.

[12] V. Baltazart, P. Nicolle, L. Yang, "Ongoing tests and improvements of the MPS algorithm for the automatic crack detection within grey level pavement images," In *Proceedings of the 25th European Signal Processing Conference (EUSIPCO)*, Kos, Greece, 2017, pp. 2016–2020. https://doi.org/10.23919/EUSIPCO.2017.8081563.

[13] D. Zhang, Q. Zou, H. Lin, X. Xu, L. He, R. Gui, Q. Li, "Automatic pavement defect detection using 3D laser profiling technology," *Automation in Construction*, vol. 96, pp. 350-365, 2018. https://doi.org/10.1016/j.autcon.2018.09.019.

[14] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "Deep crack: Learning hierarchical convolutional features for crack detection," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1498–1512, 2018. https://doi.org/10.1109/TIP.2018.2878966.

[15] B. Varona, A. Monteserin, A. Teyseyre, "A deep learning approach to automatic road surface monitoring and pothole detection," *Pers Ubiquit Comput*, vol. 24, pp. 519–534, 2020. https://doi.org/10.1007/s00779-019-01234-z.

[16] O. Ronneberger, P. Fischer, T. Brox, "U-net: Convolutional networks for biomedical image segmentation," In *Proceedings of the Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Munich, Germany, 5–9 October 2015; Springer: Cham, Switzerland,

2015, vol. 9351, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.
[17] R. Augustauskas, A. Lipnickas, "Improved pixel-level pavement-defect segmentation using a deep autoencoder," *Sensors*, vol. 20, 2557, 2020. https://doi.org/10.3390/s20092557.
[18] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Deep Autoencoding Models for Unsupervised Anomaly Segmentation in Brain MR Images," arXiv preprint arXiv:1804.04488, 2018. https://doi.org/10.1007/978-3-030-11723-8_16.
[19] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *Proceedings of the International Conference on Learning Representations*, 2014.
[20] N. Navab, J. Hornegger, W. Wells, A. Frangi, "Medical image computing and computer-assisted intervention," *MICCAI 2015*, *Lecture Notes in Computer Science*, vol. 9351, Springer, Cham, 2015. https://doi.org/10.1007/978-3-319-24571-3.
[21] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," *Proceedings of the 2017 IEEE, Visual Communications and Image Processing (VCIP)*, St. Petersburg, FL, USA, 2017, pp. 1-4, 2017. https://doi.org/10.1109/VCIP.2017.8305148.
[22] H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, "Pyramid scene parsing network," *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 6230-6239. https://doi.org/10.1109/CVPR.2017.660
[23] R. Zhang, L. Du, Q. Xiao, J. Liu, "Comparison of backbones for semantic segmentation network," Journal of Physics: Conference Series, 1544, 2020. https://doi.org/10.1088/1742-6596/1544/1/012196.
[24] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, H. Ling, "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE Trans. Intell. Transp. Syst*., vol. 21, pp. 1525–1535, 2020. https://doi.org/10.1109/TITS.2019.2910595.
[25] D.M.W. Powers, "Ailab evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," Inf. Markedness Correl., vol. 2, pp. 37–63, 2011.
[26] H. Rezatofighi, N. Tsoi, J.-Y. Gwak, A. Sadeghian, I. Reid, S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2019, pp. 658-666. https://doi.org/10.1109/CVPR.2019.00075.
[27] ImageNet large scale visual recognition competition 2014 [Online]. Available at: URL: http://imagenet.org/challenges/LSVRC/2014/.

SVETLANA MUSTAFINA, S. A. a scientist in the field of mathematical modeling and optimal control of chemical-technological processes. A methodology for determining the optimal modes of the processes of chemical and petrochemical industries has been created, criteria for monitoring and regulating quality, qualitative stability and instability of the optimal modes of chemical production have been introduced.

ANDREY AKIMOV, *Graduated from Bashkir State University. Research interests: artificial intelligence, boundary value problems.*

SOFIA MUSTAFINA, *a student. Research interests: artificial intelligence.*

ALEXANDRA V. PLOTNIKOVA graduated from Southwest State University. Research interests: artificial intelligence.