

Machine Learning for Remote Monitoring of Agricultural Fields with Explosive Tunnels

Nikolay Kiktev ¹, Oleksiy Opryshko ¹, Alla Dudnyk ¹ and Volodymyr Reshetiuk ²

¹ National University of Life and Environmental Sciences of Ukraine, Department of Automation and Robotic Systems, Kyiv, Ukraine

² Szkoła Główna Gospodarstwa Wiejskiego w Warszawie, Warsaw, Poland

Abstract

The article is devoted to the study of the application of neural networks for recognizing craters resulting from explosions in agricultural fields using satellite or UAV images. The aim of the study is to assess the state of destruction based on data on the real state of fields obtained as a result of remote monitoring, using machine learning tools. It has been established that recognition of craters from explosions in images with a resolution of 0.5 m / pixel is possible using neural networks. In the course of the study, round craters were studied, but during military operations other shapes are also possible, which require further study. In further studies, it is advisable to introduce an additional parameter - the crater shape index. The results obtained indicate the prospects for introducing this approach in the post-war restoration of agricultural lands in Ukraine.

Keywords

Neural network, agricultural land, image recognition, blast craters, training, dataset

1. Introduction

The food market in the world is in an unstable state. According to the results of the research work of F. Trajkovikj (2023) in [1] regarding the European food market, serious anomalies regarding price gouging have been observed in recent years. One of the important reasons for this phenomenon is large-scale hostilities in Europe starting in 2022 and, accordingly, withdrawal of large amounts of agricultural land. The situation with damage to the fields has a long-term nature, so according to Dries Claeys (2019) in [2] field funnels from artillery shells in the fields of Western Europe have been preserved since the First World War.

Taking into account the significantly increased explosive power of projectiles and the appearance of ballistic, cruise missiles and powerful aerial bombs, the situation with damage to fields in modern wars is fundamentally worsening.

According to estimates given in the work of Deepak Rawtani (2022) in [3], the war in Ukraine is the most large-scale armed conflict in Europe since the Second World War in terms of destruction and damage to infrastructure. In order to ensure the stability of the food market, it is necessary to promptly introduce fields affected by military operations into circulation. Taking into account limited human and material resources, farmers need to take into account the cost of restoration, assess the state of destruction and calculate optimal routes as shown in the works of V. Mezhujev (2020) in [4], S. Lienkov (2022) in [5]. Such measures are possible only if there is data on the real state of the fields that can be obtained from the results of remote monitoring, the development of approaches to which was the goal of our work.

The purpose of the work is to assess the state of destruction based on data on the real state of the fields, obtained as a result of remote monitoring, using machine learning tools.

Dynamical System Modeling and Stability Investigation (DSMSI-2023), December 19-21, 2023, Kyiv, Ukraine

EMAIL: nkiktev@ukr.net (A. 1); ozon.kiev@gmail.com (A.2); dudnikalla@nubip.edu.ua (A.3); volodymyr_reshetiuk@sggw.edu.pl (A.4)

ORCID: 0000-0001-7682-280X (A. 1); 0000-0001-9797-3551 (A.2); 0000-0001-6433-3566 (A.3); 0000-0002-3183-9744 (A.4)



© 2023 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

2. Materials and Methods

2.1. State of the problem

According to Giacomo Certini (2013) in [6], physical disturbances of the soil include compaction due to the construction of defense infrastructures, digging trenches or tunnels, compaction by the movement of machinery and troops or the formation of bomb craters, etc. Places where fortified areas were placed may have fortifications: anti-tank embankments, ditches, etc. The location of fortified areas for military reasons should be located taking into account the topography of the area and the existing road infrastructure and, as a rule, not directly on the fields, accordingly, in this work, attention was focused on funnels. The work of Nataliia Kussul (2023) in [7] was devoted to research specifically on sinkholes. Thanks to the free Sentinel-2 satellite data, the affected fields were identified based on changes in the dynamics of vegetation indices. Analogous studies with similar tools were also carried out in the work of Maksym Solokha (2023) in [8]. In both cases, similar results were obtained, but in fact the total area of the sinkholes is determined, which to some extent can be useful in view of the legal and economic aspect of the problem, but is poorly adapted for soil restoration calculations.

When using vegetation indices to assess the state of crops, the method of obtaining them with its inherent limitations should also be taken into account. Thus, in the work of N.A. Pasichnyk (2021) in [9], the results are presented according to which the influence of atmospheric correction on the determination of vegetation indices is very significant, and the possibility of traditional correction of data based on ground research at the sites of recent hostilities is doubtful.

It is also unknown and the composition of vegetation and their state of organogenesis can also be determined only approximately, and according to Pasichnyk N. (2023) in [10], not taking these indicators into account will also affect the accuracy and general suitability of the data.

Better identification of sinkholes based on the results of satellite monitoring is shown in the work of Erik C. Duncan (2023) in [11], but based on the results, the model created better determines the presence of craters and not their shape and dimensions. Accordingly, craters with an area of more than 60 m² are reliably identified, which is useful to know for the needs of chemical decontamination, but the disposal of such craters is doubtfully appropriate in view of plant breeding practices. So, Figure 1 shows photos from the Internet of the funnels of various explosive devices.



Figure 1. Funnels from explosive devices. The funnel from the most common 155 mm high-explosive projectile (left) and from the Shahid-136 UAV (right)

If a funnel from a 155 mm high-explosive projectile can be eliminated relatively easily using the available resources of farms, it is relatively difficult to implement a funnel from a ballistic missile several meters deep. Accordingly, as a result of explosions, there may be craters on the field that can and should be leveled with a bulldozer, and craters that are dangerous for such equipment due to their dimensions. Therefore, it is necessary to calculate trajectories in case of possible conflict situations as shown in N. Pasichnyk (2023) in [12] and in the presence of restrictions as shown in S.A. Shvorov (2019) in [13]. Monitoring in the works presented above was carried out using high-resolution satellite data of 0.3 - 0.5 m/pixel, however, starting from 2022, such a Google Earth Pro product stopped distributing high-resolution images for the territory of Ukraine. An alternative to obtaining such data can be UAVs, which, when using aircraft platforms, can survey up to 10,000 hectares per day.

The experience of using UAVs for monitoring for the needs of humanitarian demining is presented in the work of N. Kiktev (2023) in [14], where it was proposed to detect anti-tank mines laid on the surface of the earth using a thermal imaging tool. During the experiments, the authors established that funnels from explosive devices are well identified, but the selectivity is low due to the presence of foreign objects such as car skids, parts of equipment, etc. Accordingly, an approach was used regarding the use of neural networks to track the presence of the location of objects characteristic of mine barriers.

Despite the possibility of UAVs for monitoring, including the consequences of actions, they are used relatively little, but such experience is present, in particular, in the study of volcanic activity, as shown in the works of Ruli Andaru (2021) in [15] and A. Román (2022) in [16]. For volcanoes, UAVs are used because they allow you to work quickly and under a layer of clouds or smoke during an eruption. Therefore, even with the limited availability of satellite data, there are available technical means for obtaining visual information.

- So, based on the literature review, we can draw conclusions about:
- the possibility of identifying craters caused by military actions on high-resolution images;
- craters from explosions persist in fields for decades and will severely limit agricultural practices in fields;
- recognition can be made by the shape characteristic of such craters;
- it is necessary to determine the dimensions of the craters, at least in horizontal projections, since not all craters can be repaired with farm resources and they must be taken into account when planning routes.

To solve the problem, we will use pattern recognition methods and algorithms - a section of computer science that develops the foundations and methods of classification and identification of objects, phenomena, processes, signals, situations and other objects that are characterized by a finite set of certain properties and features using various sections of applied mathematics [17, 18].

2.2. Organization of experimental studies and processing of results.

Since high-intensity hostilities have been taking place on the territory of Ukraine since 2014, archival satellite survey data obtained from the Google Earth Pro service for the years 2013-2017 were used for the work. Fields with coordinates 47.927794, 38.746100 should be studied (Fig. 2-4).



Figure 2. Starting field without the influence of hostilities (08,2013)



Figure 3. Field with craters from explosive devices of different power, smaller on the opposite side of the green field and more powerful from below on the dark field (07.2014).



Figure 4. Field with craters with tracks of ground equipment during attempts to neutralize the craters (08.2014)

3. Results

3.1. Data recruitment preparation

To process graphic data, it was decided to generate a neural network in Python. Below is a description and code snippets of the software. Figure 5 presents a fragment of the code for downloading data with images.

```

In [8]: import numpy as np
        from matplotlib import pyplot as plt

In [9]: data = tf.keras.utils.image_dataset_from_directory('data')

Found 305 files belonging to 2 classes.

In [10]: data_iterator = data.as_numpy_iterator()

In [11]: batch = data_iterator.next()

In [12]: fig, ax = plt.subplots(ncols=4, figsize=(20,20))
        for idx, img in enumerate(batch[0][:4]):
            ax[idx].imshow(img.astype(int))
            ax[idx].title.set_text(batch[1][idx])

```

Figure 5. The screenshot of the code, which loads data with images, converts them to the format of the Numpy array.

Used `tf.keras.utils.image_dataset_from_directory` to download data set from DATA Directory containing an array of files divided into 2 classes. The data is converted to the Numpy array with `data.as_numpy_iterator()`. The first set of data is obtained from the iterator using `Data_iterator.next()`. Used `PLT.SUBPLOTS` to create subplot (4 columns, figure size 2020). The for and enumerate cycle shows the first 4 images from a data set (Fig.6).

```

In [13]: data = data.map(lambda x,y: (x/255, y))

In [ ]: data.as_numpy_iterator().next()

In [15]: train_size = int(len(data)*.7)
        val_size = int(len(data)*.2)
        test_size = int(len(data)*.1)

In [16]: train_size

Out[16]: 7

In [17]: train = data.take(train_size)
        val = data.skip(train_size).take(val_size)
        test = data.skip(train_size+val_size).take(test_size)

```

Figure 6. A screenshot of the software code for organizing machine learning: data scaling and distribution.

At the next stage, the images are normalized, that is, the values of each pixel are divided by 255. This is done so that the pixel values are in the range from 0 to 1, which helps the model learn faster and better. After scaling, the images are stored in a format that is convenient for processing (NumPy arrays).

Next, the data is divided into three parts: training, validation and test sets. The training set (70% of the data) is used to train the model. The validation set (20% of the data) is used to tune the model and check its accuracy during training. The test set (10% of the data) is used to finally check the accuracy of the model after training is complete. Distribution process. First, the dimensions of each of these parts are calculated (how many images will be in each part). Then the data is actually distributed: first, data is taken for the training set, then for the validation set, and finally for the test set. This approach helps create a robust model that can perform well on new, unknown data.

3.2. Building a model of deep learning

The examination of the educational data is presented in Fig. 7.

```
In [18]: train

Out[18]: <TakeDataset element_spec=(TensorSpec(shape=(None, 256, 256, 3), dtype=tf.float32, name=None), TensorSpec(shape=(None,), dtype=tf.int32, name=None))>

In [19]: from tensorflow.keras.models import Sequential
         from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, Dropout

In [20]: model = Sequential()

In [21]: model.add(Conv2D(16, (3,3), 1, activation='relu', input_shape=(256,256,3)))
         model.add(MaxPooling2D())
         model.add(Conv2D(32, (3,3), 1, activation='relu'))
         model.add(MaxPooling2D())
         model.add(Conv2D(16, (3,3), 1, activation='relu'))
         model.add(MaxPooling2D())
         model.add(Flatten())
         model.add(Dense(256, activation='relu'))
         model.add(Dense(1, activation='sigmoid'))

In [22]: model.compile('adam', loss=tf.losses.BinaryCrossentropy(), metrics=['accuracy'])
```

Figure 7. Code implementation for creating a deep learning model for image recognition using the TensorFlow library.

The top row shows that the training data used is a set of 256x256 pixel images. The following lines show the import of the main components for building the model, including libraries for building the different layers of the neural network. Creation of a consistent model. The model = Sequential() line creates a model structure that allows you to add layers one by one. Adding layers to the model:

- First layer: This is a convolutional layer that extracts the important features of the images.
- Maxpooling: Reduces the dimensionality of images, which helps reduce the amount of computation and prevent overtraining.
- Second and third layers: again convolutional layers and maxpooling, which helps to extract even more complex image features.
- Antialiasing Layer: Converts 2D data to 1D so that it can be passed to dense layers.
- Dense layers: include neurons that combine all the previous features and make a decision about which class the image belongs to.
- Compiled model: In the final step, the model is compiled using the adam optimizer and the BinaryCrossentropy loss function suitable for two-class classification. The model will also be evaluated for accuracy. This process creates a neural network that can learn from images and use its knowledge to classify new images. Fig. 8 shows a summary table that helps to understand the structure of the model, the number of parameters in each layer and their form, which is important for checking the correctness of the model construction and its evaluation.

In [23]:

```
model.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 16)	448
max_pooling2d (MaxPooling2D)	(None, 127, 127, 16)	0
conv2d_1 (Conv2D)	(None, 125, 125, 32)	4640
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 32)	0
conv2d_2 (Conv2D)	(None, 60, 60, 16)	4624
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 16)	0
flatten (Flatten)	(None, 14400)	0
dense (Dense)	(None, 256)	3686656
dense_1 (Dense)	(None, 1)	257

=====
Total params: 3,696,625
Trainable params: 3,696,625
Non-trainable params: 0
=====

Figure 8. Program code for the deep learning model

Model Description:

- Model type: sequential (sequential model)
- Model structure
- Model layers: Conv2D (Conv2D):
- Output form: (None, 254, 254, 16)
- Number of parameters: 448

It is a concentration layer that highlights the main features of the image. Max_pooling2D (Maxpooling2D):

- Output form: (None, 127, 127, 16)
- Number of parameters: 0

It is a layer of size reduction that reduces the spatial size of the selected Fig.

- CONV2D_1 (CONV2D):
- Output form: (None, 125, 125, 32)
- Number of parameters: 4640

Another concentration layer for a deeper selection of Fig.

- Max pooling2d 1 (Maxpooling2D):
- Output form: (None, 62, 62, 32)
- Number of parameters: 0

Layer of size reduction.

- CONV2D 2 (CONV2D):
- Output form: (None, 60, 60, 16)
- Number of parameters: 4624

The third concentration layer.

- Max pooling2D 2 (Maxpooling2D):
- Output form: (None, 30, 30, 16)
- Number of parameters: 0

Another layer of size reduction.

- Flatten (Flatten):
- Output form: (None, 14400)
- Number of parameters: 0

This layer translates multidimensional data into a one -dimensional vector.

- Dense (Dense):
- Output form: (None, 256)
- Number of parameters: 3,686,656

Dense (fully connected) layer that processes smoothed data.

- Dense 1 (Dense):
- Output form: (None, 1)
- Number of parameters: 257

The last layer that produces the final classification result.

- General information
- Total number of parameters: 3,696,625
- Number of parameters that study: 3,696,625
- Number

3.3. Display results.

In Fig. 9 shows a graphical reflection of the learning outcomes of the deep learning model, in particular, changes in loss (loss) and accuracy (Accuracy) during training and validation. The first code creates a schedule of losses during training (LOSS) and validation losses (Val_loss). The graph shows how these indicators change with each era (iteration of training). The graph shows that both curves decrease, indicating that the model is gradually learning and improving. The second code creates a graph of accuracy during training and validation accuracy. The graph demonstrates how the accuracy of the model with each era changes. The curves on the graph show the growth, which means that the model becomes more accurate in its forecasts. Based on the graphed results, it can be seen that both training and validation losses are decreasing, indicating that the model is learning and improving with each epoch. Regarding accuracy, it can be seen that both accuracy curves are increasing, which indicates an improvement in the model's ability to make correct predictions. These graphs are important tools for evaluating a model's training performance and help understand whether the model is continuing to improve or has already reached its maximum.

The testing process is carried out according to the following algorithm:

- Image import and reading:
- The cv2 library is imported.
- Image 154006829.jpg is read using cv2.imread and displayed using plt.imshow.

Changing the image size:

- The image is resized to 256x256 pixels, which is the size used to train the model.
- The changed image is displayed again.

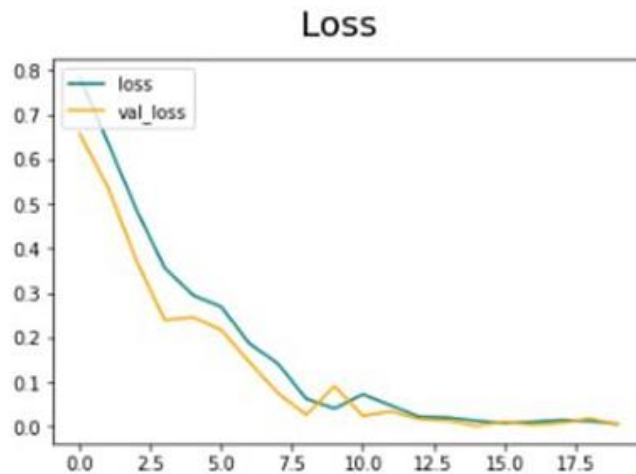
Class prediction:

- The model uses the modified image to predict the class (model.predict).

The prediction result (yhat) is compared with a threshold value of 0.5:

- If the result is greater than 0.5, the image is classified as "Sad".
- Otherwise, the image is classified as "Happy".


```
In [27]: fig = plt.figure()
plt.plot(hist.history['loss'], color='teal', label='loss')
plt.plot(hist.history['val_loss'], color='orange', label='val_loss')
fig.suptitle('Loss', fontsize=20)
plt.legend(loc="upper left")
plt.show()
```



```
In [28]: fig = plt.figure()
plt.plot(hist.history['accuracy'], color='teal', label='accuracy')
plt.plot(hist.history['val_accuracy'], color='orange', label='val_accuracy')
fig.suptitle('Accuracy', fontsize=20)
plt.legend(loc="upper left")
plt.show()
```

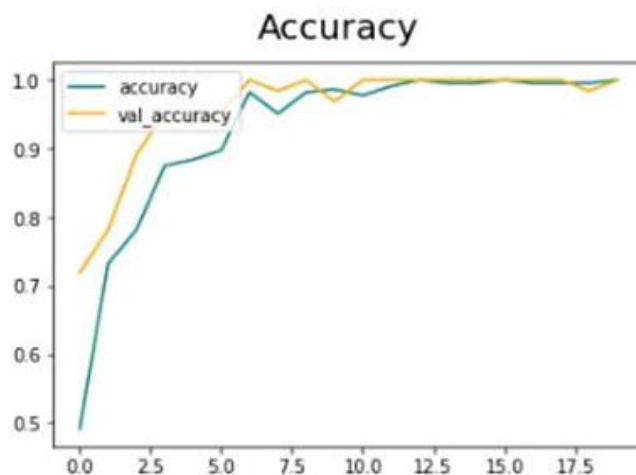


Figure 9. Graphic display of deep learning model training results, in particular, changes in loss and accuracy during training and validation

4. Discussion

Directly, the problem of identification on photographs of funnels from projectiles was almost not considered before, since practically after the 2nd World War, there were no high-intensity combat operations in Europe with the consumption of 152-mm projectiles of more than 5,000 per day. The most

similar technical solutions relate to the identification of craters from meteorites and other cosmic bodies on the surface of the planets from photographs. In the review work of Atal Tewari (2023) in [19], an analysis of the most common algorithms for machine learning in relation to the search for craters, namely based on semantic segmentation, based on object detection and based on classification, was carried out. However, the system was trained on a common database of images obtained, as a rule, from space telescopes. In our case, we will have to take into account the peculiarities of the means of obtaining visual information and the peculiarities of meteorological conditions. In our opinion, this explains the difference between the results obtained by us and the information obtained in the review work. That is, the evaluation of each architecture and its potential applications for the detection of craters created for space objects require adaptation for fields with traces of combat operations. This statement coincides with the results of E. Emami (2019) in [20], where the evaluation was carried out on different data sets from third-party providers.

A universal crater detection scheme based on the newly proposed Segmentation of Everything Model (SAM) from META AI is shown in the work of Iraklis Giannakis (2023) in [21]. SAM is a segmentation system with cues and zero-shot generalization to unfamiliar objects and images without the need for additional training. Unlike meteorites, which can hit the surface at an arbitrary angle, howitzer shells, which account for the majority of craters in experimental fields, fall almost vertically, which limits the nomenclature of crater geometry, and accordingly, in our case, training can be facilitated. Similar results to ours were obtained in the work of Atal Tewari (2022) in [22], where the angle at which the contact occurred was taken into account along with the images used for training the model.

5. Conclusion

It has been established that recognition of sinkholes in images with a resolution of 0.5 m/pixel is possible using neural networks. During the previous studies, it was possible to obtain high accuracy, but the images from the google earth service provide only data obtained under the most favorable conditions of satellite photography, which probably under normal conditions can affect the obtained results. During the research, round craters were studied, however, other forms of craters are also possible during combat operations, which need to be further investigated. In further studies, it is advisable to introduce an additional parameter - the crater shape index.

6. References

- [1] F. Trajkovikj et al., "A comprehensive study of food prices and food fraud in the European Union," 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, 2023, pp. 4557-4566, doi: 10.1109/BigData59044.2023.10386666.
- [2] Dries Claeyss, Celina Van Dyck, Gert Verstraeten, Yves Segers "The importance of the Great War compared to long-term developments in restructuring the rural landscape in Flanders (Belgium)" *Applied Geography*, Vol. 111, 2019, 102063, doi: 10.1016/j.apgeog.2019.102063.
- [3] Deepak Rawtani, Gunjan Gupta, Nitasha Khatri, Piyush K. Rao, Chaudhery Mustansar Hussain "Environmental damages due to war in Ukraine: A perspective" *Science of The Total Environment*, Vol. 850, 2022, 157932, doi: 10.1016/j.scitotenv.2022.157932.
- [4] Mezhyuev V., Gunchenko Y., Shvorov S., Chyrchenko D. "A method for planning the routes of harvesting equipment using unmanned aerial vehicles" (2020) *Intelligent Automation and Soft Computing*, 26 (1), pp. 121 - 132, doi: 10.31209/2019.100000133.
- [5] Lienkov S., Shvorov S., Sieliukov O., Tolok I., Lytvynenko N., Davydenko T. *Learning of Neural Networks Using Genetic Algorithms* (2022) *CEUR Workshop Proceedings*, 3312, pp. 155 – 164
- [6] Giacomo Certini, Riccardo Scalenghe, William I. Woods "The impact of warfare on the soil environment, *Earth-Science Reviews*" Vol. 127, 2013, pp. 1-15, doi: 10.1016/j.earscirev.2013.08.009.
- [7] Nataliia Kussul, Sofiia Drozd, Hanna Yailymova, Andrii Shelestov, Guido Lemoine, Klaus Deininger, "Assessing damage to agricultural fields from military actions in Ukraine: An integrated approach using statistical indicators and machine learning" *International Journal of*

- Applied Earth Observation and Geoinformation, Vol. 125, 2023, 103562, doi: 10.1016/j.jag.2023.103562.
- [8] Maksym Solokha, Paulo Pereira, Lyudmyla Symochko, Nadiya Vynokurova, Olena Demyanyuk, Kateryna Sementsova, Miguel Inacio, Damia Barcelo, "Russian-Ukrainian war impacts on the environment. Evidence from the field on soil properties and remote sensing " Science of The Total Environment, Vol. 902, 2023, 166122, doi: 10.1016/j.scitotenv.2023.166122.
- [9] Pasichnyk N., Komarchuk D., Opryshko O., Shvorov S. and Kiktev N., "Methodology for Software Assessment of the Conformity of Atmospheric Correction from the UAV's Zenith Sensor," 2021 IEEE 6th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD), Kyiv, Ukraine, 2021, pp. 1-5, doi: 10.1109/APUAVD53804.2021.9615177.
- [10] Pasichnyk N., Opryshko O., Shvorov S., Dudnyk A., Teplyuk V. "Remote field monitoring results feasibility assessment for energy crops yield management" (2023) Machinery and Energetics, 14 (2), pp. 46 - 59, doi: 10.31548/machinery/2.2023.46
- [11] Erik C. Duncan, Sergii Skakun, Ankit Kariryaa, Alexander V. Prishchepov "Detection and mapping of artillery craters with very high spatial resolution satellite imagery and deep learning" Science of Remote Sensing, Volume 7, 2023, 100092, doi: 10.1016/j.srs.2023.100092.
- [12] D.S.Komarchuk, Y.A.Gunchenko, N.A.Pasichnyk, O.A.Opryshko, S.A.Shvorov and V.Reshетиuk, "Use of Drones in Industrial Greenhouses," 2021 IEEE 6th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD), Kyiv, Ukraine, 2021, pp. 184-187, doi: 10.1109/APUAVD53804.2021.9615418
- [13] S.A.Shvorov, N.A.Pasichnyk, S.D.Kuznichenko, I.V.Tolok, S.V.Lienkov and L.A.Komarova, "Using UAV During Planned Harvesting by Unmanned Combines," 2019 IEEE 5th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), Kiev, Ukraine, 2019, pp. 252-257, doi: 10.1109/APUAVD47061.2019.8943842.
- [14] Kiktev N., Opryshko O., Pasichnyk N., Dudnyk A., Komarchuk D. "Remote Monitoring of Mines in Fields with Using Neural Networks" (2023) CEUR Workshop Proceedings, 3624, pp. 239 – 249. https://ceur-ws.org/Vol-3624/Paper_20.pdf
- [15] Ruli Andaru, Jiann-Yeou Rau, Devy Kamil Syahbana, Ardy Setya Prayoga, Heruningtyas Desi Purnamasari "The use of UAV remote sensing for observing lava dome emplacement and areas of potential lahar hazards: An example from the 2017–2019 eruption crisis at Mount Agung in Bali", Journal of Volcanology and Geothermal Research, Vol. 415, 2021, 107255, doi: 10.1016/j.jvolgeores.2021.107255.
- [16] A. Román, A. Tovar-Sánchez, D. Roque-Atienza, I.E. Huertas, I. Caballero, E. Fraile-Nuez, G. Navarro "Unmanned aerial vehicles (UAVs) as a tool for hazard assessment: The 2021 eruption of Cumbre Vieja volcano, La Palma Island (Spain)", Science of The Total Environment, Vol. 843, 2022, 157092, doi: 10.1016/j.scitotenv.2022.157092.
- [17] E.V. Ivokhin, O.V. Oletsky. Restructuring of the Model "State-Probability of Choice" Based on Products of Stochastic Rectangular Matrices. Cybern Syst Anal. Vol.58 (2), pp.242-250 (2022). <https://doi.org/10.1007/s10559-022-00456-z>
- [18] O. Oletsky. On Constructing Adjustable Procedures for Enhancing Consistency of Pairwise Comparisons on the Base of Linear Equations. CEUR Workshop Proceedings, 2021, 3106, pp. 177–185. https://ceur-ws.org/Vol-3106/Short_1.pdf
- [19] Atal Tewari, K. Prateek, Amrita Singh, Nitin Khanna. (2023). Deep Learning based Systems for Crater Detection: A Review. doi: 10.48550/arXiv.2310.07727.
- [20] E. Emami, T. Ahmad, G. Bebis, A. Nefian and T. Fong, "Crater Detection Using Unsupervised Algorithms and Convolutional Neural Networks," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 8, pp. 5373-5383, Aug. 2019, doi: 10.1109/TGRS.2019.2899122.
- [21] Iraklis Giannakis, Anshuman Bhardwaj, Lydia Sam, Georgios Leontidis. (2023). Deep learning universal crater detection using Segment Anything Model (SAM). doi: 10.48550/arXiv.2304.07764.
- [22] Atal Tewari, Vinay Verma, Pradeep Srivastava, Vikrant Jain, Nitin Khanna "Automated Crater detection from Co-registered optical images, elevation maps and slope maps using deep learning", Planetary and Space Science, Vol. 218, 2022, 105500, doi: 10.1016/j.pss.2022.105500.