Artificial Intelligence, Medical Engineering and Education
Z.B. Hu et al. (Eds.)

doi:10.3233/ATDE231310

Neural Networks for Environmental Monitoring: A Web Application for Image Recognition

Nataliya BERNATSKA[a], Elvira DZHUMELIA[b], Iryna TYPILO[a], Orest KOCHAN[c d] [1], Ivan SALAMON[e]

[a] *Department of Physical, Analytical and General Chemistry, Institute of Chemistry and Chemical Technologies, Lviv Polytechnic National University, Lviv, 79000, Ukraine*
[b] *Software Department, Institute of Computer Science and Information Technologies, Lviv Polytechnic National University, Lviv, 79000, Ukraine*
[c] *Department of Information-Measuring Technologies, Institute of Computer Technologies, Automation and Metrology, Lviv Polytechnic National University, Lviv, 79000, Ukraine*
[d] *School of Computer Science, Hubei University of Technology, Wuhan 430068, China*
[e] *Department of Ecology, Faculty of Humanities and Natural Sciences, University of Presov, Presov, 080 01, Slovakia*

ORCiD ID: Nataliya BERNATSKA https://orcid.org/0000-0002-6062-1971
Elvira DZHUMELIA https://orcid.org/0000-0003-3146-8725
Orest KOCHAN https://orcid.org/0000-0002-3164-3821
Ivan SALAMON https://orcid.org/0000-0001-5379-3989

Abstract. Environmental monitoring is increasingly reliant on image recognition through neural networks, automating the identification and classification of environmental threats. This automation optimizes resource allocation, enhances monitoring accuracy, and empowers researchers to address climate-related challenges. In response to this, we present a web application built using React.js and TensorFlow.js, which enables real-time image recognition. This application facilitates tasks ranging from climate analysis to disaster response and wildlife conservation, offering valuable insights into environmental trends. Moreover, it engages the public, promotes awareness, and contributes to conservation efforts, making it a versatile and cost-effective tool for environmental monitoring. The application leverages advancements in image recognition technology, availability of datasets, open-source frameworks, and cloud computing infrastructure. It supports a wide range of applications, promotes global collaboration, and ensures data security. While challenges exist, the potential of neural networks in environmental monitoring is promising. The web application combines the power of React.js and artificial neural networks, enhancing environmental monitoring efficacy and fostering sustainable practices on a global scale. Ongoing research and standardization efforts are essential for refining and expanding this approach.

Keywords. Environmental Monitoring, React.js, Tensorflow.js, Machine Learning, Image Recognition

[1] Corresponding Author: Orest KOCHAN, orest.v.kochan@lpnu.ua.

1. Introduction

Image recognition using artificial neural networks (ANNs) is becoming increasingly important in environmental monitoring due to their ability to process large amounts of data, detect patterns, and extract meaningful information from images. The use of neural networks automates the analysis of huge amounts of image data, reducing the need for manual inspection. ANNs help analyse images related to climate change, including monitoring glacial retreat, sea ice changes, and changes in vegetation patterns. This helps climate researchers to understand and overcome the effects of climate change.
Image recognition helps in predicting and responding to natural disasters. ANNs can analyse satellite or drone imagery to identify potential risk factors, assess the extent of damage during and after natural disasters, and facilitate timely and effective response.
ANNs contribute to the monitoring of protected areas by analysing images for signs of illegal activity, encroachment, or violations. This strengthens conservation efforts and helps protect biodiversity in environmentally sensitive regions.
Image recognition based on ANNs can be integrated into educational tools and public engagement initiatives. Access to interactive applications allows the public to engage in environmental monitoring, increasing awareness and participation in conservation.

2. Literature Review

Developing a web application for image recognition in environmental monitoring is important for several reasons:
- real-time monitoring: a web application allows you to analyse environmental data in real time using image recognition [1, 2]. This capability allows for timely detection and response to changes in environmental conditions, such as deforestation, pollution, or natural disasters [3, 4];
- data accessibility: the web-based platform makes environmental data available to a wider audience. Researchers, policy makers, and the general public can access and interpret image-based information, contributing to greater awareness and understanding of environmental issues [5-7];
- early detection of environmental changes: Image recognition can help in the early detection of environmental changes, allowing proactive measures to be taken to mitigate the impact of factors such as deforestation, habitat loss or climate-related events [8];
- species monitoring and conservation: the use of image recognition to monitor wildlife and plant species supports conservation efforts. It helps to track population trends, identify endangered species, and evaluate the effectiveness of conservation initiatives [9, 10];
- efficient resource management: image recognition can help in the effective management of natural resources. For example, it can be used to monitor agricultural practices, assess land use changes, and optimize resource allocation for sustainable development [11];
- improved decision-making: access to accurate and up-to-date environmental data enables decision makers to make informed choices. This is especially important for policy makers, land managers, and environmentalists who need reliable information for effective environmental planning and management [12, 13];
- public engagement and education: web applications with image recognition can engage the public in environmental monitoring. Visualizing environmental data through user-

friendly interfaces encourages public participation, citizen science initiatives, and environmental education [14, 12];
- cost-effective surveillance: web-based image recognition systems can offer a cost-effective alternative to traditional surveillance methods. Automated analysis of large data sets can replace manual monitoring, reducing costs and increasing the efficiency of monitoring programs [15, 16];
- global collaboration: Web-based applications facilitate global collaboration in environmental monitoring. Researchers and organizations from around the world can share and access data, collaborate on projects, and contribute to a more complete understanding of global environmental trends [17-19].

The development of a web application for image recognition in environmental monitoring has great potential for solving various environmental problems. Here are some key considerations that emphasize its feasibility [18]:

1) Development of image recognition technology:
- Image recognition technology, especially in the field of machine and deep learning, has made significant progress. Algorithms such as Convolutional Neural Networks (CNNs) are excellent at image analysis tasks, making them suitable for environmental monitoring applications [20].

2) Availability of datasets:
- Datasets for environmental imagery, such as satellite imagery, drone footage, or ground-based camera data, are becoming increasingly available. These datasets can be used to train and tune machine learning models, improving the accuracy of image recognition [21, 22].

3) Open source tools and frameworks:
- Numerous open-source tools and frameworks, such as TensorFlow and PyTorch, facilitate the development and deployment of machine learning models for image recognition. These tools provide a solid foundation for building web applications with powerful image analysis capabilities [23, 24].

4) Cloud computing infrastructure:
- Cloud computing services such as AWS, Google Cloud, and Azure offer a scalable and cost-effective infrastructure for hosting web applications with image recognition capabilities. Using these services can make it easier to deploy and maintain your environmental monitoring application [25].

5) Potential use cases:
- Image recognition in environmental monitoring can be applied to a variety of use cases, including deforestation detection, wildlife tracking, pollution monitoring, and land cover analysis. The diverse range of applications demonstrates the versatility of the technology.

6) Real-time monitoring [25, 26]:
- Web-based applications can provide real-time monitoring capabilities, allowing users to get instant information about changes in the environment. This is crucial for timely decision-making and intervention in response to emerging environmental issues [27].

7) User-friendly interface:
- A well-designed web interface can make the application user-friendly, allowing non-specialists to access and interpret environmental data. This democratization of information helps to raise public awareness and engage the public in environmental conservation efforts [22, 28, 29].

8) Integration with IoT devices:

- The web application can be integrated with Internet of Things devices such as sensors and cameras to expand data collection capabilities. This integration provides a more comprehensive approach to environmental monitoring.

9) Security and privacy:

- Given the sensitivity of environmental data, it is important to prioritize security and privacy. Implementing robust security measures and adhering to data protection regulations will help increase the credibility of the app [15].

Neural networks have a number of advantages over traditional methods of environmental monitoring [30, 31]:

- High accuracy: neural networks can be trained on large data sets, which allows them to achieve high accuracy in classification and forecasting tasks;
- speed: neural networks can process large amounts of data quickly and efficiently;
- flexibility: neural networks can be adapted to various environmental monitoring tasks.

However, there are also some challenges associated with using neural networks for environmental monitoring [12-15]:

- Data requirements: neural networks require large data sets to train.
- Overtraining: neural networks can be prone to overtraining, which can lead to reduced accuracy.
- Understanding: It is not always easy to understand how a neural network came to a certain decision.

Despite these challenges, neural networks are a promising tool for environmental monitoring. They have the potential to improve our understanding of environmental problems, predict future changes, and develop more effective solutions for environmental protection [15, 32].

A web application for a neural network is software that allows users to interact with a neural network through a web browser [16-17]. Web applications can be used for various purposes, such as training, testing, and utilizing neural networks [19-37].

The aim of the study was to create a web application for image recognition using artificial neural networks to improve the efficiency of environmental monitoring.

3. Materials and Methods

Libraries and technologies used:

1) React.js is an open-source JavaScript library for creating user interfaces (UI). It is based on the concept of components, which are independent and interacting blocks that can be used to create complex UIs.

2) React-transition-group is a React.js library that allows you to create smooth transitions between components. It uses CSS transitions to create smooth effects such as smoothing, expanding, and contracting.

Main features of react-transition-group:

- Smooth transitions: react-transition-group uses CSS transitions to create smooth effects.
- Transition management: react-transition-group allows you to control transitions with JavaScript.
- Support for various components: react-transition-group supports various components such as text boxes, buttons, and images.

3) Material UI is an open-source React.js library that allows you to create user interfaces (UIs) based on Google's Material Design system. It contains sets of

components that can be used to create various UI elements such as buttons, input fields, menus, and navigation.

Main features of Material UI:

- Components: Material UI contains sets of components that you can use to create different UI elements.
- Style: Material UI supports Google's Material Design design system.
- Expansiveness: Material UI has a large and active community that develops many libraries and tools for Material UI.

4) TensorFlow.js: 1.0.1 is an open-source JavaScript library for machine learning that allows you to create and implement machine learning models in web browsers and Node.js. It is based on TensorFlow, a popular machine learning library for Python.

Main features of TensorFlow.js:

- Ability to create and implement machine learning models in web browsers and Node.js: TensorFlow.js allows you to create machine learning models that can be used in web browsers and Node.js.
- Supports various machine learning tasks: TensorFlow.js supports a wide range of machine learning tasks such as classification, regression, pattern recognition, and speech recognition.
- Ease of use: TensorFlow.js is relatively easy to use.

5) MobileNet: 1.0.0 is a neural network architecture for pattern recognition developed by Google. It was first introduced in 2017. It is designed for use in mobile devices. It is resource efficient, which allows it to run on devices with limited resources, such as smartphones.

MobileNet 1.0.0 consists of two main components:

- MobileNet base model: This is the base model that consists of 12 consecutive convolution layers.
- MobileNet extended model: This is an extended model that consists of 22 consecutive convolution layers.

MobileNet 1.0.0 was trained on the ImageNet dataset, which consists of over 1.2 million images and their classifications.

Main features of MobileNet 1.0.0:

- Resource-efficient: MobileNet 1.0.0 is resource efficient, enabling it to run on devices with limited resources.
- Ability to be used in mobile devices: MobileNet 1.0.0 has been designed to be used in mobile devices.
- Good results in pattern recognition: MobileNet 1.0.0 has shown good pattern recognition results on the ImageNet dataset.

4. Results and Discussion

The main components of a web application for a neural network are:

- Client interface: Users interact with the neural network through a client interface.
- Web server: The web server processes requests from the client interfaces and passes them on to the neural network.
- Neural network: neural network performs image recognition tasks.

Important factors when developing a web application for a neural network:

- Reliability: the web application must be reliable and work without failures.
- Efficiency: the web application should be efficient and perform tasks quickly.

- Convenience: the web application should be easy to use and easily understandable for users.

The effectiveness of a web application for a neural network depends on several factors, including:

- The accuracy of the neural network: The more accurate a neural network is, the more efficient it will be.
- The speed of the neural network: The faster a neural network can perform tasks, the more efficient it will be.
- Ease of use of the web application: The easier it is to use a web application, the more effective it will be.

Benefits of using a web-based image recognition application for environmental monitoring (Fig. 1-5):

- Automation of tasks: web app can automate the task of recognizing objects in images. This can lead to significant time and cost savings.
- Improved accuracy: Web-based image recognition applications can use neural networks trained on large datasets. This can lead to improved object recognition accuracy.
- Speed: our web application can process large amounts of data quickly. This can be important for tasks that require a quick response.

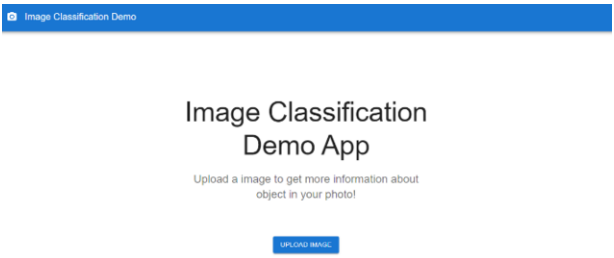

Fig. 1. Image Classification Demo App

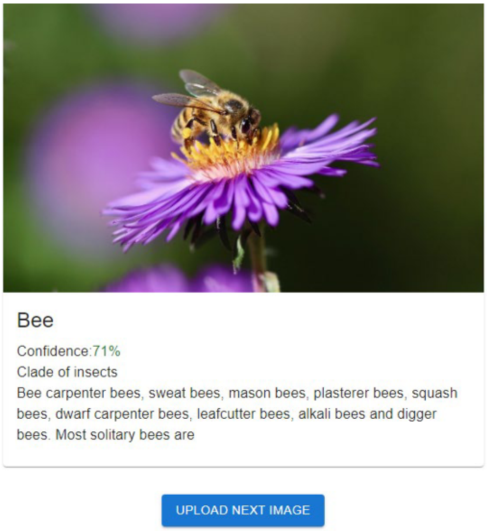

Fig. 2. Screenshot of the web application for image of bee recognition

Fig. 3. Screenshot of the web application for image of bird recognition

Fig. 4. Screenshot of the web application for image of flower recognition

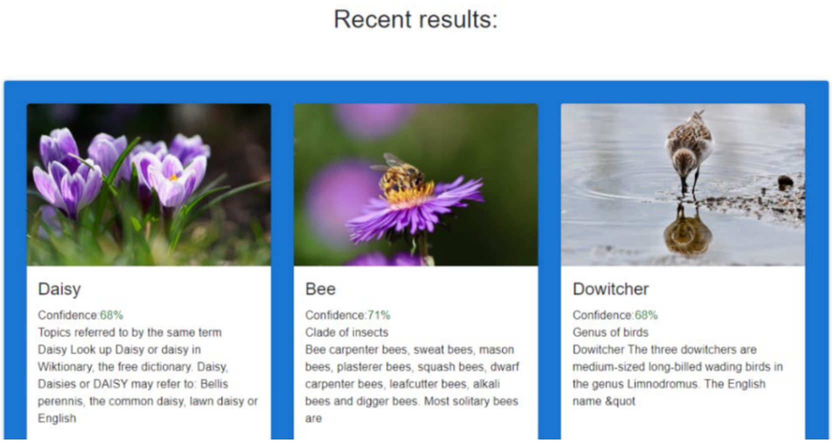

Fig. 5. Screenshot of the web application for image recognition

We used Tensorflow.js together with React.js to build a one-page web application based on TensorFlow.js and the Mobilenet 1.0 artificial neural network model trained on the Imagenet2012 dataset to classify uploaded images and provide a short description about them from the MediaWiki API. The application stores the recent classification in the gallery. MobileNet is a class of convolutional neural network that was open-sourced by Google, and therefore, provides an excellent starting point for training classifiers that are insanely small and insanely fast.

React and artificial neural networks are two powerful tools that can be used to build web applications. React is a JavaScript framework for building user interfaces, and ANNs are a type of machine learning that can be used to train models that can predict or classify data.

Some of the main benefits of using React to build user interfaces include:

- Declarative Model: React uses a declarative model that makes the code more understandable and easier to maintain.
- Component-based architecture: React uses a component-based architecture that makes code more organized and scalable.
- Virtual DOM: React uses a virtual DOM that makes code more efficient and productive.
- Large community: React has a large and active community of developers who are constantly creating new tools and libraries.

One way to use React with ANNs is to create user interfaces that adapt to individual user needs. For example, an ANN can be used to train a model that can predict which UI features will be most useful to a particular user. This model can then be used to tailor the user interface to the user's individual needs.

Another way to use React with ANNs is to create user interfaces that can interact with the real world. For example, an ANN can be used to train a model that can recognize objects on a camera. This model can then be used to create a user interface that allows users to interact with objects on the camera.

TensorFlow.js is an open-source library developed by Google that allows for the training and deployment of machine learning models directly in the browser or on Node.js. It enables developers to build and run machine learning applications using JavaScript. MobileNet is a family of neural network architectures designed for efficient on-device computer vision, particularly suited for mobile and embedded applications.

Key Features:

- flexibility: TensorFlow.js supports both training and inference in the browser, which is useful for applications requiring real-time updates and interactivity;
- cross-platform: It works seamlessly on various platforms, including web browsers and Node.js, enabling a wide range of applications;
- TensorFlow.js can be integrated with other web technologies, allowing for the creation of interactive and dynamic web applications.

MobileNet is a family of neural network architectures designed for mobile and embedded vision applications. MobileNetV1, in particular, is known for its efficiency and ability to perform image classification with a relatively small computational footprint.

Key Features:

- MobileNetV1 uses depthwise separable convolutions, which reduce the number of parameters and computations compared to traditional convolutional layers.
- The architecture is designed to be lightweight and efficient, making it suitable for deployment on resource-constrained devices such as mobile phones and IoT devices.
- Pre-trained versions of MobileNetV1 are available, allowing developers to leverage transfer learning for tasks like image recognition.

TensorFlow.js provides pre-trained models, and MobileNetV1 is one of them. Developers can use MobileNetV1 in the browser for tasks such as image classification without the need for server-side computation. There is the web app code example using TensorFlow.js and React.js (fig. 6-7).


```
import ResultsGallery from './ResultsGallery';
import SingleImage from './SingleImage';
import { CSSTransition, TransitionGroup } from 'react-transition-group';

const MobileNet = require('@tensorflow-models/mobilenet');

const MainApp = () => {
    const [results, setResults] = useState([]);
    const [gigaState, setGigaState] = useState(false);
    const fileInputRef = useRef(null);
    const showWelcomeTitlesFlag = useRef(false);
    const isImageBeingProcessed = useRef(false);
    const [loadMobileNetModel, setMobileNetModel] = useState(null);
    const [image, setImage] = useState({ src: null, title: null, excerpt: null, description: null, confidence: null });


    useEffect(() => {
        setMobileNetModel(async () => await MobileNet.load());
    }, []);

    useEffect(() => {

        if (showWelcomeTitlesFlag.current) {
            showWelcomeTitlesFlag.current = false;
            isImageBeingProcessed.current = false;
```

Fig. 6. Screenshot of the project code

```
Code  Blame  136 lines (111 loc) · 5.16 KB                                    Raw
const MainApp = () => {
            setGigaState(true);
            setResults(results => [...results, image])
        }
        return;
    }, [image]);



    const loadMobilenetImage = (src) => {
        return new Promise((resolve, reject) => {
            const img = new Image();
            img.addEventListener('load', () => resolve(img));
            img.addEventListener('error', (err) => reject(err));
            img.src = src;
        })
    }

    //This function could be memoized
    const capitalizeFirstLetter = (string) => {
        return string.charAt(0).toUpperCase() + string.slice(1);
    }

    const handleImageChange = (e) => {
        isImageBeingProcessed.current = true;
```

Fig. 7. The project code

Ensuring the reliability and accuracy of results in image recognition for environmental monitoring is crucial. Here are some critical steps to achieve this:

1) High-Quality Training Data. The start with a comprehensive and diverse dataset for training your neural network. The dataset should represent the types of images your application will encounter. Quality data is the foundation of accurate results.

2) Data Preprocessing. Preprocessing and cleaning the data to remove noise, artifacts, and inconsistencies. This step is essential to ensure the neural network learns from clean and relevant information.

3) Data Augmentation. Augmenting the dataset with variations of the original images, such as lighting, rotation, and scale changes. Data augmentation helps improve the model's generalization and robustness.

4) Cross-validation. Implementing cross-validation techniques to assess the model's performance. This involves splitting the dataset into multiple subsets for training and testing, providing a more comprehensive evaluation of the model's accuracy.

5) Hyperparameter Tuning. The experiment with various hyperparameters, such as learning rate, batch size, and network architecture. Fine-tuning these parameters can significantly impact the model's performance.

6) Regularization Techniques. Using regularization methods like dropout and L2 regularization to prevent overfitting. Overfit models may perform well on the training data but poorly on new, unseen data.

7) Ensemble Models. Considering using ensemble methods to combine predictions from multiple models. Ensemble methods often lead to more accurate results by reducing individual model biases.

8) Continuous Monitoring. Regularly monitoring the model's performance in a real-world environment. Retraining the model periodically to adapt to changing conditions and improve accuracy over time.

9) Confidence Scores. Implementing a confidence scoring system to gauge the model's certainty about its predictions. This allows us to filter out low-confidence results or seek human verification when necessary.

10) Human Oversight. Maintaining a human oversight component in the system. Experts review and validate a portion of the results to ensure accuracy, especially for critical applications.

11) Feedback Loops. Establishing feedback loops where user-provided feedback is used to improve the model. Correctly identifying inaccuracies or false positives can be fed back into the training process.

12) Error Analysis. Conducting thorough error analysis to identify common failure patterns. This can guide improvements in the model and data collection strategies.

13) Interpretability. Using interpretable models or techniques to make the model's decisions more understandable. Explainable AI can enhance trust in the model's results.

14) Documentation. Maintaining comprehensive documentation for the entire pipeline, including data sources, preprocessing steps, model architecture, and evaluation metrics. This ensures transparency and reproducibility.

15) Compliance and Regulations. Adhering to relevant regulations and standards for environmental monitoring, data privacy, and model deployment. Compliance contributes to the reliability of our results.

16) User Education. Educating users and stakeholders about the model's capabilities and limitations to manage expectations and enhance trust.

React and ANNs are rapidly evolving areas of technology. As these technologies improve, they have the potential to revolutionize web application development by making it more efficient, accessible, and informative.

5. Conclusions

The development of a web application for image recognition in environmental monitoring represents a significant advancement in the field. The application's capabilities offer several key advantages:

Task Automation: Automation is at the core of our web application, streamlining the identification and classification of objects in images. This not only results in substantial time and cost savings but also reduces the margin for human error.

Improved Accuracy: Our web-based image recognition application leverages neural networks trained on extensive datasets, which translates to a marked improvement in the precision of object recognition. This is particularly critical in environmental monitoring, where data accuracy is paramount.

Speed and Efficiency: Our application excels in processing vast volumes of data swiftly, a crucial aspect for tasks that demand immediate response, such as disaster management and early intervention in environmental challenges.

Image recognition technologies have rapidly become an integral part of contemporary environmental monitoring. These technologies are poised to significantly impact our understanding of environmental issues, forecast future changes, and craft more effective solutions for environmental protection.

However, it is vital to acknowledge and address the challenges inherent in the use of neural networks for environmental monitoring. Data requirements, overtraining, and the interpretability of neural network decisions are hurdles that must be overcome to harness the full potential of this technology. Moving forward, continued research is imperative to refine and enhance the accuracy and efficiency of neural network models. Furthermore, the establishment of standards for the training and evaluation of neural networks in the realm of environmental monitoring is essential to ensure the credibility and accuracy of these models.

In conclusion, the fusion of web development technologies and image recognition through artificial neural networks represents a substantial step forward in environmental monitoring. It provides a wealth of information crucial for solving environmental issues, promoting sustainability, and safeguarding ecosystems both locally and globally. As technology continues to evolve, the prospects for more accurate and efficient models in environmental monitoring remain promising.

Acknowledgment

This paper is supported by the National Research Foundation of Ukraine, project number 0123U103529 (2022.01/0009).

References

[1] Zhang et al. A review of deep learning for environmental monitoring. 2022, Sensors.
[2] Sun, L., Qin, H., Przystupa, K., Majka, M., & Kochan, O. Individualized Short-Term Electric Load Forecasting Using Data-Driven Meta-Heuristic Method Based on LSTM Network. Sensors, 22(20), 2022, 7900.
[3] Zhengbing, H., Jotsov, V., Jun, S., Kochan, O., Mykyichuk, M., Kochan, R., & Sasiuk, T. (2016, September). Data science applications to improve accuracy of thermocouples. In 2016 IEEE 8th International Conference on Intelligent Systems (IS), pp. 180-188.
[4] Kunrong Z., Tingting H., Shuang W., Songling W., Bilan D., Qifan Y. (2018). Application research of image recognition technology based on CNN in image location of environmental monitoring UAV. EURASIP Journal on Image and Video Processing. 50. pp. 1-11.
[5] Liu et al. Deep learning for environmental monitoring: A survey. 2022, IEEE Access.
[6] Zhengbing Hu, Yevgeniy V. Bodyanskiy, Nonna Ye. Kulishova, Oleksii K. Tyshchenko, "A Multidimensional Extended Neo-Fuzzy Neuron for Facial Expression Recognition", International Journal of Intelligent Systems and Applications (IJISA), Vol.9, No.9, 2017, pp.29-36.
[7] Wang et al. Application of deep learning in environmental monitoring. 2021, Earth Science Informatics.
[8] Zhang et al. Deep learning for environmental monitoring: A review of recent advances. 2020, IEEE Access
[9] Wang et al. Deep learning for environmental monitoring: A survey of methods and applications. 2019, IEEE Transactions on Geoscience and Remote Sensing.
[10] Zhang et al. Deep learning for remote sensing of the environment. 2023, Remote Sensing
[11] Wang et al. Deep learning for environmental data analysis. 2022, Environmental Science & Technology
[12] Zhang et al. Deep learning for environmental modeling. 2021, Environmental Modelling & Software
[13] Zhengbing Hu, Mykhailo Ivashchenko, Lesya Lyushenko, Dmytro Klyushnyk. Artificial Neural Network Training Criterion Formulation Using Error Continuous Domain", International Journal of Modern Education and Computer Science (IJMECS), Vol.13, No.3, 2021, pp. 13-22.
[14] Zhengbing Hu, Sergii V. Mashtalir, Oleksii K. Tyshchenko, Mykhailo I. Stolbovyi, "Video Shots' Matching via Various Length of Multidimensional Time Sequences", International Journal of Intelligent Systems and Applications (IJISA), Vol.9, No.11, pp.10-16, 2017.
[15] Karpinski M., V. Pohrebennyk V., Bernatska N., Ganczarchyk J., Shevchenko O. Simulation of Artificial Neural Networks for Assessing the Ecological State of Surface Water. 18th International Multidisciplinary Scientific GeoConference SGEM 2018. 2 July – 8 July 2018, Albena, Bulgaria. P. 693-700.
[16] Spencer MCDonald. Web-based decision support system tools: The Soil and Water Assessment Tool Online visualization and analyses (SWATOnline) and NASA earth observation data downloading and reformatting tool (NASAaccess). Environmental Modelling & Software. 2019. Volume 120. P. 1-12.
[17] Jun, S., & Kochan, O. (2015). Common mode noise rejection in measuring channels. Instruments and Experimental Techniques, 58, 86-89.
[18] Srivastava S., Vaddadi S., Sadistap S., Smartphone based System for water quality analysis. Applied Water Science, 2018., P. 129-130.

[19] Crimi A, Jones T. Sgalambro A. Designing a Web Spatial Decision Support System Based on Analytic Network Process to Locate a Freight Lorry Parking. Sustainability. 2019. 11. P. 1-14.
[20] Al-Rubaie F. Application of Artificial Neural Network and Geographical Information System Models to Predict and Evaluate the Quality of Diyala River Water, Iraq / F. Al-Rubaie, N. Al-Musawi // Association of Arab Universities Journal of Engineering Sciences, 2019, 26(1), pp. 160–169.
[21] Bernatska N., Dzhumelia E., Kochan O. Development of the web interface of the information-analytical system of environmental monitoring using the REACT library. Information technologies and society, 2023, No. 1 (7), pp. 6–12. (In Ukr.)
[22] Pohrebennyk, V., Dzhumelia, E. Environmental assessment of the impact of tars on the territory of the Rozdil state mining and chemical enterprise “Sirka” (Ukraine). Studies in Systems, Decision and Control, 2020, 198, pp. 201–214.
[23] Pohrebennyk, V., Korchenko O., Mitryasova O., Bernatska N., Kordos M. An Analytical Decision Support System in Prognostication of Surface Water Pollution Indicators. 19th International Multidisciplinary Scientific GeoConference SGEM 2019, 2019, 2.1, pp. 49-56.
[24] Pohrebennyk, V., Koszelnik, P., Mitryasova, O., Dzhumelia, E., & Zdeb, M. Environmental monitoring of soils of post-industrial mining areas. Journal of Ecological Engineering, 20(9), 2019, 53–61. https://doi.org/10.12911/22998993/112342.
[25] Al-Ashwal N.H., Al Soufy K.A.M., Hamza M.E., Swillam M.A. Deep Learning for Optical Sensor Applications: A Review. Sensors. 2023, 23(14):6486. https://doi.org/10.3390/s23146486.
[26] Chen, P. Visualization of real-time monitoring datagraphic of urban environmental quality (2019) Eurasip Journal on Image and Video Processing, (1), 2019, art. no. 42. doi: 10.1186/s13640-019-0443-6
[27] Chen, L., Zhang, L. Design of Ecological Environment Monitoring System Based on Internet of Things Technology. Wireless Communications and Mobile Computing, 2022, art. no. 2531425.
[28] Havrys, A., Yakovchuk, R., Pekarska, O., Tur, N. Visualization of Fire in Space and Time on the Basis of the Method of Spatial Location of Fire-Dangerous Areas. Ecological Engineering and Environmental Technology, 24 (2), 2023, pp. 28-37.
[29] Novak, R., Petridis, I., Kocman, D., Robinson, J.A., Kanduč, T., Chapizanis, D., Sarigiannis, D. Harmonization and visualization of data from a transnational multi-sensor personal exposure campaign. International Journal of Environmental Research and Public Health, 18 (21), 2021, art. no. 11614.
[30] Shpynkovska, M., Bakunova, A. Information system for visualization of the ecological state of the region (2015) Automation of Technological and Business Processes, 7, pp. 75-81.
[31] Sohor, A., Brydun, A., Yarema, N., Lozynskyi, V., Sohor, M. (2020) Interactive map of environmental pollution of the surface waters in Lviv Region, pp. 1-5.
[32] Fayyad, J., Jaradat, M.A., Gruyer, D., et al. Deep learning sensor fusion for autonomous vehicle perception and localization: Areview. Sensors, 20(15), 2020, 4220 (3).
[33] Priya R. Maidamwar, Prasad P. Lokulwar, Kailash Kumar. Ensemble Learning Approach for Classification of Network Intrusion Detection in IoT Environment. International Journal of Computer Network and Information Security, Vol.15, No.3, pp.30-46, 2023.
[34] Mohammed Abduljabbar Zaid Al Bayati, Muhammet Çakmak. Real-Time Vehicle Detection for Surveillance of River Dredging Areas Using Convolutional Neural Networks. International Journal of Image, Graphics and Signal Processing, Vol.15, No.5, pp. 17-28, 2023.
[35] Dhekra El Hamdi, Ines Elouedi, Mai K Nguyen, Atef Hamouda. A Conic Radon-based Convolutional Neural Network for Image Recognition. International Journal of Intelligent Systems and Applications, Vol.15, No.1, pp.1-12, 2023.
[36] Banothu Balaji, T. Satyanarayana Murthy, Ramu Kuchipudi. A Comparative Study on Plant Disease Detection and Classification Using Deep Learning Approaches. International Journal of Image, Graphics and Signal Processing, Vol.15, No.3, pp. 48-59, 2023.
[37] Ihor Tereikovskyi, Zhengbing Hu, Denys Chernyshev, Liudmyla Tereikovska, Oleksandr Korystin, Oleh Tereikovskyi. The Method of Semantic Image Segmentation Using Neural Networks. International Journal of Image, Graphics and Signal Processing, Vol.14, No.6, pp. 1-14, 2022.