A Literature Review on Real-Time Image Classification for Dragonfly Species Using TensorFlow.js and Biodiversity Monitoring.

Orion Liçi, Inva Bilo, Karafil Kareçi and Ana Ktona

July 3, 2025

A Literature Review on Real-Time Image Classification for Dragonfly Species Using TensorFlow.js and Biodiversity Monitoring.

Orion Lici[1], Inva Bilo[2], Karafil Kareci[3] and Ana Ktona[4]

1,3,4 Department of Informatics, Faculty of Natural Sciences, University of Tirana, Tirana, Albania

2 Department of Informatics, Faculty of Natural Sciences, University of Gjirokastra, Tirana, Albania

Abstract

In this paper we will explore TensorFlow.js, a JavaScript library for using machine learning in browsers. The main goal is to compare and classify different species of dragonfly based on their visual characteristics by using machine learning models and this paper prepares the ground. The study investigates the effectiveness of TensorFlow.js for analyzing and classifying images in real time and shows the potential of Artificial Intelligence to assist in ecological studies, biodiversity conservation, and entomological research. Based on that classification, images can be stored in a back-end application, by having a store confirmation button in the front-end application. The system should use a camera to capture images, classify them using a convolutional neural network (CNN) model, and store the classified images. The performance of the system will be evaluated based on accuracy, speed, and scalability.

Key words**:** TensorFlow.js, machine learning, browser-based, biodiversity conservation, entomological research, dataset diversity, convolutional neural network

Introduction

The rapid advancement of AI has opened the door for more efficient and automated processes in various fields like medicine, education, tourism [Ktona et al., 2023; Kika et al., 2023; Ktona et al., 2022; Koroveshi et al., 2021], biodiversity from environmental monitoring to wildlife protection [Reynolds et al., 2025; Olawade et al., 2024; Xhina et al., 2023], and so on. One type of application is the real-time classification and detection of different species of insects, such as dragonflies, using AI image recognition technologies. Dragonflies, with their diversity and ecological significance, serve as an ideal subject for studying the potential of real-time image classification in biodiversity research.

This literature review explores the different usage of TensorFlow.js, a powerful JavaScript library, for creating real-time image classification and object detection system. The focus is on comparing various dragonfly species by leveraging pre-trained models and custom datasets. TensorFlow.js offers huge opportunity to run machine learning models directly in the browser through the development of lightweight, real-time solutions without consuming extensive server-side infrastructure, and offering more data privacy as the row data are processed on client browser.

The main goal of this literature review is to contribute to the growing field of real-time wildlife monitoring through innovative AI techniques and provide a clear path for implementing similar solutions in various ecological fields.

This paper is organized in “Background” section that mentions different AI technologies. Then a comparison between them according to our needs, following up with the limitations of each of them. Also describing actual role of IA and possible improvement by having a look at similar systems.

1. Background

Convolutional Neural Networks (CNNs) are a special class of deep neural networks created to handle grid data structures, such as images, with remarkable success. One of the first works in CNNs dates to 1989. It was the development of the **LeNet-5** architecture, which paved the way for modern image classification tasks (LeCun et al., 1989).

CNNs are applied in different domains, from medical image analysis (Tajbakhsh et al., 2016) to autonomous driving systems (Tran et al., 2021). CNNs have also been applied to classify and detect various animal species from images (Tannous et al., 2023). Their flexibility in recognizing different patterns has made CNNs the most used architecture for image classification tasks in academic research and practical applications.

TensorFlow.js is a JavaScript library developed by Google. It can be used for both training and deploying machine learning models in the browser. This way, it powers users with machine learning features without the need for server-side infrastructure. This feature is very useful for real-time applications, such as the live monitoring of species, where fast data processing is a key point.

TensorFlow.js gives you the possibility to deploy pre-trained models as well as training new models directly within the browser environment. This is how TensorFlow significantly reduces latency and improves the user experience. It sustains a variety of tasks, among them, image classification, object detection, and natural language processing, making it one of the best tools for different types of application.

One of the main advantages of using TensorFlow.js is the fact that it can be integrated into new and existing web-based applications with minimal effort. As we already know, Web-based applications can easily incorporate real-time image classification and object detection functionalities without the need of an external server or expensive infrastructure. This gives us a lightweight application for real-time environmental monitoring and wildlife tracking, as is the case in the dragonfly species classification explored in this thesis.

TensorFlow.js also allows us to use trained models for object detection tasks on other platforms such as e.g., MobileNet, COCO-SSD (Rivera, 2020) These models, trained on TensorFlow or other platforms, can be improved by using the dragonfly image dataset through the technique of transfer learning (Zhuang et al., 2021; He et al., 2021; Hosna et al., 2022) However, the ability to train models directly within the browser offered by TensorFlow offers the AI developer unique opportunities to interact with and customize the application.

2. Comparison between TensorFlow.js and other Machine Learning Frameworks for Web-Based Applications

TensorFlow.js is one of the most powerful frameworks for web-based applications but, as in all fields, here too there exist competitors. Each has their own distinctive features. Let's try to wrap up the differences among them, as well as their advantages and drawbacks.

- TensorFlow.js vs. TensorFlow (Python-based) - TensorFlow (the Python-based library) is practically the predecessor of TensorFlow.js. (Abadi et al., 2016). It offers a comprehensive ecosystem, lots of tools, libraries and pre-trained models. All these features make it one of the most perfect frameworks for complex machine learning tasks. TensorFlow offers a lot of flexibility and power by easing the training process for large models and manipulating large datasets. However, it is a server-side framework and it requires sending information from client to server, processing it on the server side and sending the result back to the client. as a result, it requires considerable bandwidth and expose the end user to security concerns. On the other hand, TensorFlow.js's ability to run directly in the browser without requiring server-side processing provides us with enhanced security features and significantly less resources needed for the network and server-side. This is perfect for applications that require real-time inference without reliance on a network connection.
- TensorFlow.js vs. PyTorch (Python-based) - PyTorch is another powerful deep learning framework. Its main strength is dynamic computational graph, and it is very easy to use. While it offers competitive tools for training models, it is still a server-side framework (Paszke et al., 2019). This constrains force developers to a usforcinger web-based framework such as ONNX or simply using it as a backend, thus creating the same drawback as TensorFlow (Python-based). On the other hand, TensorFlow.js natively supports real-time deployment within the browser. This feature makes it more efficient for web-based applications that require low-latency predictions.
- TensorFlow.js vs. Keras.js - Another frontend library in this field is Keras.js (Smilkov et al., 2019). It offers a lightweight solution for the interface. The main drawback of this framework is that it cannot train the models by themselves. Instead, it relies on pretrained models on Keras (Python-based). Meanwhile, TensorFlow.js is more versatile due to its functionality in supporting both model training and interface. Furthermore, it also offers the functionality to use pre-trained models on TensorFlow (Python-based). This extended capability makes it far more competitive than Keras.js
- TensorFlow.js vs. ONNX.js - ONNX.js is a Javascript library for running ONNX models on browsers and on Node.js (Goh et al., 2023). It has adopted WebAssembly and WebGL technologies to provide an optimized ONNX model inference runtime for both CPUs and GPUs. With ONNX.js, web developers can score pre-trained ONNX models directly on browsers with various benefits including a reduction in server-client communication and protection of user privacy, as well as offering an install-free and cross-platform in-browser ML experience. The main drawback compared with TensorFlow.js is that it cannot train new models on the web. In addition, the lack of active maintenance is a downside because this library is in process to be replaced by the "ONNX Runtime Web" library.

3. Existing Methods for Species Identification and Their Limitations

Species identification is a challenging task. It is crucial in ecological studies, especially when dealing with biodiversity monitoring, conservation efforts and environmental protection. Manual methods for species identification are derived from field-based knowledge. It requires experts or taxonomists to visually inspect different species and classify them based on physical characteristics, behavior or other distinguishing features. This method requires one of the following techniques.

Field Observations - This method requires the expert to observe species in the wild. This assumes that experts master a sharp knowledge of the visual characteristics, behavior and ecology. In some situations, for specific species, it could be effective, but it is a time-consuming task. It is also limited by the physical location of the habitat and its level of accessibility. Due to these limitations the study by McClinton et al. (2022) uses both field observations (on-the-ground data) and remote sensing tools to identify the most significant threats to critically endangered or rare plant species in Nevada.

Morphological Identification – Physical characteristics are the focus of this technique including -but not limited to- size, color, shape, etc. (Juan Liu et al., 2015). Yang et al. (2022) developed a convolutional neural network method. Morphological and molecular data for species identification are integrated into the morphology-molecule network (MMNet). This is extremely difficult, especially for species that mimic others or for species at different life stages. Furthermore, it has been proven extremely difficult for closely related species. All of these complications can lead to non-feasible usage of this methodology in the field.

Taxonomic Keys - These are dichotomous keys that allow the user to determine the identity of items using a sequence of alternative choices (Dalton et al., 2024). Dichotomous keys always give two mutually exclusive choices in parallel statements. The user makes a choice about a particular characteristic of an organism and is led to a new branch or couplet of the key. This technique is complex, especially for non-experts and field-amateurs.

Despite the constant improvements that are leading to better and faster classification, these techniques have proven to be prone to errors and is overall a slow process. Another huge limitation is the level of expertise required for this task. All these limitations lead to the lack of scalability and efficiency of these techniques.

4. Role of Machine Learning in Biodiversity Monitoring

Over the last number of years, machine learning (ML) has dramatically improved species identification, offering a highly performant process (Dalton et al., 2024). Another benefit is that it can be scalable very easily. The latest improvements in ML also offer a higher level of accuracy than that offered by traditional methods. This latest development in ML enabled large-scale biodiversity monitoring by significantly reducing costs. Applying machine learning in biodiversity monitoring requires various techniques and areas, which includes image recognition and classification, audio analysis and environmental data processing.

4.1 Advantages of ML for Biodiversity Monitoring

Image Recognition and Classification - One of the most useful applications of machine learning is Convolutional Neural Networks (CNNs) for image-based identification. This important feature is also applied in species identification. The application of CNN in combination with camera traps, drones or smartphones has been very useful for image-based species identification. Based on these techniques, many models have been designed, such as ResNet, Inception, and MobileNet**.** They offer an automatic species identification in large datasets, which would have been impossible to process and analyze manually.

Object Detection - Image classification is a very useful feature, but it is not enough. Object detection models like YOLO (You Only Look Once) and Faster R-CNN, can distinguish and precisely locate multiple species in a single image. Working beyond image classification, these models can distinguish different species in addition to identifying their precise location within the image. This data can be used for other biological analysis regarding animal behavior, their density per square meters, or biodiversity distribution (Redmon et al., 2016).

Environmental Data Integration - Machine learning can be used to predict species distribution and monitor biodiversity over time based on environmental changes. These environmental changes can include -but are not limited to- climate change, soil pollution, vegetation type, etc. Some techniques like Random Forests and Gradient Boosting are used to develop models that predict species frequency or their presence based on environmental factors (Elith et al., 2006).

By using a combination of machine learning and existing ecological data, we could achieve some advantages as set out below:

Scalability - Machine learning models do not have the limitations faced by human experts. They can process huge datasets, examine millions of images or other electronic data far faster and without complaining. This makes it possible to monitor large areas that have multiple species to monitor.

Automation - Machine learning algorithms can offer real-time species identification. This leads us to automatic selection. So, while using camera traps or live audio recordings we can instantly gather results from the field about the species we are interested in.

Accessibility - By simplifying the selection process through machine learning, we can settle in this process also non-expert but enthusiastic naturalists. This is important when many citizens are invited to contribute to their local environment.

Improved Accuracy - Machine learning models, when trained properly and highly tuned with real life datasets, especially those that are considerably large, can offer astonishing levels of accuracy in species identification. They always surpass the field expert.

That being said, while machine learning in biodiversity can be a very powerful tool, there are also some limitations which include:

Data Quality and Quantity - Machine learning models must be provided with large, high-quality datasets for training. In some cases, such datasets may be difficult to provide. This can jeopardize the creation of new models for new or rare species. (Tabak et al., 2018).

Model Generalization - One model trained in specific environmental data may not perhaps be useful in gathering new environment data or similar species beyond its particular dataset. This limitation requires our models to be frequently tuned with new data.

Bias in Training Data - There are some cases when training data is biased regarding habitat or environmental conditions. Machine learning could wrongly develop new models based on biased predictions. This could lead to potentially incorrect classifications for that species.

One famous tool for developing, training and deploying machine learning models in web browsers is TensorFlow.js. It is built by the same team that previously built the TensorFlow (Python-based) library, which is why they share so many common features. However, they have their differences too. TensorFlow supports a number of programming languages, including Python and Java.TensorFlow.js is built only on JavaScript, this way it can be widely used via the web-browser. This feature makes it the perfect framework for real-time machine learning applications. This model enables us to empower environmental monitoring and ecological research.

Real-Time Inference - The capability to operate on the front-end, offering real-time inference without the need of a back-end server support makes it one of the perfect tools in this field. Processing machine learning models directly on the front-end eliminates the latency required by other back-end-based competitors. This way it offers immediate feedback from machine learning predictions. It is this feature that gives TensorFlow.js a huge advantage in ecological monitoring where data captured by cameras could be processed in real time.

4.2TensorFlow.js approach

Browser-Based Deployment - The main advantage of TensorFlow.js is the fact that it can run machine learning models directly in the web-browser or on Node.js. This makes it usable on almost all mobile devices that have their own operating system, such as smartphones or other portable devices. It eliminates the necessity for a cutting edge back-end infrastructure like the cloud and makes it the perfect tool for processing machine learning models on devices with reduced resources. TensorFlow.js enables lightweight web applications to be powered with real-time application processing, making it accessible on any device with a web browser. This factor is the main reason for such a huge expansion in ecological monitoring projects.

No Server-Side Dependency - TensorFlow.js runs on the client-side, in this way it does not require a cutting edge back-end environment. All it needs is simply a normal back-end to host it as a normal web application. All calculations are made on the client-side, making it in practice a real-time and low-latency application. This feature is extremely beneficial in scenarios where data is generated and manipulated in the field and thus perfectly matches our case. It also keeps maintenance and operational costs at an extremely low level, because we can use many small devices to minimize server bottlenecks.

Ease of Integration with Web Technologies - TensorFlow.js is practically a JavaScript Library. This way it can perfectly be integrated into web technologies with other JavaScript libraries, HTML5 and CSS3. It makes it possible for developers to easily incorporate machine learning models into web-based interfaces, providing developers an easy way to create intuitive applications with which to interact. For example, in ecological studies where we need to process locally data captured in the field and send the results back to the server, TensorFlow.js offers a platform to develop real-time applications.

Support for Pre-Trained Models - TensorFlow.js is created by the same team that created TensorFlow(python-based). All models created with other machine learning are also compatible with TensorFlow.js. Therefore, pre-trained models (such as MobileNet, COCO-SSD, or PoseNet) that are already optimized can be easily used in TensorFlow.js by simply importing them into the web browser.

Cross-Platform Compatibility - A wide range of platforms support TensorFlow.js from computer browsers to smartphones, leading to the use of that kind of application across diverse environments. This feature is crucial for ecological research as it offers the possibility to use the system in both environments, in field settings and laboratory conditions.

5. Applicability of TensorFlow.js with Biodiversity Monitoring

Artificial intelligence applications, especially those in machine learning frameworks built on TensorFlow.js for real-time ecological research are becoming the normal standard nowadays. Next, we will take a look at some interesting types of machine learning used for monitoring biodiversity and species identification.

Camera Trap Species Identification - One very serious work on this field is that of Tabak et al. (2018) who applied deep learning algorithms to camera trap images. The purpose was to identify wildlife species in real-time, quite the same idea that we aim to achieve in our work. Their system offers automatic classification of the animals directly from camera traps, in this way eliminating the need for expert classification. TensorFlow.js, can be perfectly used in such scenarios to power a browser-based system with real-time identification features from camera trap images. This feature will speed up the process and make research more economically feasible and faster than traditional methods.

Insect Species Detection - When it comes to insect species detection, the identification process is far more difficult due to their small size and similar appearances across species. By using a combination of machine learning models and high-resolution images we could achieve our goal. TensorFlow.js is a very powerful tool for implementing this kind of real-time system in mobile applications or web platforms, providing researchers the possibility to collect and analyze data in the field. In this way it contributes to a more accurate assessment of biodiversity.

6. Conclusion

TensorFlow.js is a powerful tool for creating real-time applications for ecological research. It is especially valuable in species identification and biodiversity monitoring. Its feature to power in-browser machine learning, without the need for a cutting-edge server-side infrastructure makes it a very popular tool used in real-time applications. The case studies presented here are a perfect example that shows the power of machine learning in ecological research. It becomes more crucial when it comes to automated species identification in real-time. More and more ecological research projects use web-based technologies in combination with machine learning, so it is expected to lead to an exponential growth of TensorFlow.js usage, making the entire process more accessible, scalable and efficient. Additionally, the veteran tool in this field, CNN, is a consolidated solution since it runs on the server-side and can be used for more complex solutions such as high-resolution images with complicated features. The accuracy level of both TensorFlow.js and CNN are identical, as both have been built by the same team behind TensorFlow (python-based); that being said, CNN can run faster because it is generally sustained by a more powerful hardware infrastructure. The advantage of TensorFlow.js is that it offers low latency. As system availability and accuracy improves and models are becoming more and more accurate, it is expected that this technology will play a pivotal role in global conservation efforts and ecological research.

References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. Proceedings of the 12th USENIX Symposium on Operating Systems, November 2‑4, 2016, Savannah, GA, USA. arXiv preprint arXiv:1603.04467.
- Dalton, D., Berger, V., Kirchmeir, H., Adams, V., Botha, J., Halloy, S., Hart, R., Švara, V., Torres Ribeiro, K., Chaudhary, S., & Jungmeier, M. (2024). A framework for monitoring biodiversity in protected areas and other effective area-based conservation measures: Concepts, methods and technologies. *UCN WCPA Technical Report Series* No. 7, Gland, Switzerland: IUCN.
- Elith, J., Phillips, S.J., Hastie, T., Dudík, M., Chee, Y.E., & Yates, C.J. (2011). A statistical explanation of MaxEnt for ecologists. *Diversity and Distributions*, 17: 43-57. https://doi.org/10.1111/j.1472-4642.2010.00725.x
- Goh, H.A., Ho, C.K., & Abas, F.S. (2023). Front-end deep learning web apps development and deployment: a review. *Applied intelligence*. Dordrecht, Netherlands, *53*(12), 15923–15945. https://doi.org/10.1007/s10489-022-04278-6
- He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., & Neubig, G. (2021). Towards a Unified View of Parameter-Efficient Transfer Learning. ArXiv, abs/2110.04366.
- Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., & Azim, M.A. (2022). Transfer learning: a friendly introduction. *J Big Data* 9, 102. https://doi.org/10.1186/s40537-022-00652-w

- Kika, A., Dashi, F., & Alimehmeti, R. (2023). Graph Convolutional Network Approach for Predicting the Severity of Carpal Tunnel Syndrome. *Transactions on Engineering and Computing Sciences*, 11(5), 38–44. https://doi.org/10.14738/tecs.115.15441
- Koroveshi, J. & Ktona, A. (2021). Training an intelligent tutoring system using reinforcement learning. *International Journal of Computer Science and Information Security, 19*(3), 10–18.
- Ktona, A., Mitre, A., Shehu, D., & Xhaja, D. (2022). Support allergic patients, using models found by machine learning algorithms, to improve their quality of life. *International Journal of Intelligent Systems and Applications Engineering, 10*(4), 512–517. https://www.scopus.com/sourceid/21101021990
- Ktona, A., Muça, E., Çollaku, D., Shahini, I., & Boboli, I. (2023). Support the creation of appropriate tourism offers by finding a model, using machine learning algorithms, to forecast spending by tourists. *International Journal of Technology Marketing (IJTMKT), 17*(1), 1–10. https://doi.org/10.1504/IJTMKT.2022.10048142
- Liu, J., Yan, H.-F., Newmaster, S.G., Pei, N., Ragupathy, S., Ge, X.J. & Lowe, A. (2015). The use of DNA barcoding as a tool for the conservation biogeography of subtropical forests in China. *Diversity & distributions*, *21*(2), 188-199.
- McClinton, J.D., Kulpa, S.M., Grames, E.M., & Leger, E.A. (2022). Field observations and remote assessment identify climate change, recreation, invasive species, and livestock as top threats to critically imperiled rare plants in Nevada. *Frontiers in Conservation Science*, Vol.3 https://www.frontiersin.org/journals/conservation-science/articles/10.3389/fcosc.2022.1070490, DOI=10.3389/fcosc.2022.1070490
- Olawade, D.B, Wada, O.Z, Ige, A.O., Egbewole, B.I., Olojo, A., & Oladapo, B. (2024). Artificial Intelligence in Environmental Monitoring: Advancements, Challenges, and Future Directions. *Hygiene and Environmental Health Advances* 12. 100114. 10.1016/j.heha.2024.100114.
- Paszke, A., Gross, S., & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *NIPS*.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, 779-788, doi: 10.1109/CVPR.2016.91.
- Reynolds, S.A., Beery, S., Burgess, N., Burgman, M., Butchart, S.H.M., Cooke, S.J., Coomes, D., Danielsen, F., Di Minin, E., Durán, A.P., Gassert, F., Hinsley, A., Jaffer, S., Jones, J.P.G., Li, B.V., Mac Aodha, O., Madhavapeddy, A., O'Donnell, S.A.L., Oxbury, W.M., Peck, L., Pettorelli, N., Rodríguez, J.P., Shuckburgh, E., Strassburg, B., Yamashita, H., Miao, Z., & Sutherland, W.J. (2025). The potential for AI to revolutionize conservation: A horizon scan. *Trends in Ecology & Evolution, 40*(2), 191–207. https://doi.org/10.1016/j.tree.2024.11.013
- Rivera, Juan. (2020). Practical TensorFlow.js: Deep Learning in Web App Development. 10.1007/978-1-4842-6273-3.

- Smilkov, D., Thorat, N., Assogba, Y., Nicholson, C., Kreeger, N., Yu, P., Zhang, K., Cai, S., Nielsen, E., Soergel, D., Bileschi, S., Terry, M., Nicholson, C., Gupta., S.N., Sirajuddin, S., Sculley, D., Monga, R., Corrado, G., Viégas., F., & Wattenberg, M.M. (2019). Tensorflow. js: Machine learning for the web and beyond. *Proceedings of Machine Learning and Systems*, *1*, 309-321.
- Tabak, M.A, Norouzzadeh, M.S., Wolfson, D.W., Sweeney, S.J., VerCauteren, K.C., Snow, N.P., Halseth, J.M., Di Salvo, P.A., Lewis, J.S., White, M.D., Teton, B., Beasley, J.C., Schlichting, P.E., Boughton, R.K., Wigh, B., Newkirk, E.S., Ivan, J.S., Odell, E.A., Brook, R.K., Lukacs, P.M., Moeller, A.K., Mandeville, E.G., Clune, J., & Miller, R.S. (2018). Machine learning to classify animal species in camera trap images: applications in ecology. *Methods in Ecology and Evolution*, *10*(4), 585-590.
- Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B., & Liang, J. (2016). Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? IEEE Transactions on Medical Imaging, 35, 1299-1312. 10.1109/TMI.2016.2535302.
- Tannous, M., Stefanini, C., & Romano, D. (2023). 'A Deep-Learning-Based Detection Approach for the Identification of Insect Species of Economic Importance'. *Insects* 14(2): 148. https://doi.org/10.3390/insects14020148
- Tran, D.A., Fischer, P., Smajic, A., & So, Y. (2021). Real-time Object Detection for Autonomous Driving using Deep Learning.
- Yang, B., Zhang, Z., Yang, C-Q., Wang, Y., Orr, M.C., Wang, H., & Zhang, A.B. (2022). Identification of Species by Combining Molecular and Morphological Data Using Convolutional Neural Networks, *Systematic Biology*, *71*(3), May, 690–705, https://doi.org/10.1093/sysbio/syab076
- Xhina, E., Bilo, I., Ktona, A., Paparisto, A., Liçi, O., Qirinxhi, X., & Haxhiu, D. (2023). Machine learning methods for understanding biodiversity and its conservation. *International Journal of Ecosystems and Ecology Science (IJEES), 13*(3), 39–44.
- LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., & Jackel, L.D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4):541-551, Winter.
- LeCun, Y., Jackel, L.D., Boser, B., Denker, J.S., Graf, H.P., Guyon, I., Henderson, D., Howard, R.E., & Hubbard, W (1989). Handwritten digit recognition: Applications of neural net chips and automatic learning. *IEEE Communication*, 41-46, November. invited paper.
- LeCun, Y. (1989). Generalization and network design strategies. Technical Report CRG-TR-89-4, Department of Computer Science, University of Toronto.
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A Comprehensive Survey on Transfer Learning. Proceedings of the IEEE. 1-34. 10.1109/JPROC.2020.3004555.