International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-14 Issue-12, November 2025

Comparison of Convolutional Neural Network Architectures for Underwater Image Classification

Krystian Kozakiewicz

Abstract: ***Convolutional neural networks (CNNs) play an essential role in classifying images collected in real-world environments. This article presents a performance comparison of selected CNNs for image classification tasks related to marine flora and fauna, using recordings from an Unmanned underwater vehicle (UUV). An attempt was made to find suitable CNN architectures for processing images of a poor-visibility marine environment among five commonly used architectures. The research was based on a uniform model training system: the same dataset and identical optimisation parameters were used to demonstrate the learning capabilities of each architecture. Thanks to the uniform CNN learning system, their direct learning capabilities for specific images can be more accurately estimated. This means that the conducted experiments showed that, in the early stage of training, the analysed networks achieved similar learning results, whereas the differences concerned the final training accuracy. The best results were achieved with models such as ResNet50, which have the most advanced architecture. Advanced models achieve improved classification of complex and distorted images by leveraging more parameters. The results provide insights into the performance of different architectures in underwater image classification and serve as a reference for further research on deep learning applications in marine environment monitoring.***

Keywords: ***Artificial Intelligence, Deep Learning, Convolutional Neural Networks, Unmanned Underwater Vehicles.***

Nomenclature:
UUV: Unmanned Underwater Vehicle
CNNs: Convolutional Neural Networks

I. INTRODUCTION

In recent years, deep neural networks have become an essential part of visual data analysis, enabling significant advances in tasks such as image classification and object detection. Among them, CNNs have revolutionised the field of computer vision [1]. The development of architectures for transfer learning has enabled the use of efficient models based on ready-made neural network structures. This paper focused on underwater image classification from video frames recorded by a UUV. The aim was to detect elements of marine flora and fauna that are important for both ecological research and environmental monitoring [2].

Manuscript received on 02 November 2025 | Revised Manuscript received on 09 November 2025 | Manuscript Accepted on 15 November 2025 | Manuscript published on 30 November 2025.
*Correspondence Author(s)
Krystian Kozakiewicz*, Department of Autonomous Systems, Gdynia Maritime University, Gdynia (Pomorskie), Poland. Email ID: k.kozakiewicz@wi.umg.edu.pl, ORCID ID: 0009-0009-3020-5764

Such images pose challenges: limited water transparency, variable lighting conditions, and a wide variety of biological objects [3]. To develop the most effective approach in this context, five commonly used CNN models were compared: MobileNetV2, Resnet50, EfficientNetB0, InceptionV3, and DenseNet121. This paper used only one training system: all models were trained on the same dataset and with identical optimisation parameters for the same number of epochs, allowing comparison of training accuracy across the chosen architectures. The main factor studied in this work is the network's ability to learn from low-visibility underwater images.

II. THEORETICAL BACKGROUND

The functioning of biological nervous systems inspires the neural networks used in machine learning. With individual computational "neurons" in layers, a neural network allows establishing relationships between input data and expected results by assigning weights to selected neurons. Over time, deep neural networks started to be designed. Their multilayer structure allows the creation of very complex patterns. One of the essential types of architecture is the CNN, which has become the standard for analysing images. A typical CNN includes the following basic components: convolutional layers, activation layers, and pooling layers. Convolution layers extract local image features using filters. Non-linear activation layers allow the network to approximate complex functions; common types include ReLU. Pooling layers reduce the dimensionality of feature maps. Fully connected layers perform the final classification based on the high-level features learned in previous stages.

Architecture built this way allowed CNNs to detect simple patterns such as edges and textures. Simultaneously, this structure enabled the classification of more abstract structures, including animals and plants. As a result, CNNs have found wide application in medicine, satellite image analysis, environmental monitoring, and autonomous systems. Many improved variants of CNNs have been proposed in the literature, differing in their information flow, depth, and computational efficiency. Among them, some are widely used. MobileNetV2 is optimised for mobile devices and resource-constrained devices [4]. ResNet50 enables training deep networks without vanishing gradients [5]. EfficientNetB0 will balance width and depth, offering a high accuracy-to-complexity ratio [6]. InceptionV3 enables efficient feature extraction across different scales via multiscale filters [7]. DenseNet-12, with dense connectivity in which each layer receives Input from all previous layers enables improved feature reuse [8]. Comparing these

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

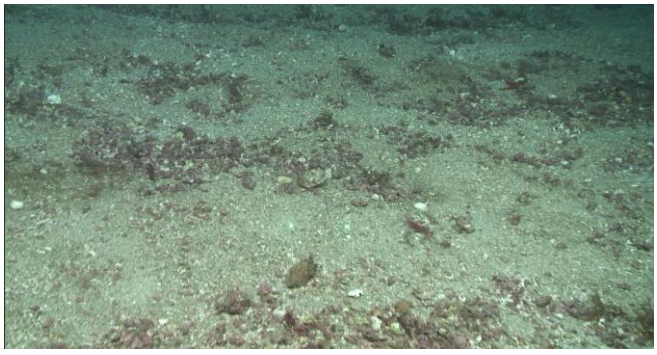
five architectures provides a broad perspective for evaluating learning performance. Most comparative studies have focused on datasets such as ImageNet, but this work extends the line by assessing their performance in classifying underwater images under challenging visual conditions.

III. METHODOLOGY

All stages of the experiments were performed using the Python programming language, with libraries including TensorFlow, Keras, NumPy, Pandas, and matplotlib. TensorFlow and Keras were used to build models and perform transfer learning. NumPy and Pandas served as utilities for working with the dataset. Matplotlib was used for visualisation using graphs. To make the results repeatable, random seeds of models were fixed.

The ICM Benchmark used in this study is publicly available on the Kaggle platform [9]. It includes video frames recorded by a UUV showing various species of marine flora and fauna. Frames from the identical sequences were assigned to selected classes, allowing the division of the dataset into species of aquatic organisms: *Spatangus purpureus* (purple heart urchin), *Echinaster spesitus* (Mediterranean red sea star), *Cerianthus membranaceus* (cylinder anemone), *Bonellia virdis* (green spoonworm), *Scyliorhinus canicula* (small-spotted catshark), and *Ophiura ophiura* (serpent star). To provide a control sample, a class for only seabed fragments, with no evidence of any species present, was also created. This class is crucial in classification to identify objects of biological interest and the environmental background. The complete set of species for this dataset is shown in Figure 1. In addition, Figures 2 and 3 compare an image containing only the seabed background with a photo showing a selected species. These demonstrate the difficulty of classification, even for the human eye.

[Fig.1: ICM Benchmark Species]



[Fig.2. Example of an Image Showing Only the Seabed]

[Fig.3: Example of an Image Showing Echinaster Spesitus]

Preprocessing included resizing input images to the required resolution of each architecture (224 x 224 pixels for MobileNetV2, ResNet50, EfficientNetB0, and DenseNet121, and 299 x 299 pixels for InceptionV3). Each model employed the appropriate normalization function provided in TensorFlow. To enhance generalization capability, data augmentation was applied during training [10]. This augmentation involved random rotations, horizontal flips, cropping, and adjustments to brightness and contrast. The transformations were designed to reflect the variability of underwater environments, particularly changes in illumination and water turbidity. All five CNN architectures were trained ten times each under identical conditions, with the same optimization settings and on the same dataset. The classification head was adapted to the number of classes, consisting of a global average pooling layer, a dropout layer to mitigate overfitting, and a dense layer with SoftMax activation. Each model was trained for 20 epochs, without early stopping. The optimization process used the Adam optimizer, and the loss function was categorical cross-entropy suitable for multi-class classification.

To minimise the influence of stochastic factors, each experiment was repeated multiple times with different random initialisations of weight and data batching. Performance analysis was based on the mean and standard deviation of model accuracies. In addition, learning curves were examined across epochs to assess training dynamics. The primary focus was on training accuracy to point out the best CNN architecture for training a model on this dataset. Results are presented in both tabular and graphical form in the next part.

IV. RESULTS

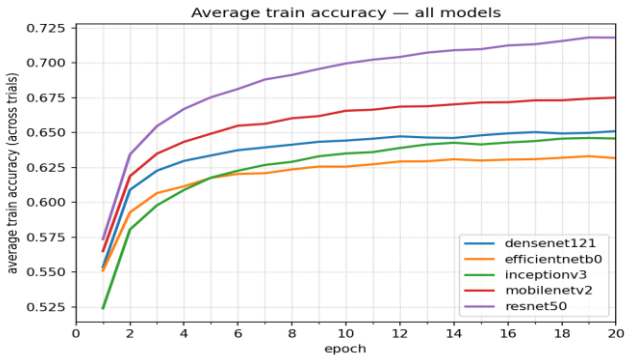
This section presents results from experiments on the learning accuracy of five CNN architectures. The average training accuracy per epoch for each model is presented in Table I. Figure 4 shows the average results across models. This graph enables a direct comparison of the five architectures in terms of their learning effects. Figures 5-9, however, represent the individual average results of the architectures along with them exact curves for each of the 10 trials.

Retrieval Number: 100.1/ijitee.A117515011225
DOI: 10.35940/ijitee.A1175.14121125
Journal Website: www.ijitee.org

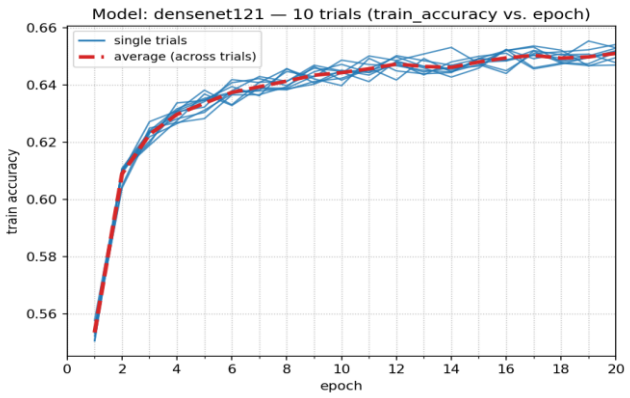
Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Table I: Model's Architectures Train Accuracy Per Epoch

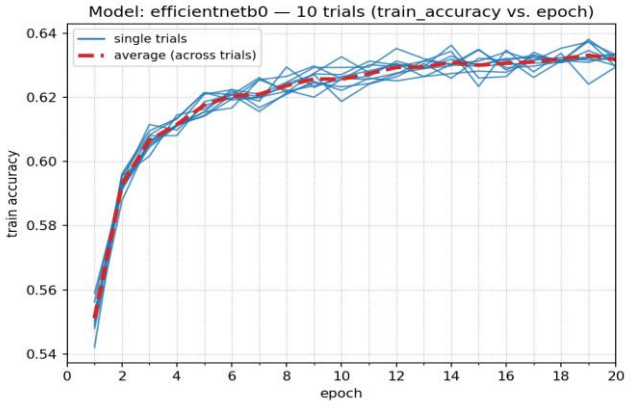
Epoch	DenseNet121	EfficientNetB0	Inceptionv3	MobileNetv2	ResNet50
1	0.5536	0.5511	0.5241	0.5650	0.5736
2	0.6090	0.5929	0.5806	0.6188	0.6343
3	0.6227	0.6067	0.5979	0.6349	0.6546
4	0.6297	0.6115	0.6089	0.6434	0.6669
5	0.6336	0.6176	0.6177	0.6492	0.6754
6	0.6374	0.6204	0.6227	0.6549	0.6813
7	0.6394	0.6209	0.6269	0.6563	0.6881
8	0.6414	0.6236	0.6291	0.6603	0.6913
9	0.6434	0.6257	0.6330	0.6618	0.6956
10	0.6443	0.6256	0.6351	0.6657	0.6995
11	0.6457	0.6273	0.6360	0.6664	0.7023
12	0.6473	0.6293	0.6390	0.6686	0.7042
13	0.6465	0.6294	0.6415	0.6689	0.7073
14	0.6462	0.6309	0.6427	0.6702	0.7090
15	0.6481	0.6300	0.6415	0.6716	0.7098
16	0.6495	0.6307	0.6429	0.6718	0.7125
17	0.6503	0.6310	0.6439	0.6731	0.7133
18	0.6494	0.6320	0.6457	0.6731	0.7156
19	0.6498	0.6331	0.6462	0.6744	0.7182
20	0.6511	0.6317	0.6458	0.6751	0.7181



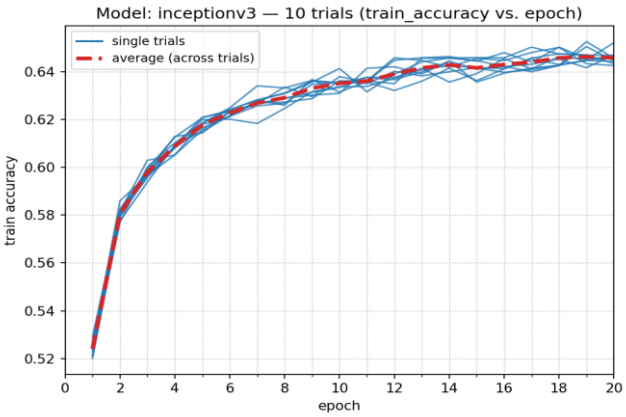
[Fig.4: Average Train Accuracy of all Architectures]



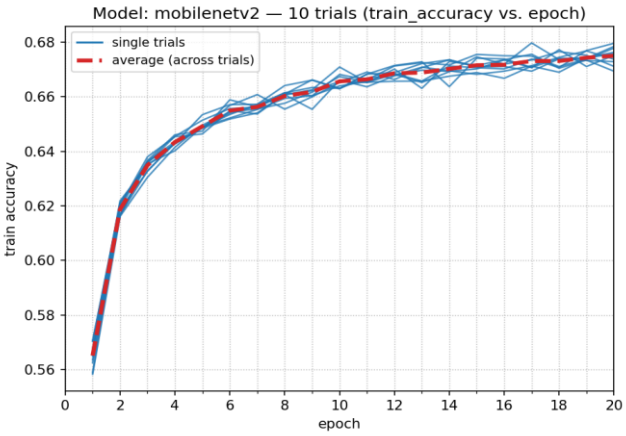
[Fig.5: Train Accuracy of DenseNet121]



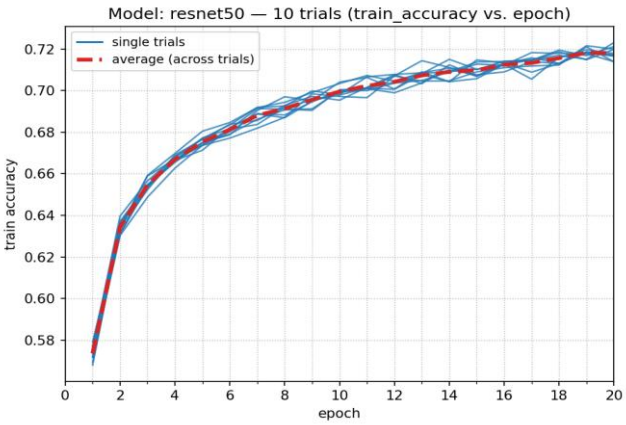
[Fig.6: Train Accuracy of EfficientNetB0]



[Fig.7: Train Accuracy of Inceptionv3]



[Fig.8: Train Accuracy of MobileNetv2]



[Fig.9: Train Accuracy of ResNet50]

The initial training accuracy values range from 0.524 to 0.574, indicating that no architecture has a clear advantage in the very beginning of learning. Among all models, InceptionV3 achieved the lowest initial value. DenseNet121, EfficientNetB0, and MobileNetV2 achieved average initial values. ResNet50 achieved the highest initial training accuracy. As training progressed, all networks gradually improved their accuracy. After 20 epochs, training accuracies ranged from 0.632 to 0.718. At this stage, differences between architectures became more pronounced. EfficientNetB0 achieved the lowest final training accuracy, while InceptionV3 was somewhat better. DenseNet121 and MobileNetV2 got

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

intermediate results. ResNet50 considerably outperformed previous models, achieving the best outcome. It attained a difference of 0.033, indicating 3% better training accuracy than the second-best result for MobileNetV2.

In summary, though all five architectures showed very similar initial results, the final results show significant differences in model training effectiveness. ResNet50 proved to be the most effective model under the experimental conditions, achieving the highest average accuracy, suggesting that its architecture is best suited for training models on difficult images of the underwater marine environment.

V. CONCLUSION

This paper compares the performance of selected CNN architectures. The objective was to assess how effectively the models can be trained to classify underwater images from frames extracted from UUV recordings. Five models representative of different design approaches have been considered; all were trained on the same dataset and with the same training parameters. The methodology followed allowed direct comparison of model performance, excluding variability introduced by different training times or hyperparameter settings. The results showed that all architectures exhibited very similar learning curves and training accuracies during the first epochs. The differences became more pronounced only in the later stages of training, where the networks reached different final values of training accuracy [11]. These results also show that models with higher structural complexity outperformed simpler ones, suggesting they are better at recognising complex biological structures. This analysis provides a basis for subsequent investigations into the potential of deep learning for marine environment monitoring. Future work could expand this study to include newer network architectures or appropriate hyperparameter selections for the chosen architecture. Besides, observing validation and test accuracy would be advantageous for comparing various architectures under real conditions.

DECLARATION STATEMENT

I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** Yes, I have received financial assistance for this article. Uniwersytet Morski w Gdyni NIP: 586-001-28-73 ul. Morska 81-87 81-225 Gdynia. https://umg.edu.pl/kontakt
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed solely.

REFERENCES

1. Y. LeCun, Y. Bengio, G. Hinton (2015). "Deep learning". Nature 521, 436–444 (2015). DOI: https://doi.org/10.1038/nature14539
2. M. J. Islam, S. Sakib Enan, P. Luo, and J. Sattar, "Underwater Image Super-Resolution using Deep Residual Multipliers," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 900-906, DOI: https://doi.org/10.1109/ICRA40945.2020.9197213.
3. C. Li et al., "An Underwater Image Enhancement Benchmark Dataset and Beyond," in IEEE Transactions on Image Processing, vol. 29, pp. 4376-4389, 2020, DOI: https://doi.org/10.1109/TIP.2019.2955241.
4. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. Chen. "MobileNetV2: Inverted Residuals and Linear Bottlenecks" Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520. DOI: https://doi.org/10.48550/arXiv.1801.04381
5. M. Elpeltagy, H. Sallam, "Automatic prediction of COVID-19 from chest images using modified ResNet50". Multimed Tools Appl 80, 26451–26463 (2021). DOI: https://doi.org/10.1007/s11042-021-10783-6
6. S. Upadhyay, J. Jain, R. Prasad. (2024). "Early Blight and Late Blight Disease Detection in Potato Using Efficientnetb0". International Journal of Experimental Research and Review, 38, 15–25. DOI: https://doi.org/10.52756/ijerr.2024.v38.002
7. C. Wang et al., "Pulmonary Image Classification Based on Inception-v3 Transfer Learning Model," in IEEE Access, vol. 7, pp. 146533-146541, 2019, DOI: https://doi.org/10.1109/ACCESS.2019.2946000.
8. G. Huang, Z. Liu, L. van der Maaten, K. Weinberger; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4700-4708. DOI: https://doi.org/10.48550/arXiv.1608.06993
9. D. Serrano, "ICM-Benchmark-20," Kaggle Datasets, Oct. 2024. [Online]. Available: https://www.kaggle.com/datasets/tsunamiserra/icm-benchmark-20 [Accessed: 31-Oct-2025].
10. Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. J Big Data 6, 60 (2019). DOI: https://doi.org/10.1186/s40537-019-0197-0
11. J. Madewell, R.A. Feagin, T.P. Huff, B. Balboa; Estimating Freshwater Inflows for an Ungauged Watershed at the Big Boggy National Wildlife Refuge, USA. J. Mar. Sci. Eng. 2024, 12, 15. DOI: https://doi.org/10.3390/jmse12010015

AUTHOR'S PROFILE

Krystian Kozakiewicz completed his Master's degree at the Faculty of Electrical Engineering at the Maritime University of Gdynia, specialising in Automation, Electronics, Electrical Engineering, and Space Technologies. His research interests focus on artificial intelligence and automation, and autonomous systems. He is actively involved in the development and experimental evaluation of AI models designed for processing both numerical data and image data. His work includes implementing and optimising neural networks for tasks such as pattern recognition, anomaly detection, and sensor data fusion in real-time environments. He works with autonomous and semi-autonomous mobile robots for navigation, control, and task execution.