A web-based application to classify cendrawasih birds using deep learning

Ardiyanto[1,a)], Yuliska[2,b)]

[1]*Magister Terapan Teknik Komputer, Politeknik Caltex Riau, Pekanbaru, Indonesia*
[2]*Teknik Informatika, Politeknik Caltex Riau, Pekanbaru, Indonesia*

[a)]Corresponding author: ardiyanto23mttk@mahasiswa.pcr.ac.id
[b)]yuliska@pcr.ac.id

Abstract. This research aims to develop a deep learning-based model to classify images of birds of paradise (Cendrawasih). Three different model architectures were employed in this study: Convolutional Neural Network (CNN), InceptionResNetV2, and MobileNetV2. The dataset consists of several species of birds of paradise, which were processed using data augmentation techniques to enhance the variety and quality of the training data. The model training and evaluation processes were conducted using TensorFlow and Keras, with the application of callbacks such as EarlyStopping to prevent overfitting. Evaluation results indicate that the MobileNetV2 and InceptionResNetV2 models achieved the highest accuracy, with an average score above 90%. The system implementation involved developing a web application based on Flask and React JS to facilitate real-time image prediction. This study demonstrates the effectiveness of using deep learning models for bird of paradise image classification and their potential for application in limited computing environments.

Keywords: deep learning, image classification, Convolutional Neural Network (CNN), InceptionResNetV2, MobileNetV2, data augmentation, TensorFlow.

INTRODUCTION

The Bird-of-Paradise (Cendrawasih) species, known for their striking visual features, are of critical importance to biodiversity and conservation efforts in Papua. Identifying and classifying these birds through traditional means can be labor-intensive and prone to human error. In this study, we leverage deep learning models to automate the classification of Cendrawasih species using image data. We explored multiple architectures, including Convolutional Neural Networks (CNN), InceptionResNetV2, and MobileNetV2, with MobileNetV2 achieving the highest accuracy of 90.62%.

This model was selected for its balance between computational efficiency and classification performance, making it ideal for real-time applications. Furthermore, the model was deployed in a web-based system using Flask and React.js, enabling users to upload bird images for classification via a user-friendly interface. This system enhances the accessibility and scalability of bird identification, aiding conservation efforts and species monitoring. In this paper, we present the development pipeline, from data preprocessing and augmentation techniques to model evaluation, and discuss the potential impact of automated classification in conservation biology.

METHODS

In this research will go through several processes, the following is a flowchart of the research path process.

Figure 1 Reseach Path

1. Problem Definition

Birds of paradise (Cendrawasih) are unique birds with several species that are difficult for the general public to distinguish visually. To help address this issue, an automatic classification system is needed, capable of recognizing bird species from images quickly and accurately. Manual methods for identifying bird species are not only time-consuming but also prone to errors.The main task of this project is to develop a web-based application that can automatically classify different species of birds of paradise using deep learning methods. This application is designed for users to upload bird images, and the system will identify the species of the bird

2. Data Collection

The data used for this project is the bird of paradise (Cendrawasih) dataset, which was manually collected from Google. This dataset contains 6 labels, which consist of cendrawasih belah rotan, cendrawasih botak, cendrawasih kerah, cendrawasih goldi, cendrawasih merah, and cendrawasih raja. The detailed number of images collected for each label can be seen in the following table.

Table 1 Data Collection

No	Data	Image	Total
1	Cendrawasih Belah Rotan		203
2	Cendrawasih Botak		234

			
3	Cendrawasih Goldi		259
4	Cendrawasih Kerah		222
5	Cendrawasih Merah		237

6	Cendrawasih Raja		223

3. Modelling

Modelling is done using 3 algorithms namely:

1) Convolutional Neural Network (CNN)
2) InceptionResNetV2
3) MobileNetV2

The exploration of deep learning models was conducted using three different architectures: CNN, InceptionResNetV2, and MobileNetV2, to classify birds of paradise (Cendrawasih). Each model was tested and evaluated based on accuracy and loss performance on training and testing data. CNN was used as the baseline model, while InceptionResNetV2 offers multi-scale feature extraction and residual connections for deeper networks.

3.1 Convolutional Neural Network (CNN)

CNN works by utilizing a series of convolutional layers to extract features from images. CNNs are specifically designed for image data processing. The main components of CNN include:

1) Convolutional Layer: Uses filters to extract local features from images, such as edge patterns or shapes. The filter moves across the image and produces a feature map that highlights specific patterns.
2) Pooling Layer: Typically used after the convolutional layer to reduce data dimensions and retain the most important features. Max pooling is often used to select the maximum value within a specific area.
3) Fully Connected Layer: In this layer, all neurons are connected to each other, and the resulting feature maps are converted into classification decisions.
4) Activation Function (ReLU): The ReLU function adds non-linearity to the network to capture complex relationships between features.

Here is the model architecture of CNN.

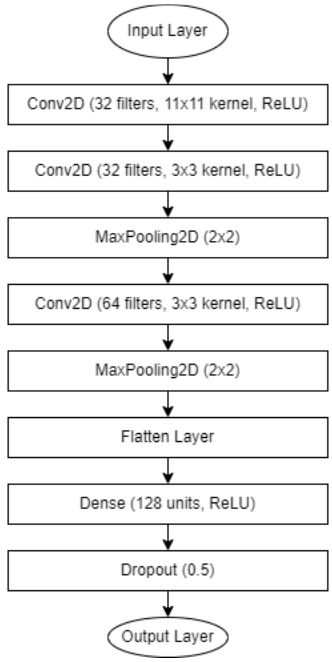

Figure 2 Architecture CNN

The CNN model starts with an input image processed through two consecutive convolutional layers to extract features, followed by a MaxPooling layer to reduce the feature size. Then, two convolutional layers with 64 filters are used to extract deeper features, and the size is again reduced by MaxPooling. The result is flattened through the Flatten Layer and passed to a Dense Layer with 128 units for further processing. Dropout is applied to prevent overfitting, and finally, the Output Layer with SoftMax activation produces the classification predictions.

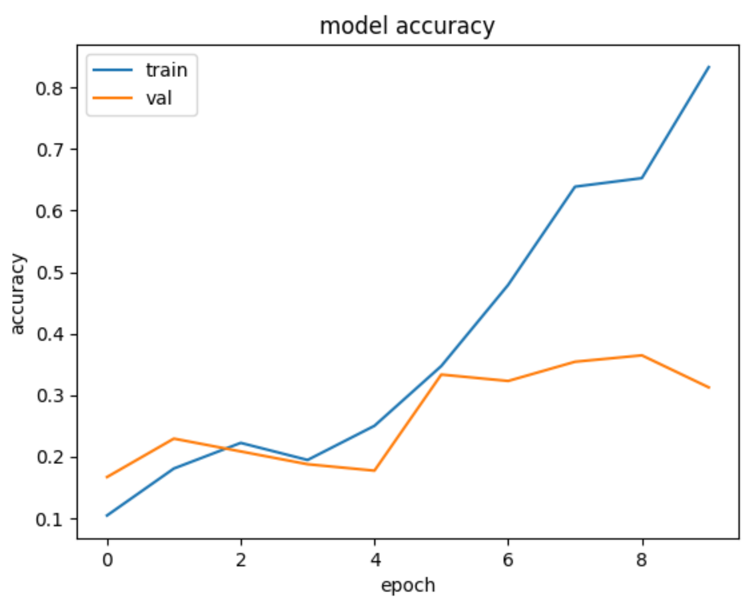

Figure 3 Model Accuracy CNN

The results indicate that the CNN did not perform optimally on the Cendrawasih bird dataset, with relatively low accuracy and possible overfitting due to a lack of variation or complexity in the data or model architecture.

3.2 InceptionResNetV2

InceptionResNetV2 combines the Inception and ResNet architectures:

1) Inception Modules: These modules use multiple filters of different sizes (1x1, 3x3, and 5x5) within a single layer, allowing the model to capture features at various scales of the image. This enhances the model's ability to recognize more complex patterns.
2) Residual Connections (ResNet): Residual connections allow information to bypass several layers, helping to address the vanishing gradient problem in deep networks. This enables the model to learn more efficiently without losing important information during training.

Here is the model architecture of InceptionResNetV2.

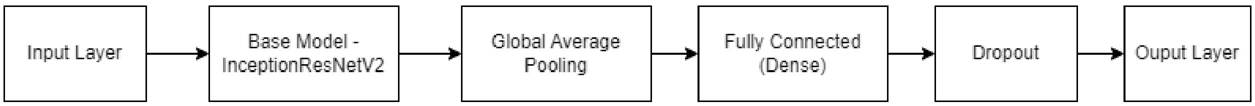

Figure 4 Architecture InceptionResNetV2

The model architecture starts with an input image processed by InceptionResNetV2 as a feature extractor. Then, the extracted features are summarized using Global Average Pooling. After that, the features are processed by a Fully Connected Layer with 128 units, followed by Dropout to prevent overfitting. Finally, the Output Layer with softmax activation generates classification predictions based on the number of bird-of-paradise species classes.

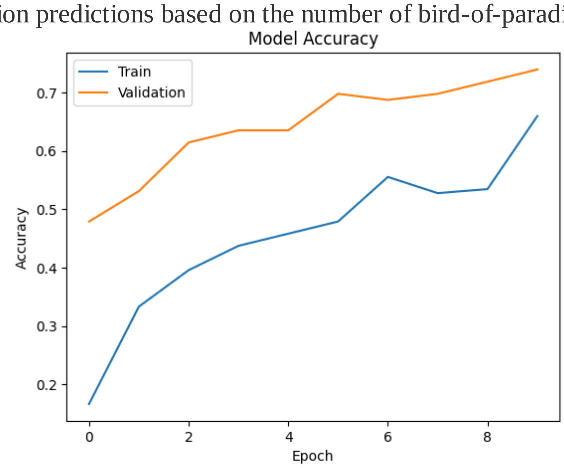

Figure 5 Model Accuracy InceptionResNetV2

The InceptionResNetV2 shows a significant improvement in accuracy compared to CNN. The combination of its ability to capture features at various scales and residual connections gives this model an advantage in detecting complex patterns in the Bird of Paradise dataset.

3.3 MobileNetV2

MobileNetV2 is designed for computational efficiency, particularly for devices with limited resources. Its main features include:

1) Depthwise Separable Convolution: Instead of using standard convolution, MobileNetV2 splits the convolution process into two stages: depthwise convolution (performed on each input channel separately) and pointwise convolution (1x1 convolution to combine the depthwise results). This significantly reduces computational load and model size.
2) Inverted Residuals with Linear Bottleneck: MobileNetV2 introduces residual connections with a narrower bottleneck layer to enhance efficiency while maintaining good accuracy. These inverted residual blocks ensure a more compact representation without sacrificing the network's ability to capture information.

Here is the model architecture of MobileNetV2.

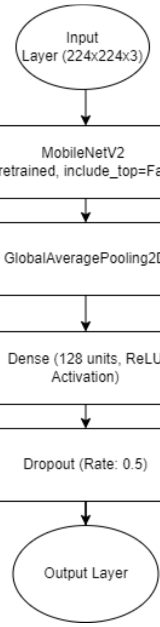

Figure 6 Architecture MobileNetV2

This model uses MobileNetV2 without the top layers, with pre-trained weights from ImageNet, where the top classification layers are removed and the input size is 224x224x3. The initial layers are frozen to retain the pre-trained weights during transfer learning. Additional layers include global pooling, a fully connected layer with 128 neurons and ReLU activation, a 50% dropout to prevent overfitting, and an output layer with SoftMax activation, where the number of neurons matches the number of classes. Finally, the model is combined with the input from MobileNetV2 and the newly added classification output layer. Here is the accuracy graph of the MobileNetV2 model applied in this study.

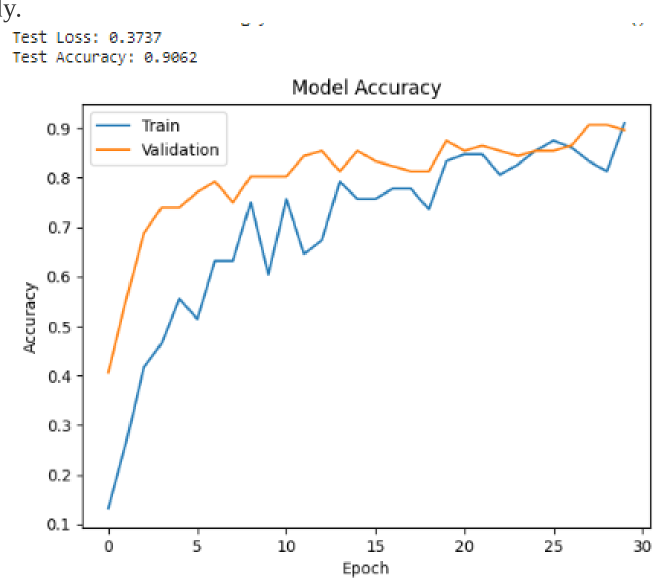

Figure 7 Model Accuracy MobileNetV2

.

4. Model Evaluation

From the testing results conducted on several methods used, the following outcomes were obtained:

1) CNN: Accuracy of 31.25%, indicating that this model is less optimal for this dataset.

2) InceptionResNetV2: Accuracy of 73.96%, showing a significant improvement in accuracy due to a deeper network and the use of residual connections.
3) MobileNetV2: Highest accuracy of 90.62%, proving that this method is highly effective for image classification with high computational efficiency.

Table 2 Model Evaluation

No	Metode	Loss	Accurasy
1	CNN	1.9794	0.3125
2	InceptionRestNetV2	0.7809	0.7396
3	MobileNetV2	0.3737	0.9062

Based on the tests conducted and the evaluation results, it was determined that the most suitable method to use is MobileNetV2, with an accuracy of 0.9062 or 90.62%.

5. Web Deployyment

Web Deployment involves integrating both the backend and frontend of the Cendrawasih bird image classification application. The backend, developed using Python and Flask, handles web requests and image prediction with TensorFlow and MobileNetV2, while tools like Ngrok allow public access via a URL, and Keras Preprocessing and Werkzeug manage image processing and file security. The frontend, built with React.js, uses Axios or Fetch API for backend communication, with HTML/CSS, JavaScript, and frameworks like Bootstrap enhancing responsiveness. This integration enables efficient user interaction and seamless image prediction processing. Here is an overview of the resulting website.

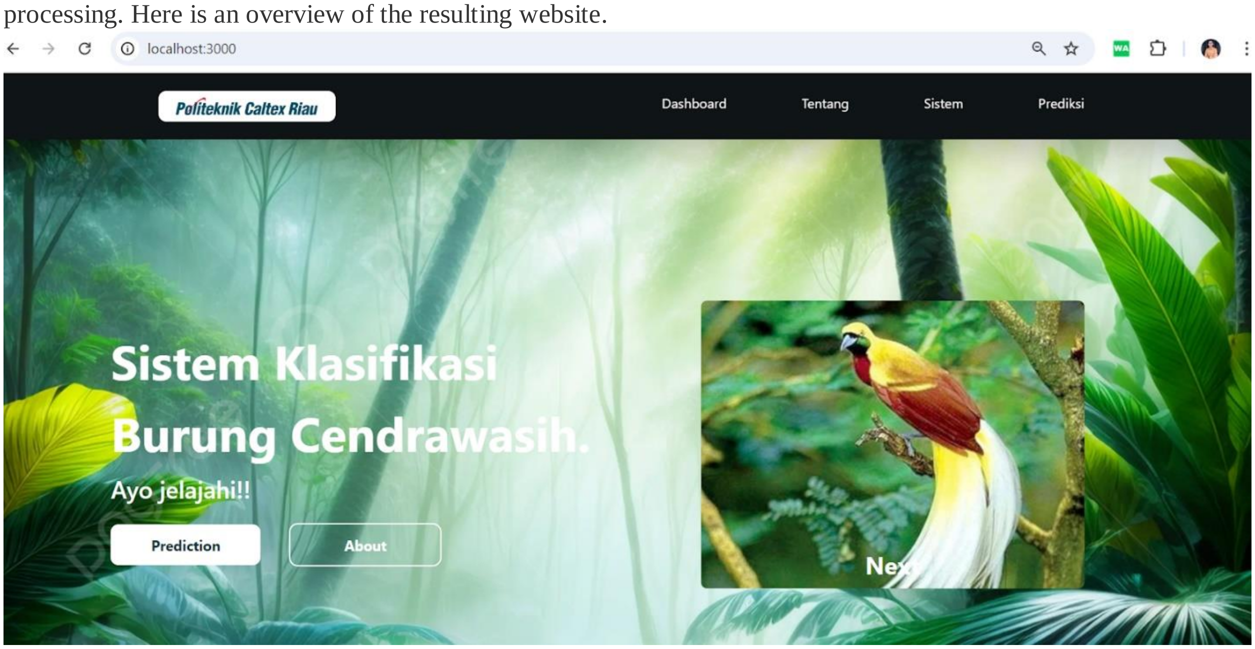

Figure 8 Home Website

RESULTS AND DISCUSSION

In this study, the classification of Cendrawasih bird images was performed using three deep learning models: Convolutional Neural Network (CNN), InceptionResNetV2, and MobileNetV2. The models were evaluated based on accuracy and loss metrics. The results of the model evaluation are as follows:

1) CNN Model:

Accuracy: 31.25%
Loss: 1.9794
2) InceptionResNetV2 Model:
Accuracy: 73.96%\
Loss: 0.7809
3) MobileNetV2 Model:
Accuracy: 90.62%
Loss: 0.3737

From these results, MobileNetV2 outperformed the other models, achieving the highest accuracy and lowest loss. Therefore, MobileNetV2 was chosen as the best model for the bird classification task in this web-based application.

The CNN model, while being a simpler architecture, did not perform well in this classification task. This result can be attributed to its relatively shallow structure, which limits its ability to capture complex features from the bird images. As a result, it yielded the lowest accuracy (31.25%) and the highest loss (1.9794), indicating poor generalization to new data. The InceptionResNetV2 model showed significant improvement over the CNN model, achieving an accuracy of 73.96%. This improvement is due to the combination of the Inception module, which allows multi-scale feature extraction, and ResNet's residual connections, which prevent the vanishing gradient problem in deep networks. The model's ability to capture detailed features at different scales contributed to its better performance.

However, the MobileNetV2 model outperformed both CNN and InceptionResNetV2, achieving the highest accuracy (90.62%) and the lowest loss (0.3737). MobileNetV2's architecture is particularly suited for this task, as it uses depth wise separable convolutions and inverted residual blocks with linear bottlenecks. These features allow the model to be computationally efficient while maintaining high accuracy, making it ideal for mobile and low-power devices. The success of MobileNetV2 in this study demonstrates its capacity to generalize well across different bird species, even with limited computational resources. The choice of MobileNetV2 as the final model for deployment in this web-based application aligns with the need for an efficient, high-performing model suitable for mobile or embedded systems, as it balances both accuracy and computational cost. In conclusion, while all models provided valuable insights, MobileNetV2 was chosen for its superior performance in accuracy, making it the best fit for the bird classification application. Future work may explore further optimization or fine-tuning of this model to improve performance even further.

After the model is obtained, the next step is to use the model in the backend, which will later be implemented on the website. The link generated from the backend will be used in the frontend. Once the React.js program is run and the accuracy is calculated, a page like the following will appear

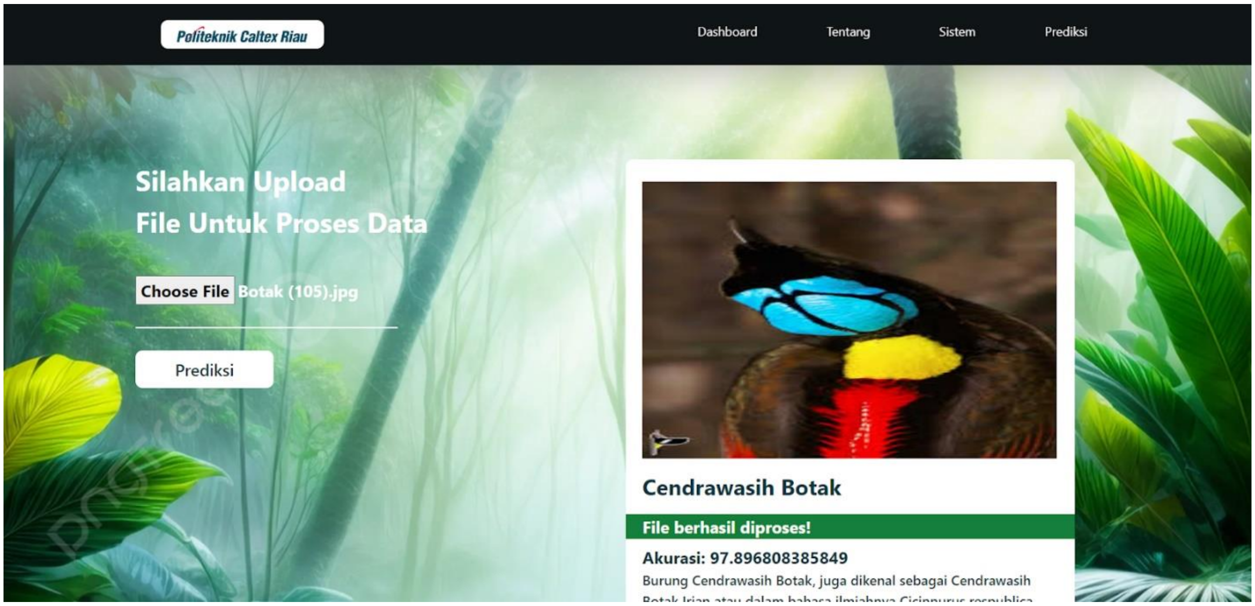

Figure 9 Web Prediction

CONCLUSIONS

The use of deep learning techniques in classifying Cendrawasih bird images provides varying results depending on the model architecture used. Among the three models tested (CNN, InceptionResNetV2, and MobileNetV2), MobileNetV2 demonstrated the best performance with an accuracy of 90.62% and a lower loss value compared to the other models. This makes MobileNetV2 the best choice for implementation in the developed web-based application.

Additionally, the application utilizes a Python and Flask-based backend to process images and predict bird species using the deep learning model. The application's interface is built with React.js, connected to the backend, allowing users to upload images and receive prediction results interactively.

With the lighter and more efficient MobileNetV2 architecture, this application is suitable for use on devices with limited computational power, such as mobile phones or embedded devices, without sacrificing accuracy. The well-integrated combination of frontend and backend ensures a seamless user experience in automatically classifying Cendrawasih bird images.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to Politeknik Caltex Riau, particularly to the Machine Learning course instructors for their invaluable guidance and insights throughout this project. Lastly, my heartfelt appreciation goes to my family and friends for their continuous support during the completion of this project.

REFERENCES

[1] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. https://doi.org/10.1038/nature14539

[2] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1). https://doi.org/10.48550/arXiv.1602.07261

[3] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4510-4520). https://doi.org/10.1109/CVPR.2018.00474

[4] Albahli, S., Albattah, W., Masood, M., & Mohammed, M. (2020). Web-based application for deep learning in image classification using convolutional neural networks. *International Journal of Interactive Multimedia and Artificial Intelligence*, 6(5), 47-55. https://doi.org/10.9781/ijimai.2020.10.003

[5] Li, X., Liu, Y., Zhang, H., & Zhang, J. (2021). Web-based platform for real-time image classification using deep learning techniques. Journal of Systems and Software, 177, 110994. https://doi.org/10.1016/j.jss.2021.110994

[6] Bhatt, D., Patel, S., Talsania, M., & Joshi, P. (2020). Deploying deep learning models as web applications: A survey and implementation. Future Generation Computer Systems, 109, 209-223. https://doi.org/10.1016/j.future.2020.06.005

[7] Rajpurkar, P., Irvin, J., Ball, R. L., Zhu, K., Yang, B., Mehta, H., & Lungren, M. P. (2019). Deep learning in web-based medical image classification: A case study of chest X-rays. IEEE Access, 7, 88835-88845. https://doi.org/10.1109/ACCESS.2019.2909960

[8] Gupta, R., Singh, D., Sharma, P., & Sood, M. (2021). Design and implementation of a web-based application for image classification using deep learning. Procedia Computer Science, 179, 92-100. https://doi.org/10.1016/j.procs.2021.02.090

[9] Lee, J., Lee, S., & Park, H. (2021). A scalable web-based platform for deploying deep learning models: Application to image classification and object detection. Journal of Visual Communication and Image Representation, 78, 103123. https://doi.org/10.1016/j.jvcir.2021.103123

[10] Khan, M. A., Sharif, M., Raza, M., Damaševičius, R., & Balas, V. E. (2019). A web-based tool for automatic classification of skin lesions using deep learning. Pattern Recognition Letters, 134, 70-76. https://doi.org/10.1016/j.patrec.2019.04.027