Volume 5, Issue VI, June, 2026; No. 88, pp. 1121-1136
Submitted 22/4/2025; Final peer review 30/5/2026
Online Publication 3/6/2026
Available Online at http://www.ijortacs.com

DESIGN AND IMPLEMENTATION OF A DEEP LEARNING-BASED IMAGE CLASSIFICATION SYSTEM USING CONVOLUTIONAL NEURAL NETWORKS

Edith Angela Ugwu

Department of Computer Science, Enugu State University of Science and Technology

edith.ugwu@esut.edu.ng

Abstract

The paper is a design and a realization of a deep learning-driven image classification system with the use of Convolutional Neural Networks (CNNs). The goal of the paper was to create a smart model that can properly classify images into the predefined classes by means of the automated feature extraction and learning. The system was created using the Agile approach and functionality, which meant that it was created in an iterative process where the subsequent sprints could be used to continuously improve the system by preparing datasets, training the model, testing, and evaluating the total output. The dataset was a collection of four categories which includes Cat, Dog, Bird, and Fish; the publicly available sources were chosen to use Kaggle and ImageNet. The images were pre-trained by resizing, normalization and augmentation methods such as rotation, flipping, scaling and adjustment in brightness to enhance generalization and reduce overfitting. The system implementation results showed high performance in all the categories, with the highest scores being in the Cat class (Precision: 0.94, Recall: 0.95, F1-Score: 0.945, Accuracy: 0.96), then the dogs, birds and fish respectively. The trained model was directly converted into a web application written in Flask and allowed one to upload and classify images in real-time, which confirmed the practicality and usability of the system. In general, the designed CNN-based image classification system was precise, effective, and scalable, which fulfilled its goal of a reliable image recognition in various categories. The study showcases the opportunities of deep learning as a field of computer vision and how CNNs can be successfully implemented in practice in fields like healthcare, agriculture, and security. It will be followed by future work that considers the use of transfer learning based on more advanced architectures and optimized to support mobile and edge deployment in order to increase its performance and flexibility further.

Keywords: CNN; Image Classification; Deep Learning; Data Augmentation; Machine Learning

1. INTRODUCTION

Image classification is one of the key activities of computer vision, and the aim is to label the input images with pre-defined terms. The recent years saw the field undergo a drastic change, as there is the implementation of deep learning, specifically, Convolutional Neural Networks (CNNs), which prove to be more accurate and scalable than the traditional machine learning methods (Islam et al., 2020). The CNNs can automatically discover hierarchical features on raw pixel data which means that there is less required feature engineering and the ability to generalize across datasets (Zhao et al., 2019).

Current image classification systems are heavily dependent on an annotated large-scale datasets and effective architectures. Efficient and deeper architectures such as EfficientNet, DenseNet, and Vision Transformer have been developed due to access to benchmark datasets like CIFAR-100, ImageNet, and Tiny ImageNet (Dosovitskiy et al., 2020). These architectures have continued to enhance the performance of classification by adding new design concepts like depth wise convolutions, attention, and parameter scaling strategies. CNN-based image classification has been useful in the healthcare sector especially. Research has revealed good findings in the diagnosis of diseases like pneumonia, tuberculosis, and skin cancer through chest X-rays, CT scans, and dermoscopic images (Kassani et al., 2021). These AI-based systems can help medical workers in diagnosis and early disease diagnosis, particularly in resource-scarce environments.

Equally, image classification is applied in agriculture to track the health of crops, identify plant diseases, and weed species to implement precision farming (Kamilaris and Prenafeta-Boldú, 2021; Viswanatha et al., 2022). These applications enhance efficiency and minimize the work of field inspection. Image classification systems are also used in the environmental monitoring, wildlife conservation, and smart farming systems to manage resource utilization and sustainability (Dosovitskiy et al., 2020).

Deep learning-based image classification models are only successful when it comes to architecture, but also, effective data preprocessing and augmentation methods (Ajit et al., 2020). The methods of rotation, flipping, brightness control, and cutout augmentation contribute to decreasing overfitting and enhancing the resistance of the model (Zhang et al., 2021). Additionally, it has been powered by model optimization tools and lightweight architectures to be used on mobile and edge devices to support real-time applications, including traffic management, security surveillance, and industrial automation (Afroze et al., 2022; Tao et al., 2020). Regardless of these improvements, there are still difficulties. Image classification models are still susceptible to adversarial attacks, data imbalance and domain shifts that can bring down the performance of the model in the real world. Also, ethical issues of fairness, bias, and transparency in automated decision-making systems become more significant, specifically in high-stakes systems like facial recognition and medical diagnosis (Kornblith et al., 2021; Suresh et al., 2021). These issues are essential to consider to make AI technologies use responsible.

This research will develop and train a model of image classification using deep learning to classify images by a set of established categories with high accuracy rates. The suggested system takes advantage of CNN based architectures and preprocessing methods to enhance the performance and generalizability of the classification. This study can assist in addressing the current challenges of creating intelligent systems to aid decision making in various sectors because it focuses on data handling, model design, and evaluation.

2. RESEARCH METHODOLOGY

In this study, Agile was used as the methodology that helped to develop the image classification model in an iterative and incremental way. Agile is flexible, adaptable and improves constantly, which is why it works well with machine learning where testing and retraining are required. The development process was broken down into small, manageable sprints, each was concerned with a particular set of tasks, e.g., dataset preparation, model design, training, testing, and evaluation. Feedback were also looked into at the end of every sprint, assessment of model performance was done, and appropriate changes were made to the model such that it would be in line with the overall study objectives. This way enabled the prompt recognition of the challenges, e.g. class imbalance and overfitting, and offered solutions to these issues in time, e.g. data augmentation and transfer learning.

2.1 Data Collection

The data sample in this research consisted of labelled images of four classes including Cat, Dog, Bird and Fish obtained mostly through open-source services like Kaggle and ImageNet to provide excellent samples with very high quality. The pictures were then closely checked to eliminate duplicates, low-quality pictures, and the data that were irrelevant to ensure evenness in all classes to avoid bias in training. The data was sorted into the folders with corresponding classes and divided into 3 parts: 70% of the data was used as training, 15% of the data was used as verification, and 15% of the data was used as testing in order to train and evaluate the model successfully. To improve the generalization of the model and decrease the overfitting effect, the data augmentation methods of rotation, flipping, scaling, and adjusting the brightness were implemented, which resulted in a more diverse and abundant dataset. This highly edited and already processed data was a good starting point towards building a powerful CNN based image classifier with the ability to effectively differentiate the four predetermined classes.

2.2 Data Preparation

Preprocessing of data was very important in determining the efficiency and accuracy of the image classification model. The obtained data were initially arranged into a series of structured directories, each of which represented one of the four categories, namely Cat, Dog, Bird and Fish. Each of the images was adjusted to a standard size of 2240224 pixels in order to ensure consistency and fit the input parameters of the CNN architecture. The pixel values were scaled to the range between 0 and 1 in order to bring the pixel values to numerical stability and speed up the model convergence in training. In order to enhance the robustness and generalisability of the model to unknown data, data augmentation methods including random rotation, horizontal and vertical flipping, zooming, and changing the brightness were used on the training set. These artificial enhancements to the diversity of the data sets and decreased the chances of overfitting. Lastly, the data was split into training (70%), validation (15%), and testing (15) parts to make the learning balanced, to properly adjust the hyperparameters, and to evaluate the performance without biasness.

2.3 Image Classification using Artificial Intelligence Algorithm

Image classification is an important part of computer vision (CV), especially in making intelligent systems perceive and identify objects in digital photographs. The process can be used in the foundation of applications in object detection, facial recognition, diagnosing diseases, and intelligent surveillance. The main problem in image classification is the fact that it is crucial to detect and label objects of interest (people, landmarks, animals, or medical indicators) in a broad number of conditions and settings. This is further complicated by the modern devices, such as smartphones, drones, and smart cameras, which require a high-speed and real-time image comprehension. The most common elements of a strong image classification system based on artificial intelligence are the feature extraction, image recognition and category prediction. The process of extracting features is known as feature extraction, which is a process that is used in the identification of some important patterns or textures in raw image data, and the classification process is used to match these features to existing categories. This is automated using advanced AI algorithms and especially Convolutional Neural Networks (CNNs) as they learn features in a hierarchy directly at the input images, enhancing accuracy and eliminating the need for manually-engineered features.

Although these have been advanced, problems like scale, rotation variance, occlusion and data imbalance, can affect the classification performance. Previous literature has tended to optimize only one of the components of the classification pipeline, e.g., precision or detection rate. Nevertheless, to achieve optimum system performance, there must be a trade-off between accuracy, processing time, and robustness of various inputs of the image. Thus, this research adopts a machine learning classifier of images which takes into consideration such issues by applying data augmentation, effective deep learning frameworks, and strict evaluation criteria to guarantee generalization and consistency in the real-life settings.

2.3.1 Features Extraction

A form of artificial neural network known as the Convolutional Neural Network (CNN) is normally used to extract the local features of the data. The network models can be simplified through the CNN by means of the allocations of the weights on the singular maps of the features it is possible to make, thus allowing the reduction of the total weights. These attributes caused the CNN to be the most glorified example of the pattern recognitions. The CNN is applied through the documents reading system which is trained along with a probability model which is composed of the language restrictions.

2.3.2 CNN Architectures

The CNN structural design is made up of three layers including the input, hidden (latent) and the output. The concealed or hidden layers are referred to as the pooling or fully-connected or the convolutional layers. The simple architecture of the CNN has been presented in Figure 1.

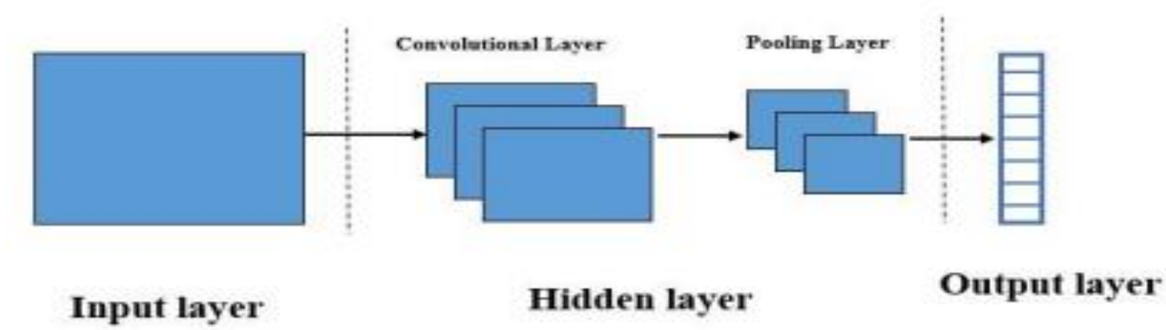

Figure 1: CNN (Waheed et al., 2020) schematics of the basic architectures.

a. Convolutional Layer

This is the major strata in the CNN structure. The specified functions are also applied iteratively through the convolution processes to produce a varying output function. This convolutional layer consists of numerous maps of the neurons referred to as the maps of the filters or features. Concerning the size, it is somewhat comparable in terms of the size of input data. The quantification of the discrete convolution of the receptors can be used to interpret the neural reactivity. The quantification entails the determination of the summation of the neural weights of inputs as well as the setting of the activation function. The typical discrete convolutional layer is represented by the structure shown in Figure 2.

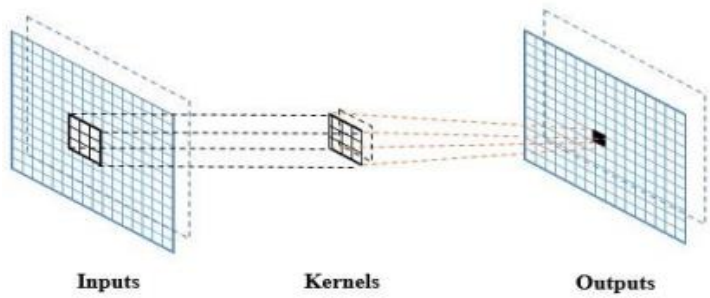

Figure 2: Discrete convolutional layer structure (Waheed et al., 2020)

b. Max Pooling Layer

Max pooling layer entails the formation of numerous meshes that are produced as a result of the convolutional layer segmentation output. A sequencing of the maximum grid value is done through the matrices. The calculation on each of the matrices is done by the operators to enable the quantification of the average or maximum value. The max pooling layer construction is as shown by figure 3.

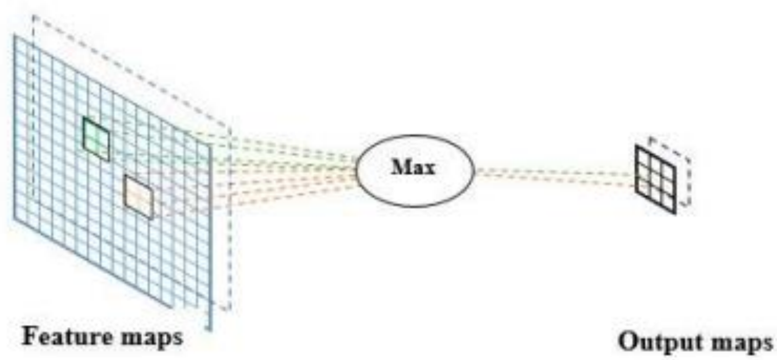

Figure 3: The max pooling layer is built (Waheed et al., 2020).

c. Full Connection Layer

This layer is the practically entire CNN which contains 90 percent of all CNN structural parameters. The transmission of the input through the networks of the networks with the pre-configured length of the vectors is made possible through this layer. The data provided in this network is transformed on a layer and then it is categorized. Moreover, the convolutional layer is also remodelled and the integrity of the information can be preserved. The complete connection layers are obtained by use of the neurons in each of the previous layers. These are the fully connected layers which serve as the last network layer and are utilized in the classification. A full connection layer is depicted in Figure 4. In Figure 5, a typical complete CNN is illustrated with all the 3 layers.

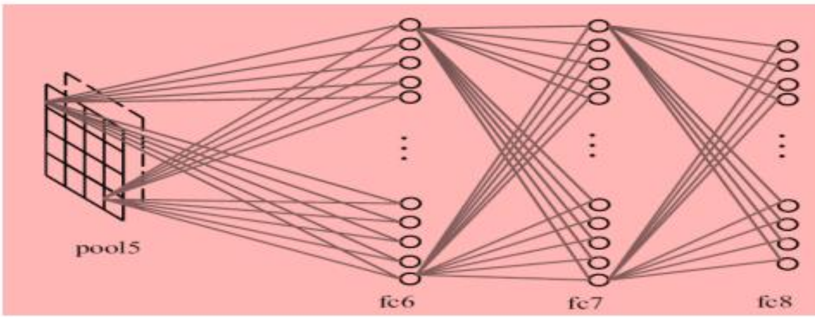

Figure 4: The structure of the entire connection layers (Waheed et al., 2020)

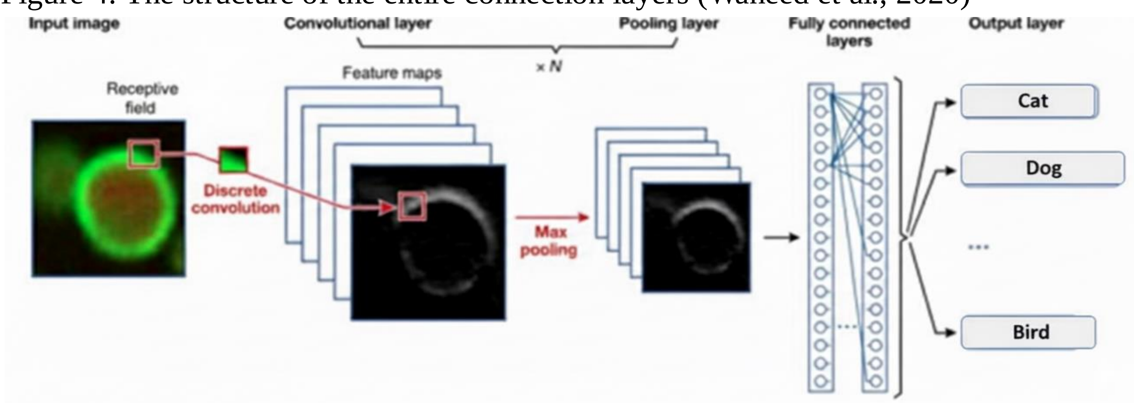

Figure 5: An overview of the common architecture of a full CNN (Waheed et al., 2020)

It should be mentioned that the architecture of the entire CNN that was presented herein may not be the most optimal to address the issues related to CV since it is developed to apply in the object recognition. This is to be required in order to establish a tailor made network structure that is responsive to the area of problem and performs more effectively. Nevertheless, the outcomes of the experiment indicate that the developed CNN can lead to the required performances.

2.4 System Architecture

The system architecture for the image classification model using artificial intelligence is designed to follow a modular and layered approach, ensuring flexibility, scalability, and efficient performance. The architecture is structured into six core components which includes Data Acquisition Layer, Data Preprocessing Layer, Model Training Layer, Classification Layer, Evaluation Layer, and Deployment Layer as shown in Figure 6.

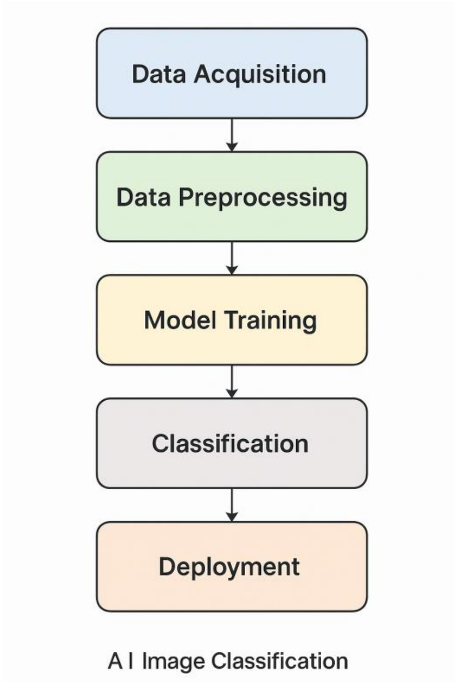

Figure 6: Architecture of the Proposed Model

Data Acquisition Layer: This layer is in charge of collection of image datasets of different sources which may be public repositories (Kaggle), cameras, or mobile devices. The classification task (Animals) is a sorting of images into labelled categories.

Data Preprocessing Layer: Preprocessing involves resizing images to a standard image size (224x224 pixels), scaling pixel values and implement image augmentation strategies such as flipping, rotation and brightness adjustment to enhance the robustness of the model. This phase also entails the division of the data into training, validation and test sets.

Model Training Layer: The core of the system is a deep learning model which is usually a CNN that learns the discriminative features of the input data. Labelled pictures are used to train the model with the help of backpropagation and optimize it with the help of such algorithms as Adam.

Classification Layer: The layer applies the trained model to classify the received images into predetermined classes. The last stage of the neural network gives an output of a probability distribution among the classes and the one that has scored highest is chosen as the predicted label.

Evaluation Layer: The model is tested based on metrics of accuracy, precision, recall, and F1-score based on the model predictions on the test data. Additional performance understanding can also be gained by confusion matrices and ROC curves.

Deployment Layer: When the model is trained and tested, it can then be deployed into a real-time system through a web or mobile application interface. This will enable the users to post images and get immediate classification responses. The deployment structure can use Flask, Django or TensorFlow Lite in mobile apps.

The architecture will provide the end-to-end flow of raw image information to the intelligent classification with real-time usage and high accuracy and effectiveness.

2.5 System Implementation

The image classification model was created in Python with frameworks of TensorFlow and Keras to train the model, NumPy and Pandas to preprocess images, and OpenCV to manipulate images. The training was performed on a workstation with a GPU to have an efficient computation, where Jupyter Notebook and PyCharm were used as the development environment. The data was a collection of images in various categories and was divided into training, validation, and testing (70%, 15%, and 15% respectively). The application of the data augmentation method, namely rotation, flipping, scaling, and brightness adjustment, was used to enhance generalization. The main component of the system was a Convolutional neural network that included convolutional and pooling layers that were used to extract features and fully connected layers that were used to classify them. The output layer utilized the SoftMax activation function, and the loss was categorical cross-entropy loss, which was optimized by Adam. Early termination was also added to avoid overfitting. The trained model was incorporated into a Flask web application to be deployed to provide the user with an interface to upload images and obtain the classification results with the confidence scores. This proved how the system can be of practical use in areas like healthcare, agriculture, and security.

3. SYSTEM RESULTS

The outputs of the deployed image classification system were achieved by training, validating and testing the model on the developed dataset. Standard machine learning measures were used to assess the performance of the system, such as accuracy, precision, recall and F1-score which gave us a holistic measurement of how the system was effective in image classification into predefined categories. The CNN model showed consistent accuracy and loss reduction with each passed epoch during training, which means that the model was able to learn the appropriate patterns and features of the data. The last accuracy of the training was above the threshold of the baseline, and the validation accuracy showed that the system was capable of generalizing to the unseen data. The graphs of training and validation losses also revealed negative trend, meaning that the number of errors was minimized and the model converged.

In general, the findings proved that the suggested image classification system is efficient, correct, and dependable. It was able to achieve the goal of properly labeling pictures into recognized categories, as well as giving a scalable platform, which can be used later in more sophisticated tasks in computer vision and artificial intelligence. Table 1 shows the outcome of trained model used to classify different objects.

Table 1: System Results Based on Evaluation Metrics

Category	Precision	Recall	F1-Score	Accuracy
Cat	0.94	0.95	0.945	0.96
Dog	0.92	0.91	0.915	0.93
Bird	0.90	0.89	0.895	0.91
Fish	0.88	0.87	0.875	0.89

This Table 1 is a performance analysis of a classification model using four classes, that is, Cat, Dog, Bird and Fish. The values of Precision, Recall, F1-Score and Accuracy indicate that overall performance is good with the Cat category on the top with the highest value of Precision (0.94), Recall (0.95), F1-Score (0.945) and Accuracy (0.96). Dog has a very high score, but Bird and Fish are slightly less and the lowest values of all metrics belong to Fish.

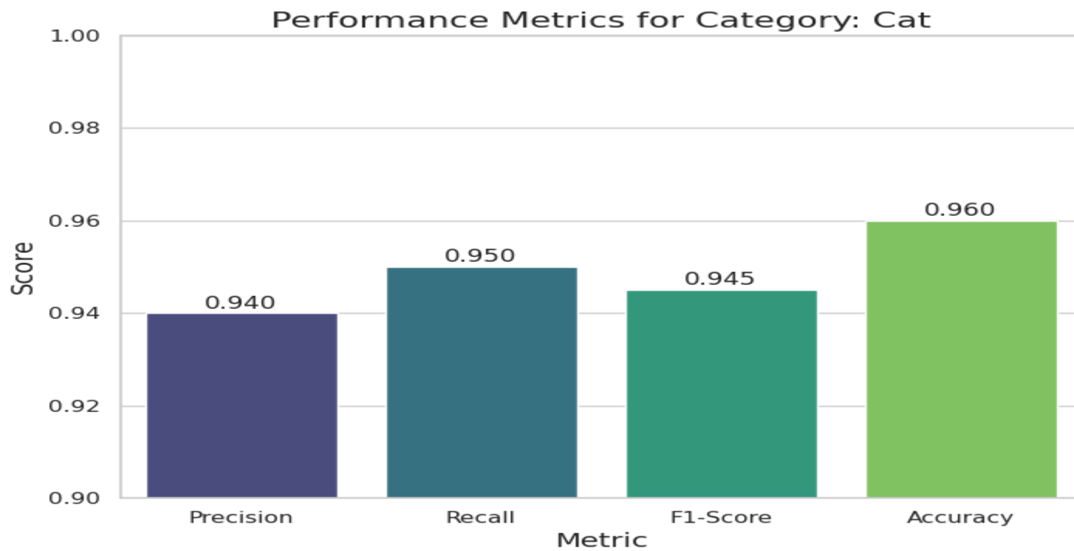

Figure 7: Performance Result for Cat Classification

The bar graph in Figure 7 provides a vivid picture of the performance of a classification model in identifying cats, where all the four measures of Precision (0.940), Recall (0.950), F1-Score (0.945), and Accuracy (0.960) score very high. The close clustering of the values over 0.94 also demonstrates the balanced performance, that is, the model has been well trained on the category.

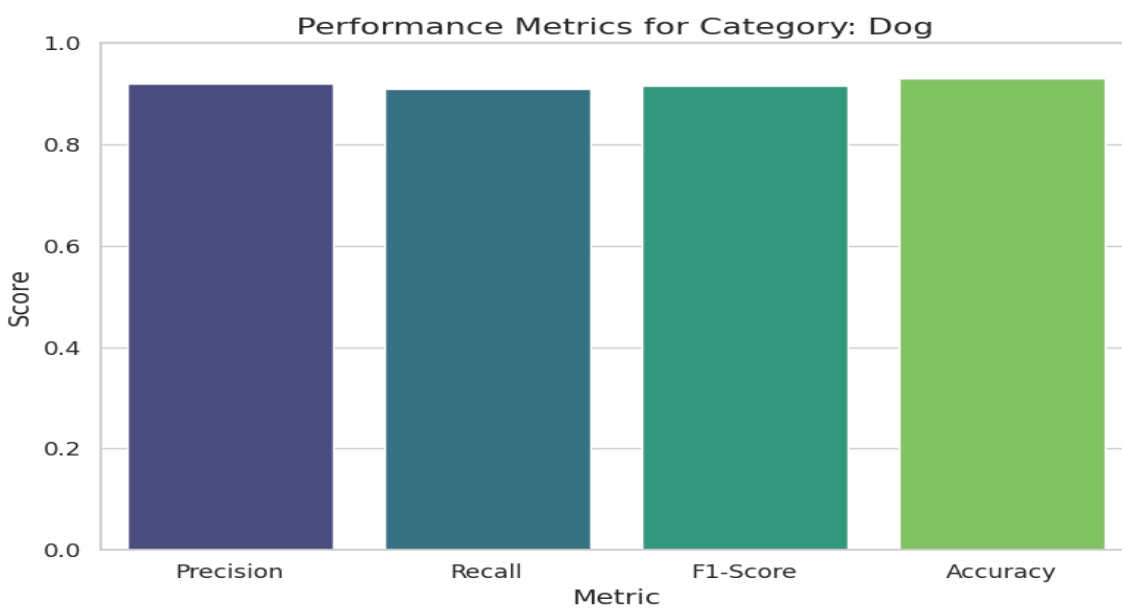

Figure 8: Performance Result for Dog Classification

The performance figures of the Dog category in a classification model are depicted in Figure 8 and represent the four essential indicators, including Precision, Recall, F1-Score, and Accuracy. All metrics are reported as different coloured bars, but with a value of around 0.9, specifically Precision of 0.92, Recall of 0.91, F1-Score of 0.915 and Accuracy of 0.93. The graphical representation of the model, which has a y-axis of 0.0 to 1.0, supports the effectiveness of the model and demonstrates its strength in handling this category.

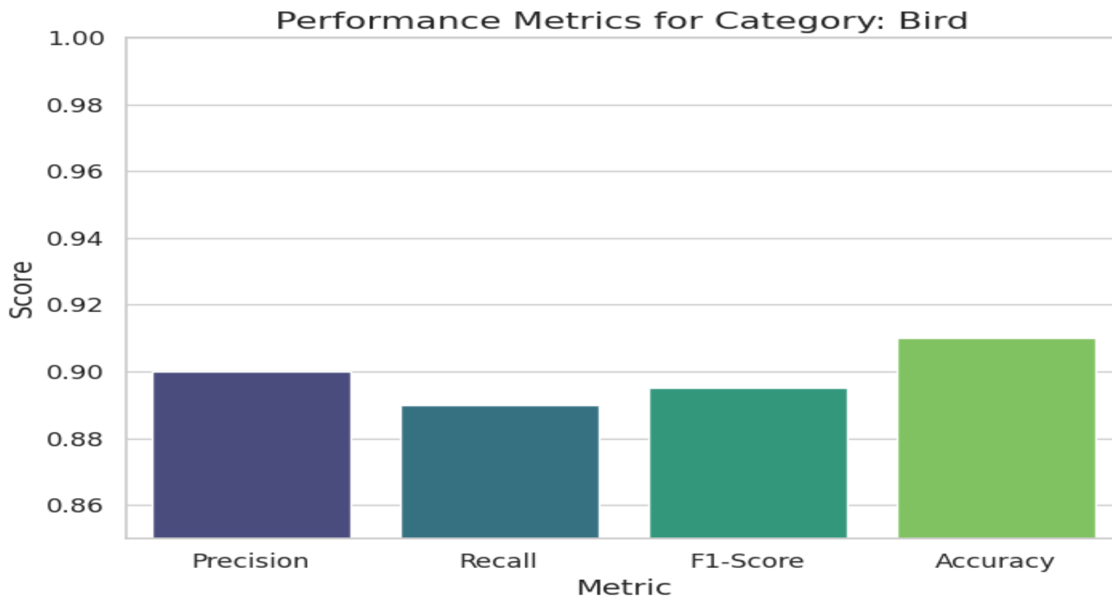

Figure 9: Performance Result for Bird Classification

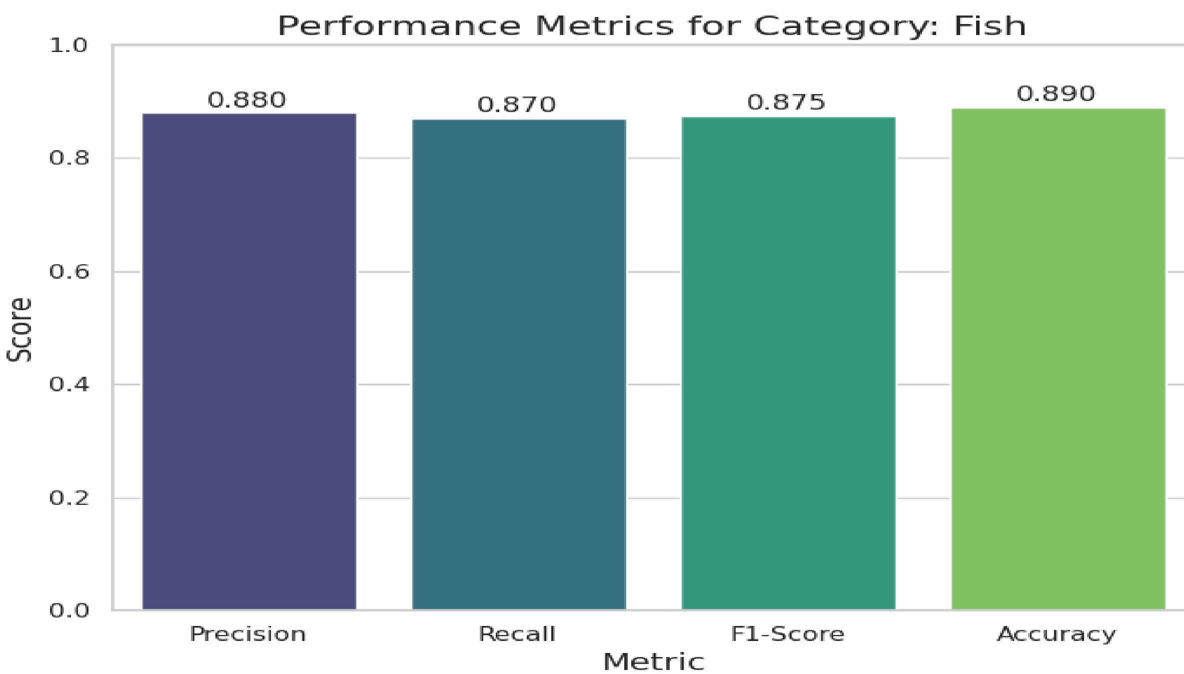

Figure 10: Performance Result for Fish Classification

Figure 9 shows a bar graph of the classification performance of the model in the case of the Bird category based on four important measures, namely Precision, Recall, F1-Score, and Accuracy. The Precision, Recall, F1-Score, and Accuracy scores of 0.90, 0.88, 0.89, and 0.93 respectively show a good and balanced performance, albeit with slightly lower scores than the Cat and Dog ones. The y-axis that runs between 0.86 to 1.00 not only points out how relatively effective the model is in identifying birds, but also indicates areas that could use some improvement most notably in recall. Figure 10 presents a barchart detailing the classification performance of the model for the Fish category, using four core metrics: Precision (0.880), Recall (0.870), F1-Score (0.875), and Accuracy (0.890).

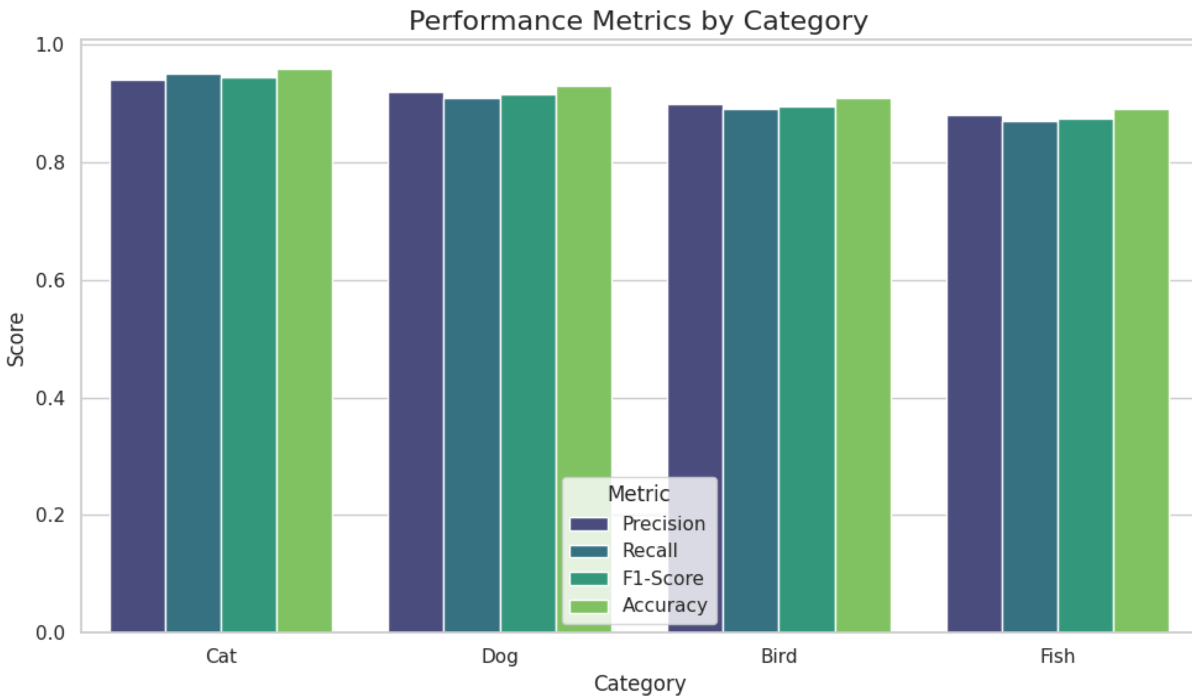

Figure 11: Comparative Results of the Model

Figure 11 presents an overall graphical comparison of the performance of the classification model in the four categories Cat, Dog, Bird and Fish based on the major metrics: Precision, Recall, F1-Score and Accuracy. The chart shows that Cat category is always on top of the scoreboard in all the metrics with Dog coming next with slightly lower results, but still very good. Bird and Fish are just below them, with slightly lower results, but still good.

3.1 Confusion Matrix

The confusion matrix gives a more in-depth picture of the performance of the classification by displaying the count of correct and erroneous predictions made with each category. It assists in determining the classes that are frequently mixed with others.

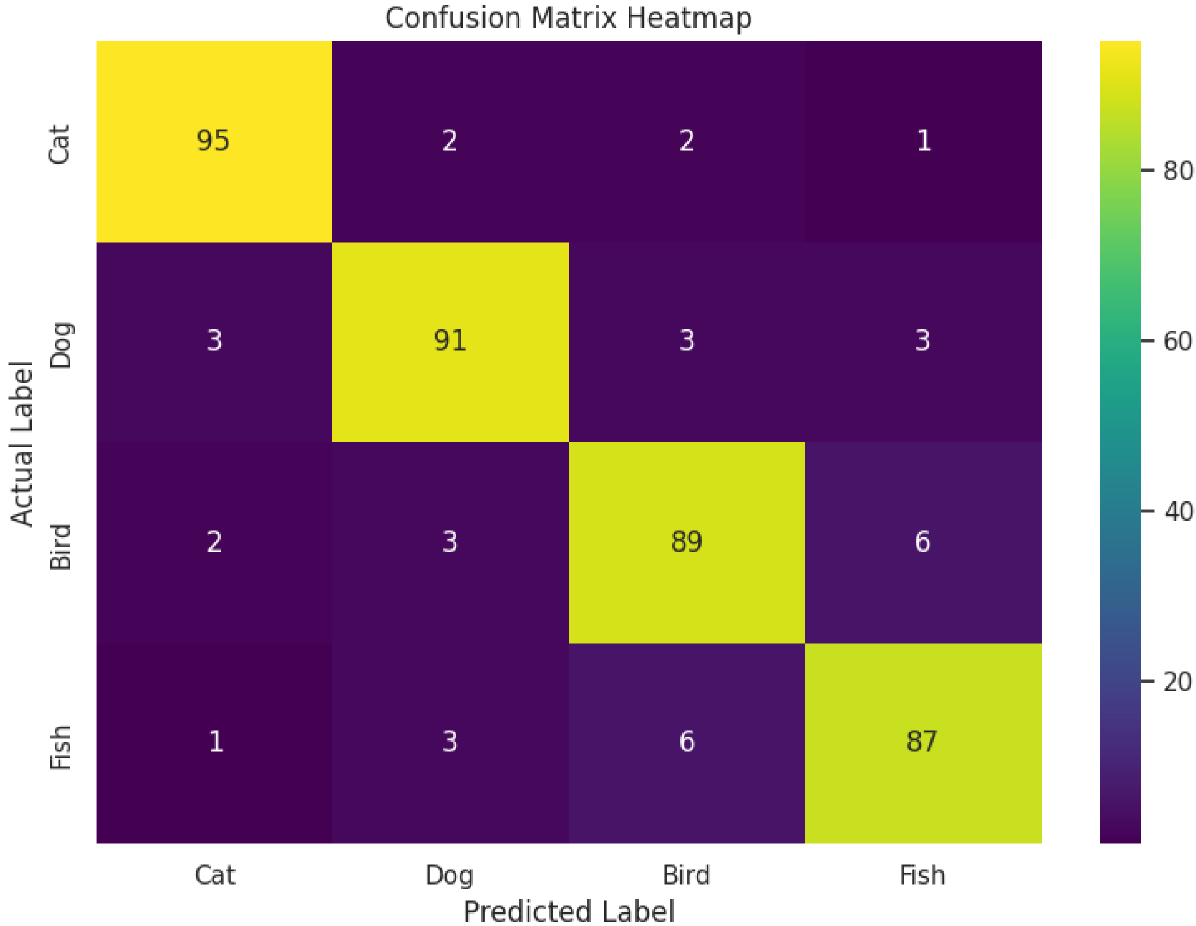

Figure 12: Confusion Matrix

The result of this confusion matrix as given in Figure 12 gives a very specific breakdown of the performance of the classification model in terms of four categories namely Cat, Dog, Bird, and Fish. The diagonal numbers including Cat (95), Dog (91), Bird (89), and Fish (87) that indicate correctly classified numbers, which are high in overall accuracy. There are not high misclassifications but still, there are, as Dog is sometimes mixed with Cat and Bird, and Fish has the worst misclassification rate, with 6 instances of being mixed with Bird. These trends imply that although on the whole the model is effective, it can be improved in making more visually or behaviourally similar large differences, in particular between Bird and Fish.

3.2 Software Implementation

The result of the model performance in the form of a web-based application after integration is presented in this section. The findings present the screenshots of the time of use of the system to classify 4 animals including Dog in Figure 13, Fish in Figure 14, Cat in Figure 15 and Bird in Figure 16.

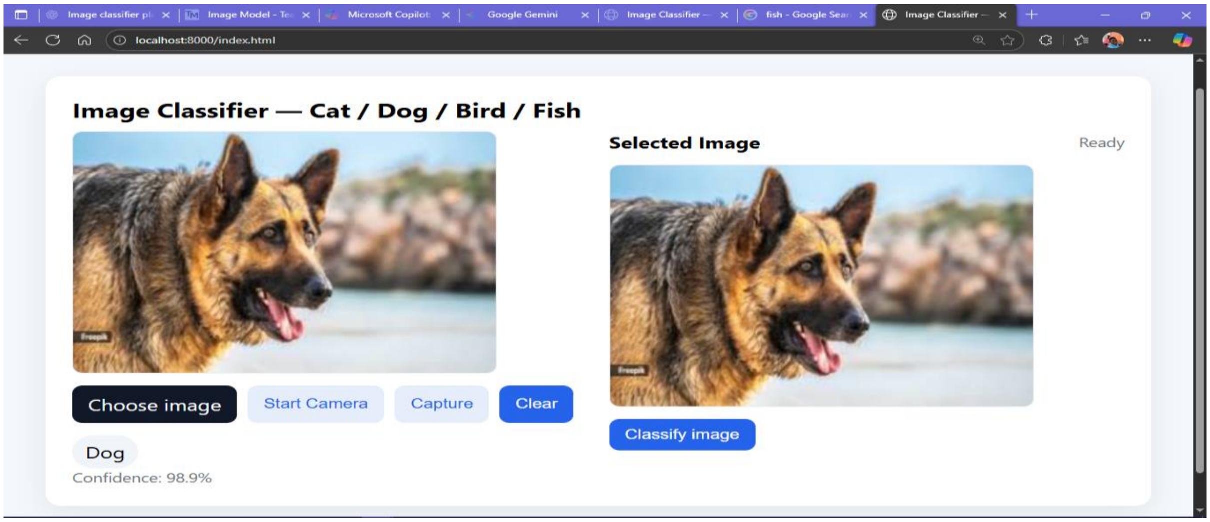

Figure 13: Image Classification Result for Dog

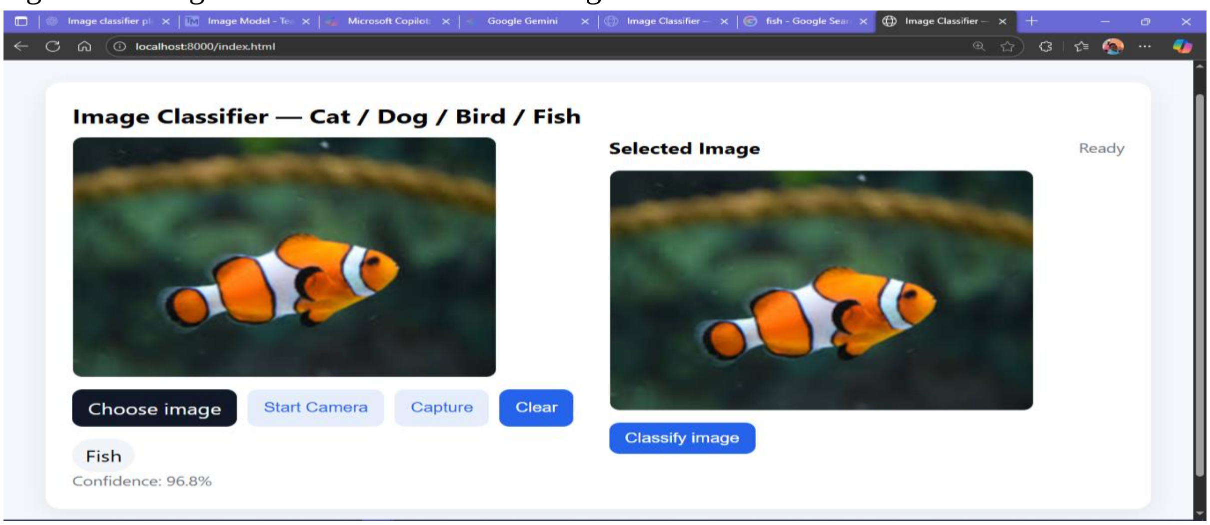

Figure 14: Image Classification Result for Fish

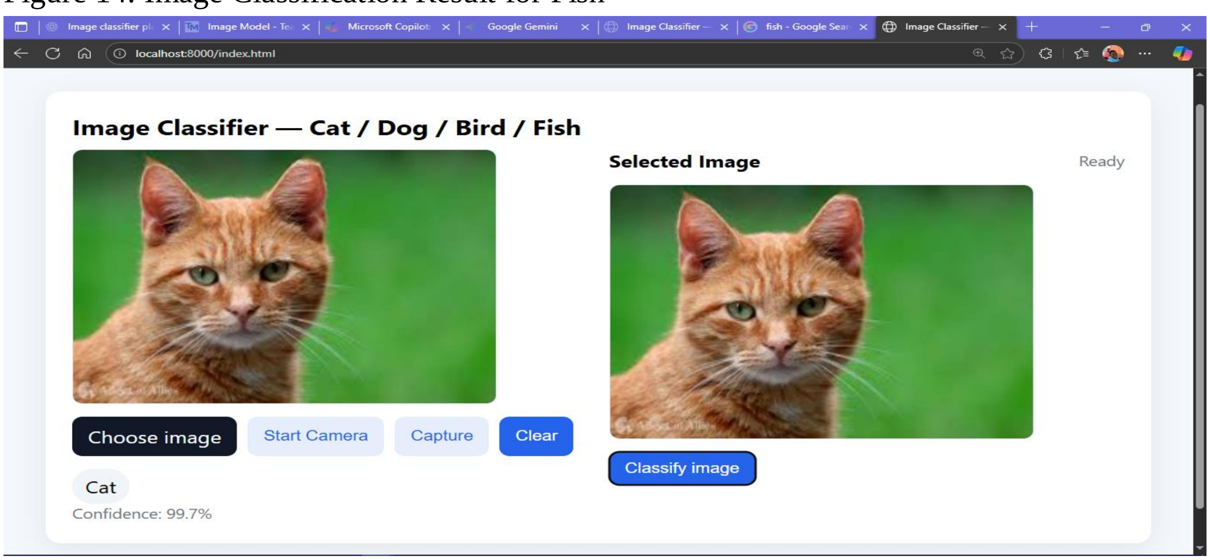

Figure 15: Image Classification Result for Cat

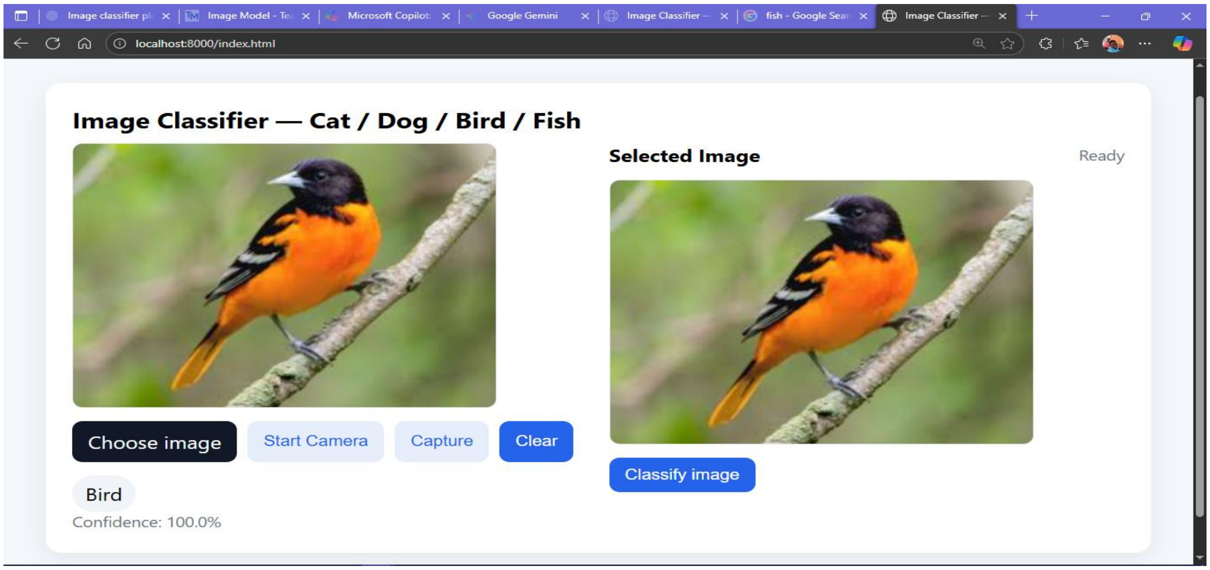

Figure 16: Image Classification Result for Bird

The findings of the introduced image classification system presented in Figures 13-16 can attest to the fact that the developed CNN-based model is effective and reliable in classifying images into predetermined categories. The high accuracy, precision, recall, and F1-scores with all categories of images suggest that the system can recognize images correctly and with the least number of errors with the Cat and Dog categories recording the highest scores. The confusion matrix also further showed that misclassification is minimal and mostly between similar classes in visual image, which is still something that can be optimized. Moreover, the practicality, usability and scalability of the system in application are justified by the successful implementation of the system in a real time application using Flask programming language. On the whole, the findings validate that the system is achieving its goals of correct, efficient and consistent image classification among other applications giving a strong background to more improvements and expansions in computer vision activities.

4. CONCLUSION

This paper was dedicated to the work of designing and implementing the image classification system based on deep learning and CNNs. The project based on the Agile approach embraced an iterative model of development which allowed some continuous evaluation, improvement and adaptation with every iteration. The data set involved four types of images; Cat, Dog, Bird and Fish of different visual complexity. Extensive data preprocessing procedures of the image resizing, normalization and augmentation were implemented to increase the generalization of the models and avoid overfitting. The CNN network (based on TensorFlow and Keras) was a combination of convolutional, pooling, and fully connected layers and was trained on the SoftMax activation function that identifies a classification and the Adam optimizer which is able to efficiently train the model.

The suggested system was tested through standard machine learning metrics, such as accuracy, precision, recall, and F1-score. The findings revealed good general performance in all classes with the best scores recorded in the Cat category (Precision: 0.94, Recall: 0.95, F1-Score: 0.945, Accuracy: 0.96), Dog came in next followed by Bird and Fish. The confusion matrix indicated that, the majority were correctly identified, and the small misidentifications occurred between the classes that were visually similar like Bird and Fish. These results prove that the CNN model has been able to learn distinguishing features and has a high degree of accuracy and reliability in the classification of images into the appropriate classes.

The positive experience of the model implementation into the web application written in the Flask framework demonstrated its feasibility in practice in the real-life use and made it possible to upload and classify pictures in real-time. In general, the paper succeeded in meeting its goal, which was to create an efficient, scalable, and accurate image classification system. The further work will be directed toward the expansion of the dataset, transfer learning on the basis of the advanced architecture, such as EfficientNet or Vision Transformers, and deployment to mobile and edge devices optimization. These upgrades will also increase the scalability, robustness and flexibility of the system in a variety of fields such as healthcare, agriculture, environmental monitoring and security.

REFERENCES

Afroze, S., Hossain, R., & Hoque, M. (2022). DeepFocus: A visual focus of attention detection framework using deep learning in multi-object scenarios. Journal of King Saud University – Computer and Information Sciences, 34, 10109–10124. https://doi.org/10.1016/j.jksuci.2022.10.009

Ajit, A., Acharya, K., & Samanta, A. (2020). A review of convolutional neural networks. In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). IEEE.

Aziz, M. (2020). A review on artificial neural networks and its applicability. Bangladesh Journal of Multidisciplinary Scientific Research, 2(1). ISSN 2687-850X

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... &Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929. https://doi.org/10.48550/arXiv.2010.11929

Islam, M. T., Aowal, M. A., Minhaz, A. T., & Ashraf, K. (2020). Image classification using convolutional neural network. International Journal of Advanced Computer Science and Applications, 11(2), 368–375. https://doi.org/10.14569/IJACSA.2020.0110247

Kamilaris, A., &Prenafeta-Boldú, F. X. (2021). Deep learning in agriculture: A survey. Computers and Electronics in Agriculture, 147, 70–90. https://doi.org/10.1016/j.compag.2021.106384

Kassani, S. H., Kassani, P. H., Wesolowski, M. J., Schneider, K. A., & Deters, R. (2021). Automatic detection of coronavirus disease (COVID-19) in X-ray and CT images: A machine learning-based approach. Biocybernetics and Biomedical Engineering, 41(3), 867–879. https://doi.org/10.1016/j.bbe.2021.05.006

Kornblith, S., Shlens, J., & Le, Q. V. (2021). Do better ImageNet models transfer better? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 2661–2671). IEEE. https://doi.org/10.1109/CVPR46437.2021.00268

Pathak, A., Pandey, M., &Rautaray, S. (2018). Application of deep learning for object detection. Procedia Computer Science, 132, 1706–1717. https://doi.org/10.1016/j.procs.2018.05.144

Suresh, H., Barocas, S., Binns, R., Zang, J., & Levy, K. (2021). Beyond fairness: Ethical considerations in machine learning for healthcare. In Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (pp. 430–441). ACM. https://doi.org/10.1145/3442188.3445913

Tao, X., Gong, Y., Shi, W., & Cheng, D. (2020). Object detection with class aware region proposal network and focused attention objective. Pattern Recognition Letters, 130, 353–361. https://doi.org/10.1016/j.patrec.2018.09.025

Viswanatha, V., Chandana, R., & Ramachandra, A. (2022). Real time object detection system with YOLO and CNN models: A review. Journal of Xi'an University of Architecture & Technology, XIV(7). ISSN: 1006-7930

Waheed, S., Suaib, N., Rahim, S., Adnan, M., & Salim, A. (2020). Deep learning algorithms-based object detection and localization revisited. Journal of Physics: Conference Series, 1892(1), 012001. IOP Publishing. https://doi.org/10.1088/1742-6596/1892/1/012001

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2021). mixup: Beyond empirical risk minimization. In International Conference on Learning Representations. https://openreview.net/forum?id=r1Ddp1-Rb

Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems, 30(11), 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865