Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

Image based Bird Species Identification using Convolutional Neural Network

Satyam Raj[1], Saiaditya Garyali[2], Sanu Kumar[3]
[1, 2, 3] B.E. Scholar
Department of Computer Science and Engineering
Sir M. Visvesvaraya Institute of Technology
Bengaluru, Karnataka, India

Sushila Shidnal
Assistant Professor
Department of Computer Science and Engineering
Sir M. Visvesvaraya Institute of Technology
Bengaluru, Karnataka, India

***Abstract* — Life's routine tempo appears to be rapid and energetic and includes diverse tasks. Bird-watching is a popular hobby which offers relaxation in everyday life. Innumerable people visit bird sanctuaries to observe the elegance of different species of birds. To provide birdwatchers with a convenient tool for identifying the birds in their natural habitat, we developed a Deep Learning model to help birders recognize 60 bird species. We implemented this model to extract information from bird images using the Convolutional Neural Network (CNN) algorithm. We gathered a dataset of our own using Microsoft's Bing Image Search API v7. We created an 80:20 random split of the data. The classification accuracy rate of CNN on the training set was observed to be 93.19%. The accuracy on testing set was observed to be 84.91%. The entire experimental research was carried out on Windows 10 Operating System in Atom Editor with TensorFlow library.**

Keywords* — *Deep Learning, CNN Model, Classification and Prediction, TensorFlow, Keras

I. INTRODUCTION

Deep Learning is a Machine Learning subfield which is in turn a subfield of Artificial Intelligence. Deep learning can be visualized as a platform where artificial, human brain-inspired neural networks and algorithms learn from large amounts of data. Deep Learning allows computers to solve complex problems even though they use a very diverse, unstructured, and interconnected data set. The more Deep Learning algorithms learn, the better they perform.

Nowadays, bird species identification is seen as a perplexing problem which often leads to confusion. Birds allow us to search for certain species within the ecosystem as they react rapidly to changes in the atmosphere; but collecting and gathering information on birds needs tremendous human effort. Many people visit bird sanctuaries to look at the birds, while they barely recognize the differences between different species of birds and their characteristics. Understanding such differences between species can increase our knowledge of birds, their ecosystems and their biodiversity. The identification of birds with bare eyes is based solely on the basic characteristics due to observer constraints such as location, distance and equipment, and appropriate classification based on specific characteristics is often found to be tedious. Ornithologists have also faced difficulties in distinguishing bird species. To properly identify a particular bird, they need to have all the specificities of birds, such as their distribution, genetics, breeding climate and environmental impact. A robust system is needed for all these circumstances that can provide processing of large scale bird information and serve as a valuable tool for scholars, researchers and other agencies. The identification of the bird species from the input of sample data therefore plays an important role here.

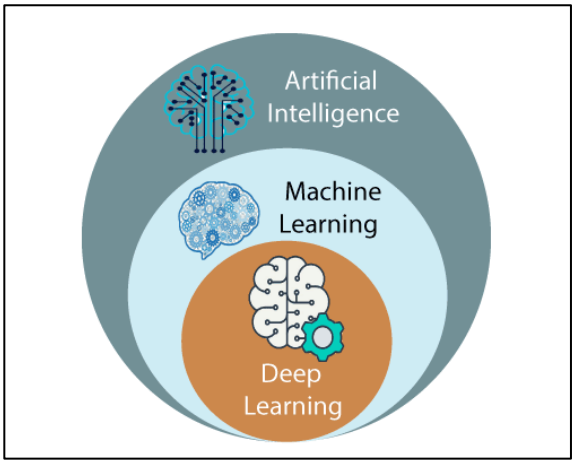

Fig 1: A Venn diagram defining deep learning as a machine learning subfield which is in effect a subfield of artificial intelligence.

Bird identification can generally be done with the images, audio, or video. The audio or video processing technique makes it possible to detect birds by analyzing audio and video signals; however, the processing of such information is made more complicated by mixed sounds such as insects and the presence of other real-world objects in the frame. People are typically more effective at finding images than audios or videos. Therefore, it is easier to use a picture over audio or video to identify birds.

To predict the birds in their natural habitats, we developed an interface to extract information from bird images using the Convolutional Neural Network (CNN) algorithm. First, a vast dataset of birds were gathered and localized. Second, CNN architecture was designed similar to the VGGNet Network. Now that the network was implemented, we trained the CNN model with the bird dataset using Keras, and subsequently the classified, trained data was stored on the disk to identify a target object. Ultimately, the client-server architecture navigates a sample bird image submitted by an end-user to retrieve information and predict the bird species from the qualified model stored on the disk. This method allows the autonomous identification of birds

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

from the captured images and can provide important, useful knowledge about bird species.

II. RELATED WORKS

A. Bird Species Recognition Using Support Vector Machines [1]

This paper examines automatic detection of bird species through their vocalization. Recognition is performed at each node in a decision tree with a Support Vector Machine (SVM) classifier that classifies between two species. Recognition is tested with two collections of bird species which have previously been studied using different methods. Recognition resulted in a better or equivalent output indicated by the proposed approach relative to then existing reference models.

B. Bird Species Classification Based on Color Features [2]

The proposed approach to classifying bird species is based on color characteristics drawn from unconstrained images. A color segmentation algorithm is applied to the image in the first step of the technique to remove background elements and delimit candidate regions where the bird may be present. The image is then bifurcated into component planes and standardized color histograms from each plane are calculated from these candidate regions. Eventually, aggregation processing is used to cut down the number of histogram intervals to a fixed number of bins. Instead, a learning algorithm uses these histogram bins as feature vectors to differentiate between the various bird types.

C. Image Recognition with Deep Learning Techniques[3]

Deep Learning methodology has been used in this research work for the recognition of images. Two versions of deep learning neural networks were considered: Convolutional Neural Network (CNN) and Deep Belief Network (DBN). Caltech101 dataset was chosen to train and test the above proposed models. The SVM-KNN algorithm was considered as the benchmark model, selected by the Caltech101 database issuer. After several dataset preprocessing techniques, using the above proposed approach, a correct recognition score of 67.23% was obtained, which was an increase of 1% over the recognition score obtained by the chosen benchmark algorithm.

D. Bird species recognition based on SVM classifier and decision tree [4]

In this paper, the ratio of the distance of the eye to the root of the beak and the distance of the width of the beak was used to distinguish different bird species. A new bird species recognition algorithm was proposed to achieve the final recognition result by integrating these new features into the multi-scale decision tree and the SVM framework. The proposed approach has achieved a correct classification rate of around 84%.

E. Bird Species Categorization Using Pose Normalized Deep Convolutional Nets [5]

In this study, architecture was proposed for fine-grained visual categorization. The approach reflected human expert success in classifying bird species. First, the architecture calculates an estimate of the pose of the object which is further used to calculate features of local images. In addition, these characteristics are used for classification purposes. The features are determined by applying deep convolutionary nets to patches of the image that the pose locates and normalizes. A thorough analysis was carried out for state-of-the-art implementations of deep convolution technologies and fine-tuning feature learning for fine-grained classification. The experiments advance the state-of-the-art success on bird species identification, with a substantial increase in the right levels of classification over the previous methods (75% vs. 55-65%).

F. Bird Identification by Image Recognition [6]

The principal plot of this project was to classify the bird species from the user's picture as input. Transfer Learning is the technology used to fine-tune a pretrained model (AlexNet). Used for classification is SVM (Support Vector Machine) which is a supervised machine learning algorithm. MATLAB was used because it is suitable for the implementation of advanced algorithms and gives good accuracy in numerical precision. The developer has reached an accuracy of about 80% - 85%. This project extends much breadth as it meets the intent. This concept can be implemented with camera traps in wildlife research and also in monitoring to keep records of wildlife movements in specific habitats and behaviors of any species.

G. Automatic Classification of Flying Bird Species using Computer Vision Technique [7]

This work aimed at developing a reliable and automated system capable of classifying individual species of birds, using video data during flight. This piece introduced a new and rich collection of video-classification appearance features. Features of motion including curvature and frequency of wing beat were added. Seven organisms made up the dataset. The experimental evaluations of the appearance and motion features were presented in combination with the Normal Bayes classifier and a Support Vector Machine classifier. A classification rate of 92 percent and 89 percent, respectively, was achieved using Normal Bayes and SVM classifiers.

H. Audio Based Bird Species Identification using Deep Learning Techniques [8]

This paper introduced a new method of audio classification for the identification of bird species. While most approaches used the nearest neighbor matching or decision trees for each individual bird species with extracted templates, the authors of this paper used speech recognition techniques in the field of deep learning. After incorporation of all data preprocessing requirements and data increase methods, a Convolutional Neural Network was formed. The network architecture achieved a 0.686 accuracy score while predicting each sound file's main species and scored 0.555 when background species were used as the extra target.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

III. PROPOSED DEEP LEARNING MODEL

The Convolutional Neural Network (CNN) is a deep learning algorithm which includes an input image and assigns the weights and the distinctions to the various aspects of the images and can then distinguish one image from another. The pre-processing required in CNN compared with other classification algorithms is much lower. In primitive methods, filters were usually hand-engineered; on the other hand, CNN has the ability to learn these filters on its own when subjected to enough number of trainings.

CNN's architecture is quite similar to that of the pattern of neuron connectivity in the human brain, in which individual neurons respond only to stimuli in the receptive field. These receptive areas collectively overlap the entire visual area.

The initial parameters to be known are the elements that are significant part in the operation of Convolutional Neural Networks.

- Input Image
- CNN
- Output Label (Image Class)

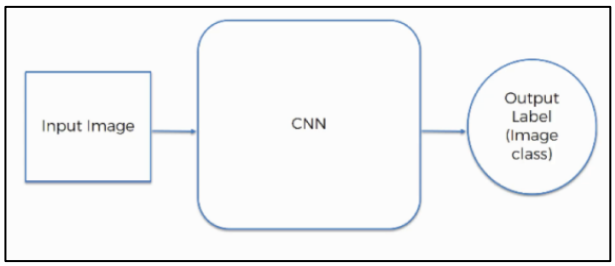

Fig 2: A diagram depicting interaction of the elements

Steps involved in the process of developing Convolutional Neural Networks are:

- Convolution followed by the application of Rectifier Function
- Pooling
- Flattening
- Full Connection

Convolution followed by ReLU:

Convolution is the first step in the process. Mathematically, convolution is a defined as an operation on two functions, which produces a third function expressing how the shape of one is modified by the other.

The mathematical expression of Convolution goes as:

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau \qquad (1)$$

Input Image, feature sensor and function map are the three elements that are used in the convolution operation. Convolution is important because it helps in reducing the size of the input image. The feature detector's very motive is to slide through the input image attributes and filter the integral parts into the feature map, and exclude the rest. Convolutional Neural Networks develop multiple feature detectors and use them to develop multiple feature maps known as convolutional layers. Through training, the network determines all the features that are important for them in order to scan images and detect them precisely. In many instances, the network considered features are left unnoticed by the human eye, and therefore convolutional neural networks are so incredibly helpful.

The Rectified Linear Unit or the ReLU is an additional step to the Convolution process. This corrective function is used so that the nonlinearity of the images is increased. Images are naturally non-linear and the rectifier serves to make up for the linearity that might have been inflicted in the image when it underwent the convolution operation. The fundamental difference between the unrectified and rectified versions of the image is the progression of colours.

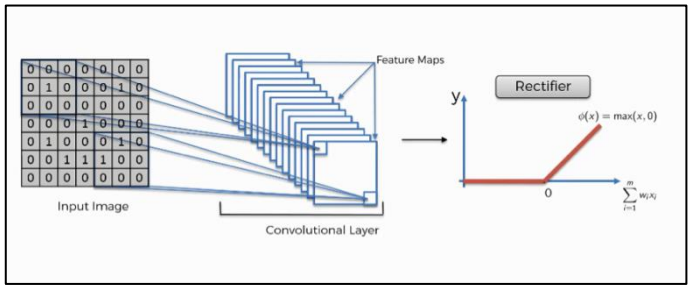

Fig 3: A diagram depicting Convolution followed by the application of Rectifier Function

Pooling:

A Convolutional Neural Network should incorporate the property of Spatial Variance. This means that, during prediction, the network should remain unaffected with the location of features, irrespective of their position in the training set of images. If the features are slightly angled and if the function is very different in texture; whether the features are a little closer or if they are a little further apart, the network trained should not be affected. So, even if the feature itself is a bit distorted, the neural network must possess some level of flexibility to be able to precisely detect that feature and this is what pooling is all about. Pooling offers the spatial variance competence to the convolutional neural network. In addition, Pooling serves to minimize the size of the image and the number of parameters through dimensionality reduction, in order to reduce the computational power required to process data which, in turn, prevents over-fitting.

There are three types of Pooling, which include:

- Mean Pooling
- Max Pooling
- Sum Pooling

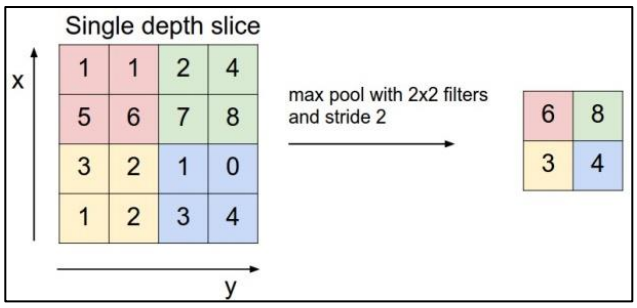

Fig 4: A diagram depicting Max Pooling

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

Flattening:

In the Flattening operation, we re-organize the Pooled Feature Map into a column of values. The result of the flattening operation is a long vector of input data which is meant for passing through the artificial neural network for further processing.

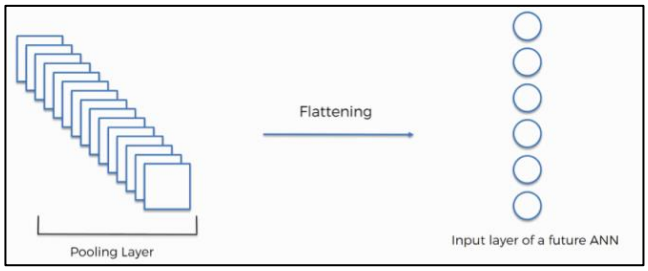

Fig 5: A diagram depicting Flattening of Pooled Feature Maps

Full Connection:

This is the final step in the process of creating a convolutional neural network. It's in this step where we add the artificial neural networks to the convolutional neural networks.

There are three layers in the Full Connection Step:

- Input Layer
- Fully Connected Layer
- Output Layer

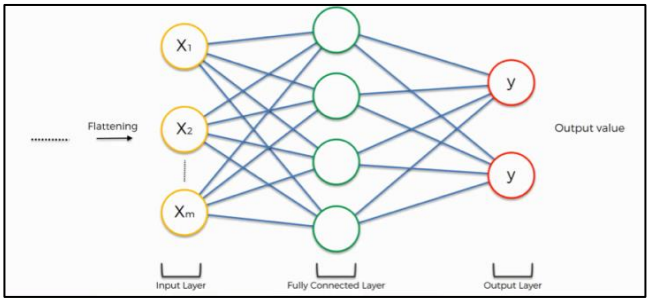

Fig 6: Fully Connected Layer

The input layer contains a data vector resulting from the flattening process. The role of the artificial neural network is to accept this data and merge all the features into a wider range of features, making the convolutional network more capable of classifying images. The output layers that make forecasts are at the end of the network.

The full connection process works as follows:

- The receptor neuron detects a certain feature in a fully-connected layer and retains its value.
- This value is communicated to all the classes by the neuron.
- The classes control and determine if the value of the feature is applicable to them.

After the Convolutional Neural Network has finally been developed from scratch, the final step is to test the effectiveness of the network.

IV. METHODOLOGY

The principal aim of this project is identification of images of birds and their classification into the individual species. This project has been developed on the mainstream algorithm of Deep Learning, which is Convolutional Neural Network (CNN).

The entire system is built atop Python3 in Atom Editor and deployed in the Django web framework.

Hardware Requirements:-

- CPU: 8th Generation Intel® Core™ i7 Processor
- RAM: 8 GB or above
- Hard Disk: 500 GB or above
- GPU: NVIDIA GTX 960

Software Requirements:-

- Operating System: Windows 10
- IDE: Atom Editor
- Language: Python v3.8.2
- Backend Libraries: Keras, TensorFlow, NumPy, Scikit-Learn, OpenCV
- Frontend: HTML5, CSS3, JavaScript

The execution of the entire project comprises four main steps:

A. Gathering and Localizing the Bird Dataset.

In order to build our Deep Learning image dataset, we utilized Microsoft's Bing Image Search API v7. Bing Image Search API is a comprehensive Cognitive Service family of Microsoft. The dataset mostly includes the birds found in Asian sub-continent. The entire dataset houses 60 species of birds and consists of 8218 images.

B. Implementing the CNN architecture.

The CNN architecture to be developed is a smaller and more portable version of the VGGNet network [9].

Characteristics of VGGNet architectures are:

- 3X3 Convolutional Layers stacked on each other at increasing depth.
- Max Pooling to minimize the size of the image and the number of parameters.
- Fully Connected Layers at the end of the network prior to a softmax classifier.

TensorFlow Backend is used for implementing this architecture. Multiple layers of Convolution and ReLU are stacked together in order to learn an affluent set of attributes.

The Convolution Layer has 32 filters with 3X3 feature detector for the first convolution block. This operation is followed by application of ReLU function. The Pooling layer incorporates a 3X3 pool to reduce the spatial dimensions from 96X96 to 32X32 (96X96X3 dimensional images were used to train the network).

Another Convolution Layer is stacked onto this, where filter size is increased from 32 to 64 but the feature detector still being 3X3 dimensional. This is again followed by ReLU function. Further, Max Pooling is applied where the pool window size is decreased from 3X3 to 2X2 at strides of 2.

A final Convolution Layer is used where filter size is increased to 124 followed by implementation of ReLU function. Max Pooling is applied again with the pool window

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

size of 2X2 at strides of 2. A Dropout Layer is used to prevent overfitting with the dropout value of 0.25.

Finally, the Fully Connected Layers are added using the Dense Layer of size 1024. A Dropout Layer is implemented again with the value of 0.50.

At the end, Softmax classifier is used for predicting a single class out of various mutually exclusive classes.

C. Training the CNN Model.

After successful deployment the CNN Model, the network was all set to be trained with the bird images using Keras using Adam Optimizer. All the necessary packages were imported in the training script. Matplotlib backend was used for saving figures in the background.

For data augmentation, the ImageDataGenerator class has been used to increase the diversity of the information available for training models significantly without actually collecting new information. This technique also helps in preventing overfitting.

A common practice is to divide the dataset into training and testing sets when implementing deep learning. An 80:20 random split of the dataset was created with the help of train_test_split function, rendering 80% of the data for training and the remaining 20% for testing.

After the CNN finished training, both the Model and Label Binarizer files were saved to the local disk as it is required to load them in the framework, whenever the network is tested on images extrinsic to the training and testing dataset.

D. Testing the Efficacy of the Trained Model.

Now that the CNN model was trained, a classification script was implemented to identify images of birds.

The user browses and uploads a sample image through the web portal. Ultimately, the client-server architecture navigates the submitted sample bird image to the testing script. The script retrieves information from the trained model and label binarizer file stored on the disk thereby successfully predicting the bird species.

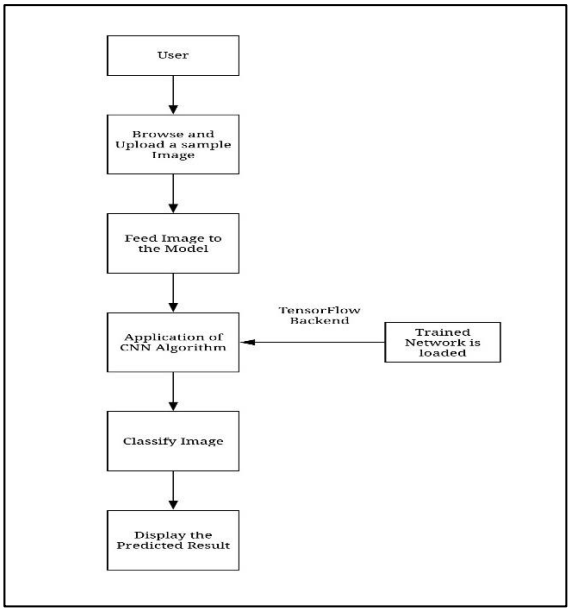

Fig 7: Flow of System

V. EXPERIMENTAL RESULTS AND ANALYSIS

The Bird Dataset comprises 8218 images for 60 bird species. We used TensorFlow Backend for the CNN architecture and Adam Optimizer while training the model. An 80:20 random split of data was enforced for training and testing respectively. The network was trained for a total of 60 epochs. The initial learning rate was initialized with the default value of 1e-3. The Batch Size was assigned the value 32. The spatial dimensions of the input image were constrained to 96X96 pixels with 3 channels (namely RGB). The entire training took around 5 hours on our machine configuration.

The classification accuracy rate of CNN on the training set was observed to be 93.19%. The accuracy on testing set was observed to be 84.91%. The observed metrics were plotted on a graph which is depicted in the following figure.

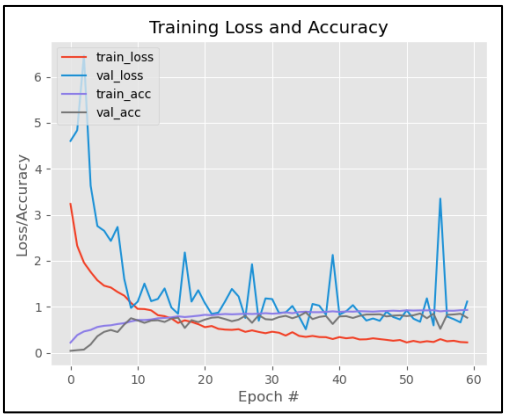

Fig 8: Training Loss and Accuracy Plot

The sample image input by the end user is parsed through the client server architecture to the prediction script in form of argument. Here the image is loaded and preprocessed. The trained model and the label binarizer file are loaded in the memory. Consequently, the prediction is made.
The output image is displayed in a new window instantiated by cv2.imshow() function with the prediction imprinted in it.

As an example, let's consider the below Fig 9, as the input image fed to the system for its prediction.

Fig 9: Input Image

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 06, June-2020

Looking into the evaluation by the model, the system generates a new window with prediction imprinted in it along with the accuracy metrics.

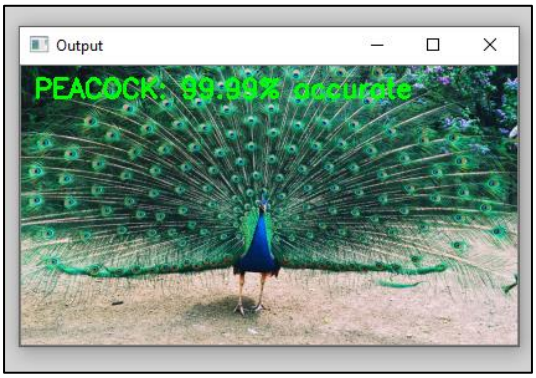

Fig 10: Output Terminal depicting the Prediction

VI. CONCLUSION AND FUTURE ENHANCEMENTS

In this paper, we have proposed a method to predict the bird species from images using the most sought algorithm of Deep Learning, Convolutional Neural Network. We developed the entire CNN Model from scratch, imparted training to it and finally tested its efficacy. The application developed is generating results, with a high accuracy of 93.19% on training set and 84.91% on the testing set.

The application can be enhanced in the following possible ways:

- Currently we are predicting 60 species of birds on our system configuration. The number of species can be invariably extended.
- Presently we have attained an accuracy of around 93%. The number of image per class can be increased to attain a higher accuracy.
- Google Maps API can be used to display the locations where birds are found.
- Description about the predicted bird can be displayed which can be either manually added or extracted from internet sources.
- An extensive database can be developed to keep track of users and their activities.
- This application developed for a PC can be further devised in form of Mobile Application.

REFERENCES

[1] Fagerlund, Seppo. "Bird species recognition using support vector machines." EURASIP Journal on Advances in Signal Processing 2007, no. 1 (2007): 038637.

[2] Marini, Andréia, Jacques Facon, and Alessandro L. Koerich. "Bird species classification based on color features." In 2013 IEEE International Conference on Systems, Man, and Cybernetics, pp. 4336-4341. IEEE, 2013.

[3] Barar, Andrei Petru, Victor Neagoe and Nicu Sebe. "Image Recognition with Deep Learning Techniques." Recent Advances in Image, Audio and Signal Processing: Budapest, Hungary, December 10-2 (2013).

[4] Qiao, Baowen, Zuofeng Zhou, Hongtao Yang, and Jianzhong Cao. "Bird species recognition based on SVM classifier and decision tree." First International Conference on Electronics Instrumentation & Information Systems (EIIS), pp. 1-4, 2017.

[5] Branson, Steve, Grant Van Horn, Serge Belongie, and Pietro Perona. "Bird species categorization using pose normalized deep convolutional nets." arXiv preprint arXiv: 1406.2952 (2014).

[6] Madhuri A. Tayal, Atharva Mangrulkar, Purvashree Waldey and Chitra Dangra. “Bird Identification by Image Recognition.” Helix Vol. 8(6): 4349- 4352

[7] Atanbori, John, Wenting Duan, John Murray, Kofi Appiah, and Patrick Dickinson. "Automatic classification of flying bird species using computer vision techniques." Pattern Recognition Letters (2016): 53-62.

[8] Sprengel, Elias, Martin Jaggi, Yannic Kilcher, Thomas Hofmann. “Audio Based Bird Species Identification using Deep Learning Techniques.” *CLEF* (2016).

[9] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv: 1409.1556 (2014).

(This work is licensed under a Creative Commons Attribution 4.0 International License.)