Development of a Bird species recognition system using deep learning

\*Note: Sub-titles are not captured in Xplore and should not be used

line 1: 1st Given Name Surname   
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 4th Given Name Surname  
line 2: *dept. name of organization*  
*(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 2nd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 5th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 3rd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 6th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

*Abstract*—the problem of knowing the species of birds in a garden during the autumnwatch was expressed and the solution was designed using AI algorithms to define a species of the birds. This paper provides an overview of the related researches that has been carried out in the past as well as it elaborates the architecture used for the detection of bird species. The dataset we received consisted of for thousand images and we did the Exploratory Data Analysis of the data and simultaneously applied data pre-processing techniques to shape the data to be in perfect shape to train the model. After discussing these processes, we also discuss the selection of model architecture to be used to train the model as we selected faster RCNN from the options of the three best object detection models faster RCNN, SSD and YOLO. Finally, the discussion of the future work is conducted and explains a plan for training the model.

Keywords—component, formatting, style, styling, insert (key words)

# Introduction (*Heading 1*)

The system proposed is used for detection of birds in a garden for the watchers and present them with name of the bird in real-time. Whereas, the system is only able to detect the birds using images by using object detection api. However, in similar research birds can also be detected using the fusion of images and acoustic signals generated by them to communicate amongst each other hence it will increase the chances of making a correct prediction. Additionally, to detect foreground vocal activity in the recording, adaptive thresholding is used to process the raw signal. The signal is then examined in order to extract short-term features based on brief overlapping windows. Zero-crossing rate (ZCR), energy, energy entropy, spectral centroid, spectral spread, spectral entropy, spectral flux, spectral roll-off, 13 Mel frequency cepstral coefficients (MFCCs), 12 chroma vectors, and chroma deviation are some of the short-term properties[1]. Moreover in [2], the systems have evolved further, developers have been able to detect and classify the animal also predict the respective species of the input image using two-stage object detection. For the detection stage, they use the ResNet-50 network as their model backbone network. In order to increase the network's responsiveness to small targets without considerably increasing the complexity of the networks, they constructed an attention mechanism and a feature pyramid network. By explicitly modelling how the channel characteristics are interdependent to capture channel-wise interactions and provide discriminatory object identification functions, the feature pyramid improves the characteristic responses. By selecting more discriminatory features based on the attention mechanism, the regression of bounding boxes and the classification of microscopic species have improved yields. To further enhance network performance and manage the region proposal network, they adopted the Focal Loss as the loss function. Moreover, Dense blocks improve network classification accuracy during the classification stage. It is important to notice they created a two-tiered image pyramid from input, which is useful for identifying large animal species. After analysing these two systems the problem entailed was that the research by [2] is a more complex ensemble learning system as it is for a more complex problem than ours hence it will cost considerably more than using conservative AI.

The paper will focus more on the exploratory data analysis of the dataset and also propose hypothetically the best suitable for the furture work to train the model.

# Background

In [3] the BBC, birding and the study of wildlife in the UK are covered through a series of programmes called "Watches," which include Springwatch, Autumnwatch, and Winterwatch. Nevertheless, in addition to its group of naturalists, the programme has begun to rely on another type of birdwatcher: artificial intelligence (AI). Machine learning technologies developed by BBC Research and Development are currently being utilised to picture and identify birds captured by remote cameras located around the country. Now, as part of our work in explainable machine learning, we have designed an interactive tool for recognising garden birds and finding out what a computer truly understands about birdwatching. Our goal with this tool is to find out what a machine knows about birdwatching. Moreover, in the next research [4] they have worked on the same problem and their model was based on the SVM as they tried to solve this problem in 2015 when the models such as faster RCNN, SSD and YOLO were not invented hence they used traditiional machine learning algorithm called Support Vector Machine(SVM) and bayes. They compared the performance of both the models and selected SVM as the difference between the accuracy was not that huge the selection of both models were selected for their problem under the given circumstances as they did not have any other object detection models as today. In [5] the author First, a colour segmentation technique is used to attempt to remove background components and define possible zones in which the bird may be present. Next, the picture is divided into component planes, and normalised colour histograms are produced from these candidate locations for each component plane. Following aggregation processing, the number of histogram intervals is reduced to a given number of bins. A learning system uses the histogram bins as feature vectors to attempt to identify between the various bird species. The segmentation method obtains a 75% accurate segmentation rate based on experimental data from the CUB-200 dataset. Moreover, the categorization rate of bird species varies between 90% and 8% depending on the number of groups considered. The same problem as the previous research is detected with this research as the technology present at the time of this research was not using any of the model architectures present currently they had a limitation of the system performance as the current models are able to give a better performance.

# Methodology

To design an intelligent system that can effectively decide the species of the birds we have been provided with a dataset of four thousand images and our dataset is divided amongst four species of the birds Erithacus rebicula, Periparus ater, Pica pica and Turdus medulla. To analyse the dataset we conduct the exploratory data analysis hereafter referred to as EDA which allows us to use the dataset for training the model for most accurate results.

## Data type

The first phase of EDA is to analyse the kind of data available and plan the next steps. The dataset required to train the concerned model is provided in the form of photos, requiring us to pay attention to the resolution and choose whether the images in the dataset should be in colour or grayscale.

## Data size

The second stage is to determine whether the dataset is sufficient to train the model in the future. However, if the dataset is insufficient, there are several strategies we may use to increase its size, such as data augmentation techniques or adding more data from websites such as Kaggle, given that we are working with photos. To investigate the present dataset, which has enough data consisting of four thousand photographs, one thousand for each of the categories.

## Data Cleaning

At this point, we are certain that the amount of data that is readily available is adequate, so we go on to the process of data pre-processing, which allows us to convert the data into a format that is standardised and devoid of errors. In light of the fact that we are dealing with photographs rather than tabular information, it was necessary for us to check the data type before we could clean it. We were required to check through all of the pictures in order to determine whether or not we have the appropriate images or whether or not there were any anomalies. Following the examination of the dataset consisting of 4,000 photographs, we found 112 images that were classified as outliers. These images were made up of pictures that were not associated with the Concern. Had pictures of birds that were not part of the class, eggs, eggshells, feathers, stones, oscilloscope screenshots, and other corrupted photographs that might have caused the training data to become unusable. We used a smartphone application known as Merlin in order to assess whether or not the birds belonged to the same class. With the use of this programme, we were able to determine whether or not the birds belonged to the same class.

## Check the class distribution

After cleaning the data, we were confronted with a class imbalance due to the fact that each class had a different number of erroneous photographs that were filtered out, totaling 112. This imbalance was caused by the fact that each class had a different number of images that were eliminated. To be more precise, the class of Erithacus rubecula had 16 photographs that did not belong to the class, the class of Periparus ater contained 10 images that were unusable, Pica pica datasets contained 34 unusable images, and the dataset of Turdus merula contained 52 images that were removed.

## Normalising the class distribution

In order to level the playing field between the different courses, we took the same amount of pictures from the Internet and put them in the folders that corresponded to those classes. We went through the GettyImages website and downloaded an equal amount of the photos that were deleted, then placed them in the primary folder.

## Data Tagging

The whole data set was labelled by using the tensor flow object identification API for this particular data set. This data collection has four classes, and in order to label them all manually, we had to do it ourselves. We trained the system on the environment in addition to training it on the images themselves. We believe that if we train the data of the environment, it will lead to an increase in the overall accuracy of the system. The various techniques that were used while labelling the data included labelling all of the unused areas in the images and training both the system and the images themselves.

## Averaging the pixels

After we have normalised the class distribution, the next step is to determine the picture's overall average pixel value. Using the numpy library in Python, we converted all of the images into an array in order to check the average number of pixels in the entire dataset. We then used numpy to calculate the mean average of all of the pixels in each class, and we converted the overall image sizes by using reshape and reshaping the previous images to the new dimensions extracted using the mean average values. Finally, we used numpy to convert the overall image sizes.

As can be seen in Figure 1, the resolution of photographs belonging to the species Erithacus rubecula exhibits a wide range of differences among themselves. Following our examination of the graph, we moved on to the next step of the process, which was to determine the average resolution of all of the images contained within this class by using numpy. We found that the average width of the image was 796.6 pixels and the average height was 926.9 pixels; consequently, we decided to apply an average resolution of 800x930 pixels in RGB to the dataset, which enables the system to produce more accurate results. Moving on, we carried out the same procedures on the graph depicted in figure 2 for the class of Periparus after the results showed that the average width of the resolutions in this class was 744.6 pixels and the average height of the images in this class was calculated to be 957.3 pixels; consequently, based on the results analysis, we reshaped the images to have a resolution of 750 by 960 to get the best results possible..

Chart, scatter chart

Description automatically generated

Figure (Scatter plot of all the images in the class of Erithacus rubecula)

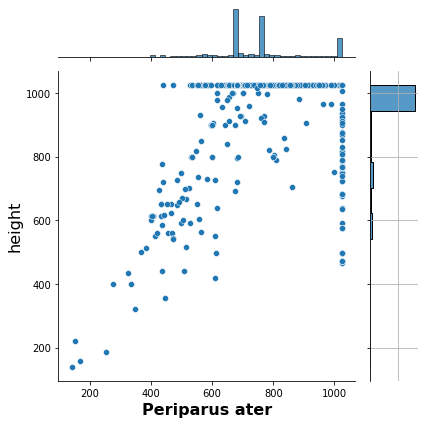


Figure (the scatter plot to view the resolution of all the files in the category of Periparus ater)

Figure 3 demonstrates that the resolution of pictures that belong to the Pica pica class may vary widely from one another in a very wide range. Following our examination of the graph, we moved on to the next step of the process, which was to determine the average resolution of all the images contained within this class by using numpy. We found that the average width of the image was 800.03 pixels, and the average height was 915.3 pixels; consequently, we assigned an average resolution of images to the dataset of 800x920 in RGB, which enables the system to produce more accurate results. Moving on, we carried out the same procedures on the graph depicted in Figure 4 for the class of Turdus merula. The results showed that the average width of the resolutions in this class was 823.9 pixels, and the average height of the images in this class was calculated to be 875.4 pixels; consequently, based on the analysis of the results, we reshaped the images to have a resolution of 830 by 880 in order to get rid of the variation in the image resolutions..

Chart, scatter chart

Description automatically generated

Figure (the scatter plot for all the image resolution in the class of Pica pica)

Chart, scatter chart

Description automatically generated

Figure (this figure displays a scatter plot of resultion of all images in the class of Turdus merula)

## Data augmentation

Data augmentation methods, which enable us to change the existing dataset and increase the total pictures accessible by rotating, rescaling, and flipping the size of the whole training data set, may be used to expand the size of our data site so that it can be used in the training process. We utilised the data picture generator from the keras package in tensorflow to accomplish the data augmentation, and we built the artificially variated image to expand the size of the dataset. Figure 5 depicts the picture in its original state before any augmentation was made, and Figure 6 shows the image after several augmentation methods have been applied to it.



Figure (the original image before implementing augmentation)



Figure (the image after implementing data augmentation techniques)

In order to get these photos with augmentations I went to shampoo then we rescaled the image then shared it by 0.1 then zoom that by 10% followed by a flipping of the image and for the final step we filled the rest of the image with the model off nearest fill. First, we rotated the image by 45 degrees. Next, we shifted the pitch picture width by 10%. Finally, we filled the rest of the image with the model off nearest fill. After performing all of these different actions, the final product can be seen in Figure 6.

## Model training architecture

The model architecture chosen for this system is faster-RCNN as our problems are that the object to be detected is quite small in size and secondly, the input image resolution cannot be fixed to one resolution hence the best possible model architecture available is the the faster RCNN.

# Discussion

There are numerous other applications that may make use of architectures that are similar to the object detection model. SSD, faster-RCNN, and YOLO were the possibilities that were being thought about. As a result, we would want to investigate all of the possibilities, and after doing so, we will defend our decision to go with the quicker RCNN by considering the benefits and drawbacks of each alternative.

## SSD

(Liu et al., 2016) With their study, they show that while these methods are accurate, they are too computationally demanding for embedded systems, and even when using high-end hardware, they are still too sluggish for real-time applications. Detection speed for these methods is often measured in frames per second, and even the method with the highest possible accuracy, called Faster R-CNN, only functions at a rate of seven frames per second (FPS). The SSD is an one shot detector, and since the inferencing time it requires is far less than that required by other designs, such as R-CNN, it is suitable for application in real-time detection. In contrast to previous designs, this one does not permit increasing the picture size while it is being trained; rather, it requires that it be kept constant. The SDD object identification process consists of two steps: first, the feature maps are extracted, and then, convolution filters are applied to the maps in order to identify objects. SSD extracts the feature maps, which are subsequently utilised for object classification. The feature maps are extracted using VGG16 (but alternative base networks may be used). The technique is repeated many times so that it may handle a larger size. This indicates that even if a picture has two dogs, one in the front and one in the background, it is still possible to recognise both of the canine subjects. It provides four forecasts for each individual site. In order to account for scale, SSD makes use of a variety of size filters when predicting class membership. In order to do this, SSD includes six more convolution layers following the VGG16. There will be an additional five of them added for object detection. The detection of tiny objects is the responsibility of feature maps with a higher resolution. The spatial size of the first layer for object detection, conv4 3, is just 38 by 38, which is a rather significant decrease from the original input picture. Because of this, SSD often has poor performance when compared to other detection technologies for detecting tiny things. In SSD, only the layers with a greater resolution are able to detect the presence of tiny things..

## Faster-RCNN

In [7] The following will provide an explanation of the architecture of this model. The Faster R-CNN network is made up of two different components. The first module is a deep fully convolutional network that makes area suggestions, and the subsequent module is a Fast R-CNN detector that makes use of those region suggestions. A Region Proposal Network (RPN) is used by the Faster R-CNN in order to produce a predetermined number of regions. In order to create a collection of region suggestions, the RPN makes advantage of the convolutional features that are provided by the base network. The RPN is programmed to function as a fully convolutional network, which can accurately anticipate object limits objectness scores at each place. The Faster R-CNN design includes a component called the Region Proposal Network (RPN), which is responsible for locating candidate regions by using previously learned characteristics from the base network (ResNet/Inception). The RPN is intended to take the place of the selective search strategy that was used in early R-CNN networks. In these networks, region suggestions were input at the pixel level as opposed to the feature map level. The RPN makes an effort to locate bounding boxes inside the picture, where the individual boxes may have varying dimensions and aspect ratios. The anchors for the fixed bounding boxes may be found in various parts of the picture. Each anchor is represented by a bounding box that is one of nine distinct sizes and has a variable aspect ratio. When making its first predictions on the locations of objects, the RPN refers to them.

## YOLO family

In [8] The following is an explanation of the YOLOv3 architecture. YOLOv3's base network is comprised of its 53 convolutional layers, which are collectively referred to as Darknet-53. While the convolutional layer with a stride of 2 is used to down sample feature maps, no type of pooling is used in this approach. Regardless of the dimensions of the supplied picture, YOLO will provide the same results. In actual use, however, it is best to keep to a predetermined size for the picture. The picture is resampled at a lower resolution by the network by a factor equal to the stride of the network. The height and breadth of the newly created feature map using this kernel are the same as those of the previously produced feature map. It has detecting capabilities along the depth, as was previously mentioned. The YOLO v3 algorithm generates predictions at three different scales, which are obtained by dividing the original image's dimensions by 32, 16, and 8. If you use a stride that is 32 by 32, you will end up with a layer that is 13 by 13, which is enough for the identification of bigger things. A stride size of 8 by 8 will give you a layer size of 52 by 52, which is suitable for identifying tiny items since it is more sensitive. The layers 82, 94, and 106 are where the forecasts are made. The YOLO versions 1 and 2 both have the drawback of being unable to identify things at varying scales, which is something that the three distinct layers aim to remedy. The up sampled layers concatenate with the preceding layers, which helps to maintain the fine-grained characteristics. This is accomplished by raising the stride, which in turn increases the number of cells in the grid. In other words, you are not just relying on the feature map that corresponds to that specific aspect ratio or size; rather, you are also including the characteristics that correspond to other aspect ratios or sizes in order to assist in the identification of objects. This is especially helpful in situations where it is often difficult to identify extremely tiny things using very large features. The YOLO algorithm makes predictions for boxes on three distinct scales. The number of boxes that may be predicted for an image with a resolution of 416 by 416 pixels is 10,647 (13x13x3 + 26x26x3 + 52x52x3). Because it is obvious that not all of these boxes have any practical use, YOLO makes use of NMS in order to cut down on redundant detections and numerous detections of the same photos.

The faster-RCNN has given us the option of using this model architecture to achieve an exceptionally high level of accuracy for the system. This was necessary for our problem, which involves the recognition of patterns on birds. In addition to this, it enables us to adjust the resolution of the input photos to suit our needs, which is very helpful given the wide range of resolutions included in our dataset. However, despite the fact that YOLO v3 is one of the best two stage detectors currently available and gives the best accuracy of Intersection Over Union (IOU) of 0.5, it is not able to identify the birds of different species because it is only able to determine whether or not the object in question is a bird due to the use of linear regression. Because of this issue with the architecture, we decided not to take it into consideration any further.

# Future work

Using EDA and data pre-processing techniques, we have prepared the data that will be used to train the model as the first stage in developing an AI system for identifying the species of birds. In the future, we want to train the model using the previously described faster-RCNN model architecture as justified earlier. However, we cannot rely exclusively on the performance of a single model since we cannot compare the findings to determine whether we have the greatest performance available. Consequently, the second model architecture we will employ is YOLOv3. To end our future work, we will construct each of these models individually and present the results to substantiate our model architecture choice.

##### Acknowledgment

I would like to give acknowledgement to my professors Dr. Paul Fergus and Dr. Carl Chalmers as they have been highly supportive during the time of writing this report and always nudged me to the right direction for my research.

##### References

[1] N. I. Hasan, ‘Bird Species Classification And Acoustic Features Selection Based on Distributed Neural Network with Two Stage Windowing of Short-Term Features’. arXiv, Jan. 01, 2022. doi: 10.48550/arXiv.2201.00124.

[2] C. C. Ukwuoma *et al.*, ‘Animal species detection and classification framework based on modified multi-scale attention mechanism and feature pyramid network’, *Scientific African*, vol. 16, p. e01151, Jul. 2022, doi: 10.1016/j.sciaf.2022.e01151.

[3] Ben Hughes, ‘A machine’s guide to birdwatching’, *BBC R&D*, 2021. https://www.bbc.co.uk/rd/blog/2021-05-a-machines-guide-to-birdwatching (accessed Nov. 18, 2022).

[4] John Atanbori, Wenting Duan, John Murray, Kofi Appiah, and Patrick Dickinson, ‘Automatic classification of flying bird species using computer vision techniques | Elsevier Enhanced Reader’, 2015. https://reader.elsevier.com/reader/sd/pii/S0167865515002743?token=295D21BF83391B008CD0A5F5551B3DCDFD6822A5D6E7616974B79BB1DA57C1B47DCBFF9754C5119B95B44A57D094CE32&originRegion=eu-west-1&originCreation=20221202201002 (accessed Dec. 02, 2022).

[5] A. Marini, J. Facon, and A. L. Koerich, ‘Bird Species Classification Based on Color Features’, in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2013, pp. 4336–4341. doi: 10.1109/SMC.2013.740.

[6] W. Liu *et al.*, ‘SSD: Single Shot MultiBox Detector’, in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 21–37.

[7] S. Ren, K. He, R. Girshick, and J. Sun, ‘Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks’. arXiv, Jan. 06, 2016. Accessed: Dec. 03, 2022. [Online]. Available: http://arxiv.org/abs/1506.01497

[8] J. Redmon and A. Farhadi, ‘YOLOv3: An Incremental Improvement’. arXiv, Apr. 08, 2018. Accessed: Dec. 03, 2022. [Online]. Available: http://arxiv.org/abs/1804.02767