# Large Scale Bird Species Classification Using Convolutional Neural Network with Sparse Regularization

M. Muazin Hilal Hasibuan, Novanto Yudistira(✉), and Randy Cahya Wihandika

Informatics Department, Faculty of Computer Science, Brawijaya University, Jalan Veteran 8, 65145 Malang, Indonesia
yudistira@ub.ac.id

**Abstract.** Bird is one of many creatures with so many species worldwide. Every bird species has many differences, from the shape of its limbs, behavior, and food. Sometimes some scientists have difficulty when making their observations. To this end, accurate artificial intelligence system using deep learning has to be developed to help scientists detect the existence of certain birds automatically. Convolutional Neural Networks (CNN) can help classify the species of birds based on their characteristics. Nevertheless, to train a CNN that can classify the species of birds correctly, it requires a large dataset. 285-birds is a suitable dataset consisting of 43780 digital images with 285 class labels. Bird classification training utilizes a Transfer-learning method using pre-trained weights on ImageNet such as AlexNet and Resnet34. In addition, the hyper-parameters of training of 100 epochs, an Adam optimizer with a learning rate of 0.00001, a batch size of 64, and a Cross-Entropy loss. Utilizing a Sparse regularization in loss function improves the performance model by reducing unnecessary features while also focussing on the important ones. The result of this research is that ResNet model with sparse regularization can recognizes large number of wild birds with the most robust performance compared to other models and thus we suggest our proposed methods to be applied in large scale birds recognition systems.

**Keywords:** Bird Classification · Convolutional Neural Network · Transfer Learning · Sparse Regularization

## 1 Introduction

Birds are divided into 29 orders of 158 families, one of the vertebrate animal classes. Birds belong to the Aves class, subPhylum Vertebrata, and belong to the Phylum Chordata, which is descended from two-legged animals. Birds are warm-blooded and reproduce through eggs. Its body is covered with feathers and has various adaptations for flight. Birds have a fast exchange of substances because flying requires a lot of energy. His body temperature is high and fixed, so he needs a lot of food [1].

Birds are one the animals that have many variations of species, each type of bird has many differences, ranging from the shape of its limbs, behavior, and food to sound.


Y. A. Yusran et al. (Eds.): BIC 2022, AEBMR 235, pp. 651–663, 2023.
https://doi.org/10.2991/978-94-6463-140-1_65

Therefore, scientists often also have difficulty in making observations in nature. For example, to find out what bird species exist in an area, they must be present and explore every corner [2]. Sometimes their presence in the place for a long time disturbs the existing birds, and the birds even leave the place before they can be observed. One way to detect birds in an area without disturbing the presence of birds is to use a tool. For example, scientists can use a camera to take pictures of the surrounding environment or a voice recorder to record the birds' sounds. The next problem is that if we use a camera, then the system that we install must also be able to recognize that a type of bird belongs to a particular species of bird. For this reason, it is necessary to build a system using an algorithm such as the Convolutional Neural Network to identify the bird species correctly [2].

Based on the description, this research will use the CNN method and Transfer-learning to classify large-scale bird species. Convolutional Neural Network (CNN) takes advantage of the convolution process by moving a convolutional kernel (filter) of a specific size to an image. Then, the computer gets new representative information by multiplying that part of the image with the filter used. Meanwhile, Transfer-learning is used to utilize a model that has been trained on a dataset to solve other similar problems by using it as a starting point, modifying and updating its parameters so that it fits the new dataset. This study evaluates two CNN architectures to achieve the best results: AlexNet and ResNet34. AlexNet was used because the research conducted by Gong et al. reached 99.99% accuracy. ResNet34 was used because the research conducted by Jiang had an accuracy of 99.34% [3].

## 2 Related Works

Studies have been conducted to classify the taxonomy of bird species. Raj [4] developed a deep learning model to help birdwatchers identify his 60 species of birds. The researcher collected this dataset using Microsoft's Bing Image Search API v7 to create a random his 80:20 data split. The entire dataset hosts 60 species of his birds and consists of 8218 images. A basic experimental study was performed using the TensorFlow library in the Atom editor on the Windows 10 operating system [4].

The developed CNN architecture is a smaller and more portable version of the VGGNet network. A TensorFlow backend is used to implement this architecture. Stack multiple levels of Convolution and ReLU to learn different attributes. The convolutional layer has 32 filters with $3 \times 3$ feature detectors for the first convolutional block. Application of the ReLU function follows this operation. The pooling layer then contains $3 \times 3$ pools that reduce the spatial dimension from $96 \times 96$ to $32 \times 32$ ($96 \times 96 \times 3$ dimensional images were used to train the network). Another convolutional layer is stacked on top of this, increasing the filter size from 32 to 64, but the feature detector is still $3 \times 3$ dimensional. The ReLU function again follows the convolutional layers.

Also, max pooling is applied and the pool window size is decreased by 2 from 3x3 to 2x2. A final convolutional layer is used, the filter size is increased to 124, and then the ReLU function is implemented. Max Pooling is again applied to Strides of 2 with a pool window size of 2x2. The dropout layer prevents overfitting with a dropout value of 0.25. Then a fully connected layer is added using a dense layer of size 1024. The dropout layer is implemented again with a value of 0.50. Finally, softmax classifiers predict a single class from different mutually exclusive classes [4].

After successfully deploying the CNN model, researcher is ready to train the network on bird images using Keras and the Adam Optimizer. All required packages have been imported into the training script. Matplotlib backend was used to store the numbers in the background. After the CNN finished training, both the model and label binary files were saved to the local disk. This is because the network needs to be loaded into the framework each time it is tested with images acquired outside of training and a test data set. For data augmentation, the ImageDataGenerator class significantly increased the variety of information available for model training without collecting new information. This technique also helps avoid overfitting. A common practice is to split the dataset into a training set and a test set when implementing deep learning. A random 80:20 split of the dataset was created using the train-test split function, with his 80% of the data rendered for training and the remaining 20% for testing [4].

Researchers have proposed a method to predict bird species based on images using CNN, one of the most popular deep learning algorithms in image classification. Raj (2020) built the entire CNN model from scratch, provided training, and finally tested its effectiveness. The developed application produces a high accuracy of 93.19% on the training set and 84.91% on the test set [4].

This research is also inspired by Covid-19 detection research. This is important for early identification of suspected Covid-19 patients so that further action can be taken [5]. Her one way to detect it is by taking an x-ray of her lungs. However, in addition to the need for an algorithmic model that can produce high accuracy, it requires simple computations to apply to the detector. Deep CNN models can detect accurately, but tend to require significant memory usage. CNNs with fewer parameters can save storage and memory usage from real-time processing by perception devices and decision-making systems on the cloud. Furthermore, due to its small parameters, CNN is applicable to FPGAs and other hardware with limited storage capacity. To achieve accurate COVID-19 detection in X-ray images with simple computation, researcher propose a small but reliable CNN architecture using a channel shuffling technique called ShuffleNet. In this study, researcher tested and compared the capabilities of ShuffleNet, EfficientNet, and ResNet. This is because it has fewer parameters than popular deep CNNs such as VGGNet and FullConv, which use fully convolutional layers with robust detection capabilities. Using 1125 X-ray images, researcher achieved an accuracy of 86.93% using multiple model parameters, 18.55 times better than EfficientNet and 22.36 times better than ResNet50, with a 5-fold crossover for Covid-19, pneumonia, researcher detected the usual three categories. -inspection. The memory required by each CNN architecture to perform discovery is linearly related to the number of parameters. ShuffleNet only uses 0.646 GB of GPU memory. That is, 0.43 times more than ResNet50, 0.2 times more than EfficientNet, and 0.53 times more than FullConv. Additionally, ShuffleNet performs the fastest detection in 0.0027 s [5].

## 3 Methods and Solutions

### 3.1 285-Birds Dataset

The dataset in this study was obtained from bird images of various types of birds. The data set consists of 43780 images with 285 bird species, and every image has a dimension of 224 by 224 pixels in jpg format. The creator has divided this dataset into three parts, which consist of 40930 train data, 1425 validation data, and 1425 test data.

From Fig. 1 shows an overview of the 285-birds and some of bird's images with several species that will support the research of bird classification.

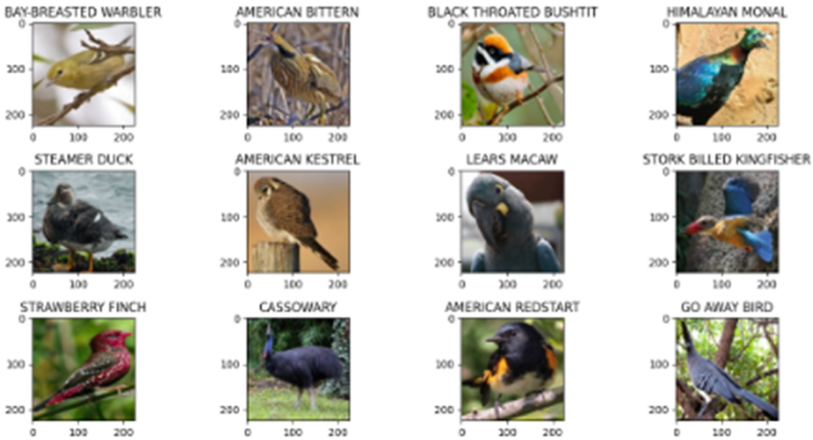


**Fig. 1.** 285-birds sample

### 3.2 Hardware Specification

We trained our model on Intelligent System laboratory server owned by Faculty of Computer Science of Brawijaya University. Our specification of server is NVIDIA QUADRO RTX 8000 48 GB RAM, Intel Xeon Gold 6230R CPU @2.10 GHz, and memory of 250 GB.

### 3.3 Convolutional Neural Network

Convolutional Neural Network (CNN) is the Multilayer Perceptron (MLP) development designed to process two-dimensional data. CNN was developed under NeoCognitron by Kunihiko Fukushima, a researcher from the NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan. Then the concept was finalized by Yann LeChun, a researcher from AT&T Bell Laboratories in Holmdel, New Jersey, USA. LeChun successfully applied the CNN model with the name LetNet in his research on number and handwriting recognition [6]. Furthermore, the CNN method has been proven to outperform other Machine Learning methods such as SVM in the case of object classification in images. In 2012, Alex Krizhevsky, with his CNN application, won the ImageNet Large Scale Visual Recognition Challenge 2012.

CNN is divided into two stages, feature learning, and classification. Feature learning is the stage that allows the model to find the representation needed for feature detection. After the representation is found, the data will be classified based on its class. Figure 2 is an overview of the CNN architecture.

Based on Fig. 2, we can conclude that what will enter the input into the convolution layer will then be entered into the pooling layer using the max-pooling method by taking the largest value from that section. Then it is fed through a Flatten Layer, where this layer will reshape the feature into a vector to be used as input from the fully-connected layer. The fully-connected layer is where all activity neurons from the previous layer are connected to neurons in the next layer like an artificial neural network.

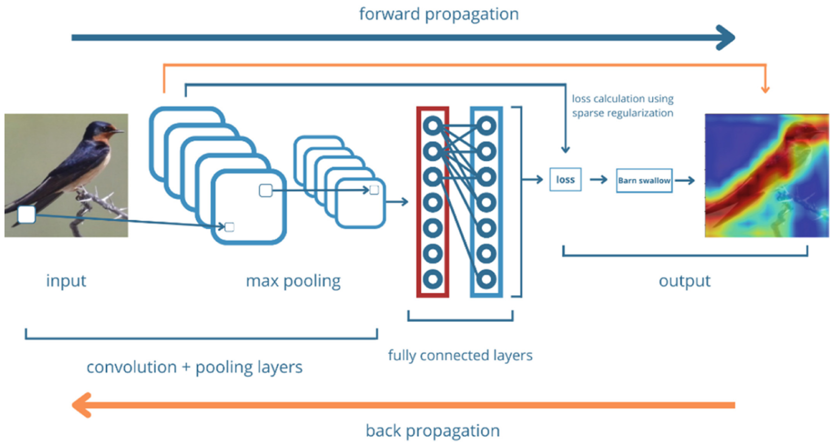


**Fig. 2.** CNN architecture

### 3.3.1 Convolutional Layer

The Convolutional layer is the leading layer in charge of receiving input which will then process the input as an image. The Convolutional process utilizes filters from an image. Filters are dimensions of the image, such as a specific height, width, and thickness. Four hyperparameters can affect the output of the Convolutional layer [7]. The hyperparameters are seen in Table 1. To get a new output based on the previous input image, calculate the dimensions of the feature map using the following Eq. 1.

Based on Table 1 that shows parameters in Convolutional layer where the Filter which smaller than the input data and the type of multiplication applied between a filter-sized patch of the input and the filter is a dot product. A filter must always have the same number of channels as the input, often referred to as Depth. Zero padding which is a large filter shift in convolution. And Stride for the spatial size of the Filter include width or height [8].

$$Output = \frac{W - F + 2P}{s} + 1 \tag{1}$$

**Table 1.** Parameters in Convolutional Layer

| No | Parameter | Object accuracy |
|---|---|---|
| 1 | Filter | The number of zeros added to the image |
| 2 | Depth | Number of filters used |
| 3 | Zero-Padding | Large filter shift in convolution |
| 4 | Stride | Spatial size of the filter (width/height) |

#### 3.3.2 ReLU

ReLU (Rectification Linear Unit) is an activation function that can be used anywhere. Although, in the process, it only changes the negative value to 0. This function is the most popular and practical in the hidden layer process. ReLU is the most frequently used activation function and is the default choice for most neural network layers [8].

ReLU has an almost linear form which, in normal activation functions, cannot take the smoothness of an image, but this is not the case for ReLU. As a result, the ReLU activation function works better than the sigmoid activation function in classification problems [9]. This activation function equation can be seen in Eq. 2.

$$f(x) = 0, z \tag{2}$$

#### 3.3.3 Softmax Activation Function

Softmax function has a function to calculate the probability of each target class over all possible target classes and helps to determine the target class according to the given input. The main advantage of using Softmax is that the output probability ranges from 0 to 1, and the sum of all probabilities will be equal to one [10]. The calculation of the Softmax function can be seen in Eq. 3.

$$Softmax(x_i) = \frac{\exp(x_i)}{\sum_j \exp\left(x_j\right)} \tag{3}$$

#### 3.3.4 Cross Entropy Loss

The artificial neural network architecture can get each optimal weight by using an optimization algorithm to maximize or minimize an objective function. When minimized, it can be referred to as a loss/error/cost function.

Cross-Entropy Loss measures the performance of the model classification, which has an output in the form of a probability value between 0 and 1. The value of the cross-entropy loss will increase if the probability value predicted by the model can be kept far from the actual label. Equation 4 shows the formula for calculating cross-entropy loss.

$$L(y, \hat{y}) = -\sum_{i=1}^{N_c} y_i \log(\widehat{y_i}) \tag{4}$$

From the Eq. 4, it can be explained that L is stands for loss, N is the size of the test set, $y_i$ is the truth label and $\widehat{y_i}$ is the Softmax propability for the $i^{th}$ class.

### 3.3.5 Gradient Descent

Gradient descent is an optimization algorithm used to find the parameter values (coefficients) of a function (f) that minimize the cost function. Gradient descent is best used when the parameters cannot be calculated analytically, such as using linear algebra, and must be searched for with optimization algorithms [11].

### 3.3.6 Sparse Regularization

Sparse regularization, also known as spareness, is one type of regularization method. Spares were first discovered to prevent an increase in the number of parameters of a model. There are several types of spareness. The first is spareness in the weight section, often known as weight decay, spareness at the output or hidden unit output, and the last is spareness in the input of the ReLU activation function proposed by Kurita (2017) [12]. Using spareness on the input of the ReLU activation function can prevent an excessive increase in the value of the output of the ReLU activation function. An excessive increase in the value will cause overfitting. Therefore it is necessary to prevent an increase in the value that is too excessive so that it can have a batch normalization effect and increase the generalization ability of the model [12]. The equation used to add sparse regularization to the ReLU input can be seen in Eq. 5.

$$E = L + \lambda \sum_{k=0}^{n} S(h_k) \tag{5}$$

From Eq. 5, it can be explained that E stands for total error or total loss. L is the result of Cross-Entropy Loss from Eq. 4. $\lambda$ is regularization parameter control. $\sum k$ is the sum of all grids resulting from Activation function output. Finally, $S(h_k)$ is the output of Activation function.

Too many layers in a network can lead to problems like overfitting and the need for network pruning before deploying to production. Then there is also the problem of feature selection. Feature selection searches for a subset of all the important features for modelling. This is usually done as a separate pipeline before modelling starts [13]. Sparse regularization tackles problems like optimizing neural network weights, optimizing the number of neurons for each hidden layer, and selecting features. In neural networks, feature selection can be considered as the pruning of nodes at the input layer, and that is how it is addressed in sparse regularization [13].

Sparse regularization works if all the incoming and outgoing neuron weights are 0, then that node can be removed. Sparse regularization is one way to achieve the sparsity of weights. In this technique, the sum of absolute values of the weights is penalized during training [13].

### 3.4 Transfer Learning

Transfer-learning is a technique or method that utilizes a model that has been trained on a dataset to solve other similar problems by using it as a starting point, modifying, and updating its parameters so that it fits the new dataset [14].

#### 3.4.1 AlexNet

AlexNet is a CNN architecture made by Krizhevsky. AlexNet has eight feature extraction layers. This layer consists of five convolution layers and three pooling layers. In its classification layer, AlexNet has two Fully Connected layers, each of which has 4096 neurons. At the end of the layer, there is a classification into 20 categories using softmax activation. The average accuracy of the classification results reached 85%. In comparison, the identification accuracy has reached 90%, obtained from testing 40 images.

#### 3.4.2 ResNet34

ResNet stands for Residual Network and is a classic neural network. This model was the winner of the 2015 ImageNet Challenge. The fundamental breakthrough for ResNet was the ability to train very deep neural networks with more than 150 layers. Before ResNet training, deep neural networks were complicated because of the gradient loss problem. The Resnet architecture shows that this neural network is easy to optimize and can achieve significantly improved accuracy from the deepest depths [3]. Resnet34 is a ResNet architecture that has 34 layers inside the architecture.

#### 3.4.3 EfficientNet-b0

EfficientNet-b0 is a convolutional neural network trained with over 1 million images from the ImageNet database. Note that the image input size for the network is 224x224 [15]. The network then can categorize images into 1000 object categories, such as keyboards, mice, pencils, and many animals. As a result, the network has learned a rich feature representation of a wide range of images. The neural network, not engineers, designed the EfficientNet-B0 architecture. Engineers created this model using a multi-target neural architecture search that optimizes accuracy and floating-point arithmetic [16].

#### 3.4.4 Fine Tuning

Fine-tuning will train the network with a small learning rate and a reduced number of training periods. In several studies conducted for remote sensing data, Fine-tuning has provided several advantages for pre-trained models from CNN [17].

#### 3.4.5 Freeze Layer

Freeze-layer is an implementation method of Transfer-learning by not updating the weights and bias during the back-propagation process. The freeze can be done at particular layers of the Transfer-learning architecture [18].

### 3.5 Gradient-Weighted Class Activation Mapping (Grad-Cam)

Grad-Cam uses the gradients of each target that flows into the least convolutional layer to produce a bearish localization map, highlighting important regions in the image for concept prediction display [19].

$$w_k^{(c)} = \frac{1}{H.W} \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{\partial Y^{(c)}}{\partial A_k(i,j)} \tag{6}$$

$$L_{GradCAM}^{(c)}(x, y) = ReLU\left(\sum_k w_k^{(c)} A_k(x, y)\right) \tag{7}$$

Equation 6 shows the calculation of $w_k^{(c)}$ where $H$ is the image height and $W$ is the image width. This formula is necessary to summarize the selected layer gradient matrix. Equation 7 multiplies Eq. 6 by the per-channel activations before summing. Finally, ReLU activation returns 0 if the value used is less than 0. This activation removes unnecessary gradients in order to focus only on the most important parts of the gradient map on the image.

## 4 Experimental Results

The training technique used in all cases uses a batch size of 64 and the Adam optimizer with a maximum learning rate of 0.00001 and a weight decay of 0.00001, as well as a gradient clipping of 1 to limit changes in gradient values that are not too large. In addition, data augmentation such as random rotation, random resize crop, and random horizontal flip was also carried out at all stages of the test to increase the amount of data and create more learning models from different images. Data augmentation is excellent to do on image data because it can improve the quality of the data while avoiding overfitting. The normalization process is also carried out because at the testing stage, we will use Transfer-learning with pre-trained models from ImageNet so that the training process can be more efficient.

### 4.1 Evaluation Results

The evaluation carried out on each model is based on the highest accuracy and lowest loss from each epoch, and the value taken is only based on the results of the training validation. Four experimental methods exist for each pre-trained model: Fine-tuning, Freeze-layer, Fine-tuning with Sparse, and Freeze-layer with Sparse.

#### 4.1.1 AlexNet Evaluation Results

In AlexNet, each pre-trained model has four experimental methods: Fine-tuning, Freeze-layer, Fine-tuning with Sparse, and Freeze-layer with Sparse. Table 2 is the result of AlexNet's evaluation.

**Table 2.** AlexNet Evaluation Results

| Method | Epoch | Train accuracy | Train loss | Val accuracy | Val loss |
|---|---|---|---|---|---|
| Fine Tuning | 87 | 83.24% | 4.6859 | 86.45% | 4.6792 |
| Freeze Layer | 92 | 83.50% | **4.6848** | 86.45% | **4.6778** |
| Fine Tuning + Sparse | 82 | 82.63% | 4,6871 | 85.73% | 4.6816 |
| Freeze Layer + Sparse | 90 | **83.76%** | 4.6853 | **87.12%** | 4.6785 |

**Table 3.** ResNet34 Evaluation Results

| Method | Epoch | Train accuracy | Train loss | Val accuracy | Val loss |
|---|---|---|---|---|---|
| Fine Tuning | 76 | 96.36% | 4.6668 | 97.57% | 4.6661 |
| Freeze Layer | 85 | 96.55% | 4.6657 | 97.64% | 4.6657 |
| Fine Tuning + Sparse | 94 | 96.49% | **4.6656** | 97.57% | **4.6651** |
| Freeze Layer + Sparse | 91 | **96.72%** | 4.6660 | **97.98%** | 4.6664 |

Based on Table 2, it can be seen that the Freeze-layer method with Sparse has the highest validation accuracy value. Although the validation loss obtained is not the smallest, the difference between each validation loss is not too far away, so it can be concluded that the Freeze-layer method with Sparse has the best performance compared to other methods.

#### 4.1.2 ResNet34 Evaluation Results

In ResNet34, there are four experimental methods for each pre-trained model: Fine Tuning, Freeze-layer, Fine-tuning with Sparse, and Freeze-layer with Sparse. Table 3 is the result of the evaluation of ResNet34.

Based on Table 3, it can be seen that the Freeze-layer method with Sparse has the highest validation accuracy value. Although the validation loss obtained is not the smallest, the difference between each validation loss is not too far away, so it can be concluded that the Freeze-layer method with Sparse has the best performance compared to other methods.

#### 4.1.3 EfficientNet-b0 Evaluation Results

In EfficientNet-b0, there are four experimental methods for each pre-trained model: Fine Tuning, Freeze-layer, Fine-tuning with Sparse, and Freeze-layer with Sparse. Table 4 is the result of the evaluation of EfficientNet-b0.

Based on Table 4, it can be seen that the Freeze-layer method with Sparse has the highest validation accuracy value. Although the validation loss obtained is not the smallest, the difference between each validation loss is not too far away, so it can be

**Table 4.** EfficientNet-b0 Evaluation Results

| Method | Epoch | Train accuracy | Train loss | Val accuracy | Val loss |
|---|---|---|---|---|---|
| Fine Tuning | 91 | 92.02% | 4.6910 | 95.73% | 4.6730 |
| Freeze Layer | 92 | 92.04% | **4.6809** | 95.80% | 4.6734 |
| Fine Tuning + Sparse | 92 | **92.06%** | 4.6814 | **96.35%** | **4.6721** |
| Freeze Layer + Sparse | 99 | 91.97% | 4.6816 | 96.14% | 4.6734 |

concluded that the Freeze-layer method with Sparse has the best performance compared to other methods.

### 4.2 Grad-Cam Results

Gradient-weighted Class Activation Mapping (Grad-Cam) uses the gradients of any target concept flowing into the final convolutional layer to produce a coarse localization map highlighting the crucial regions in the image for predicting the concept.

Figure 3 shows Grad-CAM results for AlexNet and ResNet34. AlexNet has more minor differences than ResNet34 because ResNet34 has skip connection and the distribution of the gradient became swelling. Then if the models are compared with sparse regularization, ResNet34 makes more differences than AlexNet. Table 3 shows that AlexNet with Freeze layer method using sparse and without sparse, Freeze layer with sparse has the higher accuracy with 87.12% and without sparse only got 86.45%. For ResNet34 with Freeze layer method using sparse and without sparse, Freeze layer with

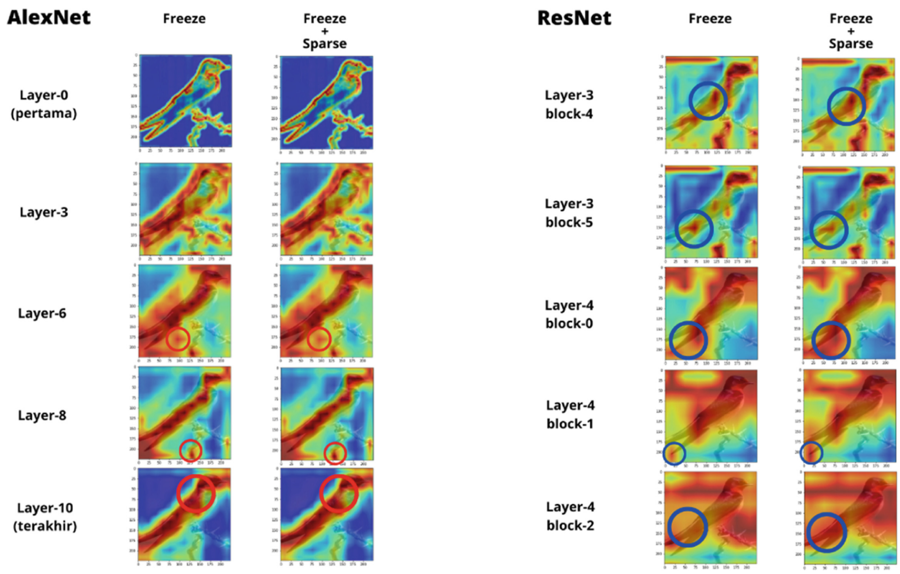


**Fig. 3.** Grad-Cam result of each Models

sparse has the higher accuracy with 97.98% and without sparse got slightly smaller accuracy at 97.57%.

## 5 Conclusion

Based on the implementation and the results of the tests that have been carried out, conclusions can be drawn from this research: AlexNet, ResNet34, and EfficientNet-b0 models have a very significant difference in performance in terms of the resulting accuracy. AlexNet has the highest accuracy with 85% to 87% of each method used, while the ResNet34 model has the highest accuracy of 97% in each method performed, and EfficientNet-b0 has the highest accuracy with 95% to 96% of each method used. Based on the results of the tests, it can be concluded that the ResNet34 model performs better than the AlexNet and EfficientNet-b0 model. Regarding computational time, the AlexNet, ResNet34, and EfficientNet-b0 models have very different training times. AlexNet, with 100 epochs and 64 batch sizes, requires more than 3 h of training time to complete the entire training process, while ResNet34, with the same number of epochs and batch size, takes 8 h, and EfficientNet-b0 takes 4 h. More to complete the entire training process. From these differences, it can be concluded that the AlexNet model is far superior in terms of computational time gain for model training compared to ResNet34 and EfficientNet-b0. Using Sparse in the model could maximizing loss to the best value by reduce the number of less significant parameters.

For the future works this research can be improved by try to use other pre-trained models such as VGG, Densenet, MobileNet, etc.; try to use other optimizers like SGD, RMSProp, and others; and also try the version of the latest bird species dataset, because this dataset is quite updated. For implementation, the trained model can be applied on mobile software application or surveillance camera for animal research or counting existence of wild birds in forest.

## References

1. Suhaerah, Lilis. 2016. “Zoologi Vertebrata.” Bandung: FKIP Unpas.Another reference.
2. Santoso, A. 2018. “Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah.” Surakarta: Universitas Muhammadiyah Surakarta.
3. Jiang, Y., Chen, L., Zhang, H., dan Xiao, X. 2019. “Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module.” PloS One. 14(3), e0214587.
4. Raj, S., Garyali, S., Kumar, S. 2020. “IImage based Bird Species Identification using Convolutional Neural Network.” Department of Computer Science and Engineering.
5. Yudistira, N., Widodo, A.W. and Rahayudi, B. 2020. “Deteksi Covid-19 pada citra sinar-x dada menggunakan deep learning yang efisien.” Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK). 7(6).
6. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D. 1990. “Handwritten digit recognition with a back-propagation network.” Morgan Kaufmann Publishers. 396–404.
7. Arrofiqoh, E. N., Harintaka, H. 2018. “Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi.” Geomatika. 24(2), 61.

8. Ramachandran, P., Zoph, B. and Le, Q.V., 2017. Searching for activation functions. *arXiv preprint* arXiv:1710.05941.
9. Johannes Schmidt-Hieber. 2020. “Nonparametric regression using deep neural networks with ReLU activation function.” Ann. Statist. 48 (4) 1875 - 1897, August 2020.
10. Sofia, N. 2018. “CONVOLUTIONAL NEURAL NETWORK.”
11. Jason Brownlee. 2019. “Gradient Descent For Machine Learning.” Machine Learning Algorithms.
12. Kurita, T. 2017. “Improvement of learning for CNN with ReLU activation by sparse regularization.” Proceedings of the International Joint Conference on Neural Networks. 2684–2691.
13. Simone Scardapane, Danilo Comminiello, Amir Hussain, Aurelio Uncini. 2016. “Group Sparse Regularization for Deep Neural Networks.” arXiv:1607.00485.
14. Sena, S. 2017. “Deep Learning Introduction: Convolutional Neural Network(CNN).”
15. Mingxing Tan and Quoc V. Le. 2019. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” https://arxiv.org/abs/1905.11946.
16. Mingxing Tan, Quoc V. Le. 2020. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” International Conference on Machine Learning.
17. Biserka Petrovska, Tatjana Atanasova-Pacemska, Roberto Corizzo, Paolo Mignone, Petre Lameski and Eftim Zdravevski. 2020. “Aerial Scene Classification through Fine-Tuning with Adaptive Learning Rates and Label Smoothing.” Appl. Sci. 2020, 10(17), 5792.
18. Yan-Mao Huang, Wen-Ren Jong, and Shia-Chung Chen. 2021. “Transfer Learning Applied to Characteristic Prediction of Injection Molded Products.” Polymers (Basel). 13(22): 3874.
19. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D. and Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618–626).