Bird Species Classification with MobilenetV2 for Mobile Vision Application

Research-based □ Application-based ☑

Name	WAN KAR HOU	LIM MING YUEN	CHAN YIH HUI
Programme	CS	CS	CS
ID	1607221	1803655	1603975
Contribution	1/3	1/3	1/3

1. Introduction

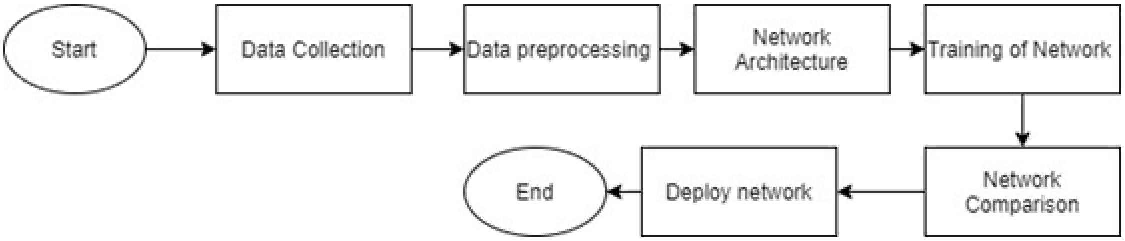
Ornithologists and bird watchers are enthused to find the various colours of beautiful birds and their characteristics of birds. However, people rarely have the knowledge about the birds and have difficulty identifying the types of birds and their characteristics. Even the ornithologists have the identification problem because they need to know all the details of the birds such as biology, distribution and others. Bird classification can be done with picture, video and audio. Picture is more productive than audio and video because audio and video consist of other noise such as insects and wind sounds to interrupt the classification in deep learning. Furthermore, birds can detect other organisms in the natural environment to recognise the environmental changes .(Fagerlund, 2007). The users also are convenient to use applications outdoors.

The dataset used is sub-part of Caltech-UCSD Birds 200(CUS-200-2011). 19 categories were selected from 200 categories of birds to set as our project dataset, challenging parts of identifying bird species are they are similar to each other in appearance. Moreover, there are many features in the images such as colour, head ,body and others. Nowadays, as most of our smartphone allows us to take a high quality of picture, yet an effective bird species classification mobile application could be ideal for ornithologists and bird watchers to predict the bird species to different categories by images. Besides this, for integration of deep learning model into mobile application, a lightweight model with small size but with good accuracy considered suitable for this task. The main objective is to develop a modified light-weight model with smaller size which is MobilenetV2 that could be implemented in IDE such as android studio to develop a mobile vision application that classify the bird species to particular classes with good accuracy.

2. RELATED WORK

Most of the related works are proposed with convolution neural network architecture to classify bird images. The authors presented a bird species classification network with convolution neural networks. They searched the 20 species of birds images on the Internet and merged over 10000 images in total as the dataset. They did the image segmentation depending on RGB and grey scales methods.(Gavali et al., 2019) At the same time, the author also used different convolution neural network to develop bird species classification applications. The images were collected from open sources such as Google's Open Image. It is a mobile application so the author decided to apply Mobile NetV2.Others used the well-known dataset CUS-200-211 for the bird classification applied by MobileNet V2. The authors tried to speed up the training by freezing the network. The training was done on the new output layer. (Gabor , 2019) However, their limitations still remain as the network cannot identify some images due to slightly different characteristics of birds.

3. SYSTEM DESIGN



Data Collection:

In this phase, a dataset was created by selecting 19 categories of birds out of 200 categories of birds from Caltech-UCSD Birds-200-2011 (http://www.vision.caltech.edu/visipedia/CUB-200-2011.html). Several files contained in downloaded Caltech-UCSD Birds-200-2011 which are class.txt, image_class_lables.txt, images.txt, train_test_split.txt and folder contains images for different categories. In order to create dataset of this project, only 19 categories of birds selected from folder which are Crested, Auklet Bobolink, Northern Flicker, American Goldfinch, Evening, Grosbeak California Gull, Anna Hummingbird, Green Jay, Belted Kingfisher, Red breasted Merganser, Mockingbird, Baltimore Oriole, American Pipit, White necked Raven, American Redstart, Cape May Warbler, Golden winged Warbler, Red headed Woodpecker and Cactus Wren. Modification also carry out on all text files such as divided and changes "0" in second column of train_test_split.txt to "0"and "2" equally, reason for this action is to allow dividing datasets to training, validation and testing with ratio of 50%, 25% , 25%. Besides this, other files are also modified based on the 19 categories selected. Eventually, all files and folders of images were saved in a new folder known as "CUB_19_2011" in google drive, with 1118 images in total and around 60 images for each category. Dataset can be obtain from this link (https://drive.google.com/drive/folders/1Gk7nrNwBUp3hOGcTjc18T5REtuH0W_4u?usp=sharing) , by downloading folder named " CUB_19_2011"

Data preprocessing:

To create dataset, a class CUB was created by modifying code from online[3] . Data augmentation was carried out using transformer from pytorch through applying resize, random horizontal flip and normalization. Next, datasets were divided into training dataset, valid dataset and testing dataset with 50%, 25% and 25%. Data loaders were created for each three datasets.

Network Architecture:

Standard MobileNetV2 was initialized from torchvison.models. Modification performed on original MobileNetV2 by adding Global Pooling Layer and changing the classifier layer with the purpose of further reducing model parameters and avoid overfitting issues.

Training of Network Architecture:

Different learning rates, epochs were used to train the model, the combination resulted in satisfied lost value and good accuracy was selected as the best hyperparameter to be used for deep training the model. Model was evaluated based on certain criteria such as validation accuracy, testing accuracy to check for overfitting or underfitting issues. Besides this, evaluation of the model was done based on confusion matrix, Precision, Recall and F1 Score.

Compare Network:

Comparing modified MobileNetV2 with original MobileNetV2 to check the differences in terms of size and accuracy.

Deploy network:

Modified model was converted to TorchScript format for implementation into IDE such as Android Studio to develop mobile vision applications that classify bird species.

4. EXPERIMENT & EVALUATION

Throughout the experiment, the training accuracy of deep train of original Mobile NetV2 managed to reach 100% of training accuracy and modified MobileNetV2 managed to have 99.82% of training accuracy. Next, the validation accuracy of deep train of original Mobile NetV2 only reached 59.27% and modified MobileNetv2 has 54.55%. Hence, we can say that the original MobileNetV2 has higher accuracy than the modified MobileNetV2. Next, the original MobileNetV2 and modified MobileNetV2 are consider overfitting as for each model has low training error and high testing error. However, the testing error does not too high compare to most of the overfitting model as our model the validation accuracy around 60%.

Confusion Matrix gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made. Precision measures the accuracy of positive prediction while recall measures the ratio of positive samples that are correctly detected. For Precision obtained is 0.5455 (54%), which half of the samples predicted by the model is indeed positive while the Recall is 0.5455(54%) which is same with Precision. Next, F1-score for the modified MobileNetV2 is 0.5455(54%) which is average as the 0 is the worst F1-score and 1 is the best F1-score.

In terms of size for our model, we had reduced around 200,000 model parameters of modified MobileNetV2 compared to original MobileNetV2.

5. CONCLUSION

We set up a modified MobileNetV2 by adding Global Pooling Layer and changing the classifier, around 200,000 model parameters lesser than original MobileNetV2 from torchvision.models, it had achieved one of our objectives which is to build a lightweight MobileNetV2 for mobile application. Bigger epochs were required to train the MobileNetV2 in order to achieve satisfied accuracy. However, testing accuracy achieved by modified MobileNetV2 was around 47%, considered low. For future work, bigger datasets would replace current datasets to train the model and achieve greater accuracy.

6. CONTRIBUTIONS

All teammates contributed equally for coding and writing report**.**

List of online sources code incorporated into this system.

- BCNN_for_CUS-200.ipynb [5]
- PyTorch Mobile: Image classification on Android [6]

Module developed by own:

- Modification on MobileNetV2 from torchvision.models by adding Global Pooling Layer and changing the classifier layer.
- Modification performed on the download dataset which is “Caltech-UCSD Birds 200(CUS-200-2011)” to obtain a subpart of it.
- Evaluation on modified MobileNetV2 using Precision, Recall and F1 Score, confusion matrix
- Deep training on original MobileNetV2 and modified MobileNetV2.

7.REFERENCE

[1]Cireşan, D., Meier, U. and Schmidhuber, J., 2012. Multi-column deep neural networks for image classification. arXiv preprint arXiv:1202.2745.

[2]Fagerlund, S., 2007. Bird species recognition using support vector machines. EURASIP Journal on Applied Signal Processing, 2007(1), pp.64-64

[3]Gabor. C., 2019, Active Learning to Solve Class Imbalance in Bird Species Classification. Available from:<http://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26639952.pdf>. [8 April 2021].

[4]Gavali, P., Mhetre, PA., Patil, NC., Bamane, NS., & Buva, HD., 2019, Bird Species Identification using Deep Learning. Available from:<https://www.ijert.org/bird-species-identification-using-deep-learning>. [8 April 2021].

[5]BCNN_for_CUS-200. 2021. Google Colaboratory. [online] Available at: <https://colab.research.google.com/drive/1dawhVZydPfEf7xY2LjQJoXLznbtmltfA?authuser=1&fbclid=IwAR0YLwQLRge1IyEiPZMFkr14jfvv3pKnFtVBakYSA5NTEnyI3FF1CmmUNuo#scrollTo=LHHXYGsJIuSL> [Accessed 11 April 2021].

[6]Olafenwa, J., 2019. *Medium*. [online] Medium. Available at: <https://heartbeat.fritz.ai/pytorch-mobile-image-classification-on-android-5c0cfb774c5b> [Accessed 11 April 2021].