Dog Breed Classification using ResNet and MobileNet

Ruchik Jani
Master of Science in Robotics
Northeastern University
Boston, USA
jani.ru@northeastern.edu

Anuj Patel
Master of Science in Robotics
Northeastern University
Boston, USA
patel.anuj2@northeastern.edu

***Abstract*—In this project, we implemented deep learning to solve the Dog Breed Multi-Class Classification problem. The dataset used was from the Kaggle website with due credit given to the creators of the Stanford Dogs Dataset. We have compared the performance of three neural network architectures on this problem: ResNet18, ResNet50 and MobileNetV2. We have tested the performance of the trained models on different images and videos. Additionally, we have implemented tracking in the videos by using the YOLO object detection model.**

Keywords—ResNet, MobileNet, Classification, Dog Breed, YOLOv3, Object Detection, Tracking

I. Introduction

The Kaggle competition organizers have utilized the Standford Dogs Dataset [1] for one of their prediction competitions. This dataset consists of a total of 20,580 images with 120 dog breed categories. The Kaggle website has modified this dataset for their competition by providing training and test dataset folders, and a CSV file which consists of the labels for the images in the training dataset. The training dataset consists of 10,222 images used in our model training code to save the weights. We have loaded the ResNet18, ResNet50 and MobileNetV2 architectures from the ImageNet database. We have plotted the accuracy and loss curves for both datasets.

Later, we have tested the models on different images and videos. We have also implemented the YOLO object detection model for tracking the dogs while also predicting their breed through the trained models.

II. Related Work

A. Classification

The MobileNetV2 architecture, as described by Sandler et al. [2], is designed for efficient deployment on mobile and embedded devices. The key contributions of the paper include introducing inverted residual blocks, which use linear bottlenecks to improve information flow and reduce computational costs, and proposing model scaling techniques using input resolution and width multipliers. Our project aligns with these concepts by implementing the MobileNetV2 architecture, training and evaluating the model and handling data preprocessing. The project follows the MobileNetV2 design principles outlined in the paper to adapt the architecture for the dog breed identification task.

The concept of residual learning and residual neural networks (ResNets) was introduced by He et al. [3] to address the degradation problem that occurs when training very deep neural networks. The key idea is to use shortcut connections that bypass one or more layers, allowing the network to learn residual functions instead of unreferenced functions. The paper demonstrates that ResNets can be trained to much greater depths than traditional convolutional neural networks (CNNs) without suffering from degradation in performance. This enables the ResNets to achieve better accuracy on image recognition tasks like ImageNet and CIFAR-10. Our project aims to leverage the power of residual learning and deep ResNet architectures for dog breed identification. The ResNet-50 model, which is a deeper variant with 50 layers, has the potential to achieve better accuracy than the shallower ResNet-18 model, which is a shallower variant with 18 layers. We have explored and described this in our project.

B. Detection and Tracking

YOLOv3 [4], which is an improved version of the YOLO (You Only Look Once) object detection system, is the algorithm used in our code for detecting dogs in video frames or images. As described in the paper, YOLOv3 introduces enhancements to previous YOLO versions such as improved bounding box prediction and a more powerful backbone network (Darknet-53). We have implemented dog detection & tracking by loading the pre-trained YOLOv3 weights and configuration. The paper's technical insights inform the dog detection & tracking aspects of the code, which extends YOLOv3's functionality with breed classification.

III. Methods

A. Data Preprocessing

We have downloaded a ‘labels.csv’ file from the Kaggle website [5] which consists of the Image ID and its corresponding label. We have developed 3 different programs for training the ResNet18, 50 and MobileNetV2 models and saving their weights respectively taking reference from the Kaggle competition [6]. The code reads this CSV file and encodes categorical labels (dog breeds) into numerical values. It then constructs a dictionary mapping these encoded labels back to their original breed names while adjusting image file paths and removing redundant columns. Subsequently, there is a function which partitions the dataset into training (9710 images, 95%) and validation sets (512 images, 5%) via stratified sampling to ensure balanced class distribution. For training, it applies random transformations like resizing, rotation, and flipping to augment data and enhance model generalization. Meanwhile, for validation, it applies standard transformations without augmentation.

B. Model Training

The neural network architecture for all 3 models is different and has been defined considering the base model. See Fig. 1. for MobileNet V2 architecture as an example. The neural network model is initialized with the specified number of output classes. Here, we have used all the 120 classes. During training, the base model parameters are frozen, and additional fully connected layers are added for classification. This is essentially transfer learning. The chosen Loss Function, Cross Entropy Loss, and Optimizer, Adam, are defined to optimize the model's parameters. The training loop

iterates over the specified number of epochs, where for each epoch, the model is trained on the training dataset and evaluated on the validation dataset. We have trained the models on 20 epochs. We selected this to avoid overfitting the model. Training and validation losses are computed, and the model's performance metrics, including accuracy and training time per epoch, are recorded. The process concludes with the trained model and optimizer being saved to disk for further testing. Additionally, the training progress is visualized through plots showing the training and validation loss curves, as well as the training and validation accuracy curves over each epoch.

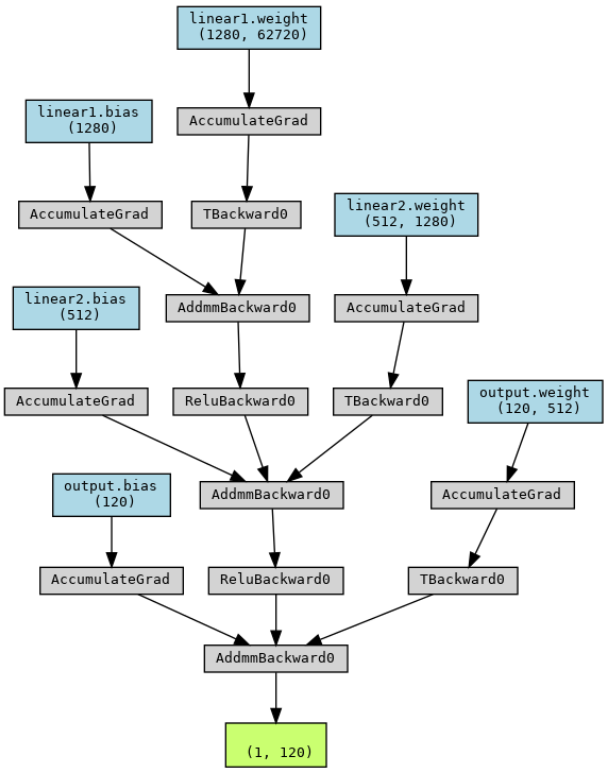

Fig.1. Custom model architecture with MobileNetV2 as base

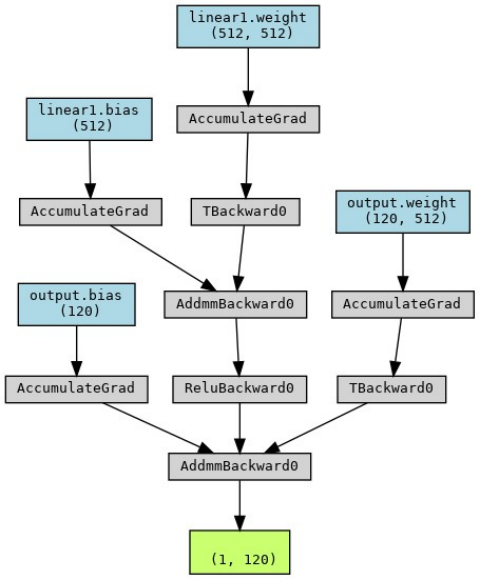

Fig.2. Custom model architecture with ResNet 18 as base

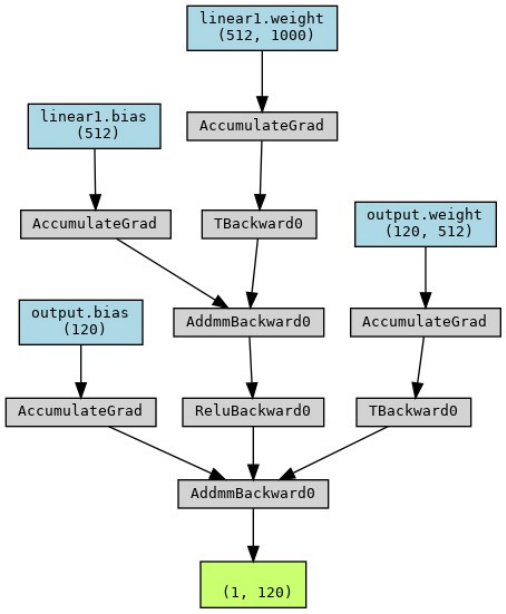

Fig.3. Custom model architecture with ResNet 50 as base

To compare the performance of all the networks we have kept the parameters constant in all 3 models like learning rate (0.0003), loss function, optimizer, number of classes, batch size (64), and number of epochs.

C. YOLO Object Detection and Tracking

We have developed a separate program for the YOLO object detection for testing the 3 trained models on different images and videos. There are 3 different testing codes which consist of all essential functions for detecting and classifying dog breeds in images and videos. We initialize a pre-trained model for feature extraction and construct a custom classification model using it as a base. We then load the YOLOv3 model for object detection and tracking. Image transformations are computed for consistent input formatting.

The code iteratively processes frames or images, providing breed predictions and confidence levels. The user is given a choice for selecting the input: image or video. The code will process all the images present in the directory provided in the code. It will process a single video wherever it is located.

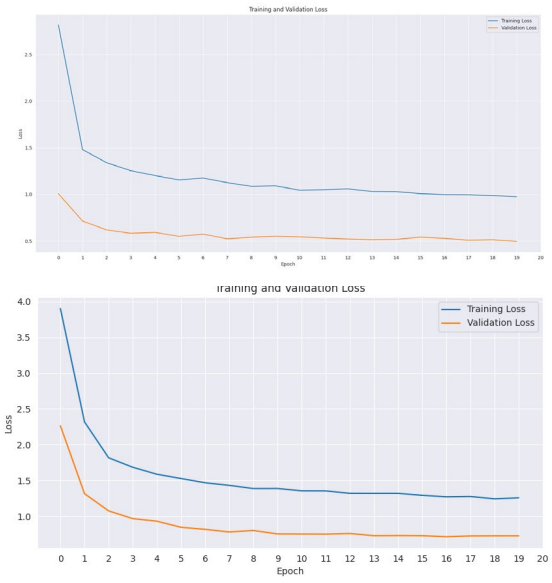

Fig. 4. Training and Validation Loss curves for MobileNetV2, ResNet18 and ResNet50

IV. Experiments And Results

A. Experiment-1: Training Loss and Validation Loss

In this experiment, we will compare the average training and validation loss of the model across 20 epochs and the loss characteristics of all 3 models. From Fig. 2, we can observe that for all 3 models, the training loss decreases rapidly within the first 2 epochs and then stabilizes at 1-1.25. The validation curve follows the same trajectory however it starts at a low initial value and stabilizes at approximately 0.5. The loss curves fluctuate a lot for the MobileNet V2 model, whereas, for ResNet 18 and 50 models the curve is much smoother with minor fluctuations here and there. The average loss, accuracy and training time for each model is captured in Table I.

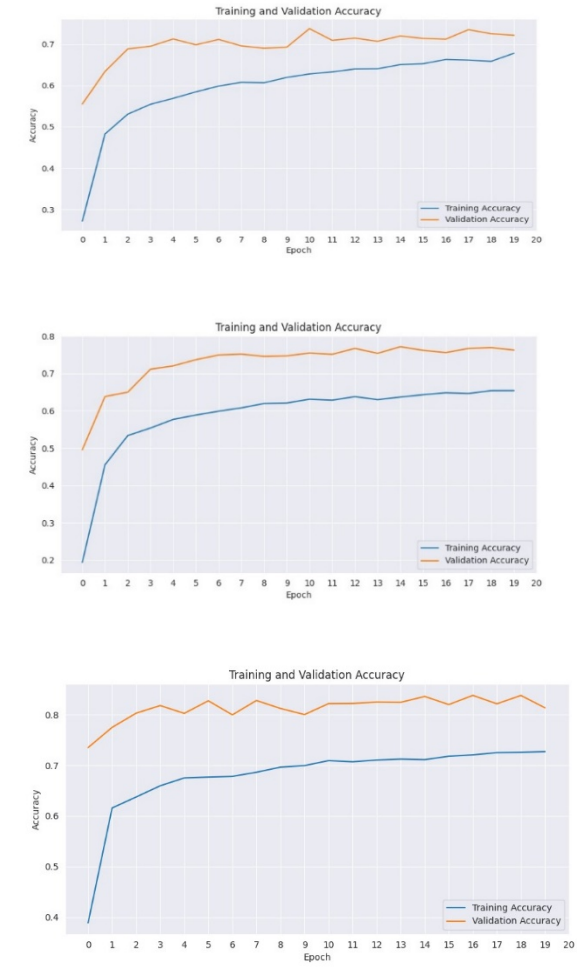

Fig. 5. Training and Validation Accuracy curves for MobileNetV2, ResNet18 and ResNet50

B. Experiment-2: Training Accuracy and Validation Accuracy

In this experiment, we will compare the average training and validation accuracy of the model across 20 epochs and the loss characteristics of all 3 models. The training accuracy increases rapidly within the first 2 epochs and then stabilizes at 65-75%. The validation curve follows the same trajectory however it starts at a high initial value and stabilizes at approximately 70-80%. The training curves for all 3 models are mostly flat with very minor spikes. The validation curves fluctuate a lot for the MobileNet and Res-50 models, whereas, for the ResNet 18 model the curve is much smoother.

C. Experiment-3: Training Time

In this experiment, we are comparing the average training time and the range of training times for each model. The ResNet 18 model is the fastest model with a minimum training time of 39.28s and a maximum of 43.02s. The MobileNet V2 model takes slightly longer to train with a minimum of 46.76s and a maximum of 50.37s. The ResNet 50 model is the slowest model with a minimum of 70.81s and a maximum of 82.74s.

TABLE I
TABULAR COMPARISON OF DATA OBTAINED FROM MODEL TRAINING

Model	Average Values of Training Parameters				
	Training Accuracy	***Validation Accuracy***	***Train Loss***	***Val Loss***	***Training Time***
MobileNetV2	59.6%	69.8%	1.475	0.942	47.89s
ResNet 18	58.8%	72.8%	1.576	0.894	41.11s
ResNet 50	***67.9%***	***81.3%***	***1.195***	***0.573***	***78.45s***

Based on the above table and the observations from the first 3 experiments, we can conclude that ResNet 50 is the best model w.r.t. all parameters except for the training time.

Fig. 6. Example of an image successfully classified with a confidence score and bounding box

D. Experiment-4: Evaluation of Model on Images

In this experiment, we are going to compare the performance of the models on a test dataset consisting of images. We have tested the model on 10 dog images belonging to different breeds. The YOLOv3 model was successful since all the images were displayed with bounding

boxes and the confidence score was displayed based on the classification model used for breed prediction. There are two tables which we will use for this experiment.

In Table II, the dog breeds along with their corresponding confidence scores have been displayed against the classification model. In Table III, the misclassification labels have been highlighted in red against the corresponding dog breed. The confidence score represents the level of certainty or probability assigned by a model to its prediction for a particular class. The code shows a label even when there is no dog image.

TABLE II.
TABULAR COMPARISON OF CONFIDENCE SCORES OF DIFFERENT CLASSES ON IMAGE DATA

Dog Breed	**ResNet18**	**MobileNetV2**	**ResNet50**
Chihuahua	90.55%	73.66%	62.48%
Lhasa	49.98%	63.05%	93.08%
Japanese Spaniel	98.87%	99.01%	92.17%
Boston Bull	87.32%	85.96%	98.75%
Airedale	79.07%	62.97%	67.92%
Sheepdog (Collie)	61.76%	53.51%	86.82%
Doberman	99.91%	99.95%	99.95%
Vizsla	98.23%	95.90%	99.96%
Rottweiler	98.46%	91.98%	99.13%
Malinois	48.81%	91.06%	83.76%

TABLE III.
TABULAR REPRESENTATION OF MISCLASSIFICATION ON IMAGE DATA

Dog Breed	**ResNet18**	**MobileNetV2**	**ResNet50**
Chihuahua	*Pembrooke*	Chihuahua	Chihuahua
Lhasa	*Pug*	*Pug*	*Pug*
Japanese Spaniel	Japanese Spaniel	Japanese Spaniel	Japanese Spaniel
Boston Bull	Boston Bull	*French Bulldog*	boston_bull
Airedale	*Wire_Haired_Fox_Terrier*	*Whippet*	*Wire_Haired_Fox_Terrier*
Sheepdog (Collie)	Collie	Collie	Collie
Doberman	*Black-and-Tan Coonhound*	*Black-and-Tan Coonhound*	Doberman
Vizsla	Vizsla	Vizsla	Vizsla
Rottweiler	Rottweiler	*Brabancon Griffon*	Rottweiler
Malinois	*Bloodhound*	*Brabancon Griffon*	Malinois

From both the tables, it is clear that ResNet 50 is by far the best-performing model since its accuracy on the test set is 80%. The accuracy of ResNet 18 is 50% and that of MobileNet V2 is 40% on the test set. One thing to be noted is that even though ResNet 50 has a high accuracy, it wrongly classifies the 'Lhasa' breed with a confidence score of 93.08% and the 'Airedale' breed with a confidence score of 67.92%.

Fig. 7. Example of dog detection on video using YOLOv3 along with successful classification

E. Experiment-5: Evaluation of Model on Videos

In this experiment, the goal is the same as that of model evaluation on images. In this case, we have observed that the tracking accuracy of the YOLOv3 model is pretty good. It almost always detects the dogs except for a case when the dog is white which matches the background colour. The classification accuracy is great when there is a single dog in the video frame. We have chosen not to display more than one prediction since the results were dissatisfactory. Tables IV and V show the confidence scores and misclassification results as before.

TABLE IV.
TABULAR COMPARISON OF CONFIDENCE SCORES OF DIFFERENT CLASSES ON VIDEO DATA

Dog Breed	**ResNet18**	**MobileNetV2**	**ResNet50**
Golden Retriever	41.16%	95.41%	92.87%
German Shepherd	85.45%	98.08%	88.07%
Labrador Retriever	69.24%	84.73%	96.83%
Siberian Husky	65.41%	77.80%	66.17%
English Bulldog	90.88%	95.43%	72.81%
Rottweiler	95.37%	97.54%	93.38%
Pomeranian	86.48%	70.90%	93.82%
Chihuahua	99.79%	59.17%	70.66%
Pug	98.32%	81.32%	98.80%
Beagle	78.51%	62.84%	96.02%
French Bulldog	98.98%	99.93%	79.60%

American Pitbull Terrier	75.26%	93.51%	95.76%
Australian Shepherd Dog	46.05%	62.29%	64.08%
Pembroke Welsh Corgi	98.35%	86.71%	99.85%
Border Collie	80.93%	82.73%	87.17%

We have implemented this breed prediction and detection algorithm on a test video which we randomly downloaded from YouTube. We have tested on 15 different dog breeds taken from the video. The prediction accuracy is also dependent on the tracking. When the images zoom in and move, the confidence score decreases.

TABLE V.
TABULAR REPRESENTATION OF MISCLASSIFICATION ON VIDEO DATA

Dog Breed	**ResNet18**	**MobileNetV2**	**ResNet50**
Golden Retriever	Golden Retriever	Golden Retriever	Golden Retriever
German Shepherd	German Shepherd	German Shepherd	German Shepherd
Labrador Retriever	*Great Dane*	Labrador Retriever	Labrador Retriever
Siberian Husky	Siberian Husky	Siberian Husky	Siberian Husky
English Bulldog	*Boxer*	*French Bulldog*	*Bull Mastiff*
Rottweiler	Rottweiler	Rottweiler	Rottweiler
Pomeranian	*Samoyed*	*Samoyed*	Pomeranian
Chihuahua	Chihuahua	Chihuahua	Chihuahua
Pug	Pug	*French Bulldog*	Pug
Beagle	*English Fox Hound*	*Walker Hound*	*English Fox Hound*
French Bulldog	French Bulldog	French Bulldog	French Bulldog
American Pitbull Terrier	*Rhodesian Ridgeback*	*Rhodesian Ridgeback*	*Vizsla*
Australian Shepherd Dog	*Bernese Mountain dog*	*Bernese Mountain dog*	*Bernese Mountain dog*
Pembroke Welsh Corgi	Pembroke	Pembroke	Pembroke
Border Collie	Border Collie	Border Collie	Border Collie

The ResNet 50 model has the highest classification accuracy amongst all the 3 models with 79% on the test video. Both ResNet 18 and MobileNet V2 have 64% accuracy. The test accuracy for ResNet 50 is the same in both videos and images. The test accuracy for the other 2 models has increased in video. There is also an outlier in our test dataset: the 'English Bulldog.' Surprisingly though, all 3 models classify it differently. The 3 different breeds are similar to the outlier with Bull Mastiff being the most genetically similar breed as it was developed through cross-breeding of English Bulldog with English Mastiff. This suggests that the classification model can accurately classify outliers.

V. DISCUSSION AND SUMMARY

A. *Classification*

ResNet 50 outperforms MobileNetV2 and ResNet 18 across various metrics such as training accuracy, validation accuracy, and validation loss. This suggests that ResNet 50 is better suited for the given classification task compared to the other models. ResNet 50 exhibits smoother loss curves with minor fluctuations, indicating better stability during training. In contrast, MobileNetV2 shows higher loss fluctuations, potentially indicating instability or convergence issues. ResNet 50 takes significantly longer to train compared to MobileNetV2 and ResNet 18. While this may not be an issue in all scenarios, it could be a consideration in applications where fast model training is essential.

To enhance the performance of MobileNetV2 and ResNet 18, several strategies can be explored:

- Fine-tuning hyperparameters such as learning rate, batch size, or optimizer settings to better suit the characteristics of each model.
- Experiment with different architectures or variations of MobileNetV2 and ResNet 18 to find configurations that yield better results.
- Regularization techniques such as dropout or weight decay to prevent overfitting and improve generalization.

Overall, while ResNet 50 emerges as the top performer in this study, there is room for further exploration and optimization to maximize the performance of all models across various metrics.

B. *Detection and Tracking*

ResNet 50 is the clear winner as evident from the test dataset evaluation results. ResNet50 achieved an accuracy of 80%, while ResNet18 and MobileNetV2 achieved 50% and 40%, respectively. Despite its high accuracy, it misclassified some breeds with high confidence scores. This suggests that while the model performs well overall, there is still room for improvement, especially for certain breeds that may be more challenging to classify correctly. However, the model is still very accurate since the prediction for 'Airedale' is quite close. The prediction as well as the true label in this case belong to the same family of dogs having similar appearances: 'Terriers.' The ResNet 18 model also gives the same result, whereas, the MobileNet V2 model gives a completely erroneous result. The misclassification for 'Lhasa' can be attributed to the fact that both of them are white-colored. This misclassification is particularly consistent across all the 3 models. These can be solved with further fine-tuning of the model to the misclassified classes. All the other misclassifications from the ResNet 18 and MobileNet V2 models also have similar characteristics: either the breeds are related or look similar, which is true in the majority of the

cases, or they can be completely wrong in a few cases. We obtained similar results when the model was applied to the test video. The prediction accuracy is also dependent on tracking, and the confidence score changes when the bounding box shifts. It changes with the color of the dog w.r.t. the background, when there are multiple objects to be detected, and with changes in object scale, motion, and camera angle.

To improve upon the issues described above, several potential approaches could be explored:

- *Enhanced Data Augmentation:* Increasing the diversity and quantity of training data by applying various augmentation techniques (e.g., rotation, scaling, colour jittering) could help the model generalize better to different scenarios, including challenging cases like dogs with colours matching the background.
- *Ensemble Methods:* Combining multiple models or architectures through ensemble techniques (e.g., model averaging, boosting) could help mitigate individual model weaknesses and enhance overall accuracy and robustness.
- *Multi-Object Detection and Tracking:* Exploring more advanced object detection and tracking algorithms specifically designed for handling multiple objects could improve performance in scenarios with multiple dogs present. We can explore the combination of YOLOv3 with algorithms like Deep SORT (Simple Online and Realtime Tracking) for improving tracking.
- *Incorporating Temporal Information:* For video analysis, incorporating temporal information and leveraging techniques like recurrent neural networks or attention mechanisms could potentially improve tracking accuracy and confidence scores, especially in cases where objects move or change scale.

Overall, the experiments provide valuable insights into the performance of different models and the challenges faced in object detection, tracking, and classification tasks. There is a lot of scope for improvement in our model training process as well as the testing and tracking process. We can explore the various methods described in this section to optimize the performance of all 3 neural networks for the dog-breed prediction task, as well as improve the YOLO tracking accuracy to create a better classifier.

REFERENCES

[1] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao and Li Fei-Fei. Novel dataset for Fine-Grained Image Categorization. First Workshop on Fine-Grained Visual Categorization (FGVC), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

[3] He, K., Zhang, X., Ren, S., & Sun, J. (2015)., "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition

[4] Redmon, J. and Farhadi, A., 1804. Yolov3: an incremental improvement. 2018. arXiv preprint arXiv:1804.02767, 20.

[5] https://www.kaggle.com/c/dog-breed-identification/data

[6] https://www.kaggle.com/code/jackttai/dog-breed-classifier-with-pytorch-using-resnet50