Dog Breed Identification Using Convolutional Neural Networks

HARI KIRAN KEERTHIPATI, University Of North Texas, USA
SASIDHAR YALAMANCHILI, University Of North Texas, USA
VAMSI KRISHNA REDDY BOGASAMUDRAM, University Of North Texas, USA

This report proposes a method for dog breed identification using convolutional neural networks (CNNs). Our approach involves training deep neural networks on a dataset of dog images to learn different features for distinguishing between different dog breeds. We use a combination of data augmentation techniques and transfer learning to improve the performance of our model. This work has important applications in areas such as veterinary medicine, animal behavior studies, and pet breeding.

Additional Key Words and Phrases: cnns, neural networks, resnet50, vgg16, convolution

ACM Reference Format:
Hari Kiran Keerthipati, Sasidhar Yalamanchili, and Vamsi Krishna Reddy Bogasamudram. 2023. Dog Breed Identification Using Convolutional Neural Networks. 1, 1 (May 2023), 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The general project idea is to use convolutional neural networks (CNNs) to develop a system that can accurately identify the breed of a dog from a given image. The problem we are trying to solve is the difficulty and time-consuming process of identifying dog breeds manually. This problem is relevant because there are hundreds of dog breeds, and some of them may look similar to one another, making it challenging for even experienced dog breeders and veterinarians to accurately identify a breed.

Our contribution to solving this problem is to use deep learning techniques such as CNNs to automatically classify dog breeds from images with high accuracy. This will provide a fast and efficient way to identify dog breeds, making it easier for veterinarians, breeders, and dog owners to identify dogs' breeds accurately. Additionally, this project has the potential to contribute to other fields, such as animal behavior studies and pet breeding, where accurate identification of dog breeds is essential. We tried several pre-trained models which are robust and proven to work and using Transfer Learning approaches we adapted these models to our dog image dataset. We also built some of the CNN models on our own and we used pytorch library to build these architectures on our own. But, we didn't get satisfactory results with our own models and we ran into OutOfMemory issues and compute issues when training these models. But, using pre-trained models is the best approach to Computer Vision problems because someone has spent massive amount of computation power and energy to build these models and these models are time-tested to work.

2 RELATED WORKS

Several studies have shown the use of CNNs to solve a wide variety of classification tasks including dog breed identification. Hasbi Ash Shiddieqy et al. (2017) [1] developed two different CNNs with 2 layers and 5 layers and concluded that CNNs with more layers will be able to classify things with much higher accuracy. Bickey Kumar shah et al. (2020) [2] implemented Convolutional Neural Networks with Deep Learning to identify breeds of a dog from an image. P. Prasong and K. Chamnongthai [3] proposed a system to identify dog breeds using size and position of each local part such as left ear, right ear etc. and PCA to identify the breeds of the dogs faster than the conventional approaches.

Shuo Wang [4] used VGG19 pretrained model, and added 2 additional layers at the end and achieved an accuracy of 73% by just using 8,351 dog images of 133 breeds. Clay Mason [5] used Inception V3 model and used transfer learning on the 10,222 dog images of 120 breeds and was able to achieve an accuracy of 77%. Amanjot Singh [6] used Resnet50 and VGG19 and also developed a simple web application where you can select an image and the web application processes the image and shows the breed of the dog as predicted by the model.

3 METHODS

We evaluated the performance of four different CNN models for the task of dog breed identification: VGG16, InceptionV3, ResNet50, and EfficientNet. VGG16 is a deep CNN architecture with 16 layers, and it has been shown to achieve high accuracy on image classification tasks. InceptionV3 is a more recent architecture that uses a combination of convolutional and pooling layers to improve performance while minimizing the number of parameters in the model. ResNet50 is a variant of the ResNet architecture that uses skip connections to improve gradient flow during training and prevent vanishing gradients. EfficientNet is a family of models that achieves state-of-the-art performance on image classification tasks while minimizing the computational cost of training the model. We also built our own models to see if they work for dog breed identification.

Each of the four models was adapted for the dog breed identification task by modifying the final layer to output probabilities for each of the 120 breeds in our dataset. We also applied various data augmentation techniques to increase the size of the dataset and improve the models' generalization performance. Some models were trained using Batch Stochastic Gradient Descent(BSGD), some others were trained using Adam Optimizer and some others using RMSProp . We have used Cross Entropy Loss as a loss function in

Authors' addresses: Hari Kiran Keerthipati, HariKiranKeerthipati@my.unt.edu, University Of North Texas, USA; Sasidhar Yalamanchili, SasidharYalamanchili@my.unt.edu, University Of North Texas, USA; Vamsi Krishna Reddy Bogasamudram, VamsiKrishnaReddyBogasamudram@my.unt.edu, University Of North Texas, USA.

XXXX-XXXX/2023/5-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

all of our models. We have used a batch size of 70 and evaluated the performance of each model using accuracy, classification report, Precision, Recall, F1-Score and Confusion Matrix.

3.1 VGG16

VGG16 is a convolutional neural network architecture that was proposed by Simonyan and Zisserman in 2014. The model consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. The convolutional layers are arranged in groups of 2-3, with max pooling layers in between. The final layer is a softmax layer that outputs a probability distribution over the 1000 classes in the ImageNet dataset, which was the original task for which VGG16 was designed.

For our dog breed identification task, VGG16 was adapted for the dog breed identification task by modifying the final layer to output probabilities for each of the 120 breeds in our dataset. The dataset was split into 6517 training, 1533 validation, and 2172 testing images. We applied data augmentation techniques such as random rotation, horizontal flipping, and random cropping to increase the size of the dataset and improve the model's generalization performance. The model was trained using the RMSProp optimizer with a learning rate of 0.01 for 150 epochs and a batch size of 100.

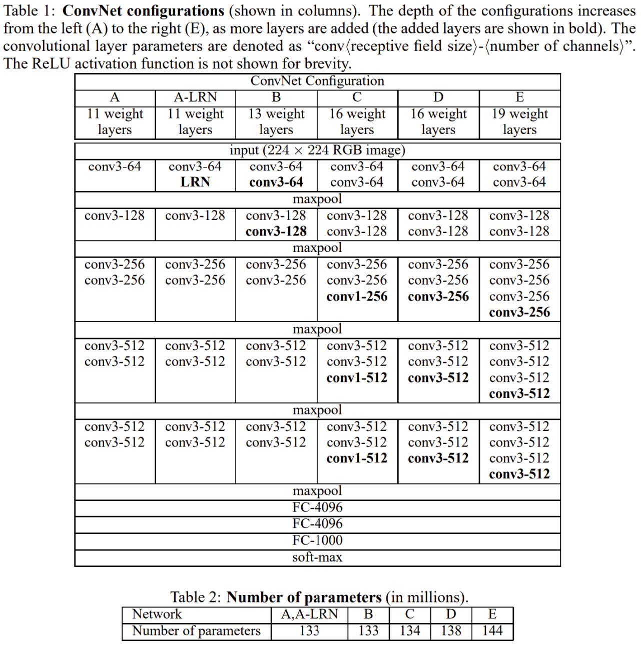


Table 1: **ConvNet configurations** (shown in columns). The depth of the configurations increases from the left (A) to the right (E), as more layers are added (the added layers are shown in bold). The convolutional layer parameters are denoted as "conv⟨receptive field size⟩-⟨number of channels⟩". The ReLU activation function is not shown for brevity.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

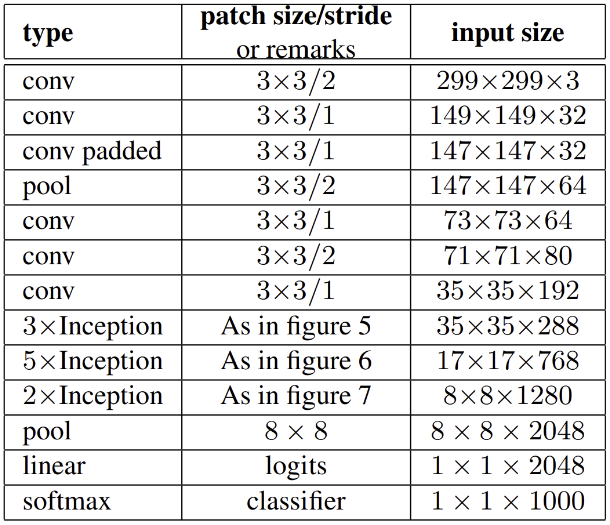
Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Fig. 1. VGG16 Architecture & Total Parametes

3.2 Inception V3

Inception V3 is a convolutional neural network architecture that was proposed by Szegedy et al. in 2015. The model consists of multiple branches of convolutional layers with different kernel sizes and pooling strategies, which are then concatenated to produce a single feature map. This design allows the model to capture both fine-grained and coarse-grained features from the input images.

For our dog breed identification task, we adapted the Inception V3 model by modifying the final layer to output probabilities for each of the 120 breeds in our dataset. The dataset was split into 6517 training, 1533 validation, and 2172 testing images. We applied data augmentation techniques such as random rotation, horizontal flipping, and random cropping to increase the size of the dataset and improve the model's generalization performance. The model was trained using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01 for 50 epochs and a batch size of 70.



type	**patch size/stride or remarks**	**input size**
conv	3×3/2	299×299×3
conv	3×3/1	149×149×32
conv padded	3×3/1	147×147×32
pool	3×3/2	147×147×64
conv	3×3/1	73×73×64
conv	3×3/2	71×71×80
conv	3×3/1	35×35×192
3×Inception	As in figure 5	35×35×288
5×Inception	As in figure 6	17×17×768
2×Inception	As in figure 7	8×8×1280
pool	8 × 8	8 × 8 × 2048
linear	logits	1 × 1 × 2048
softmax	classifier	1 × 1 × 1000

Fig. 2. Inception V3 Architecture

3.3 ResNet50

ResNet50 is a convolutional neural network architecture that was proposed by He et al. in 2016. The model consists of 50 layers and utilizes a residual block design, which allows for better propagation of gradients and reduces the vanishing gradient problem. The residual blocks contain skip connections that enable the gradient to flow directly from the input to the output of the block, improving the model's ability to learn complex features from the input images.

For our dog breed identification task, we adapted the ResNet50 model by modifying the final layer to output probabilities for each of the 120 breeds in our dataset. The dataset was split into 6517 training, 1533 validation, and 2172 testing images. We applied data augmentation techniques such as random rotation, horizontal flipping, and random cropping to increase the size of the dataset and improve the model's generalization performance. The model was trained using the Adam optimizer with a learning rate of 0.01 for 60 epochs and a batch size of 100.

3.4 EfficientNetB0

EfficientNet is a family of convolutional neural network architectures that was proposed by Tan et al. in 2019. The models utilize a

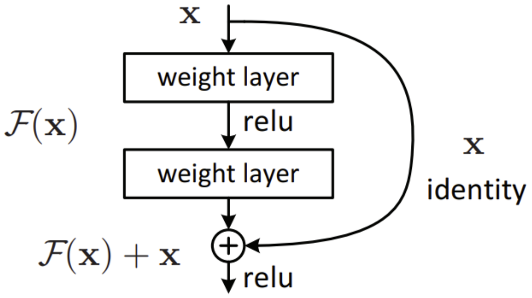
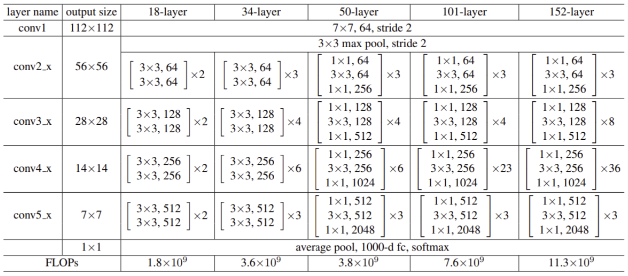

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 3. Residual Block of a Resnet



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 4. Architecture of Resnet50

novel compound scaling method that optimizes the depth, width, and resolution of the network simultaneously, resulting in better performance with fewer parameters. The architecture also includes a set of efficient blocks that utilize mobile inverted bottleneck convolutional layers to reduce the computational cost of the model.

For our dog breed identification task, we adapted the EfficientNetB0 model by modifying the final layer to output probabilities for each of the 120 breeds in our dataset. The dataset was split into 6517 training, 1533 validation, and 2172 testing images. We applied data augmentation techniques such as random rotation, horizontal flipping, and random cropping to increase the size of the dataset and improve the model's generalization performance. The model was trained using the Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01 for 70 epochs and a batch size of 100.

3.5 MyCNN

In addition to the pre-trained models, we also developed our own CNN architecture called MyCNN. MyCNN is a relatively shallow network, consisting of three convolutional layers followed by three fully connected layers. The first three layers of the network are convolutional layers with 3x3 filters and 96, 128, and 64 output channels, respectively. The convolutional layers are followed by batch normalization layers to improve the stability of the network during training. ReLU activation functions are used after each convolutional layer to introduce non-linearity into the network. After the convolutional layers, the network has max-pooling layers that downsample the feature maps. At the final fc layer we will pass the num of classes that we want as output.

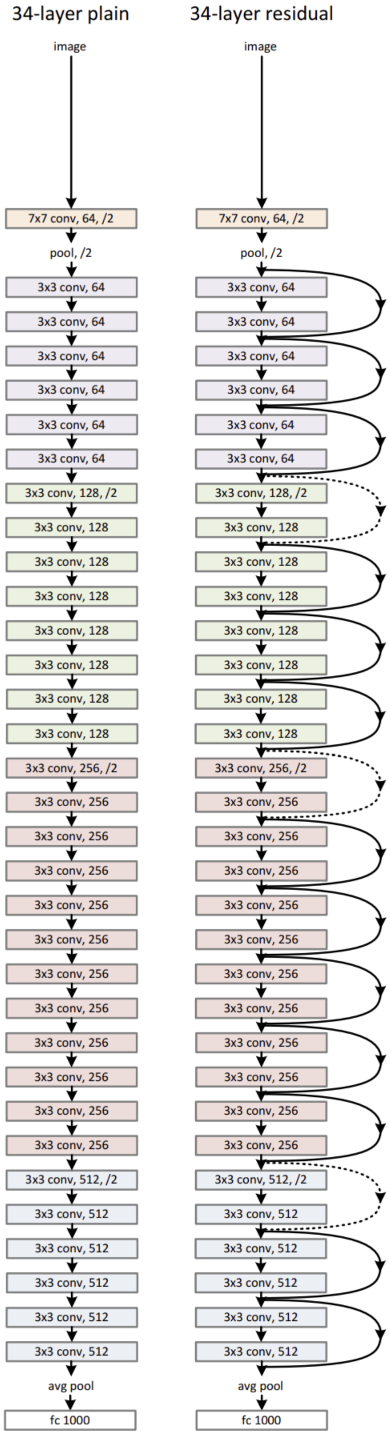


Fig. 5. Plain Network Vs. Residual Network

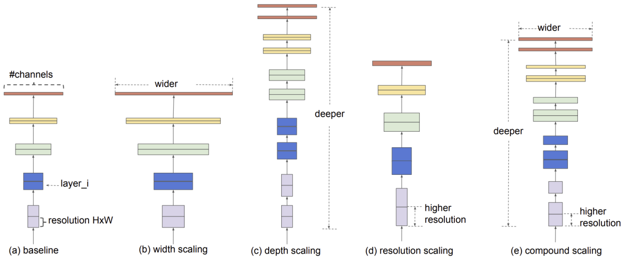


Fig. 6. Model Scaling in Others Vs. EfficientNet

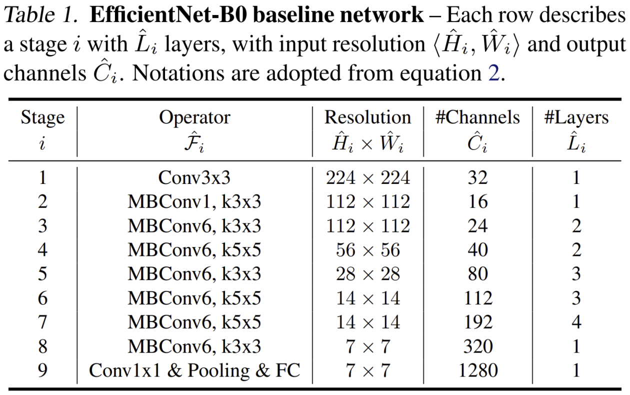
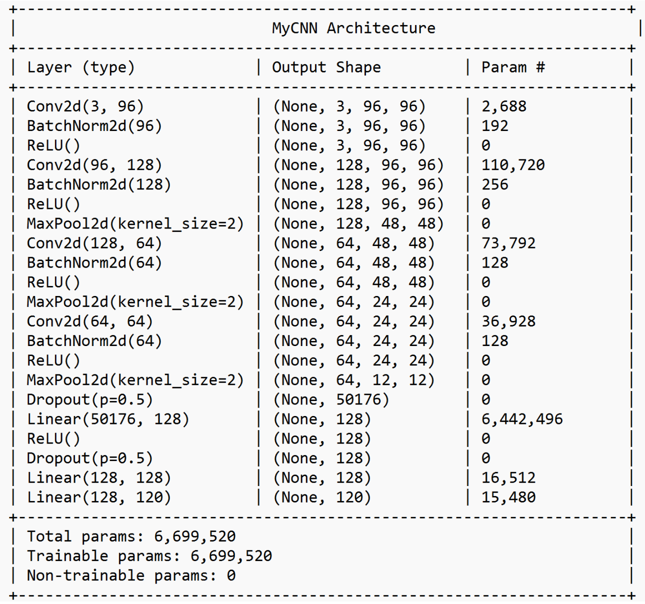


Table 1. **EfficientNet-B0 baseline network** – Each row describes a stage i with $\hat{L}_i$ layers, with input resolution $\langle \hat{H}_i, \hat{W}_i \rangle$ and output channels $\hat{C}_i$. Notations are adopted from equation 2.

Stage i	Operator $\hat{\mathcal{F}}_i$	Resolution $\hat{H}_i \times \hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv3x3	224 × 224	32	1
2	MBConv1, k3x3	112 × 112	16	1
3	MBConv6, k3x3	112 × 112	24	2
4	MBConv6, k5x5	56 × 56	40	2
5	MBConv6, k3x3	28 × 28	80	3
6	MBConv6, k5x5	14 × 14	112	3
7	MBConv6, k5x5	14 × 14	192	4
8	MBConv6, k3x3	7 × 7	320	1
9	Conv1x1 & Pooling & FC	7 × 7	1280	1

Fig. 7. EfficientNet Architecture



```
+-----------------------------------------------------------------+
|                        MyCNN Architecture                        |
+-----------------------------------------------------------------+
| Layer (type)               | Output Shape         | Param #      |
+-----------------------------------------------------------------+
| Conv2d(3, 96)              | (None, 3, 96, 96)    | 2,688        |
| BatchNorm2d(96)            | (None, 3, 96, 96)    | 192          |
| ReLU()                     | (None, 3, 96, 96)    | 0            |
| Conv2d(96, 128)            | (None, 128, 96, 96)  | 110,720      |
| BatchNorm2d(128)           | (None, 128, 96, 96)  | 256          |
| ReLU()                     | (None, 128, 96, 96)  | 0            |
| MaxPool2d(kernel_size=2)   | (None, 128, 48, 48)  | 0            |
| Conv2d(128, 64)            | (None, 64, 48, 48)   | 73,792       |
| BatchNorm2d(64)            | (None, 64, 48, 48)   | 128          |
| ReLU()                     | (None, 64, 48, 48)   | 0            |
| MaxPool2d(kernel_size=2)   | (None, 64, 24, 24)   | 0            |
| Conv2d(64, 64)             | (None, 64, 24, 24)   | 36,928       |
| BatchNorm2d(64)            | (None, 64, 24, 24)   | 128          |
| ReLU()                     | (None, 64, 24, 24)   | 0            |
| MaxPool2d(kernel_size=2)   | (None, 64, 12, 12)   | 0            |
| Dropout(p=0.5)             | (None, 50176)        | 0            |
| Linear(50176, 128)         | (None, 128)          | 6,442,496    |
| ReLU()                     | (None, 128)          | 0            |
| Dropout(p=0.5)             | (None, 128)          | 0            |
| Linear(128, 128)           | (None, 128)          | 16,512       |
| Linear(128, 120)           | (None, 120)          | 15,480       |
+-----------------------------------------------------------------+
| Total params: 6,699,520                                          |
| Trainable params: 6,699,520                                      |
| Non-trainable params: 0                                          |
+-----------------------------------------------------------------+
```

Fig. 8. MyCNN Architecture

Table 1. Models Performance on Dog Breed Dataset

Model	Validation Accuracy	Testing Accuracy
Baseline Model	1.09%	1.09%
Resnet50	57.14%	58.84%
Inception V3	**82.19%**	**85.59%**
EfficientNet	**65.68%**	**74.54%**
VGG16	60.59%	64.21%
MyCNN	1.11%	didn't check

4 RESULTS

In this section, we will present the results of our experiments with the four pre-trained CNN models and the MyCNN model. We used a dataset of 10, 222 images of dogs from 120 different breeds, with approximately 100 images per breed. The images were resized and normalized so that they fit the requirements of the model. For example, some models take 224 X 224 images while others require 299 X 299 images as inputs.

We found the dataset on Kaggle and then We started by exploring the dataset and found that it was almost evenly distributed across all 120 breeds. We also observed that some breeds had more similar physical characteristics than others, which could make their classification more challenging.

Then, we trained the five models on the dataset and compared their performance to a baseline model. The baseline model was a very simple model which always produces the most frequently present dog in the image regardless of the image you give to it.

The results are summarized in Table 1 below, which shows the accuracy of each model on the test set.

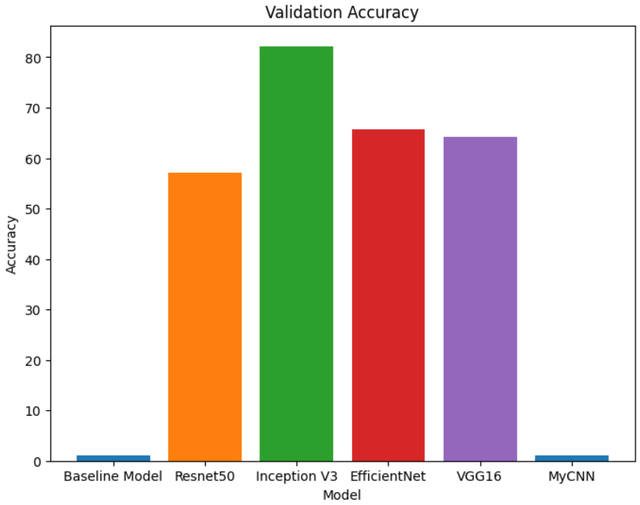


Fig. 9. Bar Chart Validation Accuracies

5 DISCUSSION

The models that we have used produced a relatively high accuracy given the time that we have trained them on. We have not exceed 50 epochs for most models and we have not exceeded 70 epochs for any model. The models InceptionV3 and Efficientnet produced an accuracy of 85.59% and 74.54%. This goes on to show that these

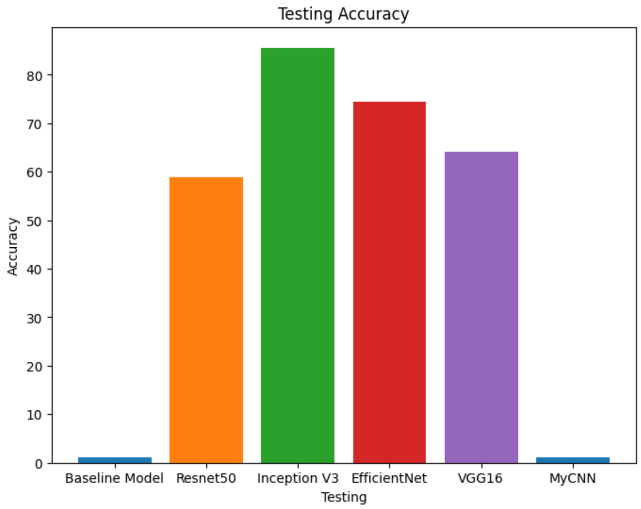


Fig. 10. Bar Chart Testing Accuracies

models are robust and efficient for the dog breed classification and other classification tasks as well.

One of the limitations of this study is that the dataset we used was relatively small. Generally, for deep learning models, the more the data that we have the better they will be able to learn and accurately classify the images.

This model that we have built by hand shows that unless you have large amounts of data and can spend a lot of money for computation, you shouldn't consider building your own models. Instead, it is better to use pretrained models and use transfer learning to use these models for your own classification or image recognition tasks.

For future research, it would be valuable to collect more data and join it together and then train the model. For example, there is standford dogs dataset which we could have used and then we could have scraped the web for more dogs. Finally, efforts to optimize the hyperparameters and adding more layers at the end of a pre-trained model could lead to more efficient and accurate models.

6 ACKNOWLEDGMENTS

Special Thanks to Professor Yan Huang and to the Teaching Assistants Riyad Bin Rafiq and Zhaomin Xiao

REFERENCES

[1] Shiddieqy, H. A., Hariadi, F. I., & Adiono, T. (2017, November). Implementation of deep-learning based image classification on single board computer. In 2017 International Symposium on Electronics and Smart Devices (ISESD) (pp. 133-137). IEEE. doi: 10.1109/ISESD.2017.8253319

[2] Shah, B. K., Kumar, A., & Kumar, A. (2020, December). Dog Breed Classifier for Facial Recognition using Convolutional Neural Networks. In 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS) (pp. 508-513).

IEEE. doi: 10.1109/ICISS49785.2020.9315871

[3] Prasong, P., & Chamnongthai, K. (2012, July). Face-recognition-based dog-breed classification using size and position of each local part, and PCA. In 2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (pp. 1-5). IEEE. doi: 10.1109/ECTICon.2012.6254212

[4] Shuo Wang. 2019. Dog breeds classification with CNN transfer learning. (January 2019). Retrieved April 30, 2023 from

https://medium.com/@wangshuocugb2005/dog-breeds-classification-with-cnn -transfer-learning-92217cba3129

[5] Clay Mason. 2018. Dog breed image classification. (December 2018). Retrieved April 30, 2023 from

https://medium.com/@claymason313/dog-breed-image-classification-1ef7dc1b1967

[6] Amanjot Singh. 2020. Dog breed classification using RESNET50 and VGG19. (June 2020). Retrieved April 30, 2023 from

https://medium.com/@amanjot.uf/dog-breed-classification-using-resnet50-and-vgg19-42133f631e7b

[7] Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint arXiv:1409.1556

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016. doi: 10.1109/CVPR.2016.90.

[8] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[9] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning (ICML 2019) (pp. 6105–6114). Association for Computing Machinery.

https://doi.org/10.1145/3292500.3330891

[10] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.

[11] Keiron O'Shea and Ryan Nash. 2015. An introduction to Convolutional Neural Networks. arXiv preprint arXiv:1511.08458 (2015). Retrieved April 30, 2023 from https://arxiv.org/abs/1511.08458

[12] Anon. Dog breed identification. Retrieved April 30, 2023 from https://www.kaggle.com/competitions/dog-breed-identification

[13] Mingxing Tan and Quoc V. Le. 2018. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018, pp. 8697-8710. Retrieved May 1, 2023 from

[14] Shubham Jangid, Rohan Koli, Prathamesh Chaugule, and Shubham Rawal. 2022. Dog Breed Recognition using Deep Learning. International Journal of Creative Research Thoughts (IJCRT) 10, 1 (2022), 1052-1057. https://ijcrt.org/papers/IJCRT2205839.pdf