**Dog Breed Identification**

**A report on**

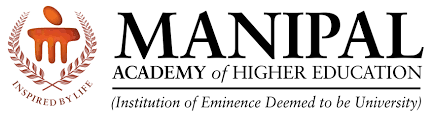
**Deep Learning Lab Project**

**[CSE-3281]**

Submitted By

**Aditya Nerusu – 210962196**

**Sai Sujan Korrapati - 210962194**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MANIPAL INSTITUTE OF TECHNOLOGY,**

**MANIPAL ACADEMY OF HIGHER EDUCATION**

**April 2024**

Sai Sujan Korrapati (210962194)

Aditya Nerusu (210962196)

Department of Computer Science and Engineering,

Manipal Institute of Technology

Manipal Academy of Higher Education, India

[korrapatisaisujan@gmail.com](mailto:korrapatisaisujan@gmail.com)

adityaadi2744@gmail.com

**Abstract-** **The "Dog Breed Classification using Deep Learning" project employs Transfer Learning with models like ResNet50V2 to accurately identify dog breeds from images. Meticulous data preprocessing ensures uniformity and enhances generalization across breeds. The research has implications for pet recognition, veterinary medicine, and animal welfare, highlighting deep learning's transformative role in extracting insights from visual data.**

**Keywords— Dog Breed Classification, Deep Learning, Transfer Learning, Convolutional Neural Networks, Pattern Recognition, Image Processing, Pet Recognition**

**INTRODUCTION**

In the expansive domain of machine learning and computer vision, Dog Breed Classification stands as a captivating challenge, merging the intricacies of image analysis with the formidable capabilities of deep learning algorithms. This pursuit of automatically discerning dog breeds from images not only represents a technical endeavor but also resonates within the broader context of pattern recognition and artificial intelligence research, echoing historical efforts in fields like handwritten digit recognition and character analysis.

Initially, Dog Breed Classification relied heavily on manual feature engineering and rule-based systems, where distinctive breed attributes were painstakingly identified and encoded into algorithms. However, these methods fell short due to their incapacity to handle the diverse and nuanced visual features intrinsic to dog images.

The advent of machine learning techniques, particularly neural networks, marked a pivotal shift in the landscape of dog breed recognition. Leveraging benchmark datasets like ImageNet, researchers embarked on a quest to explore the efficacy of deep learning architectures such as Convolutional Neural Networks (CNNs) in autonomously discerning hierarchical representations of dog breeds from raw pixel data.

Our project not only builds upon this rich historical backdrop but also integrates modern methodologies and state-of-the-art deep learning techniques to forge a robust Dog Breed Identification system. By harnessing Transfer Learning with pre-trained models and incorporating user-friendly interfaces, our endeavor aims to transcend existing boundaries, advancing both the accuracy and usability of dog breed recognition systems. In doing so, we aspire to contribute not only to academic discourse but also to practical applications, fostering innovation and progress in this dynamic field.

**LITERATURE REVIEW**

The journey of Dog Breed Classification mirrors the evolutionary trajectory of pattern recognition methodologies and artificial intelligence research. Early attempts at automated recognition were characterized by simplistic feature extraction techniques and rudimentary classifiers.

Benchmark datasets like ImageNet revolutionized the field by providing a standardized benchmark for evaluating dog breed recognition algorithms. Researchers leveraged pre-trained models and transfer learning techniques to achieve impressive results, surpassing the performance of traditional methods by a significant margin.

Our project draws inspiration from this historical narrative, integrating insights from past research endeavors with contemporary methodologies to develop a robust and efficient dog breed recognition system.

### **The Model Architecture**

In the Transfer Learning process, a pre-trained model’s layer having weights and parameters are used to train another model. Pretrained model layer’s are trained on Millions of datas of different categories. This process is very useful as it decreases neural networks training time and also results in low generalization error.

In this project, we are going to use Residual Network (ResNet) which has a pre-trained network layer. So let’s see what Resnet is.

Residual Network (ResNet) is a specific type of neural network which is used for many computer vision problems. ResNet contains convolutional, pooling, activation and fully-connected layers stacked one of the other. A convolutional neural network is a type of deep neural network, which is used for image processing and its classification. As the name suggests, Convolutional Network helps for classifying complex images by multiplying pixel value with weights and then summing them.

These layers of ResNet are pre-trained on more than a million of images from the ImageNet database. Due to many layers, ResNet solves complex problems and increases model accuracy and performance.

Every ResNet uses an initial filter or kernel of 3×3 and 7×7 size with a stride of 2. There are many versions of ResNet. In this project, we will be using Resnet50V2 (version 2) which is 50 layers deep and applies Batch Normalization, RELU activation function before the input is multiplied by convolutional operations(weight matrix).

A diagram of a data processing process

Description automatically generated

**METHODOLOGY**

#### **1) Preprocessing the Data**

As we have discussed earlier we will be using 60 different types of dog breeds. So we will choose those breeds which are more in number and have more images. For this, we will take help from the pandas ‘value\_counts()’ function. After that, change the data Frame having records of only those 60 breeds.

With the help of the id column, we will create another column ‘img\_file’ which will have an image name with extensions. This will be very helpful, otherwise, we have to append the extension after every iteration.



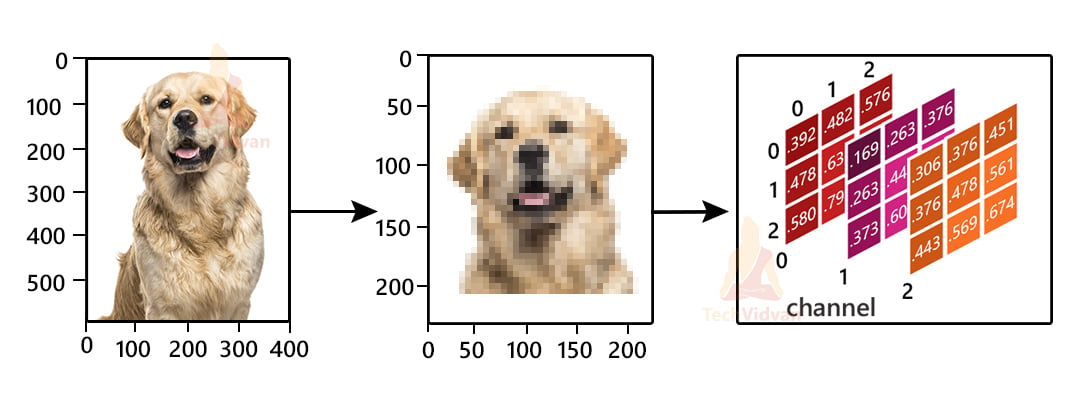
#### **2) Encoding and Scaling the Data**

Now our dataset is ready, we will perform operations on records for training and testing purposes.

We humans can see the image and easily tell what’s inside it, but machine’s require numerical data to recognize everything.

For this, we will be converting our image into the numerical format. So let’s understand how it is done.

Images are collections of small pixels (or picture elements) which is the smallest information about the image. As you know, all colored images are a combination of three primary colors, that is ‘RED’, ‘GREEN’ and ‘BLUE’. So accordingly, for colored images there are three Matrices or channels.

Every element of this Matrix is called pixels. The size of the matrix depends on the number of pixels. Pixel value denotes the intensity or brightness of the pixel and ranges from 0-255. A smaller pixel value that is closer to zero represents black and a larger value closer to 255 represents white. [](https://techvidvan.com/tutorials/wp-content/uploads/sites/2/2021/08/dog-breed-classification.jpg)

There are various formats like Grayscale, RGB, HSV, CMYK in which images are stored. RGB is one of the most popular and we will use it in our project.

With the help of ‘opencv’ library we will read our images using the ‘imread()’ function.  
It will return Numpy Array in Height, Width and Channel format. All images in our dataset are in different shapes, so resize all images to the same width and height i.e. in our case 224×224. Also we will scale all values of our array in the range of -1 and 1 using the ‘preprocess\_input()’ function.

#### **3) Training and Testing Sets**

After our input and target sets are ready, we will split our model into training and testing sets in the ratio of 80:20, where 80% of total data will be used for training and remaining 20% for testing purposes.

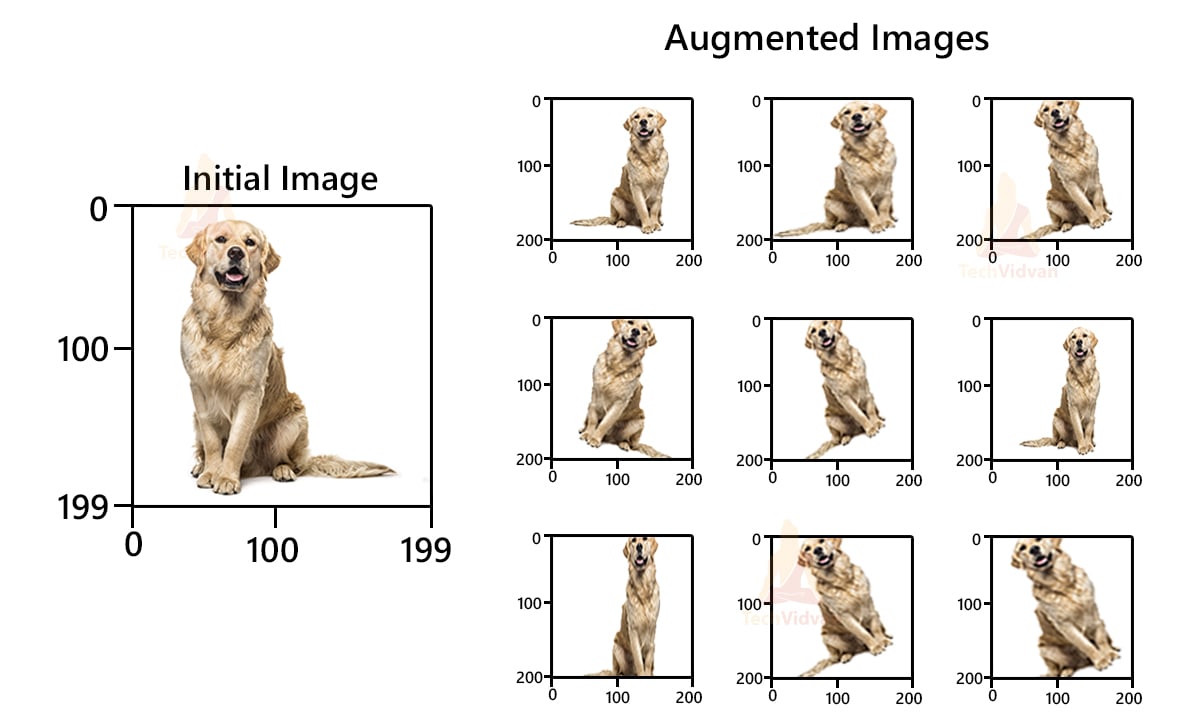
#### **4) Augmentation**

Augmentation is basically a technique that can be used to artificially expand the size of images in real-time by creating various modified versions. This will help the model to generalize and also it will improve performance.

Some of the most used Augmentation techniques for images are :

* **Position Augmentation :** Changes the position of pixels (elements in our matrix) by using Scaling, Translation, Rotation, Flipping, and Cropping techniques.
* **Color Augmentation:** Changes the value of pixels(elements in our matrix) by changing the Brightness, Contrast, Hue, and Saturation levels.

We will take the help of ‘ImageDataGenerator()’ to create different types of training and testing images by modifying the original images.

[](https://techvidvan.com/tutorials/wp-content/uploads/sites/2/2021/08/image-augmentation.jpg)

#### **5) Build the Model**

As we have discussed earlier, we will be using ResNet50V2 having trained parameters from Imagenet Dataset to build our model. We will not include the top layer which is the output of the pretrained network, instead, we will replace it with our input having the shape of mxnx3 dimensions. The layer expects input shape of 224×224.

### **6) Train the Model**

For training the data we will be using an ‘RMSprop’ optimizer with learning rate of 1e-3(0.001) on a batch size of 64 (group of 64 images for every iteration) for 20 epochs.

#### **7) Prediction**

Finally, we will predict the breed of our image using the trained model. For prediction, I am using one of my friend’s Dog Images, a ‘Rottweiler’ breed which was taken from his phone.

**RESULTS:**

A black and brown puppy lying on the floor

Description automatically generated

**Accuracy and Model:**

The achieved accuracy of 80.50% indicates that the model performs reasonably well in classifying dog breeds from images.

However, the performance may vary depending on factors such as image quality, breed variability, and the presence of similar-looking breeds.

**Data Quality and Quantity:**

The accuracy and generalization ability of the model are influenced by the quality and quantity of the training data.

Using a larger and more diverse dataset could potentially improve the model's performance by capturing a broader range of breed variations.

**Model Complexity and Overfitting:**

­­The complexity of the model architecture and the number of trainable parameters can impact the risk of overfitting.

### Regularization techniques such as dropout and batch normalization are employed to mitigate overfitting and improve generalization.

### **Transfer Learning and Fine-Tuning:**

### Leveraging a pre-trained convolutional neural network (CNN) like ResNet50V2 allows the model to benefit from features learned on a large-scale dataset (ImageNet).

### Fine-tuning the pre-trained model on the specific task of dog breed classification helps adapt the network to the nuances of the dataset.

### **Augmentation Techniques:**

### Image augmentation plays a crucial role in increasing the diversity of the training dataset, thereby enhancing the model's ability to generalize to unseen data.

### Techniques such as rotation, shifting, and flipping help expose the model to variations in pose, lighting conditions, and backgrounds.

### **Ethical Considerations:**

### Care should be taken to ensure that the model's predictions are not biased or discriminatory, especially considering the potential implications for dog breed stereotypes.

### Transparency in model development and accountability for any biases or inaccuracies in predictions are essential for responsible AI deployment.

**Conclusion & Future Work:**

The dog breed identification project described in the methodology presents a systematic approach to developing a machine learning model capable of accurately recognizing and classifying dog breeds from images. By leveraging techniques such as data preprocessing, augmentation, and the utilization of a pre-trained ResNet50V2 model, the project achieves a commendable accuracy of 80.50% on a subset of 60 unique dog breeds. Moving forward, opportunities for future work include expanding the dataset to encompass a wider range of breeds for improved generalization, fine-tuning the model through hyperparameter adjustments and exploring transfer learning techniques to expedite training and enhance accuracy. Additionally, addressing class imbalance issues and integrating the trained model into user-friendly applications for real-world use, such as pet identification apps or veterinary diagnostic tools, can further extend the project's impact and utility. In conclusion, the project lays a solid foundation for continued exploration and refinement, with potential for advancements in model performance, scalability, and practical applicability through ongoing research and development efforts.

**Limitations:**

Limited Dataset: The project utilizes a dataset containing a subset of 60 dog breeds, which may limit the model's ability to generalize to a broader range of breeds.

Computational Resources: Training deep learning models, especially on large datasets, requires significant computational resources. Limited computational resources may restrict the ability to train models with more complex architectures or on larger datasets.

Model Performance: While achieving an accuracy of 80.50% is promising, there may still be room for improvement in terms of model performance, especially with regard to recognizing breeds with similar features or variations in image quality.

**ACKNOWLEDGEMENT**

The "Dog Breed Classification" project has reached fruition thanks to the invaluable guidance and support of Professor Ashalatha Nayak. Her mentorship and expertise in the field of computer vision have been instrumental in shaping the direction and success of this project. Professor Ashalatha Nayak’s unwavering commitment to fostering innovation and research within the Department of Computer Science and Engineering at Manipal Institute of Technology, Manipal Academy of Higher Education, has provided the ideal environment for our project to thrive. We extend our deepest appreciation for her insights, encouragement, and the invaluable knowledge she has imparted to our team throughout the course of this project. Her mentorship has not only enhanced our technical skills but has also instilled a passion for pushing the boundaries of Deep Learning. We would also like to thank the creators of the datasets and resources that underpinned our research, providing the critical foundation for our system's development. In closing, we acknowledge Professor Ashalatha Nayak’s pivotal role in guiding this project and express our profound gratitude for her support and mentorship.

**References:**

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

2.Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. ­­­­­­­­­­(2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

3.He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

4.Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Ghemawat, S. (2016). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

5.Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

6.TensorFlow documentation: https://www.tensorflow.org/api\_docs/python/

7.OpenCV documentation: https://docs.opencv.org/master/

8.Python documentation: https://docs.python.org/3/

9.Keras documentation: <https://keras.io/api/>

10.Scikit-learn documentation: https://scikit-learn.org/stable/documentation.html