

**TRIBHUVAN UNIVERSITY**

**I0E, PURWANCHAL CAMPUS**

**DHARAN, NEPAL**

**A MINOR PROJECT REPORT ON**

**“DOG BREED CLASSIFICATION USING CNN ALGORITHM”**

SUBMITTED BY:

DEEPAK SHARMA [PUR076BCT029]

DIPESH ADHIKARI [PUR076BCT031]

KRITARTHA BIKRAM SHAH [PUR076BCT041]

LIJESH SUBEDI [PUR076BCT043]

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

PURWANCHAL CAMPUS

DHARAN

SUBMISSION DATE:

JUNE,2023

# CERTIFICATE OF APPROVAL

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a project report entitled “Dog Breed Classification” submitted by Deepak Sharma, Dipesh Adhikari, Krithartha Bikram Shah, Lijesh Subedi in partial fulfillment of the requirements for the Bachelor’s Degree in Computer Engineering.

….........................................................

Head of Department: **Manoj Kumar Guragai**

Department of Electronic and Computer Engineering

Institute of Engineering, Purwanchal Campus

# COPYRIGHT

The authors have agreed that the Library, Department of Electronics and Computer

Engineering, Institute of Engineering, Purwanchal Campus may make this report freely

available for inspection. Moreover, the authors have agreed that permission for extensive

copying of this project report for scholarly purpose may be granted by the supervisors who

supervised the project work recorded herein or in their absence, by the Head of the

Department wherein the project report was done. It is understood that the recognition will

be given to the authors of this project and to the Department of Electronics and Computer

Engineering, Purwanchal Campus, Institute of Engineering in any use of the material of this

report. Copying or publication or the other use of this report for financial gain without

approval of the Department of Electronics and Computer Engineering, Institute of

Engineering, Purwanchal Campus and authors’ written permission is strictly prohibited.

Request for permission to copy or to make any use of the material in this project in whole or

part should be addressed to Department of Electronics and Computer Engineering, IOE,

Purwanchal Campus.

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude and appreciation to all individuals and organizations who contributed to the successful completion of our project on dog breed classification using Convolutional Neural Networks (CNNs).

This project wasn’t possible on its own, we would like to express our sincere gratitude to Mr. Manoj Kumar Guragain, Head of Department, and Mr. Pravin Sangroula, Deputy Head of Department, who gave us the golden opportunity to do this wonderful project that aims to create the Dog Breed Classification. We’re thankful to the Department of Computer and Electronics Engineering, Purwanchal Campus for granting us this good opportunity. Further, we are obliged to our friends and classmates who constantly helped without any hesitation and expectations instantly helped us solve and tackle the problems we faced.

Besides, we are indebted to the open-source community for providing the necessary tools, libraries, and frameworks that enabled us to implement the CNN model effectively. The availability of well-documented resources and the generosity of developers worldwide greatly facilitated our progress.

In conclusion, we acknowledge the invaluable contributions of everyone involved in this project. Their support, guidance, and cooperation have been vital to its successful completion.

# ABSTRACT

Dog breed classification is a challenging task due to the wide variety of dog breeds with subtle differences in appearance. Convolutional Neural Networks (CNNs) have shown remarkable performance in image classification tasks, making them an ideal choice for dog breed recognition. In this project, we propose a CNN-based approach for dog breed classification.

The project 'Dog Breed Classification Using CNN' aims to develop a mobile application using the Flutter framework for accurate identification of dog breeds from images. This report presents the methodology, implementation details, and evaluation of the developed application. The Convolutional Neural Network (CNN) model was trained using a comprehensive dataset of dog images to enable accurate breed classification. The application provides an intuitive user interface, allowing users to capture or upload dog images for real-time breed identification. Evaluation results demonstrate the effectiveness of the CNN model, achieving high accuracy in breed classification. Additionally, the report discusses the challenges faced during the development process and proposes potential areas for future enhancements. The 'Dog Breed Classification Using CNN' project showcases the integration of deep learning techniques, mobile app development, and breed recognition, offering practical applications in various domains such as pet care, animal welfare, and canine research.

|  |  |
| --- | --- |
| TABLE OF CONTENTS  [CERTIFICATE OF APPROVAL i](#_Toc137124577)  [COPYRIGHT ii](#_Toc137124578)  [ACKNOWLEDGEMENT iii](#_Toc137124579)  [ABSTRACT iv](#_Toc137124580)  [1. INTRODUCTION 1](#_Toc137124581)  [1.1 Background: 1](#_Toc137124582)  [1.2 Objective: 1](#_Toc137124583)  [1.3 Problem Statement: 1](#_Toc137124584)  [2. LITERATURE REVIEW: 2](#_Toc137124585)  [3. METHODOLOGY 4](#_Toc137124586)  [3.1 General Description: 4](#_Toc137124587)  [3.2 Hardware Requirement: 4](#_Toc137124588)  [3.3 Software Requirement: 4](#_Toc137124589)  [3.4 Network Architecture: 5](#_Toc137124590)  [3.5 Basic Steps: 6](#_Toc137124591)  [3.5.1 Data Collection and Preparation: 6](#_Toc137124592)  [3.5.2 Data Preprocessing: 7](#_Toc137124593)  [3.5.3 Building the CNN Model: 7](#_Toc137124594)  [3.5.4 Training the CNN Model: 9](#_Toc137124595)  [3.5.5 Evaluation: 10](#_Toc137124596)  [3.6 Block Diagram: 11](#_Toc137124597)  [3.7 Usecase Diagram: 12](#_Toc137124598)  [3.8 Sequence Diagram: 13](#_Toc137124599)  [3.9 Activity Diagram: 14](#_Toc137124600)  [4. OUTPUT AND ANALYSIS: 15](#_Toc137124601)  [5.CONCLUSION: 16](#_Toc137124602)  [REFERENCES: 17](#_Toc137124603) |  |

# 1. INTRODUCTION

Dog breed classification is a challenging computer vision task that involves identifying the breed of a dog from an input image. The development of Convolutional Neural Networks (CNNs) has revolutionized image classification problems, including dog breed classification. This project proposes to develop a dog breed classification system using CNNs.

## 1.1 Background:

The American Kennel Club (AKC) recognizes around 197 dog breeds, each with unique physical characteristics and temperament. Accurately identifying the breed of a dog can be challenging and there is a need for a computer vision system that can classify dogs by their breed. Traditional image recognition techniques, such as feature engineering and Support Vector Machines, have been used to classify dog breeds with some success, but they often require manual feature selection and are not scalable for large datasets. CNNs have shown great promise in solving image classification problems, including dog breed classification, without requiring manual feature selection.

## 1.2 Objective:

* Breed Classification
* Real-Time Classification
* Offline Functionality

## 1.3 Problem Statement:

Identifying the breed of a dog from an image is challenging due to the high variation in appearance within a breed and similarities between different breeds. Traditional image recognition techniques have limitations in terms of accuracy and scalability. The development of a CNN-based dog breed classification system can help address these challenges and improve the efficiency and accuracy of identifying dog breeds. In summary, this project proposes to develop a CNN-based dog breed classification system to address the challenges of identifying dog breeds from input images. The system will leverage the power of CNNs to learn breed-specific features and improve the accuracy and efficiency of dog breed classification.

# 2. LITERATURE REVIEW:

Dog breed classification is a well-studied problem in the field of computer vision, and various approaches have been proposed to tackle this task. In recent years, Convolutional Neural Networks (CNNs) have shown great promise in solving image classification problems, including dog breed classification. One of the earliest approaches to dog breed classification was based on hand-crafted feature extraction techniques. This approach involved extracting features such as color, texture, and shape from the input image and using them to train a classifier. However, manual feature selection in this approach can be time-consuming and may not be scalable for large datasets. CNNs have become a popular approach for image classification tasks, including dog breed classification. CNNs consist of multiple layers of convolutional filters that learn hierarchical representations of the input image. They have demonstrated state-of-the-art performance on various image classification tasks, including dog breed classification. One of the pioneering studies in using CNNs for dog breed classification was conducted on the Oxford-IIIT Pet Dataset, which contains 12,000 images of cats and dogs from 37 different breeds. The study employed a pre-trained CNN model, VGG-16, and fine-tuned it on the pet dataset to classify dog breeds. The approach achieved an accuracy of 59.21% on the test set, representing a significant improvement over traditional feature-based methods. Subsequent studies have built upon this work, exploring different CNN architectures and techniques to improve accuracy. For instance, the InceptionV3 architecture was proposed and achieved state-of-the-art accuracy on the Stanford Dogs dataset. Another study introduced a multi-scale CNN architecture capable of handling images of different sizes and achieved high accuracy in the Kaggle Dog Breed Identification Challenge. The Kaggle Dog Breed Identification Challenge has been a popular benchmark dataset in the literature. Various studies have utilized pre-trained CNN models, such as VGG-19, and fine-tuned them on the Kaggle dataset, achieving accuracies as high as 83.7% on the test set. Furthermore, novel CNN architectures like DenseNet have been proposed and demonstrated state-of-the-art accuracy of 92.5% on the Kaggle dataset, surpassing other architectures like VGG and ResNet. In conclusion, the Kaggle Dog Breed Identification Challenge dataset has been widely used in the literature to develop and evaluate dog breed classification models. CNNs have consistently shown state-of-the-art performance on this dataset, and the use of transfer learning and novel architectures, such as DenseNet, have further improved the accuracy of these models.

# 3. METHODOLOGY

The waterfall model is a sequential software development approach where the project progresses through distinct phases, including requirements gathering, system design, implementation, testing, deployment, and maintenance. Each phase is completed before moving on to the next, with a focus on careful planning and documentation.

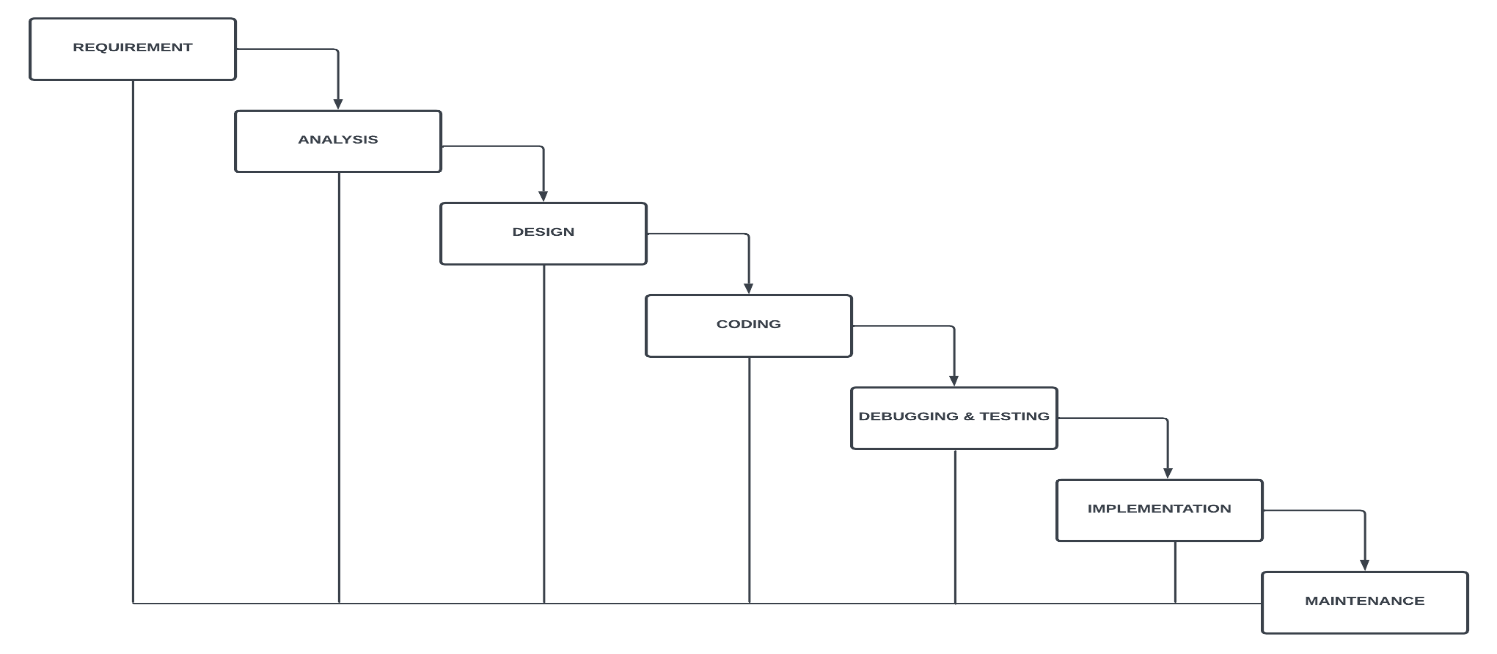
****

Fig: Waterfall model

## 3.1 General Description:

Here, the application predicts the dog breed through the image and live feed along with the accuracy level of given input.

## 3.2 Hardware Requirement:

There are certain hardware requirements needed in order to classify the dog breed. Laptop, smart phone, camera are a few devices that is needed in this system.

## 3.3 Software Requirement:

The software required in the project are Python, Flutter, Dart, Android Studio, Jupyter Notebook etc.

## 3.4 Network Architecture:

In this project, we used MobileNetV2 for model development. MobileNetV2 (Mobile Network Version 2) is a network architecture designed to enable efficient and lightweight deep learning models for computer vision tasks. It builds upon the original MobileNet architecture by introducing inverted residual blocks and linear bottlenecks. MobileNetV2 enables efficient deployment on resource-constrained devices without sacrificing performance, making it a popular choice for mobile and embedded applications requiring real-time image processing. MobileNetV2 uses depthwise separable convolutions, while ResNet integrates features by summation through skip connections.

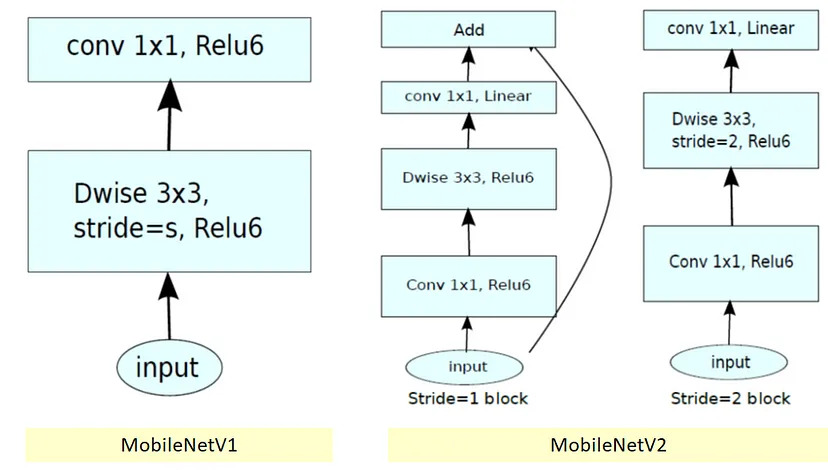


Fig: Comparison between Version1 and Version2 of MobileNet

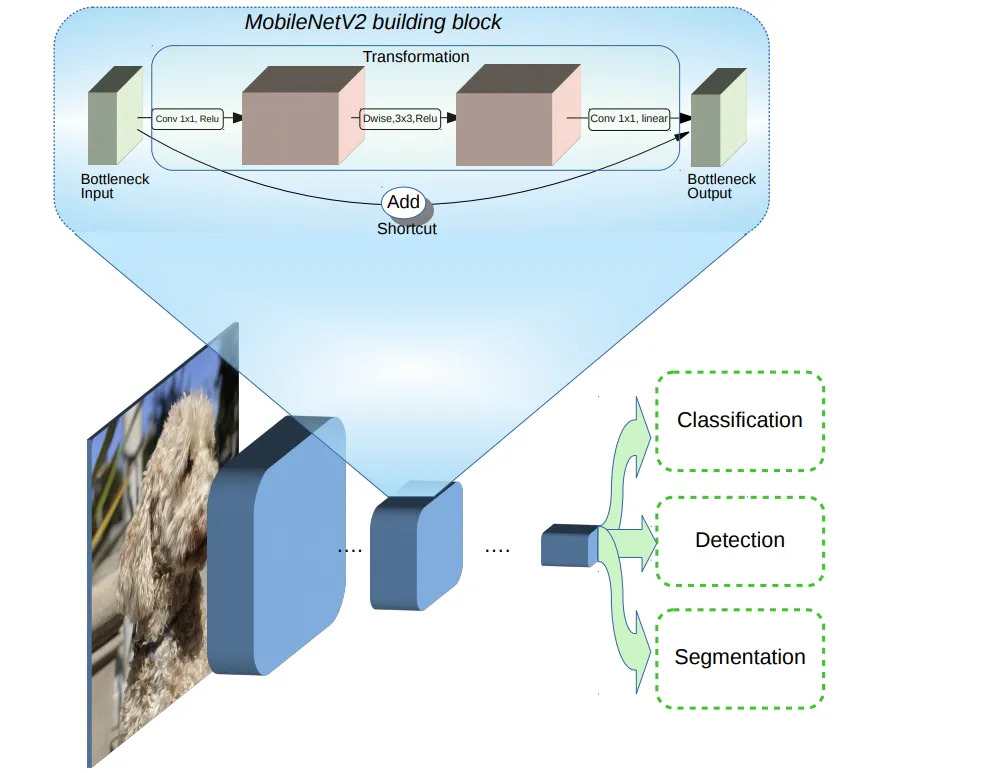


Fig: Architecture of MobileNetV2

## 3.5 Basic Steps:

### 3.5.1 Data Collection and Preparation:

Images used in this project were obtained from the Stanford Dogs Dataset which is a public dataset available online along with other sites like Facebook, Google Images and Adobe Images. This dataset is divided into 74 classes. Some of the classes are Afghan, Yorkshire Terrier, Bulldog, Labrador, Himalayan Sheepdog etc along with Human being class. This Dog Breed Dataset consists of 14,388 Dogs images with breed labels from 72 unique dog breeds.

### 3.5.2 Data Preprocessing:

The total size of obtained data was 4.91 GB. The labels for all the images were arranged in separate subdirectories based on their classes or label. ImageDataGenerator was used to perform data augmentation and normalization on the image data. The flow\_from\_directory function was then used to generate the training and validation data generators.

• Now the data was split. First, group shuffle split was used and a train set was made with

80% of the total data. The remaining 20% of data was split into two halves for test and

validation sets using train test split.

• Next, the images were normalized and resized to a target size of 224x224 pixels.

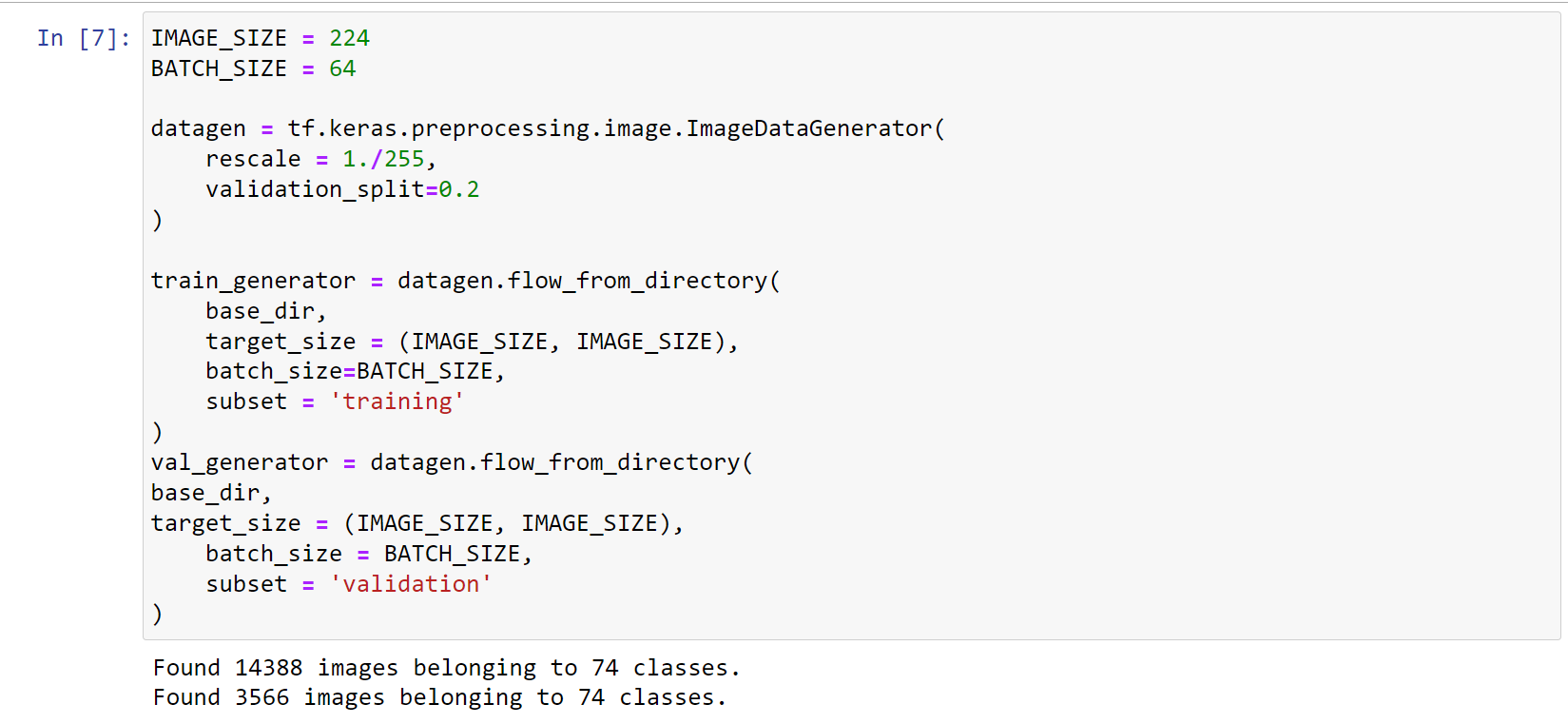
****

Fig: Dataset Preparation

### 3.5.3 Building the CNN Model:

Design the architecture of the CNN. It typically consists of alternating convolutional layers, activation functions (e.g., ReLU i.e. Rectified Linear Unit), and pooling layers to extract relevant features from the images. As our model consist of four sequential layers that include Convo2D, Dropout, GlobalAveragePooling2D and Dense in the final layer.

**3.5.3.1** **Convo2D:** Convo2D is a 2D convolution layer that creates a convolution kernel that is a wind with layers input which helps produce a tensor of outputs. It is trying to understand the image’s patterns.

**3.5.3.2 Dropout:** Dropout layer prevents Neural Networks from Overfitting, i.e. being too precise to a point where the NN is only able to recognize images that are present in the dataset and no other images. The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting.

**3.5.3.3 GlobalAveragePooling2D:** GlobalAveragePooling2D layer calculates the average output of each feature map in the previous layer, thus reducing the data significantly and preparing the model for the final layer.

**3.5.3.4 Dense:** Dense layer is a deeply connected layer in which each neuron receives input from all neurons of its previous layer.

****

Fig: Building the Model

### 3.5.4 Training the CNN Model:

Train the CNN using the labeled training dataset. This involves passing the images through the network, comparing the predicted outputs with the ground truth labels, and adjusting the network's weights and biases to minimize the prediction error. The optimization process is typically performed using an algorithm like stochastic gradient descent (SGD), Adagrad(Adam Gradient), Adamax, Adadelta, Nadam(NAG and Adam) or Adam. Here, we used the Adam(i.e., Adaptive Moment Estimation) optimizer Algorithm and Softmax activation was used to assign higher probabilities to the correct class and lower probabilities to the incorrect classes based on the input image features.The Adam optimizer algorithm provides adaptive learning rates that can handle sparse gradients, noisy gradients, and different scales of parameters. It has demonstrated good performance and convergence properties for a wide range of deep learning tasks.

Mean update: m = β1 \* m + (1 - β1) \* gradient

Variance update: v = β2 \* v + (1 - β2) \* gradient^2

Bias correction: m\_hat = m / (1 - β1^t), v\_hat = v / (1 - β2^t)

(Where, t is the current iteration step)

Parameter update: parameter = parameter - learning\_rate \* m\_hat / (sqrt(v\_hat) + epsilon)

In these formulas, β1 and β2 are the decay rates for the first and second moments (typically set to 0.9 and 0.999, respectively), and learning\_rate is the step size or learning rate for updating the parameters. The epsilon term is a small constant added to the denominator for numerical stability.

We define the number of epochs as 35, which indicates the number of times the model will iterate over the entire training dataset.

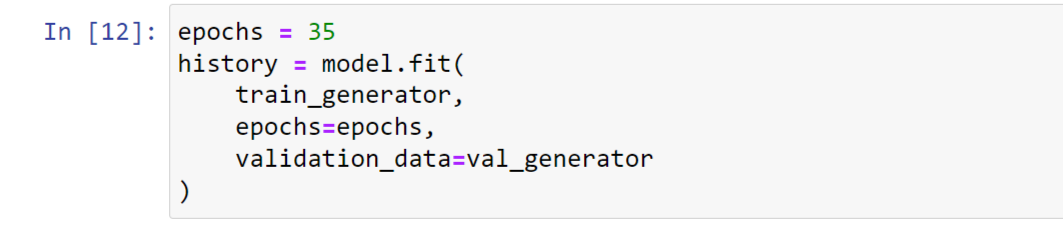


Fig: Training the Model

### 3.5.5 Evaluation:

Evaluated the trained CNN model using the labeled testing dataset to assess its performance. Calculated metrics such as training loss, training accuracy, validation loss and validation accuracy to measure how well the model performs in classifying the dog breeds. The obtained validation loss and validation accuracy was found to be 0.8339 and 87.89% respectively while training loss and training accuracy was found to be 0.0378 and 97.66% respectively for the 35th epoch.

## 3.6 Block Diagram:

This block diagram shows how the classification process is conducted for user-end.

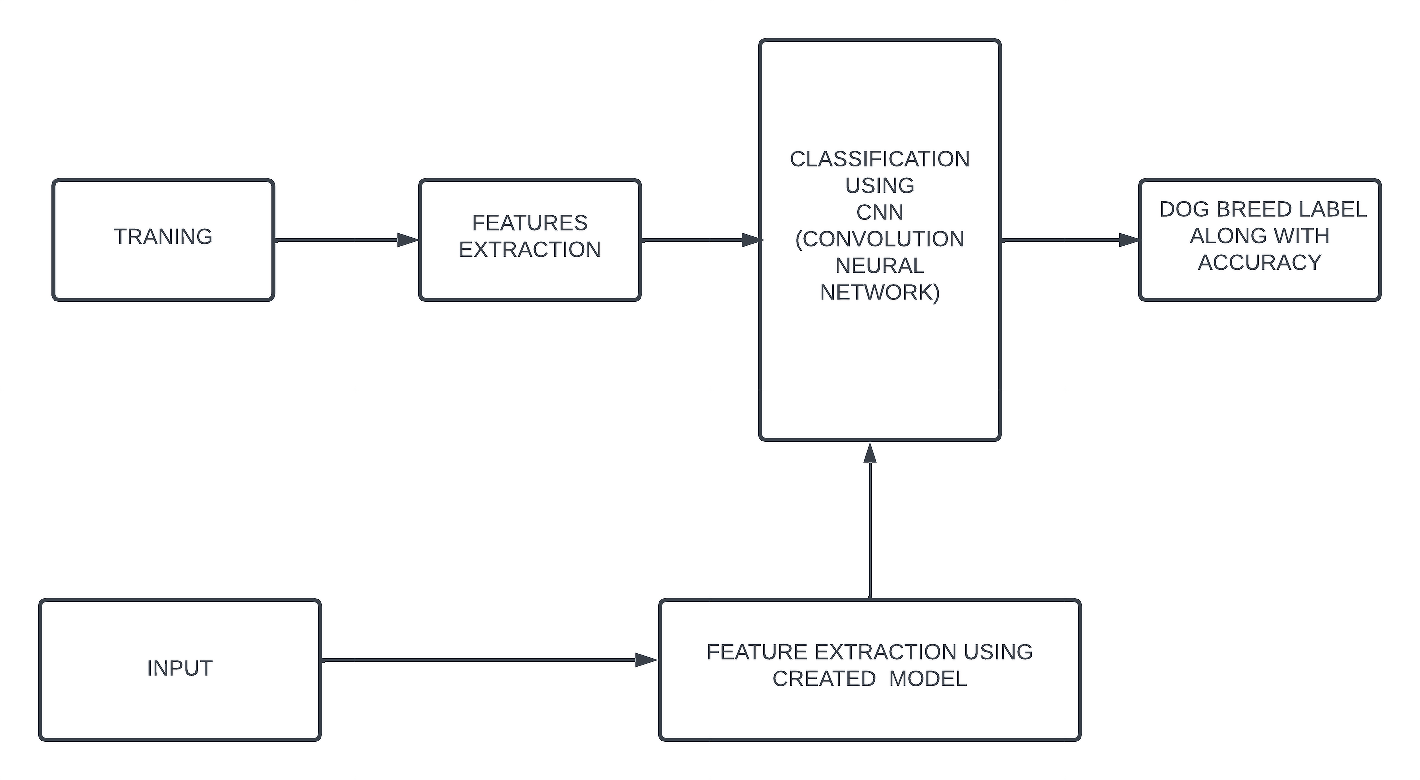


Fig: Block Diagram for Dog Breed Classification

## 3.7 Usecase Diagram:

This use case diagram shows how voter and admin interact with the system.

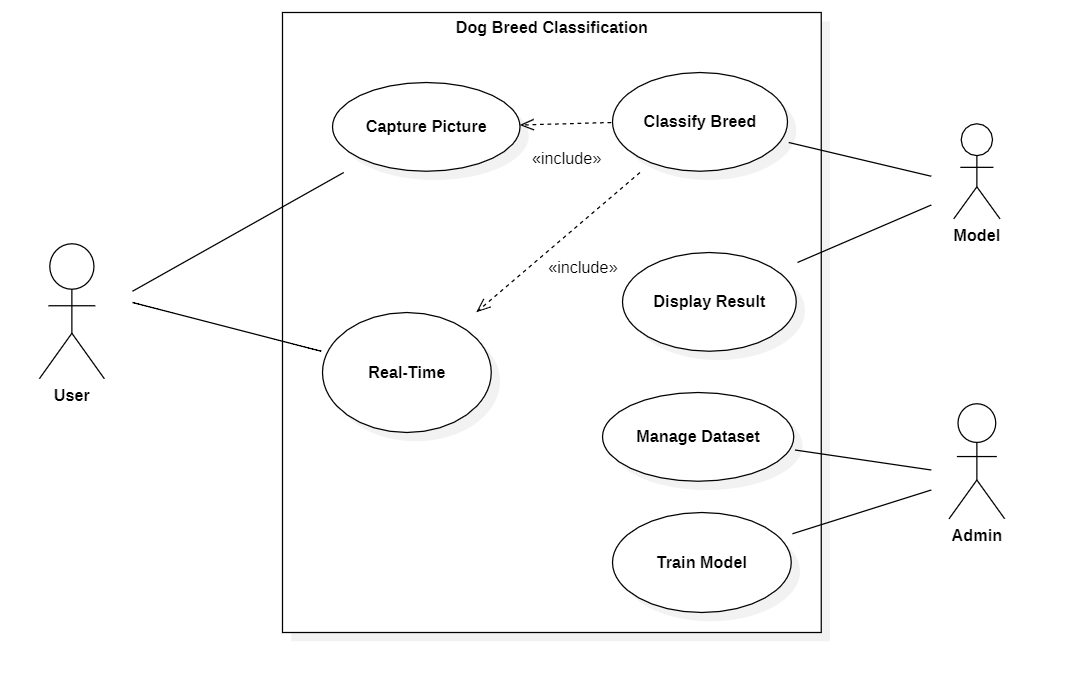
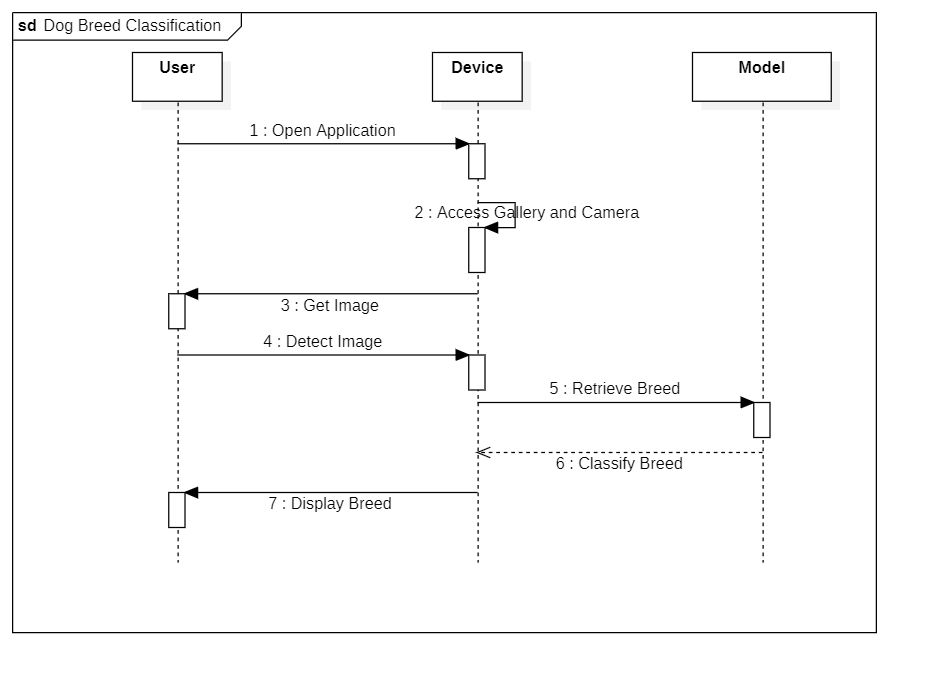
****

Fig: Use Case Diagram

## 

## 3.8 Sequence Diagram:

The sequence diagram shows different sequential flow of this project.

**** Fig: Sequence Diagram

## 

## 3.9 Activity Diagram:

Activity Diagram shows the flow process at which program works.

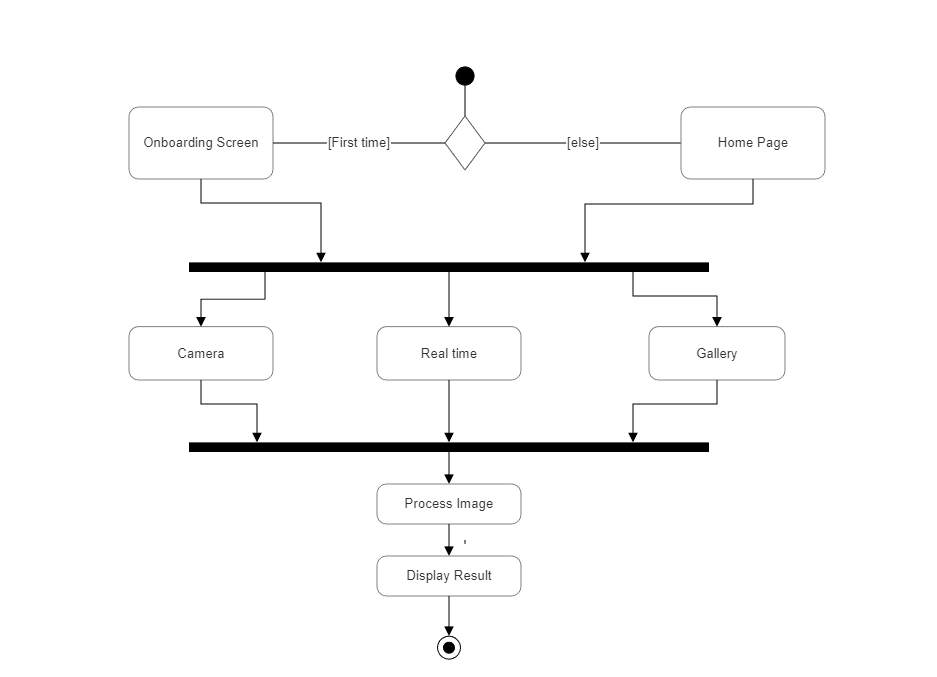
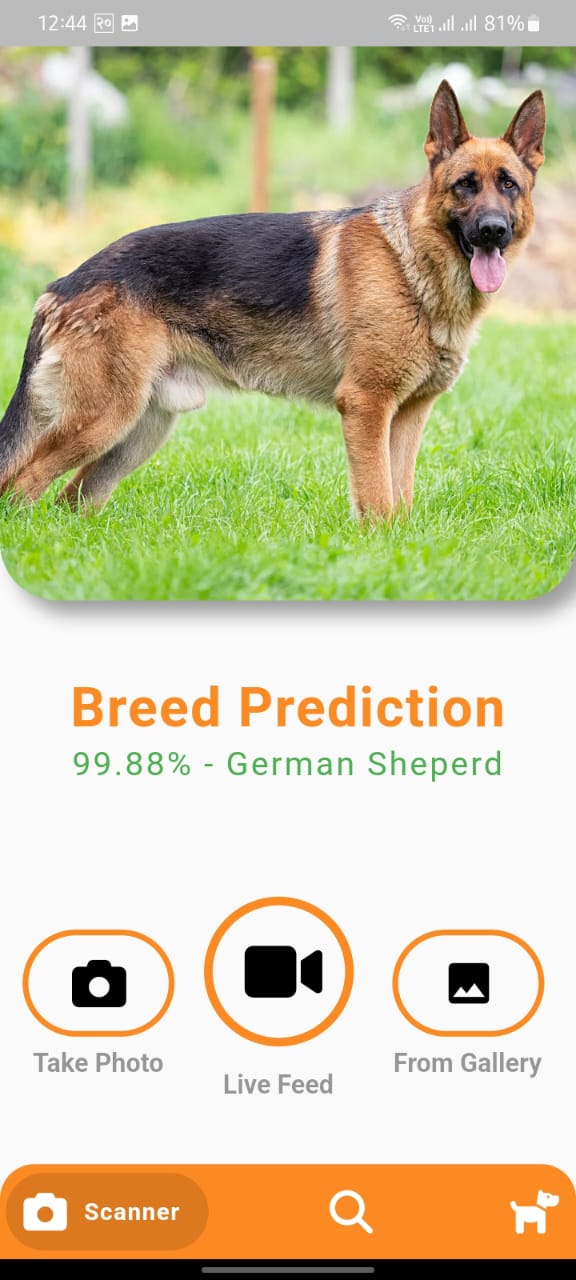
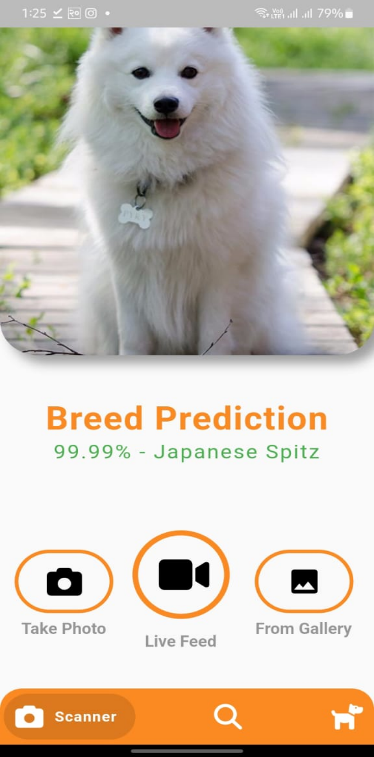


Fig: Activity Diagram

# 4. OUTPUT AND ANALYSIS:

The screenshot of output through the gallery and live feeds are given as below. The Application predicts the breed along with the accuracy level in percentage as shown below.

**  **

**  **

# 5.CONCLUSION:

In conclusion, the development of the dog breed classification app has been a resounding success. The app effectively achieves its goal of accurately identifying the breed of a dog based on an uploaded image. By following a systematic approach that involves preprocessing the images, extracting relevant features, and utilizing a pre-trained deep learning model, the app demonstrates the power of machine learning in solving real-world problems. The user interface is intuitive and user-friendly, allowing for a seamless experience.

# REFERENCES:

​• K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

• S. Ghosh, "Dog Breed Classification with Deep Learning," Medium, Aug. 27, 2021.

• A. Levy, "Dog breed classification using Convolutional Neural Networks," Medium, 2021. Available: <https://ariehlevyp.medium.com/dog-breed-classification-using-convolutional-neural-networks-cef89891d609>

• D. Jamil, "Deep learning for dog breed classification," 2020. [Online]. Available: https://pub.towardsai.net/deep-learning-for-dog-breed-classification-77ef182a2509

• S.-H. Tsang, "Review: MobileNetV2 - Light Weight Model (Image Classification)," Towards Data Science, 2019. [Online]. Available: https://towardsdatascience.com/review-mobilenetv2-light-weight-model-image-classification-8febb490e61c