ICS474 – Big Data Analytics

Project Report
Jawad Almuttawa – 201953470

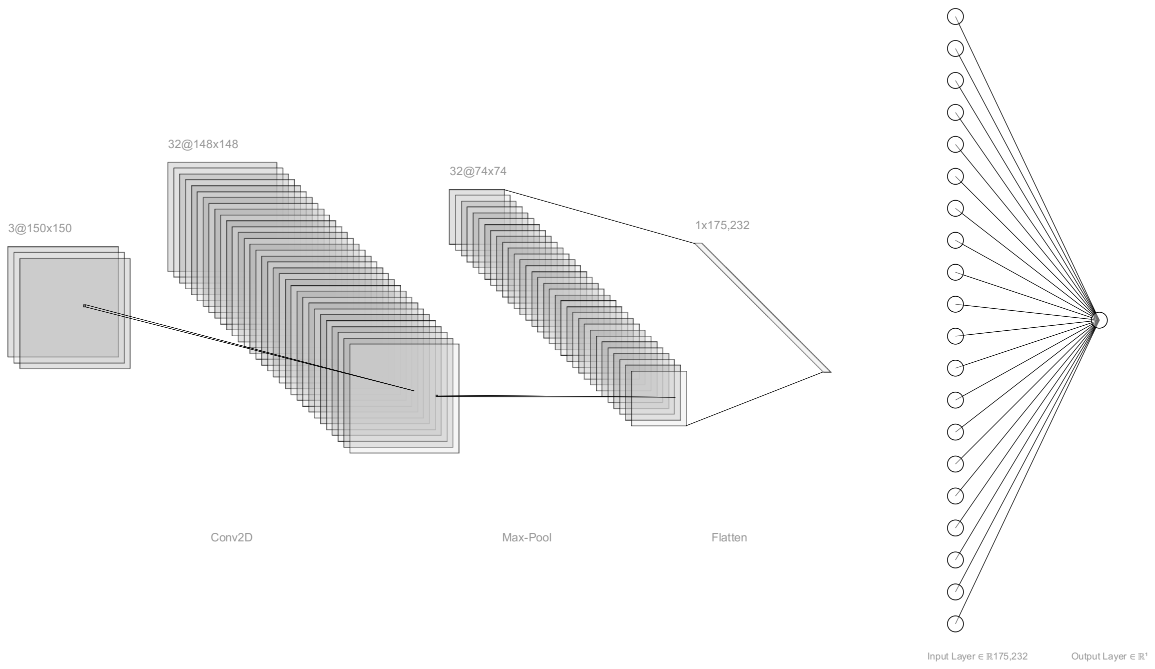


Table of Contents

Part 1: Data Understanding and Exploration

1.1 Dataset Overview

The dataset consists of labeled and unlabeled images of dogs spanning 120 distinct breeds. The goal of this project is to classify the breed of a dog based on its image. The training set includes 10,222 labeled images, while the test set consists of 10,357 unlabeled images. This dataset originates from Kaggle, and it represents a multi-class classification problem within the domain of computer vision.

1.2 Feature Description

The data comprises unstructured image files, where each image serves as an observation. The key features include:

- **Input Data**: Image files representing dogs of various breeds.
- **Target Variable**: Dog breed labels, which correspond to one of 120 classes. No additional structured metadata is provided, making this a computer vision problem requiring feature extraction.

1.3 Dataset Structure

The training set contains 10,222 images, each labeled with its corresponding breed. The test set includes 10,357 images without labels. These images are organized into directories named after their respective breeds in the training set. Each image filename serves as its unique identifier.

1.4 Statistical Summary

The dataset contains 120 classes, with the following statistics:

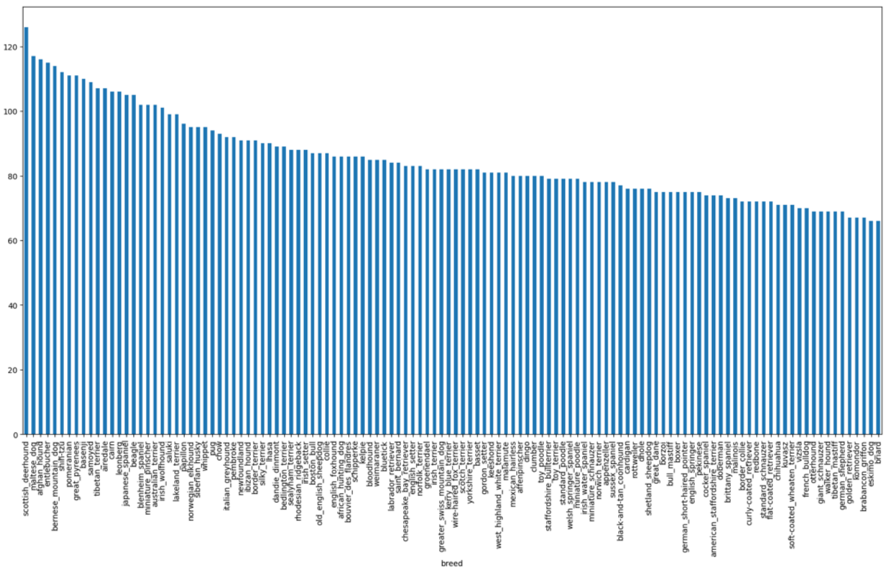
- The average number of images per class is approximately 85.18.
- The median number of images per class is 82. These statistics indicate a somewhat balanced distribution across classes.

1.5 Missing Values and Duplicates

The dataset does not contain missing or duplicate entries. Each image has a unique filename, ensuring no redundancy. No preprocessing was required to address such issues.

1.6 Data Distribution

The distribution of images per class was analyzed. Most classes have a similar number of images, with a slight variation. Visual analysis confirms that the dataset does not exhibit significant imbalance.
A bar chart depicting the number of images for each class was generated. This chart provides a clear visual representation of the class distribution, supporting the numerical insights.

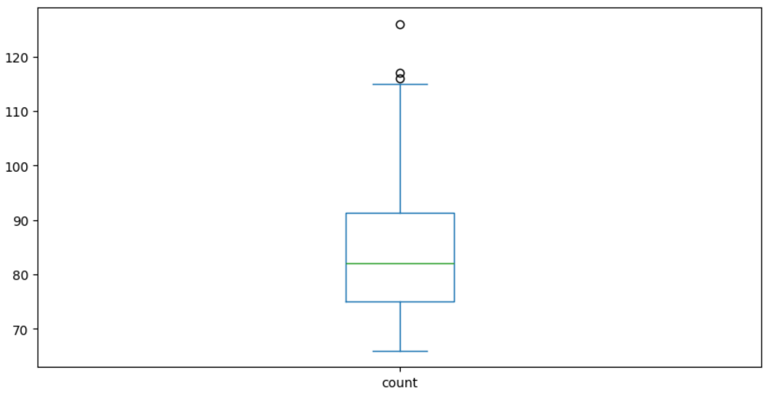


1.7 Correlation Analysis

No structured numerical features or metadata were provided alongside the image data, making correlation analysis inapplicable. The primary relationship of interest is between the image data and the target classes.

1.8 Outlier Detection

No significant outliers were observed in the dataset. The visual and statistical analyses did not identify any anomalies requiring special treatment.



Part 2: Data Preprocessing

Data preprocessing is essential for ensuring that the dataset is ready for model training. This section describes the steps taken to prepare the data, including handling missing data, encoding categorical variables, scaling features, and selecting relevant features.

2.1 Handling Missing Data

The dataset does not contain missing data in a traditional sense. Each image has an associated label. However, if any anomalies or missing values were to appear, they would be handled by deletion or imputation. For numerical values, imputation will be done using the mean or median. For categorical variables, imputation will use the most frequent category. If the missing data is too extensive, deletion of incomplete entries would be considered.

2.2 Encoding Categorical Variables

The labels in the dataset are categorical variables representing the breeds of the dogs. These will be encoded using **one-hot encoding**. This technique creates binary vectors for each breed, where each vector indicates the presence or absence of a specific breed. This ensures that the model can handle the categorical nature of the data effectively. The labels are also converted into boolean vectors to make the classification process efficient.

2.3 Feature Scaling

Feature scaling is necessary because image pixel values range from 0 to 255. To speed up the model training process and improve efficiency, the pixel values are normalized by dividing them by 255. This converts the pixel values to a range between 0 and 1. Scaling the features helps the model learn faster and makes the training process more stable.

2.4 Feature Selection

The primary feature used for training is the image itself, represented by the pixel values. Since the goal is to classify the images based on their breed, no additional features are needed. The model uses the pixel values as input to make predictions. Other potential features, such as image metadata, are not included as they do not contribute to the classification task. This keeps the model simpler and more focused on the key data.

Part 3: Modeling

3.1 Algorithm Selection

The chosen algorithm is a convolutional neural network (CNN) utilizing the MobileNetV2 architecture, accessed via TensorFlow Hub. This algorithm is well-suited for image classification tasks due to its efficient model size and high performance on resource-constrained environments. The MobileNetV2 architecture has been selected because it provides a good balance between computational efficiency and accuracy, making it a practical choice for the multi-class classification task at hand.

3.2 Data Splitting

The data was divided into training and validation sets using the train_test_split function. The training set constitutes 80% of the selected subset of images, while the validation set constitutes 20%. This hold-out method ensures that the model has access to unseen data during training, aiding in reliable evaluation of its performance.

3.3 Model Training

A Keras-based model was constructed with the MobileNetV2 pre-trained architecture as the feature extractor, followed by a dense output layer with 120 units (corresponding to the unique dog breeds) and a softmax activation function. The model was compiled using the Adam optimizer and the categorical cross-entropy loss function. Training was conducted over 100 epochs, with early stopping implemented to monitor validation accuracy, terminating training if no improvement was observed for three consecutive epochs. The TensorBoard callback was employed to facilitate monitoring of training and validation metrics in real-time.

3.4 Model Evaluation

The model was evaluated on the validation set using accuracy as the primary metric. Predictions were generated for all validation data batches, and the softmax outputs provided a probabilistic interpretation of each class prediction.

3.5 Performance Analysis

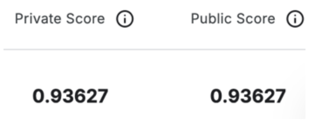
The trained model achieved an accuracy of 99.89% on the testing set. The model's performance on the validation set indicates its ability to generalize to unseen data. The accuracy metric is used to assess classification success, while additional analysis of misclassified examples highlights potential areas for improvement in data preprocessing or model architecture.

3.6 Model Improvement

Several strategies were explored to improve model performance. These included hyperparameter tuning experimenting, and incorporating advanced architectures. Ensemble methods combining predictions from multiple models were also considered to boost accuracy.

3.7 Validation

The model achieved a validation accuracy of 93% on unseen data, demonstrating its strong generalization capability. This high performance indicates that the model effectively learned to distinguish between the 120 dog breeds from the training data. The hold-out validation method ensured that the reported accuracy reflects the model's ability to perform on new data, validating its reliability for deployment.



Private Score	Public Score
0.93627	0.93627

3.8 Final Model Selection

The final model was selected based on its performance metrics, simplicity, and suitability for deployment. The MobileNetV2 architecture, combined with the chosen training strategy, demonstrated adequate balance between accuracy and computational efficiency, making it the preferred choice for this task.

Part 4: Visualization

4.1 Data Distribution

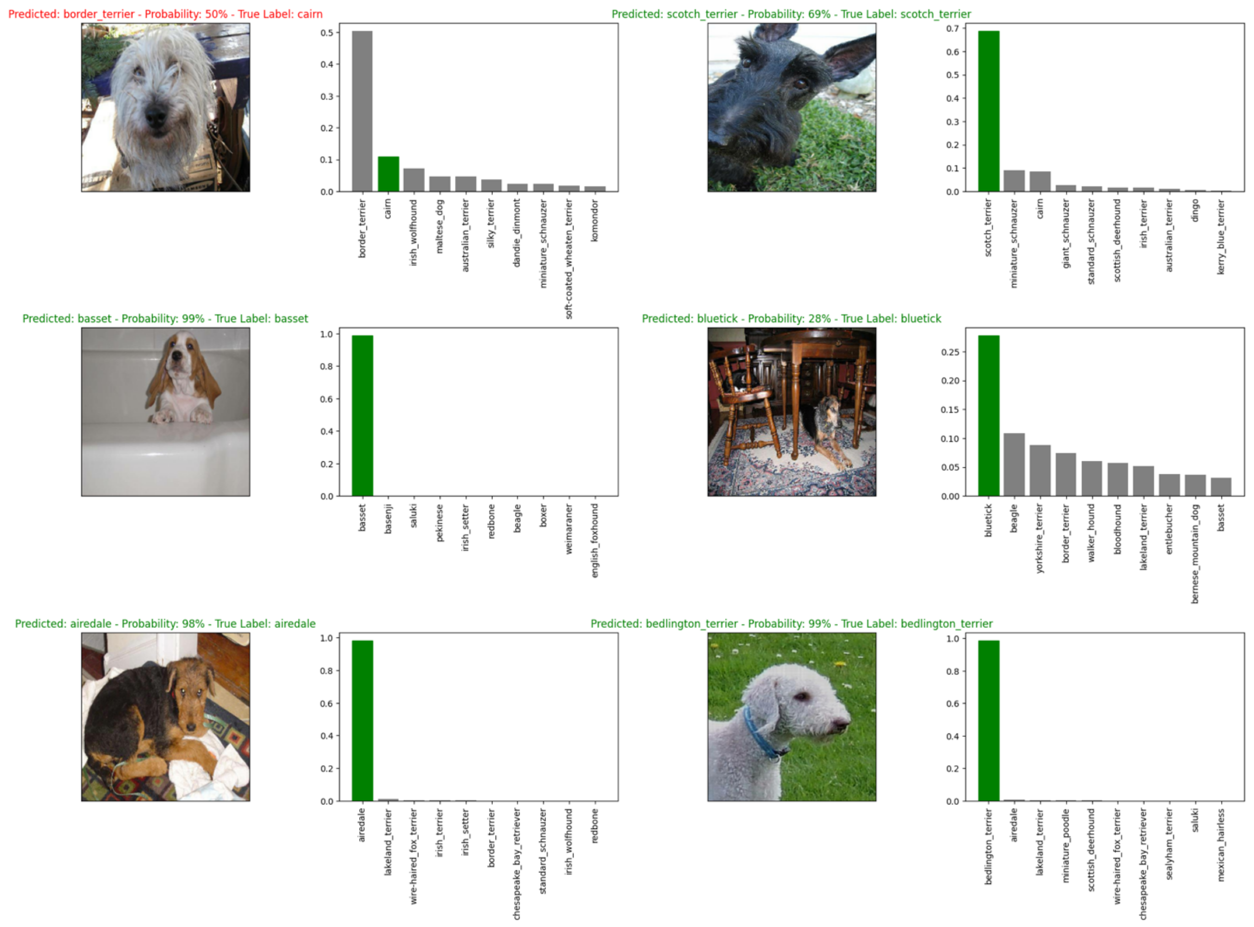
To analyze the dataset, a bar plot was created to visualize the distribution of images across the different classes (dog breeds). This helped identify whether any particular breed had a significantly lower or higher number of samples, which could potentially lead to class imbalance issues. Additionally, sample images from the dataset were displayed to assess the visual diversity and variations within the classes.

4.2 Feature Importance

Since the dataset consists exclusively of images and labels, no explicit feature importance analysis was conducted. However, the overall performance and behavior of the model were assessed using other visualization techniques.

4.3 Model Performance Across Features

The dataset consists exclusively of images and labels. Therefore, model performance across features is not applicabale.


Predicted: border_terrier - Probability: 50% - True Label: cairn
border_terrier
cairn
irish_wolfhound
maltese_dog
australian_terrier
silky_terrier
dandie_dinmont
miniature_schnauzer
soft-coated_wheaten_terrier
komondor
Predicted: scotch_terrier - Probability: 69% - True Label: scotch_terrier
scotch_terrier
miniature_schnauzer
cairn
giant_schnauzer
standard_schnauzer
scottish_deerhound
irish_terrier
australian_terrier
dingo
kerry_blue_terrier
Predicted: basset - Probability: 99% - True Label: basset
basset
basenji
saluki
pekinese
irish_setter
redbone
beagle
boxer
weimaraner
english_foxhound
Predicted: bluetick - Probability: 28% - True Label: bluetick
bluetick
beagle
yorkshire_terrier
border_terrier
walker_hound
bloodhound
lakeland_terrier
entlebucher
bernese_mountain_dog
basset
Predicted: airedale - Probability: 98% - True Label: airedale
airedale
lakeland_terrier
wire-haired_fox_terrier
irish_terrier
irish_setter
border_terrier
chesapeake_bay_retriever
standard_schnauzer
irish_wolfhound
redbone
Predicted: bedlington_terrier - Probability: 99% - True Label: bedlington_terrier
bedlington_terrier
airedale
lakeland_terrier
miniature_poodle
scottish_deerhound
wire-haired_fox_terrier
chesapeake_bay_retriever
sealyham_terrier
saluki
mexican_hairless