**DL Lab Project:**

**Dog Breed Identification**

* Problem statement:

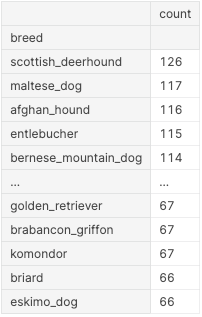
Is it a Dalmatian? A great Dane probably? We’ve all sat behind our screens looking at cute and cuddly dog photos on different social media platforms and wondered, which breed might it be?

The aim of this project is to build a deep learning model to predict the breed of a dog when given an input image.

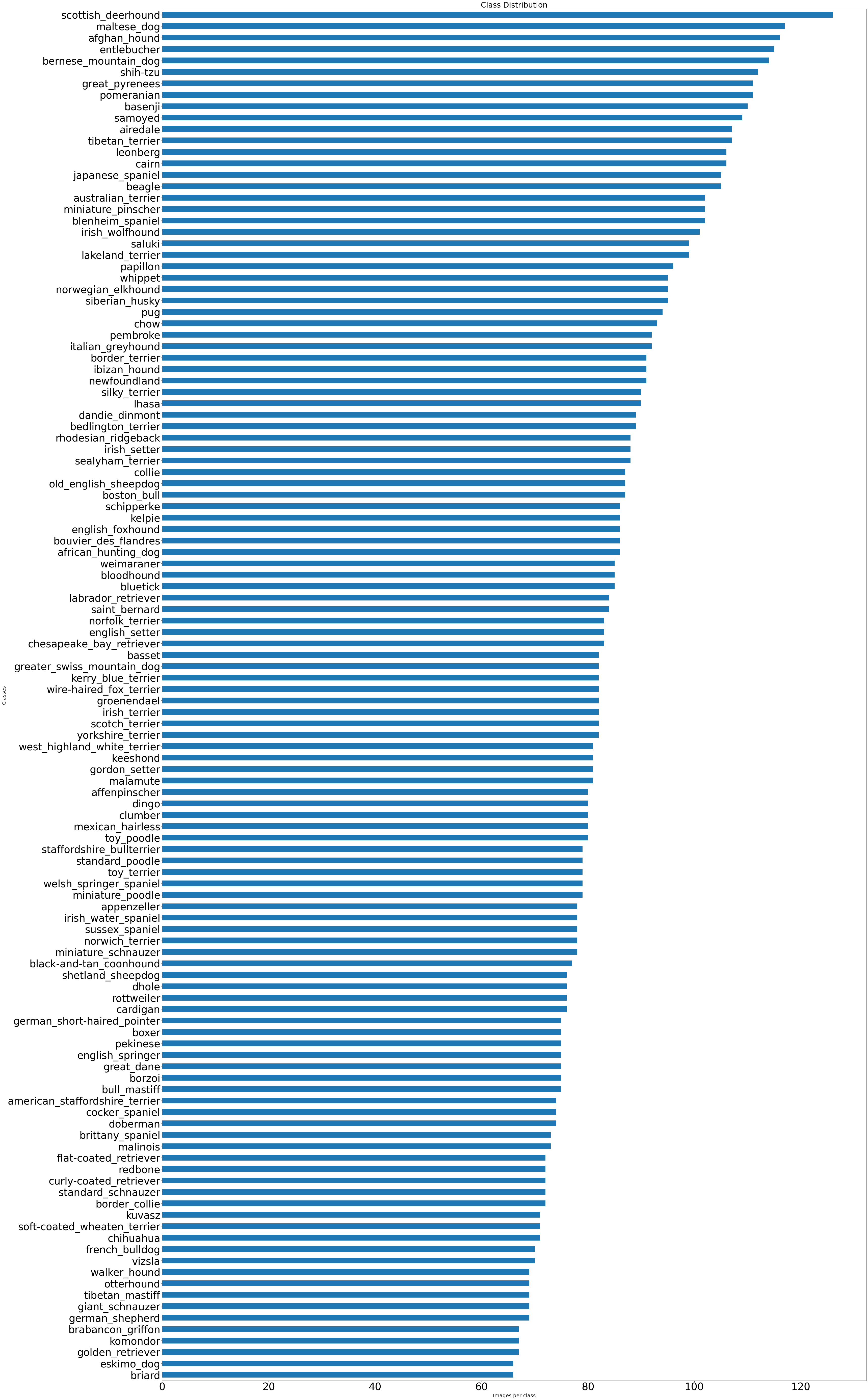
* Project objectives:

To build an efficient and accurate model to predict the breed of the dog in the input image.

* Exploratory analysis:
* Number of dog breeds to classify- 120
* Number of training images- 10222
* Images per breed:



* Breed vs Image count distribution:



* Preprocessing pipeline:
  1. Data loading – load data by providing path
  2. Label encoding- numeric label to each species/breed
  3. Optimization – To make the photo as light as possible for the model to work quickly and efficiently by cropping and rescaling

Phase 2.1:

**Literature review:**

1. **Introduction**

Nowadays, Convolutional Neural Networks (CNNs) are popular in different aspects of deep learning, especially in image recognition. The reason is its property of having general reusable architectures which are viable for different problems.

This model presents the results of CNNs implemented over different architectures using the Kaggle dataset.

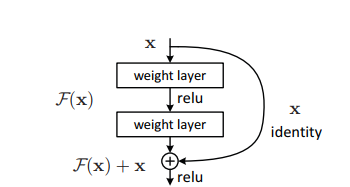
1. **Implemented Works**

* A significant amount of work has been done before in the said field by Whitney Larow (et. Al 2016) which uses an SVM with a linear kernel with an accuracy of 52% and 90% of the time, the correct Breed is in the top ten predictions.
* Csaba Sulyok (et. Al 2018) uses a deep learning model (CNN) and fine-grained image recognition using the inception resnetv2 architecture.
* Suyash S.B (et. Al 2021) implements deep learning using CNN over a VGG-16 architecture to find the predicted output over an input image.

3. Models to be implemented

It has been planned to implement a CNN for this problem over different architectures:

* + **ResNet50** - 50 layers deep. The problem of training very deep networks has been relieved with the introduction of these Residual blocks and the ResNet model is made up of these blocks.

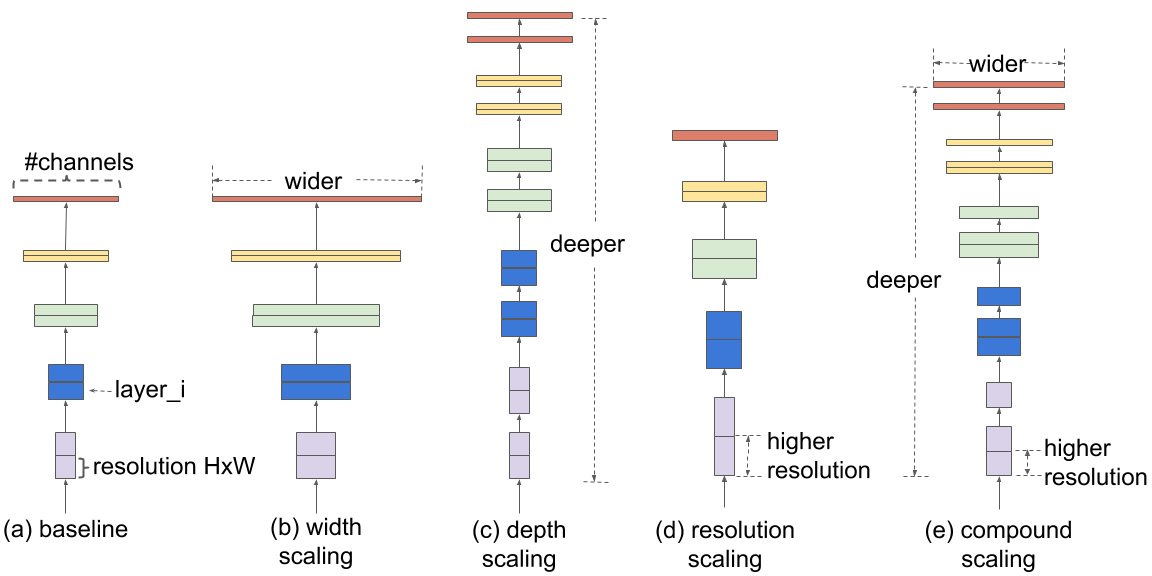


Pros:

* + - * + The ResNet architecture does not need to fire all neurons in every epoch. This greatly reduces the training time and improves accuracy.
        + The complexity of an identical VGG network caused the degradation problem which was solved by residual learning.

Cons :

* + - * + deeper network usually requires weeks for training.
* **EfficientNet** - uniformly scales all dimensions of depth/width/resolution using a *compound coefficient*. Unlike conventional practice that arbitrary scales these factors, the EfficientNet scaling method uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients.

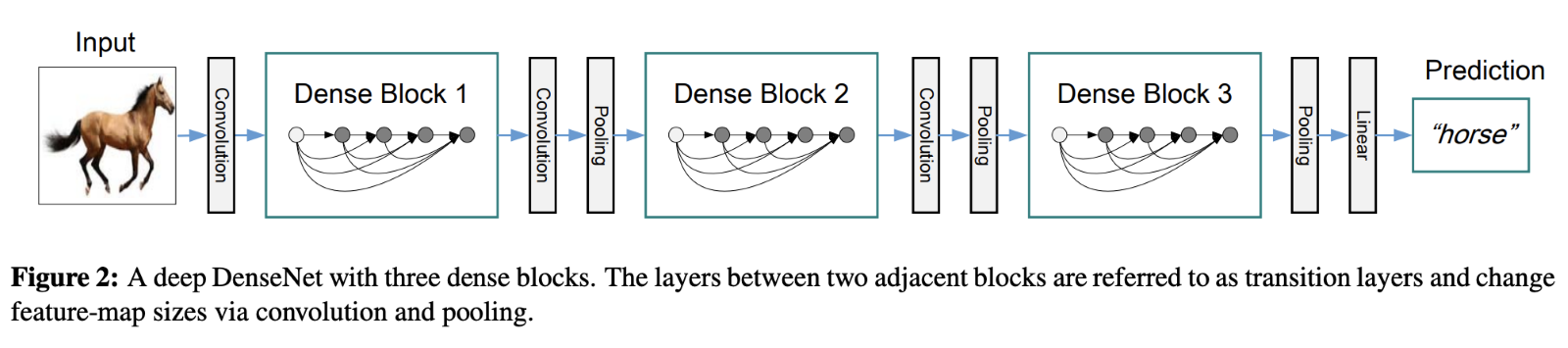


Pros:

* + - * Reduces parameter size and FLOPS by an order of magnitude

Cons:

* + - * + perform poorly on hardware accelerators
* **DenseNet -** utilises dense connections between layers, through Dense Blocks, where we connect all layers (with matching feature-map sizes) directly with each other.

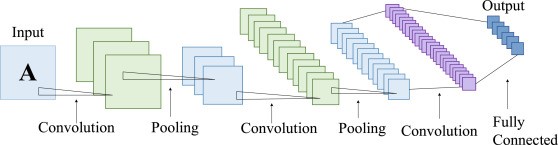


Pros:

* + - * DenseNets simplify the connectivity pattern between layers introduced in other architectures
      * DenseNets *require fewer parameters* than an equivalent traditional CNN, as there is *no need to learn redundant feature maps*.

Cons:

* + - * Densenets tend to face the problem of overfitting
* **InceptionV3** - **Inception-v3** is a convolutional neural network architecture from the Inception family that makes several improvements including using Label Smoothing, Factorized 7 x 7 convolutions, and the use of an auxiliary classifer to propagate label information lower down the network (along with the use of batch normalisation for layers in the sidehead).



Pros:

* It has a deeper network compared to the Inception V1 and V2 models, but its speed isn't compromised.
* It is computationally less expensive.
* It uses auxiliary Classifiers as regularizes.

Cons:

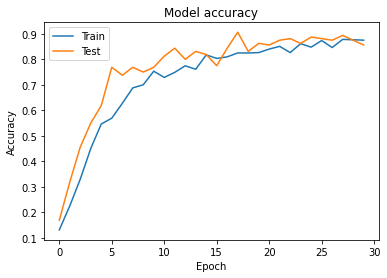
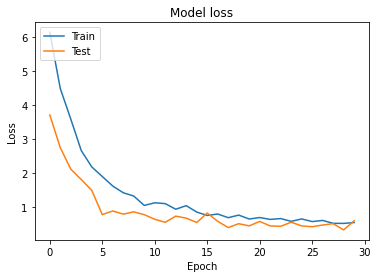
* can quickly accumulate substantial computational cost.

4. Working pipeline

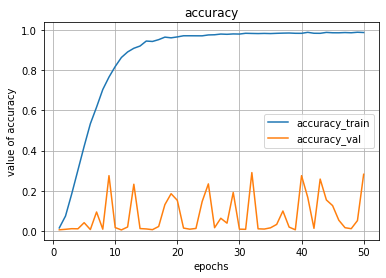
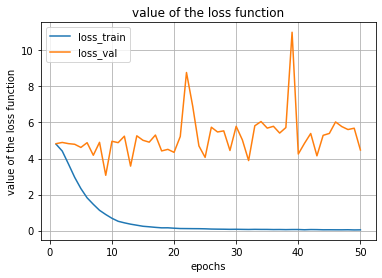
1. Loading dataset
2. Exploratory Data Analysis
3. Preprocessing
4. Label Encoding
5. Splitting data
6. Implementation of models
7. Accuracy comparision

5. Accuracy

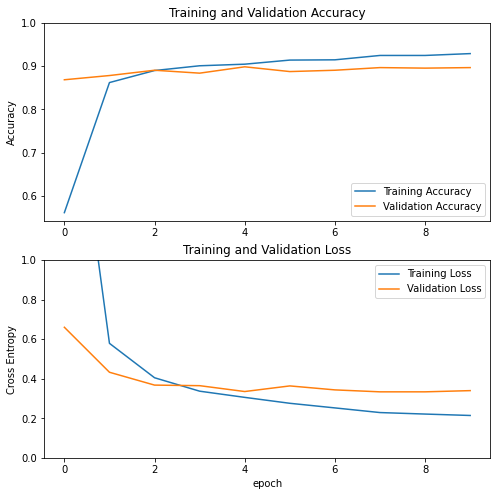
1. RESNET50 – 88%



2. EfficientnetB3 – 98%



3. InceptionV3- 92%



Note – A model using Ensemble methods with logloss scores has also been implemented whose python notebook is attached in the final submission assignment tab.