Final Project Report Template

1. Introduction

1.1. Project overviews

- The project basically classifies the dog breeds using transfer learning.In this project CNN Architectures like VGG-16 , Resnet-50 , Inception and Xception. 1.2. Objectives
- The objective of the project is to classify the different breeds of dogs so as to solve the real-world problems.

2. Project Initialization and Planning Phase

2.1. Define Problem Statement

- The problem statement for the following project can be any of the below two:
- To create an online platform to categorize the dog breed available for adoption based on the uploaded image.
- A veterinarian needs assistance in identifying the breed of the dog brought in for health checkup.

2.2. Project Proposal (Proposed Solution)

Project Overview	
Objective	The objective of the project is to classify and identify the dog breed from images using transfer learning.
Scope	The project has a wider scope. The model can identify the provided 8 breeds of dog. To identify more breeds, we will need larger dataset.
Problem Statement	
Description	The problem statement that we worked on is Dog Breed Identification using the Transfer learning.
Impact	Solving the problem can make the users identify the dog breed accurately without any discomfort.
Proposed Solution	
Approach	The images are taken as input and the breed of the dog is identified. Different CNN architectures such as VGG-16, Resnet50, Inception and Xception were used to identify the breed. Among which Xception gave the best accuracy. So deployed the application with that model.
Key Features	The accuracy of the model is around 99.9% which makes the solution accurate and precise.

2.3. Initial Project Planning

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Project Initiation and Planning	USN-1	Project is initiated and the planning is done	2	High
Sprint-2	Data Collection and Preprocessing Phase	USN-2	Data for the project is collected form Kaggle. It contains images of eight breeds of dog.	1	High
Sprint-3	Model development	USN-3	The transfer learning is used to build the models. VGG-16,	2	High

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Project Initiation and Planning	USN-1	Project is initiated and the planning is done	2	High
Sprint-2	Data Collection and Preprocessing Phase	USN-2	Data for the project is collected form Kaggle. It contains images of eight breeds of dog.	1	High
Sprint-3	Model development	USN-3	The transfer learning is used to build the models. VGG-16,	2	High

3. Data Collection and Preprocessing Phase

3.1. Data Collection Plan and Raw Data Sources Identified

Section	Description
Project Overview	The project identifies the breed of the dog when the image of the dog is uploaded as an input.
Data Collection Plan	The dataset has been collected from Kaggle.
Raw Data Sources Identified	The dataset is from Kaggle. It contains 8 different classes of breed.

Raw Data Sources Template

Source Name	Description	Location/URL	Format	Size	Access Permissions
Dataset 1	It contains 8 classes of dog breeds	https://www.kaggle.com/datasets/mohamedchahed/dog-breeds	Image	86 MB	Public

3.2. Data Quality Report

Data Source	Data Quality Issue	Severity	Resolution Plan
Kaggle	There are different number of images for different dog bread s	Low	Random function is used to separate the testing and training data which makes sure it is evenly distributed.

3.3. Data Preprocessing

Section	Description
Data Overview	The dataset is from Kaggle. It contains 541 images with 8 classes. The eight classes of breed of dog are beagle, bulldog, dalmatian, german-sheperd, husky, labrador-retriever, poodle, rottweiler
Resizing	The image is resized into a target size of 224 x 224 x 3.
Normalization	Normalized pixel value between 0 to 1.
Data Augmentation	Applied Data augmentation techniques such as flipping, rotation, shifting, zooming, or shearing.

Data Preprocessing Code Screenshots	

Loading Data	`# download dataset`
	`!kaggle datasets download -d 'mohamedchahed/dog-breeds'`
	`# unzip dataset`
	`!unzip dog-breeds.zip`

Resizing	`# Define the image dimensions and batch size`
	`img_height = 224`
	`img_width = 224`

Normalization	`train_datagen = ImageDataGenerator(rescale=1./255,`
	`rotation_range=20,`
	`width_shift_range=0.2,`
	`height_shift_range=0.2,`
	`shear_range=0.2,`
	`zoom_range=0.2,`
	`horizontal_flip=True)`
	`test_datagen = ImageDataGenerator(rescale=1./255)`

Data Augmentation	`train_datagen = ImageDataGenerator(rescale=1./255,`
	`rotation_range=20,`
	`width_shift_range=0.2,`
	`height_shift_range=0.2,`
	`shear_range=0.2,`
	`zoom_range=0.2,`
	`horizontal_flip=True)`
	`test_datagen = ImageDataGenerator(rescale=1./255)`

4. Model Development Phase

4.1. Model Selection Report

Model	Description
Model 1	This model is build using the VGG-16 architecture by applying transfer learning. The top layer is replaced with the dense layer with 8 neurons and sigmoid activation function. The model got an accuracy of 100 for 10 epochs.
Model 2	This model is build using the ResNet-50 architecture by applying transfer learning. The top layer is replaced with the dense layer with 8 neurons and sigmoid activation function. The model got an accuracy of 42 for 10 epochs.
Model 3	This model is build using the Inception architecture by applying transfer learning. The top layer is replaced with the dense layer with 8 neurons and sigmoid activation function. The model got an accuracy of 28.5 for 10 epochs.
Model 4	This model is build using the Xception architecture by applying transfer learning. The top layer is replaced with the dense layer with 8 neurons and sigmoid activation function. The model got an accuracy of 100 for 10 epochs.

4.2. Initial Model Training Code, Model Validation and Evaluation Report

Initial Model Training

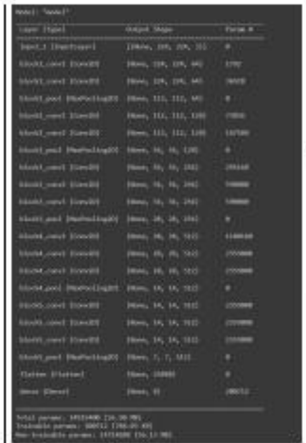
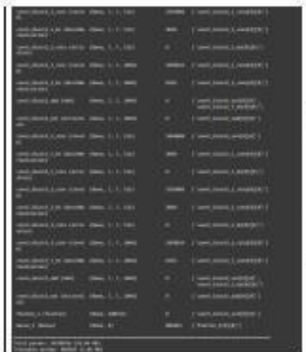
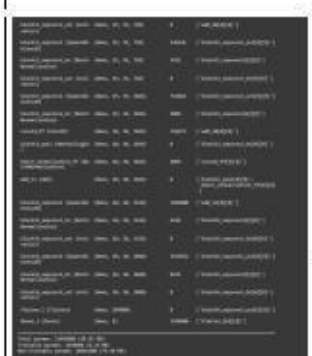
```
vgg16.fit(train_generator,validation_data = test_generator,epochs=10 )
```

```
resnet.fit(train_generator,validation_data = test_generator,epochs=10 )
```

```
inception.fit(train_generator,validation_data = test_generator,epochs=10 )
```

```
xception.fit(train_generator,validation_data = test_generator,epochs=10 )
```

Model Validation and Evaluation Report

Model	Summary	Training and Validation Performance Metrics
Model 1		
Model 2		
Model 3		
Model 4		

5. Model Optimization and Tuning Phase

5.1. Tuning Documentation

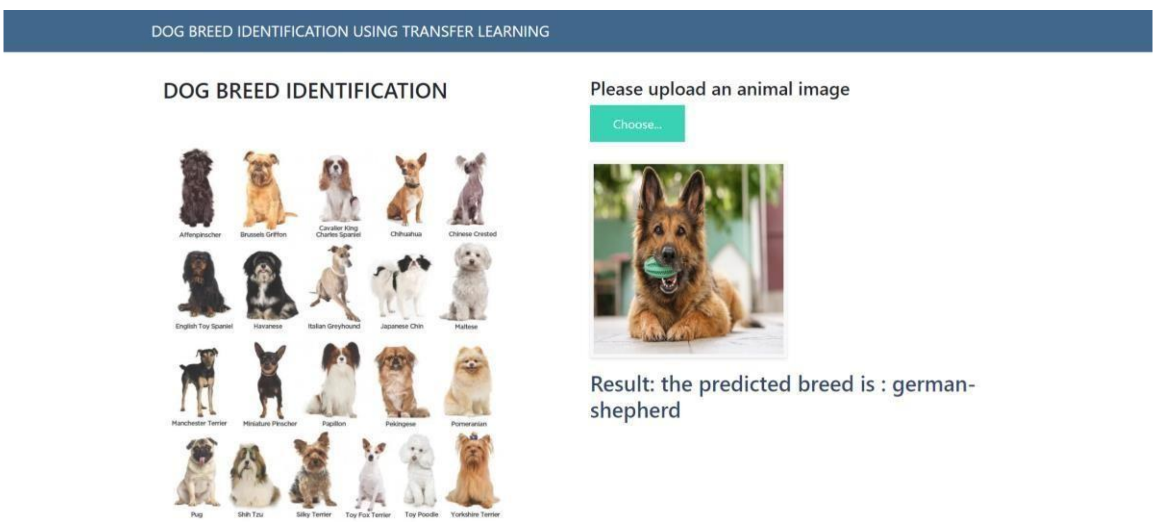
Model	Tuned Hyperparameters
Model 1(VGG-16)	Used Adam optimizer, which gave better accuracy than SGD and ran for 10 epochs.
Model 2(ResNet50)	Used Adam optimizer, which gave better accuracy than SGD and ran for 10 epochs.
Model 3(Inception)	Used Adam optimizer, which gave better accuracy than SGD and ran for 10 epochs.
Model 4(Xception)	Used Adam optimizer, which gave better accuracy than SGD and ran for 10 epochs.

5.2. Final Model Selection Justification

Final Model	Reasoning
Model 4 (Xception)	This model gave batter accuracy than other models.

6. Results

6.1. Output Screenshots



7. Advantages & Disadvantages

<u>Advantages:</u>

- Transfer learning leverages pre-trained models on large datasets (like ImageNet), allowing for quicker convergence and significantly reducing the time needed for training.

- Pre-trained models have learned rich feature representations, which can enhance the accuracy of the classification task, especially when dealing with limited data.
- Transfer learning can achieve good performance even with smaller datasets, which is beneficial if you don’t have access to a large dataset of dog breeds.
- Using complex, deep networks (like ResNet, VGG) becomes feasible without the need to train them from scratch, making advanced architectures accessible.
- The features learned from a broad dataset can help the model generalize better to different types of dog breeds, improving robustness.

Disadvantages:

- Pre-trained models might not be specialized for the task of dog breed classification and may include features irrelevant to this specific task.
- There might be a difference between the source dataset (e.g., ImageNet) and the target dataset (dog breeds), causing a performance drop due to domain shift.
- Pre-trained models are often large and computationally expensive, which might not be suitable for deployment in resource-constrained environments.
- If the target dataset is very small, there's a risk of overfitting to the small dataset despite the use of pre-trained models.
- The quality of your results is heavily dependent on the pre-trained model you choose. If the pre-trained model is not well-suited to your specific task, performance can be suboptimal.

8. Conclusion

- The project uses transfer learning to identify the breed of the dog.
- The Exception architecture gave the best result.
- So, it is used for deploying in the Flask application

9. Future Scope

Enhanced Model Accuracy:

- Continued improvements in deep learning algorithms and architectures could lead to even higher accuracy in classifying dog breeds.

Real-Time Classification:

- Development of lightweight, efficient models that can run on mobile devices, enabling real-time classification through smartphone apps.

Integration with IoT:

- Combining dog breed classification with Internet of Things (IoT) devices, such as smart collars or home cameras, for continuous monitoring and identification.

Explainable AI:

- Incorporating explainability features to provide users with insights into how the model makes its decisions, increasing trust and usability.

10. Appendix

10.1. Source Code import os import shutil import random import numpy as np import matplotlib.pyplot as plt import tensorflow as tf

from tensorflow.keras.layers import Dense from

tensorflow.keras.models import Model from tensorflow.keras.preprocessing

import image

from tensorflow.keras.preprocessing.image import ImageDataGenerator

train_dir = 'train' test_dir

= 'test'

img_height = 299

img_width = 299 batch_size =

32 train_datagen =

ImageDataGenerator(rescal

e=1./255,

rotation_range=20,

width_shift_range=0.2, height_shift_range=0.2,

shear_range=0.2, zoom_range=0.2,

horizontal_flip=True) test_datagen =

ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(train_dir,

target_size=(img_height, img_width),

batch_size=batch_size,

class_mode='categorical', color_mode='rgb')

test_generator = test_datagen.flow_from_directory(test_dir,

```
                                         target_size=(img_height, img_width),
batch_size=batch_size,
class_mode='categorical',                         color_mode='rgb')

from tensorflow.keras.applications.xception import Xception
from tensorflow.keras.layers import Dense , Flatten from tensorflow.keras.models
import Model
xception= Xception(include_top = False,input_shape=(299,299,3))

for layer in xception.layers:
  print(layer) for layer in xception.layers:
layer.trainable = False

x = Flatten()(xception.output) output =
Dense(8,activation = 'softmax')(x)

xception= Model(xception.input,output)
xception.compile(loss = 'categorical_crossentropy',optimizer = 'adam',metrics=['accuracy'])

xception.fit(train_generator,validation_data = test_generator,epochs=10 )
```

10.2. GitHub & Project Demo Link

- Github link
  https://github.com/ssricharmika/Dog_Breed_identification_using_transfer_lear
  ning