PBEL Virtual Internship

Project Title: Image Classification of Cats and Dogs using CNN

Submitted By:

Name: Tejas Pandey
College: United college of engineering and research(UCER).

Submitted To:

Name: Mr. Devnath

Declaration

I hereby declare that this project report titled “Image Classification of Cats and Dogs using CNN” is a result of my own work carried out during the IBM PBEL Virtual Internship. The project is original, and no part of it has been copied or submitted elsewhere for any other course or internship.

Acknowledgement

I would like to express my heartfelt gratitude to my Project guide, Mr. Deepanshu Kumar for their valuable support, guidance, and encouragement throughout the project. Their constructive feedback and constant motivation helped me complete this internship successfully.

I also extend my sincere thanks to the IBM PBEL team for offering this incredible opportunity to gain practical experience. My appreciation also goes to my college, KCC Institute of Technology and Management, and my peers who supported me during the course of this internship.

Table of Contents

1. Introduction

The goal of this project is to develop a **binary image classifier** that can distinguish between cats and dogs. A lightweight pretrained CNN model — **MobileNetV2** — was used to build an efficient and fast classifier. This project also includes a **web-based interface using Gradio**, allowing users to upload or capture images and get instant predictions.

---

2. Technologies Involved

Technology	Purpose
Python	Core language
TensorFlow / Keras	Deep learning model building
TensorFlow Datasets (tfds)	Dataset loading and preprocessing
Google Colab	Development and training environment
Gradio	UI for model deployment
Matplotlib	Visualization of training graphs
MobileNetV2	Pretrained CNN for efficient feature extraction

3. Problem Faced & Solution Implementation

During the development of this project, several challenges were encountered and addressed effectively. One of the major issues was **overfitting**, which started to appear when the model was trained for too many epochs. To resolve this, the number of epochs was limited to 5, and **MobileNetV2's pretrained base was frozen** to prevent unnecessary parameter updates. Another problem was that the model sometimes predicted images that were not cats or dogs as one of the classes. This was mitigated by adding a **confidence threshold** in the prediction logic—if the prediction probability was too close to 50%, the image was marked as "Invalid." Additionally, since Google Colab environments reset after every session, **saving and loading the model to Google Drive** was implemented to retain training progress. Finally, for easier and interactive model deployment, **Gradio** was used to build a clean web-based interface, which simplified testing and user input through webcam or file upload.

4. Dataset Description

- **Source**: Oxford-IIIT Pet Dataset via TensorFlow Datasets
- **Classes**: 37 pet breeds
- **Simplified to**:
  - **Cat** → label < 12
  - **Dog** → label ≥ 12
- **Split**: 80% Training / 20% Validation

5. Model Design

- **Base Model**: MobileNetV2 (frozen)
- **Custom Head**:

    - GlobalAveragePooling2D
    - Dense(128, activation='relu')
    - Dense(1, activation='sigmoid')
- **Optimizer**: Adam
- **Epochs**: 5 (can be increased)
- **Saved Model**: `Google Drive`

6. Gradio Interface

- Accepts both **webcam** and **uploaded** images
- Resizes, preprocesses, and classifies the image
- Uses a **confidence check** to filter out non-cat/dog inputs
- Output format:
    - Cat (95.3% confidence)
    - Dog (88.7% confidence)
    - Invalid image (not a cat or dog)

7. Conclusion

This project demonstrated the successful implementation of a **binary classifier** using transfer learning. MobileNetV2 proved efficient for real-time inference with limited compute. The use of **Gradio for deployment** makes it highly accessible, and **Google Drive integration** simplifies saving/loading across sessions.

This hands-on project strengthened skills in:

- CNNs and transfer learning
- Data preprocessing using tfds
- Model evaluation & deployment
- Building real-world ML applications