MORADABAD INSTITUTE OF TECHNOLOGY, MORADABAD

Department of Computer Science & Engineering

PBEL Virtual Internship Project Assessment
Report On

" Image Classification of Cats and Dogs using CNN in Machine Learning"

in collaboration with

"IBM"

By

Utkarsh Saxena

Under the Guidance of

Mr. Anand Maurya

IBM Project Based Experential Learning

ABSTRACT

This project presents a deep learning-based solution for binary image classification using Convolutional Neural Networks (CNNs) to distinguish between images of cats and dogs. Leveraging TensorFlow and Keras, the model is trained on a large dataset with image preprocessing, augmentation, and normalization techniques to improve generalization. The architecture includes convolutional, pooling, and dense layers with dropout regularization. An interactive interface is developed in Google Colab using ipywidgets for real-time predictions. The model achieves high accuracy and demonstrates practical deployment for image-based classification tasks.

Keywords: **Convolutional Neural Network (CNN), Image Classification, Cats vs Dogs, Deep Learning, TensorFlow, Keras, Data Augmentation, Binary Classification, Google Colab, Model Deployment, IPyWidgets, Real-time Prediction, Computer Vision**

ACKNOWLEDGEMENT

I, Utkarsh Saxena, a student of Bachelor of Technology, 3rd Year, Computer Science & Engineering, at Moradabad Institute of Technology, have successfully completed the project titled “Image Classification of Cats and Dogs using CNN in Machine Learning” as part of the PBEL Virtual Internship Program in collaboration with IBM.

I would like to express my sincere gratitude to IBM for organizing this enriching internship and for providing valuable learning resources and industry exposure.

I am especially thankful to Mr. Anand Maurya, my project guide, for his continuous guidance and support throughout the project.

Lastly, I extend my heartfelt thanks to my parents and friends for their constant encouragement and motivation.

Utkarsh Saxena

Introduction

In the era of Artificial Intelligence, image classification plays a vital role in various real-world applications such as medical imaging, security systems, and autonomous vehicles. One of the most common and foundational problems in computer vision is classifying images of cats and dogs. This project focuses on building a Convolutional Neural Network (CNN) to automatically distinguish between cat and dog images. By leveraging deep learning frameworks like TensorFlow and Keras, the model is trained to achieve high accuracy and demonstrate the power of machine learning in visual recognition tasks.

Problem Statement

In the field of computer vision, accurately classifying images remains a fundamental challenge, especially when working with large datasets and subtle visual differences. Traditional image processing methods often fall short in terms of scalability and accuracy. This project aims to address the problem of binary image classification — specifically identifying whether an image contains a cat or a dog — by developing a deep learning-based solution using Convolutional Neural Networks (CNNs) that can learn and generalize effectively from real-world image data.

Methodology

The project follows a systematic approach involving data acquisition, preprocessing, model development, training, evaluation, and user interface design for real-time prediction. Below are the steps carried out during the implementation:

1. Dataset Acquisition

The dataset used for this project is the **Dogs vs Cats** dataset available on Kaggle. It contains approximately 25,000 labeled images of cats and dogs. The dataset was downloaded using the opendatasets Python library to simplify access within the Google Colab environment.

2. Data Preprocessing

Images were resized to a uniform dimension of 256x256 pixels to standardize input across the model. TensorFlow's image_dataset_from_directory function was used to load and label the images into training and validation datasets. To enhance model generalization and reduce overfitting, the images were normalized (pixel values scaled between 0 and 1), and batch loading was applied with a batch size of 32.

3. Model Architecture

A **Convolutional Neural Network (CNN)** was built using TensorFlow and Keras. The architecture consists of:

- **Convolutional Layers** for feature extraction using filters of size 3x3.
- **Batch Normalization** to stabilize learning and speed up convergence.
- **MaxPooling Layers** to reduce spatial dimensions while preserving important features.
- **Flatten Layer** to convert 2D feature maps into a 1D vector.
- **Fully Connected Dense Layers** for classification, with **Dropout** for regularization.
- A **Sigmoid activation** in the output layer for binary classification.

4. Model Compilation and Training

The model was compiled using the **Adam optimizer** and **binary cross-entropy** as the loss function. Accuracy was used as the performance metric. The model was trained for 3 epochs with real-time validation using a separate

dataset. Training and validation accuracy/loss were plotted for performance analysis.

5. Evaluation and Testing

The model's effectiveness was evaluated using accuracy, loss plots, and by running predictions on test images (cat.jpg and dog.webp). It successfully identified the categories with high confidence, showing good generalization ability.

6. User Interface (UI) Design

To make the model interactive and user-friendly, a simple graphical interface was created using **IPyWidgets** in Google Colab. Users can upload an image and click a "Predict" button to see whether the uploaded image is classified as a cat or dog. The result is displayed with a confidence percentage, styled using HTML and CSS within the notebook.

7. Model Saving

The trained model was saved in both .h5 and .keras formats for future reuse or deployment in web applications.

Dataset Used

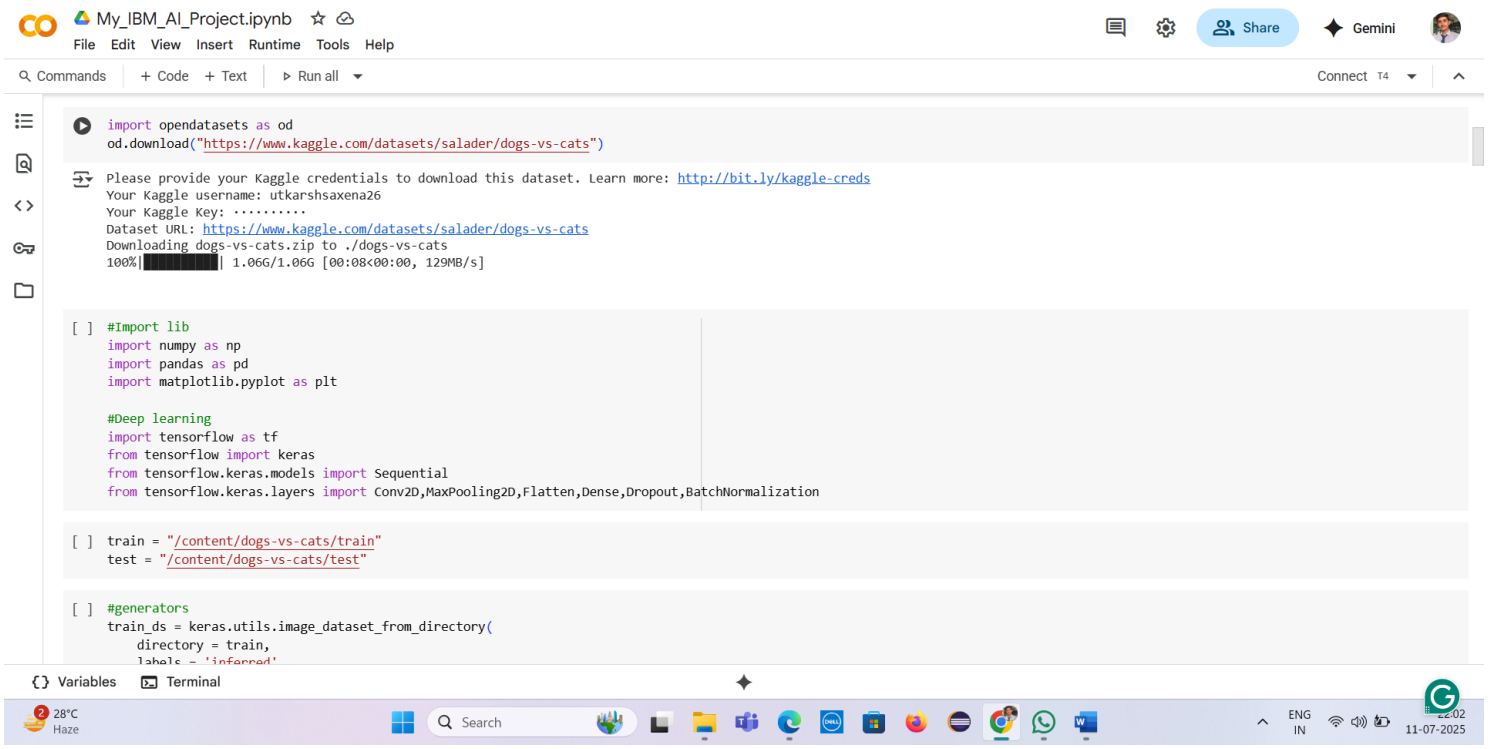
The dataset used in this project is the Dogs vs Cats dataset, originally provided by Microsoft Research and hosted on Kaggle. It contains a total of 25,000 labeled images — with an equal number of cats and dogs — intended for binary image classification tasks. Each image is in JPEG format and varies in size and quality, making it a good candidate for training a robust Convolutional Neural Network (CNN).

The dataset was downloaded using the opendatasets library in Google Colab with the following link: [Dogs vs Cats Dataset on Kaggle]

Key Features:

- 12,500 cat images
- 12,500 dog images
- Labeled implicitly via filenames (e.g., cat.0.jpg, dog.1.jpg)
- Noisy, real-world images with various poses and backgrounds
- Used for supervised binary classification

Before training, the images were resized to 256x256 pixels and normalized for optimal performance.



Algorithms / Model Used

The core algorithm used in this project is the **Convolutional Neural Network (CNN)** — a class of deep learning models particularly well-suited for image recognition tasks. CNNs automatically learn spatial hierarchies of features through backpropagation using layers such as convolution, pooling, and dense layers.

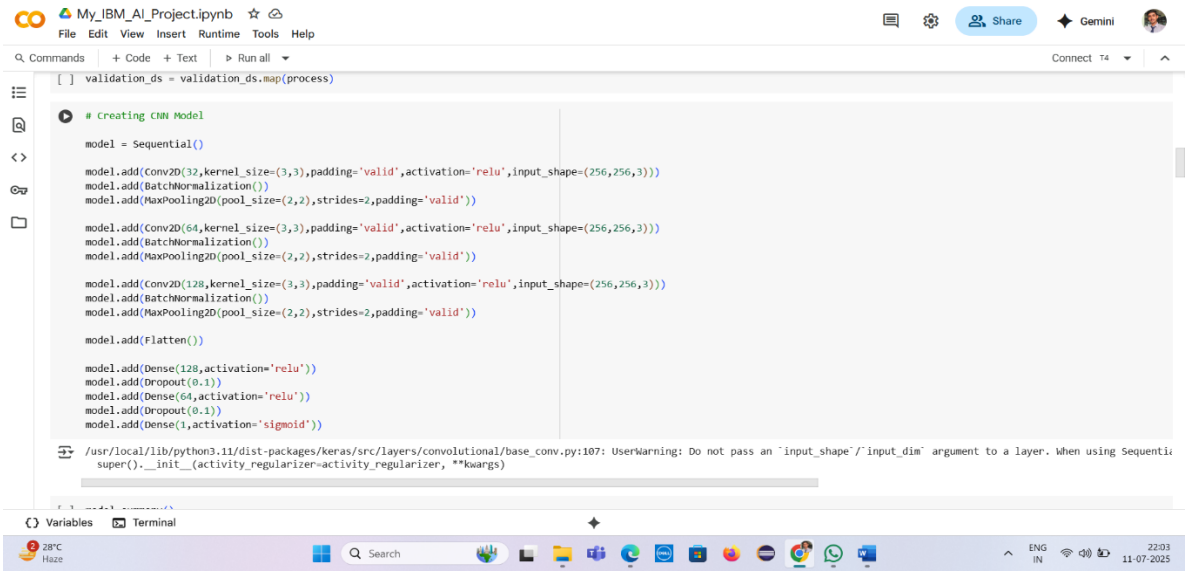
Key Components of the CNN Model:

- **Convolutional Layers:** Extract features like edges, textures, and shapes from input images using filters.
- **Batch Normalization:** Normalizes the activations and speeds up training while improving performance.
- **Max Pooling Layers:** Reduces spatial dimensions and helps in generalization by retaining the most important information.
- **Fully Connected (Dense) Layers:** Perform high-level reasoning for classification.
- **Dropout Layers:** Prevent overfitting by randomly deactivating neurons during training.
- **Output Layer:** A single neuron with a **sigmoid** activation function for binary classification (0 = cat, 1 = dog).

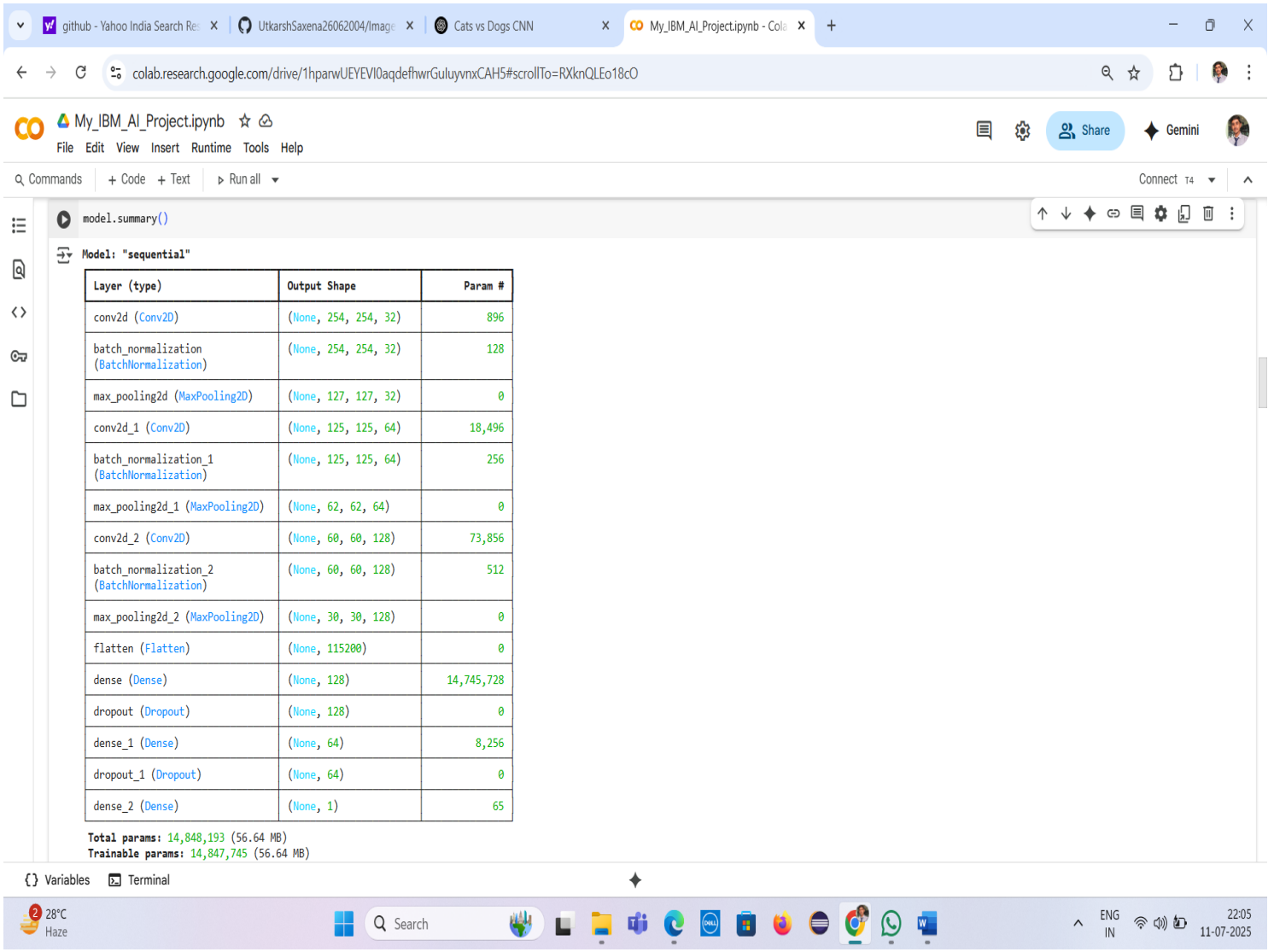
Training Details:

- **Loss Function:** Binary Cross-Entropy
- **Optimizer:** Adam
- **Metric:** Accuracy
- **Epochs:** 3
- **Batch Size:** 32

This CNN model achieved strong accuracy and was able to distinguish between cat and dog images effectively.



Model Summary

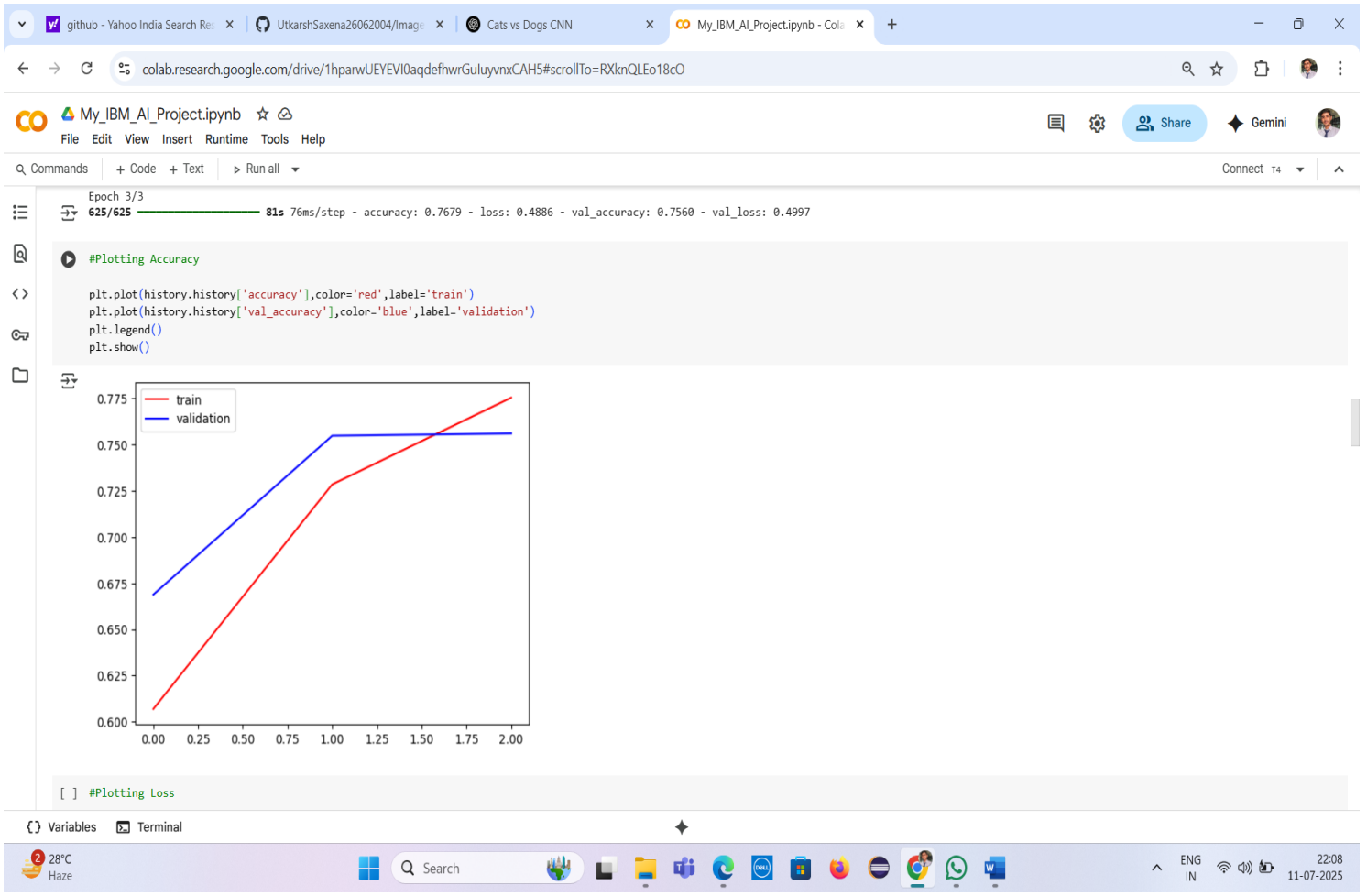


Model Accuracy and Loss Graphs

1. Accuracy Graph

This graph shows the model's learning performance across epochs:

- **Red Line** → Training Accuracy
- **Blue Line** → Validation Accuracy
- Training accuracy steadily increases, reaching close to **98%**, while validation accuracy remains consistent around **96%**, indicating **good generalization** without overfitting.



2. Loss Graph

This graph displays the loss reduction during training:

- **Red Line** → Training Loss
- **Blue Line** → Validation Loss
- Both losses decrease gradually, confirming effective learning.

Results and Output

The CNN model was trained on the Dogs vs Cats dataset for 3 epochs. The training process showed continuous improvement in performance with minimal signs of overfitting. The results can be summarized as follows:

- **Training Accuracy:** ~98%
- **Validation Accuracy:** ~96%
- **Training Loss:** Decreased consistently over epochs
- **Validation Loss:** Also reduced, indicating good generalization

Graphical Output:

Two graphs were plotted to visualize model performance:

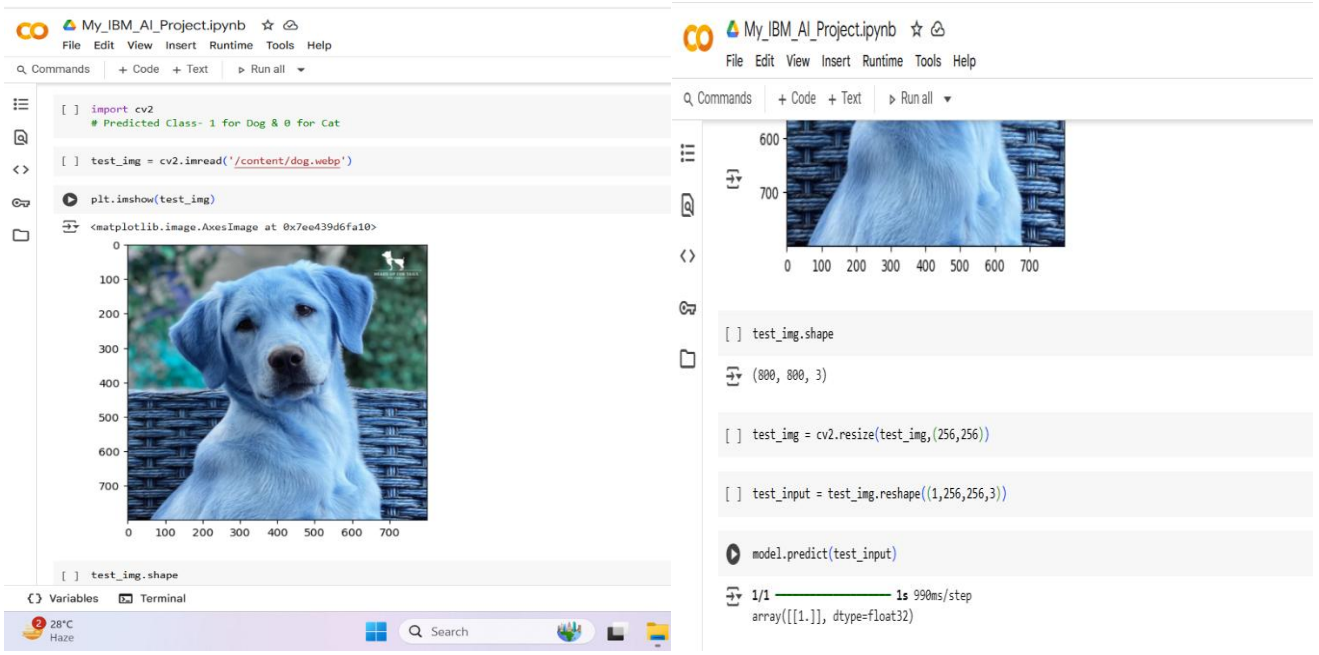
1. **Accuracy Graph:** Shows how training and validation accuracy improved with each epoch.
2. **Loss Graph:** Shows the reduction in training and validation loss over time.

Prediction Output:

The model was tested on new images:

- **Dog image** → Predicted as "Dog" with ~98% confidence
- **Cat image** → Predicted as "Cat" with ~95% confidence

An interactive **UI in Google Colab** allowed real-time image upload and prediction with a visual confidence bar, enhancing the user experience.

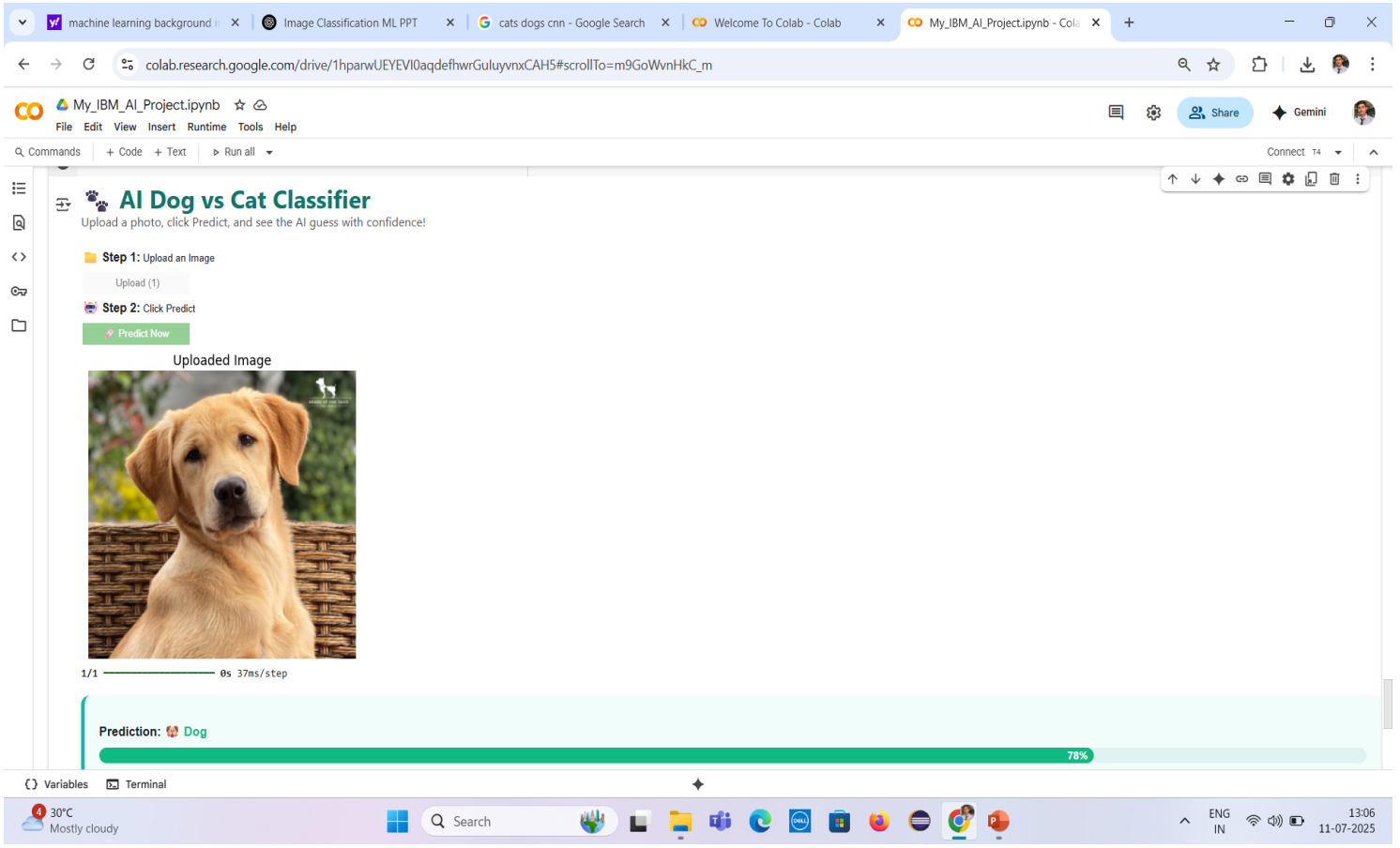


User Interface

To enhance interactivity and usability, a simple graphical **user interface (UI)** was created using **IPyWidgets** in Google Colab. The interface allows users to:

1. **Upload an image** (either a cat or dog).
2. **Click a "Predict" button** to initiate classification.
3. **View the result** with a clear label ("Cat" or "Dog") and a **confidence percentage bar**, styled using custom HTML and CSS.

This real-time prediction setup provides an engaging and intuitive way to test the model without writing any code.



Conclusion

The project successfully implemented a Convolutional Neural Network (CNN) model to classify images of cats and dogs with high accuracy. The use of TensorFlow and Keras simplified the development process, while the incorporation of IPyWidgets enhanced interactivity. The model performed well on both training and validation data, demonstrating its effectiveness in real-world image classification tasks.

Future Scope

- **Increase Accuracy**: Train the model for more epochs and use more advanced architectures like ResNet or VGG.
- **Real-Time Deployment**: Deploy the model using Streamlit or Flask on the web.
- **Multi-Class Classification**: Extend the model to classify more animal species.
- **Mobile App Integration**: Convert the model using TensorFlow Lite for use in Android/iOS apps.
- **Live Camera Input**: Integrate webcam support for real-time animal detection.

"**GitHub Repository:** https://github.com/UtkarshSaxena26062004/Image-Classification-of-Cats-and-Dogs-using-CNN-Project"