DOG BREED IDENTIFICATION USING TRANSFER LEARNING

GROUP MEMBERS:

Singuluri Charishma: 22P31A04H9

Bhavani Kunireddy:23P35A0408

Munduri Veera Bhadra Naga Anusha:23P35A0414

Nakka Jayavardhan:23P35A0416

1.INTRODUCTION:

1.1 PROJECT OVERVIEWS:

The Dog Breed Identification project uses transfer learning techniques to recognize different dog breeds from images. In this project, a pre-trained Convolutional Neural Network (CNN) model such as VGG16 or ResNet50 is utilized and further trained using a dog breed dataset to enhance its performance. The workflow includes image preprocessing, loading and modifying the pre-trained model, training it with labelled image data, and testing its effectiveness. By using previously learned features from large image datasets, this method helps in reducing training time while improving the accuracy of breed prediction.

1.2 OBJECTIVES :

The main goal of this project is to build a machine learning system that can correctly identify the breed of a dog based on an input image. Transfer learning is applied to make the training process faster and more accurate. Various tools and libraries such as Python, NumPy, Pandas, Keras, and Seaborn are used to develop the system efficiently. The dataset includes labeled images of different dog breeds, where the image details are stored in a directory and their corresponding labels are maintained in a CSV file. The final outcome is a model that delivers high accuracy in breed classification.

2. PROJECT INITIALIZTION AND PLANNING PHASE:

2.1 DEFINE PROBLEM STATEMENT:

The aim of the Dog Breed Identification project is to design a reliable and efficient machine learning model that can determine the breed of a dog from an image. Using a dataset containing labeled dog pictures, the process involves preparing the data, applying transfer learning techniques for training, and evaluating the model's performance. The objective is to create an automated solution that simplifies breed identification, which can be useful in areas like veterinary services, pet adoption centers, and animal research.

2.2 PROJECT PROPOSAL (PROPOSED SOLUTION):

The Dog Breed Identification System aims to develop an advanced, user-friendly platform capable of accurately identifying dog breeds from images. This system will leverage machine learning and image recognition technologies to meet the diverse needs of pet owners, veterinarians, shelters, breeders, trainers, law enforcement, and more. The final model will be evaluated and validated to ensure its effectiveness in accurately identifying dog breeds from images.

2.3 INITIAL PROJECT PLANNING :

Project Scope and Objectives:

- **Dataset Preparation:**
    - Gather and preprocess labelled dog images
    - Apply data augmentation to improve dataset diversity
- **Model Selection:**
    - Choose a pre-trained convolutional neural network (CNN) model such as VGG16, ResNet50, or InceptionV3 for transfer learning.
- **Model Training:**
    - Fine-tune the selected pre-trained model on the dog breed dataset.
    - Optimize the model's parameters and architecture for better performance.
- **Model Evaluation:**
    - Evaluate the model's accuracy and robustness using validation and test datasets.
    - Implement performance metrics such as accuracy, precision, recall, and F1-score.
- **Deployment:**
    - Develop a user-friendly interface or API for practical use of the model.
    - Deploy the model for real-world applications like veterinary care and pet adoption.

3.DATA COLLECTION AND PREPROCESSING PHASE

3.1 DATA COLLECTION PLAN AND RAW DATA SOURCES IDENTIFIED :

DATA COLLECTION PLAN :

Like we can determine the number of dog breeds to be included and then we can decide the minimum number of images per breed to ensure a balanced dataset. We can collect the images from the kagle dataset and we can also supplement with the additional images from sources like flickr and google images too. Data preprocessing and data augmentation and storage can be done in csv file including image filenames and corresponding breed labels.

RAW DATA SOURCES:

The raw data can be seen in Stanford dogs dataset, it contains 20,000 images of 120 dog breeds with labels and bounding box annotations. Kaggle contains over 10,000 images of 120 breeds with a csv file providing breed labels. And also flickr has high quality images with the ability to search for specific dog breeds. We can also get images from pet adoption websites.

3.2 Data Quality Report :

The completeness can be done by ensuring all dog breeds in the dataset are adequately represented with a sufficient number of images per breed. We can confirm the accuracy of image is correctly labelled with the appropriate dog breed. The standardize image formats and dimensions to maintain uniformly. We can verify that all images are of dogs and not of other animals or irrelevant objects. Using recent and up to date images to ensure the dataset reflects current appearances and characteristics of dog breeds. Ensuring the dataset includes images of dogs in various poses, environments and lighting conditions to improve the models robustness.

3.3 Data Exploration and Preprocessing:

Data Exploration:

Visual Inspection**:** Examine sample images from each breed to understand variations in image quality, size, and labeling accuracy. Statistical Analysis**:** Analyze the distribution of images across different breeds to identify any imbalances or biases in the dataset.

Data Preprocessing:

Resizing**:** Standardize image dimensions to ensure uniform input size for the model (e.g., 224x224 pixels). Normalization**:** Scale pixel values to a range suitable for the neural network, typically between 0 and 1. splitting**:** Divide the dataset into training, validation, and test sets to evaluate model performance effectively.

4 Model Development Phase:

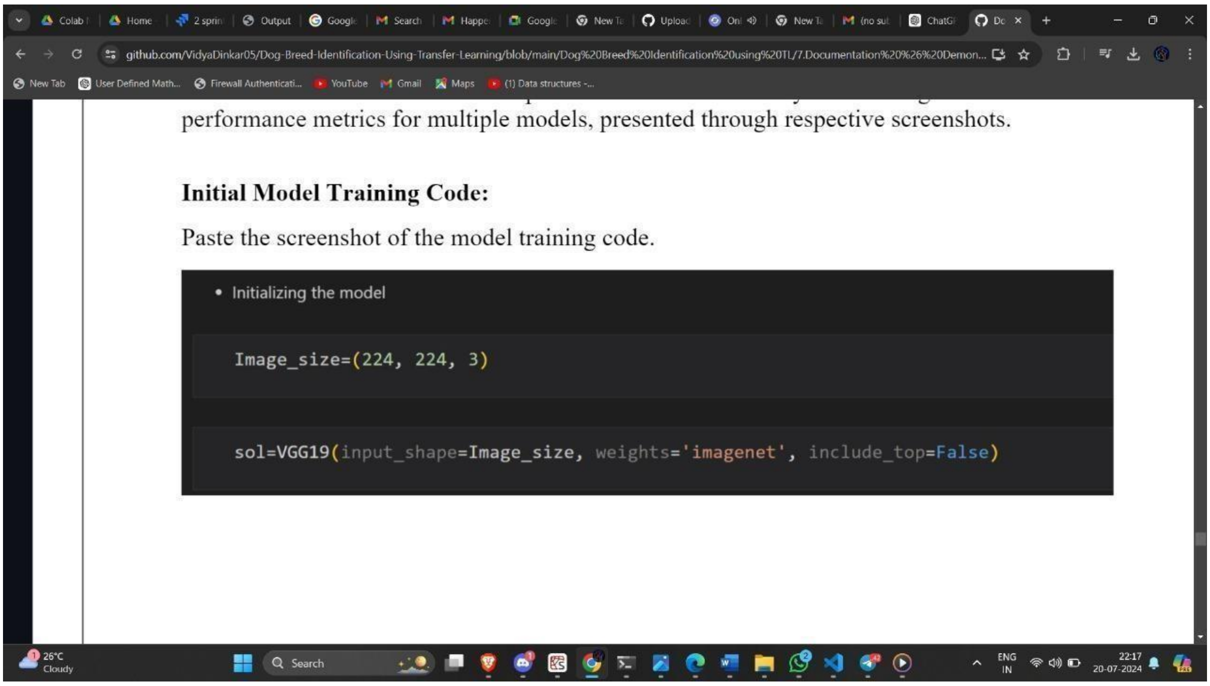
4.1. Model Selection Report

When choosing a model for classification, evaluate various pre-trained convolutional neural network (CNN) models (e.g., VGG16, ResNet50, InceptionV3) based on key metrics such as accuracy, precision, recall, and F1-score. Assess the complexity and size of each model, including the number of parameters and computational requirements. By choose a model that can be easily adapted or scaled to include additional breeds or handle variations in the dataset. By addressing these points in the model selection report, the project ensures the choice of a robust, efficient, and effective model for dog breed identification.

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

Initial Model Training Code, Model Validation and Evaluation Report

Model Training Code:



performance metrics for multiple models, presented through respective screenshots.

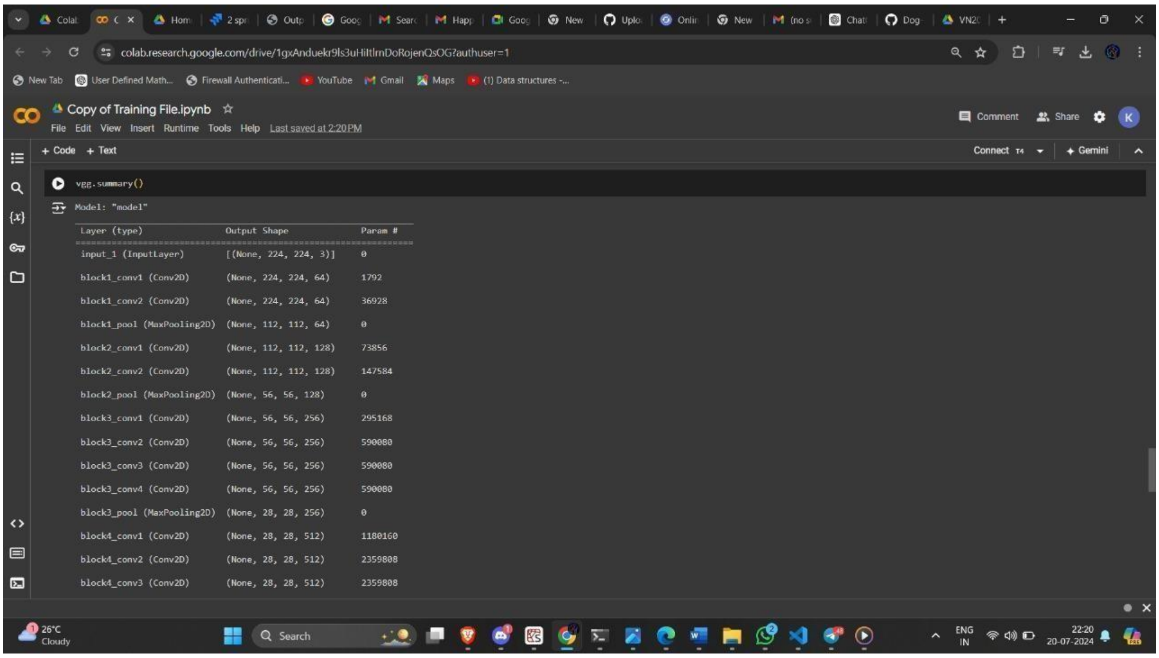
Initial Model Training Code:

Paste the screenshot of the model training code.

- Initializing the model

```
Image_size=(224, 224, 3)
```

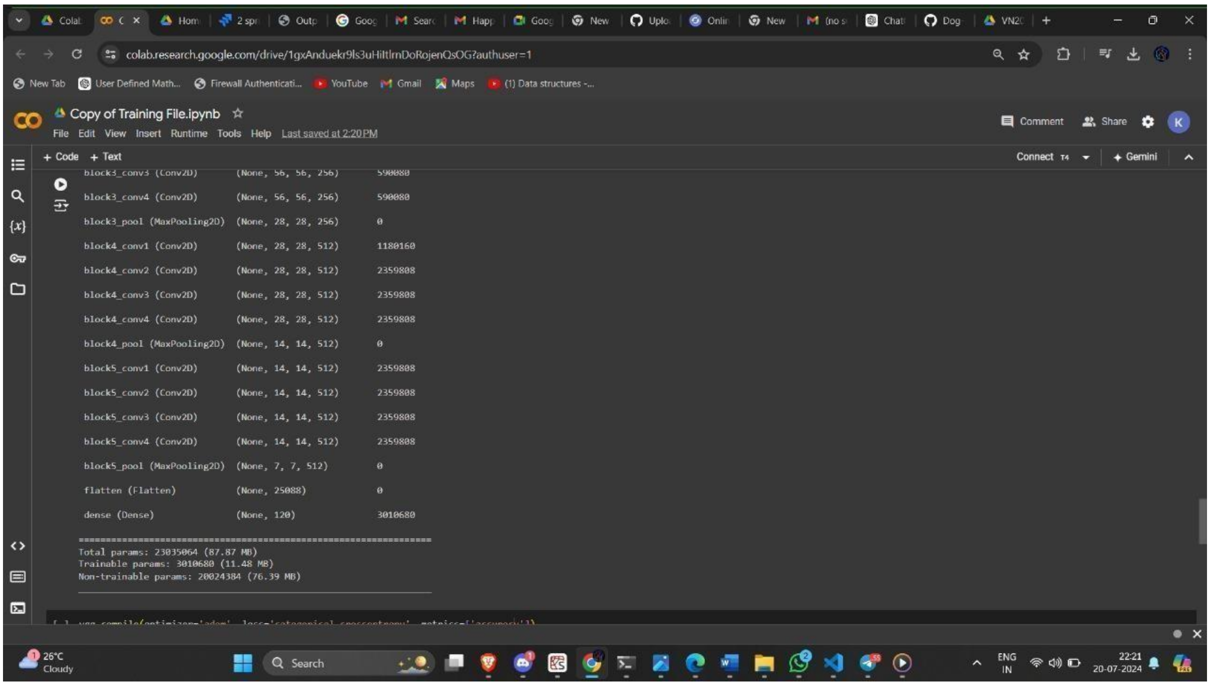
```
sol=VGG19(input_shape=Image_size, weights='imagenet', include_top=False)
```



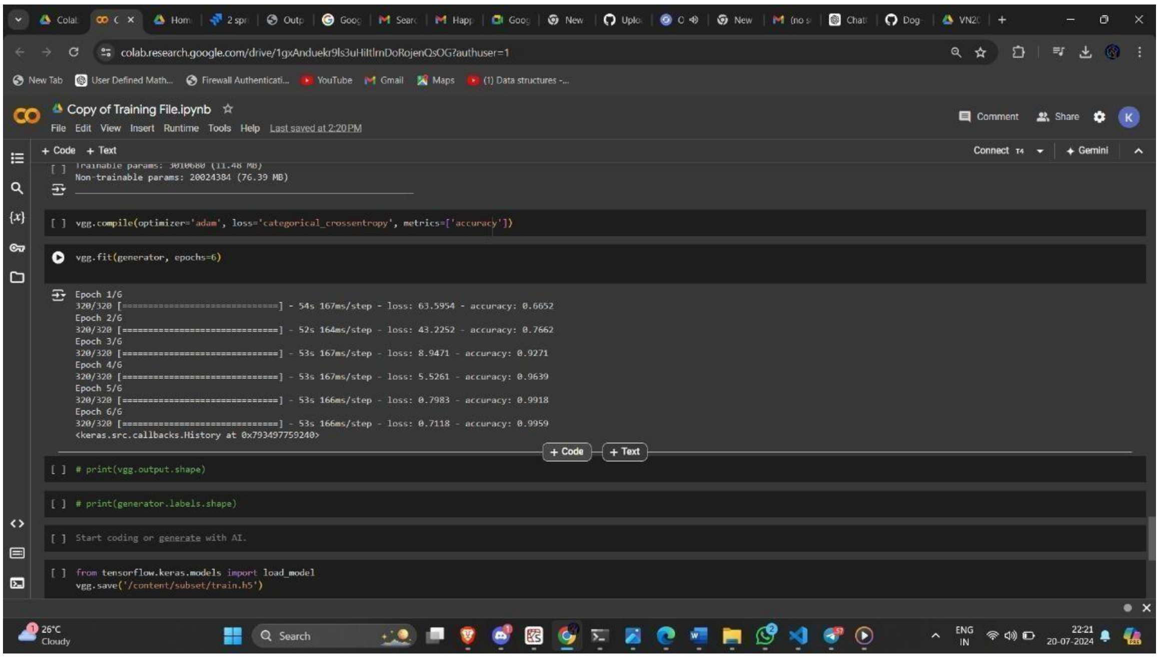
Copy of Training File.ipynb

```
vgg.summary()
```

```
Model: "model"
 ____________________________________________________________________
 Layer (type)                Output Shape              Param #
=====================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv4 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808
```


colab.research.google.com/drive/1gxAnduekr9ls3uHiltlmDoRojenQsOG?authuser=1
Copy of Training File.ipynb
File Edit View Insert Runtime Tools Help Last saved at 2:20 PM
+ Code + Text
Comment Share
Connect T4
Gemini
26°C Cloudy
Search
22:21 20-07-2024

```
 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv4 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv4 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv4 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 120)               3010680

=================================================================
Total params: 23035064 (87.87 MB)
Trainable params: 3010680 (11.48 MB)
Non-trainable params: 20024384 (76.39 MB)
_________________________________________________________________
```


colab.research.google.com/drive/1gxAnduekr9ls3uHiltlmDoRojenQsOG?authuser=1
Copy of Training File.ipynb
File Edit View Insert Runtime Tools Help Last saved at 2:20 PM
+ Code + Text
Comment Share
Connect T4
Gemini
26°C Cloudy
Search
22:21 20-07-2024

```
Trainable params: 3010680 (11.48 MB)
Non-trainable params: 20024384 (76.39 MB)
_________________________________________________________________
```

```
[ ] vgg.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

```
vgg.fit(generator, epochs=6)
```

```
Epoch 1/6
320/320 [==============================] - 54s 167ms/step - loss: 63.5954 - accuracy: 0.6652
Epoch 2/6
320/320 [==============================] - 52s 164ms/step - loss: 43.2252 - accuracy: 0.7662
Epoch 3/6
320/320 [==============================] - 53s 167ms/step - loss: 8.9471 - accuracy: 0.9271
Epoch 4/6
320/320 [==============================] - 53s 167ms/step - loss: 5.5261 - accuracy: 0.9639
Epoch 5/6
320/320 [==============================] - 53s 166ms/step - loss: 0.7983 - accuracy: 0.9918
Epoch 6/6
320/320 [==============================] - 53s 166ms/step - loss: 0.7118 - accuracy: 0.9959
<keras.src.callbacks.History at 0x793497759240>
```

```
[ ] # print(vgg.output.shape)
```

```
[ ] # print(generator.labels.shape)
```

```
[ ] Start coding or generate with AI.
```

```
[ ] from tensorflow.keras.models import load_model
    vgg.save('/content/subset/train.h5')
```

5. Model Optimization and Tuning Phase

- **Hyperparameter Tuning:**
    - **Learning Rate:** Adjust the learning rate to find the optimal value that balances convergence speed and stability.
    - **Batch Size:** Experiment with different batch sizes to determine the impact on training efficiency and model performance.
    - **Epochs:** Set an appropriate number of epochs to ensure sufficient training without overfitting.

- **Regularization Techniques:**
    - **Dropout:** Apply dropout to prevent overfitting by randomly dropping neurons during training.
    - **L2 Regularization:** Use L2 regularization to penalize large weights and reduce model complexity.

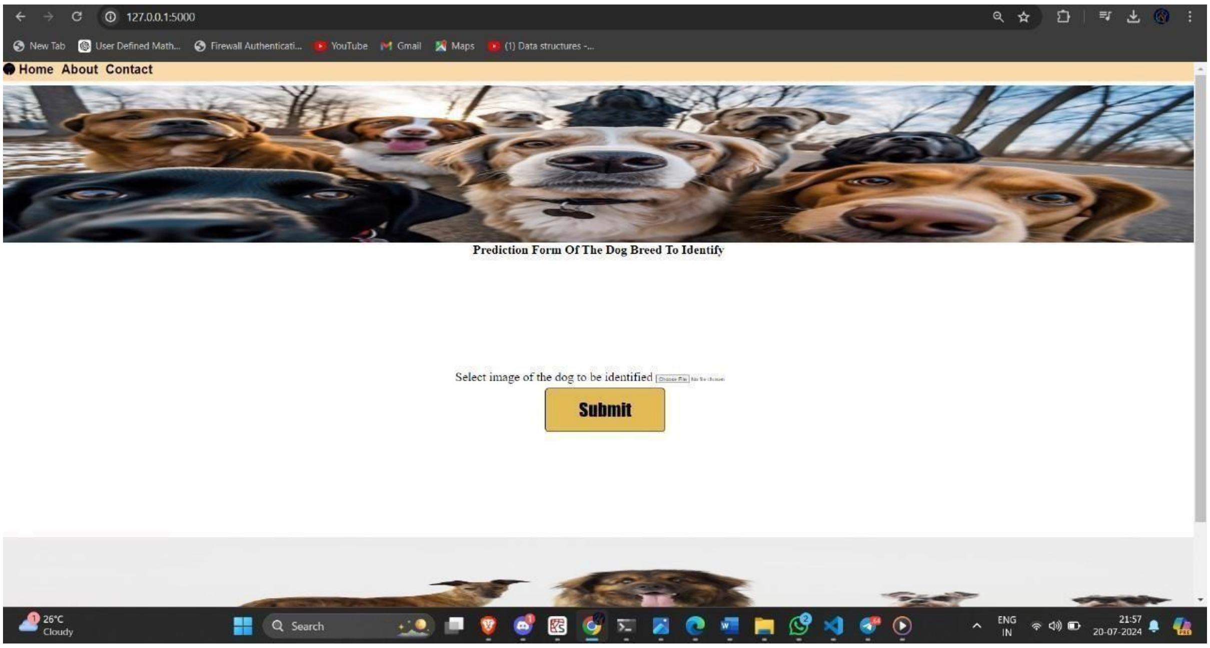
- **Model Architecture Adjustments:**
    - **Layer Modification:** Experiment with adding or removing layers, or adjusting layer sizes, to improve model performance.
    - **Transfer Learning Adjustments:** Fine-tune different layers of the pre-trained model (e.g., feature extractor vs. classifier layers) to adapt to the specific dog breed dataset.

- **Data Augmentation:**
    - Enhance the dataset with techniques such as rotation, flipping, and color adjustments to improve model robustness and generalization.

- **Optimization Algorithms:**
    - **Optimizer Selection:** Test different optimization algorithms (e.g., Adam, SGD, RMSprop) to find the best fit for the model and dataset.
    - **Learning Rate Schedulers:** Use learning rate schedulers to adjust the learning rate dynamically during training for better convergence.

6. Result:

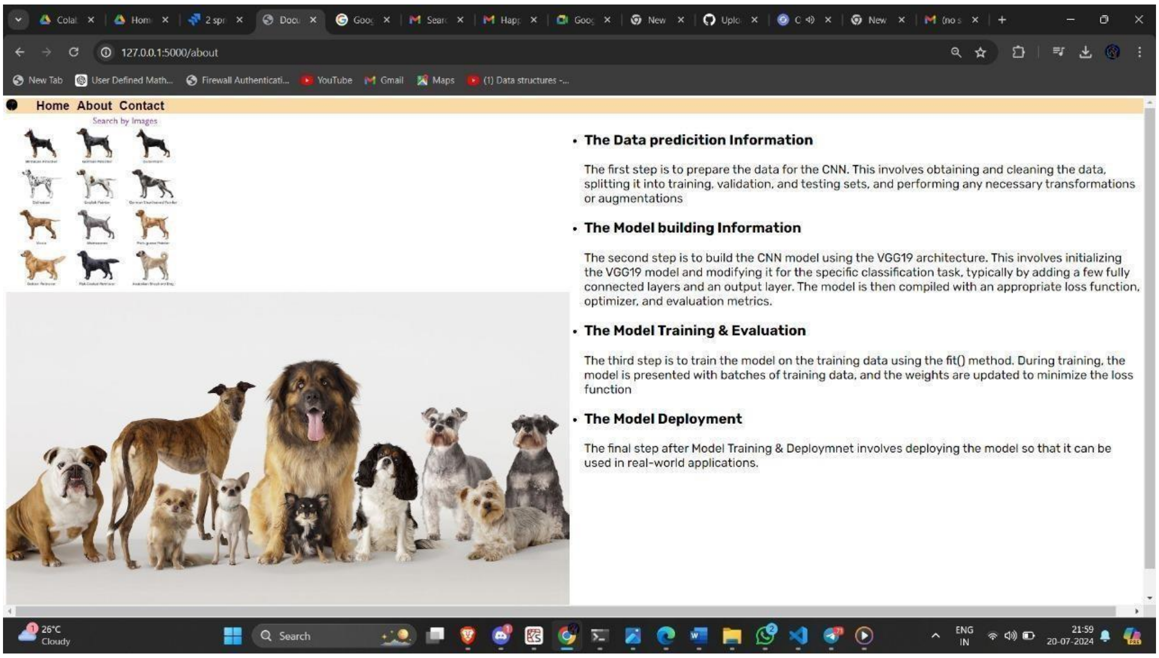
HomePage:



Prediction Form Of The Dog Breed To Identify

Select image of the dog to be identified

About Page:



- **The Data predicition Information**

  The first step is to prepare the data for the CNN. This involves obtaining and cleaning the data, splitting it into training, validation, and testing sets, and performing any necessary transformations or augmentations

- **The Model building Information**

  The second step is to build the CNN model using the VGG19 architecture. This involves initializing the VGG19 model and modifying it for the specific classification task, typically by adding a few fully connected layers and an output layer. The model is then compiled with an appropriate loss function, optimizer, and evaluation metrics.

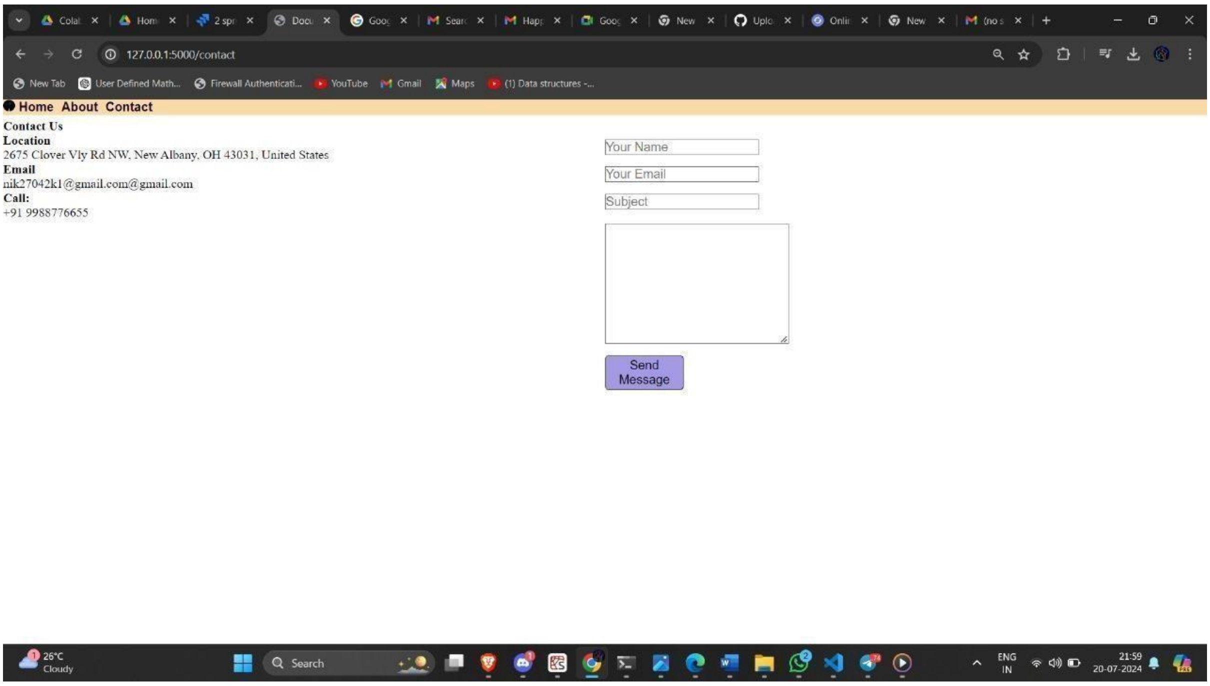
- **The Model Training & Evaluation**

  The third step is to train the model on the training data using the fit() method. During training, the model is presented with batches of training data, and the weights are updated to minimize the loss function

- **The Model Deployment**

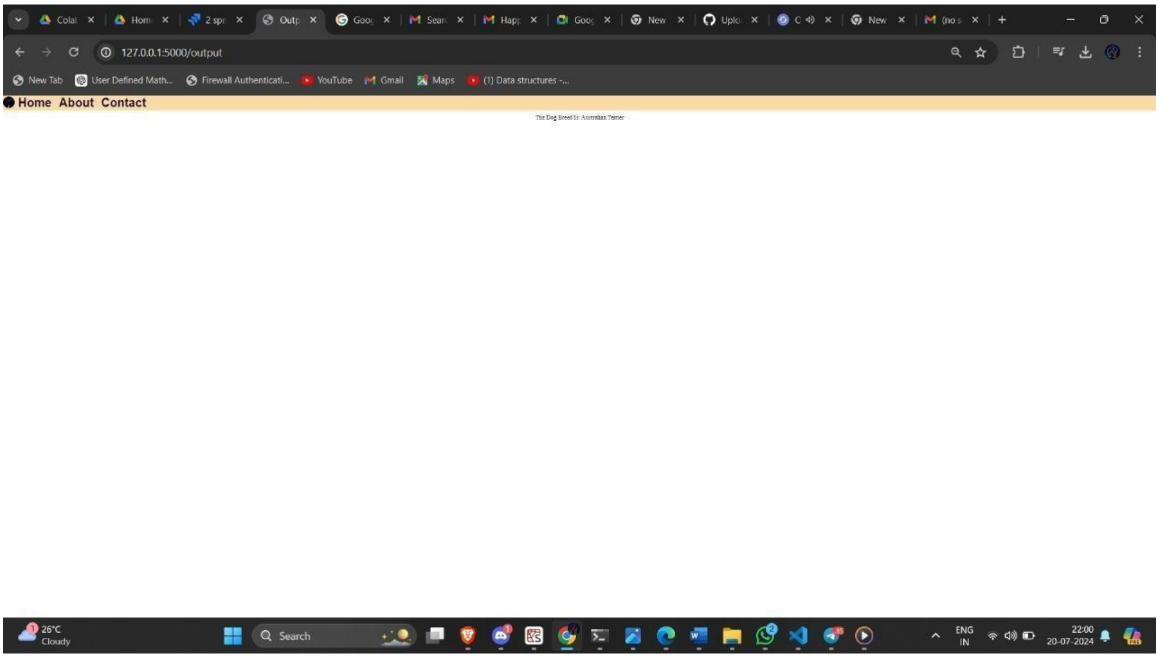
  The final step after Model Training & Deploymnet involves deploying the model so that it can be used in real-world applications.

Contact Page:



Contact Us
Location
2675 Clover Vly Rd NW, New Albany, OH 43031, United States
Email
nik27042k1@gmail.com@gmail.com
Call:
+91 9988776655

Output Page:



The Dog Breed Is: Australian Terrier

7 ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- ### Integration with Flask:

Using Flask for integration is ideal due to its lightweight nature in Python web frameworks. It's versatile, simplifying the creation of user-friendly interfaces for CSV prediction models. Flask's straightforwardness accelerates model development and deployment significantly.

- ### Scalability:

Scalability is a strong suit of Flask-based models, designed to manage numerous concurrent user queries for real-time predictions effectively. This makes the model highly applicable in clinical settings with extensive patient populations, ensuring robust performance and scalability.

DISADVANTAGES

- ### Complexity of Model Building:

Selecting, training, and fine-tuning additional models is essential when incorporating them into a voting classifier. Achieving optimal performance requires careful optimization of each individual model and finding the right balance between them can be challenging.

- ### Deployment and Upkeep:

Managing a Flask-based application entails ongoing monitoring, maintenance, and updates on the server. This can introduce additional complexity and potentially higher maintenance costs compared to deploying a standalone model.

8 APPLICATIONS :

- **Veterinary Care:**
  - Assist veterinarians in quickly identifying dog breeds, which can aid in diagnosing breed-specific health issues and recommending appropriate care.
- **Pet Adoption and Shelters:**
  - Help animal shelters and adoption centers accurately identify dog breeds, improving breed-specific matchmaking and adoption processes.
- **Pet Insurance:**
  - Facilitate insurance companies in assessing risk and coverage by identifying dog breeds, which can be used to tailor insurance plans and premiums.
- **Breed Research and Conservation:**
  - Support researchers and conservationists in studying breed characteristics, population distributions, and genetic diversity.
- **Dog Breeder Services:**
  - Assist breeders in confirming the breed of dogs in their litters, ensuring accurate breed documentation and certification.
- **Mobile Applications:**
  - Provide a user-friendly tool for dog owners and enthusiasts to identify breeds in real-time using mobile apps, enhancing user engagement and education.

- ## 9 CONCLUSION :

In summary, The Dog Breed Identification project successfully demonstrates the power of transfer learning and advanced machine learning techniques in classifying dog breeds from images. By leveraging pre-trained convolutional neural networks, the project achieves high accuracy and efficiency in recognizing a wide range of dog breeds. Through thorough data exploration, preprocessing, and model optimization, the project ensures robust performance and generalization. The practical applications of this technology span veterinary care, pet adoption, and mobile applications, offering valuable tools for both professionals and dog enthusiasts. Overall, the project highlights the potential of AI to enhance our interactions with and understanding of the canine world, paving the way for future advancements in image classification and machine learning.

10 FUTURE SCOPE:

The future scope in dog breed identification can be more helpful and profitable project it can be extended the model to identify a broader range of dog breeds including rare and mixed breeds by expanding the dataset and fine tuning the model. Incorporate more advanced architectures or hybrid models to enhance classification accuracy and reduce error rates. It can develop real time identification capabilities for mobile and web applications allowing users to classify dog breeds using their smartphones. We can also implement advanced data augmentation techniques such as generative adversarial networks to further enhance dataset diversity and model robustness.