🐶 DOG BREED IDENTIFICATION USING DEEP LEARNING

📌 Project Description

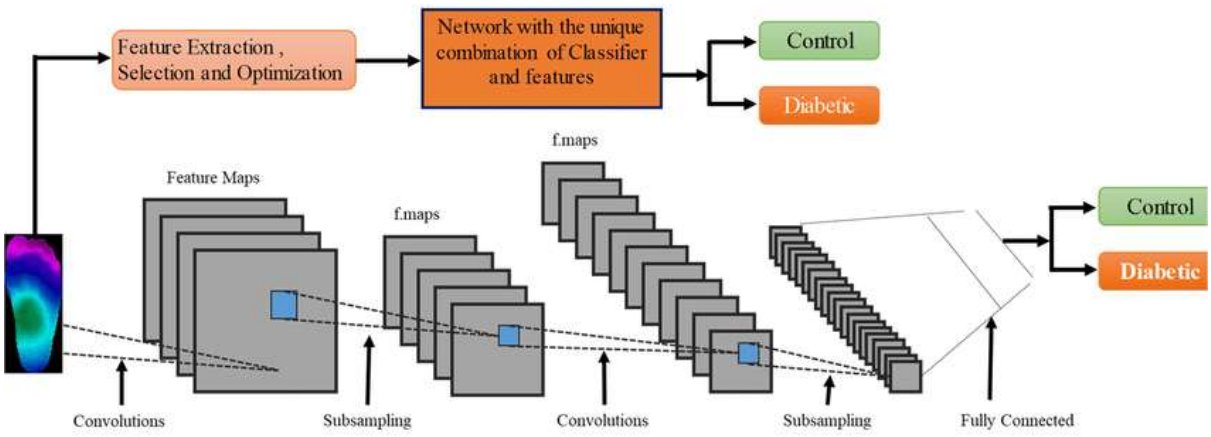
Dog Breed Identification is a deep learning-based image classification project where the system predicts the breed of a dog from an uploaded image.

This project uses:

- Convolutional Neural Networks (CNN)
- Transfer Learning (VGG19)
- TensorFlow & Keras
- Flask Web Application

The trained model is saved as dogbreed.h5 and deployed using Flask for real-time prediction.

🏗 Technical Architecture



Architecture Flow:

1. User uploads dog image
2. Flask receives image
3. Image preprocessing (Resize + Normalize)

4. Image passed to VGG19 model
5. Model predicts breed
6. Result displayed on webpage

📁 Project Structure

```
DOG_BREED_PREDICTION/
|
├── static/
├── templates/
|   ├── index.html
|   ├── predict.html
|   └── output.html
├── uploads/
├── dogbreed.h5
├── app.py
└── DogBreed.docx
```

📊 Milestone 1: Dataset

The dataset contains images of various dog breeds such as:

- Golden Retriever
- Labrador
- German Shepherd
- Pug
- Bulldog

Images are organized into folders based on breed names.

📈 Milestone 2: Data Preprocessing

Steps performed:

- Resize images to 224x224
- Normalize pixel values (1./255)
- Data Augmentation

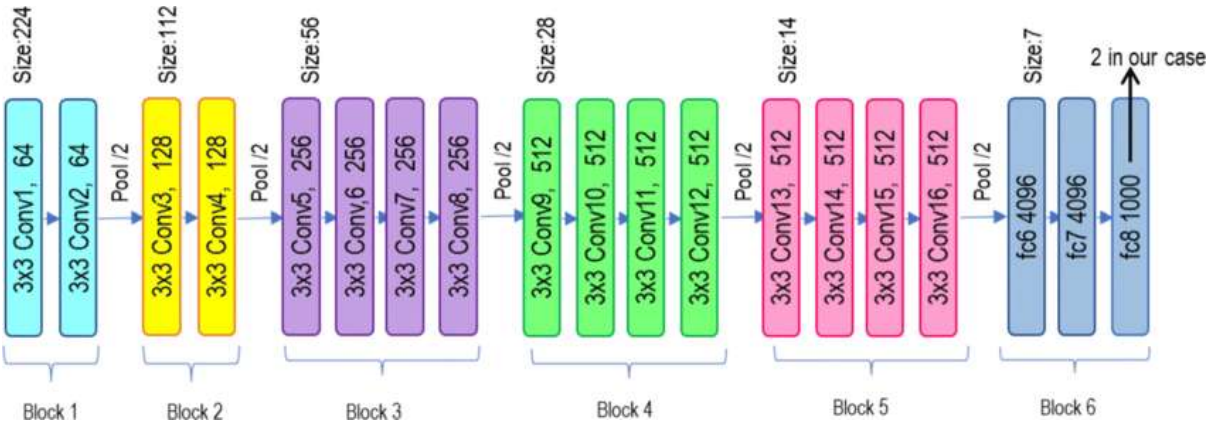
We used ImageDataGenerator for preprocessing.

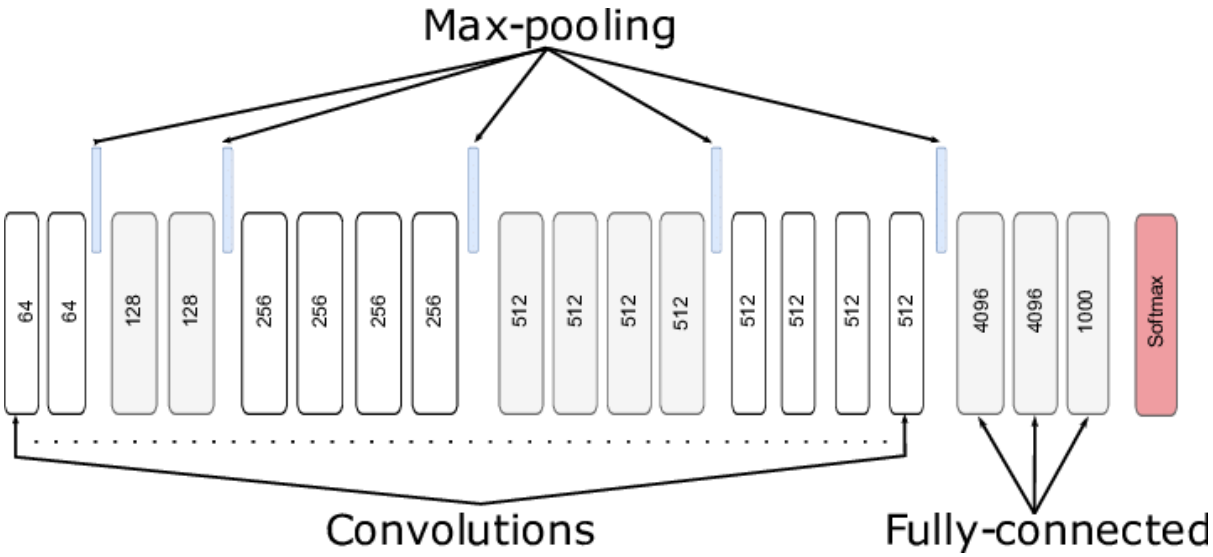
🧠 Milestone 3: Model Building

We used **Transfer Learning with VGG19**.

Steps:

1. Import VGG19 model
2. Freeze base layers
3. Add Dense layers
4. Use Softmax activation
5. Compile with Adam optimizer
6. Train model
7. Save model as dogbreed.h5

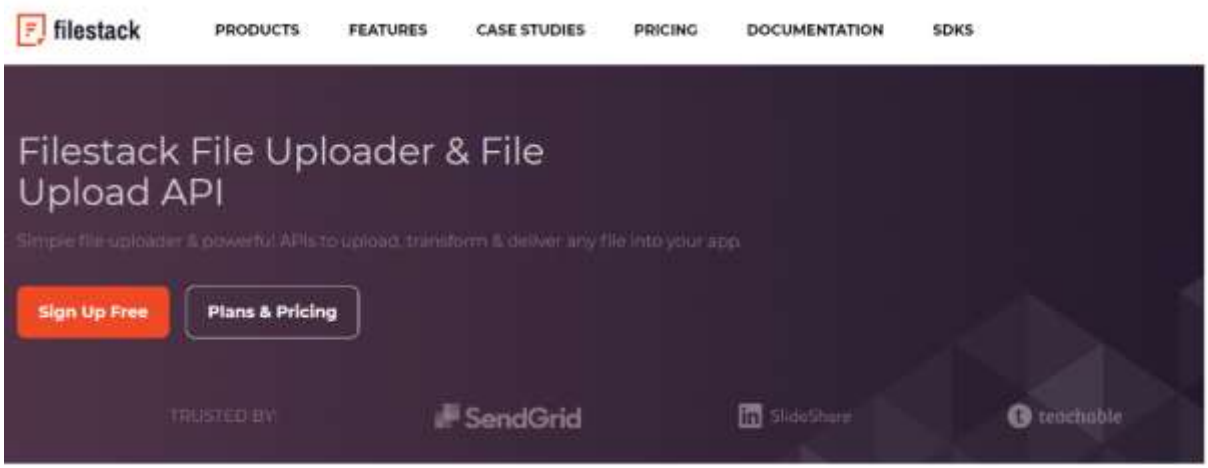


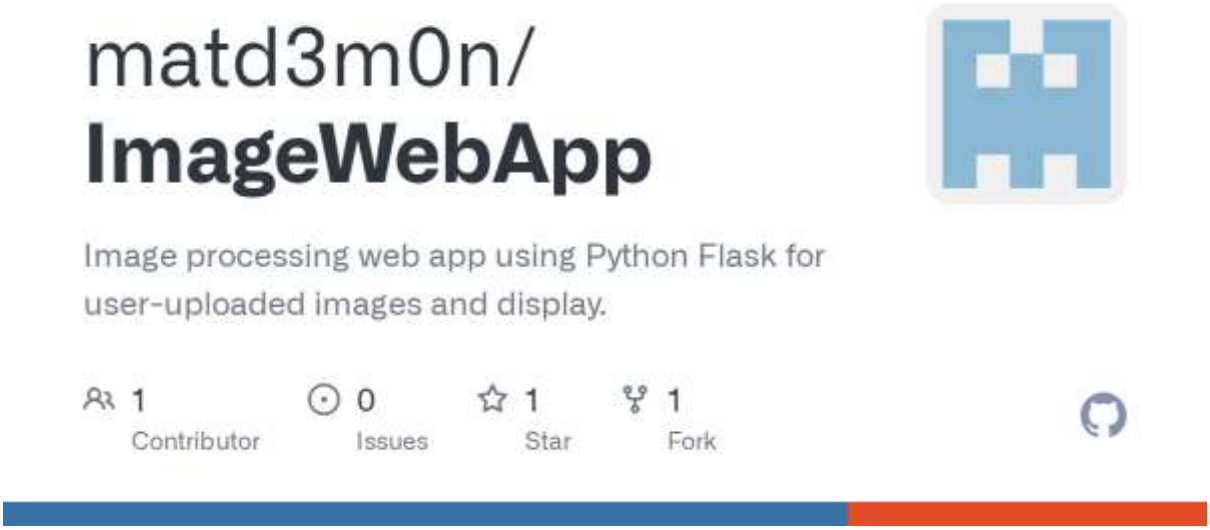


🌐 Milestone 4: Flask Application

We built a Flask web application that:

- Loads trained model
- Accepts uploaded image
- Predicts dog breed
- Displays result





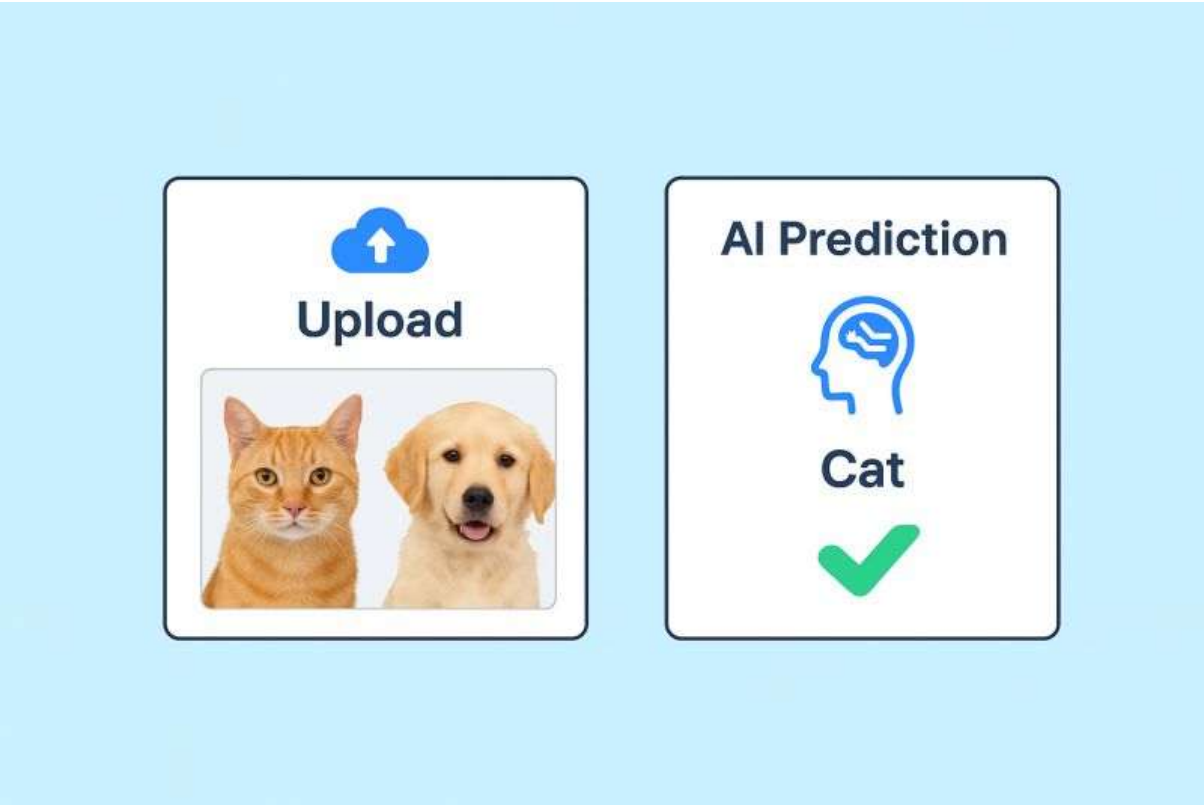
Running the Application

1. Open Anaconda Prompt
2. Navigate to project folder
3. Run:

python app.py

4. Open browser:

http://127.0.0.1:5000/



📊 **Result**

The system successfully predicts the dog breed from uploaded images with good accuracy.