**CAT AND DOG CLASSIFIER: PORJECT DOCUMENTATION**

**1. Project Content**

This project demonstrates the use of Convolutional Neural Networks (CNNs) to solve a binary image classification problem—identifying whether a given image contains a cat or a dog. The classifier is trained using thousands of labeled images and deployed via a simple web interface built with Gradio.

**2. Project Code**

pip install tensorflow pillow numpy

import tensorflow as tf

from tensorflow.keras.applications.mobilenet\_v2 import MobileNetV2, preprocess\_input, decode\_predictions

from tensorflow.keras.preprocessing import image

import numpy as np

import os

# Load pre-trained model

model = MobileNetV2(weights='imagenet')

def predict\_cat\_or\_dog(img\_path):

# Load and preprocess the image

img = image.load\_img(img\_path, target\_size=(224, 224))

img\_array = image.img\_to\_array(img)

img\_array = np.expand\_dims(img\_array, axis=0)

img\_array = preprocess\_input(img\_array)

# Predict using the model

predictions = model.predict(img\_array)

decoded\_preds = decode\_predictions(predictions, top=3)[0]

# Check if 'cat' or 'dog' is in predictions

for \_, label, confidence in decoded\_preds:

if 'cat' in label.lower():

return "Cat"

elif 'dog' in label.lower():

return "Dog"

return "Not sure – try another image"

# Example usage

if \_\_name\_\_ == "\_\_main\_\_":

img\_path = input("Enter the path of a cat or dog image: ")

if os.path.exists(img\_path):

result = predict\_cat\_or\_dog(img\_path)

print("Prediction:", result)

else:

print("Invalid image path.")

**3. Key Technologies**

* **Python**: Programming language used for the entire pipeline.
* **TensorFlow/Keras**: Used to create and train deep learning models.
* **Gradio**: Creates a simple, interactive front-end for the model.
* **NumPy and OpenCV**: Assist with data preprocessing and image manipulation.
* **Matplotlib**: For visualization of training metrics.

**4. Description**

Convolutional Neural Networks (CNNs) are a type of deep neural network designed to process pixel data. They are widely used in computer vision tasks such as image classification, object detection, and segmentation.

**Key Concepts:**

* **Convolutional Layers**: Apply filters to detect patterns such as edges, textures, and shapes.
* **Pooling Layers**: Reduce spatial dimensions to make the network faster and reduce overfitting.
* **Dropout**: A regularization technique to prevent overfitting by randomly turning off neurons during training.
* **Flatten and Dense Layers**: Convert feature maps into a 1D array and pass through fully connected layers to make final predictions.

**Model Architecture:**

* **Input Layer**: Accepts images resized to 128x128 with 3 color channels (RGB).
* **Hidden Layers**:
  + Conv2D + MaxPooling2D (multiple layers for hierarchical feature extraction)
  + Dropout (for regularization)
  + Flatten
  + Dense (ReLU activation)
* **Output Layer**: Dense layer with sigmoid activation for binary classification (Cat = 0, Dog = 1)

**5. Output**

* **Input**: User uploads a cat or dog image.
* **Output**: Model returns a label—"Cat" or "Dog"—along with a confidence score.
* **Evaluation Metrics**:
  + Accuracy
  + Precision, Recall, F1-score (optional)
  + Loss and accuracy curves for training and validation sets

EXAMPLE:

Enter the path of a cat or dog image: images/cat.jpg

Prediction: Cat

**6. Further Research**

* **Transfer Learning**: Use pre-trained models like VGG16, ResNet, or Inception to improve accuracy.
* **Object Detection**: Detect and label multiple animals within one image.
* **Multi-class Classification**: Expand dataset to classify other animals like birds, rabbits, etc.
* **Explainable AI**: Use Grad-CAM or similar techniques to highlight regions that influenced the prediction.

**7. Conclusion**

This project illustrates the practical application of deep learning in image classification. With minimal user input (an image), the model can accurately predict the category of the object. Its applications span pet adoption services, veterinary diagnostics, security surveillance, and educational platforms.

Future improvements can include larger and more diverse datasets, real-time camera integration, and edge deployment on mobile devices.