

EXPERIMENT : 1

TITLE : Write a program to demonstrate the working of CNN architecture to classify images

DATE : 20/01/25

AIM:

To demonstrate the working of CNN architecture using MobileNetV2 for image classification.

PROGRAM:

```
import tensorflow as tf
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input, decode_predictions
from tensorflow.keras.preprocessing.image import load_img, img_to_array
import numpy as np
import matplotlib.pyplot as plt
import os

def preprocess_image(image_path, target_size=(224, 224)):
    img = load_img(image_path, target_size=target_size)
    img_array = img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)
    img_array = preprocess_input(img_array)
    return img, img_array

model = MobileNetV2(weights="imagenet")

image_files = [
    "C:/Users/AI_LAB/Downloads/pokemon dataset/images/whimsicott.png",
    "C:/Users/AI_LAB/Downloads/pokemon dataset/images/whirlipede.png",
    "C:/Users/AI_LAB/Downloads/pokemon dataset/images/whiscash.png",
    "C:/Users/AI_LAB/Downloads/pokemon dataset/images/whismur.png",
    "C:/Users/AI_LAB/Downloads/pokemon dataset/images/wigglytuff.png",
    "C:/Users/AI_LAB/Downloads/pokemon dataset/images/wimpod.png"
]

for image_file in image_files:
    try:
        img, processed_image = preprocess_image(image_file)
        predictions = model.predict(processed_image)
        decoded_predictions = decode_predictions(predictions, top=3)[0]
        sorted_predictions = sorted(decoded_predictions, key=lambda x: x[2])
        plt.imshow(img)
        plt.axis('off')
        plt.title("Predictions:")
        for i, (imagenet_id, label, score) in enumerate(sorted_predictions):
            accuracy_percentage = int(score * 100)
            plt.text(0, 230 + i * 20, f"Accuracy: {accuracy_percentage}%", fontsize=12, color='blue')
        plt.show()
    except Exception as e:
        print(f"Error processing {image_file}: {e}")
```

OUTPUT:

Predictions:



Accuracy: 2%

Accuracy: 3%

Accuracy: 4%

RESULT:

The CNN model was successfully implemented to classify input images using MobileNetV2 architecture.