TITLE :- DOG VS CAT IMAGE CLASSIFIER USING VGG16

INTRODUCTION:-

The *Dog vs Cat Image Classifier* is a deep learning project that uses a pre-trained convolutional neural network (CNN) model—**VGG16**—to distinguish between images of dogs and cats. VGG16, developed by the Visual Geometry Group at Oxford, is known for its deep architecture and excellent performance on image recognition tasks.

TOOLS AND LIBRARIES:-

1. Python
2. TensorFlow / Keras
3. Jupyter Notebook / Colab
4. NumPy, Matplotlib, etc.

ALGORITHM:-

- **Step 1: Load and Preprocess Dataset**
  Collect dog and cat images.
  Resize all images to 224x224 pixels.
  Normalize pixel values to [0, 1].
  Split dataset into training and validation sets.

- **Step 2: Load Pre-trained VGG16 Model**
  Import VGG16 without the top classification layers.
  Freeze all pre-trained layers to retain learned features.
  Step 3: Add Custom Classification Layers
  Add Flatten, Dense, and Dropout layers.

  Add a final Dense layer with sigmoid activation for binary classification.

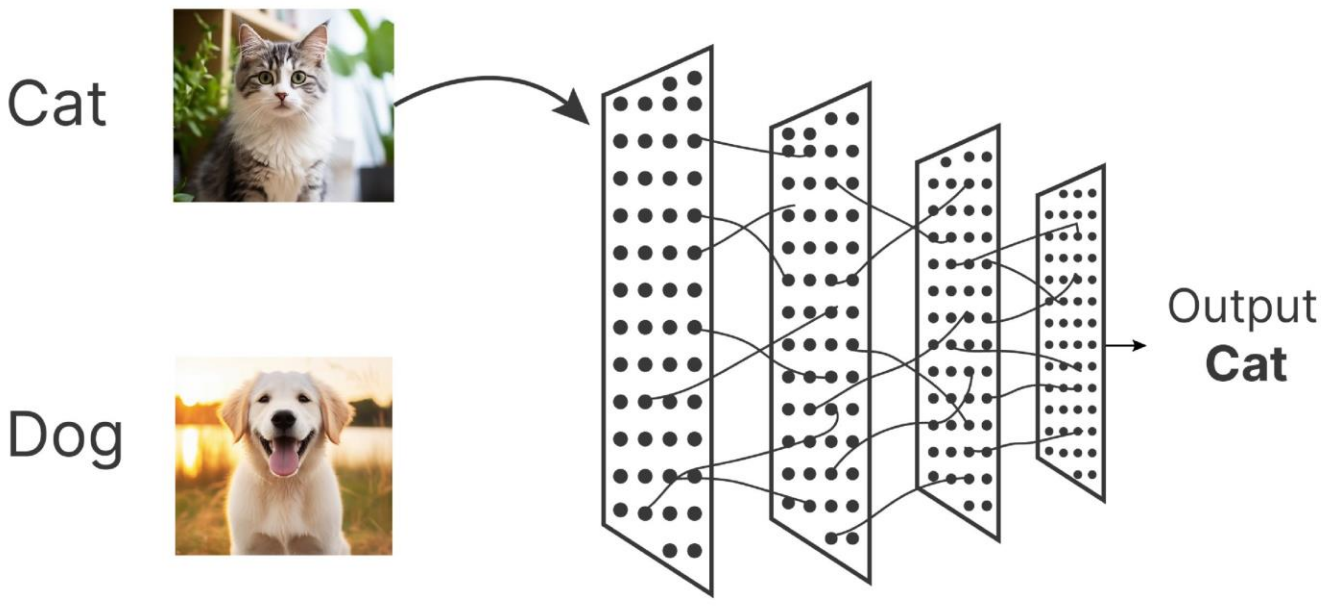
- **Step 4: Compile and Train the Model**
  Use binary cross-entropy loss and an optimizer like Adam.
  Train the model using training data and validate with validation data.
- **Step 5: Evaluate and Predict**
  Evaluate model accuracy on validation data.
  Use the model to predict whether new images are dogs or cats.

SOURCE CODE:-

```
#This will open a file picker to upload image
from google.colab import files
uploaded = files.upload()

#This step imports all the necessary libraries for image processing, deep learning, and plotting.
import numpy as np
import matplotlib.pyplot as plt
import os
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input, decode_predictions

# Load the VGG16 model pre-trained on ImageNet.

model = VGG16(weights='imagenet')
# Change this to the uploaded filename.
 img_path = 'sloth.jpg'
# Example: change to your uploaded filename.
# Check if file exists
if os.path.exists(img_path):
    print("✅ Image found. Classifying...")
 # Load and resize image to 224x224 for VGG16.
img = image.load_img(img_path, target_size=(224, 224))

# Show the image.
_plt.imshow(img)
    plt.axis('off')
    plt.title("Input Image")
    plt.show()

# Convert to array and preprocess.
_x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = preprocess_input(x)
```

```
else:
    print("❌ Image file not found.")

# Predict the class
preds = model.predict(x)

# Decode predictions (top 3 results)
decoded = decode_predictions(preds, top=3)[0]

# Show results
print("\nTop Predictions:")
for i, (imagenet_id, label, prob) in enumerate(decoded):
    print(f"{i+1}. {label}: {prob*100:.2f}%")

# Pick the most confident one
animal_name = decoded[0][1].replace("_", " ")  # Make it more readable

# Show result
print(f"\n✅ It's a {animal_name}!")
```

WHAT IS IT CAN CNN:-



CONCLUSION:-

The Dog vs Cat image classifier using the VGG16 model demonstrates the effectiveness of transfer learning in solving image classification problems with high accuracy and minimal training time. By leveraging a pre-trained convolutional neural network like VGG16, we were able to extract powerful image features and build a reliable model for binary classification. The project not only highlights the importance of deep learning in computer vision but also shows how existing architectures can be customized for specific tasks. With further tuning and more data, the model can be enhanced even further, making it suitable for real-world applications such as pet identification systems or automated animal monitoring tools.

OUTPUT:-

INPUT IMAGE:-

```
Image found. Classifying...
```

OUTPUT:-

```python
# Predict the class
preds = model.predict(x)

# Decode predictions (top 3 results)
decoded = decode_predictions(preds, top=3)[0]

# Show results
print("\nTop Predictions:")
for i, (imagenet_id, label, prob) in enumerate(decoded):
    print(f"{i+1}. {label}: {prob*100:.2f}%")

# Pick the most confident one
animal_name = decoded[0][1].replace("_", " ")  # Make it more readable

# Show result
print(f"\n✅ It's a {animal_name}!")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 871ms/step

Top Predictions:
1. lion: 99.97%
2. Arabian_camel: 0.03%
3. hyena: 0.00%

✅ It's a lion!
```