```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Importing Required Libraries

```
# General Libraries
import os
from os import listdir
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import cv2

# TensorFlow and Keras for Deep Learning
import tensorflow as tf
from tensorflow.keras.models import Sequential, Model
from tensorflow.keras.layers import (
    Dense, Dropout, Flatten, Conv2D, MaxPooling2D,
    BatchNormalization, Input, Activation
)
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.optimizers import RMSprop, Adam, SGD
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras import backend, losses, optimizers

# Google Colab Utilities
from google.colab.patches import cv2_imshow

# Error Metrics
from sklearn.metrics import (
    mean_absolute_error, mean_squared_error, accuracy_score, confusion_matrix
)

# Transfer Learning Models
from tensorflow.keras.applications import VGG16

# Visualization Tools
import seaborn as sns

```

Unzipping the Dataset and Verifying Paths

```
import os

# Define paths
source_zip_path = "/content/drive/MyDrive/Colab Notebooks/Caltech_101_Reduced.zip"
destination_path = "/content/Caltech_101_Reduced"

# Check if the zip file exists
if os.path.exists(source_zip_path):
    # Unzipping the dataset
    !unzip -q "{source_zip_path}" -d "{destination_path}"
    print(f"Dataset unzipped successfully at {destination_path}")
else:
    print(f"Zip file not found at: {source_zip_path}")

```

```
replace /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0001.jpg? [y]es, [n]o, [A]ll, [N]one, [r]ename: N
Dataset unzipped successfully at /content/Caltech_101_Reduced
```

Exploring Dataset Directory Structure and Files

```
for root, dirs, files in os.walk(destination_path):
    print(f"Directory: {root}")
    for file in files:
        # Filter to show only specific file types (optional)
        if file.endswith(('.jpg', '.png', '.txt', '.mat')):
            print(f"  File: {file}")
```

```
  File: image_0024.jpg
  File: image_0002.jpg
  File: image_0085.jpg
  File: image_0012.jpg
  File: image_0074.jpg
  File: image_0052.jpg
  File: image_0067.jpg
  File: image_0068.jpg
  File: image_0053.jpg
  File: image_0073.jpg
  File: image_0081.jpg
  File: image_0091.jpg
  File: image_0054.jpg
  File: image_0056.jpg
  File: image_0060.jpg
  File: image_0001.jpg
  File: image_0059.jpg
  File: image_0018.jpg
  File: image_0089.jpg
  File: image_0076.jpg
Directory: /content/Caltech_101_Reduced/caltech101_classification/dalmatian
  File: image_0005.jpg
  File: image_0027.jpg
  File: image_0063.jpg
  File: image_0064.jpg
  File: image_0043.jpg
  File: image_0049.jpg
  File: image_0047.jpg
  File: image_0033.jpg
  File: image_0051.jpg
  File: image_0057.jpg
  File: image_0039.jpg
  File: image_0015.jpg
  File: image_0016.jpg
  File: image_0008.jpg
  File: image_0014.jpg
  File: image_0048.jpg
  File: image_0030.jpg
  File: image_0021.jpg
  File: image_0010.jpg
  File: image_0026.jpg
  File: image_0031.jpg
  File: image_0020.jpg
  File: image_0058.jpg
  File: image_0036.jpg
  File: image_0032.jpg
  File: image_0041.jpg
  File: image_0007.jpg
  File: image_0017.jpg
  File: image_0025.jpg
  File: image_0009.jpg
  File: image_0004.jpg
  File: image_0062.jpg
  File: image_0055.jpg
  File: image_0037.jpg
  File: image_0040.jpg
  File: image_0066.jpg
  File: image_0003.jpg
  File: image_0038.jpg
```

Consolidating Annotations into a DataFrame and Saving as CSV

```
import os
import pandas as pd

# Directory paths
annotations_dir = "/content/Caltech_101_Reduced/caltech101_classification"
image_dir = "/content/Caltech_101_Reduced/caltech101_classification"

# Classes and their corresponding annotation files
classes_files = {
    'Butterfly': os.path.join(annotations_dir, 'res_butterfly.txt'),
    'Dalmatian': os.path.join(annotations_dir, 'res_dalmatian.txt'),
    'Dolphin': os.path.join(annotations_dir, 'res_dolphin.txt')
}

# Consolidate annotations into a single DataFrame
annotations = []

```

```
for class_name, file_path in classes_files.items():
    if not os.path.exists(file_path):
        print(f"Annotation file not found: {file_path}")
        continue

    with open(file_path, 'r') as f:
        for line in f:
            # Split by commas (adjusted for the file format)
            parts = line.strip().split(',')
            if len(parts) != 5:  # Expecting 5 values: filename, x_min, y_min, x_max, y_max
                print(f"Skipping malformed line in {file_path}: {line.strip()}")
                continue

            filename, x_min, y_min, x_max, y_max = parts
            # Adjust file extension if needed (e.g., replace '.mat' with '.jpg')
            filename = filename.replace('.mat', '.jpg')
            annotations.append({
                'filename': filename,
                'x_min': float(x_min),
                'y_min': float(y_min),
                'x_max': float(x_max),
                'y_max': float(y_max),
                'class': class_name
            })

# Convert to DataFrame
annotations_df = pd.DataFrame(annotations)

# Display the first few rows of the consolidated DataFrame
print("Consolidated annotations:")
print(annotations_df.head())

# Save the consolidated DataFrame to a CSV file (optional)
output_csv_path = "/content/Caltech_101_Reduced/consolidated_annotations.csv"
annotations_df.to_csv(output_csv_path, index=False)
print(f"Consolidated annotations saved to: {output_csv_path}")

```

```
Consolidated annotations:
                filename  x_min  y_min  x_max  y_max      class
0  annotation_0001.jpg    3.0   11.0  284.0  188.0  Butterfly
1   annotation_0002.jpg    2.0   65.0  215.0  230.0  Butterfly
2   annotation_0003.jpg   39.0    4.0  272.0  192.0  Butterfly
3   annotation_0004.jpg    9.0   10.0  289.0  264.0  Butterfly
4   annotation_0005.jpg   55.0   45.0  299.0  204.0  Butterfly
Consolidated annotations saved to: /content/Caltech_101_Reduced/consolidated_annotations.csv
```

```
# Print the content of the main directory
print("Image directory content:")
if os.path.exists(image_dir):
    print(os.listdir(image_dir))
else:
    print(f"Image directory not found: {image_dir}")

# List files in each class subdirectory
for class_name in ['butterfly', 'dalmatian', 'dolphin']:
    class_dir = os.path.join(image_dir, class_name)
    if os.path.exists(class_dir):
        print(f"\nFiles in {class_name} ({class_dir}):")
        print(os.listdir(class_dir))
    else:
        print(f"\nClass directory not found: {class_dir}")

```

```
Image directory content:
['res_dolphin.txt', 'dolphin', 'butterfly', 'res_butterfly.txt', 'res_dalmatian.txt', 'dalmatian']

Files in butterfly (/content/Caltech_101_Reduced/caltech101_classification/butterfly):
['image_0005.jpg', 'image_0027.jpg', 'image_0063.jpg', 'image_0064.jpg', 'image_0043.jpg', 'image_0049.jpg', 'image_0047.jpg', 'image_00

Files in dalmatian (/content/Caltech_101_Reduced/caltech101_classification/dalmatian):
['image_0005.jpg', 'image_0027.jpg', 'image_0063.jpg', 'image_0064.jpg', 'image_0043.jpg', 'image_0049.jpg', 'image_0047.jpg', 'image_00

Files in dolphin (/content/Caltech_101_Reduced/caltech101_classification/dolphin):
['image_0005.jpg', 'image_0027.jpg', 'image_0063.jpg', 'image_0064.jpg', 'image_0043.jpg', 'image_0049.jpg', 'image_0047.jpg', 'image_00
```

Correcting Filenames in Annotations to Match Dataset Files

```
def correct_filenames(annotations, image_dir):
    corrected_annotations = []

    for class_name in ['butterfly', 'dalmatian', 'dolphin']:
        # List actual files in the class subdirectory
        class_dir = os.path.join(image_dir, class_name)
        if not os.path.exists(class_dir):
            print(f"Class directory not found: {class_dir}")
            continue

        actual_files = os.listdir(class_dir)

        # Filter annotations for this class
        class_annotations = annotations[annotations['class'].str.lower() == class_name]

        for _, row in class_annotations.iterrows():
            # Find the actual filename matching the prefix
            original_prefix = row['filename'].split('.')[0]
            matched_file = next(
                (file for file in actual_files if file.startswith(original_prefix.replace('annotation', 'image'))),
                None
            )
            if matched_file:
                row['filename'] = matched_file
                corrected_annotations.append(row)
            else:
                print(f"No matching file found for {row['filename']} in class {class_name}")

    return pd.DataFrame(corrected_annotations)

# Correct filenames in annotations
annotations_df = correct_filenames(annotations_df, image_dir)

# Display the corrected annotations
print("Corrected annotations:")
print(annotations_df.head())

```

```
No matching file found for annotation_0001.jpg in class butterfly
No matching file found for annotation_0001.jpg in class dalmatian
No matching file found for annotation_0001.jpg in class dolphin
Corrected annotations:
          filename  x_min  y_min  x_max  y_max      class
1  image_0002.jpg    2.0   65.0  215.0  230.0  Butterfly
2  image_0003.jpg   39.0    4.0  272.0  192.0  Butterfly
3  image_0004.jpg    9.0   10.0  289.0  264.0  Butterfly
4  image_0005.jpg   55.0   45.0  299.0  204.0  Butterfly
5  image_0006.jpg   20.0   50.0  299.0  265.0  Butterfly
```

Scaling Bounding Box Coordinates Relative to Image Dimensions

```
def scale_bounding_boxes(annotations, img_dir):
    scaled_annotations = []

    for _, row in annotations.iterrows():
        # Include the class-specific subdirectory in the path
        class_subdir = row['class'].lower()  # Match subdirectory name
        img_path = os.path.join(img_dir, class_subdir, row['filename'])

        # Check if the image exists
        if not os.path.exists(img_path):
            print(f"[Warning] Image not found: {img_path}")
            continue

        # Read the image
        img = cv2.imread(img_path)
        if img is None:
            print(f"[Error] Failed to read image: {img_path}")
            continue

        # Get image dimensions
        height, width, _ = img.shape

```

```
        # Scale bounding box coordinates
        x_min_scaled = row['x_min'] / width
        y_min_scaled = row['y_min'] / height
        x_max_scaled = row['x_max'] / width
        y_max_scaled = row['y_max'] / height

        # Append scaled annotation
        scaled_annotations.append({
            'filename': row['filename'],
            'class': row['class'],
            'x_min': x_min_scaled,
            'y_min': y_min_scaled,
            'x_max': x_max_scaled,
            'y_max': y_max_scaled
        })

    # Convert to DataFrame
    return pd.DataFrame(scaled_annotations)

# Example Usage:
scaled_annotations_df = scale_bounding_boxes(annotations_df, image_dir)
print("Scaled annotations:")
print(scaled_annotations_df.head())

```

```
Scaled annotations:
       filename      class     x_min     y_min     x_max     y_max
0  image_0002.jpg  Butterfly  0.009050  0.216667  0.972851  0.766667
1  image_0003.jpg  Butterfly  0.130000  0.020833  0.906667  1.000000
2  image_0004.jpg  Butterfly  0.030000  0.036900  0.963333  0.974170
3  image_0005.jpg  Butterfly  0.183333  0.161871  0.996667  0.733813
4  image_0006.jpg  Butterfly  0.066667  0.184502  0.996667  0.977860
```

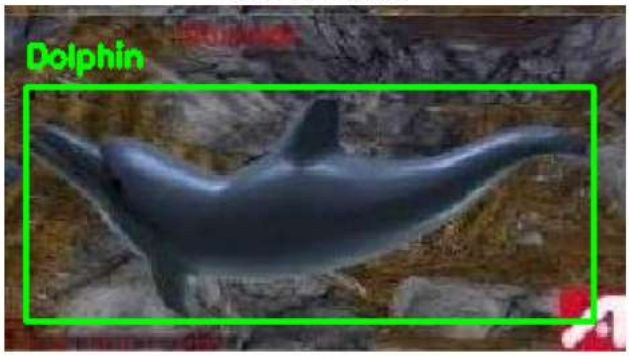
Visualizing Images with Predicted Bounding Boxes and Class Labels

```
def visualize_images_with_bboxes(annotations, img_dir, num_images=5):
    """
    Visualizes images with bounding boxes.
    Args:
        annotations (pd.DataFrame): DataFrame containing annotations with scaled bounding box coordinates.
        img_dir (str): Root directory containing class subdirectories with images.
        num_images (int): Number of images to visualize.
    """
    # Sample a subset of annotations
    sampled_annotations = annotations.sample(n=min(num_images, len(annotations)))

    for _, row in sampled_annotations.iterrows():
        # Construct the full image path
        class_subdir = row['class'].lower()
        img_path = os.path.join(img_dir, class_subdir, row['filename'])

        # Read the image
        img = cv2.imread(img_path)
        if img is None:
            print(f"[Error] Failed to read image: {img_path}")
            continue

        # Get image dimensions
        height, width, _ = img.shape

        # Convert scaled bounding box to absolute pixel coordinates
        x_min = int(row['x_min'] * width)
        y_min = int(row['y_min'] * height)
        x_max = int(row['x_max'] * width)
        y_max = int(row['y_max'] * height)

        # Draw bounding box on the image
        cv2.rectangle(img, (x_min, y_min), (x_max, y_max), (0, 255, 0), 2)
        cv2.putText(
            img, row['class'], (x_min, max(y_min - 10, 0)),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2
        )

        # Convert BGR to RGB for Matplotlib
        img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

```

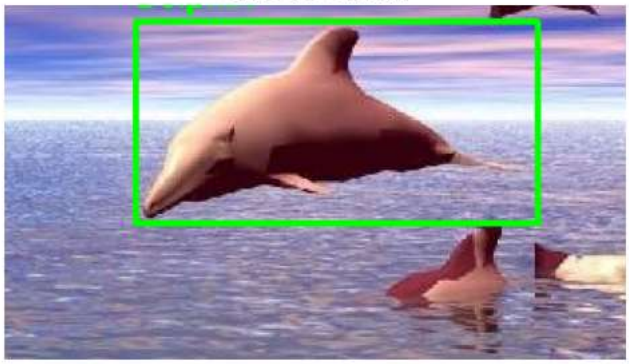
```
        # Display the image
        plt.figure(figsize=(6, 6))
        plt.imshow(img_rgb)
        plt.axis('off')
        plt.title(f"Class: {row['class']}")
        plt.show()

# Visualize a subset of images with bounding boxes
visualize_images_with_bboxes(scaled_annotations_df, image_dir, num_images=5)

```

Class: Dalmatian

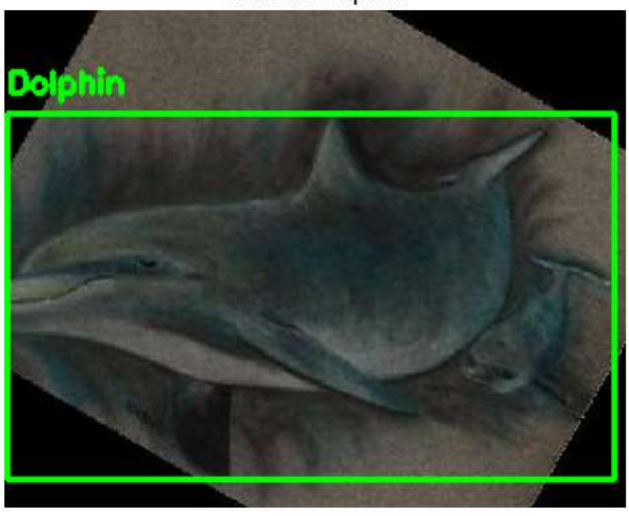
Class: Dolphin



Class: Dolphin



Class: Dolphin



Class: Dolphin

Prepare images and bounding box data for training

```
# Initialize lists to store processed data
images = []
bboxes = []
labels = []

# Define image dimensions for resizing
img_width, img_height = 224, 224

# Ensure annotations_df is a DataFrame # Changed 'annotations' to 'annotations_df'
if not isinstance(annotations_df, pd.DataFrame): # Changed 'annotations' to 'annotations_df'
    annotations_df = pd.DataFrame(annotations_df) # Changed 'annotations' to 'annotations_df'

# Print unique classes in the annotations_df # Changed 'annotations' to 'annotations_df'
print("Classes before encoding:", annotations_df['class'].unique()) # Changed 'annotations' to 'annotations_df'

# Initialize LabelBinarizer for one-hot encoding
lb = LabelBinarizer()
classes = annotations_df['class'].unique() # Changed 'annotations' to 'annotations_df'
lb.fit(classes)

# Print classes after encoding
print("Classes after encoding:", lb.classes_)

# Debugging variables
missing_files = 0
processed_files = 0

# Iterate through the annotations_df # Changed 'annotations' to 'annotations_df'
for _, row in annotations_df.iterrows(): # Changed 'annotations' to 'annotations_df'
    class_subdir = row['class'].lower()  # Match subdirectory name
    img_filename = row['filename'].replace('annotation', 'image')  # Adjust filename prefix if needed
    img_path = os.path.join(image_dir, class_subdir, img_filename)

    # Debugging: Print paths being checked
    print(f"Checking image path: {img_path}")

    if not os.path.exists(img_path):
        print(f"[Warning] File does not exist: {img_path}")
        missing_files += 1
        continue

    # Load the original image
    img = cv2.imread(img_path)
    if img is not None:
        original_height, original_width, _ = img.shape  # Get original dimensions

        # Resize and normalize the image
        img_resized = cv2.resize(img, (img_width, img_height))
        img_normalized = img_resized / 255.0
        images.append(img_normalized)

        # Normalize bounding box coordinates using original dimensions
        x_min_scaled = row['x_min'] / original_width
        y_min_scaled = row['y_min'] / original_height
        x_max_scaled = row['x_max'] / original_width
        y_max_scaled = row['y_max'] / original_height
        bboxes.append([x_min_scaled, y_min_scaled, x_max_scaled, y_max_scaled])

        # Append the one-hot encoded label
        encoded_label = lb.transform([row['class']])[0]
        labels.append(encoded_label)
```

```
        processed_files += 1
    else:
        print(f"[Error] Failed to read image: {img_path}")
        missing_files += 1

# Convert data to NumPy arrays
images = np.array(images, dtype=np.float32)
bboxes = np.array(bboxes, dtype=np.float32)
labels = np.array(labels, dtype=np.float32)

# Debugging: Print summary
print(f"Number of processed files: {processed_files}")
print(f"Number of missing files: {missing_files}")
print("Images shape:", images.shape)
print("Bounding boxes shape (normalized):", bboxes.shape)
print("Labels shape:", labels.shape)
```

```
Classes before encoding: ['Butterfly' 'Dalmatian' 'Dolphin']
Classes after encoding: ['Butterfly' 'Dalmatian' 'Dolphin']
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0002.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0003.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0004.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0005.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0006.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0007.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0008.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0009.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0010.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0011.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0012.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0013.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0014.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0015.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0016.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0017.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0018.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0019.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0020.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0021.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0022.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0023.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0024.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0025.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0026.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0027.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0028.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0029.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0030.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0031.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0032.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0033.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0034.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0035.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0036.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0037.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0038.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0039.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0040.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0041.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0042.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0043.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0044.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0045.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0046.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0047.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0048.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0049.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0050.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0051.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0052.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0053.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0054.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0055.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0056.jpg
Checking image path: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0057.jpg
```

```
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
import os

# Ensure annotations is a DataFrame (if not already)
```

```
if isinstance(annotations, list):
    annotations = pd.DataFrame(annotations)

# Prepare data from annotations
images = []  # List of processed images
bboxes = []  # List of bounding box coordinates
labels = []  # List of class labels

# Iterate over annotations to populate images, bboxes, and labels
for _, row in annotations.iterrows():
    class_subdir = row['class'].lower()  # Match subdirectory name
    img_filename = row['filename'].replace('annotation', 'image')  # Adjust filename prefix
    img_path = os.path.join(image_dir, class_subdir, img_filename)

    # Check if the image exists
    if not os.path.exists(img_path):
        print(f"[Warning] Image not found: {img_path}")
        continue

    # Read and preprocess the image
    img = cv2.imread(img_path)
    if img is None:
        print(f"[Error] Failed to load image: {img_path}")
        continue

    # Resize and normalize the image
    img = cv2.resize(img, (224, 224))
    img = img / 255.0
    images.append(img)

    # Append bounding boxes and labels
    bboxes.append([row['x_min'], row['y_min'], row['x_max'], row['y_max']])
    labels.append(row['class'])

# Convert lists to numpy arrays for `train_test_split`
images = np.array(images, dtype=np.float32)
bboxes = np.array(bboxes, dtype=np.float32)
labels = np.array(labels)  # Use a LabelBinarizer for one-hot encoding if needed

# Perform train-test split
(trainImages, testImages, trainBBoxes, testBBoxes, trainLabels, testLabels) = train_test_split(
    images, bboxes, labels, test_size=0.2, random_state=42
)

# Debugging: Print the shapes of the splits
print("Train Images Shape:", trainImages.shape)
print("Test Images Shape:", testImages.shape)
print("Train Bounding Boxes Shape:", trainBBoxes.shape)
print("Test Bounding Boxes Shape:", testBBoxes.shape)
print("Train Labels Shape:", trainLabels.shape)
print("Test Labels Shape:", testLabels.shape)

# Construct image paths for all annotations
imagePaths = [
    os.path.join(image_dir, row['class'].lower(), row['filename'].replace('annotation', 'image'))
    for _, row in annotations.iterrows()
]

# Prepare the test image paths
test_imagePaths = [
    imagePaths[i]
    for i in range(len(imagePaths))
    if i < len(testImages)
]

# Debugging: Print the length of test image paths
print("Total Test Image Paths:", len(test_imagePaths))
print("Example Test Image Paths:", test_imagePaths[:5])  # Show the first 5 test image paths

```

```
[Warning] Image not found: /content/Caltech_101_Reduced/caltech101_classification/butterfly/image_0001.jpg
[Warning] Image not found: /content/Caltech_101_Reduced/caltech101_classification/dalmatian/image_0001.jpg
[Warning] Image not found: /content/Caltech_101_Reduced/caltech101_classification/dolphin/image_0001.jpg
Train Images Shape: (153, 224, 224, 3)
Test Images Shape: (39, 224, 224, 3)
Train Bounding Boxes Shape: (153, 4)
Test Bounding Boxes Shape: (39, 4)
Train Labels Shape: (153,)
```

```
Test Labels Shape: (39,)
Total Test Image Paths: 39
Example Test Image Paths: ['/content/Caltech_101_Reduced/caltech101_classification/butterfly/\ufeffimage_0001.jpg', '/content/Caltech_10
```

```
# Save the test image paths to a text file
output_file = "testing_multiclass.txt"

try:
    with open(output_file, "w") as f:
        f.write("\n".join(test_imagePaths))
    print(f"Test image paths successfully saved to '{output_file}'")
except Exception as e:
    print(f"[Error] Failed to save test image paths: {e}")

```

```
[Error] Failed to save test image paths: [Errno 2] No such file or directory: 'testing_multiclass.txt'
```

Import VGG16 and Build Base Model

```
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Flatten, Input

# Load the VGG16 base model
vgg = VGG16(weights="imagenet", include_top=False, input_tensor=Input(shape=(224, 224, 3)))

# Flatten the output of the VGG16 base model
flatten = vgg.output
flatten = Flatten()(flatten)

```

Import Libraries and VGG16

```
from tensorflow.keras.applications import VGG16
from tensorflow.keras.layers import Flatten, Input

# Load the VGG16 base model
vgg = VGG16(weights="imagenet", include_top=False, input_tensor=Input(shape=(224, 224, 3)))

# Flatten the output of the VGG16 base model
flatten = vgg.output
flatten = Flatten()(flatten)


```

Define Bounding Box Head

```
from tensorflow.keras.layers import Dense

# Create the bounding box prediction layers
bboxHead = Dense(256, activation="relu")(flatten)
bboxHead = Dense(128, activation="relu")(bboxHead)
bboxHead = Dense(64, activation="relu")(bboxHead)
bboxHead = Dense(32, activation="relu")(bboxHead)
bboxHead = Dense(16, activation="relu")(bboxHead)
bboxHead = Dense(4, activation="sigmoid", name="bounding_box")(bboxHead)

```

Define Classification Head

```
# Create the classification head
classHead = Dense(256, activation="relu")(flatten)
classHead = Dense(128, activation="relu")(classHead)
classHead = Dense(3, activation="softmax", name="class_label")(classHead)  # Adjust the number of classes
```

Combine and Create the Model

```
from tensorflow.keras.models import Model

# Create the final multi-output model (bounding box + classification)
```

```
# Create the final multi-output model (bounding box + classification)
model = Model(inputs=vgg.input, outputs=[bboxHead, classHead])  # Use classHead if classification is needed

# Summary of the model
model.summary()

```

Model: "functional_5"

Layer (type)	Output Shape	Param #	Connected to
input_layer_12 (InputLayer)	(None, 224, 224, 3)	0	-
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792	input_layer_12[0][0]
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928	block1_conv1[0][0]
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0	block1_conv2[0][0]
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856	block1_pool[0][0]
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584	block2_conv1[0][0]
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0	block2_conv2[0][0]
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168	block2_pool[0][0]
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080	block3_conv1[0][0]
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080	block3_conv2[0][0]
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0	block3_conv3[0][0]
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160	block3_pool[0][0]
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808	block4_conv1[0][0]
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808	block4_conv2[0][0]
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0	block4_conv3[0][0]
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808	block4_pool[0][0]
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808	block5_conv1[0][0]
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808	block5_conv2[0][0]
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0	block5_conv3[0][0]
flatten_12 (Flatten)	(None, 25088)	0	block5_pool[0][0]
dense_39 (Dense)	(None, 256)	6,422,784	flatten_12[0][0]
dense_40 (Dense)	(None, 128)	32,896	dense_39[0][0]
dense_41 (Dense)	(None, 64)	8,256	dense_40[0][0]
dense_42 (Dense)	(None, 32)	2,080	dense_41[0][0]
dense_44 (Dense)	(None, 256)	6,422,784	flatten_12[0][0]
dense_43 (Dense)	(None, 16)	528	dense_42[0][0]
dense_45 (Dense)	(None, 128)	32,896	dense_44[0][0]
bounding_box (Dense)	(None, 4)	68	dense_43[0][0]

Define Metrics and Training/Testing Targets

```
# Define metrics for each output head
METRICS = {
    "class_label": "accuracy",  # Accuracy for class label prediction
    "bounding_box": "accuracy"  # Accuracy metric for bounding box (placeholder)
}

```

```
# Define the training targets for each output head
trainTargets = {
    "class_label": trainLabels,  # Training labels for class label prediction
    "bounding_box": trainBBoxes  # Training bounding boxes
}

# Define the testing targets for each output head
testTargets = {
    "class_label": testLabels,  # Testing labels for class label prediction
    "bounding_box": testBBoxes  # Testing bounding boxes
}

```

Define Loss Functions and Weights

```
from tensorflow.keras.optimizers import Adam

# Specify the initial learning rate
INIT_LR = 1e-4

# Define loss functions for each output head
losses = {
    "class_label": "categorical_crossentropy",  # Loss for class label prediction
    "bounding_box": "mean_squared_error"       # Loss for bounding box regression
}

# Define weights for each loss function
lossWeights = {
    "class_label": 1.0,  # Full weight for class label loss
    "bounding_box": 1.0  # Full weight for bounding box loss
}

```

Compile the Model

```
# Initialize the Adam optimizer
opt = Adam(learning_rate=INIT_LR)

# Compile the model
model.compile(
    loss=losses,        # Loss functions for each output head
    optimizer=opt,      # Adam optimizer with specified learning rate
    metrics=METRICS,    # Metrics to monitor during training
    loss_weights=lossWeights  # Loss weights for multi-task learning
)

# Print the model summary
model.summary()

```

Model: "functional_5"

Layer (type)	Output Shape	Param #	Connected to
input_layer_12 (InputLayer)	(None, 224, 224, 3)	0	-
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792	input_layer_12[0][0]
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928	block1_conv1[0][0]
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0	block1_conv2[0][0]
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856	block1_pool[0][0]
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584	block2_conv1[0][0]
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0	block2_conv2[0][0]
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168	block2_pool[0][0]
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080	block3_conv1[0][0]
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080	block3_conv2[0][0]
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0	block3_conv3[0][0]
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160	block3_pool[0][0]
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808	block4_conv1[0][0]
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808	block4_conv2[0][0]
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0	block4_conv3[0][0]
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808	block4_pool[0][0]
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808	block5_conv1[0][0]
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808	block5_conv2[0][0]
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0	block5_conv3[0][0]
flatten_12 (Flatten)	(None, 25088)	0	block5_pool[0][0]
dense_39 (Dense)	(None, 256)	6,422,784	flatten_12[0][0]
dense_40 (Dense)	(None, 128)	32,896	dense_39[0][0]
dense_41 (Dense)	(None, 64)	8,256	dense_40[0][0]
dense_42 (Dense)	(None, 32)	2,080	dense_41[0][0]
dense_44 (Dense)	(None, 256)	6,422,784	flatten_12[0][0]
dense_43 (Dense)	(None, 16)	528	dense_42[0][0]
dense_45 (Dense)	(None, 128)	32,896	dense_44[0][0]
bounding_box (Dense)	(None, 4)	68	dense_43[0][0]

Define Early Stopping

```
from tensorflow.keras.callbacks import EarlyStopping

# Define early stopping to prevent overfitting
early_stopping = EarlyStopping(
    monitor="val_loss",           # Monitor validation loss
    patience=5,                   # Stop training if no improvement after 5 epochs
    restore_best_weights=True     # Restore the best model weights after training stops
)

```

Train the Model

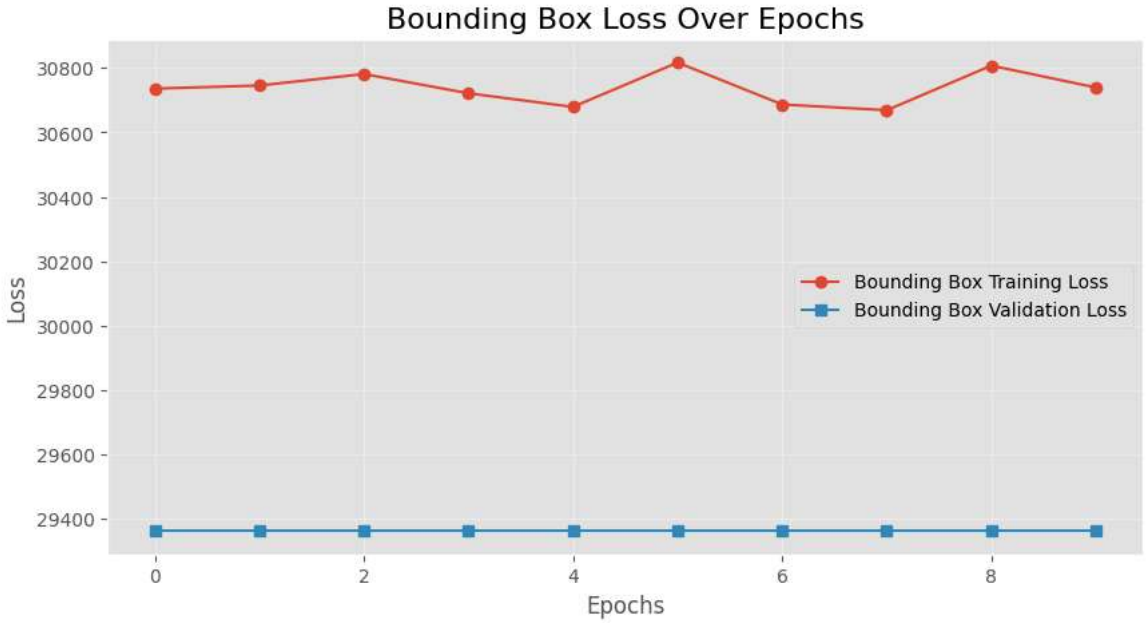
```
from sklearn.preprocessing import LabelBinarizer

# Convert string labels to numeric format
lb = LabelBinarizer()
trainLabels = lb.fit_transform(trainLabels)  # One-hot encode training labels
testLabels = lb.transform(testLabels)        # One-hot encode testing labels

# Debugging: Verify label transformation
print("Transformed trainLabels:", trainLabels[:5])
print("Transformed testLabels:", testLabels[:5])
print("Classes:", lb.classes_)

# Define training targets
trainTargets = {
    "class_label": trainLabels,
    "bounding_box": trainBBoxes
}

testTargets = {
    "class_label": testLabels,
    "bounding_box": testBBoxes
}

# Train the model
H = model.fit(
    trainImages, trainTargets,
    validation_data=(testImages, testTargets),
    batch_size=BATCH_SIZE,
    epochs=NUM_EPOCHS,
    callbacks=[early_stopping],
    verbose=1
)

```

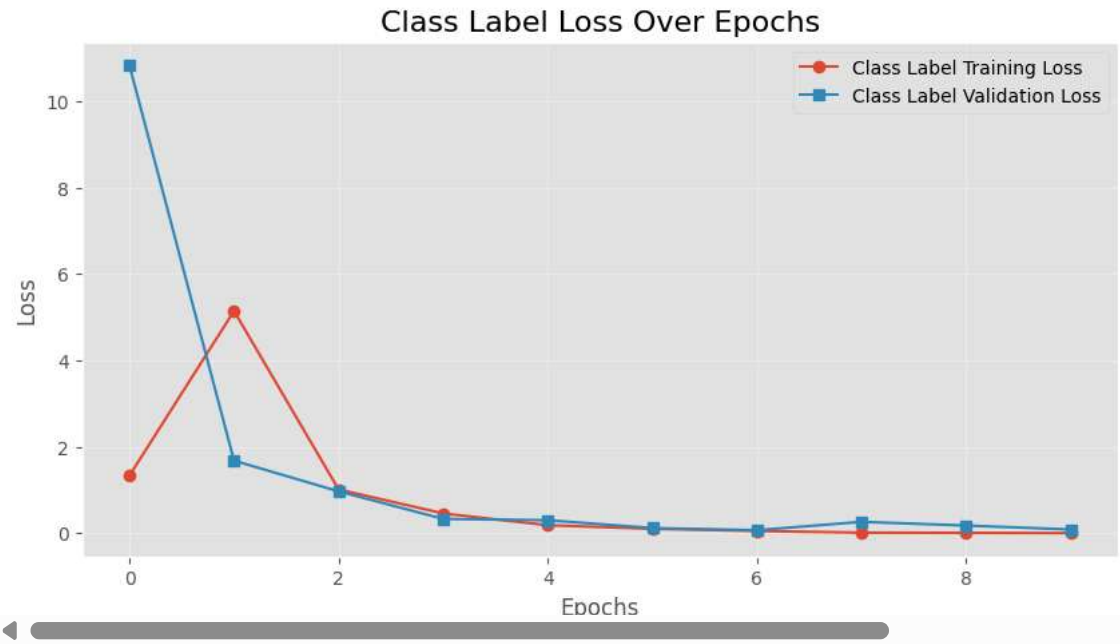
```
Transformed trainLabels: [[0 1 0]
 [0 1 0]
 [0 0 1]
 [1 0 0]
 [0 0 1]]
Transformed testLabels: [[1 0 0]
 [0 0 1]
 [0 1 0]
 [0 0 1]
 [0 1 0]]
Classes: ['Butterfly' 'Dalmatian' 'Dolphin']
Epoch 1/10
5/5 ━━━━━━━━━━━━━━━━━━━━ 18s 2s/step - bounding_box_accuracy: 0.0000e+00 - bounding_box_loss: 30799.7402 - class_label_accuracy: 0.4490
Epoch 2/10
5/5 ━━━━━━━━━━━━━━━━━━━━ 3s 539ms/step - bounding_box_accuracy: 0.0000e+00 - bounding_box_loss: 30538.4453 - class_label_accuracy: 0.464
Epoch 3/10
5/5 ━━━━━━━━━━━━━━━━━━━━ 5s 540ms/step - bounding_box_accuracy: 0.0000e+00 - bounding_box_loss: 30233.6895 - class_label_accuracy: 0.494
Epoch 4/10
5/5 ━━━━━━━━━━━━━━━━━━━━ 5s 540ms/step - bounding_box_accuracy: 0.0000e+00 - bounding_box_loss: 30724.6953 - class_label_accuracy: 0.840
Epoch 5/10
5/5 ━━━━━━━━━━━━━━━━━━━━ 5s 576ms/step - bounding_box_accuracy: 0.0000e+00 - bounding_box_loss: 31535.8242 - class_label_accuracy: 0.945
Epoch 6/10
5/5 ━━━━━━━━━━━━━━━━━━━━ 5s 547ms/step - bounding_box_accuracy: 0.0000e+00 - bounding_box_loss: 30574.8242 - class_label_accuracy: 0.980
Epoch 7/10
5/5 ━━━━━━━━━━━━━━━━━━━━ 3s 554ms/step - bounding_box_accuracy: 0.0000e+00 - bounding_box_loss: 31314.2344 - class_label_accuracy: 0.987
Epoch 8/10
5/5 ━━━━━━━━━━━━━━━━━━━━ 3s 542ms/step - bounding_box_accuracy: 0.0000e+00 - bounding_box_loss: 30974.5645 - class_label_accuracy: 1.000
Epoch 9/10
5/5 ━━━━━━━━━━━━━━━━━━━━ 3s 545ms/step - bounding_box_accuracy: 0.0000e+00 - bounding_box_loss: 30733.9082 - class_label_accuracy: 1.000
Epoch 10/10
5/5 ━━━━━━━━━━━━━━━━━━━━ 3s 573ms/step - bounding_box_accuracy: 0.0000e+00 - bounding_box_loss: 30725.3789 - class_label_accuracy: 1.000
```

Loss Visualization

```
import matplotlib.pyplot as plt

# Extract loss values from the training history
bbox_loss = H.history["bounding_box_loss"]            # Training loss for bounding boxes
val_bbox_loss = H.history["val_bounding_box_loss"]    # Validation loss for bounding boxes
class_label_loss = H.history["class_label_loss"]      # Training loss for class labels
val_class_label_loss = H.history["val_class_label_loss"]  # Validation loss for class labels

```

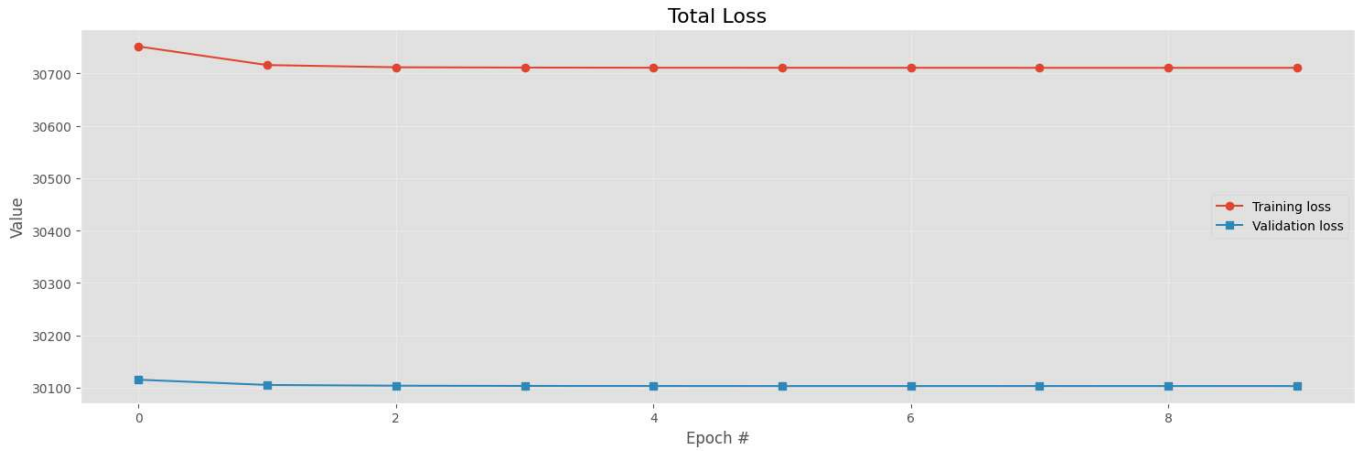
```
# Plot training and validation loss for bounding boxes
plt.figure(figsize=(10, 5))
plt.plot(bbox_loss, label="Bounding Box Training Loss", marker='o')
plt.plot(val_bbox_loss, label="Bounding Box Validation Loss", marker='s')
plt.title("Bounding Box Loss Over Epochs", fontsize=16)
plt.xlabel("Epochs", fontsize=12)
plt.ylabel("Loss", fontsize=12)
plt.legend()
plt.grid(alpha=0.3)
plt.show()

# Plot training and validation loss for class labels
plt.figure(figsize=(10, 5))
plt.plot(class_label_loss, label="Class Label Training Loss", marker='o')
plt.plot(val_class_label_loss, label="Class Label Validation Loss", marker='s')
plt.title("Class Label Loss Over Epochs", fontsize=16)
plt.xlabel("Epochs", fontsize=12)
plt.ylabel("Loss", fontsize=12)
plt.legend()
plt.grid(alpha=0.3)
plt.show()

```

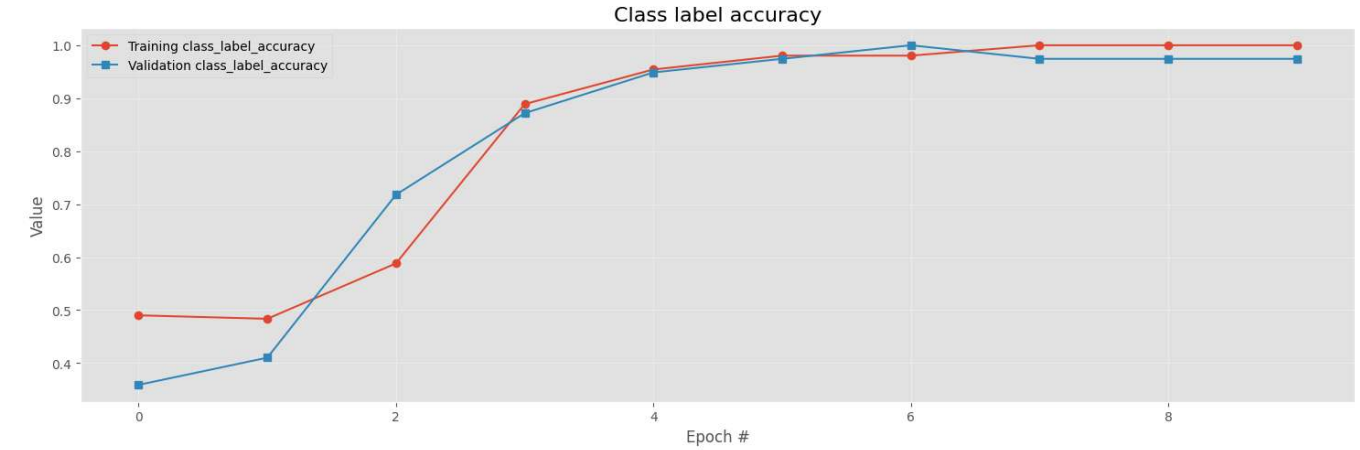


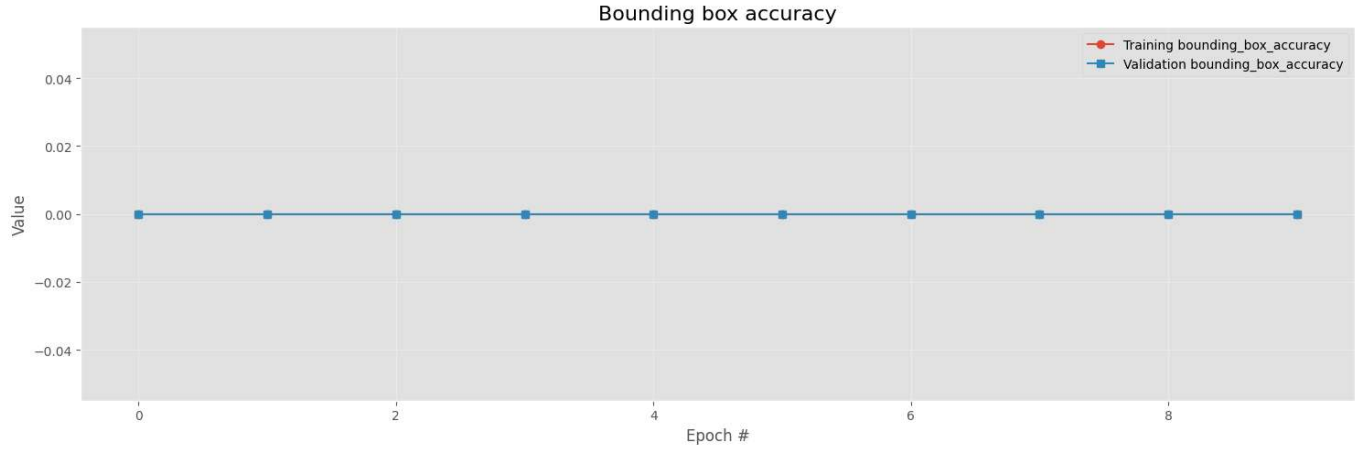


Plotting Multiple Metric

```
import matplotlib.pyplot as plt
import numpy as np

# Define the loss names and metrics to plot
lossNames = ["loss", "class_label_accuracy", "bounding_box_accuracy"]

# Define the number of epochs
N = np.arange(0, NUM_EPOCHS)

# Set the plot style
plt.style.use("ggplot")

# Create subplots for each metric
(fig, ax) = plt.subplots(len(lossNames), 1, figsize=(15, 5 * len(lossNames)))

# Loop over the loss names
for (i, l) in enumerate(lossNames):
    # Plot the metric for both training and validation data
    title = f"Total Loss" if l == "loss" else l.replace("_", " ").capitalize()
    ax[i].set_title(title, fontsize=16)
    ax[i].set_xlabel("Epoch #", fontsize=12)
    ax[i].set_ylabel("Value", fontsize=12)
    ax[i].plot(N, H.history[l], label=f"Training {l}", marker="o")
    ax[i].plot(N, H.history["val_" + l], label=f"Validation {l}", marker="s")
    ax[i].legend(fontsize=10)
    ax[i].grid(alpha=0.3)

# Adjust layout for better spacing
plt.tight_layout()
plt.show()

```


Total Loss
Training loss
Validation loss
Value
Epoch #


Class label accuracy
Training class_label_accuracy
Validation class_label_accuracy
Value
Epoch #


Bounding box accuracy
Training bounding_box_accuracy
Validation bounding_box_accuracy
Value
Epoch #

Save Model`

```
model_path = "/content/drive/MyDrive/Colab Notebooks/final_bounding_box_regression_and_classification_model.keras"
model.save(model_path)
print(f"Model saved to: {model_path}")

```

```
Model saved to: /content/drive/MyDrive/Colab Notebooks/final_bounding_box_regression_and_classification_model.keras
```

Save LabelBinarizer

```
import pickle

label_binarizer_path = "/content/drive/MyDrive/Colab Notebooks/label_binarizer.pkl"
with open(label_binarizer_path, "wb") as f:
    pickle.dump(lb, f)
print(f"LabelBinarizer saved to: {label_binarizer_path}")

```

```
LabelBinarizer saved to: /content/drive/MyDrive/Colab Notebooks/label_binarizer.pkl
```

Load Model:

```
from tensorflow.keras.models import load_model

model = load_model(model_path)
print("Model loaded successfully!")

```

```
Model loaded successfully!
```

Load LabelBinarizer:`

```
with open(label_binarizer_path, "rb") as f:
    lb = pickle.load(f)
print("LabelBinarizer loaded successfully!")

```

```
LabelBinarizer loaded successfully!
```

Ensure Correct Paths for Images and Annotations

```
image_dir = "/content/Caltech_101_Reduced"
annotations_dir = "/content/Caltech_101_Reduced"

```

Predict and Visualize Bounding Boxes

```
import numpy as np
import cv2
import imutils
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import load_img, img_to_array

# Iterate through each image path in the list of imagePaths
for imagePath in imagePaths:
    try:
        # Load and preprocess the input image
        image = load_img(imagePath, target_size=(224, 224))
        image = img_to_array(image) / 255.0
        image = np.expand_dims(image, axis=0)

        # Make predictions
        (boxPreds, labelPreds) = model.predict(image)
        print(f"Raw bounding box predictions: {boxPreds[0]}")
        print(f"Raw label predictions: {labelPreds[0]}")

        # Get the predicted bounding box and class label
        (startX, startY, endX, endY) = boxPreds[0]
```

```
        labelIndex = np.argmax(labelPreds, axis=1)[0]
        label = lb.classes_[labelIndex]
        print(f"Predicted class: {label}")

        # Load the original image
        original_image = cv2.imread(imagePath)
        if original_image is None:
            print(f"[Error] Failed to load image: {imagePath}")
            continue
        original_image = imutils.resize(original_image, width=600)
        (h, w) = original_image.shape[:2]

        # Scale bounding box coordinates
        startX = max(0, min(int(startX * w), w))
        startY = max(0, min(int(startY * h), h))
        endX = max(0, min(int(endX * w), w))
        endY = max(0, min(int(endY * h), h))
        print(f"Scaled bounding box: {(startX, startY, endX, endY)}")

        # Draw the bounding box and label
        y = startY - 10 if startY - 10 > 10 else startY + 10
        cv2.rectangle(original_image, (startX, startY), (endX, endY), (0, 255, 0), 2)
        cv2.putText(original_image, label, (startX, y), cv2.FONT_HERSHEY_SIMPLEX, 0.65, (0, 255, 0), 2)

        # Convert the image to RGB and display
        original_image = cv2.cvtColor(original_image, cv2.COLOR_BGR2RGB)
        plt.imshow(original_image)
        plt.axis("off")
        plt.show()

    except Exception as e:
        print(f"[Error] {e}")

```

```
[Error] [Errno 2] No such file or directory: '/content/Caltech_101_Reduced/caltech101_classification/butterfly/\ufeffimage_0001.jpg'
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 966ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9991226e-01 2.5683417e-07 8.7531022e-05]
Predicted class: Butterfly
Scaled bounding box: (600, 814, 600, 814)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9998724e-01 1.2600442e-05 1.3293445e-07]
Predicted class: Butterfly
Scaled bounding box: (600, 384, 600, 384)
```

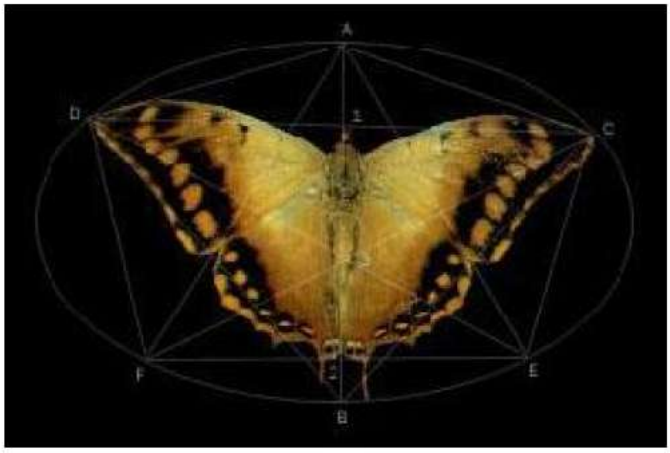

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9962425e-01 2.2968136e-04 1.4603655e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 542, 600, 542)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.7487170e-01 2.9137087e-04 2.4836939e-02]
Predicted class: Butterfly
```

```
Scaled bounding box: (600, 556, 600, 556)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9999988e-01 1.0109969e-07 3.9312766e-09]
Predicted class: Butterfly
Scaled bounding box: (600, 542, 600, 542)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9972934e-01 2.3652725e-04 3.4122593e-05]
Predicted class: Butterfly
Scaled bounding box: (600, 400, 600, 400)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9984729e-01 1.4858632e-04 4.0410291e-06]
Predicted class: Butterfly
Scaled bounding box: (600, 504, 600, 504)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [0.99251384 0.00580865 0.00167752]
Predicted class: Butterfly
Scaled bounding box: (600, 506, 600, 506)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 18ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [0.9953566  0.00246068 0.00218262]
Predicted class: Butterfly
Scaled bounding box: (600, 400, 600, 400)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9644452e-01 7.7502109e-04 2.7804219e-03]
Predicted class: Butterfly
Scaled bounding box: (600, 454, 600, 454)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9979788e-01 2.0207009e-04 7.0284500e-08]
Predicted class: Butterfly
Scaled bounding box: (600, 528, 600, 528)
```

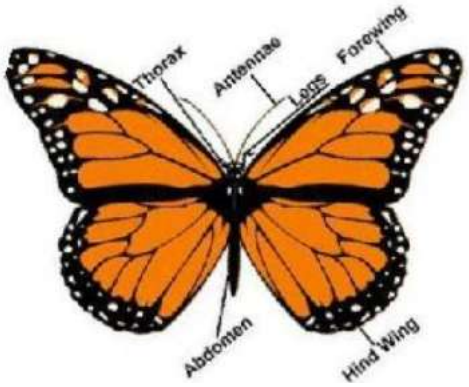
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.000000e+00 1.691577e-08 1.262069e-10]
Predicted class: Butterfly
Scaled bounding box: (600, 610, 600, 610)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9719381e-01 2.7599586e-03 4.6246092e-05]
Predicted class: Butterfly
Scaled bounding box: (600, 418, 600, 418)
```

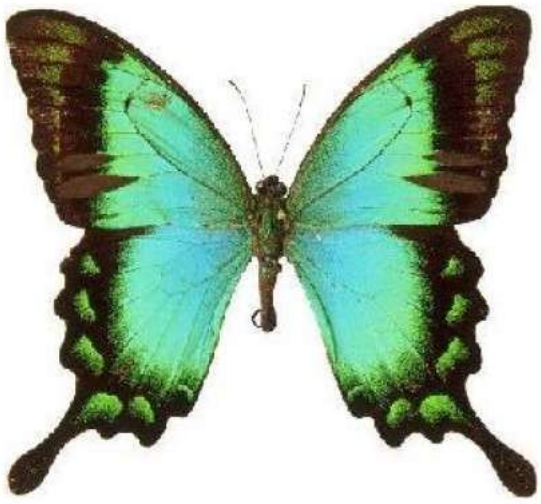
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9995780e-01 2.6638505e-05 1.5657506e-05]
Predicted class: Butterfly
Scaled bounding box: (600, 404, 600, 404)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9846292e-01 4.0941051e-04 1.1276848e-03]
Predicted class: Butterfly
Scaled bounding box: (600, 376, 600, 376)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.0000000e+00 2.3663524e-10 8.7874255e-13]
Predicted class: Butterfly
Scaled bounding box: (600, 606, 600, 606)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9999535e-01 3.1505176e-06 1.5821807e-06]
Predicted class: Butterfly
Scaled bounding box: (600, 556, 600, 556)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9887604e-01 1.1196774e-03 4.3431128e-06]
Predicted class: Butterfly
Scaled bounding box: (600, 572, 600, 572)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9991715e-01 8.4707781e-06 7.4423340e-05]
Predicted class: Butterfly
Scaled bounding box: (600, 414, 600, 414)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
```

```
Raw label predictions: [1.0000000e+00 1.5198669e-08 4.4031782e-11]
Predicted class: Butterfly
Scaled bounding box: (600, 578, 600, 578)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9327928e-01 1.7507616e-04 6.5455879e-03]
Predicted class: Butterfly
Scaled bounding box: (600, 360, 600, 360)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9844885e-01 8.9039665e-04 6.6074036e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 450, 600, 450)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9999559e-01 4.3593568e-06 1.1670805e-09]
Predicted class: Butterfly
Scaled bounding box: (600, 516, 600, 516)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.958677e-01 6.075964e-05 4.071541e-03]
Predicted class: Butterfly
Scaled bounding box: (600, 512, 600, 512)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [7.2726506e-01 1.5652306e-04 2.7257845e-01]
Predicted class: Butterfly
Scaled bounding box: (600, 659, 600, 659)
```

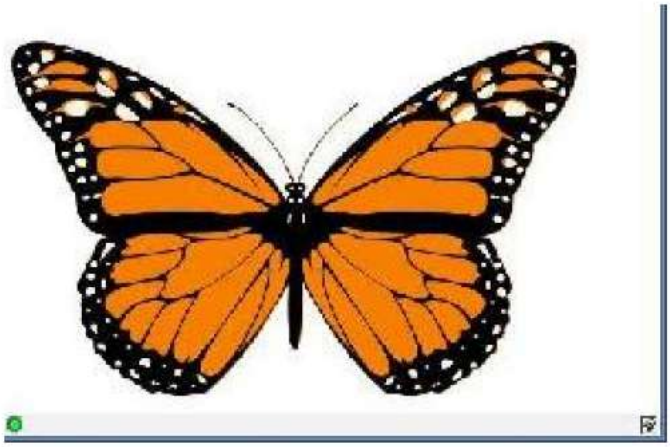
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.99630809e-01 2.64871866e-04 1.04307604e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 446, 600, 446)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9977547e-01 1.7774801e-04 4.6832618e-05]
Predicted class: Butterfly
Scaled bounding box: (600, 450, 600, 450)
```

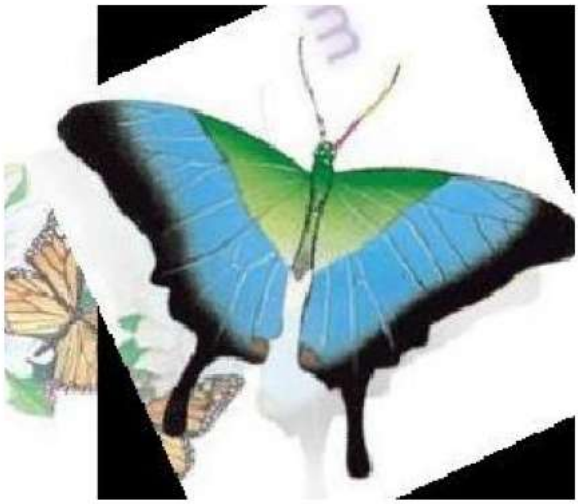

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.8963296e-01 1.0137677e-02 2.2938161e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 486, 600, 486)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9986935e-01 1.2925222e-04 1.3832450e-06]
Predicted class: Butterfly
Scaled bounding box: (600, 370, 600, 370)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.0000000e+00 5.4712201e-10 1.7473317e-12]
Predicted class: Butterfly
Scaled bounding box: (600, 396, 600, 396)
```



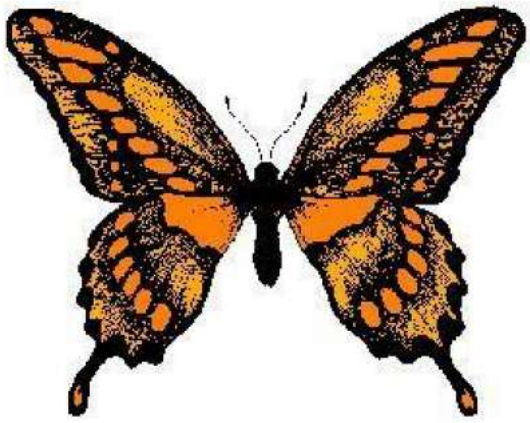
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9814057e-01 5.6040448e-05 1.8033434e-03]
Predicted class: Butterfly
Scaled bounding box: (600, 520, 600, 520)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9999726e-01 2.1389374e-06 5.5042580e-07]
Predicted class: Butterfly
Scaled bounding box: (600, 486, 600, 486)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9973112e-01 4.2078667e-05 2.2675861e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 428, 600, 428)
```


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.0000000e+00 4.4455856e-10 1.7057245e-13]
Predicted class: Butterfly
Scaled bounding box: (600, 510, 600, 510)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.980677e-01 1.197214e-03 7.351380e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 396, 600, 396)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
```

```
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9999261e-01 8.7802249e-07 6.5290583e-06]
Predicted class: Butterfly
Scaled bounding box: (600, 544, 600, 544)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [0.7149838  0.01766028 0.26735586]
Predicted class: Butterfly
Scaled bounding box: (600, 554, 600, 554)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9531269e-01 2.1338121e-04 4.4739293e-03]
Predicted class: Butterfly
Scaled bounding box: (600, 580, 600, 580)
```

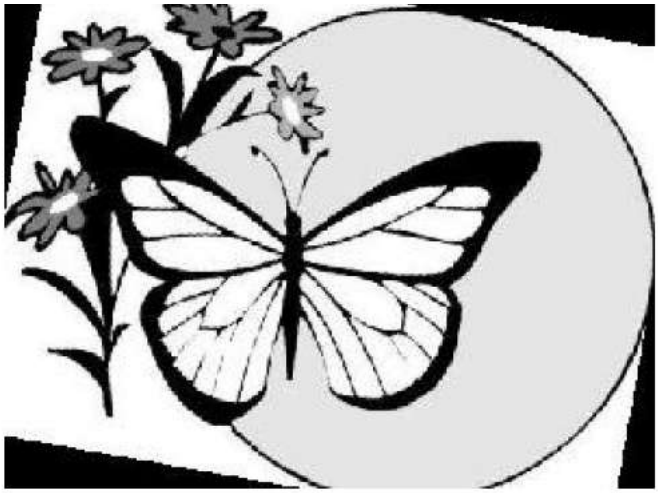
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9999928e-01 4.2923622e-07 2.2633941e-07]
Predicted class: Butterfly
```

```
Predicted class: Butterfly
Scaled bounding box: (600, 853, 600, 853)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.999993e-01 2.709093e-07 5.168475e-07]
Predicted class: Butterfly
Scaled bounding box: (600, 622, 600, 622)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.505533e-01 4.943817e-02 8.540770e-06]
Predicted class: Butterfly
Scaled bounding box: (600, 464, 600, 464)
```

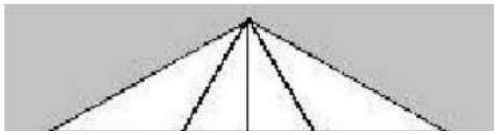
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9997747e-01 6.9386174e-06 1.5634863e-05]
Predicted class: Butterfly
Scaled bounding box: (600, 456, 600, 456)
```

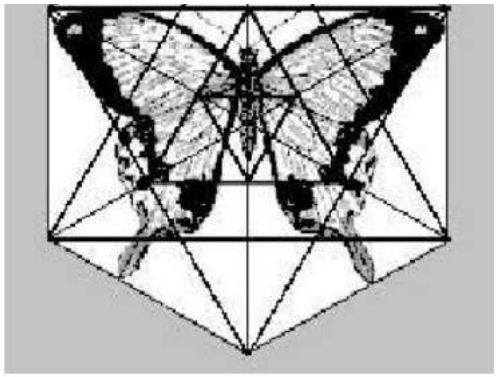


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9878484e-01 6.5839279e-04 5.5668835e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 396, 600, 396)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9511790e-01 4.6198783e-03 2.6231518e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 542, 600, 542)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9999917e-01 7.7489943e-07 3.3293489e-08]
Predicted class: Butterfly
Scaled bounding box: (600, 606, 600, 606)
```



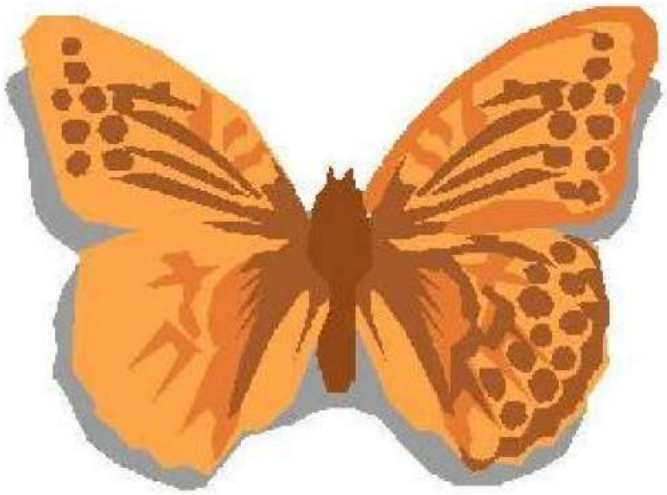


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [0.9439224  0.02972392 0.02635363]
Predicted class: Butterfly
Scaled bounding box: (600, 450, 600, 450)
```

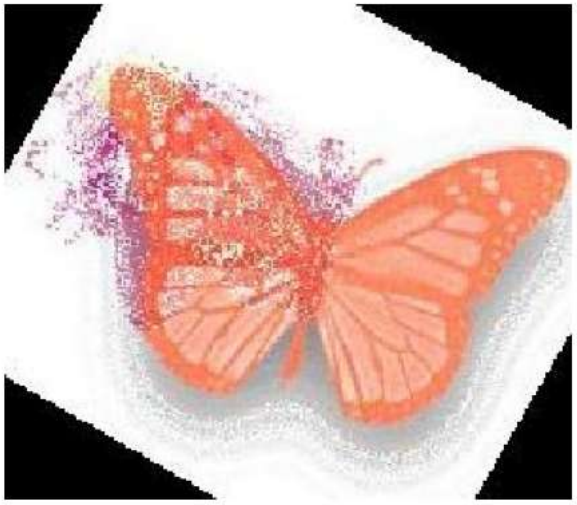
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9973398e-01 2.6050943e-04 5.4915722e-06]
Predicted class: Butterfly
Scaled bounding box: (600, 394, 600, 394)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.0000000e+00 2.9629266e-08 8.8478502e-10]
Predicted class: Butterfly
Scaled bounding box: (600, 480, 600, 480)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9303997e-01 1.2345941e-04 6.8365480e-03]
Predicted class: Butterfly
Scaled bounding box: (600, 444, 600, 444)
```

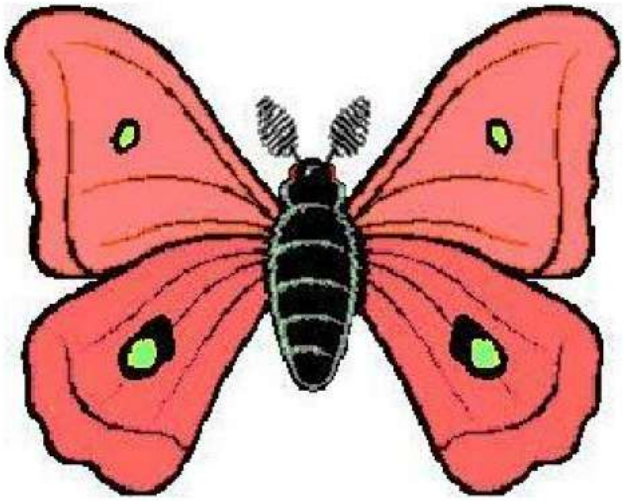


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9951196e-01 3.0895058e-05 4.5719114e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 522, 600, 522)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9993217e-01 6.6047243e-05 1.8313101e-06]
Predicted class: Butterfly
Scaled bounding box: (600, 454, 600, 454)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9999881e-01 1.2390743e-06 3.6583803e-08]
Predicted class: Butterfly
Scaled bounding box: (600, 480, 600, 480)
```

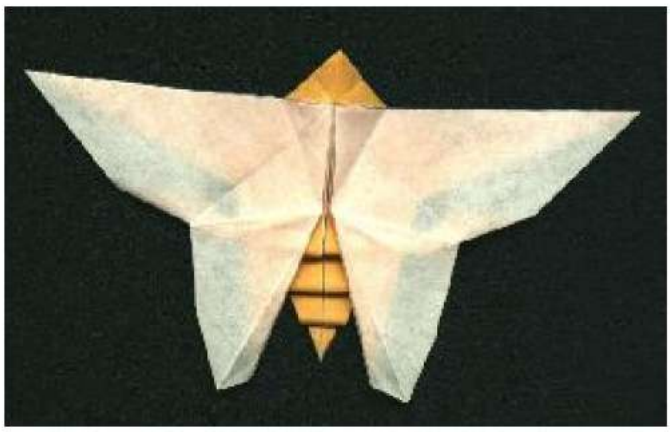


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9976879e-01 8.3899895e-06 2.2283944e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 600, 600, 600)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9999404e-01 5.9800554e-06 1.7984787e-10]
Predicted class: Butterfly
Scaled bounding box: (600, 452, 600, 452)
```

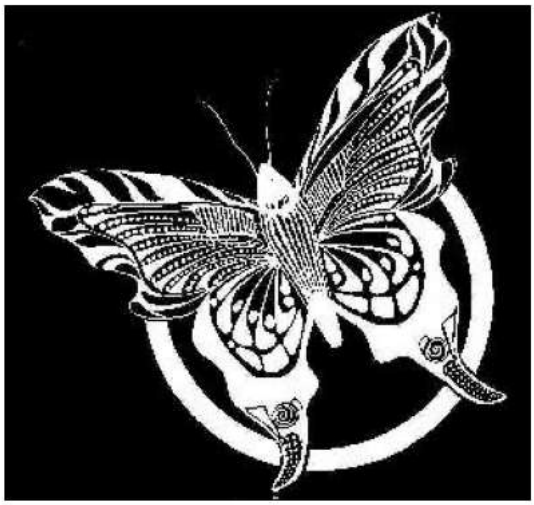
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9977881e-01 1.4452518e-05 2.0680463e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 756, 600, 756)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9878103e-01 3.3368636e-04 8.8522077e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 380, 600, 380)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9992037e-01 1.2740631e-05 6.6887347e-05]
Predicted class: Butterfly
Scaled bounding box: (600, 450, 600, 450)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9999976e-01 4.0949296e-08 2.1407986e-07]
Predicted class: Butterfly
Scaled bounding box: (600, 564, 600, 564)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9992085e-01 6.2775463e-05 1.6309066e-05]
Predicted class: Butterfly
Scaled bounding box: (600, 570, 600, 570)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9926156e-01 1.5821465e-04 5.8028474e-04]
Predicted class: Butterfly
Scaled bounding box: (600, 538, 600, 538)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
```

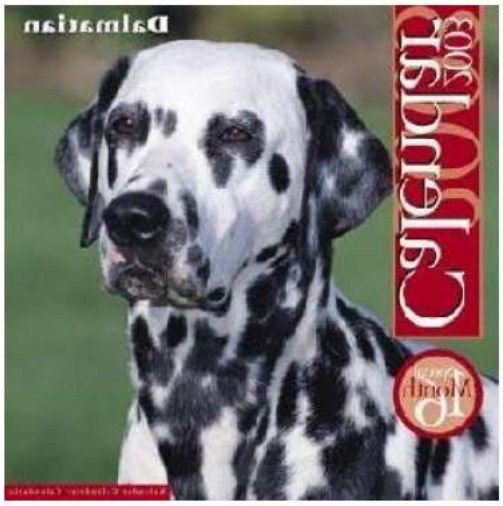
```
Raw label predictions: [9.9997556e-01 2.4490104e-05 1.4579847e-10]
Predicted class: Butterfly
Scaled bounding box: (600, 426, 600, 426)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9999082e-01 5.6379899e-06 3.5592013e-06]
Predicted class: Butterfly
Scaled bounding box: (600, 565, 600, 565)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.960741e-01 4.724394e-04 3.453498e-03]
Predicted class: Butterfly
Scaled bounding box: (600, 550, 600, 550)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.99876857e-01 1.23145262e-04 1.00901405e-10]
Predicted class: Butterfly
Scaled bounding box: (600, 606, 600, 606)
```


```
[Error] [Errno 2] No such file or directory: '/content/Caltech_101_Reduced/caltech101_classification/dalmatian/\ufeffimage_0001.jpg'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [5.091409e-10 1.000000e+00 2.174072e-11]
Predicted class: Dalmatian
Scaled bounding box: (600, 600, 600, 600)
```




```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.9215213e-08 9.9999762e-01 2.3916989e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 524, 600, 524)
```

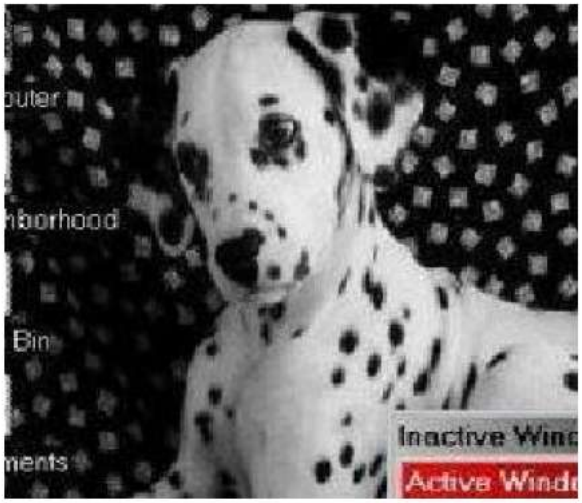

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [5.3388587e-09 1.0000000e+00 7.7625710e-13]
Predicted class: Dalmatian
Scaled bounding box: (600, 737, 600, 737)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.5388377e-03 9.9821681e-01 2.4433408e-04]
Predicted class: Dalmatian
Scaled bounding box: (600, 800, 600, 800)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.4329795e-05 9.9968255e-01 2.9320244e-04]
Predicted class: Dalmatian
Scaled bounding box: (600, 740, 600, 740)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [3.5347431e-10 1.0000000e+00 3.4364356e-10]
Predicted class: Dalmatian
Scaled bounding box: (600, 631, 600, 631)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.0211301e-08 1.0000000e+00 1.8284568e-10]
Predicted class: Dalmatian
Scaled bounding box: (600, 518, 600, 518)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.3078643e-08 1.0000000e+00 1.8083538e-11]
Predicted class: Dalmatian
Scaled bounding box: (600, 482, 600, 482)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [3.4018404e-07 9.9997008e-01 2.9610688e-05]
Predicted class: Dalmatian
Scaled bounding box: (600, 566, 600, 566)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.1797924e-07 9.9999785e-01 1.6777038e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 576, 600, 576)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9075613e-08 9.9999976e-01 7.8186531e-08]
Predicted class: Dalmatian
Scaled bounding box: (600, 544, 600, 544)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 34ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [5.177131e-11 1.000000e+00 9.096873e-13]
Predicted class: Dalmatian
Scaled bounding box: (600, 578, 600, 578)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.7646996e-10 1.0000000e+00 2.9254862e-13]
Predicted class: Dalmatian
Scaled bounding box: (600, 588, 600, 588)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [3.1674512e-07 9.9999952e-01 1.2905120e-07]
Predicted class: Dalmatian
Scaled bounding box: (600, 618, 600, 618)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.0753996e-07 9.9999917e-01 6.5602990e-07]
Predicted class: Dalmatian
Scaled bounding box: (600, 602, 600, 602)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.7433282e-06 9.9997962e-01 1.8644032e-05]
Predicted class: Dalmatian
Scaled bounding box: (600, 596, 600, 596)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [7.3270894e-05 9.9979109e-01 1.3558193e-04]
Predicted class: Dalmatian
Scaled bounding box: (600, 370, 600, 370)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.389676e-09 1.000000e+00 5.194426e-09]
Predicted class: Dalmatian
Scaled bounding box: (600, 396, 600, 396)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.7685146e-05 9.9997234e-01 1.6621067e-08]
Predicted class: Dalmatian
Scaled bounding box: (600, 498, 600, 498)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9384474e-09 1.0000000e+00 1.2247995e-09]
Predicted class: Dalmatian
Scaled bounding box: (600, 382, 600, 382)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.0066812e-07 9.9999821e-01 1.6475581e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 570, 600, 570)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [3.5520085e-07 9.9999964e-01 5.7781220e-09]
Predicted class: Dalmatian
```

```
Scaled bounding box: (600, 450, 600, 450)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.3171239e-06 9.9995041e-01 4.8269110e-05]
Predicted class: Dalmatian
Scaled bounding box: (600, 436, 600, 436)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.6142623e-09 1.0000000e+00 6.7656714e-11]
Predicted class: Dalmatian
Scaled bounding box: (600, 532, 600, 532)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.9301864e-06 9.9999583e-01 1.1418533e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 640, 600, 640)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.3266290e-07 9.9999976e-01 1.9403557e-09]
Predicted class: Dalmatian
Scaled bounding box: (600, 552, 600, 552)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.1985867e-06 9.9999571e-01 3.1320649e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 524, 600, 524)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.3840052e-05 9.9998534e-01 7.7865627e-07]
Predicted class: Dalmatian
Scaled bounding box: (600, 552, 600, 552)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.5106433e-06 9.9999750e-01 3.5090755e-09]
Predicted class: Dalmatian
Scaled bounding box: (600, 664, 600, 664)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [3.5274323e-07 9.9999762e-01 2.0735094e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 462, 600, 462)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.2137293e-10 1.0000000e+00 2.3261224e-10]
Predicted class: Dalmatian
Scaled bounding box: (600, 492, 600, 492)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [3.8705134e-06 9.9998236e-01 1.3835867e-05]
Predicted class: Dalmatian
Scaled bounding box: (600, 470, 600, 470)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [7.3802024e-05 9.9974018e-01 1.8609685e-04]
Predicted class: Dalmatian
Scaled bounding box: (600, 526, 600, 526)
```


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.9519630e-06 9.9999809e-01 1.5431704e-09]
Predicted class: Dalmatian
Scaled bounding box: (600, 446, 600, 446)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.0719696e-04 9.9987864e-01 1.4145365e-05]
Predicted class: Dalmatian
Scaled bounding box: (600, 446, 600, 446)
```

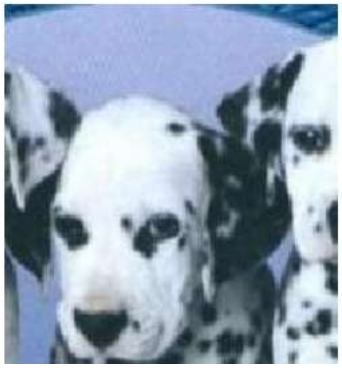
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.1464393e-06 9.9999857e-01 2.1058121e-07]
Predicted class: Dalmatian
Scaled bounding box: (600, 679, 600, 679)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [6.574985e-11 1.000000e+00 1.570322e-11]
Predicted class: Dalmatian
Scaled bounding box: (600, 414, 600, 414)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.9626692e-08 1.0000000e+00 3.1981749e-08]
Predicted class: Dalmatian
Scaled bounding box: (600, 923, 600, 923)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.6700366e-10 1.0000000e+00 4.7247641e-09]
Predicted class: Dalmatian
Scaled bounding box: (600, 779, 600, 779)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 41ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.0530455e-07 9.9999607e-01 3.6136976e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 891, 600, 891)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [5.209510e-08 1.000000e+00 7.517403e-09]
Predicted class: Dalmatian
Scaled bounding box: (600, 496, 600, 496)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 35ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.9278957e-08 1.0000000e+00 1.9158685e-08]
Predicted class: Dalmatian
Scaled bounding box: (600, 392, 600, 392)
```


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 37ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [3.949678e-07 9.999995e-01 6.504703e-08]
Predicted class: Dalmatian
Scaled bounding box: (600, 376, 600, 376)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
```

```
Raw label predictions: [1.1039371e-05 9.9952996e-01 4.5895748e-04]
Predicted class: Dalmatian
Scaled bounding box: (600, 394, 600, 394)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 32ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.0235246e-12 1.0000000e+00 1.8285777e-15]
Predicted class: Dalmatian
Scaled bounding box: (600, 352, 600, 352)
```


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [7.8648287e-07 9.9981183e-01 1.8730493e-04]
Predicted class: Dalmatian
Scaled bounding box: (600, 552, 600, 552)
```


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [5.2435206e-07 9.9999905e-01 4.8935340e-07]
Predicted class: Dalmatian
Scaled bounding box: (600, 392, 600, 392)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [7.7776895e-07 9.9999905e-01 6.0717262e-08]
Predicted class: Dalmatian
Scaled bounding box: (600, 446, 600, 446)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.1026309e-03 9.9889189e-01 5.4809630e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 659, 600, 659)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [6.0305399e-07 9.9999940e-01 5.1877884e-09]
Predicted class: Dalmatian
Scaled bounding box: (600, 348, 600, 348)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 40ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.3196924e-07 9.9998987e-01 1.0048734e-05]
Predicted class: Dalmatian
Scaled bounding box: (600, 382, 600, 382)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.05741e-08 1.00000e+00 2.50346e-10]
Predicted class: Dalmatian
Scaled bounding box: (600, 416, 600, 416)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.048978e-10 1.000000e+00 6.747771e-11]
Predicted class: Dalmatian
Scaled bounding box: (600, 430, 600, 430)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
```

```
Raw label predictions: [1.7929402e-06 9.9999666e-01 1.5561403e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 446, 600, 446)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [7.9924978e-10 1.0000000e+00 1.8903785e-09]
Predicted class: Dalmatian
Scaled bounding box: (600, 476, 600, 476)
```

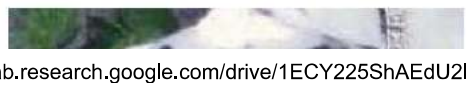
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.8833672e-06 9.9999809e-01 2.9461265e-09]
Predicted class: Dalmatian
Scaled bounding box: (600, 446, 600, 446)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.1659752e-05 9.9994636e-01 4.1981653e-05]
Predicted class: Dalmatian
Scaled bounding box: (600, 492, 600, 492)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.3659607e-06 9.9999797e-01 7.5618885e-07]
Predicted class: Dalmatian
Scaled bounding box: (600, 452, 600, 452)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.7701122e-07 9.9999964e-01 2.2753197e-07]
Predicted class: Dalmatian
Scaled bounding box: (600, 502, 600, 502)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.7335147e-04 9.9900633e-01 2.0390213e-05]
Predicted class: Dalmatian
Scaled bounding box: (600, 652, 600, 652)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [3.9943676e-05 9.9995542e-01 4.6495034e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 466, 600, 466)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.1805094e-05 9.9997520e-01 3.0122599e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 428, 600, 428)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [9.9566944e-07 9.9999654e-01 2.4505557e-06]
Predicted class: Dalmatian
Scaled bounding box: (600, 410, 600, 410)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.0345210e-07 9.9999976e-01 1.4380889e-08]
Predicted class: Dalmatian
Scaled bounding box: (600, 352, 600, 352)
```

```
[Error] [Errno 2] No such file or directory: '/content/Caltech_101_Reduced/caltech101_classification/dolphin/\ufeffimage_0001.jpg'
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [3.1683362e-06 6.0418034e-07 9.9999619e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 358, 600, 358)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.6361545e-03 1.0228384e-04 9.9826163e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 332, 600, 332)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 30ms/step
```

```
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.2064079e-04 4.7566145e-06 9.9987459e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 518, 600, 518)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.00459256e-04 1.86213333e-06 9.99897718e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 450, 600, 450)
```

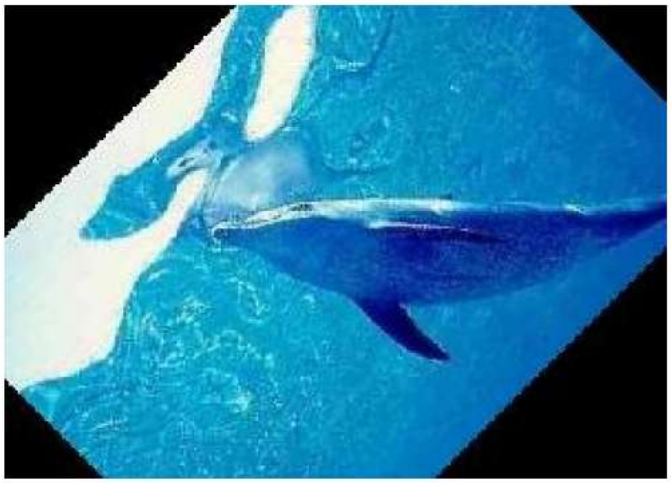
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.9101680e-04 3.6689005e-05 9.9967229e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 554, 600, 554)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [0.02914854 0.85813916 0.11271233]
```

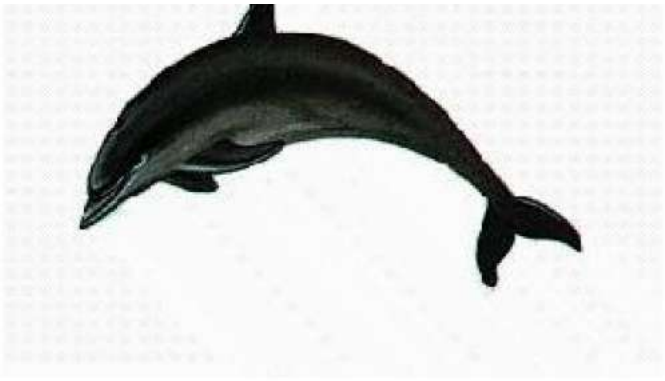
```
Predicted class: Dalmatian
Scaled bounding box: (600, 366, 600, 366)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.9812070e-06 2.0117723e-06 9.9999297e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 450, 600, 450)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.9341629e-03 9.2797582e-06 9.9805647e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 428, 600, 428)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.8471188e-04 2.0522584e-05 9.9979478e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 446, 600, 446)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.1031609e-03 2.3633814e-04 9.9866045e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 554, 600, 554)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [6.6599139e-04 2.4160920e-04 9.9909246e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 318, 600, 318)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.1248790e-04 3.8388393e-05 9.9984908e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 346, 600, 346)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.9838693e-04 6.3694883e-05 9.9973792e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 420, 600, 420)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.194017e-03 9.191742e-04 9.968868e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 440, 600, 440)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.5153151e-05 6.2824097e-06 9.9997854e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 440, 600, 440)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
```

```
Raw bounding box predictions: [0.99999976 1.         1.         1.        ]
Raw label predictions: [1.4629030e-03 3.6559047e-04 9.9817145e-01]
Predicted class: Dolphin
Scaled bounding box: (599, 248, 600, 248)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [5.2375482e-05 4.1196527e-05 9.9990642e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 320, 600, 320)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.9408583e-04 9.2100163e-05 9.9941385e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 386, 600, 386)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 39ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.3338559e-04 9.4033392e-05 9.9977261e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 402, 600, 402)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [5.6782654e-03 5.8422989e-04 9.9373746e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 422, 600, 422)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.5956361e-03 2.9457943e-04 9.9710983e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 480, 600, 480)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.4816034e-03 3.9956780e-05 9.9547845e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 434, 600, 434)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 38ms/step
```

```
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.6989898e-03 4.1481064e-05 9.9825948e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 294, 600, 294)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [8.4641384e-04 6.3707926e-03 9.9278283e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 322, 600, 322)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 31ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [5.1003019e-04 5.6475470e-05 9.9943346e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 380, 600, 380)
```

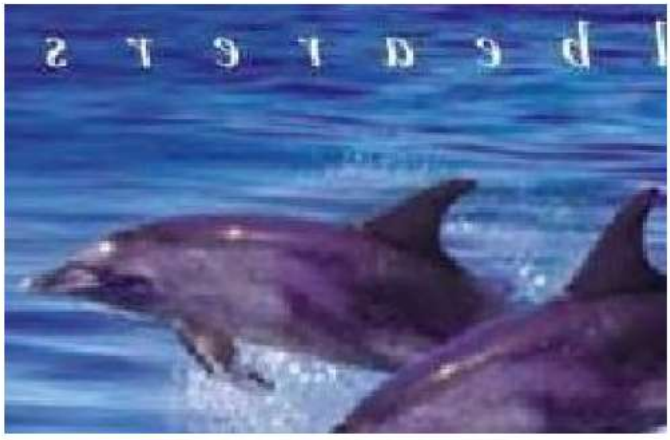
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [0.0076542  0.00443625 0.98790956]
Predicted class: Dolphin
Scaled bounding box: (600, 226, 600, 226)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [8.8226509e-05 8.3114469e-07 9.9991095e-01]
```

```
Predicted class: Dolphin
Scaled bounding box: (600, 452, 600, 452)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [8.057359e-05 7.575929e-06 9.999118e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 442, 600, 442)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.6148089e-04 5.2784844e-06 9.9983323e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 388, 600, 388)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.6288355e-03 4.4903816e-05 9.9832624e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 280, 600, 280)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [0.04474697 0.00112258 0.9541305 ]
Predicted class: Dolphin
Scaled bounding box: (600, 438, 600, 438)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.2403415e-03 7.1321166e-04 9.9504650e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 486, 600, 486)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step
Raw bounding box predictions: [0.9999994 1.        1.        1.       ]
Raw label predictions: [6.8962324e-04 1.0511041e-04 9.9920529e-01]
Predicted class: Dolphin
Scaled bounding box: (599, 452, 600, 452)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.3557462e-04 1.5861727e-03 9.9797827e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 204, 600, 204)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 33ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.7222052e-02 6.2554912e-04 9.7215241e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 480, 600, 480)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.0343044e-03 2.6361569e-04 9.9870205e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 365, 600, 365)
```

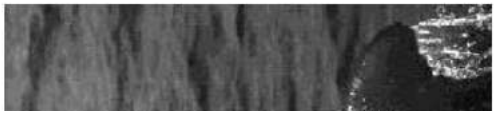
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.3794624e-04 3.3785164e-04 9.9942422e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 336, 600, 336)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.4397584e-04 1.6036285e-04 9.9959570e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 284, 600, 284)
```

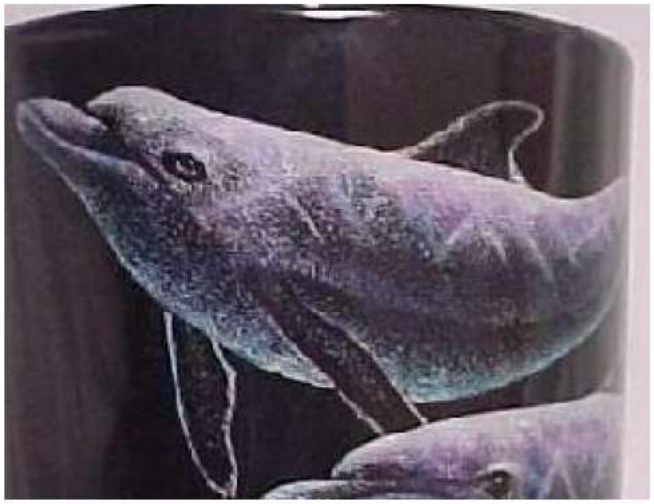
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [0.00329015 0.00151049 0.9951994 ]
Predicted class: Dolphin
Scaled bounding box: (600, 334, 600, 334)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [6.0223051e-06 3.5601574e-06 9.9999046e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 315, 600, 315)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.0814749e-04 6.0053240e-04 9.9929130e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 608, 600, 608)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.0899365e-03 9.8995632e-05 9.9781114e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 460, 600, 460)
```



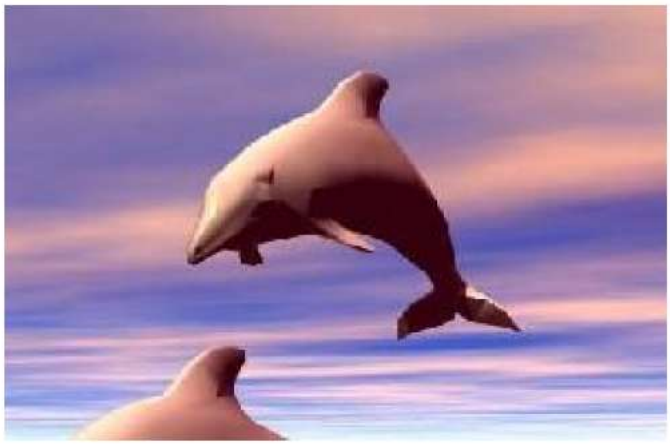
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.3403883e-04 5.9667855e-05 9.9970621e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 430, 600, 430)
```


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 25ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [6.135904e-05 2.899987e-04 9.996487e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 264, 600, 264)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [4.4184790e-06 5.8895415e-07 9.9999499e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 342, 600, 342)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [3.9874105e-04 1.5055277e-06 9.9959975e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 492, 600, 492)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [8.7793851e-05 3.1504824e-05 9.9988067e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 394, 600, 394)
```



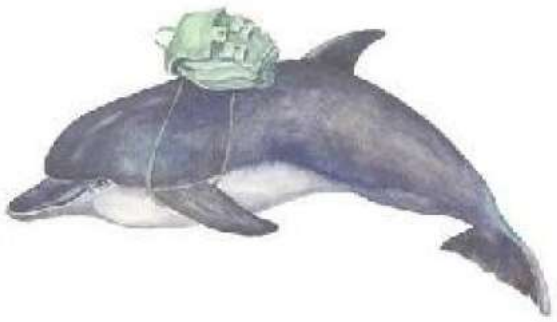
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [1.2715564e-03 9.3896793e-05 9.9863452e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 402, 600, 402)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [0.00238313 0.00292768 0.9946892 ]
Predicted class: Dolphin
Scaled bounding box: (600, 422, 600, 422)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [2.1576466e-02 9.5162285e-04 9.7747189e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 400, 600, 400)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 28ms/step
Raw bounding box predictions: [0.9999999 1.        1.        1.       ]
Raw label predictions: [0.01381231 0.00123563 0.98495203]
Predicted class: Dolphin
Scaled bounding box: (599, 442, 600, 442)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step
Raw bounding box predictions: [1. 1. 1. 1.]
Raw label predictions: [3.0776431e-04 6.9634581e-04 9.9899584e-01]
Predicted class: Dolphin
Scaled bounding box: (600, 370, 600, 370)
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 26ms/step
```