Model_Transfer Learning 01 - With data augmentation - Feature extraction

Name: Alberto Pingo
Email: 2202145 @my.ipleiria.pt

Validation dataset: train5

Directories

This section sets up the directory paths used for training, validation, and test datasets based on the repository structure.

In [ ]:
```
import os

current_dir = os.getcwd()

# TWO FOLDERS UP
data_dir = os.path.abspath(os.path.join(current_dir, os.pardir, os.pardir
test_dir = os.path.join(data_dir, 'test')
train_dir = os.path.join(data_dir, 'train')

train_dirs = []
for i in range(1, 5):
    train_dirs.append(os.path.join(train_dir, 'train' + str(i)))


validation_dir = os.path.join(data_dir, 'train', 'train5')

print(current_dir)
print(data_dir)
print(test_dir)
print(train_dir)
print(validation_dir)
```

```
/home/pws/code/IA-image-classification/notebooks/models-T
/home/pws/code/IA-image-classification/data
/home/pws/code/IA-image-classification/data/test
/home/pws/code/IA-image-classification/data/train
/home/pws/code/IA-image-classification/data/train/train5
```

Preprocessing

Load the datasets and perform initial preprocessing. Images are resized to 32x32 pixels and batched.

In [ ]:
```
from keras.utils import image_dataset_from_directory
import tensorflow as tf

# Load training datasets from train1 to train4
train_datasets = []
```

```
IMG_SIZE = 150
BATCH_SIZE = 32
train_dataset = image_dataset_from_directory(train_dirs[0], image_size=(I
# for i in range(1, 5):
#     dataset = image_dataset_from_directory(train_dirs[i-1], image_size=
#     train_datasets.append(dataset)

# train_dataset = train_datasets[0]
# for dataset in train_datasets[1:]:
#     train_dataset = train_dataset.concatenate(dataset)

# Load validation dataset
validation_dataset = image_dataset_from_directory(validation_dir, image_s


# Load test dataset
test_dataset = image_dataset_from_directory(test_dir, image_size=(IMG_SIZ

class_names = validation_dataset.class_names
class_names = [class_name.split('_')[-1] for class_name in class_names]

print(class_names)
```

```
Found 10000 files belonging to 10 classes.
Found 10000 files belonging to 10 classes.
Found 10000 files belonging to 10 classes.
['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse',
'ship', 'truck']
```

Configure the dataset for performance

In [ ]:
```
AUTOTUNE = tf.data.AUTOTUNE

train_dataset = train_dataset.cache().shuffle(1000).prefetch(buffer_size=
validation_dataset = validation_dataset.cache().prefetch(buffer_size=AUTO
test_dataset = test_dataset.cache().prefetch(buffer_size=AUTOTUNE)
```

Data Augmentation

Rendom change of flipping the image horizontally.
Random chance of moving the image horizontally and vertically [-10%, 10%].

Tried with a more complex approach to data augmentation, but the results were worse because of the small size of the images.

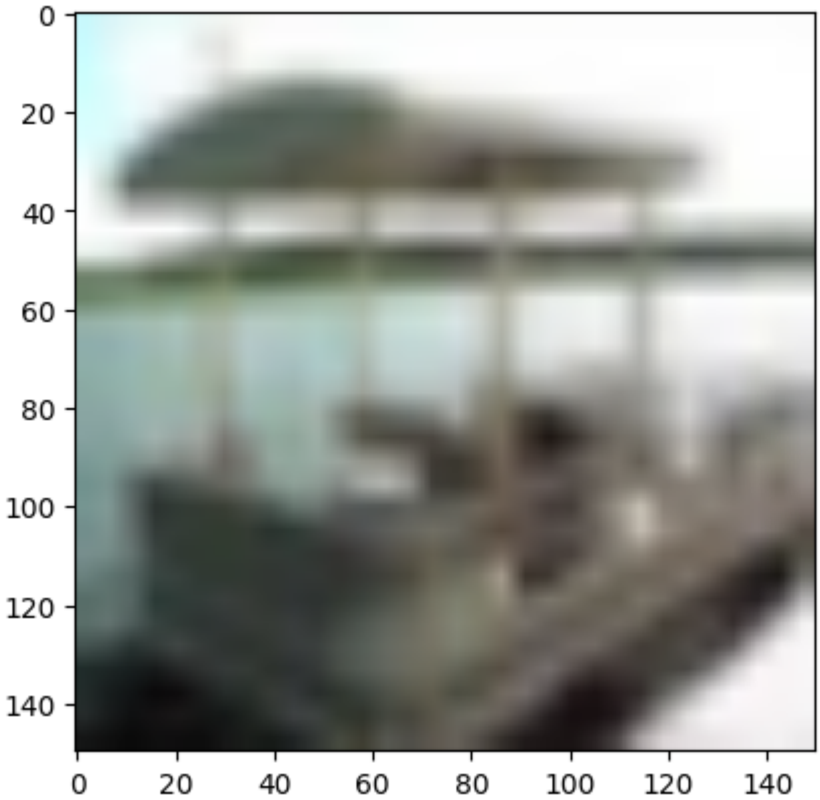
In [ ]:
```
from keras import layers

data_augmentation = tf.keras.Sequential([
    layers.RandomFlip("horizontal"),
    layers.RandomTranslation(0.1, 0.1, fill_mode='nearest'),
    layers.RandomRotation(0.03),
    layers.RandomZoom(0.03),
])
```

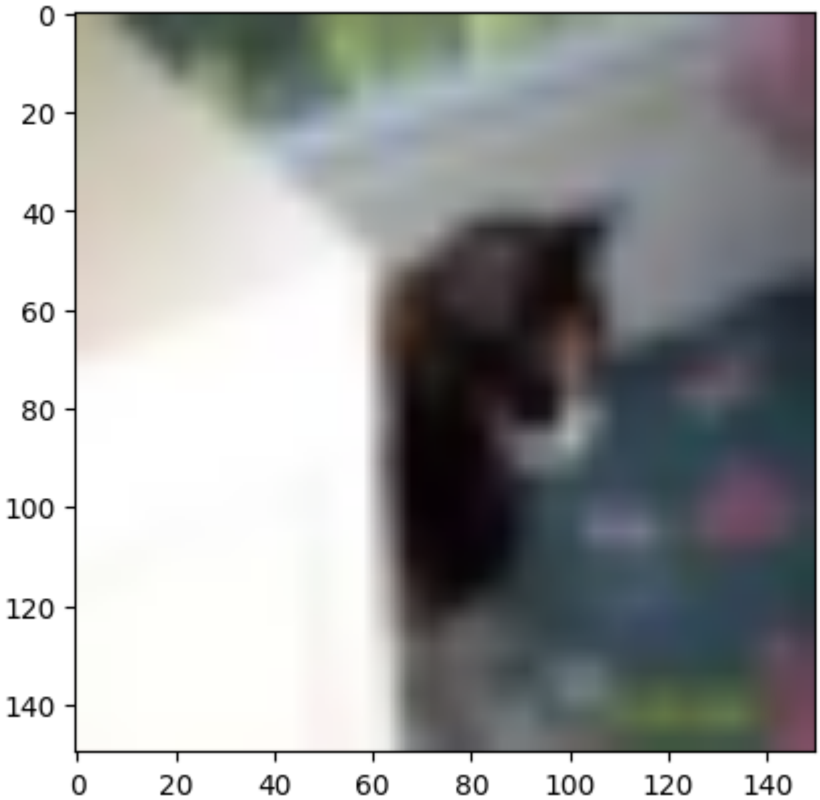
In [ ]:
```
import matplotlib.pyplot as plt
import numpy as np
```

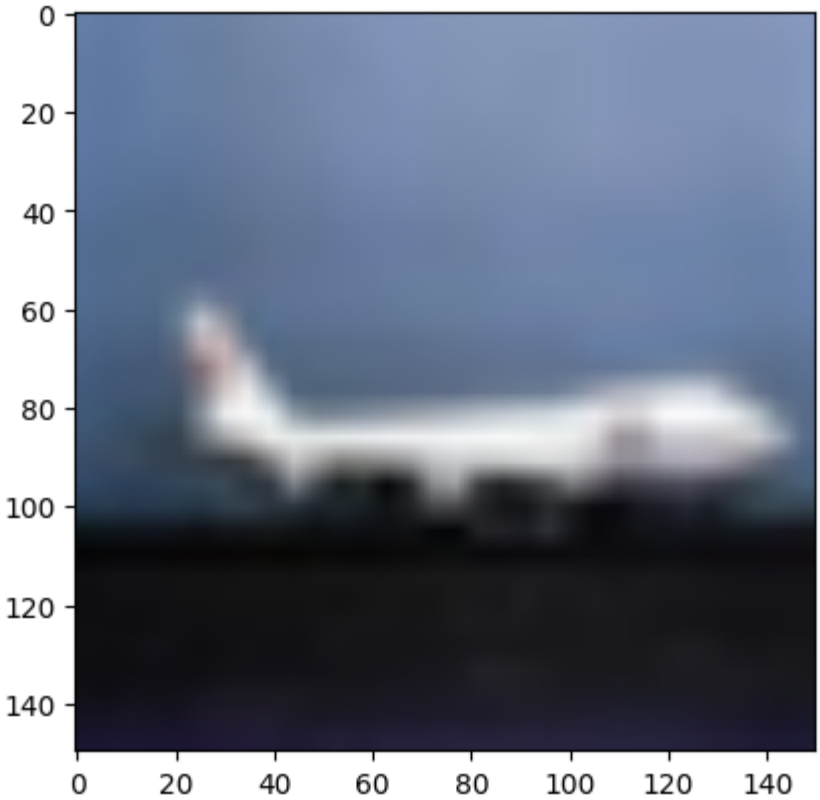
```
#Plot some Augmented images
for images, labels in train_dataset.take(1):
    plt.figure(figsize=(10, 10))
    first_image = images[0]
    for i in range(4):
        ax = plt.subplot(2, 2, i + 1)
        augmented_image = data_augmentation(tf.expand_dims(first_image, 0
        plt.imshow(augmented_image[0] / 255)
        plt.axis('off')
```

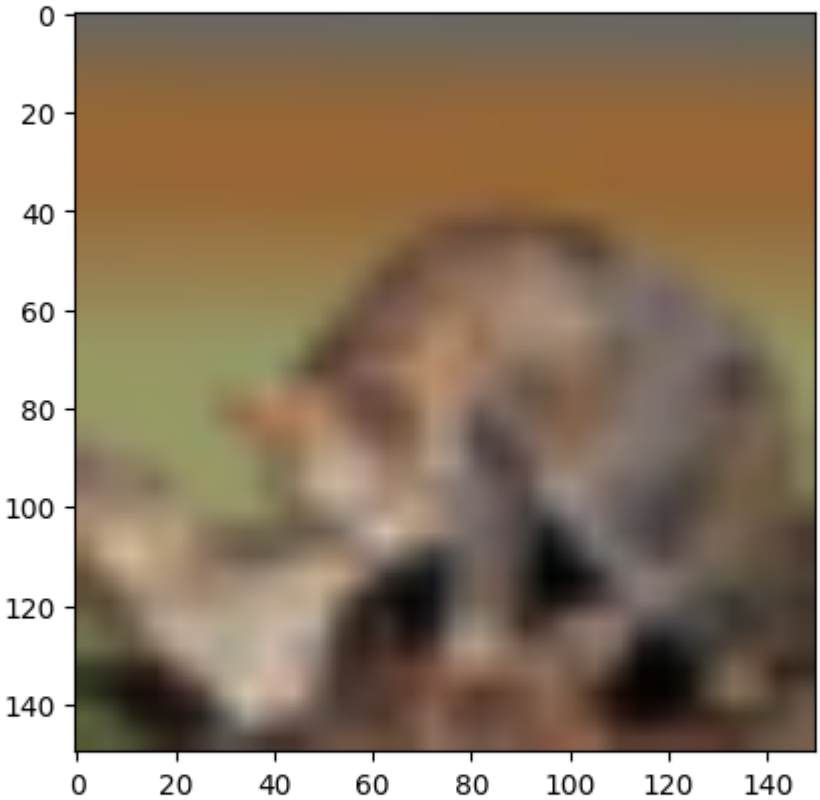
In [ ]:

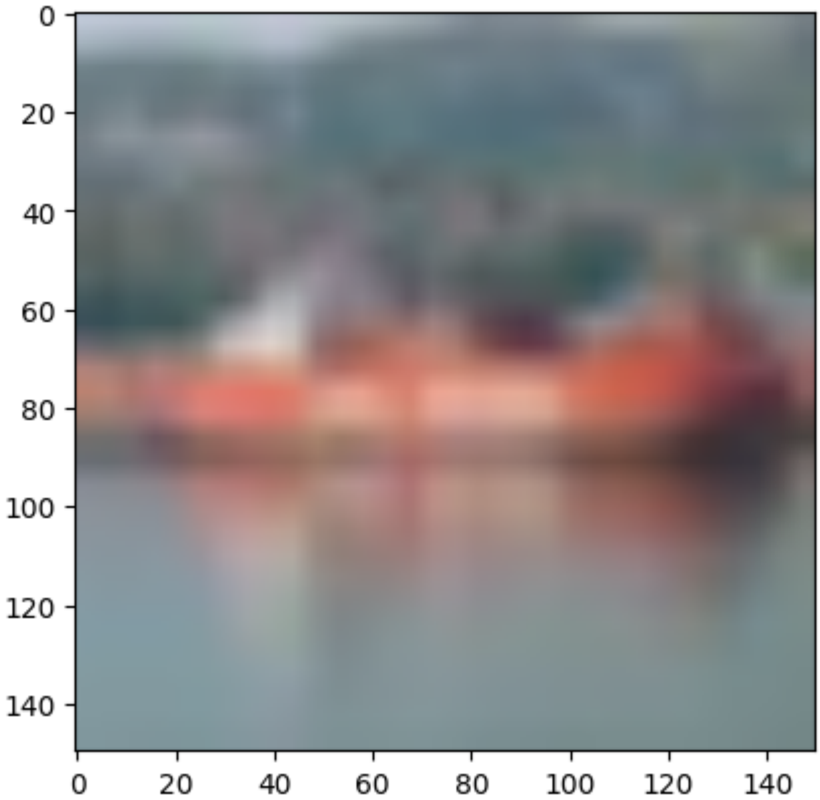
```
import matplotlib.pyplot as plt

for data, _ in train_dataset.take(1):
    for i in range(9):
        plt.imshow(data[i].numpy().astype('uint8'))
        plt.show()
    break
```

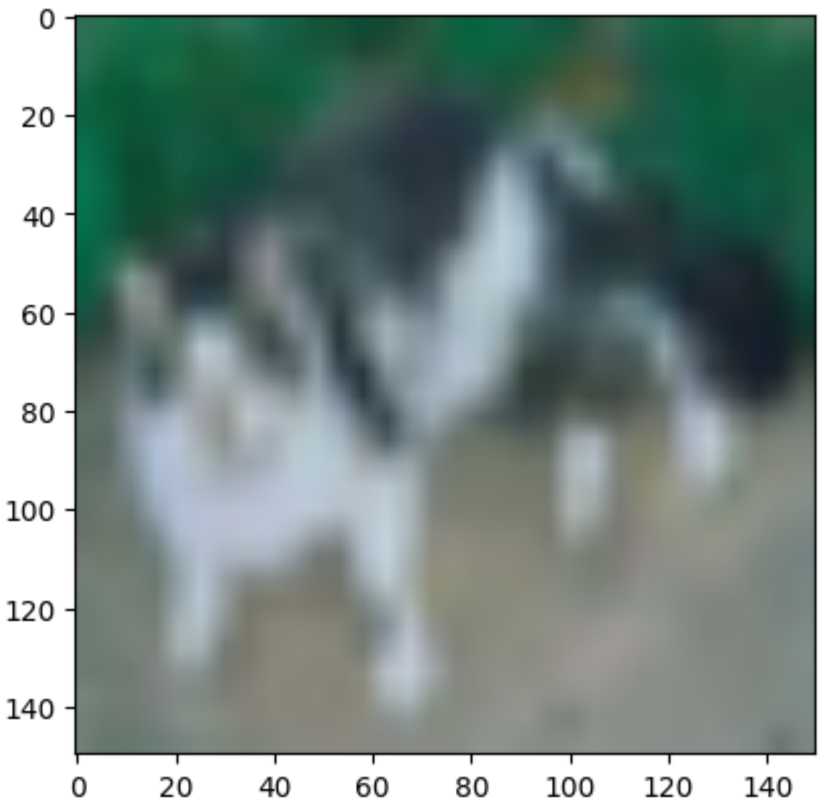

0 20 40 60 80 100 120 140
0 20 40 60 80 100 120 140

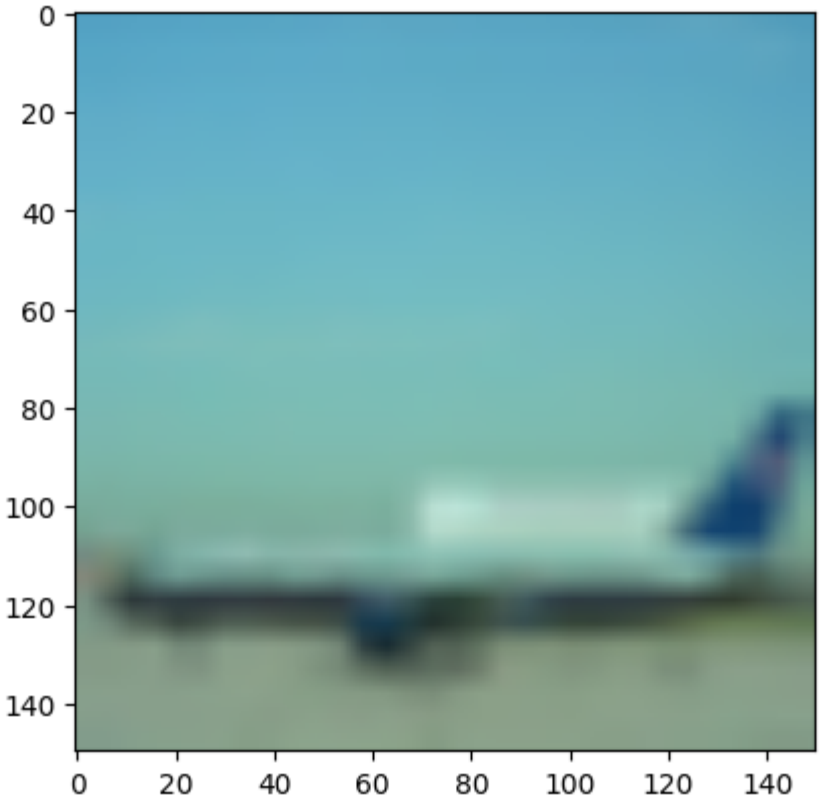

0 20 40 60 80 100 120 140
0 20 40 60 80 100 120 140

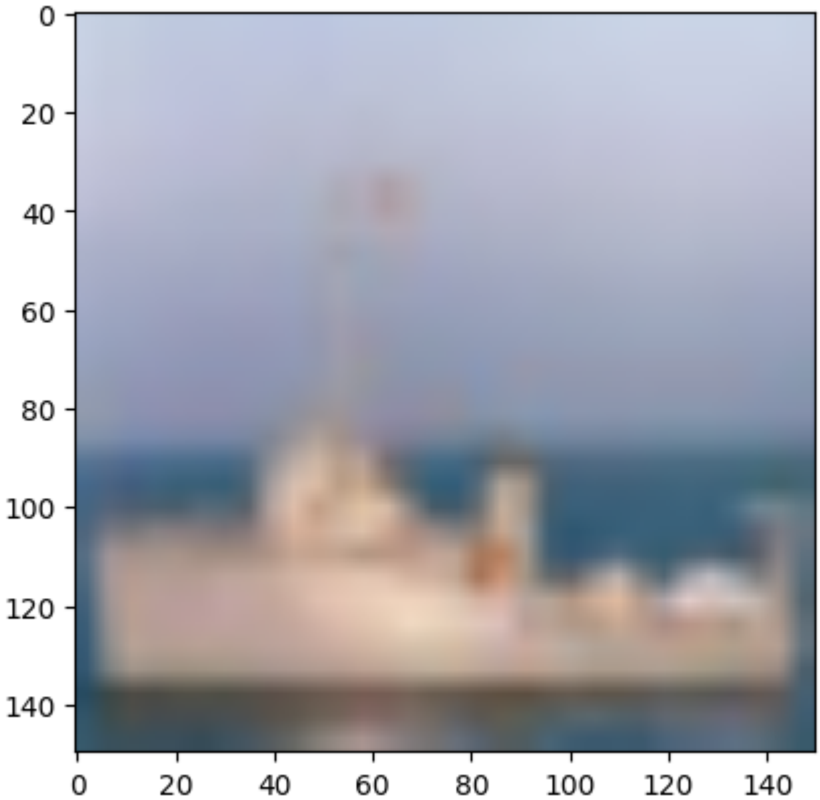

0
20
40
60
80
100
120
140
0 20 40 60 80 100 120 140

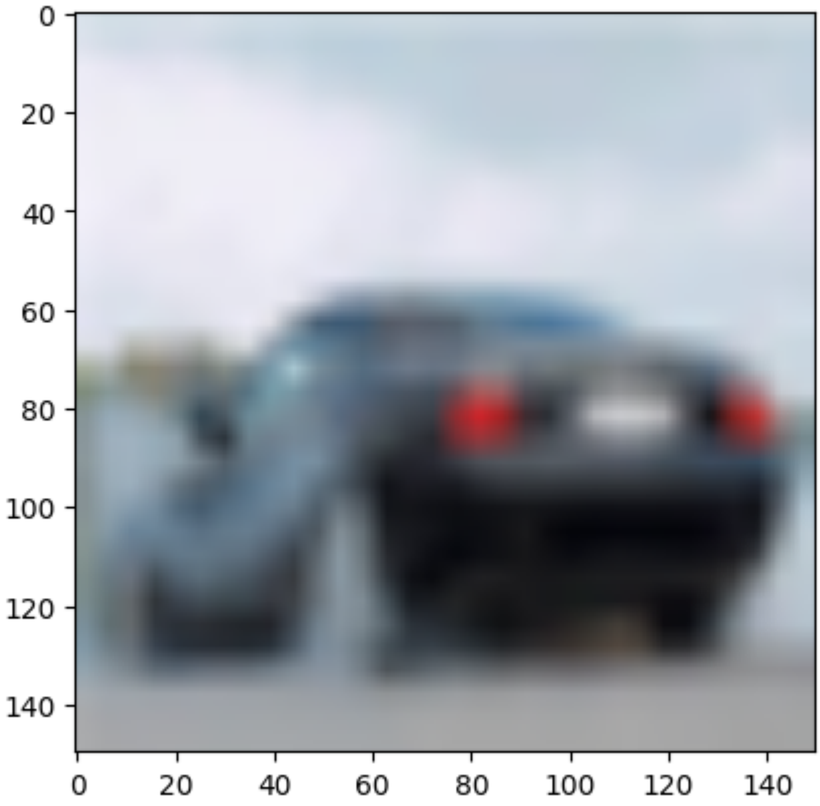

0
20
40
60
80
100
120
140
0 20 40 60 80 100 120 140


0
20
40
60
80
100
120
140
0
20
40
60
80
100
120
140


0
20
40
60
80
100
120
140
0
20
40
60
80
100
120
140


0
20
40
60
80
100
120
140
0 20 40 60 80 100 120 140


0
20
40
60
80
100
120
140
0 20 40 60 80 100 120 140



MODEL ARCHITECTURE

Build a Convolutional Neural Network (CNN) model.

Transfer Learning Model

Use the VGG16 model as a base model.
Freeze the convolutional base and train the densely connected classifier.

Data augmentation before processing the images to the model.

Use two dense layers with 256 neurons and a in-between dropout layer with a rate of 0.5.

In [ ]:
```
from tensorflow import keras
from keras.applications.vgg16 import VGG16
from keras import layers

base_model = VGG16(include_top=False, weights='imagenet')
base_model.trainable = False  # Freeze the base model


inputs = keras.Input(shape=(IMG_SIZE, IMG_SIZE, 3))
x = data_augmentation(inputs)
x = keras.applications.vgg16.preprocess_input(x)
x = base_model(x)
```

```
x = layers.Flatten()(x)
x = layers.Dense(256, activation="relu")(x)
x = layers.Dropout(0.5)(x)
x = layers.Dense(256, activation="relu")(x)
x = layers.Dropout(0.5)(x)

outputs = layers.Dense(10, activation="softmax")(x)  # Softmax for multi-

model = keras.Model(inputs=inputs, outputs=outputs)
model.summary()
```

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 input_5 (InputLayer)        [(None, 150, 150, 3)]     0         
                                                                 
 sequential_1 (Sequential)   (None, 150, 150, 3)       0         
                                                                 
 tf.__operators__.getitem_1  (None, 150, 150, 3)       0         
  (SlicingOpLambda)                                              
                                                                 
 tf.nn.bias_add_1 (TFOpLamb  (None, 150, 150, 3)       0         
 da)                                                             
                                                                 
 vgg16 (Functional)          (None, None, None, 512)   14714688  
                                                                 
 flatten_1 (Flatten)         (None, 8192)              0         
                                                                 
 dense_3 (Dense)             (None, 256)               2097408   
                                                                 
 dropout_2 (Dropout)         (None, 256)               0         
                                                                 
 dense_4 (Dense)             (None, 256)               65792     
                                                                 
 dropout_3 (Dropout)         (None, 256)               0         
                                                                 
 dense_5 (Dense)             (None, 10)                2570      
                                                                 
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 input_5 (InputLayer)        [(None, 150, 150, 3)]     0         
                                                                 
 sequential_1 (Sequential)   (None, 150, 150, 3)       0         
                                                                 
 tf.__operators__.getitem_1  (None, 150, 150, 3)       0         
  (SlicingOpLambda)                                              
                                                                 
 tf.nn.bias_add_1 (TFOpLamb  (None, 150, 150, 3)       0         
 da)                                                             
                                                                 
 vgg16 (Functional)          (None, None, None, 512)   14714688  
                                                                 
 flatten_1 (Flatten)         (None, 8192)              0         
                                                                 
 dense_3 (Dense)             (None, 256)               2097408   
                                                                 
 dropout_2 (Dropout)         (None, 256)               0         
                                                                 
 dense_4 (Dense)             (None, 256)               65792     
                                                                 
 dropout_3 (Dropout)         (None, 256)               0         
                                                                 
 dense_5 (Dense)             (None, 10)                2570      
                                                                 
=================================================================
Total params: 16880458 (64.39 MB)
Trainable params: 2165770 (8.26 MB)
Non-trainable params: 14714688 (56.13 MB)
_________________________________________________________________
```

Compile Model

Loss function:

We use the *Categorical Crossentropy* loss function because it is a `multi-class classification` problem.

Optimizer: Adam

We use the *Adam* optimizer because it is one of the best and most popular optimizers.

In [ ]:
```
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['acc'])
```

Train Model

Train the model with Early stopping, Model checkpoint, and Learning rate reduction callbacks.

In [ ]:
```
from keras.callbacks import EarlyStopping, ModelCheckpoint, ReduceLROnPla

learning_rate_reduction = ReduceLROnPlateau(
    monitor='val_acc',
    patience=3,
    verbose=1,
    factor=0.4,
    min_lr=1e-6)

early_stop = EarlyStopping(monitor='val_acc',
                           patience=8,
                           restore_best_weights=True)
model_checkpoint = ModelCheckpoint('models/T01/checkpoints/T01-DA-cp.h5',

history = model.fit(
    train_dataset,
    epochs=50,
    validation_data=validation_dataset,
    callbacks=[early_stop, model_checkpoint, learning_rate_reduction])
```

```
Epoch 1/50
313/313 [==============================] - ETA: 0s - loss: 2.6981 - acc:
0.4071
```

```
/home/pws/miniconda3/envs/tensorflow/lib/python3.11/site-packages/keras/sr
c/engine/training.py:3103: UserWarning: You are saving your model as an HD
F5 file via `model.save()`. This file format is considered legacy. We reco
mmend using instead the native Keras format, e.g. `model.save('my_model.ke
ras')`.
  saving_api.save_model(
```

```
313/313 [==============================] - 57s 177ms/step - loss: 2.6981 -
acc: 0.4071 - val_loss: 1.0745 - val_acc: 0.6791 - lr: 0.0010
Epoch 2/50
Epoch 2/50
313/313 [==============================] - 55s 177ms/step - loss: 1.5151 -
acc: 0.5776 - val_loss: 0.8892 - val_acc: 0.7617 - lr: 0.0010
Epoch 3/50
313/313 [==============================] - 53s 169ms/step - loss: 1.3404 -
acc: 0.6067 - val_loss: 0.8123 - val_acc: 0.7691 - lr: 0.0010
Epoch 4/50
313/313 [==============================] - 53s 169ms/step - loss: 1.2438 -
acc: 0.6428 - val_loss: 0.7669 - val_acc: 0.7866 - lr: 0.0010
Epoch 5/50
313/313 [==============================] - 53s 170ms/step - loss: 1.1292 -
acc: 0.6721 - val_loss: 0.7313 - val_acc: 0.7917 - lr: 0.0010
Epoch 6/50
313/313 [==============================] - 52s 167ms/step - loss: 1.0273 -
acc: 0.6853 - val_loss: 0.6603 - val_acc: 0.8050 - lr: 0.0010
Epoch 7/50
313/313 [==============================] - 52s 167ms/step - loss: 0.9732 -
acc: 0.7094 - val_loss: 0.6268 - val_acc: 0.8115 - lr: 0.0010
Epoch 8/50
313/313 [==============================] - 52s 167ms/step - loss: 0.8859 -
acc: 0.7272 - val_loss: 0.6241 - val_acc: 0.8113 - lr: 0.0010
Epoch 9/50
313/313 [==============================] - 52s 167ms/step - loss: 0.8311 -
acc: 0.7493 - val_loss: 0.5691 - val_acc: 0.8303 - lr: 0.0010
Epoch 10/50
313/313 [==============================] - 52s 166ms/step - loss: 0.7787 -
acc: 0.7570 - val_loss: 0.5724 - val_acc: 0.8255 - lr: 0.0010
Epoch 11/50
313/313 [==============================] - 53s 169ms/step - loss: 0.7659 -
acc: 0.7566 - val_loss: 0.5308 - val_acc: 0.8362 - lr: 0.0010
Epoch 12/50
313/313 [==============================] - 54s 171ms/step - loss: 0.7219 -
acc: 0.7757 - val_loss: 0.5357 - val_acc: 0.8373 - lr: 0.0010
Epoch 13/50
313/313 [==============================] - 55s 175ms/step - loss: 0.7197 -
acc: 0.7723 - val_loss: 0.5222 - val_acc: 0.8393 - lr: 0.0010
Epoch 14/50
313/313 [==============================] - 55s 175ms/step - loss: 0.6872 -
acc: 0.7885 - val_loss: 0.5169 - val_acc: 0.8463 - lr: 0.0010
Epoch 15/50
313/313 [==============================] - 54s 174ms/step - loss: 0.6878 -
acc: 0.7906 - val_loss: 0.5759 - val_acc: 0.8322 - lr: 0.0010
Epoch 16/50
313/313 [==============================] - 55s 175ms/step - loss: 0.6585 -
acc: 0.7965 - val_loss: 0.5346 - val_acc: 0.8425 - lr: 0.0010
Epoch 17/50
313/313 [==============================] - ETA: 0s - loss: 0.6559 - acc:
0.8011
Epoch 17: ReduceLROnPlateau reducing learning rate to 0.000400000018998980
5.
313/313 [==============================] - 54s 174ms/step - loss: 0.6559 -
acc: 0.8011 - val_loss: 0.5280 - val_acc: 0.8399 - lr: 0.0010
Epoch 18/50
313/313 [==============================] - 55s 175ms/step - loss: 0.5827 -
acc: 0.8172 - val_loss: 0.4923 - val_acc: 0.8531 - lr: 4.0000e-04
Epoch 19/50
313/313 [==============================] - 54s 174ms/step - loss: 0.5292 -
```

```
acc: 0.8283 - val_loss: 0.4893 - val_acc: 0.8537 - lr: 4.0000e-04
Epoch 20/50
313/313 [==============================] - 54s 173ms/step - loss: 0.5384 -
acc: 0.8275 - val_loss: 0.4746 - val_acc: 0.8543 - lr: 4.0000e-04
Epoch 21/50
313/313 [==============================] - 54s 174ms/step - loss: 0.5044 -
acc: 0.8377 - val_loss: 0.4802 - val_acc: 0.8526 - lr: 4.0000e-04
Epoch 22/50
313/313 [==============================] - 55s 176ms/step - loss: 0.4972 -
acc: 0.8363 - val_loss: 0.4868 - val_acc: 0.8523 - lr: 4.0000e-04
Epoch 23/50
313/313 [==============================] - 55s 176ms/step - loss: 0.4787 -
acc: 0.8432 - val_loss: 0.4841 - val_acc: 0.8545 - lr: 4.0000e-04
Epoch 24/50
313/313 [==============================] - 56s 178ms/step - loss: 0.4606 -
acc: 0.8497 - val_loss: 0.4708 - val_acc: 0.8567 - lr: 4.0000e-04
Epoch 25/50
313/313 [==============================] - 56s 179ms/step - loss: 0.4757 -
acc: 0.8498 - val_loss: 0.4629 - val_acc: 0.8602 - lr: 4.0000e-04
Epoch 26/50
313/313 [==============================] - 55s 176ms/step - loss: 0.4619 -
acc: 0.8522 - val_loss: 0.4729 - val_acc: 0.8572 - lr: 4.0000e-04
Epoch 27/50
313/313 [==============================] - 55s 176ms/step - loss: 0.4767 -
acc: 0.8495 - val_loss: 0.4731 - val_acc: 0.8534 - lr: 4.0000e-04
Epoch 28/50
313/313 [==============================] - ETA: 0s - loss: 0.4416 - acc:
0.8557
Epoch 28: ReduceLROnPlateau reducing learning rate to 0.000160000007599592
22.
313/313 [==============================] - 54s 174ms/step - loss: 0.4416 -
acc: 0.8557 - val_loss: 0.4616 - val_acc: 0.8587 - lr: 4.0000e-04
Epoch 29/50
313/313 [==============================] - 55s 175ms/step - loss: 0.4336 -
acc: 0.8566 - val_loss: 0.4572 - val_acc: 0.8590 - lr: 1.6000e-04
Epoch 30/50
313/313 [==============================] - 55s 176ms/step - loss: 0.4275 -
acc: 0.8613 - val_loss: 0.4562 - val_acc: 0.8589 - lr: 1.6000e-04
Epoch 31/50
313/313 [==============================] - ETA: 0s - loss: 0.4054 - acc:
0.8664
Epoch 31: ReduceLROnPlateau reducing learning rate to 6.40000042039901e-0
5.
313/313 [==============================] - 56s 178ms/step - loss: 0.4054 -
acc: 0.8664 - val_loss: 0.4561 - val_acc: 0.8579 - lr: 1.6000e-04
Epoch 32/50
313/313 [==============================] - 55s 176ms/step - loss: 0.3873 -
acc: 0.8725 - val_loss: 0.4518 - val_acc: 0.8593 - lr: 6.4000e-05
Epoch 33/50
313/313 [==============================] - 54s 174ms/step - loss: 0.3928 -
acc: 0.8714 - val_loss: 0.4520 - val_acc: 0.8609 - lr: 6.4000e-05
Epoch 34/50
313/313 [==============================] - 55s 175ms/step - loss: 0.3820 -
acc: 0.8736 - val_loss: 0.4516 - val_acc: 0.8598 - lr: 6.4000e-05
Epoch 35/50
313/313 [==============================] - 54s 174ms/step - loss: 0.3774 -
acc: 0.8736 - val_loss: 0.4491 - val_acc: 0.8603 - lr: 6.4000e-05
Epoch 36/50
313/313 [==============================] - 55s 175ms/step - loss: 0.3745 -
acc: 0.8753 - val_loss: 0.4477 - val_acc: 0.8612 - lr: 6.4000e-05
```

```
Epoch 37/50
313/313 [==============================] - 54s 174ms/step - loss: 0.3749 -
acc: 0.8784 - val_loss: 0.4501 - val_acc: 0.8598 - lr: 6.4000e-05
Epoch 38/50
313/313 [==============================] - 54s 174ms/step - loss: 0.3760 -
acc: 0.8762 - val_loss: 0.4480 - val_acc: 0.8603 - lr: 6.4000e-05
Epoch 39/50
313/313 [==============================] - ETA: 0s - loss: 0.3626 - acc:
0.8803
Epoch 39: ReduceLROnPlateau reducing learning rate to 2.560000284574926e-0
5.
313/313 [==============================] - 55s 175ms/step - loss: 0.3626 -
acc: 0.8803 - val_loss: 0.4473 - val_acc: 0.8602 - lr: 6.4000e-05
Epoch 40/50
313/313 [==============================] - 55s 175ms/step - loss: 0.3575 -
acc: 0.8765 - val_loss: 0.4459 - val_acc: 0.8617 - lr: 2.5600e-05
Epoch 41/50
313/313 [==============================] - 54s 174ms/step - loss: 0.3672 -
acc: 0.8768 - val_loss: 0.4460 - val_acc: 0.8606 - lr: 2.5600e-05
Epoch 42/50
313/313 [==============================] - 55s 175ms/step - loss: 0.3639 -
acc: 0.8814 - val_loss: 0.4457 - val_acc: 0.8610 - lr: 2.5600e-05
Epoch 43/50
313/313 [==============================] - ETA: 0s - loss: 0.3672 - acc:
0.8825
Epoch 43: ReduceLROnPlateau reducing learning rate to 1.0240000847261399
e-05.
313/313 [==============================] - 55s 175ms/step - loss: 0.3672 -
acc: 0.8825 - val_loss: 0.4442 - val_acc: 0.8614 - lr: 2.5600e-05
Epoch 44/50
313/313 [==============================] - 55s 175ms/step - loss: 0.3639 -
acc: 0.8827 - val_loss: 0.4437 - val_acc: 0.8618 - lr: 1.0240e-05
Epoch 45/50
313/313 [==============================] - 54s 173ms/step - loss: 0.3661 -
acc: 0.8756 - val_loss: 0.4441 - val_acc: 0.8617 - lr: 1.0240e-05
Epoch 46/50
313/313 [==============================] - 60s 191ms/step - loss: 0.3566 -
acc: 0.8847 - val_loss: 0.4446 - val_acc: 0.8615 - lr: 1.0240e-05
Epoch 47/50
313/313 [==============================] - ETA: 0s - loss: 0.3644 - acc:
0.8796
Epoch 47: ReduceLROnPlateau reducing learning rate to 4.09600033890456e-0
6.
313/313 [==============================] - 59s 188ms/step - loss: 0.3644 -
acc: 0.8796 - val_loss: 0.4451 - val_acc: 0.8607 - lr: 1.0240e-05
Epoch 48/50
313/313 [==============================] - 59s 190ms/step - loss: 0.3515 -
acc: 0.8854 - val_loss: 0.4449 - val_acc: 0.8605 - lr: 4.0960e-06
Epoch 49/50
313/313 [==============================] - 59s 189ms/step - loss: 0.3617 -
acc: 0.8823 - val_loss: 0.4451 - val_acc: 0.8604 - lr: 4.0960e-06
Epoch 50/50
313/313 [==============================] - ETA: 0s - loss: 0.3562 - acc:
0.8784
Epoch 50: ReduceLROnPlateau reducing learning rate to 1.6384001355618238
e-06.
313/313 [==============================] - 59s 189ms/step - loss: 0.3562 -
acc: 0.8784 - val_loss: 0.4452 - val_acc: 0.8609 - lr: 4.0960e-06
```

Save Model

In [ ]:
```
keras.models.save_model(model, 'models/T01/T01-DA-model.h5')
```

```
/tmp/ipykernel_8035/1849881407.py:1: UserWarning: You are saving your mode
l as an HDF5 file via `model.save()`. This file format is considered legac
y. We recommend using instead the native Keras format, e.g. `model.save('m
y_model.keras')`.
  keras.models.save_model(model, 'models/T01/T01-DA-model.h5')
```

In [ ]:
```
keras.models.load_model('models/T01/T01-DA-model.h5')
```

Out[ ]: <keras.src.engine.functional.Functional at 0x72a4b2d92ad0>

EVALUATION

Evaluate the model on the validation dataset.

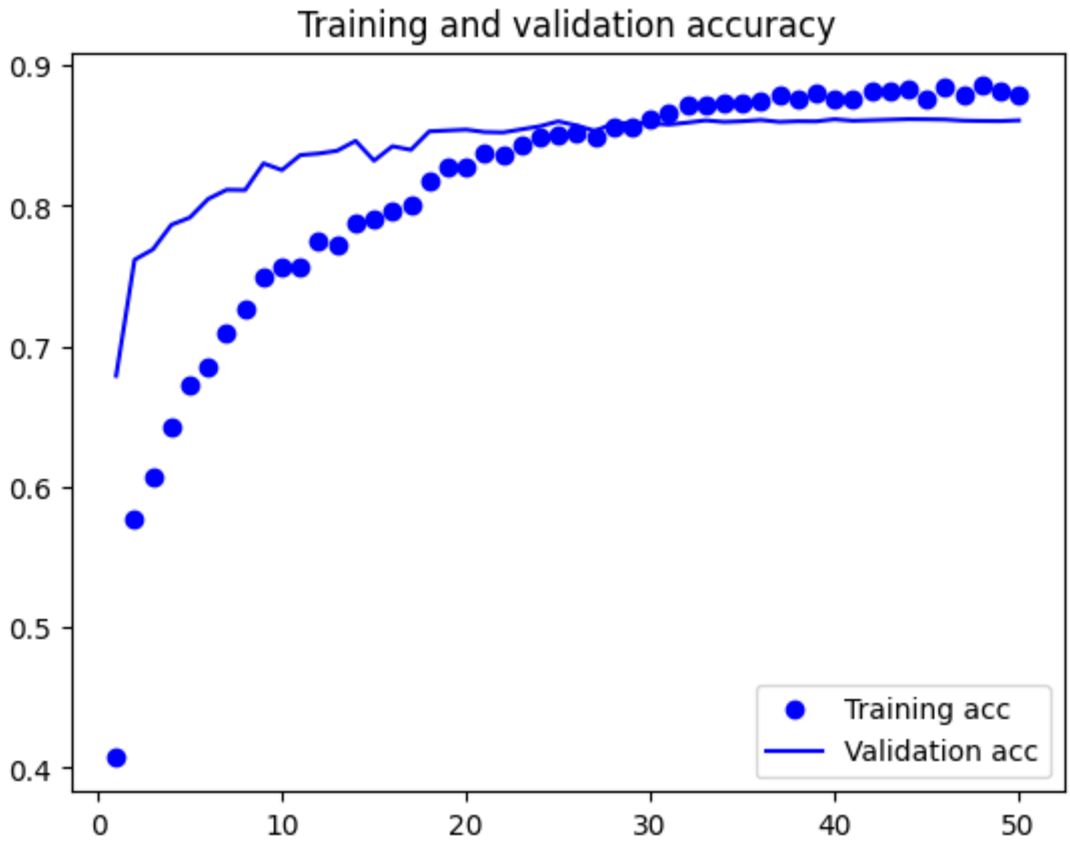
In [ ]:
```
val_loss, val_acc = model.evaluate(validation_dataset)
print('val_acc:', val_acc)
```

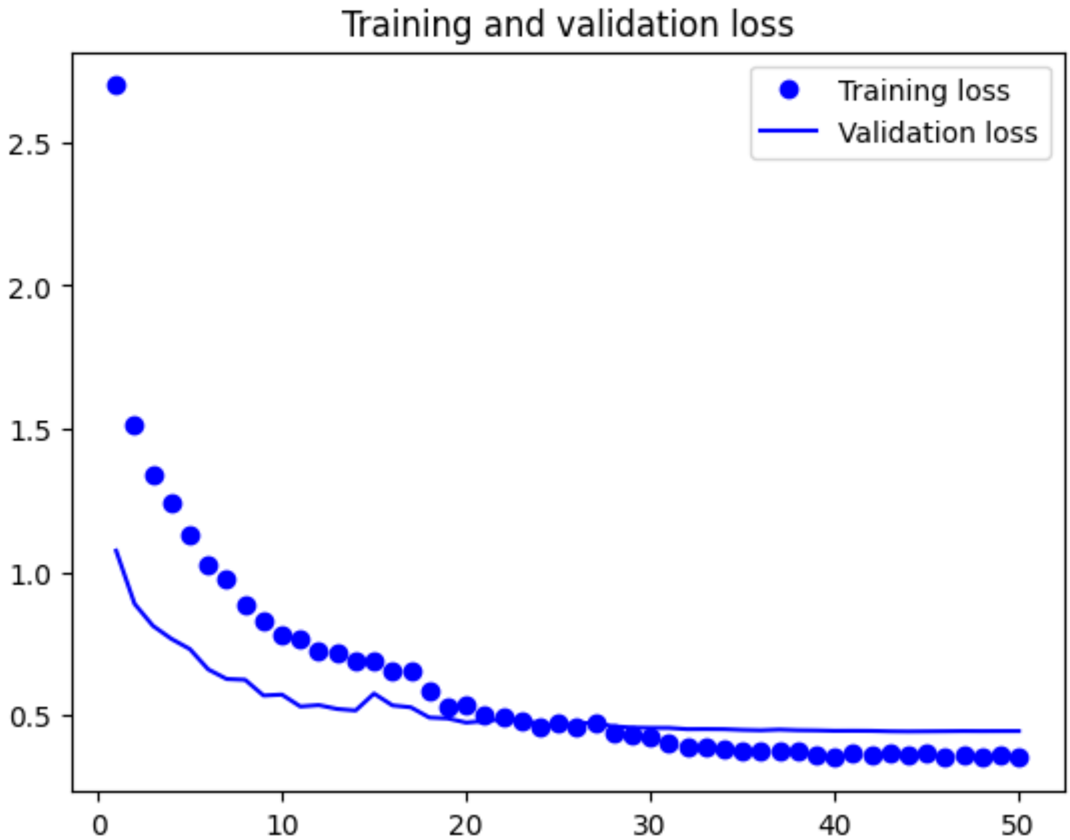
```
313/313 [==============================] - 29s 93ms/step - loss: 0.4452 -
acc: 0.8609
val_acc: 0.8608999848365784
```

Training and Validation Curves

Plot the training and validation accuracy and loss curves.

In [ ]:
```
import matplotlib.pyplot as plt

# Extract the history from the training process
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)

# Plot the training and validation accuracy
plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

# Plot the training and validation loss
plt.figure()
plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()
plt.show()
```





Confusion Matrix

In [ ]:
```
from sklearn.metrics import confusion_matrix
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
y_true = []
y_pred = []

for images, labels in validation_dataset:
    y_true.extend(np.argmax(labels, axis=1))
    y_pred.extend(np.argmax(model.predict(images), axis=1))

y_true = np.array(y_true)
y_pred = np.array(y_pred)

cm = confusion_matrix(y_true, y_pred)

plt.figure(figsize=(10, 8))
sns.heatmap(cm, annot=True, cmap='Blues', fmt='g')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```

```
1/1 [==============================] - 0s 162ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 19ms/step
```

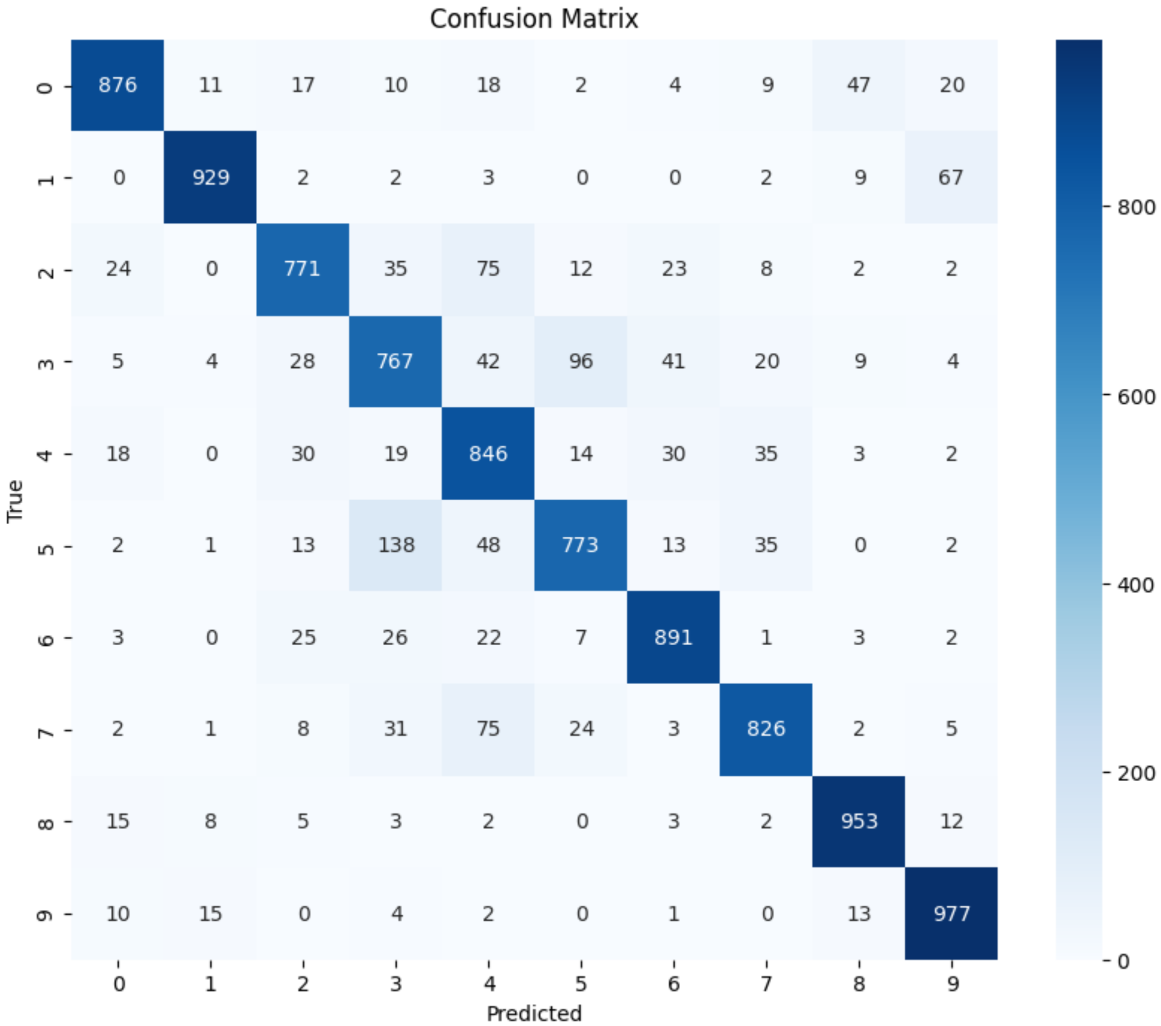
```
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 22ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
```

```
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
```

```
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 21ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 24ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
```

```
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 20ms/step
1/1 [==============================] - 0s 30ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 19ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 18ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
```

```
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 15ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 132ms/step
```



In [ ]:
```
print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[876  11  17  10  18   2   4   9  47  20]
 [  0 929   2   2   3   0   0   2   9  67]
 [ 24   0 771  35  75  12  23   8   2   2]
 [  5   4  28 767  42  96  41  20   9   4]
 [ 18   0  30  19 846  14  30  35   3   2]
 [  2   1  13 138  48 773  13  35   0   2]
 [  3   0  25  26  22   7 891   1   3   2]
 [  2   1   8  31  75  24   3 826   2   5]
 [ 15   8   5   3   2   0   3   2 953  12]
 [ 10  15   0   4   2   0   1   0  13 977]]
```

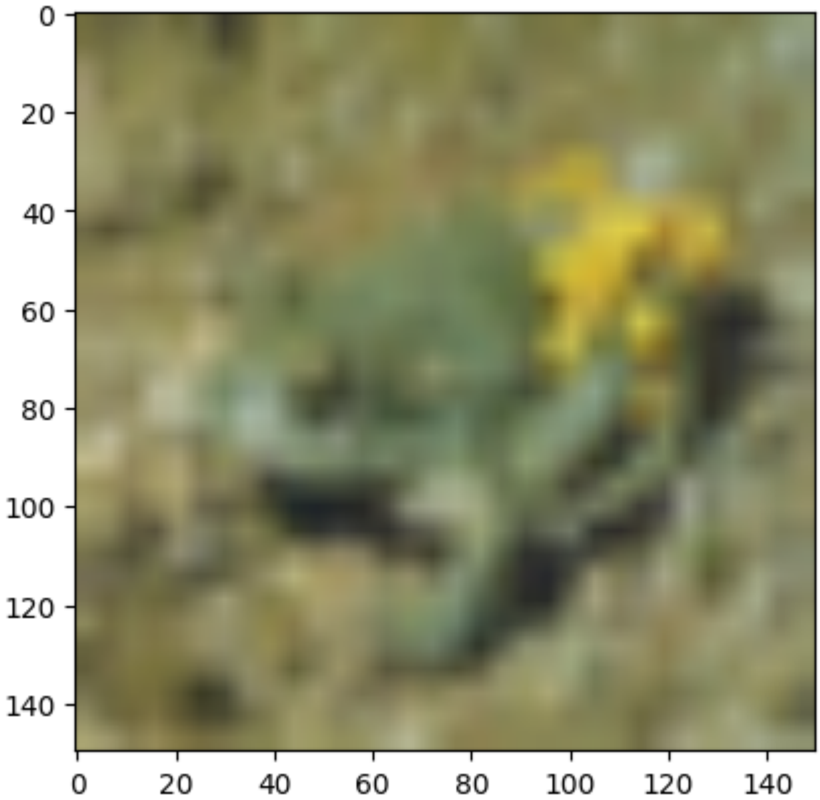
In [ ]:
```
from sklearn.metrics import classification_report

report = classification_report(y_true, y_pred, target_names=class_names)
print(report)
```

```
              precision    recall  f1-score   support

    airplane       0.92      0.86      0.89      1014
  automobile       0.96      0.92      0.94      1014
        bird       0.86      0.81      0.83       952
         cat       0.74      0.75      0.75      1016
        deer       0.75      0.85      0.79       997
         dog       0.83      0.75      0.79      1025
        frog       0.88      0.91      0.90       980
       horse       0.88      0.85      0.86       977
        ship       0.92      0.95      0.93      1003
       truck       0.89      0.96      0.92      1022

    accuracy                           0.86     10000
   macro avg       0.86      0.86      0.86     10000
weighted avg       0.86      0.86      0.86     10000
```

Predictions

Predict and visualize the results for a sample image.

In [ ]:
```
import tensorflow as tf
import matplotlib.pyplot as plt
from keras.preprocessing import image

# Load an image
img_path = train_dirs[0] + '/006_frog/alytes_obstetricans_s_000179.png'
# img_path2 = train_dirs[0] + '/000_airplane/airbus_s_000012.png'
img = tf.keras.preprocessing.image.load_img(img_path, target_size=(150, 1

# Preprocess the image
img_array = image.img_to_array(img)
img_array = tf.expand_dims(img_array, 0)

plt.imshow(img)
plt.show()

print(img_array.shape)
result = model.predict(img_array)
print("Result: ", result.round())
print("Predicted class: ", class_names[np.argmax(result)])
print("True class: ", img_path.split('/')[-2].split('_')[-1])
```



```
(1, 150, 150, 3)
1/1 [==============================] - 0s 29ms/step
Result:  [[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
Predicted class:  frog
True class:  frog
```