In [36]:	<pre>%pip install os %pip install matplotlib</pre>
	<pre>%pip install tensorflow %pip install keras %pip install numpy %pip install sklearn Requirement already satisfied: opencv-python in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (4.8.1.78) Requirement already satisfied: opencv-contrib-python in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (4.8.1.78) Requirement already satisfied: numpy>=1.17.3 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from opencv-python) (1.26.2) Note: you may need to restart the kernel to use updated packages.</pre>
	WARNING: Ignoring invalid distribution -rotobuf (c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages) WARNING: You are using pip version 22.0.4; however, version 23.3.1 is available. You should consider upgrading via the 'c:\Users\sebastian\Desktop\school\ai\DL_env\Scripts\python.exe -m pip installupgrade pip' command.
	Note: you may need to restart the kernel to use updated packages. WARNING: Ignoring invalid distribution -rotobuf (c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages) WARNING: Ignoring invalid distribution -rotobuf (c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages) ERROR: Could not find a version that satisfies the requirement os (from versions: none) ERROR: No matching distribution found for os WARNING: Ignoring invalid distribution -rotobuf (c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages)
	WARNING: You are using pip version 22.0.4; however, version 23.3.1 is available. You should consider upgrading via the 'c:\Users\sebastian\Desktop\school\ai\DL_env\Scripts\python.exe -m pip installupgrade pip' command. Requirement already satisfied: matplotlib in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (3.8.2) Requirement already satisfied: numpy<2,>=1.21 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from matplotlib) (1.26.2) Requirement already satisfied: packaging>=20.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from matplotlib) (23.2) Requirement already satisfied: importlib-resources>=3.2.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from matplotlib) (6.1.1) Requirement already satisfied: fonttools>=4.22.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from matplotlib) (4.45.0) Requirement already satisfied: pillow>=8 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from matplotlib) (10.1.0)
	Requirement already satisfied: pyparsing>=2.3.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from matplotlib) (3.1.1) Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from matplotlib) (1.4.5) Requirement already satisfied: python-dateutil>=2.7 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from matplotlib) (2.8.2) Requirement already satisfied: contourpy>=1.0.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from matplotlib) (1.2.0) Requirement already satisfied: cycler>=0.10 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from matplotlib) (0.12.1) Requirement already satisfied: zipp>=3.1.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from importlib-resources>=3.2.0->matplotlib) (3.17.0) Requirement already satisfied: six>=1.5 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0) Note: you may need to restart the kernel to use updated packages.
	WARNING: Ignoring invalid distribution -rotobuf (c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages) WARNING: You are using pip version 22.0.4; however, version 23.3.1 is available. You should consider upgrading via the 'c:\Users\sebastian\Desktop\school\ai\DL_env\Scripts\python.exe -m pip installupgrade pip' command.
	Requirement already satisfied: keras in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (2.15.0) Note: you may need to restart the kernel to use updated packages. WARNING: Ignoring invalid distribution -rotobuf (c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages)
	WARNING: You are using pip version 22.0.4; however, version 23.3.1 is available. You should consider upgrading via the 'c:\Users\sebastian\Desktop\school\ai\DL_env\Scripts\python.exe -m pip installupgrade pip' command. Requirement already satisfied: streamlit in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (1.29.0) Requirement already satisfied: toml<2,>=0.10.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (0.10.2) Requirement already satisfied: importlib-metadata<7,>=1.4 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (6.8.0) Requirement already satisfied: altair<6,>=4.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (5.2.0) Requirement already satisfied: tornado<7,>=6.0.3 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (6.3.3) Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (3.1.40) Paguirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site packages (from streamlit) (3.1.40)
	Requirement already satisfied: pandas<3,>=1.3.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (2.1.3) Requirement already satisfied: packaging<24,>=16.8 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (23.2) Requirement already satisfied: requests<3,>=2.27 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (2.31.0) Requirement already satisfied: protobuf<5,>=3.20 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (4.23.4) Requirement already satisfied: tzlocal<6,>=1.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (5.2) Requirement already satisfied: blinker<2,>=1.0.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (1.7.0) Requirement already satisfied: python-dateutil<3,>=2.7.3 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (2.8.2) Requirement already satisfied: pillow<11,>=7.1.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (10.1.0) Requirement already satisfied: pyarrow>=6.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (14.0.1)
	Requirement already satisfied: pyarrow=0.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (8.2.3) Requirement already satisfied: watchdog>=2.1.5 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (8.2.3) Requirement already satisfied: watchdog>=2.1.5 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (0.20.0) Requirement already satisfied: typing-extensions<5,>=4.3.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (4.8.0) Requirement already satisfied: pydeck<1,>=0.8.0b4 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (0.8.1b0) Requirement already satisfied: click<9,>=7.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (8.1.7) Requirement already satisfied: cachetools<6,>=4.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (5.3.2) Requirement already satisfied: rich<14,>=10.14.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (13.7.0)
	Requirement already satisfied: numpy<2,>=1.19.3 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from streamlit) (1.26.2) Requirement already satisfied: jinja2 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from altair<6,>=4.0->streamlit) (3.1.2) Requirement already satisfied: jsonschema>=3.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from altair<6,>=4.0->streamlit) (4.20.0) Requirement already satisfied: toolz in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from altair<6,>=4.0->streamlit) (0.12.0) Requirement already satisfied: colorama in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from click<9,>=7.0->streamlit) (0.4.6) Requirement already satisfied: gitdb<5,>=4.0.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from gitpython!=3.1.19,<4,>=3.0.7->streamlit) (4.0.11) Requirement already satisfied: zipp>=0.5 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from packages (from packages),>=1.3.0->streamlit) (3.17.0) Requirement already satisfied: pytz>=2020.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from packages),>=1.3.0->streamlit) (2023.3.post1)
	Requirement already satisfied: tzdata>=2022.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from pandas<3,>=1.3.0->streamlit) (2023.3) Requirement already satisfied: six>=1.5 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from python-dateutil<3,>=2.7.3->streamlit) (1.16.0) Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from requests<3,>=2.27->streamlit) (3.3.2) Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from requests<3,>=2.27->streamlit) (2.1.0) Requirement already satisfied: idna<4,>=2.5 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from requests<3,>=2.27->streamlit) (3.4) Requirement already satisfied: certifi>=2017.4.17 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from requests<3,>=2.27->streamlit) (2023.11.17) Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from rich<14,>=10.14.0->streamlit) (3.0.0) Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from gitdb<5,>=4.0.1->gitpython!=3.1.19,<4,>=3.0.7->streamlit) (5.
	Requirement already satisfied: smmap<6,>=3.0.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from gitdb<5,>=4.0.1->gitpython!=3.1.19,<4,>=3.0.7->streamlit) (5.0.1) Requirement already satisfied: MarkupSafe>=2.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from jinja2->altair<6,>=4.0->streamlit) (2.1.3) Requirement already satisfied: rpds-py>=0.7.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from jsonschema>=3.0->altair<6,>=4.0->streamlit) (0.13.2) Requirement already satisfied: jsonschema-specifications>=2023.03.6 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from jsonschema>=3.0->altair<6,>=4.0->streamlit) (2023.11.2) Requirement already satisfied: attrs>=22.2.0 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from jsonschema>=3.0->altair<6,>=4.0->streamlit) (23.1.0) Requirement already satisfied: referencing>=0.28.4 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from jsonschema>=3.0->altair<6,>=4.0->streamlit) (0.32.0) Requirement already satisfied: mdurl~=0.1 in c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages (from markdown-it-py>=2.2.0->rich<14,>=10.14.0->streamlit) (0.1.2)
	Note: you may need to restart the kernel to use updated packages. WARNING: Ignoring invalid distribution -rotobuf (c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages) WARNING: You are using pip version 22.0.4; however, version 23.3.1 is available.
	You should consider upgrading via the 'c:\Users\sebastian\Desktop\school\ai\DL_env\Scripts\python.exe -m pip installupgrade pip' command. Collecting protobuf=3.19.0 Downloading protobuf-3.19.0-cp39-cp39-win_amd64.whl (895 kB)
	Note: you may need to restart the kernel to use updated packages. WARNING: Ignoring invalid distribution -rotobuf (c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages) ERROR: Could not install packages due to an OSError: [WinError 5] Access is denied: 'C:\\Users\\sebastian\\Desktop\\school\\ai\\DL_env\\Lib\\site-packages\\google\\~-pb_message.opgo-packages\\packages\packages\\packages\\packages\\packages\\packages\\packages\\packages\\packages\\packages\\pack
	WARNING: Ignoring invalid distribution -rotobuf (c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages) WARNING: Ignoring invalid distribution -rotobuf (c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages) WARNING: Ignoring invalid distribution -rotobuf (c:\users\sebastian\desktop\school\ai\dl_env\lib\site-packages) WARNING: You are using pip version 22.0.4; however, version 23.3.1 is available. You should consider upgrading via the 'c:\Users\sebastian\Desktop\school\ai\DL_env\Scripts\python.exe -m pip installupgrade pip' command. 2. EDA & Prep data
In [1]:	display how many images are in each class import os dir = './' image_dir = os.path.join(dir, 'images')
	<pre>for subfolder in os.listdir(image_dir): subfolder_path = os.path.join(image_dir, subfolder) if os.path.isdir(subfolder_path): image_count = 0 for file in os.listdir(subfolder_path): if file.endswith('.jpg'):</pre>
	<pre>image_count += 1 print(subfolder, image_count) test_set 0 training_set 0 load in the images and split them into train, validation, and test sets</pre>
In [2]:	<pre>from keras.utils import image_dataset_from_directory batch_size = 32 image_size = (64, 64) validation_split = 0.2 # Create the training dataset from the 'train' directory train_ds = image_dataset_from_directory(</pre>
	<pre>directory='images/training_set', labels='inferred', label_mode='categorical', batch_size=batch_size, image_size=image_size, validation_split=validation_split, subset='training', seed=123</pre>
	<pre># Create the validation dataset from the 'train' directory validation_ds = image_dataset_from_directory(directory='images/training_set', labels='inferred', label_mode='categorical', batch_size='ategorical', image_size=batch_size, image_size=batch_size,</pre>
	<pre>image_size=image_size, validation_split=validation_split, subset='validation', seed=123) # Create the testing dataset from the 'test' directory test_ds = image_dataset_from_directory(directory='images'test_set'</pre>
	<pre>directory='images/test_set', labels='inferred', label_mode='categorical', batch_size=batch_size, image_size=image_size) WARNING:tensorflow:From c:\Users\sebastian\Desktop\school\ai\DL_env\lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.</pre>
	Found 500 files belonging to 5 classes. Using 400 files for training. Found 500 files belonging to 5 classes. Using 100 files for validation. Found 313 files belonging to 5 classes.
In [3]:	3. Build the model import necessary libraries import tensorflow as tf from tensorflow import keras from tensorflow.keras import optimizers from tensorflow.keras import layers
	from keras.models import Sequential from keras.layers import Conv2D from keras.layers import MaxPooling2D from keras.layers import Flatten from keras.layers import Dense from keras.layers import Dropout create model architecture
In [24]:	<pre>NUM_CLASSES = 5 IMG_SIZE = 128 HEIGTH_FACTOR = 0.2 WIDTH_FACTOR = 0.2 # Create a sequential model with a list of layers</pre>
	<pre>model = tf.keras.Sequential([# Add a resizing layer to resize the images to a consistent shape layers.Resizing(IMG_SIZE, IMG_SIZE), # Add a rescaling layer to rescale the pixel values to the [0, 1] range layers.Rescaling(1./255), # Add some data augmentation layers to apply random transformations during training layers.RandomFlip("horizontal"), layers.RandomTranslation(HEIGTH_FACTOR,WIDTH_FACTOR),</pre>
	<pre>layers.RandomZoom(0.2), # Add a convolutional layer with 32 filters, a 3x3 kernel, and relu activation layers.Conv2D(32, (3, 3), input_shape = (64, 64, 3), activation="relu"), layers.MaxPooling2D((2, 2)), layers.Dropout(0.2), layers.Conv2D(32, (3, 3), activation="relu"), layers.MaxPooling2D((2, 2)),</pre>
	<pre>layers.Dropout(0.2), layers.Flatten(), # Or, layers.GlobalAveragePooling2D() layers.Dense(128, activation="relu"), layers.Dense(NUM_CLASSES, activation="sigmoid")]) # Compile and train your model as usual model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])</pre>
In [25]:	4. Train the model history = model.fit(train_ds,
	13/13 [====================================
	Epoch 5/20 13/13 [====================================
	Epoch 9/20 13/13 [====================================
	Epoch 13/20 13/13 [====================================
	Epoch 17/20 13/13 [====================================
In [26]:	<pre># Create a figure and a grid of subplots with a single call fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10,5)) # Plot the loss curves on the first subplot ax1.plot(history.history['loss'], label='training loss')</pre>
	<pre>ax1.plot(history.history['val_loss'], label='validation loss') ax1.set_title('Loss curves') ax1.set_xlabel('Epoch') ax1.set_ylabel('Loss') ax1.legend() # Plot the accuracy curves on the second subplot ax2.plot(history.history['accuracy'], label='training accuracy')</pre>
	<pre>ax2.plot(history.history['val_accuracy'], label='validation accuracy') ax2.set_title('Accuracy curves') ax2.set_xlabel('Epoch') ax2.set_ylabel('Accuracy') ax2.legend() # Adjust the spacing between subplots fig.tight_layout()</pre>
	# Show the figure plt.show() Loss curves Accuracy curves training loss validation loss 0.60 - training accuracy validation accuracy
	0.60 - 0.55 - 0.55 - 0.45 -
	0.45 - 0.40 - 0.35 - 0.30 -
	0.35 - 0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5 Epoch
In [27]:	test_loss, test_acc = model.evaluate(test_ds) print('Test accuracy:', test_acc) 10/10 [====================================
In [34]:	<pre>import numpy as np from sklearn.metrics import confusion_matrix from sklearn.metrics import ConfusionMatrixDisplay # Generate predictions for all the test images predictions = model.predict(test_ds)</pre>
	# First, let's transform all the prediction into the winners (otherwise each prediction gives us the 10 probabilities, but we only need the winner, the one our network thinks it i pred = np.argmax(predictions, axis=1) # Now, compare the true labels of the test set, to our predicted winners cm = confusion_matrix(test_ds, pred) # print(cm) # Make the confusion matrix a little more visually attractive disp = ConfusionMatrixDisplay(confusion_matrix=cm)
	disp.plot(cmap=plt.cm.Blues) plt.show() AttributeError Traceback (most recent call last) Cell In[34], line 6 2 from sklearn.metrics import confusion_matrix 3 from sklearn.metrics import ConfusionMatrixDisplay
	> 6 ind = list(test_ds.class_indices.values()) 7 keys = list(test_ds.class_indices.keys()) 8 # Generate predictions for all the test images AttributeError: '_PrefetchDataset' object has no attribute 'class_indices' Save the model
In [30]:	model.save("model.tf") INFO:tensorflow:Assets written to: model.tf\assets INFO:tensorflow:Assets written to: model.tf\assets 5. Compare to teachable machine
	basketball 100 Image Samples Lyboad Lyboad Lyboad Lyboad
	golfball /
	Training Model Trained Advanced Approximation to the specific state of the specific s
	100 Image Samples □ Learning Rate: 0.001 ○
	poolball / Reset Defaults 100 Image Samples Under the hood Upload Upload
	tennisball / in the state of th
	La L
	Output Output
	baske golfball socce baske golfball socce
	poolb tenni poolb tenni
	Output Daske Daske
	golfball socce poolb golfball socce 100%
	tenni tenni
	TB I GOT
	Output baske
	golfball socce poolb
	Conclusion It's hard to accuratly train a model with a small amount of data. The quality of the data is also a hig part of the accuracy of the model. At the moment the model is not year, accurate, but I think that with more data and better
In []:	It's hard to accuratly train a model with a small amount of data. The quality of the data is also a big part of the accuracy of the model. At the moment the model is not very accurate, but I think that with more data and better quality data the model could be more accurate. Webscraper *pip install selenium **Reference of the model. At the model is not very accurate, but I think that with more data and better quality data the model could be more accurate, but I think that with more data and better quality data the model could be more accurate, but I think that with more data and better quality data the model could be more accurate.
	<pre># importing libraries from selenium import webdriver from selenium.webdriver.common.action_chains import ActionChains from selenium.webdriver.common.by import By import time import os import requests</pre>
	<pre>from bs4 import BeautifulSoup ## # start the chrome driver driver = webdriver.Chrome('C:/chromedriver/chromedriver.exe') # categories we want to search for and the url for each category categories = {</pre>
	"soccerball": "https://unsplash.com/s/photos/soccer-ball", "tennisball": "https://unsplash.com/s/photos/golf-ball", "golfball": "https://unsplash.com/s/photos/golf-ball", "basketball": "https://unsplash.com/s/photos/basket-ball", "poolball": "https://unsplash.com/s/photos/pool-ball" } # check if the image folder exists, if not create it
	<pre>if not os.path.exists("images"): os.makedirs("images") # loop through the categories for category, url in categories.items(): # search for the url which is the value of the category driver.get(url)</pre>
	<pre>images = [] # scroll down the page and load more images until we have 150 images while len(images) < 150: driver.execute_script("window.scrollBy(0, 1000)") time.sleep(1) try:</pre>
	<pre>el = driver.find_element(By.XPATH, '//button[text()="Load more"]') ActionChains(driver).click(el).perform() time.sleep(1) except: pass page_html = driver.page_source soup = BeautifulSoup(page_html, 'html.parser')</pre>
	<pre># find all the images on the page images = soup.findAll('img', {'class':"tB6UZ a5VGX"}) # create a folder for each category category_path = os.path.join('images', category) if not os.path.exists(category_path): os.makedirs(category_path)</pre>
	<pre># save the images in the folder i = 1 for image in images: try: response = requests.get(image['src'], stream=True) with open(category_path + '/'+ str(category)+str(i)+'.jpg', "wb") as file: file.write(response.content) i += 1 except:</pre>
In []:	driver.quit() Streamlit app import streamlit as st
	<pre>import os from keras.utils import image_dataset_from_directory import tensorflow as tf from tensorflow import keras from tensorflow.keras import optimizers from tensorflow.keras import layers from keras.models import Sequential from keras.layers import Conv2D from keras.layers import MaxPooling2D</pre>
	from keras.layers import Flatten from keras.layers import Dense from keras.layers import Dropout import matplotlib.pyplot as plt ## # Function to load and preprocess datasets
	<pre>def load_datasets(): batch_size = 32 image_size = (64, 64) validation_split = 0.2 # Create the training dataset from the 'train' directory train_ds = image_dataset_from_directory(</pre>
	<pre>labels='inferred', label_mode='categorical', batch_size=batch_size, image_size=image_size, validation_split=validation_split, subset='training', seed=123)</pre>
	<pre># Create the validation dataset from the 'train' directory validation_ds = image_dataset_from_directory(directory='images/training_set', labels='inferred', label_mode='categorical', batch_size=batch_size, image_size=image_size, validation_split=validation_split, subset='validation'.</pre>
	<pre>validation_split=validation', subset='validation', seed=123) # Create the testing dataset from the 'test' directory test_ds = image_dataset_from_directory(directory='images/test_set', labels='inferred', label_mode='categorical',</pre>
	<pre>label_mode='categorical', batch_size=batch_size, image_size=image_size) return train_ds, validation_ds, test_ds ##</pre>
	<pre># Function to create and train the model def create_and_train_model(train_ds, validation_ds, epochs): NUM_CLASSES = 5 IMG_SIZE = 128 HEIGTH_FACTOR = 0.2 WIDTH_FACTOR = 0.2</pre>
	<pre># Create a sequential model with a list of layers model = tf.keras.Sequential([layers.Resizing(IMG_SIZE, IMG_SIZE), layers.Rescaling(1./255), layers.RandomFlip("horizontal"), layers.RandomTranslation(HEIGTH_FACTOR), layers.RandomZoom(0.2), layers.Conv2D(32, (3, 3), input_shape=(64, 64, 3), activation="relu"), layers.MaxPooling2D((2, 2)),</pre>
	<pre>layers.Dense(NUM_CLASSES, activation="sigmoid")]) # Compile and train the model model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy']) history = model.fit(train_ds, validation_data=validation_ds, epochs=epochs) return model, history #</pre>
	<pre># Function to display plots def display_plots(history): # Create a figure and a grid of subplots with a single call fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5)) # Plot the loss curves on the first subplot ax1.plot(history.history['loss'], label='training loss')</pre>
	<pre>ax1.plot(history.history['loss'], label='training loss') ax1.plot(history.history['val_loss'], label='validation loss') ax1.set_title('Loss curves') ax1.set_xlabel('Epoch') ax1.set_ylabel('Loss') ax1.legend() # Plot the accuracy curves on the second subplot ax2.plot(history.history['accuracy'], label='training accuracy')</pre>
	<pre>ax2.plot(history.history['val_accuracy'], label='validation accuracy') ax2.set_title('Accuracy curves') ax2.set_xlabel('Epoch') ax2.set_ylabel('Accuracy') ax2.legend() # Adjust the spacing between subplots fig.tight_layout()</pre>
	<pre># Show the figure st.pyplot(fig) ## # Function to evaluate and display test accuracy def display_test_accuracy(model, test_ds): test_loss, test_acc = model.evaluate(test_ds)</pre>
	<pre># Button to trigger model building and training if st.button("Train Model"): # Load datasets train_ds, validation_ds, test_ds = load_datasets() # Display loading message with st.spinner("Training the model. Please wait"): # Create and train the model</pre>
	# Create and train the model model, history = create_and_train_model(train_ds, validation_ds, epochs) # Display training and validation curves st.subheader("Training and Validation Curves") display_plots(history) # Display test accuracy st.subheader("Test Accuracy") display_test_accuracy(model, test_ds)
	<pre>display_test_accuracy(model, test_ds) ifname == "main": main()</pre>