Task3_DL_RafEngelen

December 15, 2023

1 Report task 3: Compare my model to Google's model

1.1 Introduction

The purpose of this report is to compare a Deep Learning model to a model created by the Teachable Machine by Google

In this file you will find some code written in comments. This is code that is only necessary to run on Streamlit. The version on Streamlit will have that code not in comments.

Streamlit link: https://rafengelenaitask3.streamlit.app

1.2 Scraping for images

First of all, I need images so that I can train our Deep Learning model. This needs a lot of images (100+) and the more images gathered, the better the model is trained.

[1]:
```
import fastbook
fastbook.setup_book()
import os
from fastbook import *
```

Next we need our different categories we want to train the model for. I have chosen to characters in Star Wars. So the goal of our algorithm is to tell which character there is on an image. For structure, i create a folders. The main "images" folder and there are folders for each category.

[2]:
```
categories=["Darth Vader", "Yoda", "Luke Skywalker", "R2D2", "C3PO"]
data_dir="./images/"
def setup_directories(categories):
    if not os.path.exists(data_dir):
        os.makedirs(data_dir)
    for category in categories:
        directory_path = data_dir+category.replace(" ", "_")
        print(directory_path)
        if not os.path.exists(directory_path):
            os.makedirs(directory_path)


setup_directories(categories)
```

```
./images/Darth_Vader
./images/Yoda
./images/Luke_Skywalker
./images/R2D2
./images/C3PO
```

I have chosen not to create a regular web scraper to get images. I use fastbook, with this module i can connect to some API's, the API i am sending requests to is the DuckDuckGo images API. After some testing I have noticed that with regular web scrapers, the program quits after a set amount of images. For Google this was 20 and for Bing this was 35. If i wanted to gather 100+ images per category, i would have to make multiple web scrapers, so this seemed as the better option.

First i loop through the categories to gain 150 results. I have chosen 150 images, because I have noticed that not all the images can be downloaded this way. That's why I've chosen a number bigger than 100 and after executing the program I also have more than 100 images left.

After that I loop through all the image links and I send a get request to get the image. I check for errors and also for if the response status is 200. I save the image in the correct folder with a unique name. I also print errors in case they happen.

[3]:

```
category_names = []
def get_images(categories):

    for category in categories:
        # Search on DuckDuckGo for images of current category
        results = search_images_ddg(category, max_images=150)
        save_and_check_image(category=category, results=results)

def save_and_check_image(category, results):
    #setup directory path to save images, each category has it's own directory

    category_name = category.replace(" ","_")
    directory_path = data_dir+category_name
    category_names.append(category_name)
    # number of correctly downloaded image
    img_number=1

    #loop through all links and catch errors that may occur
    for index, link in enumerate(results):
        try:
            # get the images
            response = requests.get(link, timeout=10)

            # Check if the request was successful
            if response.status_code == 200:

                # Get file extension and check if they are a correct image␣
 ↪extension
                file_extension = link.split('.')[-1]
                if file_extension in ["jpeg", "jpg", "png"]:

                    # Save the image to a file with img_number as a identifier
                    filename = f"{category_name}_duckduckgo_{img_number}.
 ↪{file_extension}"
                    filepath = os.path.join(directory_path, filename)

                    with open(filepath, 'wb') as file:
                        for chunk in response.iter_content(chunk_size=128):
                            file.write(chunk)


                    print(f"Image {index + 1} downloaded: {filename}")
                    img_number+=1
            else:
                print(f"Invalid response status for image {index+1}: {response.
 ↪status_code}")
        except Exception as e:
            print(f"Error downloading image {index+1}: {str(e)}")
            continue

get_images(categories)
```

```
Image 1 downloaded: Darth_Vader_duckduckgo_1.jpg
Image 2 downloaded: Darth_Vader_duckduckgo_2.png
Image 3 downloaded: Darth_Vader_duckduckgo_3.jpg
Image 4 downloaded: Darth_Vader_duckduckgo_4.jpg
Image 5 downloaded: Darth_Vader_duckduckgo_5.jpg
Image 6 downloaded: Darth_Vader_duckduckgo_6.jpg
Image 7 downloaded: Darth_Vader_duckduckgo_7.jpg
Image 8 downloaded: Darth_Vader_duckduckgo_8.jpg
Image 9 downloaded: Darth_Vader_duckduckgo_9.jpg
Image 10 downloaded: Darth_Vader_duckduckgo_10.png
Image 11 downloaded: Darth_Vader_duckduckgo_11.png
Image 12 downloaded: Darth_Vader_duckduckgo_12.jpg
Image 13 downloaded: Darth_Vader_duckduckgo_13.jpg
Invalid response status for image 14: 403
Image 15 downloaded: Darth_Vader_duckduckgo_14.jpg
Image 16 downloaded: Darth_Vader_duckduckgo_15.jpg
Image 17 downloaded: Darth_Vader_duckduckgo_16.jpg
Image 19 downloaded: Darth_Vader_duckduckgo_17.jpg
Image 20 downloaded: Darth_Vader_duckduckgo_18.jpg
Image 21 downloaded: Darth_Vader_duckduckgo_19.jpg
Image 22 downloaded: Darth_Vader_duckduckgo_20.jpg
Image 23 downloaded: Darth_Vader_duckduckgo_21.png
Image 24 downloaded: Darth_Vader_duckduckgo_22.jpg
Image 25 downloaded: Darth_Vader_duckduckgo_23.jpg
Image 26 downloaded: Darth_Vader_duckduckgo_24.png
Image 27 downloaded: Darth_Vader_duckduckgo_25.png
Invalid response status for image 28: 406
Image 29 downloaded: Darth_Vader_duckduckgo_26.jpg
Image 30 downloaded: Darth_Vader_duckduckgo_27.jpg
Image 31 downloaded: Darth_Vader_duckduckgo_28.jpg
Invalid response status for image 32: 404
Image 33 downloaded: Darth_Vader_duckduckgo_29.jpg
Image 34 downloaded: Darth_Vader_duckduckgo_30.jpg
Image 35 downloaded: Darth_Vader_duckduckgo_31.jpg
Image 37 downloaded: Darth_Vader_duckduckgo_32.jpg
Image 39 downloaded: Darth_Vader_duckduckgo_33.jpg
Image 40 downloaded: Darth_Vader_duckduckgo_34.jpg
Image 41 downloaded: Darth_Vader_duckduckgo_35.png
Image 42 downloaded: Darth_Vader_duckduckgo_36.jpg
Image 43 downloaded: Darth_Vader_duckduckgo_37.jpg
Image 44 downloaded: Darth_Vader_duckduckgo_38.jpg
Image 45 downloaded: Darth_Vader_duckduckgo_39.jpg
Image 46 downloaded: Darth_Vader_duckduckgo_40.jpg
Image 47 downloaded: Darth_Vader_duckduckgo_41.jpg
Image 48 downloaded: Darth_Vader_duckduckgo_42.jpg
Image 49 downloaded: Darth_Vader_duckduckgo_43.png
Image 50 downloaded: Darth_Vader_duckduckgo_44.png
Invalid response status for image 51: 403
Image 52 downloaded: Darth_Vader_duckduckgo_45.jpg
Image 53 downloaded: Darth_Vader_duckduckgo_46.jpg
Image 55 downloaded: Darth_Vader_duckduckgo_47.jpg
Image 56 downloaded: Darth_Vader_duckduckgo_48.jpg
Invalid response status for image 57: 403
Image 58 downloaded: Darth_Vader_duckduckgo_49.jpg
Image 59 downloaded: Darth_Vader_duckduckgo_50.jpg
Image 60 downloaded: Darth_Vader_duckduckgo_51.jpg
Image 61 downloaded: Darth_Vader_duckduckgo_52.jpg
Image 62 downloaded: Darth_Vader_duckduckgo_53.jpg
Image 63 downloaded: Darth_Vader_duckduckgo_54.jpg
Image 64 downloaded: Darth_Vader_duckduckgo_55.jpg
Image 65 downloaded: Darth_Vader_duckduckgo_56.png
Invalid response status for image 66: 404
Image 67 downloaded: Darth_Vader_duckduckgo_57.png
Invalid response status for image 68: 403
Image 69 downloaded: Darth_Vader_duckduckgo_58.jpg
Image 71 downloaded: Darth_Vader_duckduckgo_59.jpg
Image 72 downloaded: Darth_Vader_duckduckgo_60.jpg
Image 73 downloaded: Darth_Vader_duckduckgo_61.jpg
Image 74 downloaded: Darth_Vader_duckduckgo_62.jpg
Image 75 downloaded: Darth_Vader_duckduckgo_63.jpg
Invalid response status for image 76: 403
Image 77 downloaded: Darth_Vader_duckduckgo_64.jpg
Image 78 downloaded: Darth_Vader_duckduckgo_65.png
Image 79 downloaded: Darth_Vader_duckduckgo_66.jpg
Invalid response status for image 80: 403
Image 81 downloaded: Darth_Vader_duckduckgo_67.jpg
Image 82 downloaded: Darth_Vader_duckduckgo_68.jpg
Invalid response status for image 83: 403
Image 84 downloaded: Darth_Vader_duckduckgo_69.jpg
Image 85 downloaded: Darth_Vader_duckduckgo_70.jpg
Image 86 downloaded: Darth_Vader_duckduckgo_71.jpg
Image 87 downloaded: Darth_Vader_duckduckgo_72.jpg
Image 88 downloaded: Darth_Vader_duckduckgo_73.jpg
Error downloading image 89:
HTTPSConnectionPool(host='comicvine1.cbsistatic.com', port=443): Max retries
exceeded with url:
/uploads/original/7/70801/6032361-570900._sx1280_ql80_ttd_.jpg (Caused by
SSLError(SSLCertVerificationError(1, "[SSL: CERTIFICATE_VERIFY_FAILED]
certificate verify failed: Hostname mismatch, certificate is not valid for
'comicvine1.cbsistatic.com'. (_ssl.c:1006)")))
Image 90 downloaded: Darth_Vader_duckduckgo_74.jpg
Image 91 downloaded: Darth_Vader_duckduckgo_75.jpg
Image 92 downloaded: Darth_Vader_duckduckgo_76.jpg
Image 93 downloaded: Darth_Vader_duckduckgo_77.jpg
Image 94 downloaded: Darth_Vader_duckduckgo_78.jpg
Image 95 downloaded: Darth_Vader_duckduckgo_79.png
Image 96 downloaded: Darth_Vader_duckduckgo_80.jpg
Image 97 downloaded: Darth_Vader_duckduckgo_81.png
Image 98 downloaded: Darth_Vader_duckduckgo_82.jpg
Image 99 downloaded: Darth_Vader_duckduckgo_83.png
Image 101 downloaded: Darth_Vader_duckduckgo_84.jpg
Image 102 downloaded: Darth_Vader_duckduckgo_85.jpg
Image 103 downloaded: Darth_Vader_duckduckgo_86.jpg
Image 104 downloaded: Darth_Vader_duckduckgo_87.jpeg
Invalid response status for image 105: 404
Image 106 downloaded: Darth_Vader_duckduckgo_88.jpg
Image 108 downloaded: Darth_Vader_duckduckgo_89.jpg
Image 109 downloaded: Darth_Vader_duckduckgo_90.jpg
Image 110 downloaded: Darth_Vader_duckduckgo_91.jpg
Invalid response status for image 111: 403
Image 113 downloaded: Darth_Vader_duckduckgo_92.jpg
Image 115 downloaded: Darth_Vader_duckduckgo_93.jpg
Image 116 downloaded: Darth_Vader_duckduckgo_94.jpg
Image 117 downloaded: Darth_Vader_duckduckgo_95.jpg
Image 119 downloaded: Darth_Vader_duckduckgo_96.jpg
Image 120 downloaded: Darth_Vader_duckduckgo_97.jpg
Image 121 downloaded: Darth_Vader_duckduckgo_98.jpg
Image 123 downloaded: Darth_Vader_duckduckgo_99.jpg
Image 124 downloaded: Darth_Vader_duckduckgo_100.jpg
Image 125 downloaded: Darth_Vader_duckduckgo_101.jpg
Image 126 downloaded: Darth_Vader_duckduckgo_102.jpg
Image 127 downloaded: Darth_Vader_duckduckgo_103.png
Image 128 downloaded: Darth_Vader_duckduckgo_104.jpg
Image 129 downloaded: Darth_Vader_duckduckgo_105.jpg
Image 130 downloaded: Darth_Vader_duckduckgo_106.jpg
Image 131 downloaded: Darth_Vader_duckduckgo_107.jpg
Image 132 downloaded: Darth_Vader_duckduckgo_108.jpg
Image 133 downloaded: Darth_Vader_duckduckgo_109.jpg
Image 134 downloaded: Darth_Vader_duckduckgo_110.jpg
Image 135 downloaded: Darth_Vader_duckduckgo_111.jpg
Image 136 downloaded: Darth_Vader_duckduckgo_112.jpeg
Image 137 downloaded: Darth_Vader_duckduckgo_113.jpg
Image 138 downloaded: Darth_Vader_duckduckgo_114.jpg
Image 139 downloaded: Darth_Vader_duckduckgo_115.jpg
Image 140 downloaded: Darth_Vader_duckduckgo_116.jpg
Image 141 downloaded: Darth_Vader_duckduckgo_117.jpg
Image 143 downloaded: Darth_Vader_duckduckgo_118.png
Image 144 downloaded: Darth_Vader_duckduckgo_119.jpg
Image 145 downloaded: Darth_Vader_duckduckgo_120.jpg
Image 146 downloaded: Darth_Vader_duckduckgo_121.jpg
Image 149 downloaded: Darth_Vader_duckduckgo_122.jpg
Image 150 downloaded: Darth_Vader_duckduckgo_123.jpg
Image 2 downloaded: Yoda_duckduckgo_1.jpg
Error downloading image 3: HTTPSConnectionPool(host='openclipart.org',
port=443): Max retries exceeded with url:
/image/2400px/svg_to_png/227445/Yoda.png (Caused by
SSLError(SSLCertVerificationError(1, '[SSL: CERTIFICATE_VERIFY_FAILED]
certificate verify failed: certificate has expired (_ssl.c:1006)')))
Image 4 downloaded: Yoda_duckduckgo_2.jpg
Image 5 downloaded: Yoda_duckduckgo_3.jpg
Error downloading image 6: HTTPConnectionPool(host='homepage.eircom.net',
port=80): Max retries exceeded with url: /~starwars/pictures/yoda2.jpg (Caused
by NameResolutionError("<urllib3.connection.HTTPConnection object at
0x0000021FBFB03150>: Failed to resolve 'homepage.eircom.net' ([Errno 11001]
getaddrinfo failed)"))
Image 7 downloaded: Yoda_duckduckgo_4.jpg
Image 8 downloaded: Yoda_duckduckgo_5.png
Image 9 downloaded: Yoda_duckduckgo_6.jpg
Image 10 downloaded: Yoda_duckduckgo_7.jpg
Image 11 downloaded: Yoda_duckduckgo_8.jpg
Image 12 downloaded: Yoda_duckduckgo_9.jpg
Image 13 downloaded: Yoda_duckduckgo_10.jpg
Image 14 downloaded: Yoda_duckduckgo_11.jpg
Image 15 downloaded: Yoda_duckduckgo_12.jpg
Image 17 downloaded: Yoda_duckduckgo_13.jpg
Image 18 downloaded: Yoda_duckduckgo_14.jpg
Image 19 downloaded: Yoda_duckduckgo_15.jpg
Image 20 downloaded: Yoda_duckduckgo_16.jpg
Image 21 downloaded: Yoda_duckduckgo_17.jpg
Image 22 downloaded: Yoda_duckduckgo_18.jpg
Image 24 downloaded: Yoda_duckduckgo_19.jpg
Invalid response status for image 25: 403
Image 26 downloaded: Yoda_duckduckgo_20.jpg
Image 27 downloaded: Yoda_duckduckgo_21.jpg
Image 28 downloaded: Yoda_duckduckgo_22.jpg
Image 29 downloaded: Yoda_duckduckgo_23.jpg
Image 30 downloaded: Yoda_duckduckgo_24.jpg
Image 31 downloaded: Yoda_duckduckgo_25.jpg
Image 32 downloaded: Yoda_duckduckgo_26.jpg
Image 33 downloaded: Yoda_duckduckgo_27.jpg
Image 34 downloaded: Yoda_duckduckgo_28.jpg
Image 35 downloaded: Yoda_duckduckgo_29.jpg
Image 36 downloaded: Yoda_duckduckgo_30.jpg
Invalid response status for image 37: 404
Image 38 downloaded: Yoda_duckduckgo_31.jpg
Image 39 downloaded: Yoda_duckduckgo_32.jpeg
Image 40 downloaded: Yoda_duckduckgo_33.jpg
Image 41 downloaded: Yoda_duckduckgo_34.jpg
Image 42 downloaded: Yoda_duckduckgo_35.jpg
Invalid response status for image 43: 403
Error downloading image 44: HTTPConnectionPool(host='nanozen.info', port=80):
Max retries exceeded with url: /wp-content/uploads/2010/03/yoda.jpg (Caused by
NameResolutionError("<urllib3.connection.HTTPConnection object at
0x0000021FBF749150>: Failed to resolve 'nanozen.info' ([Errno 11001] getaddrinfo
failed)"))
Image 45 downloaded: Yoda_duckduckgo_36.jpg
Image 46 downloaded: Yoda_duckduckgo_37.jpg
Image 47 downloaded: Yoda_duckduckgo_38.jpg
Invalid response status for image 48: 406
Image 49 downloaded: Yoda_duckduckgo_39.jpg
Image 50 downloaded: Yoda_duckduckgo_40.jpg
Image 51 downloaded: Yoda_duckduckgo_41.jpg
Image 52 downloaded: Yoda_duckduckgo_42.jpg
Image 53 downloaded: Yoda_duckduckgo_43.jpg
Image 54 downloaded: Yoda_duckduckgo_44.jpg
Image 55 downloaded: Yoda_duckduckgo_45.png
Image 56 downloaded: Yoda_duckduckgo_46.jpg
Image 57 downloaded: Yoda_duckduckgo_47.jpg
Image 58 downloaded: Yoda_duckduckgo_48.png
Image 59 downloaded: Yoda_duckduckgo_49.jpg
Error downloading image 60: HTTPSConnectionPool(host='clipart.world', port=443):
Max retries exceeded with url: /wp-content/uploads/2021/03/Yoda-clipart-2.png
(Caused by SSLError(SSLCertVerificationError(1, '[SSL:
CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get local issuer
certificate (_ssl.c:1006)')))
Image 61 downloaded: Yoda_duckduckgo_50.jpg
Image 62 downloaded: Yoda_duckduckgo_51.jpg
Image 63 downloaded: Yoda_duckduckgo_52.jpg
Image 64 downloaded: Yoda_duckduckgo_53.jpg
Image 65 downloaded: Yoda_duckduckgo_54.jpg
Image 66 downloaded: Yoda_duckduckgo_55.png
Image 67 downloaded: Yoda_duckduckgo_56.jpg
Image 68 downloaded: Yoda_duckduckgo_57.jpg
Image 69 downloaded: Yoda_duckduckgo_58.jpg
Image 70 downloaded: Yoda_duckduckgo_59.jpg
Image 71 downloaded: Yoda_duckduckgo_60.jpg
Image 72 downloaded: Yoda_duckduckgo_61.png
Image 73 downloaded: Yoda_duckduckgo_62.jpg
Image 74 downloaded: Yoda_duckduckgo_63.jpg
Image 75 downloaded: Yoda_duckduckgo_64.jpg
Image 77 downloaded: Yoda_duckduckgo_65.jpg
Image 78 downloaded: Yoda_duckduckgo_66.jpg
Image 79 downloaded: Yoda_duckduckgo_67.jpg
Image 80 downloaded: Yoda_duckduckgo_68.jpg
Image 81 downloaded: Yoda_duckduckgo_69.jpg
Image 82 downloaded: Yoda_duckduckgo_70.jpg
Image 83 downloaded: Yoda_duckduckgo_71.jpg
Image 84 downloaded: Yoda_duckduckgo_72.jpg
Image 85 downloaded: Yoda_duckduckgo_73.png
Image 86 downloaded: Yoda_duckduckgo_74.jpg
Image 87 downloaded: Yoda_duckduckgo_75.jpg
Image 88 downloaded: Yoda_duckduckgo_76.jpg
Image 89 downloaded: Yoda_duckduckgo_77.jpg
Image 90 downloaded: Yoda_duckduckgo_78.jpg
Image 91 downloaded: Yoda_duckduckgo_79.jpg
Image 92 downloaded: Yoda_duckduckgo_80.jpg
Image 93 downloaded: Yoda_duckduckgo_81.jpg
Image 94 downloaded: Yoda_duckduckgo_82.jpg
Image 95 downloaded: Yoda_duckduckgo_83.jpg
Image 96 downloaded: Yoda_duckduckgo_84.jpg
Image 97 downloaded: Yoda_duckduckgo_85.jpg
Image 98 downloaded: Yoda_duckduckgo_86.jpg
Image 99 downloaded: Yoda_duckduckgo_87.jpg
Image 100 downloaded: Yoda_duckduckgo_88.png
Image 101 downloaded: Yoda_duckduckgo_89.jpg
Image 102 downloaded: Yoda_duckduckgo_90.jpg
Image 103 downloaded: Yoda_duckduckgo_91.jpg
Image 104 downloaded: Yoda_duckduckgo_92.jpg
Invalid response status for image 105: 404
Image 106 downloaded: Yoda_duckduckgo_93.jpg
Image 107 downloaded: Yoda_duckduckgo_94.jpg
Image 108 downloaded: Yoda_duckduckgo_95.jpg
Image 109 downloaded: Yoda_duckduckgo_96.jpg
Image 110 downloaded: Yoda_duckduckgo_97.jpg
Image 111 downloaded: Yoda_duckduckgo_98.png
Image 112 downloaded: Yoda_duckduckgo_99.jpg
Image 113 downloaded: Yoda_duckduckgo_100.jpg
Image 114 downloaded: Yoda_duckduckgo_101.jpg
Image 115 downloaded: Yoda_duckduckgo_102.jpg
Image 116 downloaded: Yoda_duckduckgo_103.jpg
Image 117 downloaded: Yoda_duckduckgo_104.jpg
Image 118 downloaded: Yoda_duckduckgo_105.jpg
Image 119 downloaded: Yoda_duckduckgo_106.jpg
Image 120 downloaded: Yoda_duckduckgo_107.jpg
Image 121 downloaded: Yoda_duckduckgo_108.jpg
Image 122 downloaded: Yoda_duckduckgo_109.jpg
Image 123 downloaded: Yoda_duckduckgo_110.jpg
Image 124 downloaded: Yoda_duckduckgo_111.jpg
Image 125 downloaded: Yoda_duckduckgo_112.jpg
Image 126 downloaded: Yoda_duckduckgo_113.jpg
Image 127 downloaded: Yoda_duckduckgo_114.jpg
Image 128 downloaded: Yoda_duckduckgo_115.jpeg
Image 129 downloaded: Yoda_duckduckgo_116.jpg
Image 130 downloaded: Yoda_duckduckgo_117.jpg
Image 131 downloaded: Yoda_duckduckgo_118.jpg
Image 132 downloaded: Yoda_duckduckgo_119.jpg
Image 133 downloaded: Yoda_duckduckgo_120.jpg
Image 134 downloaded: Yoda_duckduckgo_121.jpg
Image 135 downloaded: Yoda_duckduckgo_122.jpg
Image 136 downloaded: Yoda_duckduckgo_123.jpg
Image 137 downloaded: Yoda_duckduckgo_124.png
Error downloading image 138: HTTPSConnectionPool(host='clipart.world',
port=443): Max retries exceeded with url: /wp-content/uploads/2020/07/yoda-
fighting.jpg (Caused by SSLError(SSLCertVerificationError(1, '[SSL:
CERTIFICATE_VERIFY_FAILED] certificate verify failed: unable to get local issuer
certificate (_ssl.c:1006)')))
Image 139 downloaded: Yoda_duckduckgo_125.jpg
Image 140 downloaded: Yoda_duckduckgo_126.png
Image 141 downloaded: Yoda_duckduckgo_127.jpg
Image 142 downloaded: Yoda_duckduckgo_128.png
Image 143 downloaded: Yoda_duckduckgo_129.jpg
Image 144 downloaded: Yoda_duckduckgo_130.jpg
Image 145 downloaded: Yoda_duckduckgo_131.jpg
Image 146 downloaded: Yoda_duckduckgo_132.jpg
Invalid response status for image 147: 403
Image 149 downloaded: Yoda_duckduckgo_133.png
Image 150 downloaded: Yoda_duckduckgo_134.jpg
Image 2 downloaded: Luke_Skywalker_duckduckgo_1.jpg
Image 3 downloaded: Luke_Skywalker_duckduckgo_2.jpg
Image 4 downloaded: Luke_Skywalker_duckduckgo_3.png
Image 5 downloaded: Luke_Skywalker_duckduckgo_4.jpg
Image 6 downloaded: Luke_Skywalker_duckduckgo_5.jpg
Image 7 downloaded: Luke_Skywalker_duckduckgo_6.png
Image 8 downloaded: Luke_Skywalker_duckduckgo_7.jpg
Image 9 downloaded: Luke_Skywalker_duckduckgo_8.jpg
Invalid response status for image 10: 403
Image 11 downloaded: Luke_Skywalker_duckduckgo_9.jpg
Image 12 downloaded: Luke_Skywalker_duckduckgo_10.jpg
Image 14 downloaded: Luke_Skywalker_duckduckgo_11.jpeg
Image 16 downloaded: Luke_Skywalker_duckduckgo_12.jpg
Image 17 downloaded: Luke_Skywalker_duckduckgo_13.jpg
Image 18 downloaded: Luke_Skywalker_duckduckgo_14.jpg
Image 19 downloaded: Luke_Skywalker_duckduckgo_15.jpg
Image 20 downloaded: Luke_Skywalker_duckduckgo_16.jpg
Image 21 downloaded: Luke_Skywalker_duckduckgo_17.jpg
Image 22 downloaded: Luke_Skywalker_duckduckgo_18.png
Image 23 downloaded: Luke_Skywalker_duckduckgo_19.jpg
Image 24 downloaded: Luke_Skywalker_duckduckgo_20.png
Image 25 downloaded: Luke_Skywalker_duckduckgo_21.jpeg
Image 27 downloaded: Luke_Skywalker_duckduckgo_22.png
Image 28 downloaded: Luke_Skywalker_duckduckgo_23.png
Image 29 downloaded: Luke_Skywalker_duckduckgo_24.jpg
Image 30 downloaded: Luke_Skywalker_duckduckgo_25.jpg
Image 31 downloaded: Luke_Skywalker_duckduckgo_26.jpg
Image 34 downloaded: Luke_Skywalker_duckduckgo_27.jpg
Image 35 downloaded: Luke_Skywalker_duckduckgo_28.png
Image 36 downloaded: Luke_Skywalker_duckduckgo_29.jpg
Image 37 downloaded: Luke_Skywalker_duckduckgo_30.jpg
Invalid response status for image 39: 404
Image 40 downloaded: Luke_Skywalker_duckduckgo_31.jpg
Image 41 downloaded: Luke_Skywalker_duckduckgo_32.jpg
Image 42 downloaded: Luke_Skywalker_duckduckgo_33.jpg
Image 43 downloaded: Luke_Skywalker_duckduckgo_34.jpg
Image 44 downloaded: Luke_Skywalker_duckduckgo_35.png
Image 45 downloaded: Luke_Skywalker_duckduckgo_36.jpg
Image 46 downloaded: Luke_Skywalker_duckduckgo_37.jpg
Image 47 downloaded: Luke_Skywalker_duckduckgo_38.jpg
Image 48 downloaded: Luke_Skywalker_duckduckgo_39.jpg
Image 49 downloaded: Luke_Skywalker_duckduckgo_40.jpg
Image 51 downloaded: Luke_Skywalker_duckduckgo_41.jpg
Image 55 downloaded: Luke_Skywalker_duckduckgo_42.jpg
Image 56 downloaded: Luke_Skywalker_duckduckgo_43.jpg
Image 57 downloaded: Luke_Skywalker_duckduckgo_44.jpg
Image 58 downloaded: Luke_Skywalker_duckduckgo_45.jpg
Image 59 downloaded: Luke_Skywalker_duckduckgo_46.jpg
Image 60 downloaded: Luke_Skywalker_duckduckgo_47.jpg
Image 61 downloaded: Luke_Skywalker_duckduckgo_48.jpg
Image 62 downloaded: Luke_Skywalker_duckduckgo_49.jpg
Image 63 downloaded: Luke_Skywalker_duckduckgo_50.jpg
Image 64 downloaded: Luke_Skywalker_duckduckgo_51.jpg
```

```
Invalid response status for image 65: 406
Image 66 downloaded: Luke_Skywalker_duckduckgo_52.jpg
Image 67 downloaded: Luke_Skywalker_duckduckgo_53.jpg
Image 68 downloaded: Luke_Skywalker_duckduckgo_54.jpg
Image 70 downloaded: Luke_Skywalker_duckduckgo_55.jpg
Image 71 downloaded: Luke_Skywalker_duckduckgo_56.jpg
Image 72 downloaded: Luke_Skywalker_duckduckgo_57.jpg
Image 76 downloaded: Luke_Skywalker_duckduckgo_58.jpg
Invalid response status for image 77: 404
Image 78 downloaded: Luke_Skywalker_duckduckgo_59.jpg
Image 79 downloaded: Luke_Skywalker_duckduckgo_60.jpg
Image 80 downloaded: Luke_Skywalker_duckduckgo_61.jpg
Image 81 downloaded: Luke_Skywalker_duckduckgo_62.jpg
Image 82 downloaded: Luke_Skywalker_duckduckgo_63.jpg
Image 83 downloaded: Luke_Skywalker_duckduckgo_64.jpg
Image 84 downloaded: Luke_Skywalker_duckduckgo_65.jpg
Image 85 downloaded: Luke_Skywalker_duckduckgo_66.jpg
Image 86 downloaded: Luke_Skywalker_duckduckgo_67.jpg
Image 87 downloaded: Luke_Skywalker_duckduckgo_68.jpg
Image 89 downloaded: Luke_Skywalker_duckduckgo_69.jpg
Image 90 downloaded: Luke_Skywalker_duckduckgo_70.jpg
Image 92 downloaded: Luke_Skywalker_duckduckgo_71.png
Image 93 downloaded: Luke_Skywalker_duckduckgo_72.jpg
Image 94 downloaded: Luke_Skywalker_duckduckgo_73.jpg
Image 95 downloaded: Luke_Skywalker_duckduckgo_74.jpg
Image 96 downloaded: Luke_Skywalker_duckduckgo_75.jpg
Image 98 downloaded: Luke_Skywalker_duckduckgo_76.jpg
Image 100 downloaded: Luke_Skywalker_duckduckgo_77.jpg
Image 102 downloaded: Luke_Skywalker_duckduckgo_78.jpg
Image 103 downloaded: Luke_Skywalker_duckduckgo_79.jpg
Image 104 downloaded: Luke_Skywalker_duckduckgo_80.png
Image 105 downloaded: Luke_Skywalker_duckduckgo_81.jpg
Image 106 downloaded: Luke_Skywalker_duckduckgo_82.jpg
Image 107 downloaded: Luke_Skywalker_duckduckgo_83.png
Image 108 downloaded: Luke_Skywalker_duckduckgo_84.jpg
Image 109 downloaded: Luke_Skywalker_duckduckgo_85.jpg
Image 110 downloaded: Luke_Skywalker_duckduckgo_86.jpg
Image 111 downloaded: Luke_Skywalker_duckduckgo_87.jpg
Image 113 downloaded: Luke_Skywalker_duckduckgo_88.jpg
Image 114 downloaded: Luke_Skywalker_duckduckgo_89.jpg
Invalid response status for image 115: 403
Image 116 downloaded: Luke_Skywalker_duckduckgo_90.jpg
Image 118 downloaded: Luke_Skywalker_duckduckgo_91.jpg
Image 119 downloaded: Luke_Skywalker_duckduckgo_92.jpg
Image 121 downloaded: Luke_Skywalker_duckduckgo_93.jpg
Image 122 downloaded: Luke_Skywalker_duckduckgo_94.jpg
Image 123 downloaded: Luke_Skywalker_duckduckgo_95.jpg
Image 124 downloaded: Luke_Skywalker_duckduckgo_96.jpg
```

```
Invalid response status for image 128: 503
Image 129 downloaded: Luke_Skywalker_duckduckgo_97.jpg
Image 131 downloaded: Luke_Skywalker_duckduckgo_98.jpg
Image 132 downloaded: Luke_Skywalker_duckduckgo_99.jpg
Image 133 downloaded: Luke_Skywalker_duckduckgo_100.jpg
Image 134 downloaded: Luke_Skywalker_duckduckgo_101.jpg
Image 135 downloaded: Luke_Skywalker_duckduckgo_102.jpg
Image 137 downloaded: Luke_Skywalker_duckduckgo_103.jpg
Image 138 downloaded: Luke_Skywalker_duckduckgo_104.jpg
Image 140 downloaded: Luke_Skywalker_duckduckgo_105.jpg
Image 142 downloaded: Luke_Skywalker_duckduckgo_106.jpg
Image 143 downloaded: Luke_Skywalker_duckduckgo_107.jpeg
Image 144 downloaded: Luke_Skywalker_duckduckgo_108.jpg
Image 145 downloaded: Luke_Skywalker_duckduckgo_109.png
Image 146 downloaded: Luke_Skywalker_duckduckgo_110.jpg
Image 147 downloaded: Luke_Skywalker_duckduckgo_111.jpg
Image 148 downloaded: Luke_Skywalker_duckduckgo_112.jpg
Image 149 downloaded: Luke_Skywalker_duckduckgo_113.jpg
Image 150 downloaded: Luke_Skywalker_duckduckgo_114.jpg
Image 1 downloaded: R2D2_duckduckgo_1.jpg
Image 2 downloaded: R2D2_duckduckgo_2.jpg
Image 3 downloaded: R2D2_duckduckgo_3.jpg
Image 4 downloaded: R2D2_duckduckgo_4.jpg
Image 6 downloaded: R2D2_duckduckgo_5.png
Error downloading image 8: HTTPSConnectionPool(host='canariasnoticias.es',
port=443): Max retries exceeded with url: /sites/default/files/2017/06/r2d2.jpg
(Caused by NameResolutionError("<urllib3.connection.HTTPSConnection object at
0x0000021FBFAD0810>: Failed to resolve 'canariasnoticias.es' ([Errno 11001]
getaddrinfo failed)"))
Image 9 downloaded: R2D2_duckduckgo_6.jpg
Image 10 downloaded: R2D2_duckduckgo_7.jpg
Image 11 downloaded: R2D2_duckduckgo_8.jpg
Image 12 downloaded: R2D2_duckduckgo_9.jpg
Image 13 downloaded: R2D2_duckduckgo_10.jpg
Image 15 downloaded: R2D2_duckduckgo_11.jpg
Invalid response status for image 16: 404
Image 17 downloaded: R2D2_duckduckgo_12.jpg
Image 18 downloaded: R2D2_duckduckgo_13.jpg
Image 19 downloaded: R2D2_duckduckgo_14.jpg
Image 20 downloaded: R2D2_duckduckgo_15.png
Image 21 downloaded: R2D2_duckduckgo_16.jpg
Image 22 downloaded: R2D2_duckduckgo_17.jpg
Image 24 downloaded: R2D2_duckduckgo_18.jpeg
Invalid response status for image 25: 404
Image 27 downloaded: R2D2_duckduckgo_19.jpg
Image 28 downloaded: R2D2_duckduckgo_20.png
Image 29 downloaded: R2D2_duckduckgo_21.jpg
Image 30 downloaded: R2D2_duckduckgo_22.jpg
Image 31 downloaded: R2D2_duckduckgo_23.png
Image 32 downloaded: R2D2_duckduckgo_24.jpg
Image 33 downloaded: R2D2_duckduckgo_25.jpeg
Image 34 downloaded: R2D2_duckduckgo_26.jpg
Image 36 downloaded: R2D2_duckduckgo_27.png
Image 37 downloaded: R2D2_duckduckgo_28.jpg
Image 38 downloaded: R2D2_duckduckgo_29.jpg
Image 39 downloaded: R2D2_duckduckgo_30.png
Image 40 downloaded: R2D2_duckduckgo_31.jpg
Image 41 downloaded: R2D2_duckduckgo_32.jpeg
Image 42 downloaded: R2D2_duckduckgo_33.jpg
Error downloading image 43: ('Connection aborted.', RemoteDisconnected('Remote
end closed connection without response'))
Image 44 downloaded: R2D2_duckduckgo_34.jpg
Image 45 downloaded: R2D2_duckduckgo_35.jpg
Image 46 downloaded: R2D2_duckduckgo_36.jpg
Image 47 downloaded: R2D2_duckduckgo_37.jpg
Image 48 downloaded: R2D2_duckduckgo_38.jpeg
Invalid response status for image 49: 406
Image 51 downloaded: R2D2_duckduckgo_39.jpg
Image 52 downloaded: R2D2_duckduckgo_40.jpg
Image 53 downloaded: R2D2_duckduckgo_41.jpg
Image 54 downloaded: R2D2_duckduckgo_42.jpg
Image 56 downloaded: R2D2_duckduckgo_43.jpg
Image 58 downloaded: R2D2_duckduckgo_44.jpg
Image 60 downloaded: R2D2_duckduckgo_45.png
Image 61 downloaded: R2D2_duckduckgo_46.jpg
Image 62 downloaded: R2D2_duckduckgo_47.jpg
Image 63 downloaded: R2D2_duckduckgo_48.jpg
Invalid response status for image 64: 403
Image 65 downloaded: R2D2_duckduckgo_49.jpg
Error downloading image 66: ('Connection aborted.', RemoteDisconnected('Remote
end closed connection without response'))
Image 68 downloaded: R2D2_duckduckgo_50.jpg
Image 69 downloaded: R2D2_duckduckgo_51.png
Image 70 downloaded: R2D2_duckduckgo_52.jpg
Image 71 downloaded: R2D2_duckduckgo_53.jpg
Invalid response status for image 72: 403
Image 73 downloaded: R2D2_duckduckgo_54.jpg
Image 74 downloaded: R2D2_duckduckgo_55.jpg
Invalid response status for image 75: 403
Image 78 downloaded: R2D2_duckduckgo_56.jpg
Invalid response status for image 79: 403
Image 80 downloaded: R2D2_duckduckgo_57.jpg
Image 81 downloaded: R2D2_duckduckgo_58.jpg
Image 82 downloaded: R2D2_duckduckgo_59.jpg
Image 83 downloaded: R2D2_duckduckgo_60.jpg
Image 84 downloaded: R2D2_duckduckgo_61.jpg
Image 85 downloaded: R2D2_duckduckgo_62.jpg
Image 86 downloaded: R2D2_duckduckgo_63.jpg
Image 87 downloaded: R2D2_duckduckgo_64.jpg
Image 88 downloaded: R2D2_duckduckgo_65.jpg
Image 89 downloaded: R2D2_duckduckgo_66.jpg
Image 90 downloaded: R2D2_duckduckgo_67.jpg
Image 91 downloaded: R2D2_duckduckgo_68.jpg
Image 92 downloaded: R2D2_duckduckgo_69.jpg
Image 93 downloaded: R2D2_duckduckgo_70.jpg
Invalid response status for image 95: 403
Image 96 downloaded: R2D2_duckduckgo_71.jpg
Image 97 downloaded: R2D2_duckduckgo_72.jpg
Image 98 downloaded: R2D2_duckduckgo_73.jpg
Image 99 downloaded: R2D2_duckduckgo_74.jpg
Image 100 downloaded: R2D2_duckduckgo_75.jpeg
Image 101 downloaded: R2D2_duckduckgo_76.jpg
Image 103 downloaded: R2D2_duckduckgo_77.jpg
Image 104 downloaded: R2D2_duckduckgo_78.jpg
Image 105 downloaded: R2D2_duckduckgo_79.png
Image 106 downloaded: R2D2_duckduckgo_80.jpg
Image 107 downloaded: R2D2_duckduckgo_81.jpg
Image 108 downloaded: R2D2_duckduckgo_82.jpg
Image 110 downloaded: R2D2_duckduckgo_83.jpg
Image 111 downloaded: R2D2_duckduckgo_84.jpg
Image 113 downloaded: R2D2_duckduckgo_85.jpg
Image 114 downloaded: R2D2_duckduckgo_86.jpg
Image 115 downloaded: R2D2_duckduckgo_87.png
Image 116 downloaded: R2D2_duckduckgo_88.jpg
Image 117 downloaded: R2D2_duckduckgo_89.jpg
Image 118 downloaded: R2D2_duckduckgo_90.jpg
Image 119 downloaded: R2D2_duckduckgo_91.jpg
Image 120 downloaded: R2D2_duckduckgo_92.jpg
Invalid response status for image 121: 406
Image 124 downloaded: R2D2_duckduckgo_93.jpg
Image 125 downloaded: R2D2_duckduckgo_94.jpg
Image 126 downloaded: R2D2_duckduckgo_95.jpg
Image 128 downloaded: R2D2_duckduckgo_96.png
Image 129 downloaded: R2D2_duckduckgo_97.jpg
Image 131 downloaded: R2D2_duckduckgo_98.jpg
Image 132 downloaded: R2D2_duckduckgo_99.jpg
Image 134 downloaded: R2D2_duckduckgo_100.jpg
Image 135 downloaded: R2D2_duckduckgo_101.jpg
Image 136 downloaded: R2D2_duckduckgo_102.jpg
Image 137 downloaded: R2D2_duckduckgo_103.jpeg
Image 138 downloaded: R2D2_duckduckgo_104.jpg
Image 139 downloaded: R2D2_duckduckgo_105.jpg
Image 140 downloaded: R2D2_duckduckgo_106.png
Image 141 downloaded: R2D2_duckduckgo_107.jpg
Image 142 downloaded: R2D2_duckduckgo_108.png
Image 143 downloaded: R2D2_duckduckgo_109.jpg
Image 144 downloaded: R2D2_duckduckgo_110.png
Image 145 downloaded: R2D2_duckduckgo_111.jpg
Image 147 downloaded: R2D2_duckduckgo_112.jpg
Image 149 downloaded: R2D2_duckduckgo_113.jpg
Image 2 downloaded: C3PO_duckduckgo_1.jpg
Image 3 downloaded: C3PO_duckduckgo_2.jpg
Image 5 downloaded: C3PO_duckduckgo_3.jpg
Image 7 downloaded: C3PO_duckduckgo_4.png
Image 8 downloaded: C3PO_duckduckgo_5.jpg
Invalid response status for image 9: 403
Image 12 downloaded: C3PO_duckduckgo_6.jpg
Image 13 downloaded: C3PO_duckduckgo_7.jpg
Image 15 downloaded: C3PO_duckduckgo_8.jpg
Invalid response status for image 16: 404
Image 17 downloaded: C3PO_duckduckgo_9.jpg
Invalid response status for image 18: 406
Invalid response status for image 19: 404
Image 20 downloaded: C3PO_duckduckgo_10.png
Image 21 downloaded: C3PO_duckduckgo_11.jpg
Image 22 downloaded: C3PO_duckduckgo_12.jpg
Image 23 downloaded: C3PO_duckduckgo_13.png
Invalid response status for image 24: 403
Image 25 downloaded: C3PO_duckduckgo_14.jpg
Image 27 downloaded: C3PO_duckduckgo_15.jpg
Image 28 downloaded: C3PO_duckduckgo_16.jpg
Image 29 downloaded: C3PO_duckduckgo_17.jpg
Image 32 downloaded: C3PO_duckduckgo_18.jpg
Image 34 downloaded: C3PO_duckduckgo_19.jpg
Image 35 downloaded: C3PO_duckduckgo_20.jpg
Image 36 downloaded: C3PO_duckduckgo_21.png
Image 37 downloaded: C3PO_duckduckgo_22.jpg
Image 39 downloaded: C3PO_duckduckgo_23.png
Image 40 downloaded: C3PO_duckduckgo_24.jpeg
Image 41 downloaded: C3PO_duckduckgo_25.jpg
Image 43 downloaded: C3PO_duckduckgo_26.jpg
Image 44 downloaded: C3PO_duckduckgo_27.png
Image 45 downloaded: C3PO_duckduckgo_28.jpg
Image 47 downloaded: C3PO_duckduckgo_29.jpg
Image 48 downloaded: C3PO_duckduckgo_30.jpg
Image 49 downloaded: C3PO_duckduckgo_31.png
Image 50 downloaded: C3PO_duckduckgo_32.jpg
Image 52 downloaded: C3PO_duckduckgo_33.jpg
Image 53 downloaded: C3PO_duckduckgo_34.jpg
Image 54 downloaded: C3PO_duckduckgo_35.jpg
Image 55 downloaded: C3PO_duckduckgo_36.jpg
Image 56 downloaded: C3PO_duckduckgo_37.jpg
Image 58 downloaded: C3PO_duckduckgo_38.jpg
Image 59 downloaded: C3PO_duckduckgo_39.jpg
Image 60 downloaded: C3PO_duckduckgo_40.png
Image 62 downloaded: C3PO_duckduckgo_41.jpg
Image 65 downloaded: C3PO_duckduckgo_42.jpg
Image 66 downloaded: C3PO_duckduckgo_43.jpg
Image 68 downloaded: C3PO_duckduckgo_44.jpg
Image 71 downloaded: C3PO_duckduckgo_45.jpg
Image 72 downloaded: C3PO_duckduckgo_46.jpg
Image 74 downloaded: C3PO_duckduckgo_47.jpg
Image 75 downloaded: C3PO_duckduckgo_48.jpg
Invalid response status for image 76: 404
Image 77 downloaded: C3PO_duckduckgo_49.jpg
Image 79 downloaded: C3PO_duckduckgo_50.jpg
Image 80 downloaded: C3PO_duckduckgo_51.jpeg
Image 81 downloaded: C3PO_duckduckgo_52.jpg
Image 82 downloaded: C3PO_duckduckgo_53.jpeg
Image 84 downloaded: C3PO_duckduckgo_54.jpg
Image 85 downloaded: C3PO_duckduckgo_55.jpg
Image 86 downloaded: C3PO_duckduckgo_56.jpg
Image 87 downloaded: C3PO_duckduckgo_57.jpeg
Image 89 downloaded: C3PO_duckduckgo_58.png
Image 90 downloaded: C3PO_duckduckgo_59.jpg
Image 91 downloaded: C3PO_duckduckgo_60.jpg
Image 92 downloaded: C3PO_duckduckgo_61.png
Image 93 downloaded: C3PO_duckduckgo_62.png
Image 94 downloaded: C3PO_duckduckgo_63.jpg
Image 95 downloaded: C3PO_duckduckgo_64.jpg
Image 96 downloaded: C3PO_duckduckgo_65.jpg
Image 98 downloaded: C3PO_duckduckgo_66.jpg
Image 100 downloaded: C3PO_duckduckgo_67.jpg
Image 101 downloaded: C3PO_duckduckgo_68.jpg
Image 103 downloaded: C3PO_duckduckgo_69.jpg
Image 104 downloaded: C3PO_duckduckgo_70.jpg
Image 105 downloaded: C3PO_duckduckgo_71.png
Image 106 downloaded: C3PO_duckduckgo_72.jpg
Image 107 downloaded: C3PO_duckduckgo_73.png
Image 108 downloaded: C3PO_duckduckgo_74.jpg
Image 109 downloaded: C3PO_duckduckgo_75.jpg
Image 110 downloaded: C3PO_duckduckgo_76.png
Image 112 downloaded: C3PO_duckduckgo_77.jpg
Image 113 downloaded: C3PO_duckduckgo_78.jpg
Image 114 downloaded: C3PO_duckduckgo_79.jpg
Image 115 downloaded: C3PO_duckduckgo_80.jpg
Image 116 downloaded: C3PO_duckduckgo_81.jpeg
Image 117 downloaded: C3PO_duckduckgo_82.jpg
Image 118 downloaded: C3PO_duckduckgo_83.jpg
Image 119 downloaded: C3PO_duckduckgo_84.jpg
Image 120 downloaded: C3PO_duckduckgo_85.jpg
Image 121 downloaded: C3PO_duckduckgo_86.jpg
Invalid response status for image 122: 403
Image 123 downloaded: C3PO_duckduckgo_87.jpg
Error downloading image 125:
HTTPSConnectionPool(host='news.entertainmentearth.com', port=443): Max retries
exceeded with url: /wp-content/uploads/2014/10/01-3.75-inch-epv_c3po.jpg (Caused
by SSLError(SSLCertVerificationError(1, "[SSL: CERTIFICATE_VERIFY_FAILED]
certificate verify failed: Hostname mismatch, certificate is not valid for
'news.entertainmentearth.com'. (_ssl.c:1006)")))
Image 126 downloaded: C3PO_duckduckgo_88.jpg
Image 127 downloaded: C3PO_duckduckgo_89.jpg
Image 128 downloaded: C3PO_duckduckgo_90.jpeg
Image 129 downloaded: C3PO_duckduckgo_91.jpg
Image 130 downloaded: C3PO_duckduckgo_92.jpg
Image 131 downloaded: C3PO_duckduckgo_93.png
Image 132 downloaded: C3PO_duckduckgo_94.jpg
Image 133 downloaded: C3PO_duckduckgo_95.png
Image 137 downloaded: C3PO_duckduckgo_96.jpg
Image 138 downloaded: C3PO_duckduckgo_97.jpg
Image 139 downloaded: C3PO_duckduckgo_98.jpg
Image 140 downloaded: C3PO_duckduckgo_99.jpg
Image 141 downloaded: C3PO_duckduckgo_100.jpg
Image 142 downloaded: C3PO_duckduckgo_101.jpg
Image 143 downloaded: C3PO_duckduckgo_102.jpg
Image 144 downloaded: C3PO_duckduckgo_103.jpg
Image 146 downloaded: C3PO_duckduckgo_104.jpg
Image 147 downloaded: C3PO_duckduckgo_105.jpg
Image 149 downloaded: C3PO_duckduckgo_106.jpg
Image 150 downloaded: C3PO_duckduckgo_107.jpg
```

Now the dataset for our program is complete.

1.3 EDA & Prep the data

1.3.1 EDA

Here I show how many images there are for each category. After that I use MatPlotLb to show some of the images.

[4]:
```python
import matplotlib.pyplot as plt
```

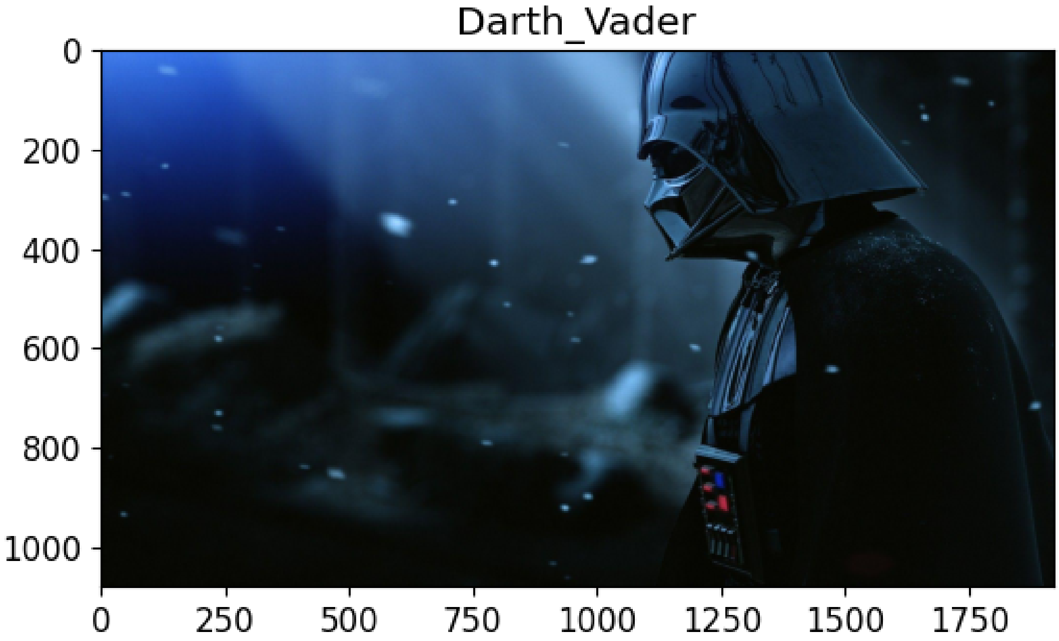
Now I want to validate that all images can be loaded. During the creation of this program, I have gotten some errors with training the model. With this extra check the errors are gone.

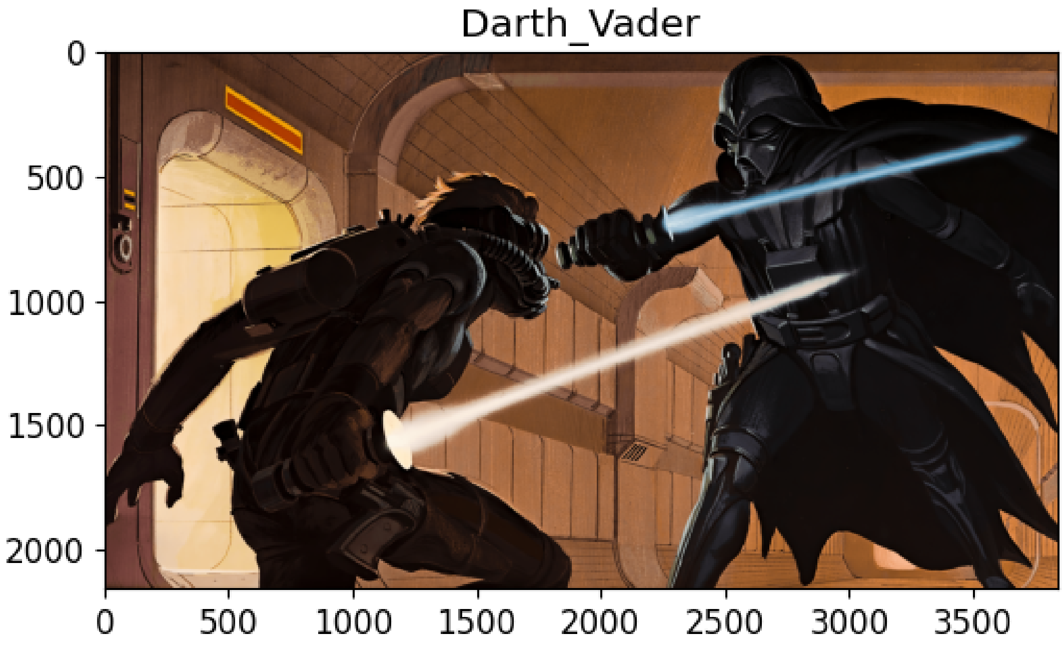
[8]:
```python
def check_image(img_path):
    # try to load images + make images RGBA
    try:
        img = Image.open(img_path)
        img.load()

        if img.mode != "RGBA":
            img = img.convert("RGBA")
        img.close()

    except (IOError, OSError) as e:
        print(f"Error loading image at path: {img_path}")
        if os.path.exists(img_path):
            os.remove(img_path)

#check every image
for category in category_names:

    category_path = os.path.join(data_dir, category)
    image_files = os.listdir(category_path)
    for image_file in image_files:
        full_path = category_path+"/"+image_file
        check_image(full_path)
```

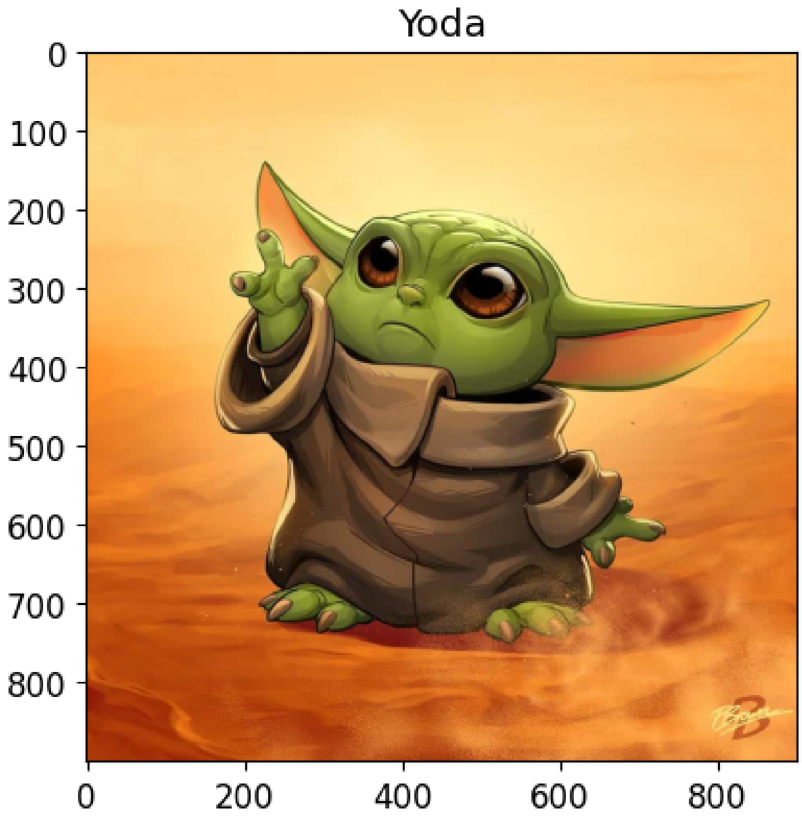
I count the amount of images that are left and display a few of them.

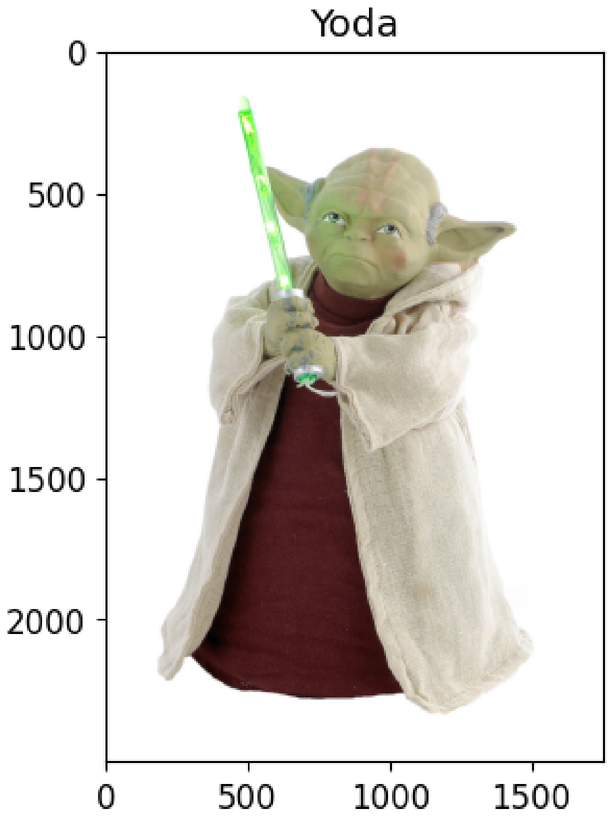
```
[9]: def count_images(category_names):

    directories= []
    #counts amount of images per directory/category
    for category in category_names:
        category_path = os.path.join(data_dir, category)
        num_images = len(os.listdir(category_path))
        print(f"Category: {category}, Number of Images: {num_images}")


        directories.append(category_path)

    display_images(category_names)

#shows 2 images per category
def display_images(category_names, amount_category_images=2):
    for category in category_names:


        category_path = os.path.join(data_dir, category)
        image_files = os.listdir(category_path)[:amount_category_images]
        for image_file in image_files:
            image_path = os.path.join(category_path, image_file)
            image = Image.open(image_path)
            plt.imshow(image)
            plt.title(category)
            plt.show()

count_images(category_names)
```

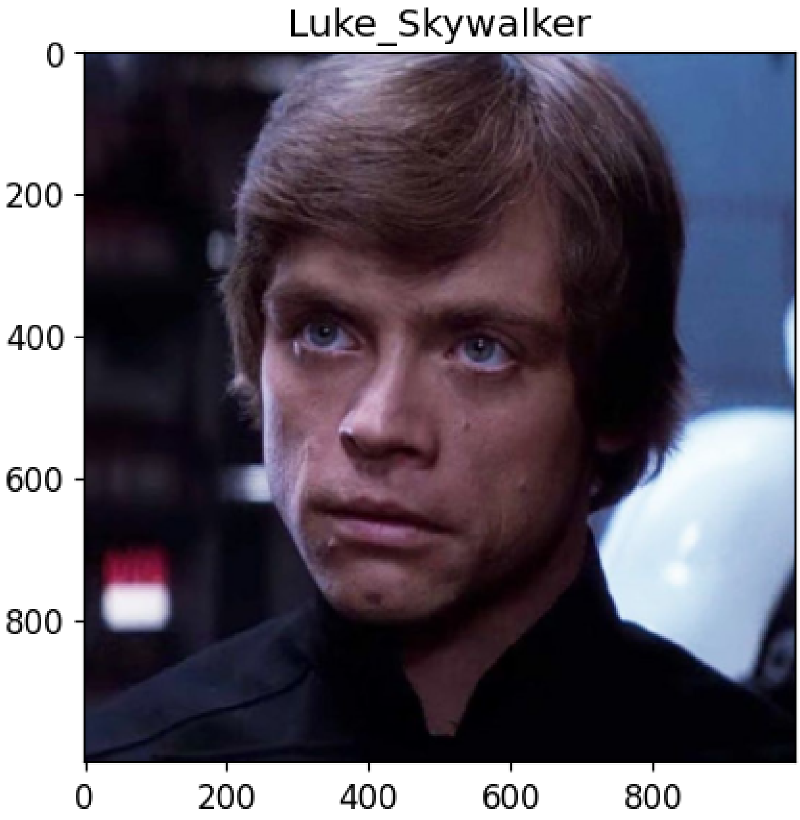
```
Category: Darth_Vader, Number of Images: 124
Category: Yoda, Number of Images: 134
Category: Luke_Skywalker, Number of Images: 114
Category: R2D2, Number of Images: 113
Category: C3PO, Number of Images: 106
```



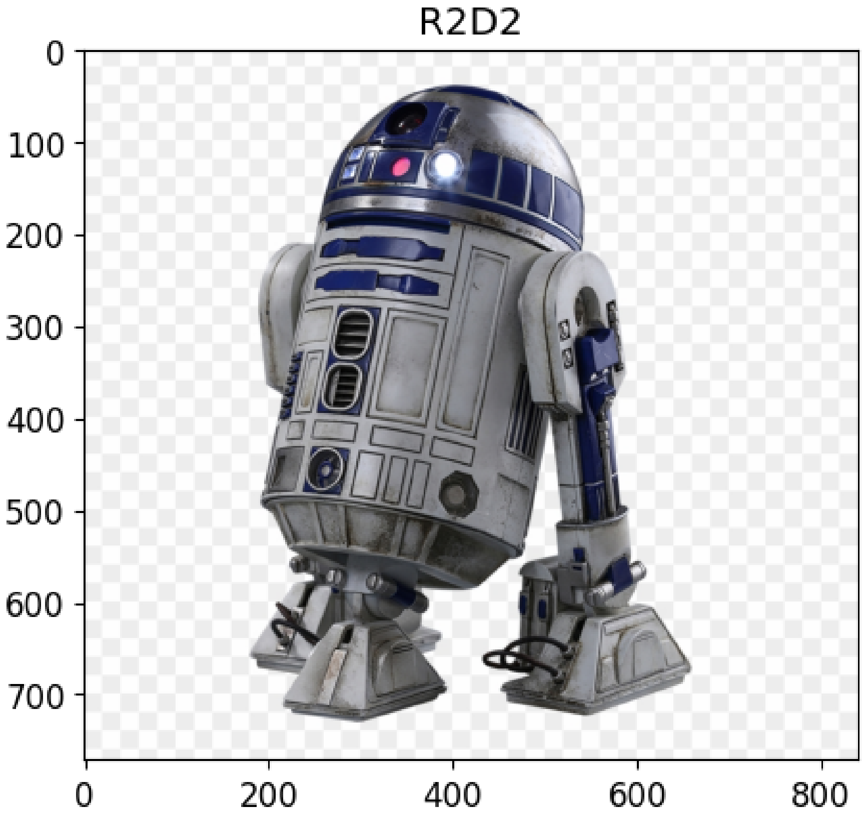

Darth_Vader
0
500
1000
1500
2000
0
500
1000
1500
2000
2500
3000
3500

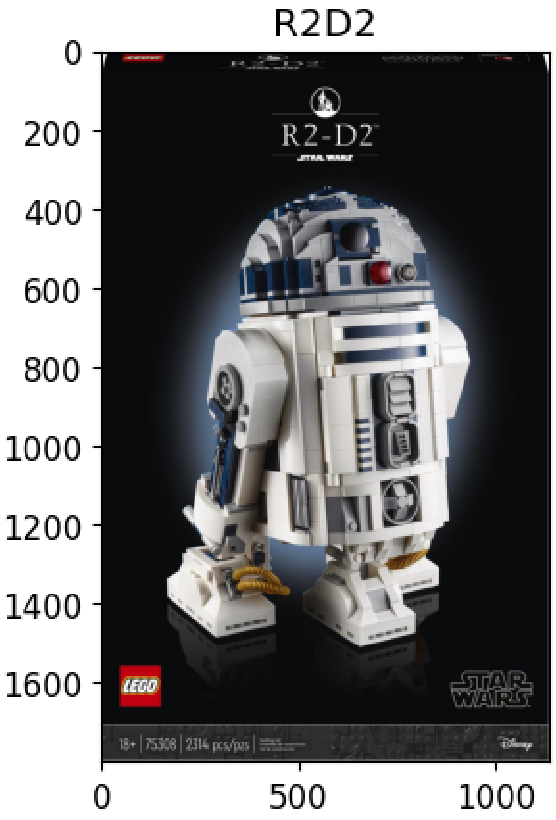

Yoda
0
100
200
300
400
500
600
700
800
0
200
400
600
800

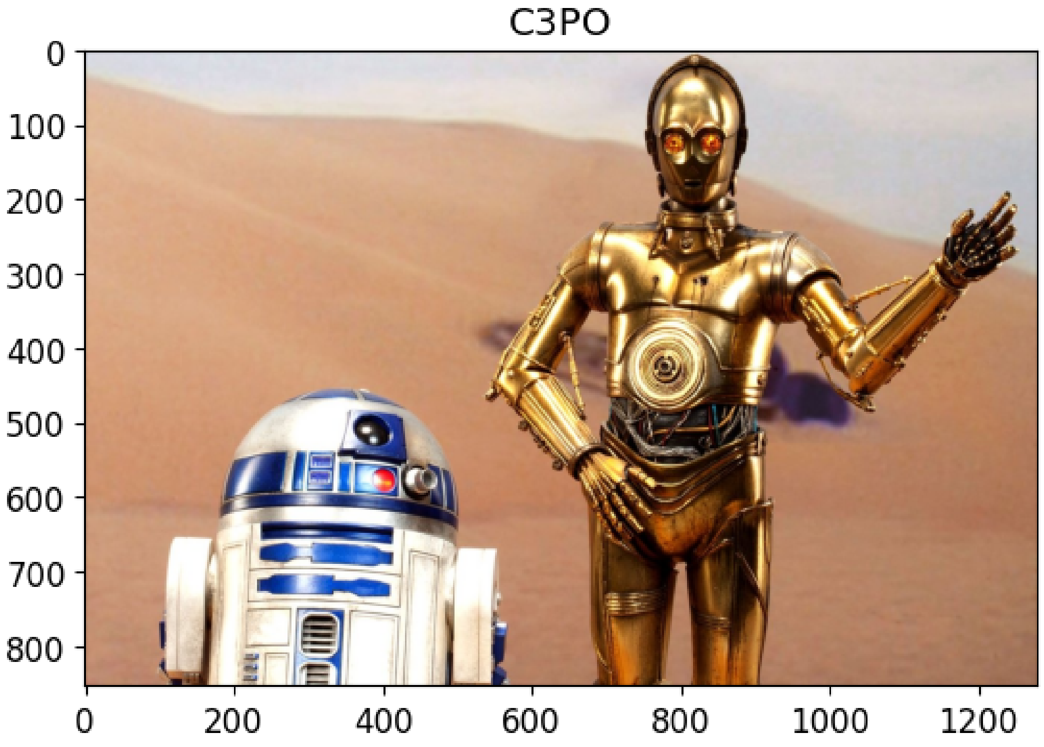

Yoda
0
500
1000
1500
2000
0
500
1000
1500

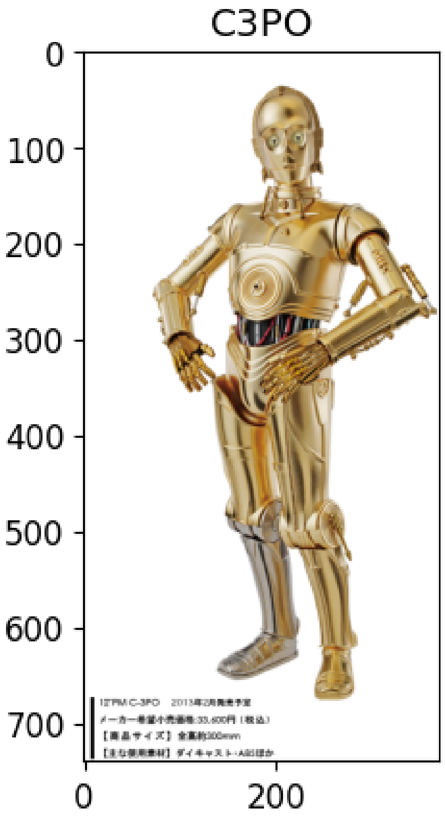

Luke_Skywalker
0
200
400
600
800
0
200
400
600
800

Luke_Skywalker
0
200
400
600
800
1000
0
200
400
600
800
STAR WARS
"LUKE SKYWALKER" REDESIGN


R2D2
0
100
200
300
400
500
600
700
0
200
400
600
800


R2D2
0
200
400
600
800
1000
1200
1400
1600
0
500
1000
R2-D2
STAR WARS
LEGO
STAR WARS
18+ | 75308 | 2314 pcs/pzs
Disney


C3PO
0
100
200
300
400
500
600
700
800
0
200
400
600
800
1000
1200



1.3.2 Prep the data

The images are in directories according to their category, not according to training and testing dataset. We can use DataFrames with the ImageDataGenerator (ImageDataGenerator, z.d.).This will however give us some problems with Teachable Machine. We don't know which images are going in which dataset so we will split the images according to directories.

[10]:
```
import os
import shutil
import random

def split_images(train_dir="./images/train", test_dir="./images/test",␣
 ↪split_ratio=0.9):
    os.makedirs(train_dir, exist_ok=True)
    os.makedirs(train_dir, exist_ok=True)

    for category in categories:
        category=category.replace(" ", "_")
        category_dir = os.path.join(data_dir, category)

        # Create train and test directories for each category
        train_class_dir = os.path.join(train_dir, category)
        os.makedirs(train_class_dir, exist_ok=True)

        test_class_dir = os.path.join(test_dir, category)
        os.makedirs(test_class_dir, exist_ok=True)

        files = os.listdir(category_dir)

        num_train_files = int(len(files) * split_ratio)

        random.shuffle(files)
        # put images in train and test directories based upon the split ratio
        for file in files[:num_train_files]:
            src_path = os.path.join(category_dir, file)
            dst_path = os.path.join(train_class_dir, file)
            shutil.move(src_path, dst_path)


        for file in files[num_train_files:]:
            src_path = os.path.join(category_dir, file)
            dst_path = os.path.join(test_class_dir, file)
            shutil.move(src_path, dst_path)




split_images()
```

I create datasets based on the different directories.

[11]:
```
from keras.preprocessing.image import ImageDataGenerator


def create_data_sets():

    train_val_datagen = ImageDataGenerator(validation_split=0.2,
                                   rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

    test_datagen = ImageDataGenerator(rescale = 1./255)


    train_dir ='./images/train'
    test_dir='./images/test'

    train_ds = train_val_datagen.flow_from_directory(
        directory=train_dir,
        subset='training',
        target_size = (64, 64),
        batch_size = 32,
        class_mode = 'categorical'
    )


    validation_ds = train_val_datagen.flow_from_directory(
        directory=train_dir,
        subset='validation',
        target_size = (64, 64),
        batch_size = 32,
        class_mode = 'categorical'
    )


    test_ds = test_datagen.flow_from_directory(
        directory=test_dir,
        target_size = (64, 64),
        batch_size = 32,
        class_mode = 'categorical'
    )
    return train_ds, test_ds, validation_ds
training_set, test_set, validation_set = create_data_sets()
```

```
WARNING:tensorflow:From
c:\Users\rafen\AppData\Local\Programs\Python\Python311\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.

Found 424 images belonging to 5 classes.
Found 105 images belonging to 5 classes.
Found 62 images belonging to 5 classes.
```

1.4 Design a CNN

[13]:
```
import tensorflow as tf
from tensorflow.keras import layers, models

# Set the number of classes
NUM_CLASSES = 1
```

```
# Create a sequential model with a list of layers
model = models.Sequential([
    layers.Conv2D(64, (3, 3), input_shape=(64, 64, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Dropout(0.25),

    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Dropout(0.25),

    layers.Flatten(),

    layers.Dense(256, activation='relu'),
    layers.Dropout(0.5),

    layers.Dense(NUM_CLASSES, activation='sigmoid')
])


model.compile(optimizer='adam', loss='binary_crossentropy',␣
 ↪metrics=['accuracy'])


model.summary()
```

```
WARNING:tensorflow:From
c:\Users\rafen\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\backend.py:873: The name tf.get_default_graph is deprecated.
Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From
c:\Users\rafen\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\layers\pooling\max_pooling2d.py:161: The name tf.nn.max_pool
is deprecated. Please use tf.nn.max_pool2d instead.

WARNING:tensorflow:From
c:\Users\rafen\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\optimizers\__init__.py:309: The name tf.train.Optimizer is
deprecated. Please use tf.compat.v1.train.Optimizer instead.

Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 64)        1792


 max_pooling2d (MaxPooling2  (None, 31, 31, 64)        0
 D)

 dropout (Dropout)           (None, 31, 31, 64)        0

 conv2d_1 (Conv2D)           (None, 29, 29, 128)       73856

 max_pooling2d_1 (MaxPoolin  (None, 14, 14, 128)       0
 g2D)

 dropout_1 (Dropout)         (None, 14, 14, 128)       0

 flatten (Flatten)           (None, 25088)             0

 dense (Dense)               (None, 256)               6422784

 dropout_2 (Dropout)         (None, 256)               0

 dense_1 (Dense)             (None, 1)                 257

=================================================================
Total params: 6498689 (24.79 MB)
Trainable params: 6498689 (24.79 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

1.5 Train model

[14]:
```
history = model.fit(training_set,
                validation_data = validation_set,
                steps_per_epoch = 10,
                epochs = 20
                )
```

```
Epoch 1/20
WARNING:tensorflow:From
c:\Users\rafen\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue
is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From
c:\Users\rafen\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\engine\base_layer_utils.py:384: The name
tf.executing_eagerly_outside_functions is deprecated. Please use
tf.compat.v1.executing_eagerly_outside_functions instead.


c:\Users\rafen\AppData\Local\Programs\Python\Python311\Lib\site-
packages\PIL\Image.py:970: UserWarning: Palette images with Transparency
expressed in bytes should be converted to RGBA images
```

```
  warnings.warn(
10/10 [==============================] - 11s 969ms/step - loss: 0.9128 -
accuracy: 0.7270 - val_loss: 0.6001 - val_accuracy: 0.8000
Epoch 2/20
10/10 [==============================] - 9s 961ms/step - loss: 0.5377 -
accuracy: 0.8000 - val_loss: 0.5607 - val_accuracy: 0.8000
Epoch 3/20
10/10 [==============================] - 9s 923ms/step - loss: 0.5250 -
accuracy: 0.8000 - val_loss: 0.5380 - val_accuracy: 0.8000
Epoch 4/20
10/10 [==============================] - 9s 851ms/step - loss: 0.5119 -
accuracy: 0.8000 - val_loss: 0.5430 - val_accuracy: 0.8000
Epoch 5/20
10/10 [==============================] - 9s 937ms/step - loss: 0.5099 -
accuracy: 0.8000 - val_loss: 0.5574 - val_accuracy: 0.8000
Epoch 6/20
10/10 [==============================] - 10s 974ms/step - loss: 0.5076 -
accuracy: 0.8000 - val_loss: 0.5447 - val_accuracy: 0.8000
Epoch 7/20
10/10 [==============================] - 10s 952ms/step - loss: 0.5062 -
accuracy: 0.8000 - val_loss: 0.5414 - val_accuracy: 0.8000
Epoch 8/20
10/10 [==============================] - 9s 894ms/step - loss: 0.5061 -
accuracy: 0.8000 - val_loss: 0.5565 - val_accuracy: 0.8000
Epoch 9/20
10/10 [==============================] - 9s 927ms/step - loss: 0.5055 -
accuracy: 0.8000 - val_loss: 0.5335 - val_accuracy: 0.8000
Epoch 10/20
10/10 [==============================] - 9s 953ms/step - loss: 0.5070 -
accuracy: 0.8000 - val_loss: 0.5382 - val_accuracy: 0.8000
Epoch 11/20
10/10 [==============================] - 9s 916ms/step - loss: 0.5061 -
accuracy: 0.8000 - val_loss: 0.5443 - val_accuracy: 0.8000
Epoch 12/20
10/10 [==============================] - 10s 1s/step - loss: 0.5052 - accuracy:
0.8000 - val_loss: 0.5367 - val_accuracy: 0.8000
Epoch 13/20
10/10 [==============================] - 10s 1s/step - loss: 0.5048 - accuracy:
0.8000 - val_loss: 0.5348 - val_accuracy: 0.8000
Epoch 14/20
10/10 [==============================] - 10s 972ms/step - loss: 0.5051 -
accuracy: 0.8000 - val_loss: 0.5352 - val_accuracy: 0.8000
Epoch 15/20
10/10 [==============================] - 10s 981ms/step - loss: 0.5048 -
accuracy: 0.8000 - val_loss: 0.5371 - val_accuracy: 0.8000
Epoch 16/20
10/10 [==============================] - 9s 928ms/step - loss: 0.5052 -
```

```
accuracy: 0.8000 - val_loss: 0.5381 - val_accuracy: 0.8000
Epoch 17/20
10/10 [==============================] - 9s 1s/step - loss: 0.5048 - accuracy:
0.8000 - val_loss: 0.5276 - val_accuracy: 0.8000
Epoch 18/20
10/10 [==============================] - 10s 1s/step - loss: 0.5041 - accuracy:
0.8000 - val_loss: 0.5326 - val_accuracy: 0.8000
Epoch 19/20
10/10 [==============================] - 9s 951ms/step - loss: 0.5051 -
accuracy: 0.8000 - val_loss: 0.5276 - val_accuracy: 0.8000
Epoch 20/20
10/10 [==============================] - 9s 903ms/step - loss: 0.5053 -
accuracy: 0.8000 - val_loss: 0.5339 - val_accuracy: 0.8000
```

[18]:

```python
from keras.models import load_model
model.save("models/modelStarWars.tf")
model = load_model("models/modelStarWars.tf")
```

```
INFO:tensorflow:Assets written to: models/modelStarWars.tf\assets
```

```
INFO:tensorflow:Assets written to: models/modelStarWars.tf\assets
```

```
WARNING:tensorflow:From
c:\Users\rafen\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\saving\legacy\saved_model\load.py:107: The name
tf.gfile.Exists is deprecated. Please use tf.io.gfile.exists instead.


WARNING:tensorflow:From
c:\Users\rafen\AppData\Local\Programs\Python\Python311\Lib\site-
packages\keras\src\saving\legacy\saved_model\load.py:107: The name
tf.gfile.Exists is deprecated. Please use tf.io.gfile.exists instead.
```

1.6 Comparison

1.6.1 Teachable Machine

First I will load in the Teachable Machine model. I have created this using the same training set. The code is provided on the Teachable Machine website when exporting the model.

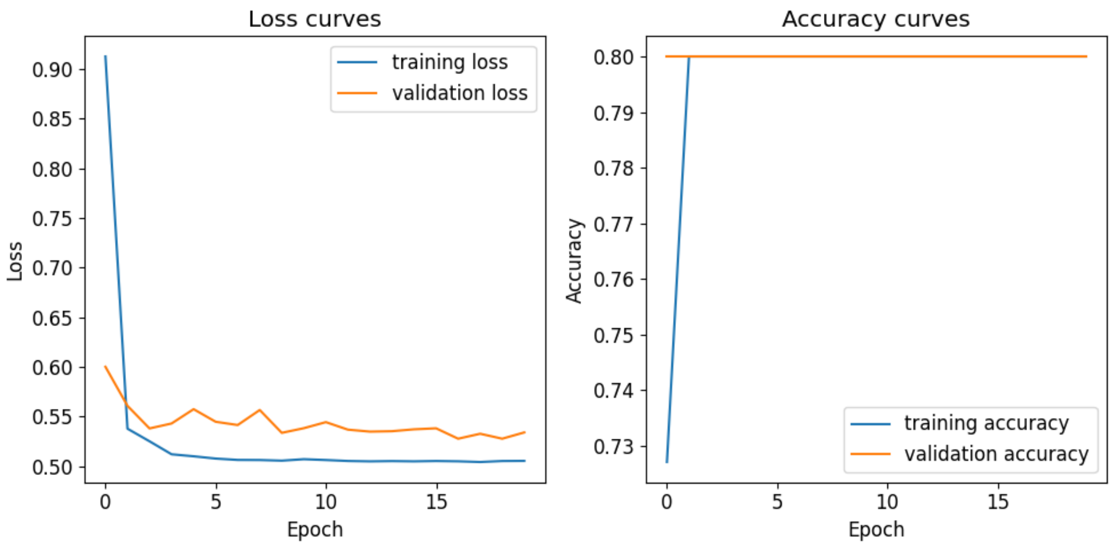
[24]:

```python
 # TensorFlow is required for Keras to work
from PIL import Image, ImageOps  # Install pillow instead of PIL
import numpy as np

def setup_tm():
    # Disable scientific notation for clarity
    np.set_printoptions(suppress=True)

    # Load the model
    tm_model = load_model("./models/teachable_machine/keras_model.h5",␣
 ↪compile=False)

    tm_model.compile(optimizer='adam', loss='binary_crossentropy',␣
 ↪metrics=['accuracy'])
    return tm_model
tm_model = setup_tm()
```

Here you can see 2 figures of the loss and the accuracy of the history of the model.

[25]:

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10,5))


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()


ax2.plot(history.history['accuracy'], label='training accuracy')
ax2.plot(history.history['val_accuracy'], label='validation accuracy')
ax2.set_title('Accuracy curves')
ax2.set_xlabel('Epoch')
ax2.set_ylabel('Accuracy')
ax2.legend()


fig.tight_layout()


plt.show()
```



1.6.2 Testing

Now I will use the test dataset to test both the Teachable Machine and my own model.

[31]:

```python
test_loss, test_acc = model.evaluate(test_set)
print('My model, test accuracy :', test_acc)
print('My model, test loss :', test_loss)

test_loss_tm, test_acc_tm = tm_model.evaluate(test_set)
print('Teachable Machine, test accuracy:', test_acc_tm)
print('Teachable Machine, test loss:', test_loss_tm)



data = {
    'Accuracy': [test_acc, test_acc_tm],
    'Loss': [test_loss, test_loss_tm]
}

df = pd.DataFrame(data, index=['My Model', 'Teachable Machine'])
```

```
2/2 [==============================] - 2s 940ms/step - loss: 0.5305 - accuracy:
0.8000
My model, test accuracy : 0.7999999523162842
My model, test loss : 0.5305071473121643
2/2 [==============================] - 2s 580ms/step - loss: 1.5608 - accuracy:
0.2581
Teachable Machine, test accuracy: 0.25806450843811035
Teachable Machine, test loss: 1.56084144115448
```

As you can see, my model scores has a higher accuracy than the Teachable Machine by Google. I am perplexed by this, because i don't think the numbers are accurate. 25% accuracy is very low, especially for the Teachable Machine which is a good Deep Learning Algorithm.

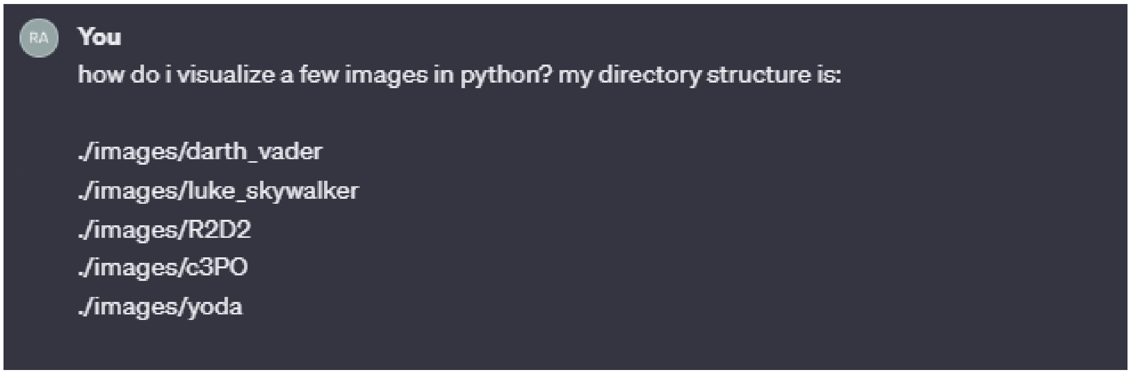
1.7 Streamlit

This is what i show on Streamlit. On streamlit, the program will not scrape images and create a new model. The model that we have saved in this document will be added to streamlit.

[32]:
```
import streamlit as st
st.header('Raf Engelen - r0901812 - 3APP01', divider='gray')
st.title("Task 3 ML: Comparing Deep Learning model to Teachable Machine")
st.write('''As you can see, my model scores has a higher accuracy than the␣
 ↪Teachable Machine by Google. I am perplexed by this, because i don't think␣
 ↪the numbers are accurate. 25% accuracy is very low, especially for the␣
 ↪Teachable Machine which is a good Deep Learning Algorithm.''')
st.write(df)
```

1.8 GenAI tools

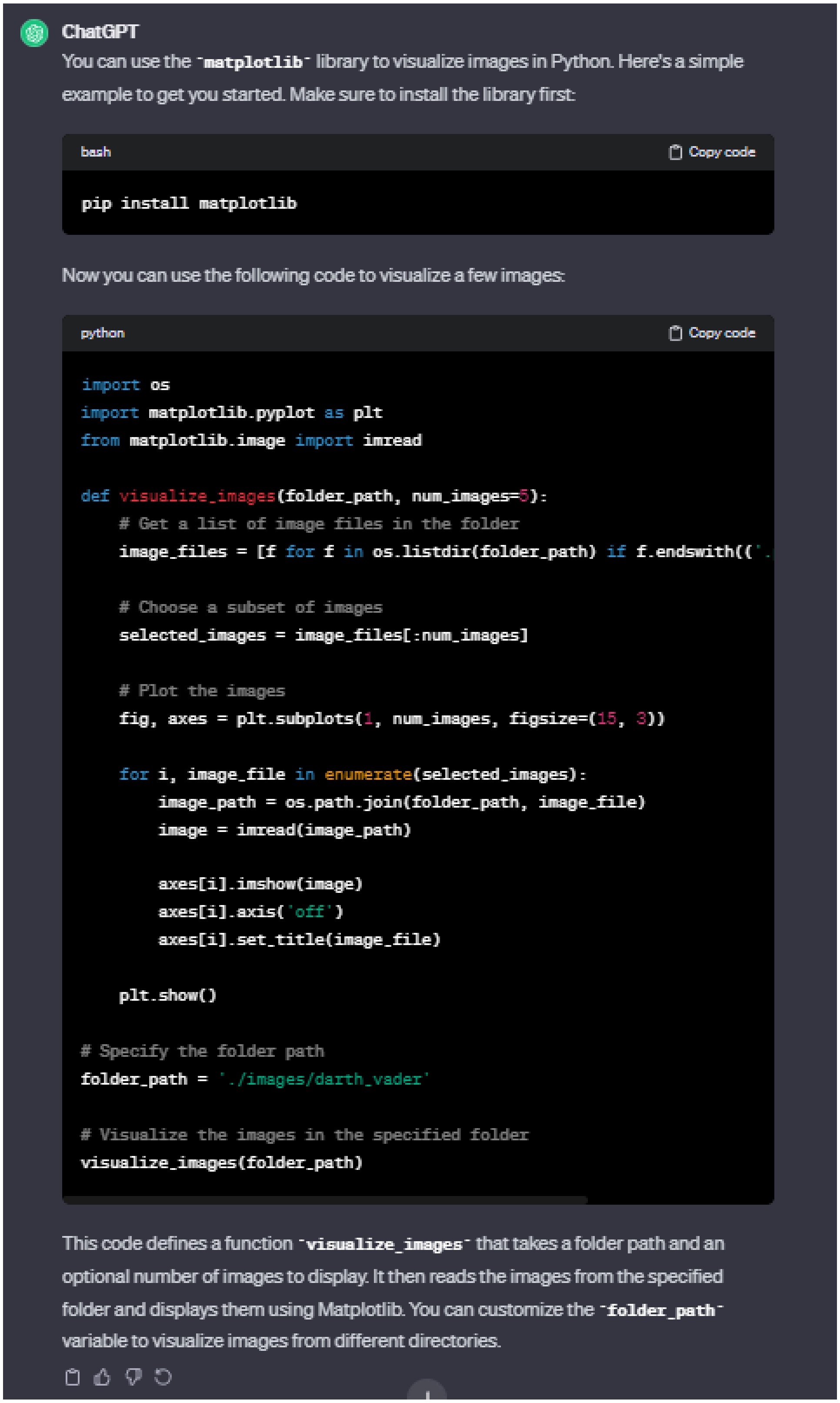
I needed a quick reminder how I could show images in python.



You

how do i visualize a few images in python? my directory structure is:

./images/darth_vader
./images/luke_skywalker
./images/R2D2
./images/c3PO
./images/yoda



You can use the `matplotlib` library to visualize images in Python. Here's a simple example to get you started. Make sure to install the library first:

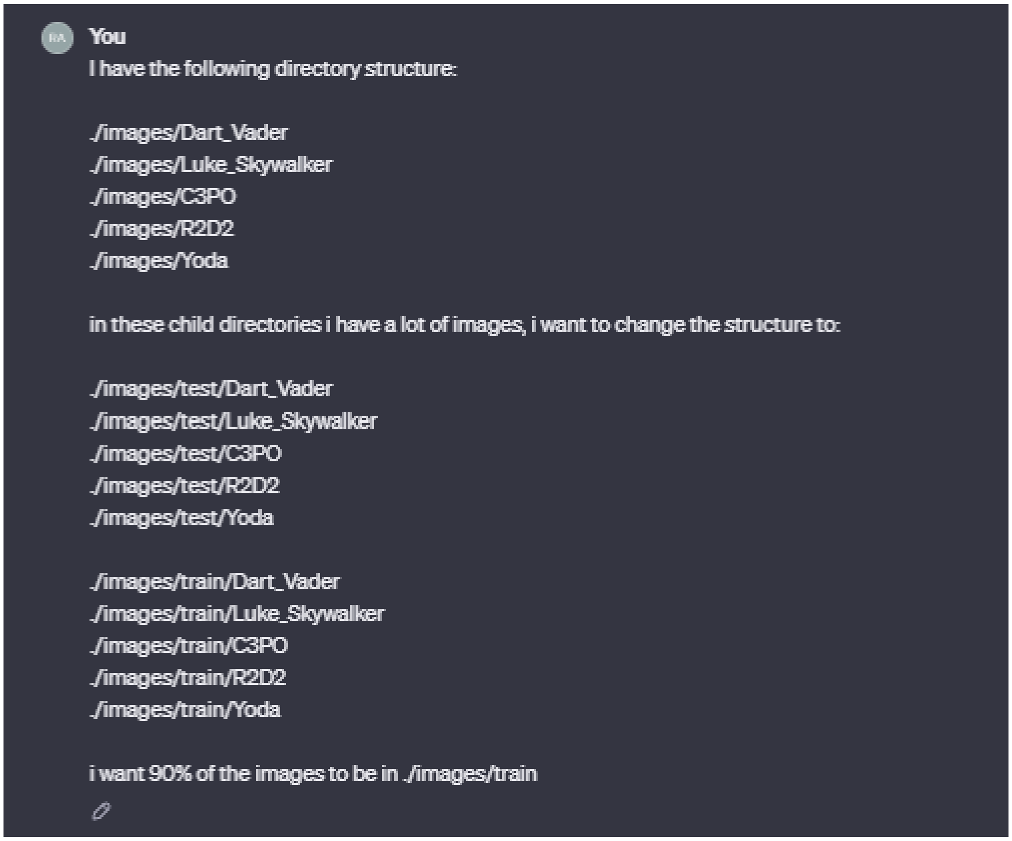
```bash
pip install matplotlib
```

Now you can use the following code to visualize a few images:

```python
import os
import matplotlib.pyplot as plt
from matplotlib.image import imread

def visualize_images(folder_path, num_images=5):
    # Get a list of image files in the folder
    image_files = [f for f in os.listdir(folder_path) if f.endswith(('.

    # Choose a subset of images
    selected_images = image_files[:num_images]

    # Plot the images
    fig, axes = plt.subplots(1, num_images, figsize=(15, 3))

    for i, image_file in enumerate(selected_images):
        image_path = os.path.join(folder_path, image_file)
        image = imread(image_path)

        axes[i].imshow(image)
        axes[i].axis('off')
        axes[i].set_title(image_file)

    plt.show()

# Specify the folder path
folder_path = './images/darth_vader'

# Visualize the images in the specified folder
visualize_images(folder_path)
```

This code defines a function `visualize_images` that takes a folder path and an optional number of images to display. It then reads the images from the specified folder and displays them using Matplotlib. You can customize the `folder_path` variable to visualize images from different directories.

After putting all the images in directories per category, I had to split them up in a training set and a testing set. This was no problem with the current structure. But the problem was that i didn't know which images were used for training, so i couldn't train the Teachable Machine properly. That's why i wanted to move the images to a test and a train directory.



You

I have the following directory structure:

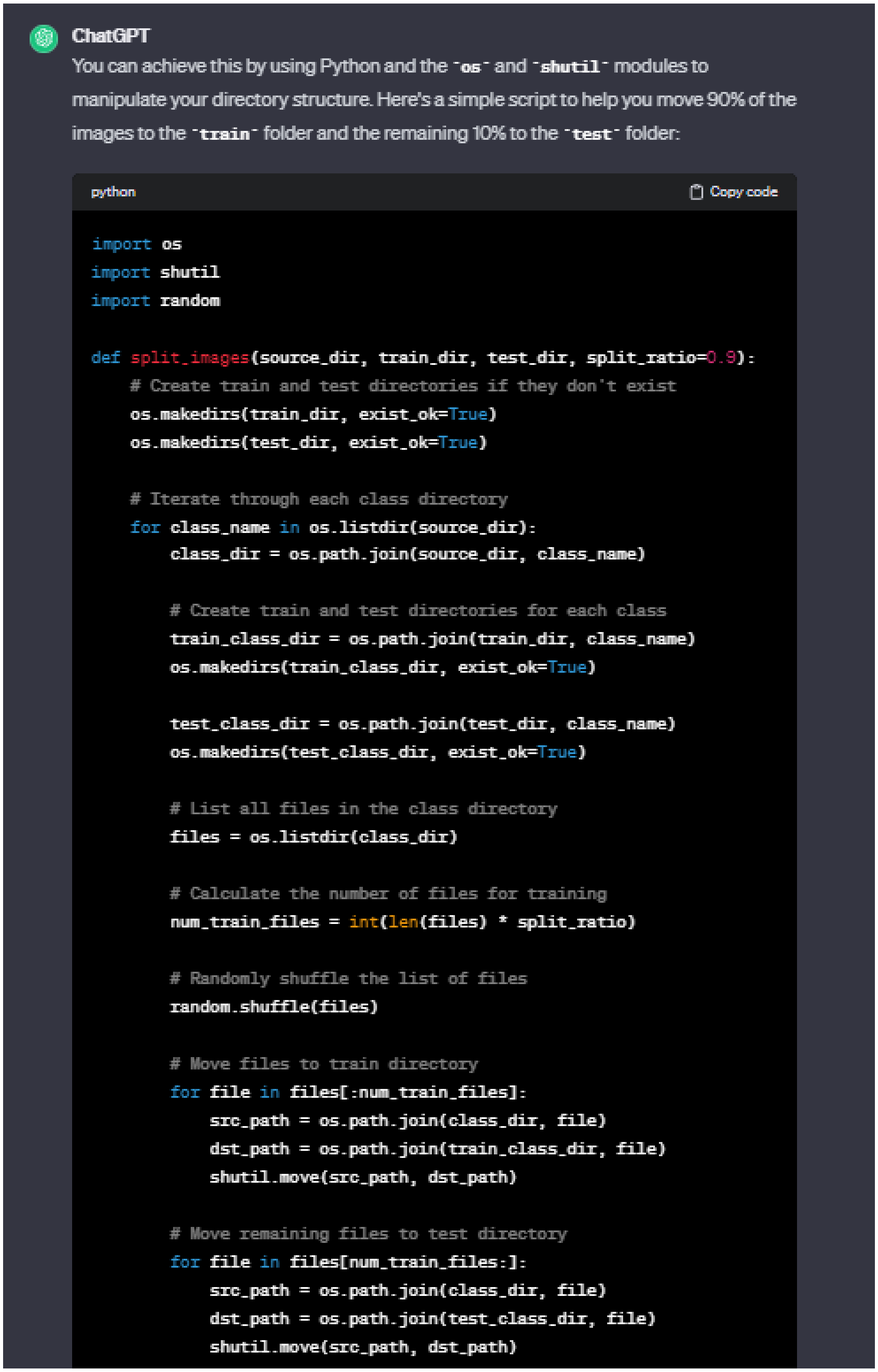
./images/Dart_Vader
./images/Luke_Skywalker
./images/C3PO
./images/R2D2
./images/Yoda

in these child directories i have a lot of images, i want to change the structure to:

./images/test/Dart_Vader
./images/test/Luke_Skywalker
./images/test/C3PO
./images/test/R2D2
./images/test/Yoda

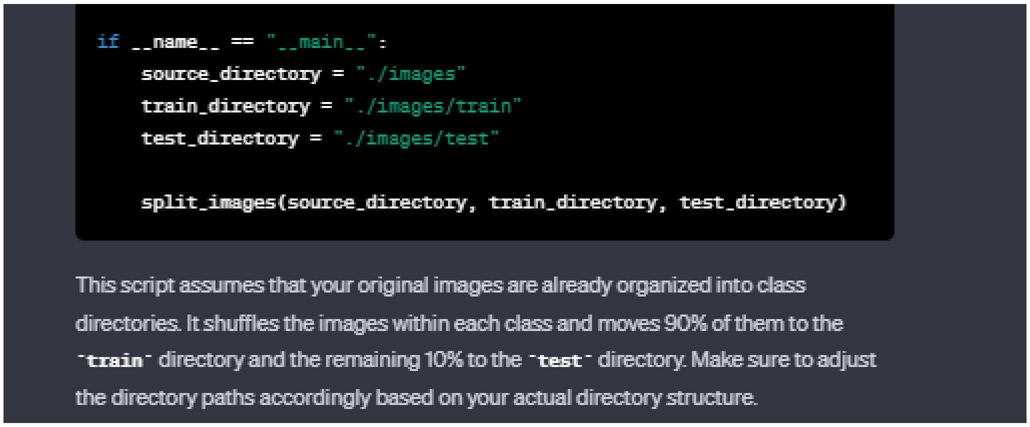
./images/train/Dart_Vader
./images/train/Luke_Skywalker
./images/train/C3PO
./images/train/R2D2
./images/train/Yoda

i want 90% of the images to be in ./images/train



You can achieve this by using Python and the `os` and `shutil` modules to manipulate your directory structure. Here's a simple script to help you move 90% of the images to the `train` folder and the remaining 10% to the `test` folder:

```python
import os
import shutil
import random

def split_images(source_dir, train_dir, test_dir, split_ratio=0.9):
    # Create train and test directories if they don't exist
    os.makedirs(train_dir, exist_ok=True)
    os.makedirs(test_dir, exist_ok=True)

    # Iterate through each class directory
    for class_name in os.listdir(source_dir):
        class_dir = os.path.join(source_dir, class_name)

        # Create train and test directories for each class
        train_class_dir = os.path.join(train_dir, class_name)
        os.makedirs(train_class_dir, exist_ok=True)

        test_class_dir = os.path.join(test_dir, class_name)
        os.makedirs(test_class_dir, exist_ok=True)

        # List all files in the class directory
        files = os.listdir(class_dir)

        # Calculate the number of files for training
        num_train_files = int(len(files) * split_ratio)

        # Randomly shuffle the list of files
        random.shuffle(files)

        # Move files to train directory
        for file in files[:num_train_files]:
            src_path = os.path.join(class_dir, file)
            dst_path = os.path.join(train_class_dir, file)
            shutil.move(src_path, dst_path)

        # Move remaining files to test directory
        for file in files[num_train_files:]:
            src_path = os.path.join(class_dir, file)
            dst_path = os.path.join(test_class_dir, file)
            shutil.move(src_path, dst_path)
```



```
if __name__ == "__main__":
    source_directory = "./images"
    train_directory = "./images/train"
    test_directory = "./images/test"

    split_images(source_directory, train_directory, test_directory)
```

This script assumes that your original images are already organized into class directories. It shuffles the images within each class and moves 90% of them to the `train` directory and the remaining 10% to the `test` directory. Make sure to adjust the directory paths accordingly based on your actual directory structure.

1.9 Literatuurlijst

ImageDataGenerator. (z.d.). tensorflow. https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/i

Teachable machine. (z.d.). https://teachablemachine.withgoogle.com/