

Matt Morrow

CIS5898

Second Progress Report

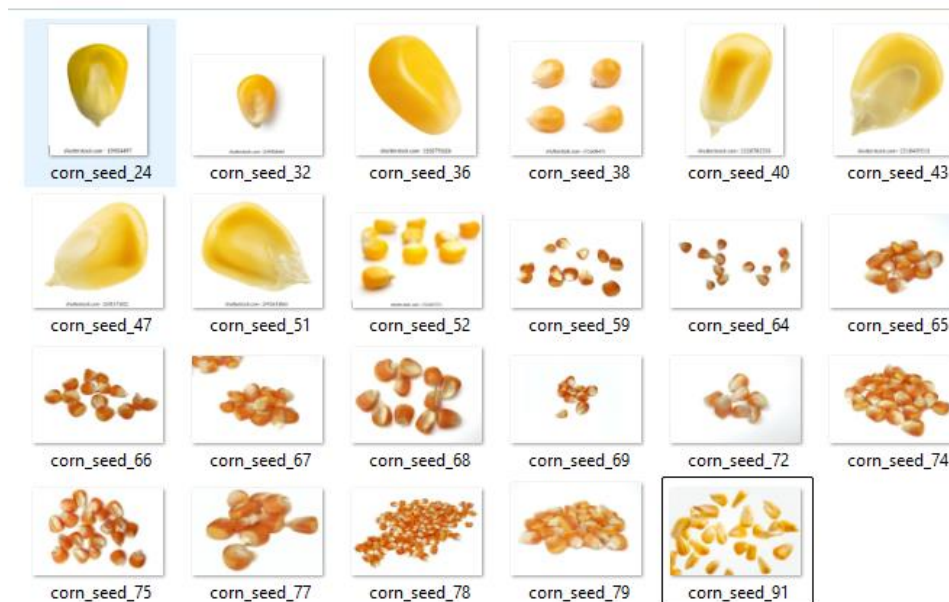
This period of the project's development focused on acquiring initial data and ensuring it can be uploaded/accessed from the Azure Database. A specialized program was needed and thus developed to perform web crawling for the purpose of collecting seed data. This custom crawler successfully retrieved images from vecteezy.com and stored them in a designated local folder for further processing and analysis.

Details of how the program works:

This code in the program automates the process of web scraping and downloading seed images using Selenium and BeautifulSoup. It starts by initializing a Selenium WebDriver with Chrome, configuring it to ignore SSL and certificate errors. The script then constructs a URL based on a specified seed type and navigates to the vecteezy.com search results page for relevant seed images.

Once the page is loaded, BeautifulSoup is used to parse the HTML content and extract all image elements (img tags). The code identifies and collects URLs of the images by checking attributes like src, data-src, and srcset, filtering for valid image formats such as .jpg and .png. After gathering the image URLs, the script creates a local folder named after the seed type and proceeds to download each image, saving them in the folder with a sequential naming convention. If an image fails to download, an error message is printed, and the script continues downloading the remaining images. Upon completion, the script outputs the total number of successfully downloaded images.

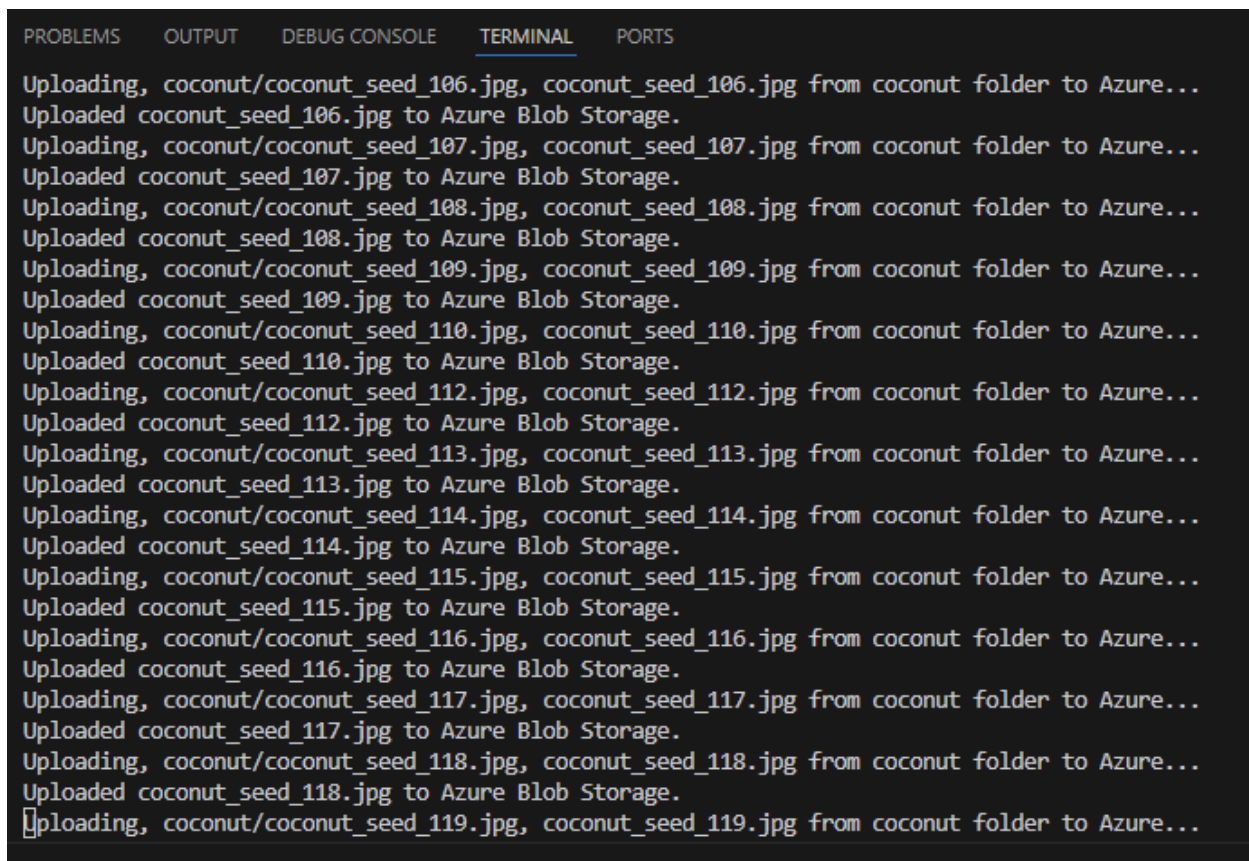
Those downloaded images show up on the local folder as such:



Then from the local folder the files are imported to the Azure database using the `upload_image` function.

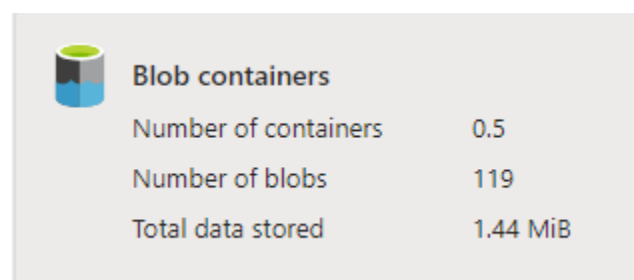
```
def upload_image(self, blob_name):  
    upload_image_to_azure(self.connection_string, self.container_name, blob_name)
```


The following screenshot illustrates that the image uploading functionality is performing as expected in the primary SeedID program, confirming that the process is operational and successfully handling uploads:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
Uploading, coconut/coconut_seed_106.jpg, coconut_seed_106.jpg from coconut folder to Azure...  
Uploaded coconut_seed_106.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_107.jpg, coconut_seed_107.jpg from coconut folder to Azure...  
Uploaded coconut_seed_107.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_108.jpg, coconut_seed_108.jpg from coconut folder to Azure...  
Uploaded coconut_seed_108.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_109.jpg, coconut_seed_109.jpg from coconut folder to Azure...  
Uploaded coconut_seed_109.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_110.jpg, coconut_seed_110.jpg from coconut folder to Azure...  
Uploaded coconut_seed_110.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_112.jpg, coconut_seed_112.jpg from coconut folder to Azure...  
Uploaded coconut_seed_112.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_113.jpg, coconut_seed_113.jpg from coconut folder to Azure...  
Uploaded coconut_seed_113.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_114.jpg, coconut_seed_114.jpg from coconut folder to Azure...  
Uploaded coconut_seed_114.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_115.jpg, coconut_seed_115.jpg from coconut folder to Azure...  
Uploaded coconut_seed_115.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_116.jpg, coconut_seed_116.jpg from coconut folder to Azure...  
Uploaded coconut_seed_116.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_117.jpg, coconut_seed_117.jpg from coconut folder to Azure...  
Uploaded coconut_seed_117.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_118.jpg, coconut_seed_118.jpg from coconut folder to Azure...  
Uploaded coconut_seed_118.jpg to Azure Blob Storage.  
Uploading, coconut/coconut_seed_119.jpg, coconut_seed_119.jpg from coconut folder to Azure...
```

So far 119 images have been uploaded for 4 different seeds that have been selected to be the initial run. Those seeds are sunflower, corn, wheat, and coconut.



	Blob containers	
Number of containers	0.5	
Number of blobs	119	
Total data stored	1.44 MiB	

Changes to the initial proposal/plan:

I opted to use Visual Studio Code over PyCharm due to its superior integration with Azure services. This choice enhances workflow efficiency, particularly in managing cloud-based resources and deployments, while also providing more seamless connectivity for development and debugging in Azure environments.