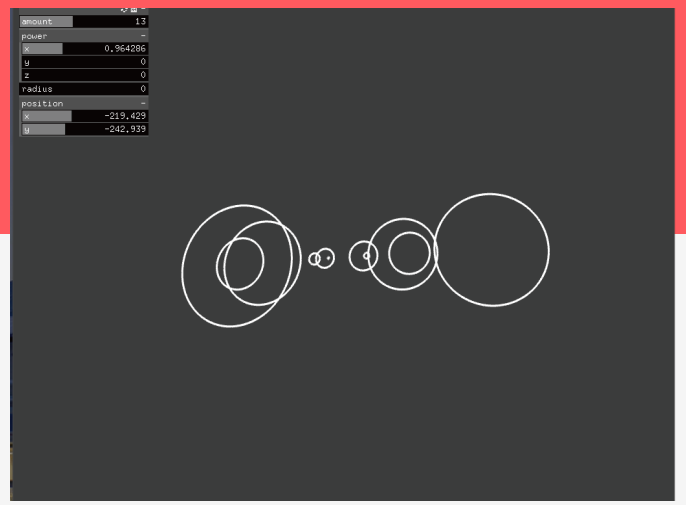
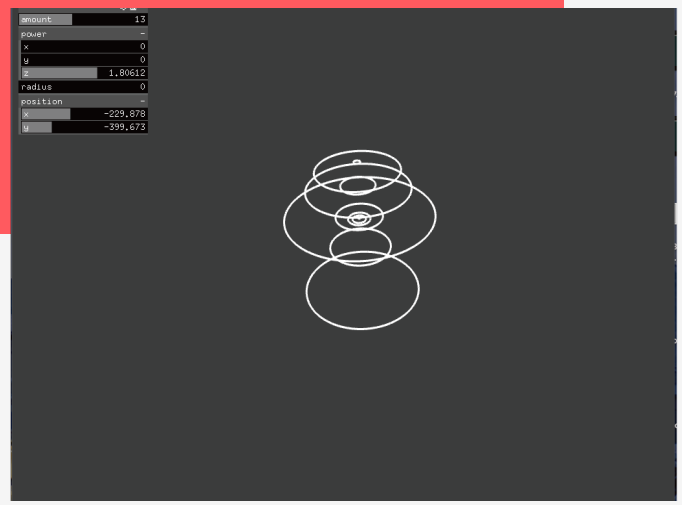
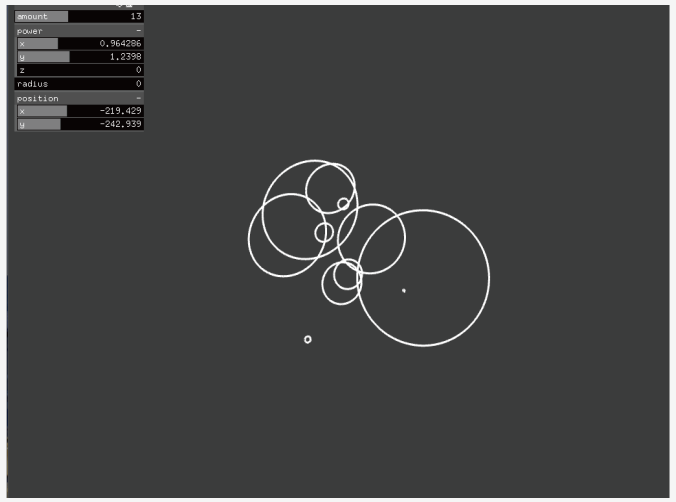
MSc Coding 2 Lab Work |

Yuchen Tan 20032211

Circle Loop✨







The name of this project is Circle Loop.

For this exercise I mainly modified last semester's 3D graphics assignment and added modifiers to modify the number of graphics and their orientation in the x, y, and z-axis in real time. It is worth mentioning that because I put in the z-axis, I can convert flat 2D graphics to 3D when using the mouse.

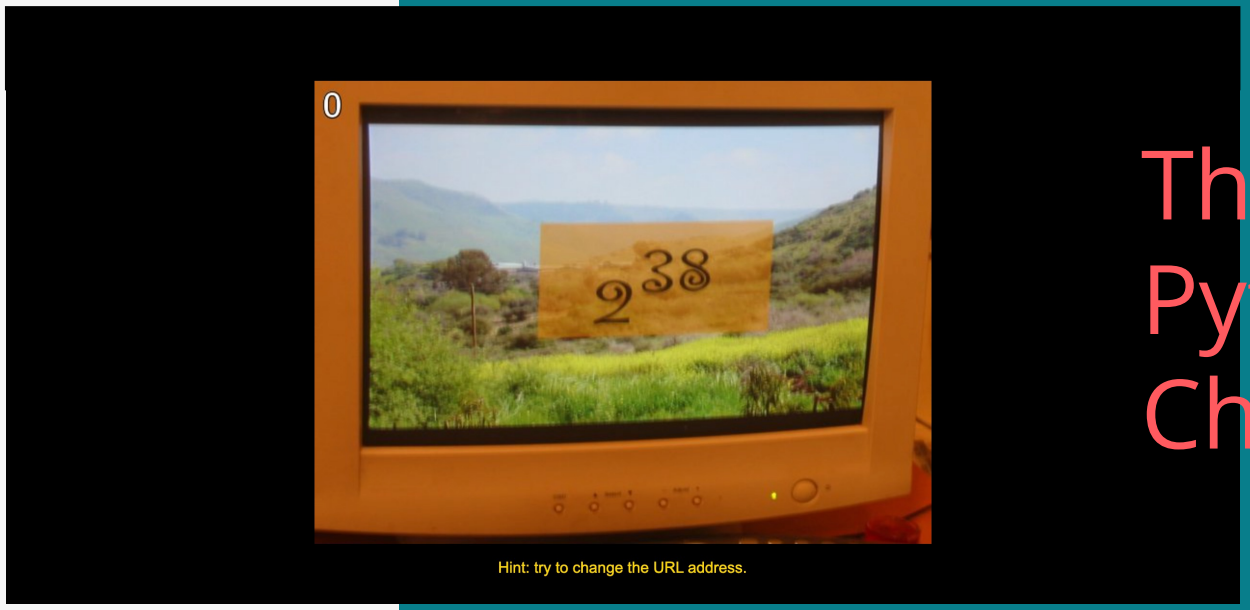
Circle Loop✨

This challenge has a total of 33 levels, each level has a picture & file & hint & nothing, need logic, imagination and write some scripts in python to get it done, and finally get a new url address, is the address of the next level.

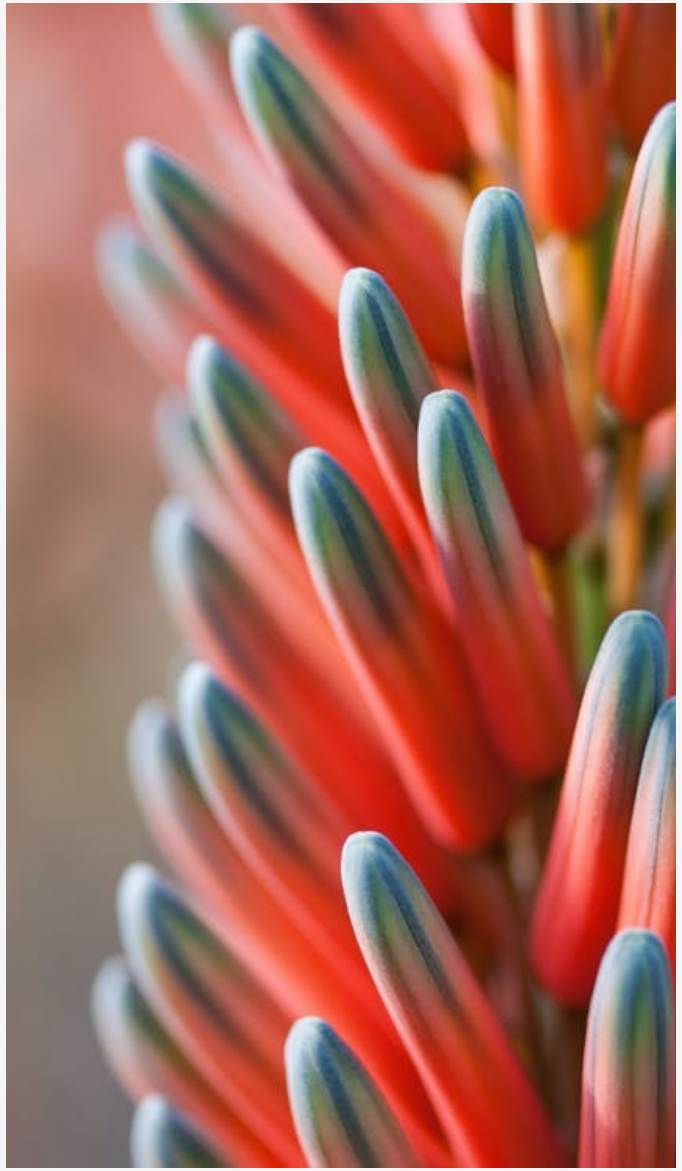
It is very helpful to improve your programming skills.

It requires good observation skills and logical thinking, and of course, imagination.

I found it interesting after I got the answer



The Python Challenge!



Current knowledge Covered :

Module: string

partitioning, indexing,
common operation functions,
mapping

Module: urllib

Access & parse

Module: pickle

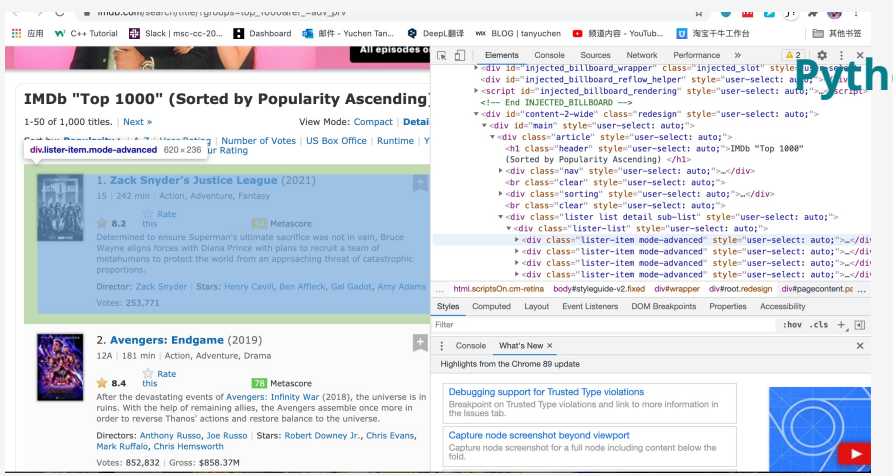
Read, parse

Module: Image (PIL library)

Image processing

Python webscraper | *Yuchen Tan 20032211*

Python webscraper



movie	year	timeMin	imdb	metascore	votes	us_gro
Zack Snyder's Justice League	2021	242	8.2	54	224695	
Avengers: Endgame	2019	181	8.4	78	845388	858.37
Sound of Metal	2019	120	7.8	82	50200	
The Trial of the Chicago 7	2020	129	7.8	77	119404	
Soul	2020	100	8.1	83	210761	
Judas and the Black Messiah	2021	126	7.6	85	29877	
Another Round	2020	117	7.8	80	52804	
The Dark Knight	2008	152	9.0	84	2331867	534.86
Avengers: Infinity War	2018	149	8.4	68	857783	678.82
Watchmen	2009	162	7.6	56	508281	107.51
Joker	2019	122	8.4	59	971358	335.45

Next was the fourth week of exercises to build a Python webscraper. I built two webscraper here.

For the first project, I'll scrape data from IMDb's "Top 1,000" movies, specifically the top 50 movies on this page.

FIELD1	movie	year	timeMin	imdb	metascore	votes	us_grossMillions
0	Zack Snyder's Justice League	2021	242	8.2	54	224695	
1	Avengers: Endgame	2019	181	8.4	78	845388	858.37
2	Sound of Metal	2019	120	7.8	82	50200	
3	The Trial of the Chicago 7	2020	129	7.8	77	119404	
4	Soul	2020	100	8.1	83	210761	
5	Judas and the Black Messiah	2021	126	7.6	85	29877	
6	Another Round	2020	117	7.8	80	52804	
7	The Dark Knight	2008	152	9.0	84	2331867	534.86
8	Avengers: Infinity War	2018	149	8.4	68	857783	678.82
9	Watchmen	2009	162	7.6	56	508281	107.51
10	Joker	2019	122	8.4	59	971358	335.45

IMDb's "Top 1,000" movies

IMFORMATION

- The title
- The year it was released
- How long the movie is
- IMDb's rating of the movie
- The Metascore of the movie
- How many votes the movie got
- The U.S. gross earnings of the movie

WEB STRUCTURE

- What's the structure of the web page that contains the data you're looking for?
- How do we get to those web pages?
- Will you need to gather more data from the next page?

Summarises and extract keywords from news using gensim

In today's digital world, we're bombarded with endless information. We've got infinitely scrolling social media feeds and a 24-hour news cycle. So there's plenty of news to stay aware of and we've got to be able to digest it quickly!

So I use bs and gensim to help me quickly summarize news and extract news keywords

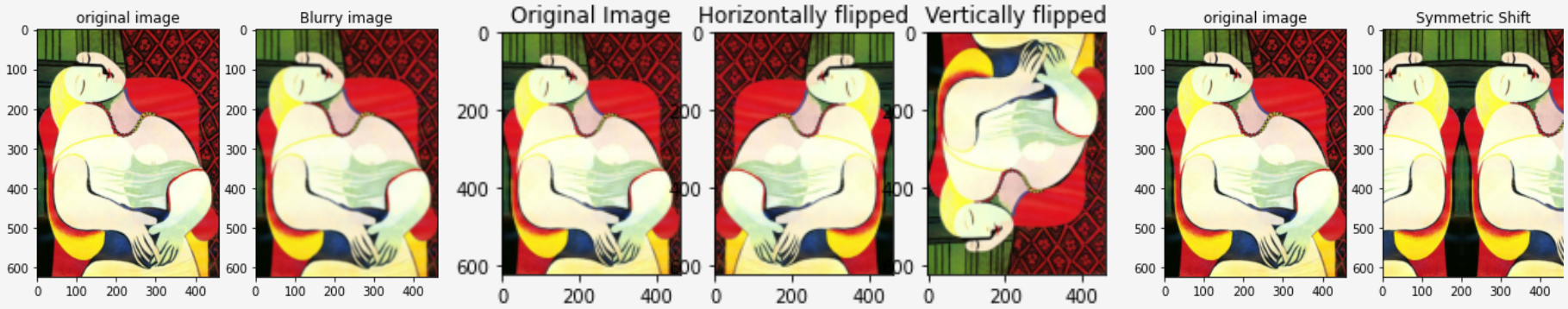
IMPORTS

```
# Imports
import requests
from bs4 import BeautifulSoup
from gensim.summarization import summarize
from gensim.summarization import keywords
```

SUMMARY

```
summary = summarize(article_text, ratio=0.3)
```

Signal Processing in Numpy and Tensorflow



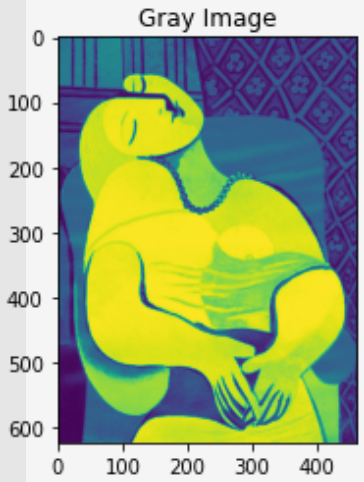
✨Image augmentation is a strategy that enables practitioners to significantly increase the diversity of images available for training models, without actually collecting new images. For training any Machine Learning model and specifically Deep Learning model, having a large dataset is very important and can improve the performance of the model dramatically. When we train a deep learning model on images, we need at least tens of thousands of images to generalise the pattern of the images.

Conclusion

Generating image data is expensive and tedious work. Machines are blind without data. In such a case, we can generate more images from existing images by applying different transformations techniques on it. There are different techniques like rotation, flipping, shifting, etc. which can help us to diversify the image data. We can generate more than 10X or 100X images if we have at least 4–5 transformations techniques. We can also use methods to blur the image and add random noise to image, to generate more images.

GRAY

```
gray= cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
plt.subplot(121)
plt.imshow(img)
plt.title("Original Image")
plt.subplot(122)
plt.imshow(gray)
plt.title("Gray Image")
plt.show()
```



WRAP

```
transform = AffineTransform(translation=(-200,0))
warp_image = warp(img,transform, mode="wrap")
plt.subplot(1,2,1)
plt.title('original image')
plt.imshow(img)
plt.subplot(1,2,2)
plt.title('Wrap Shift')
plt.imshow(warp_image)
```

OIL STYLE

I also tried to use cv2 to convert Picasso's abstract paintings into a oil style

THANKS