Image processing:

from PIL import Image , ImageFilter  
img = Image.open('C:\\Users\\gkalivar\\Desktop\\Images-Downloaded\\iron\_man.jpg')  
filter\_image = img.filter(ImageFilter.BLUR)  
filter\_image.save("C:\\Users\\gkalivar\\Desktop\\Images-Downloaded\\blur.jpg","png")  
print(img.format)

from PIL import Image , ImageFilter  
img = Image.open('C:\\Users\\gkalivar\\Desktop\\Images-Downloaded\\iron\_man.jpg')  
filter\_img = img.convert('L')  
crooked = filter\_img.rotate(180)  
#resize = filter\_img.resize((300,300))  
  
crooked.save("C:\\Users\\gkalivar\\Desktop\\Images-Downloaded\\blur.jpg","png")  
print(img.format)

resize

from PIL import Image , ImageFilter  
img = Image.open('C:\\Users\\gkalivar\\Desktop\\Images-Downloaded\\iron\_man.jpg')  
img.thumbnail((400,400))  
img.save('C:\\Users\\gkalivar\\Desktop\\Images-Downloaded\\thumbnail.jpg')

Changing the file type from jpg to png

import sys  
import os  
from PIL import Image  
  
path = sys.argv[1]  
directory = sys.argv[2]  
  
if not os.path.exists(directory):  
 os.makedirs(directory)  
  
for filename in os.listdir(path):  
 clean\_name = os.path.splitext(filename)[0]  
 img = Image.open(f'{path}{filename}')  
 #added the / in case user doesn't enter it. You may want to check for this and add or remover it.  
 img.save(f'{directory}/{clean\_name}.png', 'png')  
 print('all done!')

OpenCv is used in machine learning

**Processing PDF**

Pip install PyPDF2

import PyPDF2  
  
with open('C:\\Users\\gkalivar\\Desktop\\pdf\\dummy.pdf','rb') as file:  
 reader = PyPDF2.PdfFileReader(file)  
 print(reader.numPages)  
 print(reader.getPage(0))

Merging two pdf

import PyPDF2  
import sys  
  
inputs = sys.argv[1:]  
  
def pdfmerge(pdf\_list):  
 merger = PyPDF2.PdfFileMerger()  
 for pdf in pdf\_list:  
 print(pdf)  
 merger.append(pdf)  
 merger.write('super.pdf')

Merging the watermark with the pdf

import PyPDF2  
  
temp = PyPDF2.PdfFileReader(open('C:\\Users\\gkalivar\\Desktop\\pdf\\dummy.pdf','rb'))  
water = PyPDF2.PdfFileReader(open('C:\\Users\\gkalivar\\Desktop\\pdf\\wtr.pdf','rb'))  
  
output = PyPDF2.PdfFileWriter()  
  
for i in range(temp.getNumPages()):  
 page = temp.getPage(i)  
 page.mergePage(water.getPage(0))  
 output.addPage(page)  
  
 with open('C:\\Users\\gkalivar\\Desktop\\pdf\\watermarked\_output.pdf','wb') as file:  
 output.write(file)

Send emails in python

Links Less Secured Apps : <https://support.google.com/accounts/answer/6010255>   
Third party sites & apps: <https://support.google.com/accounts/answer/3466521>

import smtplib  
from email.message import EmailMessage  
  
email = EmailMessage()  
email['fram'] = 'dummy'  
email['to'] = 'gocool94@gmail.com'  
email['subject'] = ' i am a Bot'  
  
email.set\_content('I am good boy')  
  
with smtplib.SMTP(host='smtp.gmail.com',port=587) as smtp:  
 smtp.ehlo()  
 smtp.startts()  
 smtp.login('email','password')  
 smtp.send\_message()  
 print('mail sent')

Web scraping:

import requests

from bs4 import BeautifulSoup

import pprint

res = requests.get('https://news.ycombinator.com/news')

res2 = requests.get('https://news.ycombinator.com/news?p=2')

soup = BeautifulSoup(res.text, 'html.parser')

soup2 = BeautifulSoup(res2.text, 'html.parser')

links = soup.select('.storylink')

subtext = soup.select('.subtext')

links2 = soup2.select('.storylink')

subtext2 = soup2.select('.subtext')

mega\_links = links + links2

mega\_subtext = subtext + subtext2

def sort\_stories\_by\_votes(hnlist):

return sorted(hnlist, key= lambda k:k['votes'], reverse=True)

def create\_custom\_hn(links, subtext):

hn = []

for idx, item in enumerate(links):

title = item.getText()

href = item.get('href', None)

vote = subtext[idx].select('.score')

if len(vote):

points = int(vote[0].getText().replace(' points', ''))

if points > 99:

hn.append({'title': title, 'link': href, 'votes': points})

return sort\_stories\_by\_votes(hn)

pprint.pprint(create\_custom\_hn(mega\_links, mega\_subtext))

Exercise:

1. Read a jpeg image and print the image file
2. Merge two pdf files using python script
3. Scrape a website and store the data into DB.
4. Write queries to filter the data in db