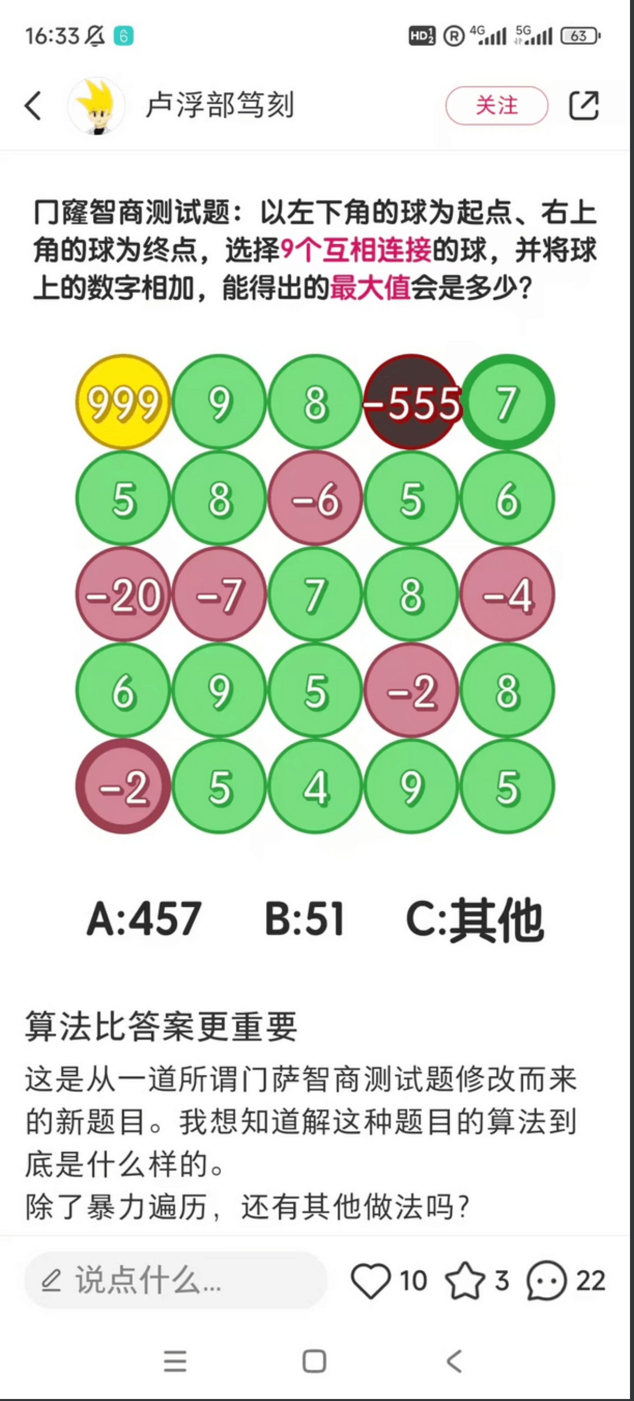
Unstructured Data



门薩智商测试题：以左下角的球为起点、右上角的球为终点，选择9个互相连接的球，并将球上的数字相加，能得出的最大值会是多少？

999	9	8	-555	7
5	8	-6	5	6
-20	-7	7	8	-4
6	9	5	-2	8
-2	5	4	9	5

A:457 B:51 C:其他

算法比答案更重要

这是从一道所谓门萨智商测试题修改而来的新题目。我想知道解这种题目的算法到底是什么样的。
除了暴力遍历，还有其他做法吗？

You are moving from the bottom-left ball to the top-right ball. Each round you can move up or move right, so you will pass 9 balls in total. Summing up these 9 numbers, what is the maximum value you can get?

Write an algorithm for it!

What’s Unstructured Data?

- Structured Data: Data that has a standardized format for efficient access by software and humans alike.
- Unstructured Data: Information that either does not have a pre-defined data model or is not organized in a pre-defined manner.
- Examples of Unstructured Data: Text data (emails, news articles, books, reviews), images, videos, audio data etc.

Processing Text Data in Python

Processing Text Data in Python

#1: Word Frequency

Word Frequency

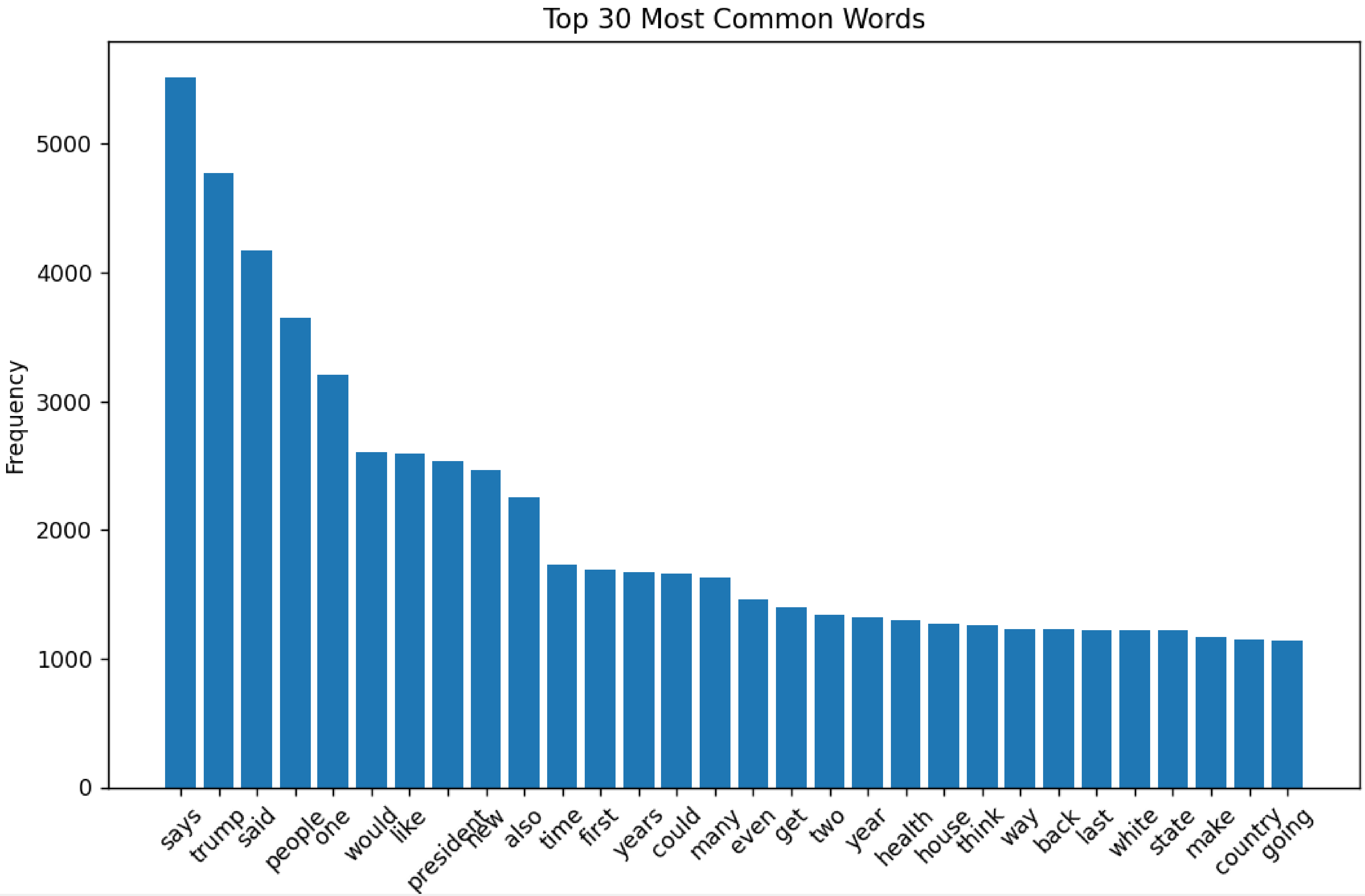
Given a document, you may want to find out the most frequently used words in the document; this is called word frequency analysis. It can be easily done with Python; but please remove the stop words (i.e., words such as “the”, “it”, “mine”, “but”) from the list!

Word Frequency

Consider a text dataset which is available here:

https://ximarketing.github.io/data/LDAshort.txt

Ask GPT to help you visualize the most frequent words used in the text document!


Top 30 Most Common Words
Frequency
0
1000
2000
3000
4000
5000
says
trump
said
people
one
would
like
president
new
also
time
first
years
could
many
even
get
two
year
health
house
think
way
back
last
white
state
make
country
going

Processing Text Data in Python

#2: Topic Models

Topic Models

Topic modeling is an unsupervised learning algorithm. It takes a large number of documents (aka corpus) as input, and automatically generates a few topics from the text. Here, each topic is described by a distribution of words (typically we describe the most common words of the topic).

Topic Models

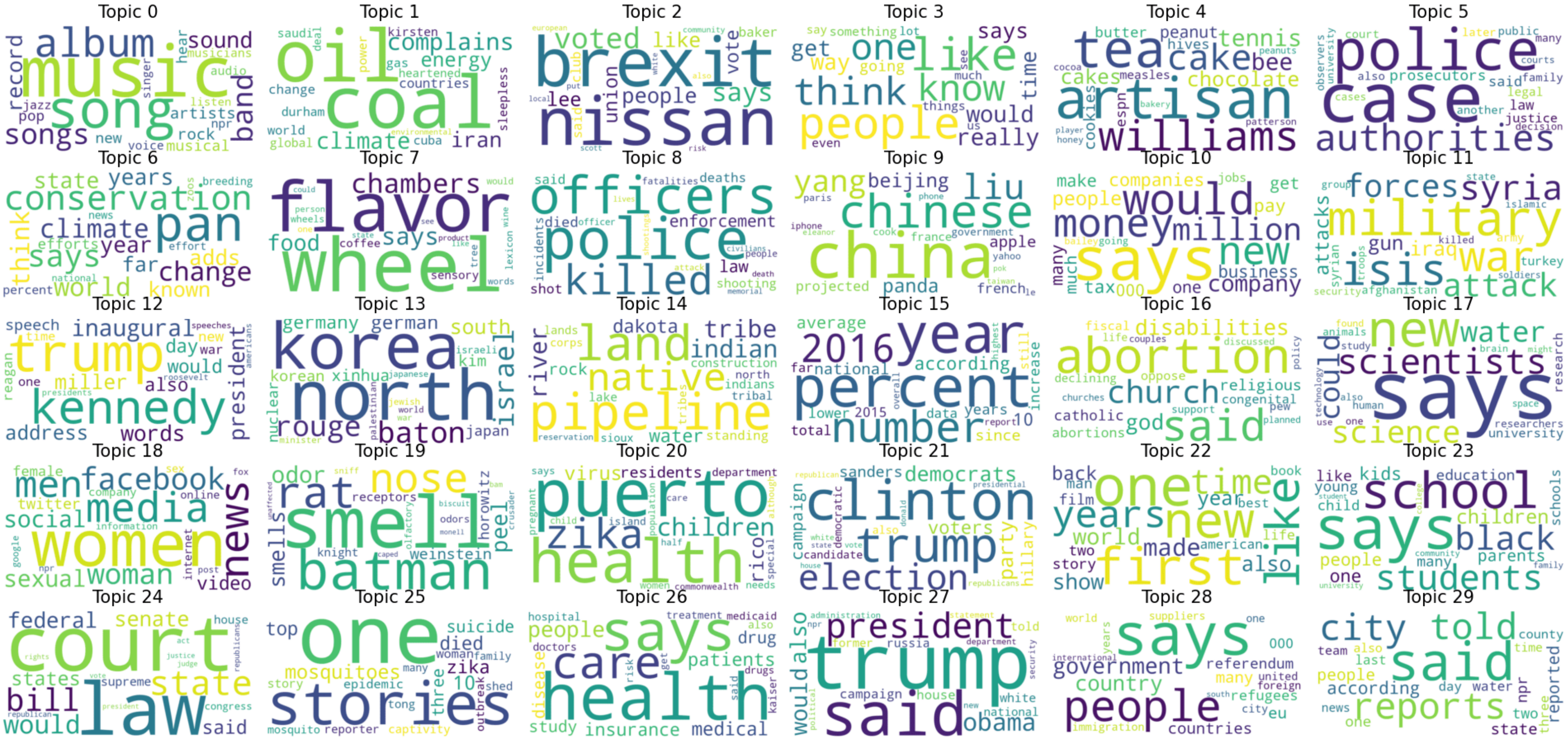
How does topic modeling work? The basic idea is, words of the same topic tend to appear simultaneously. For instance, in the topic that described “higher education”, some key words “university”, “library”, “examination”, “enrolment” often appear together.

You don’t need to understand all these. Using topic models directly to get your results!

Topic Models

The most popular method for topic modeling is latent Dirichlet allocation (LDA). Let me show you an example based on LDA analysis of articles published by Pew Research Center, an American think tank.

I did not want to read any of their articles, but I do want to know what topics they are discussing.


Topic 0
music song album songs record sound band musicians audio hear singer jazz pop artists listen npr rock new voice musical
Topic 1
oil coal complains energy kirsten saudi deal power gas heartened countries change durham world global climate environmental cuba iran sleepless
Topic 2
brexit nissan voted like vote says people union club lee european community baker white also put local said scott risk
Topic 3
one like people think know says get way going something lot say see much time things would us really even
Topic 4
tea artisan cake williams butter peanut tennis hives bee peanuts cocoa cakes measles chocolate bakery cookies espn player honey patterson
Topic 5
police case authorities court later public many observers university also prosecutors said family courts legal law cases another justice decision
Topic 6
conservation pan state years breeding zoos climate think efforts year effort says far adds change national world percent known news
Topic 7
flavor wheel chambers would could person wheels one food coffee says see product like tree wine lexicon sensory words
Topic 8
officers police killed said fatalities deaths lives died officer enforcement incidents shootings civilians people law death attack shooting memorial shot
Topic 9
china chinese yang beijing liu paris phone iphone eleanor cook france government apple yahoo pok taiwan projected panda french le
Topic 10
says would money million new make companies jobs get people pay bailey going many much tax 000 one business company
Topic 11
military isis syria forces war attack group state islamic attacks gun iraq killed army syrian troops turkey soldiers security afghanistan
Topic 12
trump kennedy inaugural speech time speeches new war day would reagan one miller also roosevelt presidents president americans address words
Topic 13
korea north germany german south israeli kim israel korean xinhua japanese nuclear jewish world war palestinian rouge baton japan minister
Topic 14
land pipeline native dakota tribe indian construction lands corps river rock north indians tribal lake tribes reservation sioux water standing
Topic 15
percent year number 2016 average far national according highest still increase overall lower 2015 data years report 10 total since
Topic 16
abortion church said disabilities fiscal life couples discussed policy declining oppose religious congenital churches pew catholic support god abortions planned
Topic 17
says new scientists science water animals found study brain might research could technology also human space one use researchers university
Topic 18
women media news facebook men female sex fox twitter company online social information google internet npr post sexual woman video
Topic 19
smell batman rat nose odor sniff bam receptors biscuit odors monell unaffected olfactory horowitz peel crusader smells knight caped weinstein
Topic 20
puerto health zika says virus residents department care island child pregnant population although children half rico special women commonwealth needs
Topic 21
clinton trump election sanders democrats republican presidential campaign donald democratic white state vote also voters party hillary candidate house republicans
Topic 22
one first new like time years back book man film year best world life two made american story also show
Topic 23
says school students kids like education young student child college children schools black people community many parents family one university
Topic 24
court law state federal senate house republicans rights act justice judge states vote supreme bill president congress republican would said
Topic 25
one stories top suicide died woman family mosquitoes many zika story epidemic 10 shed three outbreak tong mosquito reporter captivity
Topic 26
health says care hospital people treatment medicaid also drug doctors patients drugs disease risk get said kaiser study insurance medical
Topic 27
trump said president administration statement told npr former russia department security also would campaign house white political new national obama
Topic 28
says people government world suppliers one years 000 international referendum country many united foreign refugees south city eu immigration countries
Topic 29
said city told reports county team also time last people according day water news one npr two three reported state

Interpreting the topics:

Topic 0: music, song, album, band, ...

Topic 1: oil, coal, energy, climate, ...

Topic 9: china, chinese, beijing, panda, ...

Topic 18: woman, man, female, sexual, ...

Topic 21: trump, clinton, election, sanders, ...

Topic 27: trump, obama, president, russia, ...

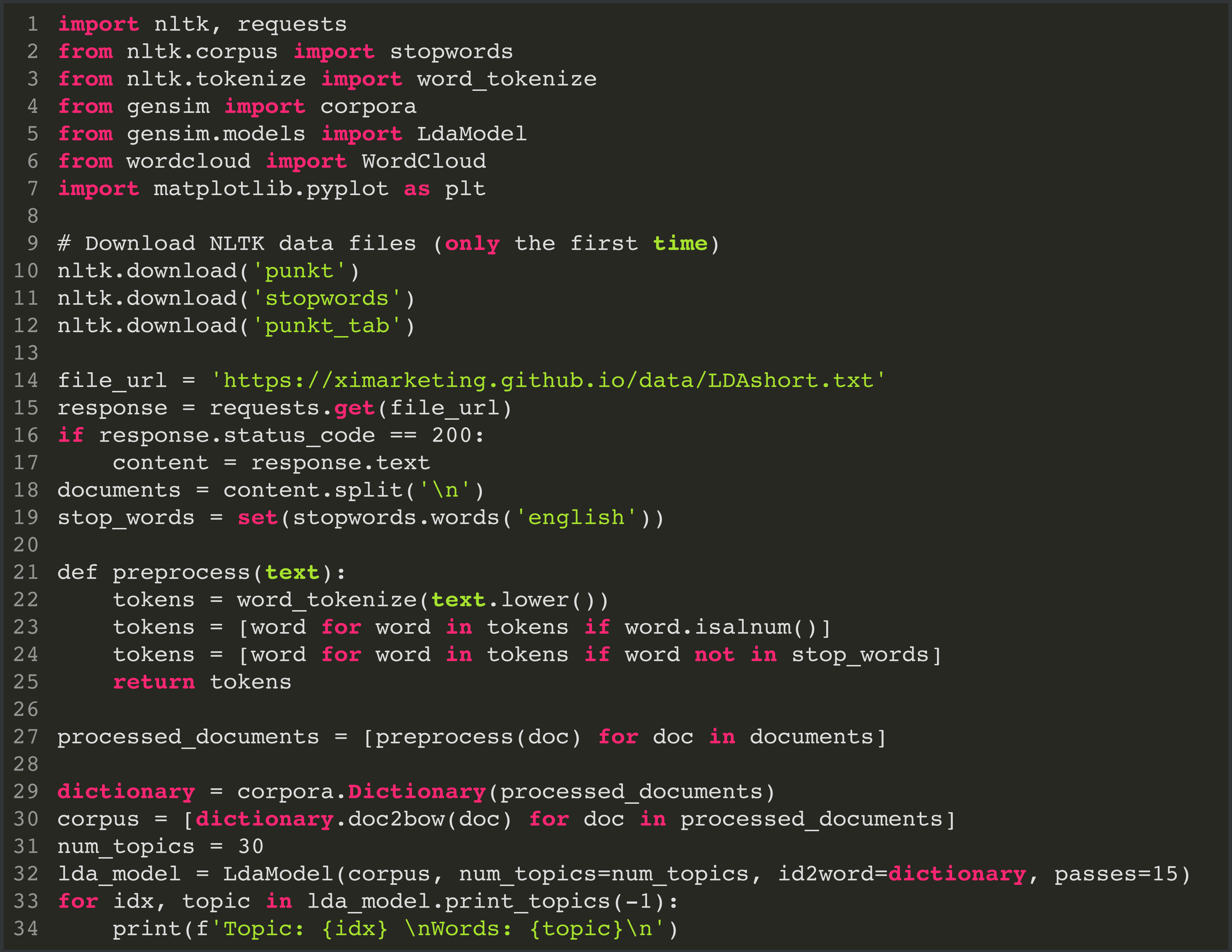
Let’s try to play with LDA!

The original dataset is too large and I am unable to share it with you. Instead, we have a shorter version of the data with it stored at the following location:

https://ximarketing.github.io/data/LDAshort.txt

Here, each line is an article and there are 1000+ articles.

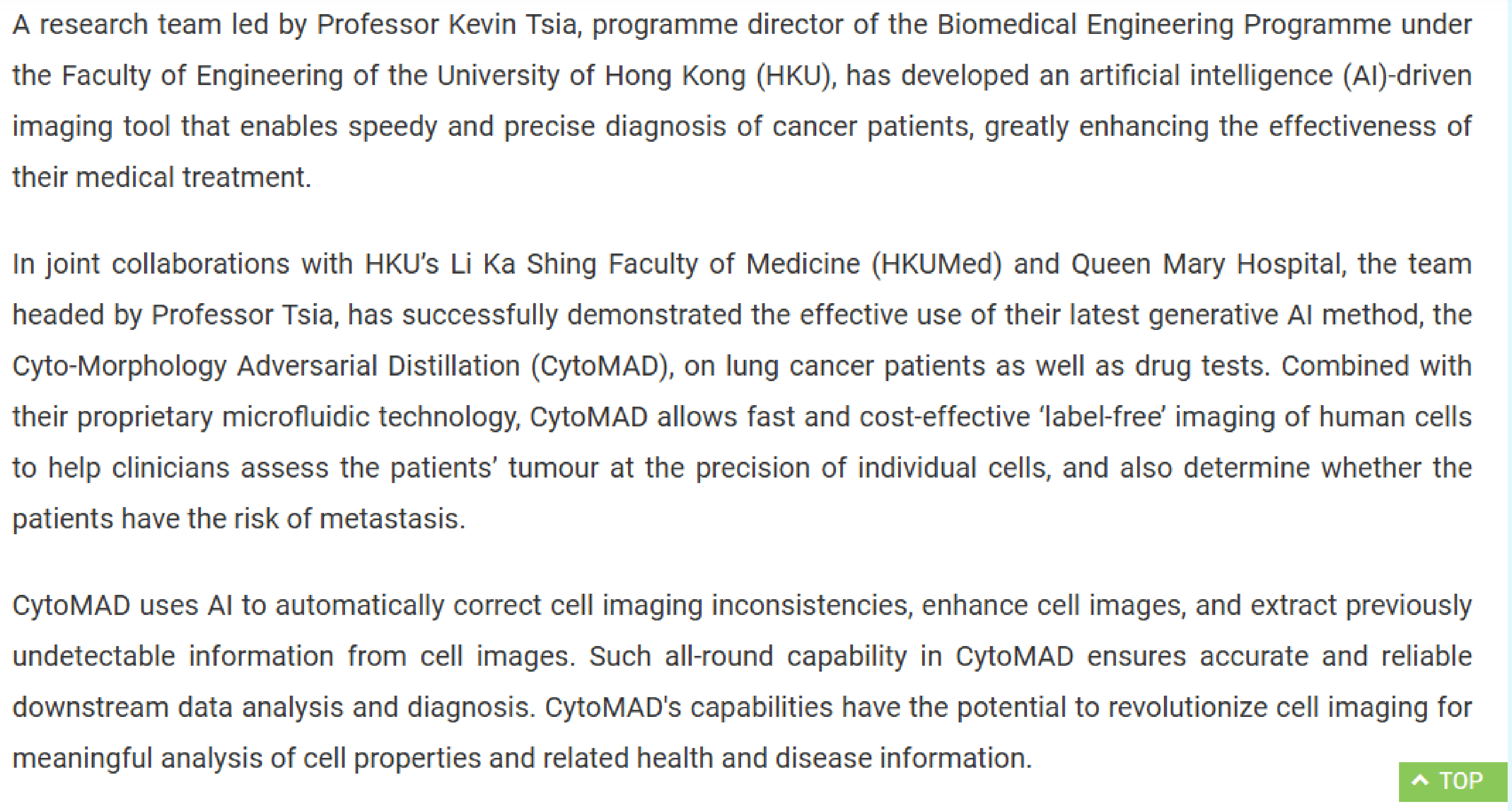
Ask GPT to develop write code for you, and visualize your result (note: you need to install several modules).



```python
import nltk, requests
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from gensim import corpora
from gensim.models import LdaModel
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Download NLTK data files (only the first time)
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('punkt_tab')

file_url = 'https://ximarketing.github.io/data/LDAshort.txt'
response = requests.get(file_url)
if response.status_code == 200:
    content = response.text
documents = content.split('\n')
stop_words = set(stopwords.words('english'))

def preprocess(text):
    tokens = word_tokenize(text.lower())
    tokens = [word for word in tokens if word.isalnum()]
    tokens = [word for word in tokens if word not in stop_words]
    return tokens

processed_documents = [preprocess(doc) for doc in documents]

dictionary = corpora.Dictionary(processed_documents)
corpus = [dictionary.doc2bow(doc) for doc in processed_documents]
num_topics = 30
lda_model = LdaModel(corpus, num_topics=num_topics, id2word=dictionary, passes=15)
for idx, topic in lda_model.print_topics(-1):
    print(f'Topic: {idx} \nWords: {topic}\n')
```

Processing Text Data in Python

#3: Summarizing Text



A research team led by Professor Kevin Tsia, programme director of the Biomedical Engineering Programme under the Faculty of Engineering of the University of Hong Kong (HKU), has developed an artificial intelligence (AI)-driven imaging tool that enables speedy and precise diagnosis of cancer patients, greatly enhancing the effectiveness of their medical treatment.

In joint collaborations with HKU's Li Ka Shing Faculty of Medicine (HKUMed) and Queen Mary Hospital, the team headed by Professor Tsia, has successfully demonstrated the effective use of their latest generative AI method, the Cyto-Morphology Adversarial Distillation (CytoMAD), on lung cancer patients as well as drug tests. Combined with their proprietary microfluidic technology, CytoMAD allows fast and cost-effective 'label-free' imaging of human cells to help clinicians assess the patients' tumour at the precision of individual cells, and also determine whether the patients have the risk of metastasis.

CytoMAD uses AI to automatically correct cell imaging inconsistencies, enhance cell images, and extract previously undetectable information from cell images. Such all-round capability in CytoMAD ensures accurate and reliable downstream data analysis and diagnosis. CytoMAD's capabilities have the potential to revolutionize cell imaging for meaningful analysis of cell properties and related health and disease information.

Here is a lengthy article about HKU research.
What does it talk about?

Text Summarization

Imagine that you want to understand the meaning of the text, but you do not really want to read the whole paragraphs to grasp its meaning. You just need a short summary of the text. This can be done with text summarization. Here is my output:

a team led by Professor Kevin Tsia has developed an artificial intelligence-driven imaging tool that enables speedy and precise diagnosis of cancer patients .

Abstractive Summarization Using Transformers

```python
from transformers import pipeline
# Load the summarization pipeline
summarizer = pipeline("summarization")
# Sample text
text = """
Input your text here
"""
# Generate a summary
summary = summarizer(text, max_length=40, min_length=15, do_sample=False)
print(summary[0]['summary_text'])
```

Processing Text Data in Python
#4: Using APIs

This will be covered later in the class.

Processing Images in Python

#1: Display an Image

Image I/O and display

```
 1 from PIL import Image, ImageFont, ImageDraw
 2 from io import BytesIO
 3 import requests
 4 import matplotlib.pylab as plt
 5 import PIL.ImageStat as stat
 6 url = "https://ximarketing.github.io/data/mainbuilding.jpg"
 7 response = requests.get(url)
 8 image = Image.open(BytesIO(response.content))
 9 print(image.width, image.height, image.mode, image.format)
10 image.show()
```

If you are reading image from a local folder, you can replace line 6 with your path:

```
image = Image.open(".../mainbuilding.jpg")
```

Processing Images in Python

#2: Color image to grayscale

A grayscale image only has two colors, black and white, and each pixel is given by a value $0 \leq x_i \leq 255$. When x_i is close to 0, the pixel is blacker, and when x_i is closer to 255, the pixel is whiter. We can use a simple formula to covert color image:

$$x_i = 0.299 \cdot r_i + 0.587 \cdot g_i + 0.114 \cdot b_i.$$

Color image to gray scale

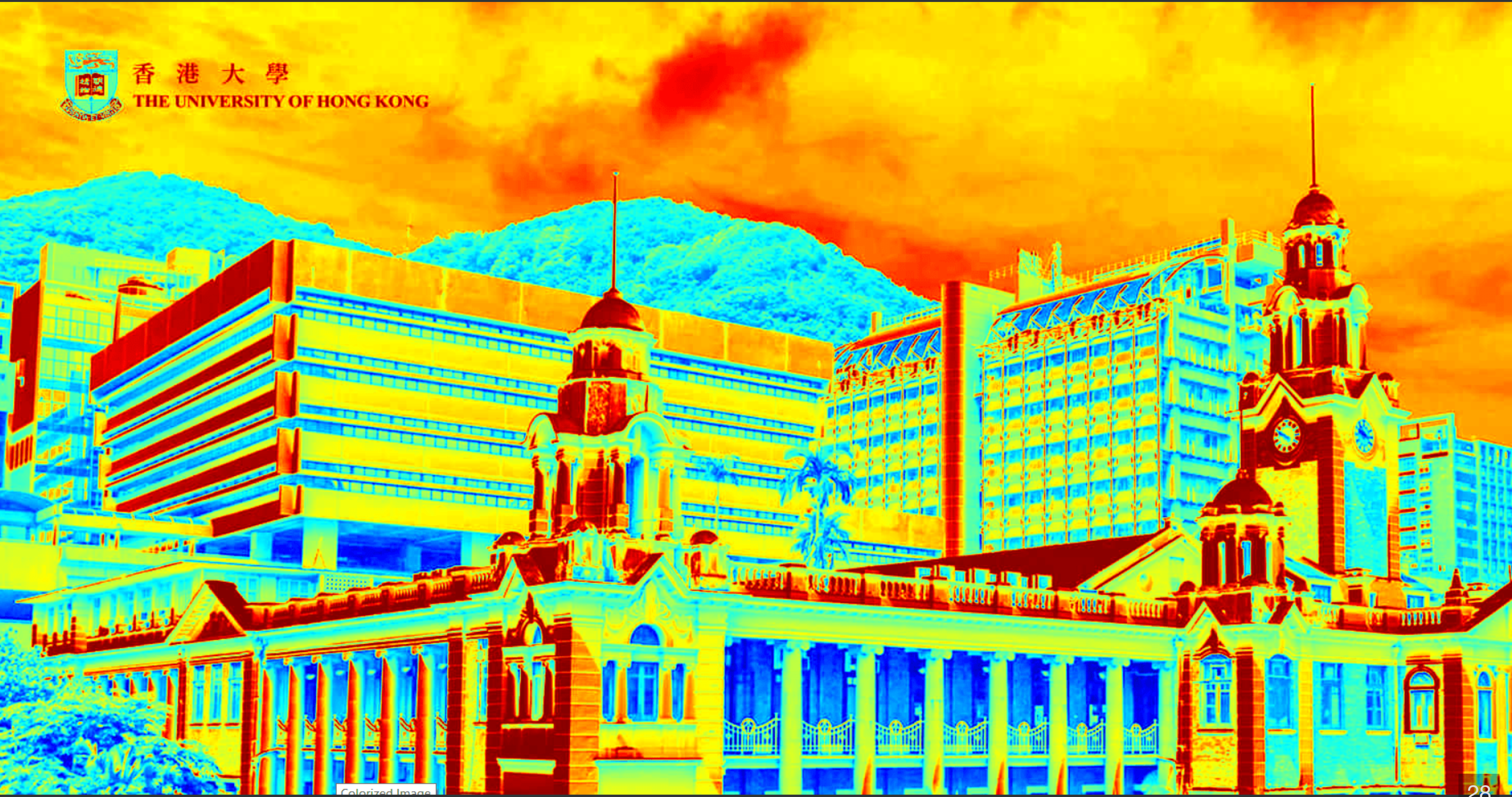
```python
from PIL import Image
import requests
from io import BytesIO
url = "https://ximarketing.github.io/data/mainbuilding.jpg"
response = requests.get(url)
img = Image.open(BytesIO(response.content))
gray_image = img.convert('L')
gray_image.save('gray_image.jpg')
gray_image.show()
```

How about adding colors to grayscale images?

This is much harder because we lose a lot of information when creating grayscale images: Given x_i, there are many combinations (r_i, g_i, b_i) that satisfy

$$x_i = 0.299 \cdot r_i + 0.587 \cdot g_i + 0.114 \cdot b_i.$$

It’s not so great...



Processing Images in Python
#3: Adjusting Brightness

How about adjusting brightness of images?

When the color of a pixel is $(0, 0, 0)$, it is black.

When the color of a pixel is $(255, 255, 255)$, it is white.

Suppose that the color of a pixel is (r, g, b). If you move the color toward $(0, 0, 0)$, it becomes darker. If you move the color toward $(255, 255, 255)$, it becomes brighter.

This makes your image brighter.

```
 1 from PIL import Image, ImageEnhance
 2 import requests
 3 from io import BytesIO
 4 # URL of the image
 5 url = 'https://ximarketing.github.io/data/mainbuilding.jpg'
 6 # Fetch the image from the URL
 7 response = requests.get(url)
 8 img = Image.open(BytesIO(response.content))
 9 image = img.point(lambda x: (255+x)/2)
10 image.show()
```

If you want darker, replace line 9 with:

```
1 image = img.point(lambda x: x/2)
```

Exercise: Develop an APP with the following functions

- The user can upload an image from his or her folder.
- There is a grayscale button. When a user clicks on the button, it converts the original image into grayscale and displays the grayscale image on a display panel to the user.
- Similarly, there is a brighter button, which makes the image brighter, and a darker button, which makes the image darker, and display the new image.

Processing Images in Python

#4: Changing colors

Making the Image Red

```python
from PIL import Image, ImageEnhance
import requests
from io import BytesIO
# URL of the image
url = 'https://ximarketing.github.io/data/mainbuilding.jpg'
# Fetch the image from the URL
response = requests.get(url)
image = Image.open(BytesIO(response.content))
for i in range(image.width):
    for j in range(image.height):
        pixel = (image.getpixel((i, j)))
        r = int((pixel[0] + 255)/2)
        g = pixel[1]
        b = pixel[2]
        image.putpixel((i, j), (r, g, b))
image.show()
```

Negating the Image

```python
from PIL import Image, ImageEnhance
import requests
from io import BytesIO
# URL of the image
url = 'https://ximarketing.github.io/data/mainbuilding.jpg'
# Fetch the image from the URL
response = requests.get(url)
image = Image.open(BytesIO(response.content))
for i in range(image.width):
    for j in range(image.height):
        pixel = (image.getpixel((i, j)))
        r = 255 - pixel[0]
        g = 255 - pixel[1]
        b = 255 - pixel[2]
        image.putpixel((i, j), (r, g, b))
image.show()
```

Processing Images in Python

#5: Rotation

Rotating the Image

```python
from PIL import Image, ImageEnhance
import requests
from io import BytesIO
# URL of the image
url = 'https://ximarketing.github.io/data/mainbuilding.jpg'
# Fetch the image from the URL
response = requests.get(url)
image = Image.open(BytesIO(response.content))
image_t = image.transpose(Image.FLIP_LEFT_RIGHT)
image_t.show()
```

Rotating the Image, 45 degree

```python
image_45 = image.rotate(45)
image_45.show()
```

Processing Images in Python
#6: Adding Information

Adding Text to Image

```python
from PIL import Image, ImageEnhance,ImageDraw, ImageFont
import requests
from io import BytesIO
# URL of the image
url = 'https://ximarketing.github.io/data/mainbuilding.jpg'
# Fetch the image from the URL
response = requests.get(url)
image = Image.open(BytesIO(response.content))
draw = ImageDraw.Draw(image)
font = ImageFont.truetype("arial.ttf", 22)
draw.text((435, 93), "teaches me Python", font = font)
image.show()
```

Adding a Thumbnail

```python
from PIL import Image, ImageEnhance,ImageDraw, ImageFont
import requests
from io import BytesIO
url = "https://ximarketing.github.io/data/trump2.jpg"
response = requests.get(url)
image_thumbnail = Image.open(BytesIO(response.content))
image_thumbnail.thumbnail((120,120))
url = "https://ximarketing.github.io/data/mainbuilding.jpg"
response = requests.get(url)
image = Image.open(BytesIO(response.content))
image.paste(image_thumbnail, (15, 55))
image.show()
```

Blending two images

Next, we are going to α-blend two images. The two images must have the same size and mode, and blending works as follows:

$$output = (1 - \alpha) \cdot image_1 + \alpha \cdot image_2$$

Code for blending Trump and HKU:

```python
from PIL import Image, ImageFont, ImageDraw
from io import BytesIO
import requests
url = "https://ximarketing.github.io/data/trump2.jpg"
response = requests.get(url)
image1 = Image.open(BytesIO(response.content))
url = "https://ximarketing.github.io/data/mainbuilding.jpg"
response = requests.get(url)
image2 = Image.open(BytesIO(response.content))
image2 = image2.resize((image1.width, image1.height), Image.BILINEAR)
image = Image.blend(image1, image2, alpha = 0.3)
draw = ImageDraw.Draw(image)
font = ImageFont.truetype("arial.ttf", 50)
draw.text((350, 93), "Make HKU Great Again!",
          font = font, fill='red')
image.show()
```

Processing Images in Python
#7: Color Channels

Separating colors in an image

```python
from PIL import Image, ImageFont, ImageDraw
from io import BytesIO
import matplotlib.pylab as plt
import requests
url = "https://ximarketing.github.io/data/parrot.jpg"
response = requests.get(url)
image = Image.open(BytesIO(response.content))
ch_r, ch_g, ch_b = image.split()
# split the RGB image into 3 channels: R, G and B
plt.imshow(ch_r, cmap = plt.cm.Reds)
plt.show()
plt.imshow(ch_g, cmap = plt.cm.Greens)
plt.show()
plt.imshow(ch_b, cmap = plt.cm.Blues)
plt.show()
```

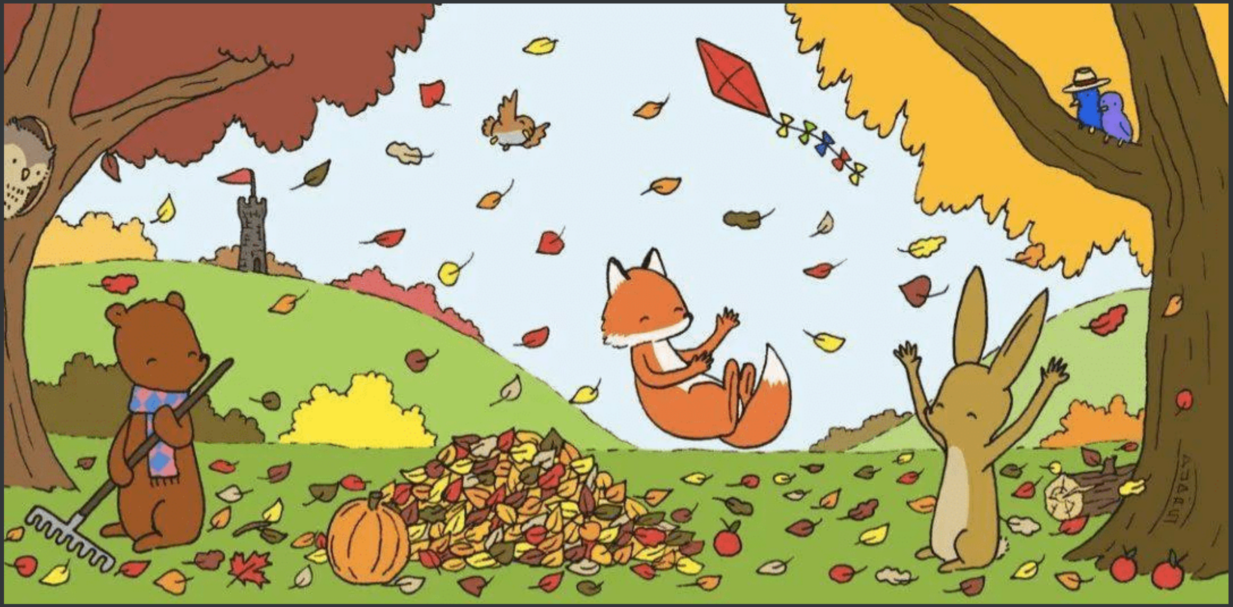
Changing the color of your parrot!

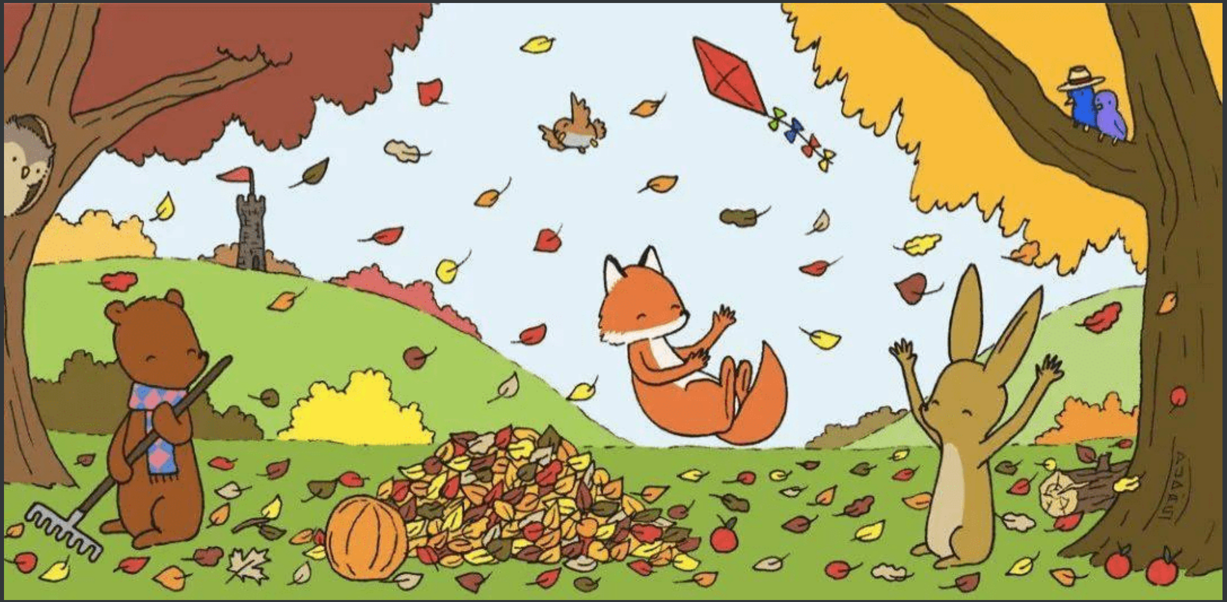
```python
from PIL import Image, ImageFont, ImageDraw
from io import BytesIO
import matplotlib.pylab as plt
import requests
url = "https://ximarketing.github.io/data/parrot.jpg"
response = requests.get(url)
image = Image.open(BytesIO(response.content))
ch_r, ch_g, ch_b = image.split()
# split the RGB image into 3 channels: R, G and B
image = Image.merge('RGB', (ch_b, ch_g, ch_r))
image.show()
```

Processing Images in Python

#8: Compare Images

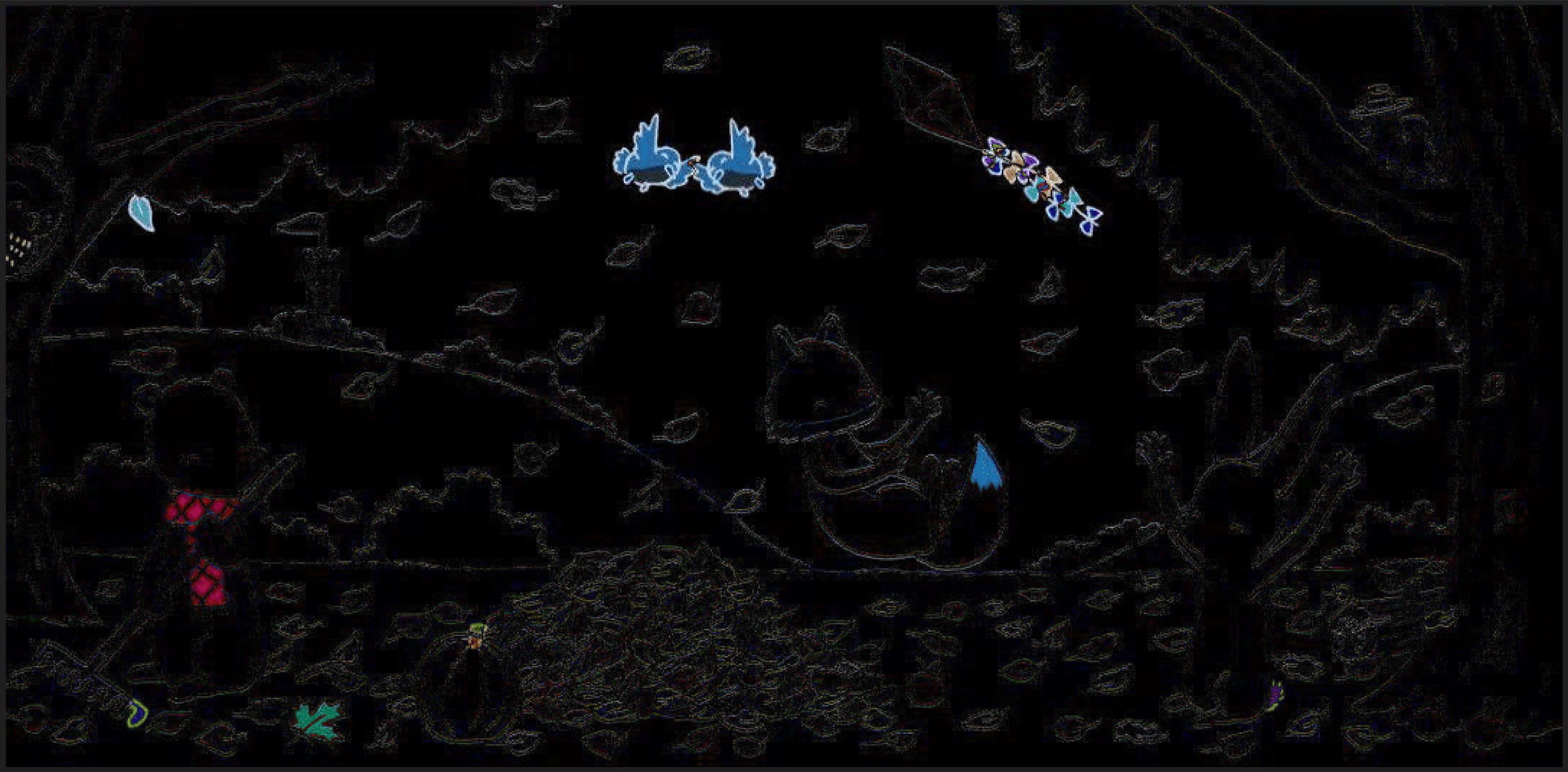
Discover the differences between two images:





Discover the differences between two images:

```python
from PIL.ImageChops import difference
from PIL import Image
from io import BytesIO
import requests
url = "https://ximarketing.github.io/data/????.jpeg"
response = requests.get(url)
image1 = Image.open(BytesIO(response.content))
url = "https://ximarketing.github.io/data/????.jpeg"
response = requests.get(url)
image2 = Image.open(BytesIO(response.content))
image = difference(image1, image2)
image.show()
```

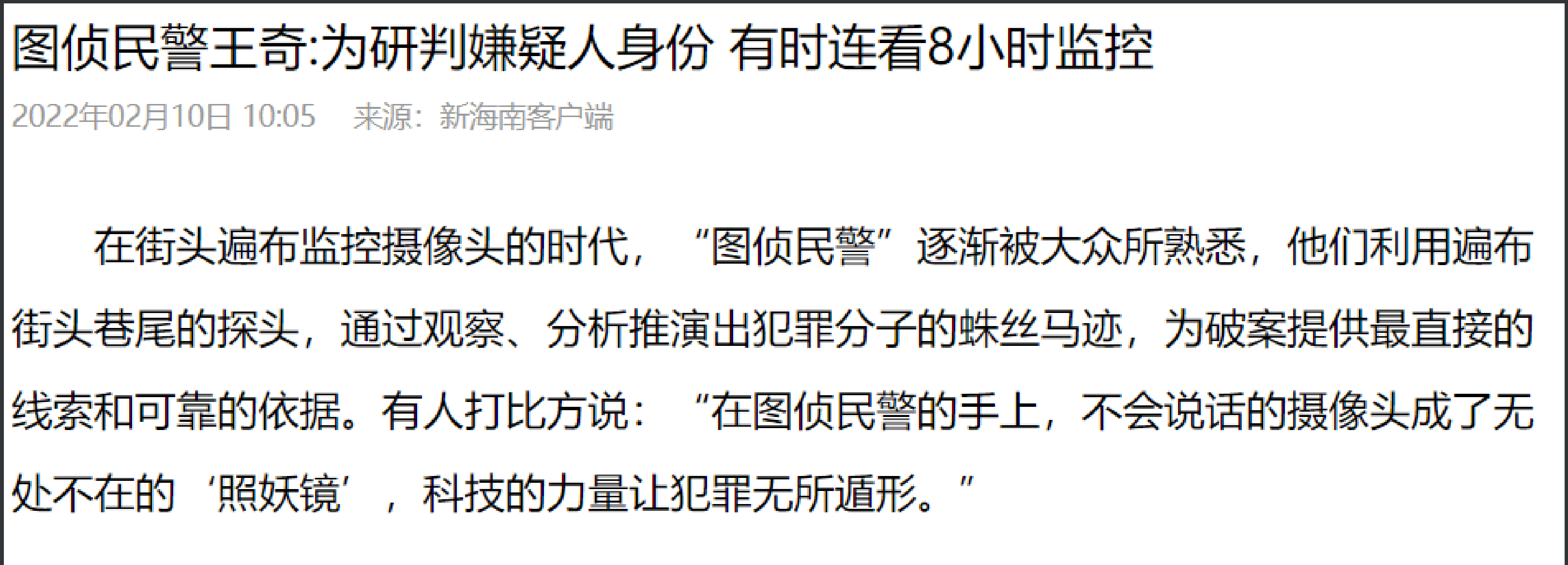


Question: What are the applications of image comparison?

Calculating Image Difference is very useful!

- Detecting and removing similar images from your smartphone storage.
- Video compression: If video frames are similar, we can just save their differences, not the entire images.
- Motion analysis: detecting anomalies in CCTV recording and sports game videos.
- Analyzing the information content of videos in computer vision.

We can help you relax :)



图侦民警王奇:为研判嫌疑人身份 有时连看8小时监控

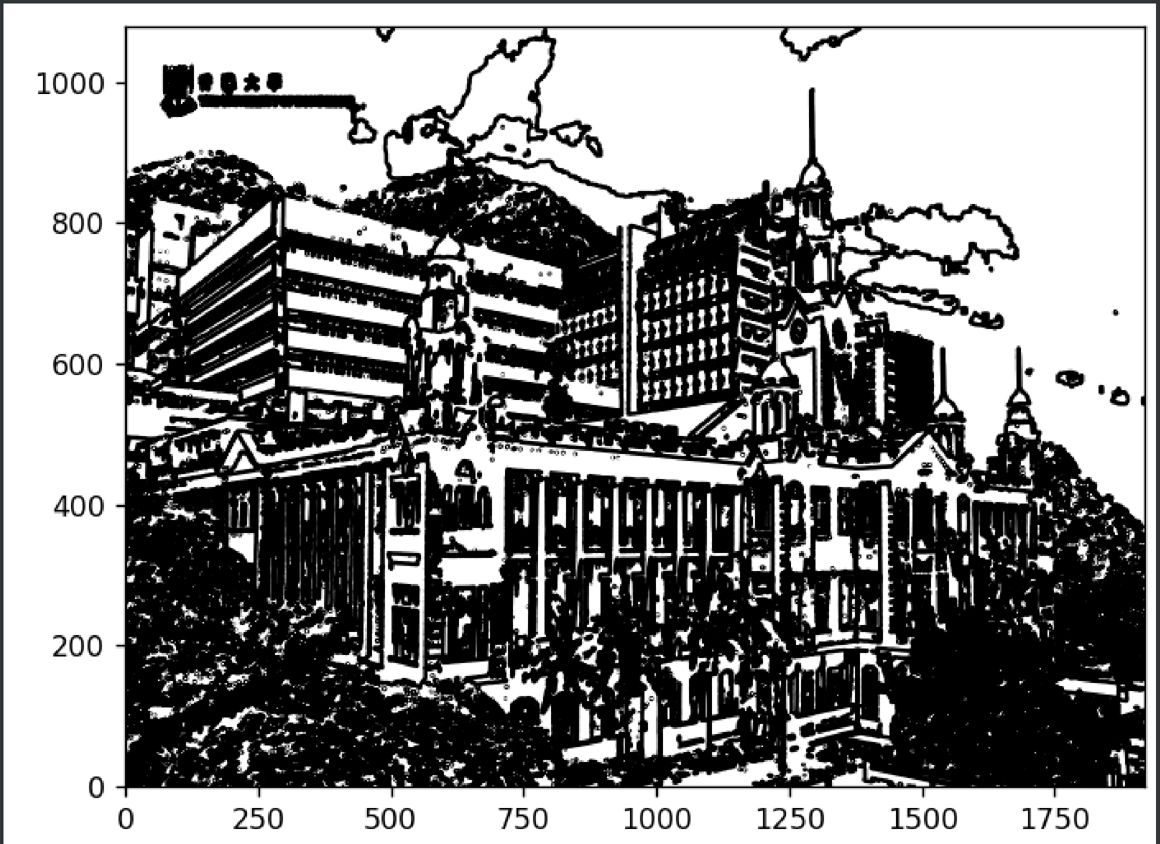
2022年02月10日 10:05　来源：新海南客户端

在街头遍布监控摄像头的时代，“图侦民警”逐渐被大众所熟悉，他们利用遍布街头巷尾的探头，通过观察、分析推演出犯罪分子的蛛丝马迹，为破案提供最直接的线索和可靠的依据。有人打比方说：“在图侦民警的手上，不会说话的摄像头成了无处不在的‘照妖镜’，科技的力量让犯罪无所遁形。”

Processing Images in Python

#9: Contour Line

How to turn your image into the following one?



Drawing Contour Line of an Image

A contour line for an image is a curve connecting all of the pixels where they have the grayscale value. We need to first convert an image into grayscale and then connect the pixels with the same grayscale intensity.

Drawing Contour Line of an Image

```python
from PIL import Image
from io import BytesIO
import requests
import matplotlib.pylab as plt
import numpy as np
url = "https://ximarketing.github.io/data/trump2.jpg"
response = requests.get(url)
image = Image.open(BytesIO(response.content))
image = image.convert('L')
plt.contour(np.flipud(image), colors ='k',
            levels = np.logspace(-15, 15, 100))
plt.show()
```

Processing Images in Python
#10: Convolution

What is convolution?

Convolution is an operation that operates on two images, one being an input image and the other one being a mask (also called the kernel) as a filter on the input image, producing an output image. It works by determining the value of a central pixel by adding the weighted values of all of its neighbors together to compute the new value of the pixel in the output image.

https://www.youtube.com/embed/yb2tPt0QVPY?enablejsapi=1

Why convolve an image?

Convolution applies a general-purpose filter effect on the input image. This is done in order to achieve various effects with appropriate kernels on an image, such as smoothing, sharpening, and embossing, and in operations such as edge detection.

Edge Detection

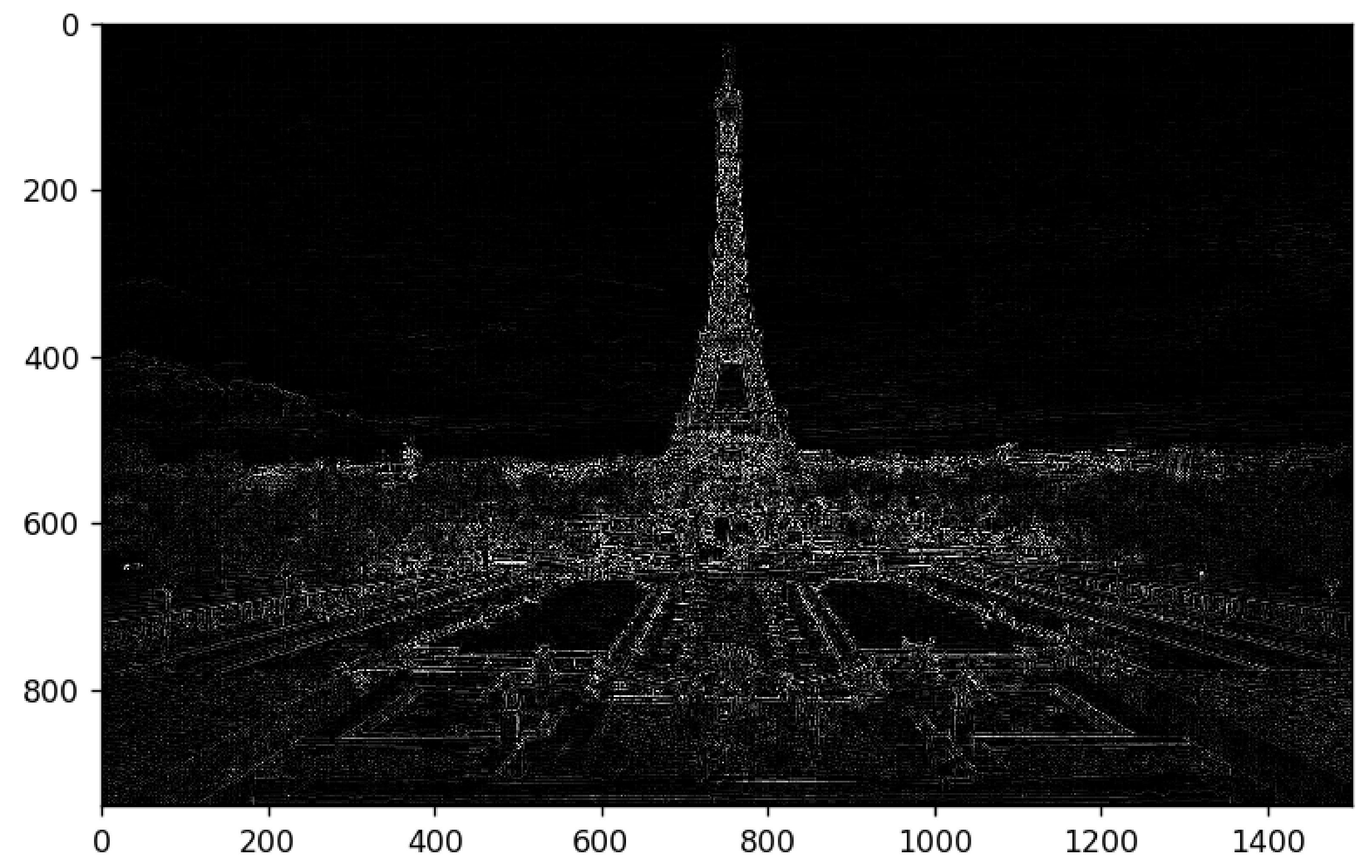
The following Laplace kernel is famous for edge detection. When applying the kernel to convolve images, you will get the edges of the original image.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Let's try the following code:

The convolve2d function allows you to convolve a 2D image.

```python
from skimage import io
from skimage.io import imshow
from scipy import signal
from skimage.color import rgb2gray
import matplotlib.pylab as pylab
import numpy as np
url = 'https://ximarketing.github.io/data/effiel.jpg'
image = rgb2gray(io.imread(url)).astype(float)
edge_laplace_kernel = np.array([[0,1,0],[1,-4,1],[0,1,0]])
image_edges = np.clip(signal.convolve2d(image, edge_laplace_kernel), 0, 1)
imshow(image_edges)
pylab.show()
```


0
200
400
600
800
0
200
400
600
800
1000
1200
1400

Emboss an Image ("图像浮雕")

The following emboss kernel will be used. When applying the kernel to convolve images, you will get the embossed verision of the original image.

$$\begin{bmatrix} -2 & -1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

Let's try the following code:

```python
from skimage import io
from skimage.io import imshow
from scipy import signal
import matplotlib.pylab as pylab
import numpy as np
url = 'https://ximarketing.github.io/data/effiel.jpg'
image = io.imread(url)/255 #
emboss_kernel = np.array([[-2,-1,0],[-1,1,1],[0,1,2]])
image_embossed = np.ones(image.shape)
for i in range(3):
    image_embossed[...,i] = np.clip(
        signal.convolve2d(image[...,i], emboss_kernel,
                          mode='same', boundary="symm"),0,1)
imshow(image_embossed)
pylab.show()
```

More Kernels...

Kernel for Sharpening ("锐化")

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Box Blur Kernel

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

0
200
400
600
800
0
200
400
600
800
1000
1200
1400

Group Project (20%)

Four Options

1. Develop a game APP which demonstrates and solves a puzzle game. It features some algorithms that we learned in class (e.g., Hanoi tower).
2. Scraping some text data yourself (to be covered later) and perform some analysis of the text. In this case, an APP is not required.
3. Develop an image APP which allows users to upload images and perform some operations on the image.
4. Develop an APP which helps users scrape data.

Submission

1. Your Python code
2. Your complete chat history with AI
3. A user manual of the APP (for options 1,3 and 4) or a report of your findings (for option 2)
4. Supplementary data files, if needed

Deadline: 23:59 PM, April 19, 2025

Next Class

Each group demonstrates the APP you have developed for the previous assignment [data analysis and visualization] to the class, followed by a peer evaluation.

1~2 minutes for each group demonstration.