

```
# Load the Drive helper and mount
from google.colab import drive

# This will prompt for authorization.
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.moun

```
import imageio as iio
import pylab
import cv2

# from documentation in: https://imageio.readthedocs.io/en/stable/

# reads from file
reader = iio.get_reader('/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/001.mp4',

#finds frames in video and appropriate spacing for 9x9 rows
cap = cv2.VideoCapture("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/001.mp4")
length = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
divs = int(length / 8)

#creates array with spaced out frames
nums = []
for i in range(length):
    if i % divs == 0:
        nums.append(i)

#iterate through frames in video
#save those
for num in nums:
    image = reader.get_data(num)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    fig = pylab.figure()
    pylab.imshow(gray)
    cv2.imwrite("/content/drive/MyDrive/CS410-Unconventional_cameras/grayshot_%s.png" % num,
pylab.show())
```

```
#original gray lab images
import numpy as np
import matplotlib.pyplot as plt

from skimage import data
from skimage.feature import match_template

import imageio as iio
```

```
# From library: https://scikit-image.org/docs/stable/auto\_examples/features\_detection/plot\_te
```

```
# importing image frames from video
```

```
image1 = plt.imread('/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray0.png')
image2 = plt.imread('/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray94.png')
image3 = plt.imread('/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray188.png')
image4 = plt.imread('/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray282.png')
image5 = plt.imread('/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray376.png')
image6 = plt.imread('/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray470.png')
image7 = plt.imread('/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray564.png')
image8 = plt.imread('/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray658.png')
image9 = plt.imread('/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray752.png')
```

```
# rough estimate for template
```

```
template = image1[965:1385,265:900]
```

```
#results for template matching
```

```
result1 = match_template(image1, template)
result2 = match_template(image2, template)
result3 = match_template(image3, template)
result4 = match_template(image4, template)
result5 = match_template(image5, template)
result6 = match_template(image6, template)
result7 = match_template(image7, template)
result8 = match_template(image8, template)
result9 = match_template(image9, template)
```

```
#creating a list of max values for the template matches
```

```
result_coord = []
```

```
result_coord.append(np.unravel_index(np.argmax(result1), result1.shape))
result_coord.append(np.unravel_index(np.argmax(result2), result2.shape))
result_coord.append(np.unravel_index(np.argmax(result3), result3.shape))
result_coord.append(np.unravel_index(np.argmax(result4), result4.shape))
result_coord.append(np.unravel_index(np.argmax(result5), result5.shape))
result_coord.append(np.unravel_index(np.argmax(result6), result6.shape))
result_coord.append(np.unravel_index(np.argmax(result7), result7.shape))
result_coord.append(np.unravel_index(np.argmax(result8), result8.shape))
result_coord.append(np.unravel_index(np.argmax(result9), result9.shape))
```

```
#these will be the x-positions
```

```
x_pos = []
```

```
for x in result_coord:
```

```
    x_pos.append(x[0])
```

```
#print(x_pos)
```

```
plt.figure(1)
```

```
plt.margins(x=0,y=0)
```

```

plt.plot(x_pos)
plt.ylabel('y-pixel position')
plt.xlabel('frame count')

#these will be the y-positions
y_pos = []
for y in result_coord:
    y_pos.append(y[1])
#print(y_pos)

plt.figure(2)
plt.margins(x=0,y=0)
plt.plot(y_pos)
plt.ylabel('x-pixel position')
plt.xlabel('frame count')

#these are the combined positions as a line plot
plt.figure(3)
plt.margins(x=0,y=0)
plt.plot(*zip(*result_coord))
plt.ylabel('x-pixel position')
plt.xlabel('y-pixel position')

#these are the combined positions as a scatter plot
plt.figure(4)
plt.margins(x=0,y=0)
plt.scatter(*zip(*result_coord))
plt.ylabel('x-pixel position')
plt.xlabel('y-pixel position')

plt.figure(5)
plt.imshow(template)

```

```

import sys
from PIL import Image

image1 = Image.open("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray0.png")
image2 = Image.open("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray94.png")
image3 = Image.open("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray188.png")
image4 = Image.open("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray282.png")
image5 = Image.open("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray376.png")
image6 = Image.open("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray470.png")
image7 = Image.open("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray564.png")
image8 = Image.open("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray658.png")
image9 = Image.open("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/gray752.png")

```

```
x = 265
```

```
y = 965
```

```
offset1 = (x-108,y-946)
```

```
offset2 = (x-445,y-1500)
```

```
offset3 = (x-328,y-1293)
```

```
offset4 = (x-45,y-1179)
```

```
offset5 = (x-283,y-642)
```

```
offset6 = (x-410,y-631)
```

```
offset7 = (x-144,y-698)
```

```
offset8 = (x-5,y-1230)
```

```
image1.paste(image2, offset1, mask = image2)
```

```
image1.paste(image6, offset5, mask = image6)
```

```
image1.paste(image7, offset6, mask = image7)
```

```
image1.paste(image8, offset7, mask = image8)
```

```
image1.paste(image3, offset2, mask = image3)
```

```
image1.paste(image4, offset3, mask = image4)
```

```
image1.paste(image5, offset4, mask = image5)
```

```
image1.paste(image9, offset8, mask = image9)
```

```
#image1.save("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/blended_inclusive.png")
```

```
#image1.save("/content/drive/MyDrive/CS410-Unconventional_cameras/Prob3/blended_selective.png")
```