

```
In [ ]: import binascii
import numpy as np
import cv2
import os
import ffmpeg
from PIL import Image
```

```
In [ ]: def strip(content):
    curx = str(content)[2:len(content)]
    return curx
```

```
In [ ]: # def form():
#     video_name = './vid.mp4'
#     images = ['1.jpg', '2.jpg']
#     fourcc = cv2.VideoWriter_fourcc(*'mp4v')
#     video = cv2.VideoWriter(video_name, fourcc, 10, (1280,720))
#     for image in images:
#         video.write(cv2.imread(image))
#     cv2.destroyAllWindows()
#     video.release()
def form_using_image_folder(image_folder):
    video_name = './vid.mp4'
    images = [img for img in os.listdir(image_folder) if img.endswith(".j
    fourcc = cv2.VideoWriter_fourcc(*'mp4v')
    video = cv2.VideoWriter(video_name, fourcc, 10, (1920,1080))
    for image in images:
        video.write(cv2.imread(os.path.join(image_folder, image)))
    cv2.destroyAllWindows()
    video.release()
```

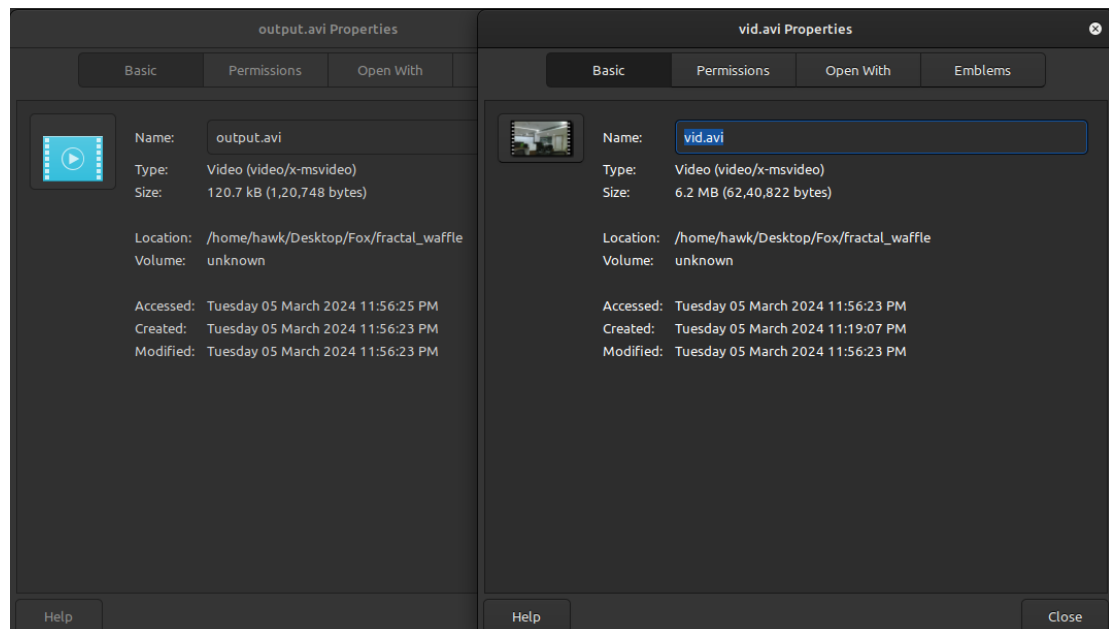
```
In [ ]: image_folder = './folder_img'
form_using_image_folder(image_folder)
```

```
In [ ]: # (
#     ffmpeg.input("vid.avi")
#     .output("output.avi",vcodec = 'h264')
#     .run()
# )
```

```
In [ ]: vid = cv2.VideoCapture('./vid.mp4')
success,image = vid.read()
c = 0
while success:
    cv2.imwrite("./out_check/Frame%05d.jpg" % c, image)
    success,image = vid.read()
    print('Reading frame: ', c)
    c = c + 1

print('done')
```

```
Reading frame: 0
Reading frame: 1
done
```



We are back to 120kB from the original 250kB images, however this time there is something different.

Somehow the frames have more information than we started with.

```
In [ ]: filename = 'output.avi'
with open(filename, 'rb') as f:
    content = f.read()
content=binascii.hexlify(content)
# print(content)
```

```
In [ ]: input_images_1 = Image.open('./out_check/Frame00000.jpg')
input_images_2 = Image.open('./folder_img/2.jpg')
pixel_map_1 = input_images_1.load()
pixel_map_2 = input_images_2.load()
point = (33,997)
a = pixel_map_1[point]
b = pixel_map_2[point]
consider_same = a == b
print(consider_same)
print(a)
print(b)
```

```
False
(98, 99, 81)
(101, 99, 86)
```

```
In [ ]: number_pixels = 1920*1080
threshold = 3
count = 0
for i in range (0,1920):
    for j in range (0,1080):
        point = (i,j)
        a = pixel_map_1[point]
        b = pixel_map_2[point]
        if ((a[0] in range (b[0]-threshold,b[0]+threshold)) & (a[1] in ra
            count = count + 1
```

```
per_equal = (count/number_pixels)*100  
print(str(per_equal)+"'% ' under threshold")
```

35.32146990740741'% ' under threshold

```
In [ ]: number_pixels = 1920*1080  
threshold = 5  
count = 0  
for i in range (0,1920):  
    for j in range (0,1080):  
        point = (i,j)  
        a = pixel_map_1[point]  
        b = pixel_map_2[point]  
        if ((a[0] in range (b[0]-threshold,b[0]+threshold)) & (a[1] in ra  
            count = count + 1  
per_equal = (count/number_pixels)*100  
print(str(per_equal)+"'% ' under threshold")
```

77.15846836419753'% ' under threshold

```
In [ ]: number_pixels = 1920*1080  
threshold = 10  
count = 0  
for i in range (0,1920):  
    for j in range (0,1080):  
        point = (i,j)  
        a = pixel_map_1[point]  
        b = pixel_map_2[point]  
        if ((a[0] in range (b[0]-threshold,b[0]+threshold)) & (a[1] in ra  
            count = count + 1  
per_equal = (count/number_pixels)*100  
print(str(per_equal)+"'% ' under threshold")
```

98.32619598765432'% ' under threshold