```
import cv2
import numpy as np
import matplotlib.pyplot as plt


def FrameCapture(path):
    # Path to video file
    vidObj = cv2.VideoCapture(path)

    # Used as counter variable
    count = 0

    # checks whether frames were extracted
    success = 1

    while success:
      try:
        # vidObj object calls read
        # function extract frames
        success, image = vidObj.read()

        # Saves the frames with frame-count
        cv2.imwrite("frame%d.jpg" % count, image)
        count += 1
      except:
        break



FrameCapture("/content/ytshorts.mp4")


def unsharp(image, sigma, strength):

    # Median filtering
    image_mf = median_filter(image, sigma)

    # Calculate the Laplacian
    lap = cv2.Laplacian(image_mf,cv2.CV_64F)

    # Calculate the sharpened image
    sharp = image-strength*lap

    # Saturate the pixels in either direction
    sharp[sharp>255] = 255
    sharp[sharp<0] = 0

    return sharp


def callunsharp(original_image):
  sharp1 = np.zeros_like(original_image)
  for i in range(3):
      sharp1[:,:,i] = unsharp(original_image[:,:,i], 5, 0.8)


import os
from os import listdir

# get the path or directory
folder_dir = "/content"
folder = '/content/processed'

count = 0
for images in os.listdir(folder_dir):
    # check if the image ends with png or jpg or jpeg
    if (images.endswith(".png") or images.endswith(".jpg")
        or images.endswith(".jpeg")):
        frame = cv2.imread(images)
        blur = cv2.medianBlur(frame,5)
        blur = cv2.bilateralFilter(blur,9,480,480)
        cv2.imwrite("outframe%d.jpg" % count, blur)
        plt.imshow(frame,cmap ='gray')
        count+=1
        # print(images)
# print(count)
```


```
img = cv2.imread('frame95.jpg')
```

```
img.shape
```

```
(480, 480, 3)
```

```
import numpy as np
import glob


frameSize = (500, 500)

out = cv2.VideoWriter('output_video.mp4',cv2.VideoWriter_fourcc(*'DIVX'), 60, frameSize)

for filename in glob.glob('/content/outframe*.jpg'):
    img = cv2.imread(filename)
    out.write(img)

out.release()


def images_to_video():
  image_folder = '/content'

  images = [img for img in os.listdir(image_folder) if img.endswith(".jpg") and img.startswith("outframe")]
  frame = cv2.imread(os.path.join(image_folder, images[0]))
  height, width, layers = frame.shape

  video = cv2.VideoWriter('project.avi',cv2.VideoWriter_fourcc(*'DIVX'), 15, (width,height))


  for image in images:
      video.write(cv2.imread(os.path.join(image_folder, image)))


  cv2.destroyAllWindows()
  video.release()


images_to_video
```

```
<function __main__.images_to_video()>
```

```
# Set the directory containing the images
image_directory = '/content'

# Get a list of all image files in the directory
image_files = [f for f in os.listdir(image_directory) if f.startswith('outframe') and f.endswith('.jpg')]
image_files.sort()  # Sort the image files

# Set the output video path
output_video_path = '/content/output_video.mp4'

# Get the first image to determine the video dimensions
first_image = cv2.imread(os.path.join(image_directory, image_files[0]))
height, width, layers = first_image.shape

# Create a VideoWriter object to write the video
fourcc = cv2.VideoWriter_fourcc(*'mp4v')  # Codec for the output video (can change to other codecs)
out = cv2.VideoWriter(output_video_path, fourcc, 30, (width, height))  # 30 frames per second

# Loop through the image files and add them to the video
```

```
for image_file in image_files:
    image_path = os.path.join(image_directory, image_file)
    image = cv2.imread(image_path)
    out.write(image)

# Release the VideoWriter and close all windows
out.release()
cv2.destroyAllWindows()

print("Video creation complete.")
```

```
Video creation complete.
```

```
image_files
```

```
['outframe0.jpg',
 'outframe1.jpg',
 'outframe10.jpg',
 'outframe100.jpg',
 'outframe101.jpg',
 'outframe102.jpg',
 'outframe103.jpg',
 'outframe104.jpg',
 'outframe105.jpg',
 'outframe106.jpg',
 'outframe107.jpg',
 'outframe108.jpg',
 'outframe109.jpg',
 'outframe11.jpg',
 'outframe110.jpg',
 'outframe111.jpg',
 'outframe112.jpg',
 'outframe113.jpg',
 'outframe114.jpg',
 'outframe115.jpg',
 'outframe116.jpg',
 'outframe117.jpg',
 'outframe118.jpg',
 'outframe12.jpg',
 'outframe13.jpg',
 'outframe14.jpg',
 'outframe15.jpg',
 'outframe16.jpg',
 'outframe17.jpg',
 'outframe18.jpg',
 'outframe19.jpg',
 'outframe2.jpg',
 'outframe20.jpg',
 'outframe21.jpg',
 'outframe22.jpg',
 'outframe23.jpg',
 'outframe24.jpg',
 'outframe25.jpg',
 'outframe26.jpg',
 'outframe27.jpg',
 'outframe28.jpg',
 'outframe29.jpg',
 'outframe3.jpg',
 'outframe30.jpg',
 'outframe31.jpg',
 'outframe32.jpg',
 'outframe33.jpg',
 'outframe34.jpg',
 'outframe35.jpg',
 'outframe36.jpg',
 'outframe37.jpg',
 'outframe38.jpg',
 'outframe39.jpg',
 'outframe4.jpg',
 'outframe40.jpg',
 'outframe41.jpg',
 'outframe42.jpg',
 'outframe43.jpg',
```

✓ 0s completed at 9:12 PM