

CV ASSIGNMENT 0

Siya Puttagunta

2021101062

1. Video ↔ Images

Question

Write a program to convert a given video to its constituent images. Your output should be in a specified folder.

Solution

cv2.VideoCapture is used to open the input video file. Video properties are obtained using video_capture.get(). The property required for this particular task is retrieved as follows:

- Total no of frames: video_capture.get(7)

The code iterates through each frame in the video and the current frame is read using video_capture.read(). Image path for the current frame is constructed and it is saved as an image using cv2.imwrite. All the images are saved in the output folder.

Challenges Faced

- Deciding how many frames to extract from the video and getting the total no of frames.

Code

```
import cv2
import os

input_video_path = "vid1.mp4"
output_folder = "out1"

video_capture = cv2.VideoCapture(input_video_path)
total_no_of_frames = int(video_capture.get(7))

if not os.path.exists(output_folder):
    os.makedirs(output_folder)

for frame_number in range(total_no_of_frames):
    ret, frame = video_capture.read()
    if not ret:
        break

    image_path = os.path.join(output_folder, f"frame_{frame_number:04d}.jpg")
    print(f"Frame {frame_number + 1}/{total_no_of_frames} saved at {image_path}")
    cv2.imwrite(image_path, frame)

video_capture.release()
```

Question

Write a program that will merge a set of images in a folder into a single video. You should be able to control the frame rate in the video that is created.

Solution

`convert_images_to_video()` is a function that takes the path to a folder containing images, the desired output video path and a frame rate which can be controlled as parameters.

A list of image filenames are retrieved from the specified input image folder and it is sorted to ensure the correct sequence of images. Frame dimensions such as width, height, and number of colour channels are determined.

A video writer object is initialized using `cv2.VideoWriter` with the specified output video path, video codec, frame rate and frame dimensions. Frames are added to the video by iterating through the sorted list of image filenames, reading each image and then writing it to the video writer object.

Challenges Faced

- Choosing the appropriate frame rate so that it matches the original video, ensuring that all images have the same resolution and format.

Code

```
import cv2
import os

def convert_list_to_video(image_folder, output_video_path, frame_rate=25):
    images_list = [img for img in os.listdir(image_folder) if img.endswith(".jpg")]
    images_list.sort()

    image_path = os.path.join(image_folder, images_list[0])
    frame = cv2.imread(image_path)
    height, width, layers = frame.shape

    video_writer = cv2.VideoWriter(output_video_path, cv2.VideoWriter_fourcc(*"mp4v"), frame_rate, (width, height))

    for image in images_list:
        image_path = os.path.join(image_folder, image)
        frame = cv2.imread(image_path)
        video_writer.write(frame)

    video_writer.release()

input_list_folder = "out1"
output_video_path = "vid1_dup3.mp4"

convert_list_to_video(input_list_folder, output_video_path)
```

2. Capturing Images

Question

Learn how to capture frames from a webcam connected to your computer and save them as images in a folder. You may use either the built-in camera of your laptop or an external one connected through USB. You should also be able to display the frames (the video) on the screen while capturing.

Solution

Webcam is opened using `cv2.VideoCapture(0)`, which is set to use the default camera. Frames are continuously captured and displayed from the webcam. Each captured frame is saved as a separate image in the specified output folder.

Challenges Faced

- Determining an appropriate frame rate for capturing images.
- Selecting the appropriate camera port.

Code

```
import cv2
import os

output_folder = "captured_images"
os.makedirs(output_folder, exist_ok=True)

cap = cv2.VideoCapture(0)
if not cap.isOpened():
    print("Error: Couldn't open the webcam.")
    exit()

while True:
    ret, frame = cap.read()
    cv2.imshow('Webcam', frame)

    image_path = os.path.join(output_folder, f"captured_frame_{len(os.listdir(output_folder)) + 1}.png")
    cv2.imwrite(image_path, frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

3. Chroma Keying

Question

Create an interesting composite of two videos using this technique.

Solution

Both the video files are loaded using `cv2.VideoCapture`. Let `video1` be the video with a green background and `video2` be the background video. Video properties such as width, height and frames per second (fps) of the videos are retrieved using `cv2.get()`. A `VideoWriter` object is initialized to write the output video. The output video will have the same resolution and fps as the input videos. A green color range is defined using arrays (`u_green` and `l_green`), representing the upper and lower bounds of the green color in the HSV color space.

Frames from both the videos are read and resized to have the same dimensions. A mask for the green background in the first video is created using `cv2.inRange`. A mask for the non-green part is created using `cv2.inRange`. The green background (background) from the first frame and the non-green part (foreground) from the second frame are extracted using `cv2.bitwise_and`.

The foreground and background are combined using `cv2.add` to create the composite frame which is written to the output video using `out.write`.

Challenges Faced

- Took a lot of time to decide the right lower and upper ranges for green colour. Incorrect ranges led to errors in the output video (such as not extracting the non-green colour part in `video1` properly, not getting fine edges correctly).
- Choosing appropriate videos suitable for the task.

Code

```
import cv2
import numpy as np

video1 = cv2.VideoCapture('green.mp4')
video2 = cv2.VideoCapture('bg.mp4')

width = int(video1.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(video1.get(cv2.CAP_PROP_FRAME_HEIGHT))
fps = int(video1.get(cv2.CAP_PROP_FPS))

out = cv2.VideoWriter('output.mp4', cv2.VideoWriter_fourcc(*'mp4v'), fps, (width, height))

u_green = np.array([140, 255, 140])
l_green = np.array([0, 100, 0])

while True:
    ret1, frame1 = video1.read()
    ret2, frame2 = video2.read()

    if not ret1 or not ret2:
        break

    frame1 = cv2.resize(frame1, (width, height))
    frame2 = cv2.resize(frame2, (width, height))

    mask = cv2.inRange(frame1, l_green, u_green)
    mask_inv = cv2.bitwise_not(mask)

    background = cv2.bitwise_and(frame1, frame1, mask=mask_inv)
    foreground = cv2.bitwise_and(frame2, frame2, mask=mask)

    composite_frame = cv2.add(foreground, background)

    out.write(composite_frame)
    cv2.imshow('Composite Video', composite_frame)

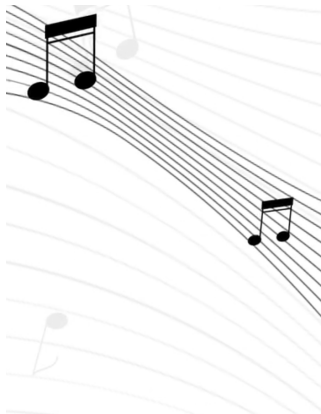
    if cv2.waitKey(25) & 0xFF == ord('q'):
        break

video1.release()
video2.release()
out.release()
cv2.destroyAllWindows()
```

Video with green background



Background Video



Output Video



Link to all code files and outputs

https://iiitaphyd-my.sharepoint.com/:f/g/personal/siya_puttagunta_students_iiit_ac_in/EkXt2m5i1bFLmzYW1VFGUP8BUT0zZWn85Q9Fr6NTRFqGce=Ns5bDu