

EE604: Assignment 1

Prof. Tushar Sandhan
sandhan@iitk.ac.in

Semester-I, 2023

Due date: 15th Sept, 2023
Due time: 11:59PM

Weight: 15%
Submission: MookIT

Introduction

You need to figure out which methods and algorithms are suitable for solving below real-world problems. Apart from this assignment, you are always welcome to practice basic image processing algorithms by your own. Obviously, no one is stopping you for that. So, go forth and pixelate, morph, and filter to your heart's content.

1 *Jai Hind*: dronepositioner [5%]

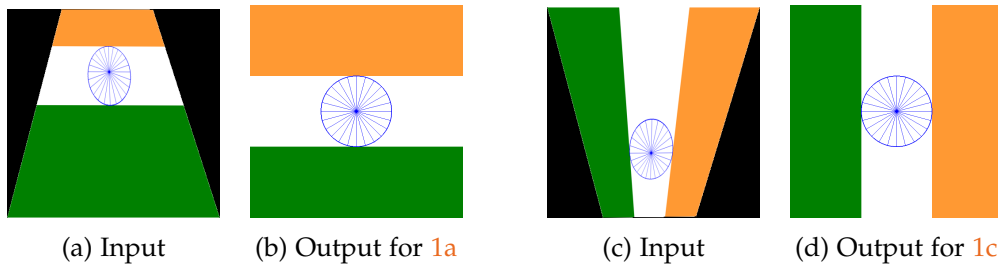


Figure 1: Two sample examples showing inputs and corresponding outputs.

The national flag holds significant cultural and symbolic importance for a nation. It serves as a powerful emblem that embodies the history, values, unity of a country via evoking a sense of patriotism and pride. Flags are more than mere pieces of cloth; they are visual representations of a nation's identity.

During aerial maneuvers, UAV or drones have to position themselves by controlling tilt and height. Experimental robotics use ArUco markers (you can consider them as QR codes) which are placed on the grounds for adjusting their position. However, fixed size visual patterns can be used in remote terrain for drone positioning. Indian flag serves as a good localizer by military drones. At the current non-optimal position, drone sees the image as shown in 'input', then it adjusts its position (tilt and height) continuously with minimum movement till it sees the final 'reference' image. Note there is no need of rotation along Z-(height)-axis (yaw). Your task is to generate the 'reference' image for the input seen by drone. Few examples are shown in Fig. 1.

Your function should take input as an image array and it should return the output as an image array. Here input size may vary but each reference image is having a size 600x600 pixels, radius of circle = 100, center of circle is (300,300), circle width = 2 and spoke width = 1 pixel. You will be given the python program, where you have to complete the function while returning the proper reference image for each input image. There will be 10 to 20 test cases used for automatic evaluation of your program. Think about all corner cases.

2 *eent ka jawab image se* : brickquality [5%]

If you punch your hostel room's wall in distress or in over-excitement, you might create a window there. That would be a very high-quality punch; on the other side, there are poor-quality bricks lying. There are four qualities of bricks used in the construction, having different costs according to their quality. Sounds interesting? So they do, via making different sounds when struck together. We will consider only 2 quality bricks here - low and high. Properly cooked red soil bricks are of high quality. The furnace melts ferrites and imparts a metal-like quality to these bricks. Therefore, when high-quality bricks are struck together, they produce a sound resembling that of metal, while low-quality bricks sound like cardboard banged together.

You will be provided with audio (.mp3 files) of these brick sounds. You can analyze it in the image domain via audio-to-spectrogram conversion using windowed Fourier transforms. You may use various python libraries for getting spectrogram, like the below pseudo-code:

```
: import librosa
: y, sr = librosa.load(audio_path, sr=None) %sr=None preserves sampling rate
: n_fft = 2048 %FFT points, adjust as you need
: hop_length = 512 %Sliding amount for windowed FFT (adjust as needed)
: spec = librosa.feature.melspectrogram(y, sr, n_fft, hop_length, fmax=22000)
: # spec_db = perform power to decibel (db) image transformation of 'spec'
: # decibel conversion  $\rightarrow 10\log(\frac{P}{\max(P)})$ , where  $P$  is the spectral power
: # inspect spec_db images via displaying them before designing your algorithm
```

Write the python program, which can process the brick sounds and outputs their quality in terms of metal (high) or cardboard (low) quality. Some representative sample inputs are given in Fig. 2. Note your program will be tested for 10 to 20 different sample inputs, where program should produce only one class output (either metal or cardboard) per sample testing.

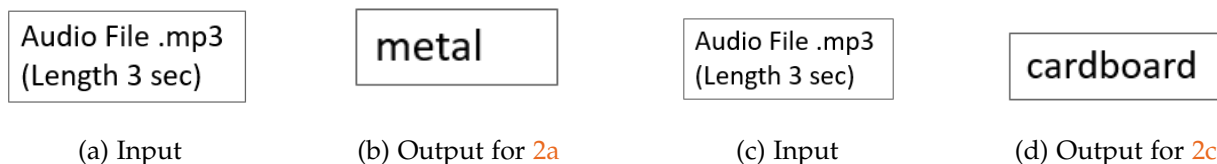


Figure 2: Sample inputs and outputs for both the classes 'metal' and 'cardboard'. Your function should take the audio input and it should return the class name as either 'metal' or 'cardboard'.

3 *Anuprasth Drishyam*: horizontalviewer [5%]

Indian ancient Sanskrit literature holds a wealth of knowledge, encrypted in the language that modern society is not fluent in. To digitize this knowledge, we need to scan the images and then use optical character recognition (OCR), an image processing algorithms to digitize it. However, many times, the scanned images are not horizontally aligned, which causes OCR to fail or produce erroneous digitization. These scanned images need to be re-rotated so that the final text appears perfectly aligned horizontally in a readable form.

Design a python program to realign scanned text always in horizontal viewing manner. Fig. 3 shows some of inputs and corresponding output for these images. Your program will be tested on various different input images where in the output any text character should not be cut and image should be horizontally aligned. Output image size can be different than input image size.

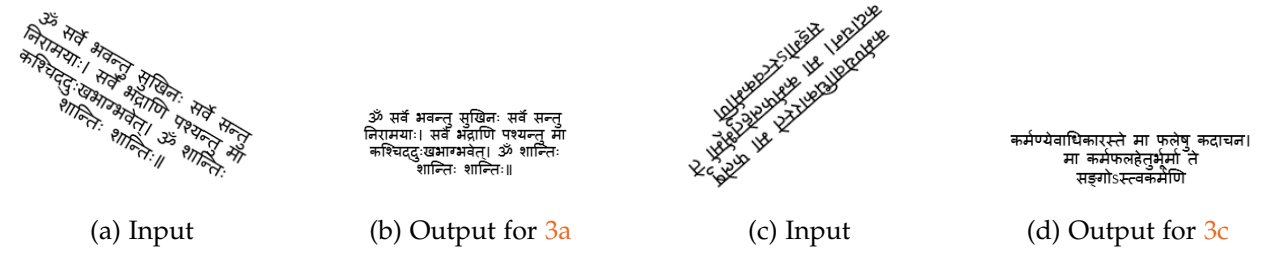


Figure 3: Sample inputs and outputs. Your function should take input as an image complete file path (string) and it should write the output in the same working directory as .jpg image.

Submission

For each question use the original images that are provided to you separately apart from this document. The only purpose of the figures here is to help with the proper explanations.

If your program does not produce output and throws errors, then you will receive a score of 0 for that question. You alone are accountable for the submitted program's proper operation. Kindly make sure it executes and outputs just the things which are being asked for.

Depending on the output quality, partial to full credit will be awarded if the program executes correctly. For plagiarized responses, even if your program does not execute or unexpectedly produces correct answer, in all cases you will be awarded full marks with an honorarium prefix as a negative sign.

Compress only the files requested into rollno.zip (eg.201234.zip) and upload it.

