CS 771 HW0 Report

Name: Debarshi Deka
ID: 9083351164

3.1 Setup the Computing Environment

All necessary packages have been installed and the list can be found in *./results/packages.txt*

3.2 Image Processing

Code for the following methods can be found in *./code/student_code.py*

Image Resizing:

There are two cases depending on the argument provided while calling the class.

1. If it's an 'int', smaller side has been adjusted to maintain the aspect ration which results in two cases i.e. height > width and height <= width.
2. If it's a 'tuple', image is resized to the provided tuple.

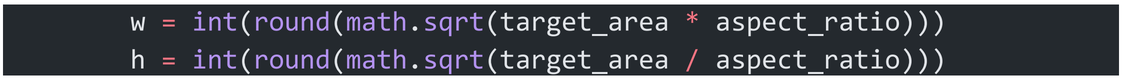
Method internally uses *cv2.resize* method.

Image Cropping:

There are two blocks depending on the success in the specified number of trials.

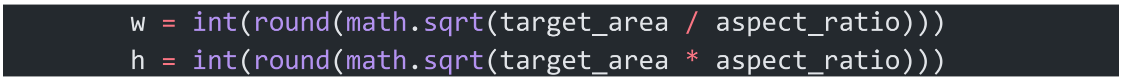
If it's a success, there are two cases possible:

- If aspect ratio <= 1,



```
        w = int(round(math.sqrt(target_area * aspect_ratio)))
        h = int(round(math.sqrt(target_area / aspect_ratio)))
```

- If aspect ratio > 1,



```
        w = int(round(math.sqrt(target_area / aspect_ratio)))
        h = int(round(math.sqrt(target_area * aspect_ratio)))
```

- After that, we crop the image and resize using the above-mentioned class.

If it's a failure, the default is to crop the patch in the centre with a square sized output depending on the argument passed i.e., 'int' or 'tuple'

Colour Jitters:

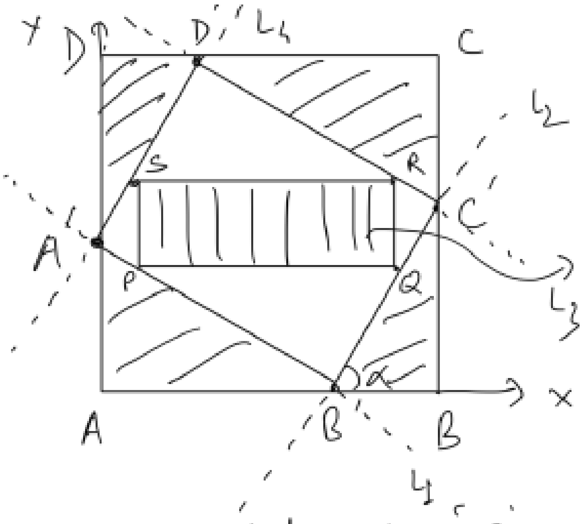
Alphas are randomly initialized for each channel, so we have 3 alphas. They are element-wise multiplied to their corresponding channels. We ensure that the values fit to uint8 datatype.

Rotation:

We first rotate using the in-built rotation matrix and warp affine methods. After that we handle the bonus part which involves cropping the max area within the image to get rid of the black pixels.

See derivation in the next page.

The computed coordinates are used to crop out the max area and then Scale class is used to resize them.



Need to find this

Let angle of rotation is 'α'

Height = h , width = w

We need to determine the coordinates of PQRS which is the intercection of lines L_1, L_3, L_2, L_4

Coordinates of $A'B'C'D'$

$A'(0, h\cos\alpha)$ $\quad$ $D'(w\cos\alpha, h\cos\alpha + w\sin\alpha)$

$B'(h\sin\alpha, 0)$ $\quad$ $C'(h\sin\alpha + w\cos\alpha, w\sin\alpha)$

Equations of L1, L2, L_3 L_4 :

$m = \frac{-h\cos\alpha}{h\sin\alpha} \Rightarrow L1: \; y = \frac{-h\cos\alpha}{h\sin\alpha}(x - h\sin\alpha)$

$m = \frac{w\sin\alpha}{w\cos\alpha} \Rightarrow L2: \; y = \frac{w\sin\alpha}{w\cos\alpha}(x - h\sin\alpha)$

$m = \frac{-h\cos\alpha}{h\sin\alpha} \Rightarrow L3: \; y - w\sin\alpha = \frac{-h\cos\alpha}{h\sin\alpha}(x - h\sin\alpha - w\cos\alpha)$

$m = \frac{w\sin\alpha}{w\cos\alpha} \Rightarrow L4: \; y - h\cos\alpha = \frac{w\sin\alpha}{w\cos\alpha}(x)$

Now, we find intersections of L1, L2, L3, L4 and we crop out that region & resize it.

3.3 Data Augmentation and Input Pipeline

Loader output images can be found in *./results/*

Attaching them in the writeup as well as per the assignment requirements.