Assignment 1: Augmentation system project documentation

**Geometrical Transformations**

**Rotation (rotation)**

Rotates the given image by the input angle (in degrees) in the counter-clockwise direction. If a negative angle value is provided, the rotation will be made in the clockwise direction.

**Parameters**

angle : float - angle of rotation in degrees. Must be in [-180,180] range

\*After rotate transformation, some pixels will end up blank

**Flip (flip)**

Flips the given image on a given axis of the image (vertical, horizontal or both).

**Parameters**

flip\_code : int

0 means flipping around the x-axis (height axis).

Positive value (for example, 1) means flipping around y-axis (width axis).

Negative value (for example, -1) means flipping around both axes.

**Shift (shift)**

Shifts the given image on a given axis (vertical, horizontal) by a shift range calculated along the axis. A shift to an image means moving all pixels of the image in one direction, such as horizontally or vertically, while keeping the image dimensions the same.

**Parameters**

axis : int - Axis along which pixels are shifted

0 means shifting around x-axis (height axis).

1 means shifting around y-axis (width axis).

shift\_range : float - represents how much (proportionally to the input axis size ) the image will be shifted.

Must be in [0,1] range.

\*After shifting, the lower edge pixels (corresponding to the shifted axis) will end up blank

**Zoom (zoom)**

Center zoom in/out of the given image and returning an enlarged/shrinked view of the image without changing dimensions.

**Parameters**

zoom\_factor : float – amount of zoom as a ratio (0 to Inf)

<1 means zooming out the given image

>1 means zooming in

\*Zooming out will downscale the image which will have the effect of blanking out the remaining pixels of the image

**Random Crop (random\_crop)**

Applies random cropping to the given image.

**Parameters**

height\_range : float - How much of the given image should be cropped horizontally. Must be in [0,1] interval

width\_range : float - How much of the given image should be cropped vertically. Must be in [0,1] interval

**Pixel-level Transformations**

**Random brightness (random\_brigthness)**

Randomly brighten the given image. The intent is to allow a model to generalize across images trained on different lighting levels.

**Parameters**

brightness\_range : array of 2 integers - specifies the range from within the brightness value (in pixels) should be chosen. -127 < brightness < +127 is good range for brightness values, therefore [-127,127] is the most appropriate range value to choose from

**Gamma correction (adjust\_gamma)**

Image gamma correction by the given gamma value

**Parameters**

gamma: float – specifies the gamma value to be mapped for each pixel value.

Must be in [0,2] range. Where 1 means that no gamma correction is applied.

**Contrast (contrast)**

Adjusts the contrast of the given image.

**Parameters**

contrast\_factor: float – specifies how much to adjust the contrast. Can be any non negative number. 0 gives a solid gray image, 1 gives the original image while 2 increases the contrast by a factor of 2.

**Gaussian blur (gaussian\_blur)**

Applies gaussian blur to the given image.

**Parameters**

kernel: int that represents the size of the squared kernel window. HAS to be an odd integer.

The bigger the integer value, the blurrier the image becomes

**Configuration file**

Configuration file is implemented using JSON, a dictionary-like format. The idea is that a highest level entry (key-value pair) in this dictionary-like format represents **one** transformation or **one** chain of transformations to be applied for each image. We therefore have as a **key** a name (can be any name) that has to be different from any other high-level keys in order to not have duplicate keys in the file. This approach is used due to the fact that there may be the case where a transformation method has to be applied more than once but with different values at the same image. As **value**, each entry (transformation) receives an object that contains one or more transformation methods and their corresponding parameters passed as subsequent object of key-value pairs. If a single transformation method and its corresponding valid parameters is present in one entry then that single transformation will be applied to the image. But, if there are multiple transformation methods and their corresponding valid parameters present in one entry, then all those transformations will be applied sequentially for the same image in the order of their listed appearance in the entry.

Example:

{

"f1":{

"rotation":{"angle":30},

"gaussian\_blur":{"kernel\_size":7}

},

"func2":{

"flip":{"flip\_code":1}

},

"fct3":{

"shift":{"axis":1,"shift\_range":0.2},

"zoom":{"zoom\_factor":1.3}

},

"func6":{

"random\_brightness":{"brightness\_range":[0,60]}

},

"f123":{

"contrast":{"contrast\_factor":0.7}

},

"f7":{

"adjust\_gamma":{"gamma":2.0},

}

}