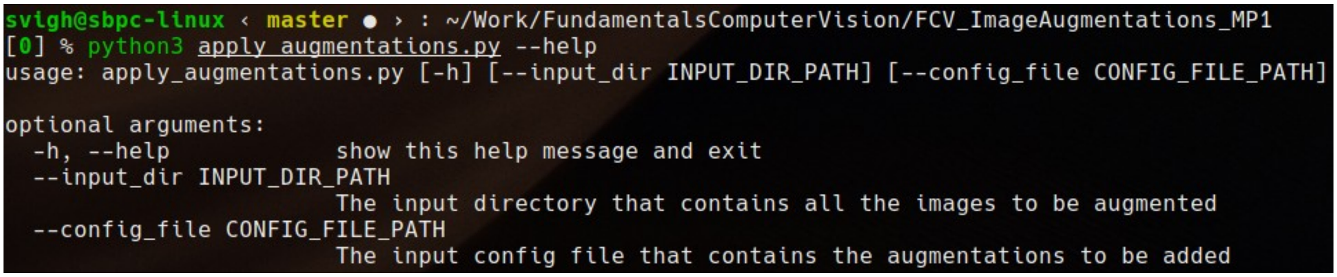
FCV ImageAugmentations

The program *apply_augmentations.py* is using OpenCV to apply a number of augmentations on the input images. Usage of the program is described by passing the –**help** command line argument when running the script.

The help argument shows the available optional arguments of the program in case the user wants to pass the input directory and config file directly at the script call, otherwise, if no arguments are given, one tkinter window will pop up for each of the parameters:



```
svigh@sbpc-linux ‹ master ● › : ~/Work/FundamentalsComputerVision/FCV_ImageAugmentations_MP1
[0] % python3 apply_augmentations.py --help
usage: apply_augmentations.py [-h] [--input_dir INPUT_DIR_PATH] [--config_file CONFIG_FILE_PATH]

optional arguments:
  -h, --help            show this help message and exit
  --input_dir INPUT_DIR_PATH
                        The input directory that contains all the images to be augmented
  --config_file CONFIG_FILE_PATH
                        The input config file that contains the augmentations to be added
```

Figure 1: Parameters overview

The program expects to receive a directory which has any number of images in it which will serve as bases on which to apply the augmentations, and a config file. The config file should have the following rules:

- each line represents an augmentation or augmentations chain;
- one augmentation represents the name of the operation followed by its parameters (some operations can accept multiple parameters), separated with spaces;
- augmentation chains can be created by having multiple augmentations on the same line, each augmentation and its parameters being separated by a ‘;’ character;
- if any operation is not recognised in a chain or single augmentation line then that line is completely ignored, even if that operation is the last one in a long chain of augmentaions. This also allows for “commenting” of lines which are not to be considered with a ‘#’ character at the beginning of the line.
- accepted operations in the config file:
  - rotate (the same with rotate_keep_size) keeps the original resolution and rotates the original image, rotate_crop (crops the image to eliminate the black corners resulted from the rotate), rotate_resize creates a bigger bounding box to fit the corners of the rotated image. All the rotates accept only the degrees parameter. E.g: rotate_crop 15
  - tint increases the channel given by the amount given, tint_abs sets the channels value to the given one. The parameters format should be channelXX where channel can be red, green and blue and XX represents any. E.g: tint red20 blue100 green10

- rescale receives two values, width and height respectivelly, and it rescales the images to have the given sizes. E.g: rescale 1024 1024
- flip will flip the images horizontally or vertically acording to the parameters given. As many parameterrs can be given to flip. E.g: flip horizontal vertical
- brighten will brighten or darken the image by a multiplier amount. E.g: brighten 0.9
- blur receives a parameter which represents the size of the gausian kernel used to blur the image which represents the intensity of the blur. E.g: blur 3
- noise receives two parameters which represent the lower and upper limits of the randomised values in the noise array. It creates a noise array of equal size with the image and it multiplies each value from the original image with the noise. E.g: noise 0.7 1.5
- CHAINING EXAMPLE: tint red100; rotate_resize 10; flip horizontal

All the operations fallback to some default values in case the parameters are not given by mistake.

The example config file has all the implemented augmentations inside it, all of them are “commented” out and need the ‘#’ character removed for each line that is wanted to be applied. When adding new lines it does not matter the capitalisation of the operations and can be written in any way as long as the words are correct.