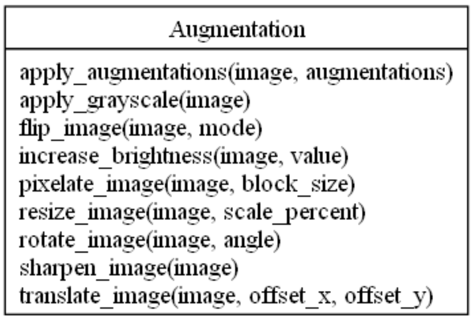
Augmentation system for CV ML algorithms

Sîrbu Radu-Mihai

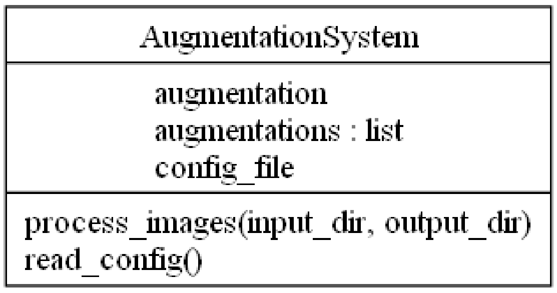
1. Design

The application is mostly built by utilizing two main classes: ***AugmentationSystem*** and ***Augmentation.*** AugmentationSystem deals with processing the images in the test folder and reading the config file, and utilizes sthe Augmentation class. Augmentation encapsulates the augmentation methods.



Augmentation

apply_augmentations(image, augmentations)
apply_grayscale(image)
flip_image(image, mode)
increase_brightness(image, value)
pixelate_image(image, block_size)
resize_image(image, scale_percent)
rotate_image(image, angle)
sharpen_image(image)
translate_image(image, offset_x, offset_y)



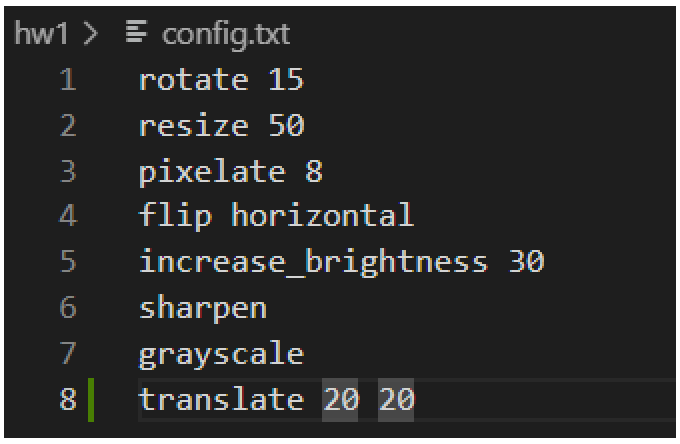
AugmentationSystem
augmentation augmentations : list config_file

process_images(input_dir, output_dir)
read_config()

These classes utilize the **numpy** and **opencv** python labraries and on top of that the **tkinter** library for enabling file selection.

2. Config file

The config file is **config.txt** and can contain multiple lines. Each line contains the augmentation method and an additional parameter (or two if it is the case of translation). Methods are: **rotate, resize, pixelate, flip horizontal/vertical, increase_brightness, sharpen, grayscale, translate.**



```
rotate 15
resize 50
pixelate 8
flip horizontal
increase_brightness 30
sharpen
grayscale
translate 20 20
```

Each augmentation method (config line) is applied to each test image and leads to a generated output.

3. Algorithms implementation
   i. Rotation: image rotation using a 2D rotation matrix and opencv library.
   ii. Resize: image resizing using the opencv library.
   iii. Flip: horizontal & vertical image flipping using numpy arrays.
   iv. Translation: low level pixel transformation, translation using offsets.

v. Pixelation: image pixelation using the opencv library.
vi. Brightness: adding brightness using the opencv library.
vii. Sharpening: applying a 2D kernel using the opencv library.
viii. Grayscale: low level pixel transformation

4. Project testing

Initial image

Sharpening

Rotation

Pixelation

Grayscale

Brightness

Translation

Horizontal flipping

Initial image

Sharpening

Rotation

Pixelation

Grayscale

Brightness

Translation

Horizontal flipping