Augmentation System Project Documentation

Efrem Dragoș-Sebastian-Mihaly

1 Framework Overview

The image augmentation framework is designed to process images using a configurable set of operations. Key components include:

- **Main script** (`augment.py`): Handles image loading, processing, and saving.
- **Configuration file** (`config.txt`): Specifies augmentation operations.
- **Augmentation functions**: Implement various image processing algorithms.

Workflow:

1. Load configuration file
2. Select input directory
3. Process each image in the directory
4. Apply specified augmentations
5. Save augmented images in a new directory

To use the framework:

1. Prepare a `config.txt` file with desired augmentations.
2. Run `augment.py`.
3. Select the input directory when prompted.
4. Augmented images will be saved in a new directory with "_aug" suffix.

2 Config File Structure

The configuration file uses a simple, line-based format:

```
operation1,param1,param2;operation2,param1
operation3,param1
```

Each line represents a set of operations applied sequentially. Multiple operations on a single line are separated by semicolons. Operation parameters are comma-separated.
Example config file:

```
brightness,1.5;rotation,30
blur,20,10.0
flip,1
scale,0.5,0.5
noise,0,25;channel_shift,2,50
shear,0.2
rotation,-30;contrast,1.5;brightness,0.8
```

3 Implemented Algorithms

- Brightness adjustment
- Color channel shift
- Gaussian noise
- Gaussian blur
- Scaling
- Flipping
- Shearing
- Rotation
- Contrast adjustment

4 Algorithm Specifications

Table 1: Image Augmentation Operations

Operation	Parameters	Description
brightness	factor	Multiplies pixel values by factor
channel_shift	channel, value	Adds value to specified channel
noise	mean, std	Adds Gaussian noise
blur	kernel_size, sigma	Applies Gaussian blur
scale	scale_x, scale_y	Resizes image
flip	flip_code	Flips image (1: horizontal, 0: vertical, -1: both)
shear	shear_factor	Applies shear transformation
rotation	angle	Rotates image by specified angle (degrees)
contrast	alpha	Adjusts image contrast

5 Implementation Details

5.1 Custom Implementations

The framework includes custom implementations for several operations:

- **Brightness Adjustment**: Scales pixel values by a specified factor to increase or decrease brightness while preserving color balance.
- **Color Channel Shift**: Adds a constant value to a specified color channel (red, green, or blue), allowing for color tint adjustments or corrections.
- **Gaussian Noise Addition**: Adds random noise based on a Gaussian distribution with specified mean and standard deviation to simulate realistic capture noise.
- **Gaussian Blur**: Uses a custom 2D convolution implementation.
- **Scaling**: Implements nearest-neighbor interpolation.
- **Flipping**: Custom implementation for horizontal, vertical, and both-axis flips.

5.2 Examples

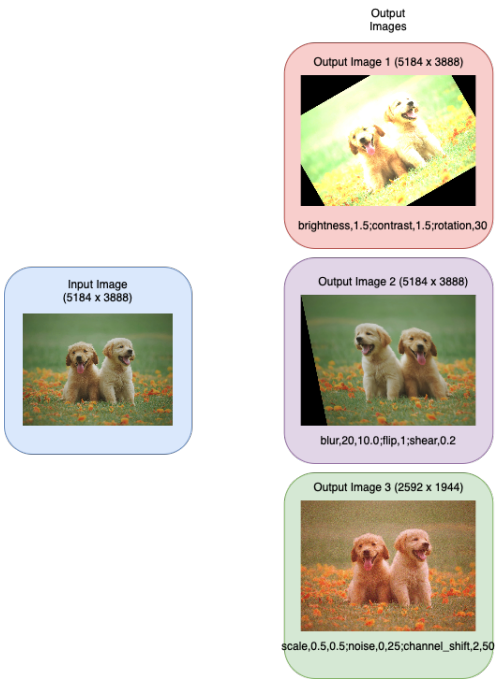

Figure 1: Image Augmentation Process

This diagram illustrates the steps involved in my image augmentation framework, from input image to the various augmented outputs.