HW2 Report– Image Analytics 1

1. Part 1
2. Generate 100 images
   1. Rotating, shifting, and scaling are utilized for this question
   2. Results
      1. Rotating

Chart, pie chart

Description automatically generated

* + 1. Shifting

A picture containing text

Description automatically generated

* + 1. Scaling

A close up of a leaf

Description automatically generated

1. Obtain local patches
   1. An image was divided 9 local patches (3 by 3)
      1. Result

A close-up of a leaf

Description automatically generated with medium confidence

1. Pre-whitening (using ZCA whitening) the patches)
   1. Result

A picture containing polygon

Description automatically generated

1. Determine the channel-by-channel distribution of the prewhitened images
   1. A sample of images was picked up for this result
      1. Result

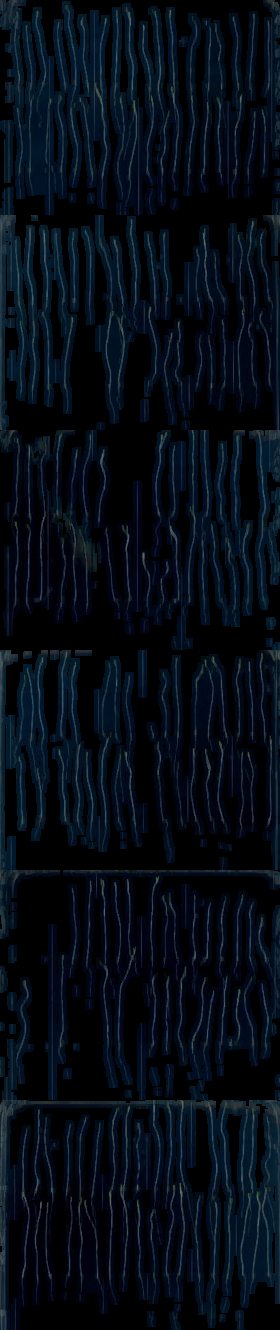
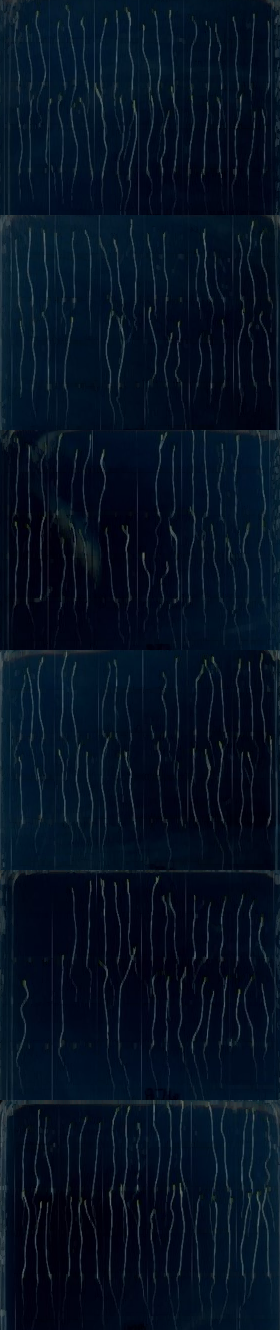
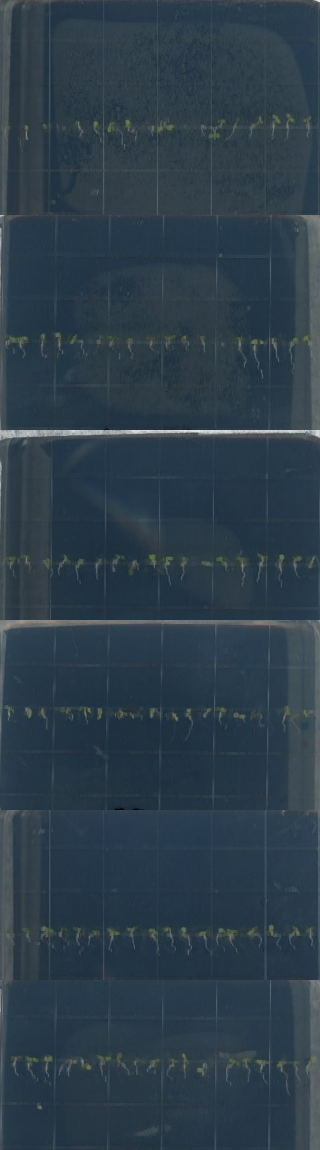
Chart

Description automatically generated

1. Determine the channel-by-channel distribution of the original images
   1. A sample of images was picked up for this result
      1. Result

Chart

Description automatically generated

1. Part 2
   * Foreground extraction from 1.jpg
   * Foreground extraction from 2.jpg
2. Discussion

Data augmentation is useful to improve performance of a machine learning model. In 1.1 we showed how we can use different techniques to augment the training dataset. In part 2, it is shown how local patches of arbitrary size can be obtained. For patching we used extract\_patches\_2d function from sklearn library.

The goal of whitening is to make the input less redundant. Whitening is a transformation of data in such a way that its covariance matrix Σ is the identity matrix. Hence whitening decorrelates features. We achieve zca whitening through library named Kornia. The RGB channel distribution profile shows how the whitening impacts the processed image.

In the image segmentation problem, the foreground was extracted from the image. Edge detection was carried out using cv.Canny() function from OpenCV. The extracted edge was then used to create a mask by identifying the contours (findContours()). The mask was then applied to the original image so that only the seedlings are visible. The segmentation is not perfect and is noisy. The reasons for this could be the following:

1. Edge detection requires further tuning to closely match the contours of the seedling profile.
2. The mask has to be tuned so that it blends at the edges instead of cutting off.
3. The vertical lines in the image look very similar to the seedlings. Therefore, contrast filters may be required to make sure the vertical lines are not identified during the edge detection process.