Assignment 6

Due Date: 2022/03/26

Assessment: 4% of total course mark

Instructions:

- For coding problems, please include the results as well as the screenshots of codes in the report
- Please upload source codes along with the report in Avenue (1 zip/rar file including codes, results and 1 PDF report file)
- The report MUST be written in Latex
- The codes MUST be written in Python language
- Please write comments for your codes!
- The explaination about the code MUST be included in the report!

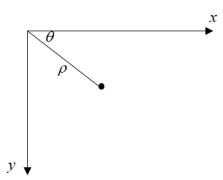
Theory (50 %)

1 Hough Transform, 15 %

Consider the following image.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 255 & 255 & 255 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 255 & 0 & 255 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 255 & 255 & 255 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 255 & 0 & 0 & 255 & 0 \\ 0 & 0 & 0 & 0 & 0 & 255 & 0 & 0 & 255 & 0 \\ 0 & 0 & 0 & 0 & 0 & 255 & 0 & 0 & 255 & 0 \\ 0 & 0 & 0 & 0 & 0 & 255 & 255 & 255 & 255 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{1}$$

If we apply the Hough transform on the image below, what would be the maximum value for the accumulator cell in the (ρ, θ) space? What is the corresponding (ρ, θ) value? The coordinate system is shown in below:



2 Optical Flow, 25 %

Consider a frame of a video is:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 255 & 255 & 255 & 0 & 0 \\ 0 & 255 & 0 & 255 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{2}$$

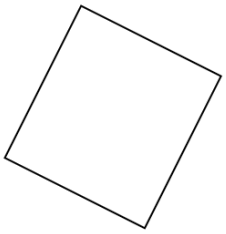
and the next frame is:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 255 & 255 & 255 & 0 \\ 0 & 0 & 255 & 0 & 255 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{3}$$

Derive I_x, I_y and I_t. Assuming the spatial coherency can be applied in evey 3×3 window. Write optical flow equation and derive u, v for each pixel (The solution of Ax=b for an over-constrained linear equation is $x = (A^T A)^{-1} A^T b$).

3 Orientation Detection, 10 %

Consider the following object:

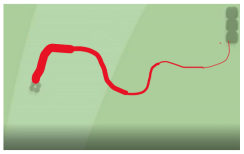


Propose a method to detect the orientation of the object.

Implementation (50 %)

1 Tracking (15 %)

Using optical flow estimation, track the snake in each frame of "video.mp4" and then show the whole path in one image. To encode the time information for the shown path, make thickness of the path proportional to the time of the video (the thickness of the path in the first second is 1 then 2, ...). The example for thickness is shown in below (the path in the example is different from your video)



2 MNIST Digit Recognition (35 %)

Download MNIST digit dataset (https://data.deepai.org/mnist.zip). Only Keep the images that has label 0, 1 or 2. Show 10 images randomly from MNIST dataset. Split MNIST to 3 parts for training, validation and test (first 70% for training, 20% for validation and 10% for testing).

2.1 Preprocessing

Apply a low pass filter to remove noise and then binarize images.

2.2 Feature Extraction

For each digit, using the contour features, extract number of pixels (X_1), width to height ratio (X_2), Area (X_3), Perimeter (X_4), Moments (X_5), width of the image (X_6). Store all features in a numpy array. You should end up with a $N \times 6$ array where N is the number of the data. For each digit (0, 1 and 2), compute the variance and mean of each feature for training data. Show the results in a table. From the table, which features do seem to be more important?

2.3 Dimension Reduction

Apply PCA on the training data and only keep 2 first components. Plot the training data in 2D space. Mark different classes with different colors.

2.4 Classification

Train an SVM classifier on the extracted features after dimension reduction. Plot the confusion matrix of the training data.

2.5 Optimizing number of components

Plot validation accuracy vs number of components in PCA and determine best value for number of components in PCA. Retrain SVM with the new number of components of PCA and report validation and training accuracy and confusion matrix.

2.6 Evaluation

Report the confusion matrix and accuracy on test dataset. Show 10 random images that are correctly classified and 10 images that are incorrectly classified.
