DataByte

The Official Machine Learning and Data Science Club of NITT

FIRST-YEAR INDUCTION

TASK-2

COMPUTER VISION

1. Automatic Segmentation of Traffic Signs or Signals using Image Thresholding

Description:

In this problem, prepare a proper dataset with annotations of images containing traffic signs or signals by web scraping techniques. Your goal is to develop an algorithm that can automatically segment and extract the traffic signs or signals from these images using image thresholding techniques.

Dataset:

The dataset must consist of a collection of images captured from different traffic scenarios. Each image can contain one or more traffic signs, and they can vary in size, shape, colour, and background clutter. The dataset must include corresponding ground truth annotations that specify the pixel-level segmentation masks for the traffic signs.

Reference : traffic signals

Objective:

Your task is to design and implement an image thresholding algorithm that can accurately identify and extract the traffic signs from the images by segmenting them from the background. The algorithm should be able to handle different lighting conditions, variations in traffic sign appearance, and background clutter. Provide the implementation of your algorithm along with the evaluation results, demonstrating the accuracy of the traffic sign segmentation. Additionally, you can showcase visualisations of the segmented traffic signs overlaid on the original images for validation.

Evaluation:

For evaluation, describe theoretically a proper method for determining the accuracy of your output, and implement it. Provide the details of accuracy in your metrics of evaluation.

Reference:

Image-based automatic traffic lights detection system for autonomous cars

Judging criteria:

- Performance and accuracy are important features but are not the prime factors of consideration as we want you to experiment here
- Understanding and innovation in the approach

\

2. Fashion MNIST

PROBLEM STATEMENT:

Develop a semi-supervised learning approach to classify the Fashion MNIST dataset using a limited amount of labelled data and a larger amount of unlabeled data. Measure and compare the classification accuracy achieved with different ratios of labelled to unlabeled data. Experiment with accuracies and implement the best. Implement the model using Pytorch. We encourage you to try something new while implementing even if the accuracy might go lower, try something new and note the observations. Experiment with loss functions and optimizers as well.

Implement the model completely using Numpy for brownie points

Details:

Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255. The training and test data sets have 785 columns. The first column consists of the class labels , and represents the article of clothing. The rest of the columns contain the pixel-values of the associated image.

- To locate a pixel on the image, suppose that we have decomposed x as x = i * 28 + j, where i and j are integers between 0 and 27. The pixel is located on row i and column j of a 28 x 28 matrix.
- For example, pixel31 indicates the pixel that is in the fourth column from the left, and the second row from the top, as in the ascii-diagram below.

Data: Fashion MNIST (Consider only the csv files)

EVALUATION METRICS:

This task is evaluated on the categorization accuracy of your predictions (the percentage of images you get correct)

Reference work:

Semi-Supervised Learning in Computer Vision - Amit Chaudhary

Judging criteria:

- Performance and accuracy are important features but are not the prime factors of consideration as we want you to experiment here
- Understanding and innovation in the approach

Don’t hesitate to do partial submissions, we appreciate your work at the beginners level.

Resources:

Python

Basics of ML

Computer vision

Sci-kit

Image thresholding

Image segmentation

Image classification

Custom semantic segmentation