Final project report

Course: Computer Vision 2020

Francesco Pham (1234004)

Automatic tree detection and recognition

This report covers my final project on object detection and recognition.

To achieve the objectives of this project, the approach I decided to follow is to use the cascade classifier originally proposed by Paul Viola and Michael J. Jone in 2004 for the face recognition. The classifier is already implemented in the OpenCV library under the name **CascadeClassifier.**

## Dataset generation

The dataset I used to train the classifier was manually created by me by downloading the pictures from google images. The number of images I managed to collect was 120 positive samples of tree images and 144 negative samples of non-tree images.

Figure 1 - Positive samples

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Figure 2 – Negative samples

In order to increase the diversity of the collected data I performed a data augmentation using the Keras library by applying some image processing operations such as rotation of ±20 degrees, shear, zoom, horizontal flip and brightness change. The final dataset consists of 260 positives and 260 negatives.

A picture containing tree, flower

Description automatically generated

Figure 3 - Augmented dataset

Finally, by using python code I generated two text files positives.txt and negatives.txt containing the list of paths to the positive and negative samples, these text files are needed for the training process.

## Training

First of all, I executed the following command to generate the positives.vec file containing all the positive samples with size 50x50.

Finally, to perform the training process I used the following command:

*opencv\_traincascade -data data -vec positives.vec -bg negatives.txt -numPos 240 -numNeg 240 -numStages 20 -w 50 -h 50 -featureType LBP -maxFalseAlarmRate 0.2*

*opencv\_createsamples -info positives.txt -vec positives.vec -w 50 -h 50 -num 264*

This command runs the training of the cascade classifier using the positive and negative samples provided. After some tests tuning the parameters, I ended up using 20 stages and max false alarm rate of 0.2. I also tested both LBP and Haar feature types and decided to use LBP because the features are much simpler, and the learning is much faster using local binary patterns. The training produced an output file cascade.xml which is our cascade classifier.