Cirlce classification for mushroom recognition using artificially generated 2d samples

Samuel Gelman

3-5-2022

Motivation

Mushroom farming is different from traditional plant based agriculture in many ways. Central of which is that a mushroom is the fruiting body of a fungal organism which lives its life underground. The organism is composed of a large mycellium network which metabolizes decaying organic matter and absorbs water from its growing medium.

Additonally, fungi do not grow from seeds, rather they are spored from spores produced in the mushrooms which are microscopic size.

This anatomy lends a large degree of unpredictability as to where a mushroom will sprout from within the growing medium.

The classification of mushrooms via image data will allow the automated system to record their location and create a unique profile to track and monitor the specific mushroom until it is ready to be harvested and shipped.

The first step is therefore the classification and identification of the mushroom. This will be done by processing image data gathered from the grow house.

Mushrooms also come in many shapes and sizes. some with one cap and some with many. Mushroom caps can be light, dark or brightly colored.

Mushrooms can also grow in very dense clusters.

This makes the task a generalized algorithm a difficult task.

For this preliminary work, simple types of mushrooms will be attempted for verification, namely, mushrooms in which there exist one, spherical body which is close to white in color.

This sharp contrast to the dark color of the growing medium will create strong gradients in the pixelated data which greatly reduces the difficulty of the classification task.

Before actual mushroom image data will be tested, algorithms will be tested on simple computer generated sample data. This data will be binary and two dimensional.

Method

1 Data Generation

As mentioned above, the mushrooms being used are simple white, semi-spherical bodies being grown on a dark surface. This data can be approximated by creating 2d image samples of white circles on a black surface.

A script was written to generate samples of this nature. Parameters for the size of the samples, the max/min circle radius and the max/min amount of circles to be generated can be entered in the script. The script also outputs text files giving the relative paths to the saved jpeg images as well as the coordinates for bounding rectangles surrounding each of the circles of interest.

A similar script was used to generated randomly placed and sized rectangles to provide training data and ensure that the model could distinguish between shapes. This will help from falsely classifying objects or anomalies in the soil that are not mushrooms.

Finally samples with a mixture of circles and squares were generated to test the algorithm.

2 Models and Algorithms

2.1 Haar Cascade

The training algorithms used were sourced from the open source library for computer vision software, OpenCV. This is a machine learning algorithm which is given positive and negative image data and classifiers are trained through this supervised learning process. Positive images are images where the object we wish to train the algorithm to detect is present while negative images are any other image.

The Haar cascade algorithm works by using an number of filters to scan an input image and generate corresponding feature maps.

At first, each individual feature extracted is known as a "weak feature". After many samples are run through the classifier cascade, these "weak classifiers" are combined to create strong classifiers which lend a higher confidence rating towards classifying the object of interest.

2.2 Circle Hough Transform

The circle Hough Transform is a feature extraction technique used for image processing tasks.

To explain the theory we start with a the mathematical description of the circle in two-dimensional space using Cartesian coordinates:

$$(x-a)^2 + (y-b)^2 = r^2.$$

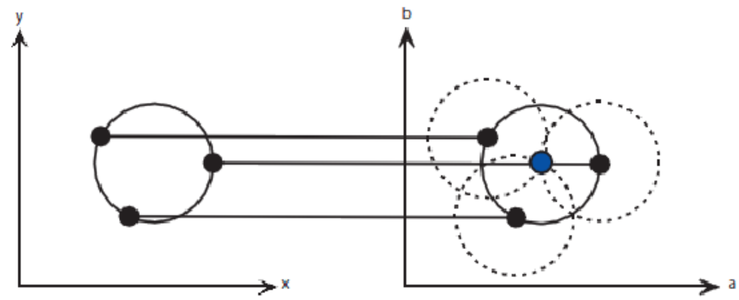

Figure 1: The circular Hough transform maps points from (x,y) space over to the (a,b) parameter space. For each point in parameter space the corresponding circles are drawn and recorded in the accumulator marix as seen by the dotted lines. The maximum marks the center of the circle marked by the blue point.

Where a and b represent the x and y coordinates to the center of the circle respectively and r is the circles radius.

When the radius is fixed, this equation defines the outline of a circle. In the event that our image data has circles which are filled in, the edges can be extracted using the Canny edge detector algorithm. This is a preliminary data preparation step built into the openCV implementation of the algorithm.

The Hough transform works by scanning over all the x-y coordinates of the edges in the input image and tracing out a circle in the parameter space. When only a single radius is being scanned for then the parameter space is two dimensional, namely a and b of which we are mapping the output to is the given values for the center of the circle given the edge points represented by x and y. This can be seen in Figure 1. The output of each of these scans is recorded into an accumulator matrix which is the same size as the sample image.

The maxima points in the accumulator matrix after all the edges have been scanned will represent the center of the circles present in the input image.

If the radius of the circle is unknown the parameter space inherits another dimension or r.

This algorithm has several downsides and room for optimizations.

For one, the parameter space that we are accumulating to is a discrete representation of space. So choosing the resolution of the accumulator is a tunable hyper-parameter.

In addition, in real life examples, circles are not perfect, in this way several circles of the same size can have differing maxima in the accumulator matrix after the algorithm has finished its scan. For this reason a threshold parameter is necessary to determine which candidate circles are accepted.

Deep neural networks have been shown to outperform the Hough Transform and are less susceptible to many of the potential false classifications inherent in the method.

Despite all of these potential drawback the algorithm is both powerful, simple and widely applied.

Results

3 Analytical Review

3.1 Haar Cascade

The first algorithm tested was the Haar Cascade classifier. While this model worked well at identifying shapes, it failed to distinguish between the circles and rectangles. This led to an extremely high "false alarm" rate. After further research, the inability for Haar cascade classifiers to distinguish between the soft edges of circles and the edges of squares was a known problem. The Haar Cascade was implemented successfully from a data preparation stand point but the inability to detect the soft edges made the model unable to distinguish between the squares and circles.

So while the algorithm managed to detect the correct number of objects it proved to be a poor choice for the purposes of this project.

3.2 Circle Hough Transform

While needing some tinkering with respect to the parameters, the Hough Transform proved to be very effective while remaining relatively easy to implement.

Some optimization was needed, but a final accuracy of ˜99.583 percent was reach. This value was determined by generating 1000 samples of size 500x500. Anywhere from one to twelve shapes were generated onto the images with a 75% chance of it being a circle and a 25% chance of it being a rectangle. The maximum radius or length/width of the shapes were 40 pixels while the minimum was 15 pixels.

The algorithm counted the number of circles found and compared this to the number of circles present. For each sample the absolute value of the difference was collected and summed. At the end, the amount of missed circles and false negatives were compared to the total number of circles accurately identified.

This metric does have a pitfall in that samples where there is both a false positive and a missed circle get recorded as having no errors. This phenomena was observed when scanning through the data but it was rare. For a production grade algorithm a more rigorous score would be formulated, but to present a rough sketch of the capabilities of this algorithm this metric felt sufficient.

Another important parameter which was tuned is setting the max radius for circles which the Hough transform can identify. This was set to one fourth of the image size which was 125 pixels. This is over triple the max radius of the mushroom. By lowering this down to 80 pixels, twice the size, the accuracy was increased to ˜99.815.

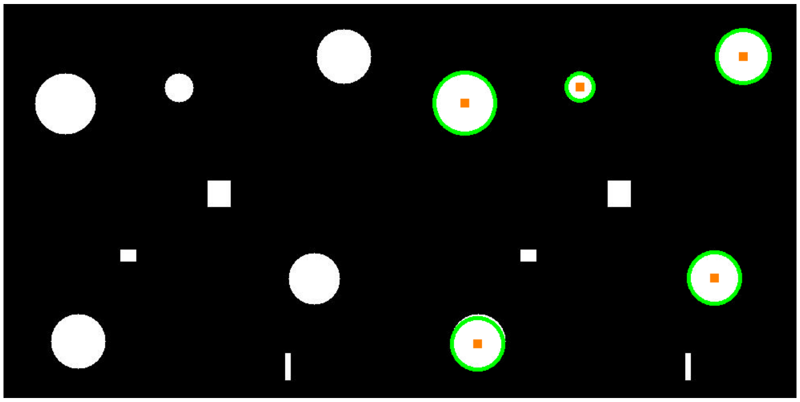

Figure 2: Image sample before and after Hough transform. Information of the circles is stored as a three dimensional coordinate of (x_center, y_center, radius)

All other parameters of the algorithm were taken via online recommendations and trial and error.

Conclusion

With only reasonable effort and through utilizing open source technologies, a frame work for data generation and circle classification was completed with a fairly high confidence rating. While better algorithms exist, the circle Hough transform worked fast and well. While real world image data is always more complicated to work with, it seems more than plausible that the working implementation of the Hough transform here could preform the task of finding mushrooms growing on a 2d grow bed.

Thank you for taking the time to read this report.

All code written is available on my github upon request.

*SammyGelman@gmail.com, github: sammygelman