Project log

14/07/2018

I needed to find a format and structure to base my EPQ, so I visited IEEE, which is a common site for publishing papers. I looked for a paper similar to my own topic, and found “Plant Disease Detection Using Image Processing” by [Sachin D. Khirade](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22First%20Name%22:%22Sachin%20D.%22&searchWithin=%22Last%20Name%22:%22Khirade%22&newsearch=true&sortType=newest); [A.B. Patil](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22First%20Name%22:%22A.B.%22&searchWithin=%22Last%20Name%22:%22Patil%22&newsearch=true&sortType=newest). This paper was structured in the following format:

1. Abstract
2. Introduction
3. Literature review
4. Proposed method
5. Experimental specification
6. Experimental results

Due to its simple to understand structure, I decided to adapt this structure into my own EPQ, as I think it is suitable to my own investigation.

16/07/2018

I began my work experience today at QBots. I had an introduction to the development team, specifically in the computing section. They gave me an introduction on how machine learning can be developed for different uses, such as predicting movement inside smart buildings. This could also be applied in crop disease by looking at the different stages of a disease within a plant. The prediction of how the disease moves up or down a plant can help categorise that specific disease. I think I could use this later on in the project to help.

21/07/2018

I began writing the beginning of my EPQ today. As to what is meant to belong in an abstract (which is what I started with) I looked for samples. Again, I used the paper before, “Plant Disease Detection Using Image Processing” by [Sachin D. Khirade](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22First%20Name%22:%22Sachin%20D.%22&searchWithin=%22Last%20Name%22:%22Khirade%22&newsearch=true&sortType=newest); [A.B. Patil](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22First%20Name%22:%22A.B.%22&searchWithin=%22Last%20Name%22:%22Patil%22&newsearch=true&sortType=newest), to see what was written there. Seeing how the authors wrote the abstract by describing first what the importance behind the project was, I decided to also look for a few other sources. I needed to consider which application/language I would use to write the code. I decided on Python mainly because I just had the most experience on it, as well as because of the fact that there is an open source distribution named “Anaconda” which makes adding modules incredibly easy.

“Crop disease detection using image segmentation” by Tushar H Jaware, Ravindra D Badgujar and Prashant G Patil also started similarly, however, due to these authors using a specific method to approach their subject, they started with a description of their method. As I am not planning to do the same, since I decided to compare various different machine learning methods. I decided to therefore do something in between both. I wrote the abstract thinking of the impact of both crop disease, and how the application of machine learning could be used to help lower the impact of this disease.

22/07/2018

I needed to prepare which methods to investigate. As I had previously written code for the methods K nearest neighbours(KNN) and random forests(RF), I felt it was appropriate to include these methods. From random forests, I could also use the decision tree classifier(CART) as well, as random forests are in the end, a lot of CART trees, with a few rule changes. In regards to the other machine learning methods, I felt it was appropriate to include the most popular machine learning methods. These were:

1. Linear regression
2. Logistic regression
3. Linear Discriminant Analysis(LDA)
4. Gaussian Naïve Bayes(NB)
5. Support vector machines(SVM)

I also had a couple lessons with tensorflow, and thought it would also be good to add tensorflow to this experiment, so in the end, my final methods were Linear regression, Logistic regression, LDA, NB, SVM, KNN and RF. I have a good grasp of most of these methods so I think these choices were made well.

23/07/2018

After doing some more extensive research, I think it would be better to merge Linear and Logistic Regression together. This is because linear regression isn’t very suitable for my research. This is because the dependent variable for Linear regression is relying on the data being binary. Due to my project requiring multiclass methods with many different plants, I think this method of classifying would be very poor. To continue this project, I also decided to use the publicly available dataset from the following URL: https://gitlab.com/huix/leaf-disease-plant-village/tree/master . I want to ideally use around 5 plants with at least 2 diseases in each. I think I will also include the healthy pictures of these plants in the dataset would be a good idea. As for the volume of the dataset I want to import, I don’t have a particularly high end computer, so I think, in order to compensate between having a large dataset and making sure my computer can still run the programs I intend to write, I may want to have to download at least 200 images, capping at around 1500. I ended up with apple, corn, tomato, potato, and grape, as my final 5 crops to use. Next up would be to start the actual main body of the writing itself.

25/07/2018

Starting on the introduction here, I think that writing something that fits the title would make sense. For example, I started the introduction here with an intro to the current situation of the world and why crops are valuable to everyone. Moving on from this, I talked about the implications on how the efficiency of the crop yield would need to be maximised to fit the exponential growing population. I think that this grabs the attention of the reader, so I moved on with talking about the consequences that would arise if these factors were to go awry. At this point it would make sense to then talk about what this disease is, which then brings the attention to the cause, and would intrigue with different questions like “If this happens then we can solve it like this”. From then on I talked about the current situation on addressing this method, and the advantages and disadvantages, to analyse how this may be useful in which areas.

05/08/2018

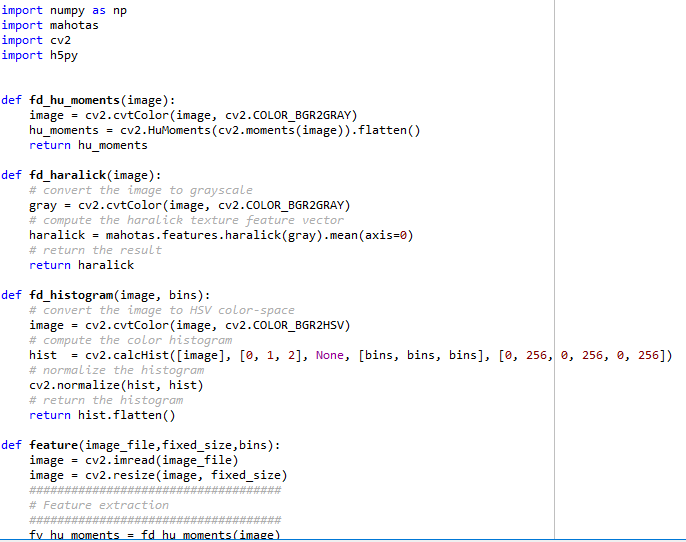
I began my literature review today. My main goal within this section is to be able to list and analyse several different methods of the machine learning methods used currently by various people around the world, and what for. I aimed to get as much variety into this section as possible, finding different image processing techniques to classify various different diseases on anything from agriculture crops to groundnuts.

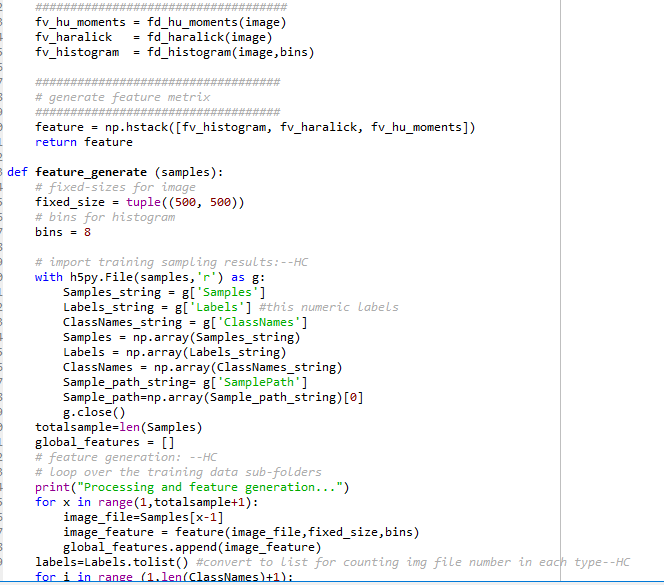
I then analysed what these methods do in summary, making it seem like it’s the surefire way to go forward with in the future. I plan to then go in depth on the advantages and disadvantages of using this machine learning in the process of detecting and classifying diseases, as the method itself is not a perfect solution.

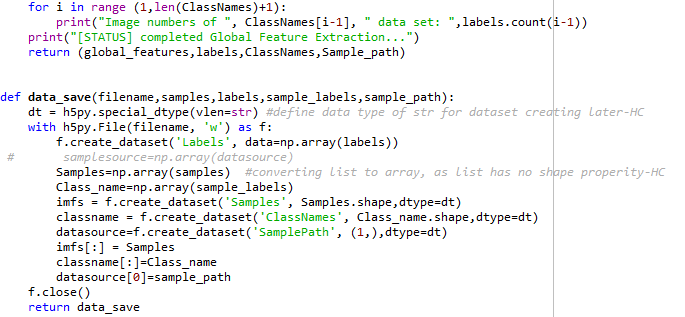
10/08/2018

I had planned to get started on the coding itself today. I assume multiple programs would have to be constructed, in order for the entire system to flow correctly. This is because I have to account for the import of features and classes, which I would also have to define. This would in general be bad practice and look very bad, so I believe separating these processes would be the ideal way to go.

The first program will be to define the actual features of the images that I plan on classifying and detecting. This program should be vital to the rest of the operations and would have to be used almost everywhere as it is the backbone of this system.







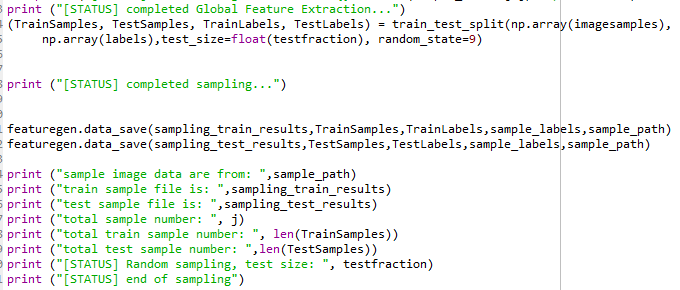
Each “def” defines a feature or process that I will be using later on, this program is more like a bank of some sort, storing the functions.

My second program will be a program that separates my entire dataset into a training and testing dataset. To do that I will need to import the following modules.



Featuregen however is not a built in package, and it is my program that I made first, which will be seeing action now.



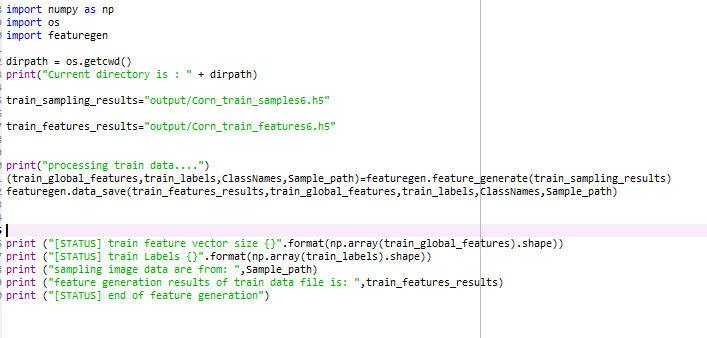


In this program I first set a variable to the location on my computer where I downloaded the dataset. Within this program, I also set an interactive element to the code, where the “test size” of the dataset/ the size of the data that will be tested against the classifier, is customizable, between 0 and 1(0 being 0% and 1 being 100%, although these values are never used). I then created empty lists of which are meant to hold feature vectors, labels and image file names. Using this, I generate the features by getting the training label, and classifying each image in that training label with the name of the label as the class. The infrastructure of the folder is as goes: Main dataset, which splits into several folders containing the names of the different plants, splitting into more folders each, containing diseases for their respective plant. This code reads all the images within that secondary subfolder and assigns those images to that plant folder. This completes this program, with some status messages at the end which are useful for debugging in case any section of that code is wrong.

13/08/2018

The next part of the code was completed today. They were named “image\_feature\_generation” and “method\_compare”

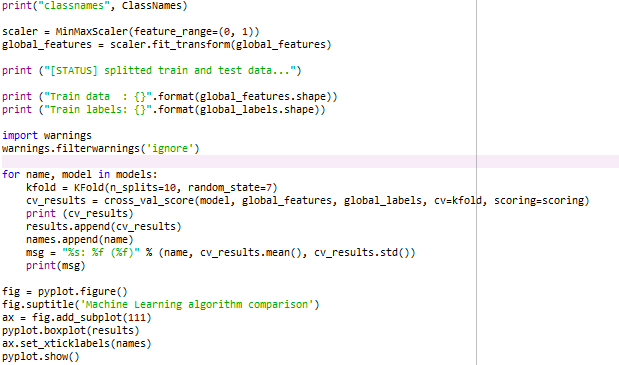
Image\_feature\_generation:



The function of this program is to be able to read and retrieve the feature vector size and label size of the training dataset samples, and be able to generate these features, which will be added to the arrays defined previously.

Method\_compare:





This method serves as a main hub for the machine learning models. It is designed to import the seven methods that I plan to investigate, and provide these models with the same scenario and data. This is done by importing the feature vectors and trained labels defined previously. However, the imported data would be very chaotic if I was to experiment on them as they were imported. Therefore the normalisation technique is used, where the outliers and variance of the data samples is reduced and each data sample going in, is of the same quality as the others.

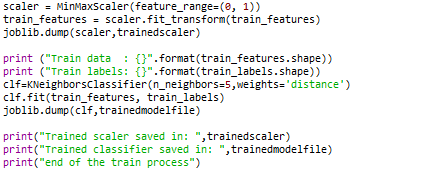
Normally an error would be made here because the imputation would raise a flag with python, due to the inputted modules not being registered. To bypass this fault, the module ‘warnings’ is inputted which ignores the message raised, and moves on to the next stage. The next stage itself uses a method called k-fold cross validation. This essentially “folds” the data, a k number of times, and by using the sections created by each fold, an accuracy is determined by running each classifier against this folded training dataset. I’m not very good with the output normally given in words however, so I decided to add a boxplot which represents the same values.

23/08/2018

The final programs were finished and refined today. They were “Train\_X” and “Train\_Y”.

Train\_X:





One of the more simplistic programs to make in this project, the goal of this program itself is to train the classifier onto the training dataset, which was all prepared with the previous programs. The name “X” in the title of the code is meant to be replaced by the initials of the classifier model used. For example “RF”= Random Forests. The coding itself is very similar to method\_compare, of course without the cross validation and k-fold technique. The “clf.fit” syntax is very handy, as it comes prebuilt into the sci-kit learn module in python, and does most of the classification work itself. It builds the new classifier model itself.

Test\_X:





Like with Train\_X, the program first imports the features extracted from the dataset, however this time the dataset is from the testing dataset. Next, the models for the machine learning algorithms are also imported because they are needed to test against the testing data. By using the built in syntax of clf.predict, a prediction is made by the algorithm, which is then tested for by the computer itself whether the prediction is true or false. This is where the accuracy score equation comes in:

Accuracy= (true positive + true negative)/(true positive + true negative + false positive + false negative)

Each model does this for each data sample, and an accuracy is generated for each model. These are then printed out at the bottom. I can then use these different values together at face value to compare accuracies.

With this done now, I won’t need to worry about coding until I need the results to summarise.

25/08/18

Following on with the titles of each section, “proposed method” is where I listed out what I plan to do. I started this off by describing machine learning in detail, and following on with a detailed description of each machine learning method. I did this because I think it is necessary for the reader to understand why I need to use this many different methods, and knowing how they all differ from each other can be very beneficial.

I then drew a flowchart of what I plan to do with the dataset given. It’s a very basic generic image, because most of the time these would be the steps that are followed in the process of machine learning. I expand on this simple diagram later on however, listing each different program and how they’ll be run. Later on I expect to refurbish my PC to be able to run the entire program.

10/09/2018

I had to prepare my computer to run the entire program later in the week today. I had to backup all my files and start a fresh account because my CPU was not good enough to handle too many tasks going on at once. After giving it a test run I estimate that the entire thing will take me a couple hours to completely run.

17/09/2018

I finished running everything in order today. I plan to analyse the results by next weekend due to the large volume of data that was outputted. In total I had around 13 thousand images to sort through for 7 different classifiers. In essence, 91 thousand pieces of data to filter through.

24/09/2018

I completed the next part of the section, which was the experimental specification (I renamed to set up because it sounds clearer). To get this info I just simply used msinfo32 on my cmd+r windows run executive. I also added the order and description of each program that will be used, as it may be useful for people to understand what each program actually does and why they are run in a specific order. To complement this, I also added the infrastructure of the folder layouts itself, for visualisation benefits.

04/10/2018

I added in the results in a table format, listing each accuracy and standard deviation for each classifier. It made things easy to comprehend and was very compact. To further help visualisation, I decided to add the boxplots for each classifier.

06/10/2018

I decided to finish my conclusion today. It was mainly just making the table of values seem easier to read, and summarising the results.

10/10/2018

Looking back at the results, I decided that including all the values into one table was not clear enough. I ran the entire code again, but this time I decided to add each different plant separately, with each different disease under that plant as the column headings, and the classifier models as the row headings.

16/10/2018

I didn’t like the disease as the headings, it looks too compact. Instead, I changed the headings to suit the accuracy, standard deviation, minimum accuracy, and maximum accuracy. I removed the boxplots to compensate for this.

30/10/2018

I trimmed many of the paragraphs down to seem not as saturated and repetitive as it was before. I added in the bibliography at the end, with circular brackets targeting each reference made. The titles of each sections were also adjusted to fit the content that was also changed in the trimming. The new titles are now the following:

1. Introduction
2. Related work
3. The application of the machine learning methods to crop disease detection
   1. Machine learning methods
   2. Proposed framework for crop disease detection
4. Implementation and experiment evaluation
   1. Implementation
   2. Experimental evaluation
5. Conclusion
6. References

12/11/2018

After reading some more papers, specifically, “Computational Sustainability and Artificial Intelligence in the Developing World” by John Quinn, Vanessa Frias-Martinez, and Lakshminarayan Subramanian, and “Fast and Accurate Detection and Classification of Plant Diseases” by H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik and Z. ALRahamneh, I realised that my conclusion does not fit what is seemingly expected of a conclusion in this sort of writing. I plan on rewriting the conclusion soon, after my UCAS problems are dealt with.

24/11/2018

I finished the first draft of the EPQ today. I tidied everything up, added in the references in square brackets with the reference at the footer of each page. I completely tidied up and rewrote the conclusion, which now accurately represents the entire investigation that I carried out. There was obstacles when Word had confused me with how to change the footer at the bottom to be entirely different for each page. By using page layout and “new page” as well as unlinking each footer to the previous, solved this. At the end I also updated the bibliography. I will rewrite any parts needed according to feedback in the future.