C7082 Assignment

17239400 - Harry Buckley

17/01/2022

https://github.com/BUCKERS99/C7082---Assignment/

Background

Potato blight is a disease that has a historically noticeable presence and has caused food shortages in a rang of countries; the most harrowing example is the Irish potato famine around 1845 (O'Neill, 2009). The current methods of controlling the disease, caused by the fungus Phytophthora infestans, (late blight) and Alternaria Solani, (early blight), is to employ a spray program that requires field wide chemical treatment of crops on a regular basis: up to twice a week, depending on the chemical being used(Leesutthiphonchai et al., 2018). With the world population increasing to around 9.1 billion by 2050 (Cohen, 2001) the need for secure food production in the future is important if we are to supply the increase in people. With genetic modification (GM) currently not an option in the UK, a line of potatoes with full resistance to blight would not be available for at least 30 years, due to the time it takes to breed in a gene (Ceccarelli, 2015; Haverkort et al., 2009).

There have been observations of reduced efficacy in the chemical control of blight when using active ingredients such as fluazinam (Schepers et al., 2018). This can be attributed to the overuse of a chemical strategy and a build-up of resistance in the fungus. The active ingredients used in the control of blight has deadly and long-lasting effects on the environment and is highly toxic to humans (EFSA, 2008). It has been seen to cause severe damage to aquatic habitats and this can easily occur in the UK where chemical application is completed in adverse conditions and creates run-off.

For the reasons of environmental sustainability and with the cost of applications adversely effecting the producers profit margins; the implementation of blight recognition using a machine learning technique could provide a future proof, precision application method of blight control. For example, the adaption of a drone to carry out fungicide application with a camera attached to recognise plants that have a blight infection could reduce the amount of fungicide used, increasing profit margins, and prevent the field wide application of the pesticide which will reduce the chance of run-off.

As the data set used in this assignment is made up of images the decision was made to use Convolutional Neural Networks (CNN) to analyse and classify the images. CNN's have been proven in their use for classification tasks and provide a high accuracy, when tuned correctly, on validation data sets (Yoo, 2015). The objectives of this assignment are:

1. Use different pre-trained models to identify an image of a healthy leaf, a late blight infected leaf and an early blight infected leaf.
2. Tune the model that produced the lowest validation accuracy, in the first run, to see if it is possible to outperform the initial validation accuracy of the best performing model (first run).

Methods

Data

The dataset was taken from Kaggle and contains images of healthy potato leaves (Kaggle, potato leaves that are infected with early blight (A. solani) and potato leaves that are infected with late blight (P. infestans). Within this data set there were 2152 files split as follows:

- Early blight - 1000 images - JPG files
- Late blight - 1000 images - JPG files
- Healthy - 152 images - JPG files

It was decided that there being 15% of amount of healthy potato leaf images compared to the diseased images had the potential to cause problems when training the model; this having been considered, the decision was made to collect more healthy potato leaf images. This was done using Kaggle where a further 816 images were taken from the training folder (Kaggle, 2021b).

- Healthy - 968 images - JPG files

The image size for most of the images was 256x256 pixels, to make sure that the images were inputted into the model with a consistent size it was decided that they should be rescaled when necessary.

File re-naming

It was found that the names of all the files were a random mix of letter and number. The decision was made to rename the files and order them numerically depending on the category they were from. This was more for the users benefit as it does affect the machine learning model. "00fc2ee5-729f-4757-8aeb-65c3355874f2____RS_HL 1864.JPG" was renamed to "Healthy_1.JPG" and this was done consecutively with the number altering in order. This was repeated for late blight leaves, early blight leaves and healthy leaves.

##File sorting The total images were split into three categories: test, train, and validation images and an 80/10/10 for train/test/validate was used. Table 1 shows the number of files in each folder for each category.

Table 1: Number of images in each folder

	Test	Train	Validate
Healthy	97	97	772
Late Blight	100	100	800
Early Blight	100	100	800

#Models ##VGG16 VGG16 has been used in many image classification problems; notably in a study conducted by Rangarajan and Purushothaman (2020) it was used to detect disease in eggplant images. VGG16 is designed for image classification with 13 stacked convolution layers. Each layer extracts features depending on the disease image it is given. A softmax function is used on the output in order for a probability score to be made to each class. An example of the architecture of VGG16 can be seen in figure 1.

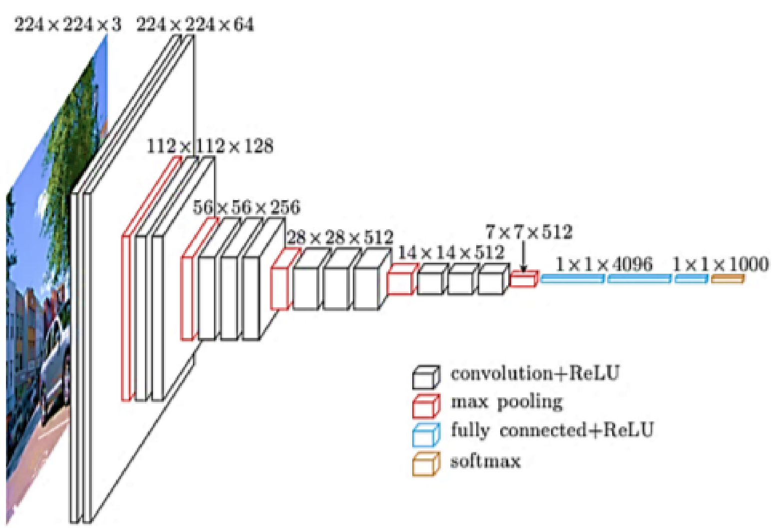

Figure 1: VGG16 architecture

The images were rescaled to 255x255 and were inputted into the model as Red, Green Blue (rgb) scale, as was the same for all the initial run of the models. The layers were frozen to prevent each layer being updated during the training of the model. To save on computing power and time, 10 epochs were chosen with 50 steps per epoch. The code can be found here for the model testing. A training accuracy of 98.4% and a validation accuracy of 80.1% was recorded after the initial running of the model. ## Xception

In comparison, both Xception and VGG16 are trained on the ImageNet (2021); however, Xception is a CNN that is said by its creator Chollet (2017) to outperform both VGG16 and Inception v3 due to its depth wise separable convolutions. For the image dataset used in this instance the learning layers were again frozen and 10 epochs with 50 steps per epoch were chosen in order to have a direct comparison to the other models. A training accuracy of 98.4% and a validation accuracy of 79.8% was recorded after the running of the model which was lower than the VGG16 model.

Inception V3

Inception v3 is a pre-trained model that was also trained on the ImageNet data set which allows for a good comparison between the three models (Szegedy et al., 2015). It is based on the idea that each layer is an inception layer where each layer's output is filtered into the input of the next layer, shown in figure 2.

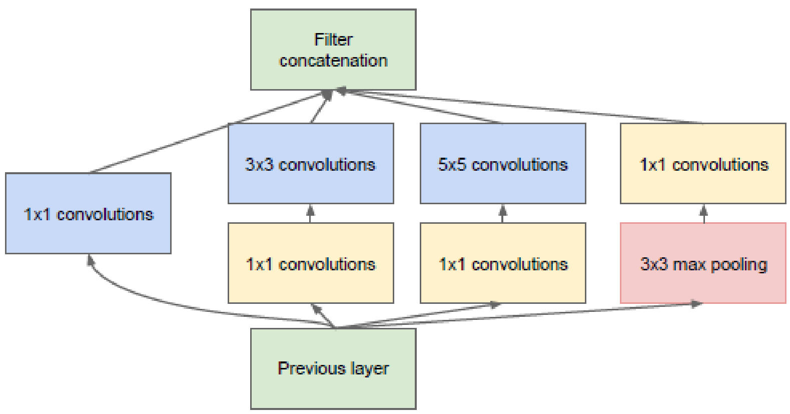

Figure 2: Inception v3 architecture

This has a negative effect of increasing the computational power needed to run the model, but should, in theory increase the accuracy.

A training accuracy of 76.9% and a validation accuracy of 68.6% was achieved using the inception pre-trained model with the same number of epochs and steps per epoch as the previous models. As the inception v3 model achieved the lowest overall accuracy it was taken forward and tuned to see if it could meet, or exceed, the benchmark of 80.1% posed by the VGG16 model. The comparison of the accuracy can be seen in figures 3 and 4.

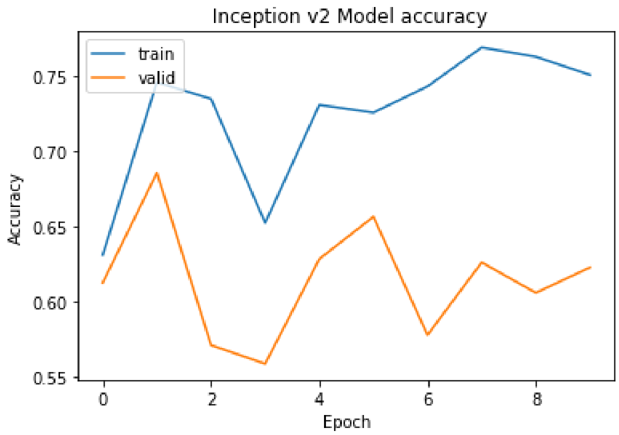

Figure 3: Inception V3 model accuracy

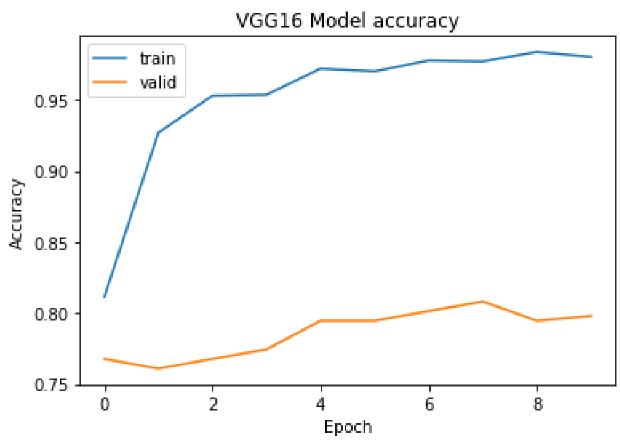

Figure 4: VGG16 model accuracy

Tuning

1

Increasing the learning rate from 0.0001 to 0.0005 produced a training accuracy of 91.4% and a validation accuracy of 84.9%. This is already 5% more accurate on the validation set than the initial Inception v3 run.

Tuning the learning rate from 0.0001 to 0.0005 has beaten the benchmark validation accuracy set by VGG16 by 4.8%. as this has completed the objective a new objective was set:

1. How close to 100% validation accuracy can we get the model to be?

2

To reduce the computing power the next model contained the early stopping function. This prevented the model running when there was little, or no improvement seen in the validation accuracy. Also contained in the second model was the activation change from relu to softmax. Having run this the training accuracy was 51.3% and the validation accuracy was 54.9% however, the early stopping call back stopped the model after 2 epochs.

3

For the third change of parameter the batch size was changed from 32 to 64, for this to work the number of steps per epoch has to be decreased, it was decreased to 25 steps per epoch. The results of this test produced a training accuracy of 82.4% and a validation accuracy of 78.9%.

4

Increasing the learning rate further to 0.001 and returning the batch size to 32 and the steps per epoch to 50 saw a training accuracy of 83.6% and a validation accuracy of 81.4%. the closing of the gap between test and validation accuracy shows that the model is over fitting less, which is a positive discovery.

5

For the fifth version of the model the number of steps per epoch were increased from 50 to 75 and the training accuracy was recorded at 90.9% and the validation accuracy at 85.1%.

6

The sixth model required the image size to be increased to 200,200 and run with the learning rate of 0.0005 and 75 steps per epoch. This gave a training accuracy of 96.7% and a validation accuracy of 85.9%.

Results

The model that achieved the highest validation accuracy was the final model, the seventh tune. The accuracy on the training data set was 96.7% and the accuracy on the validation data set was 85.9%. There is over 10% difference between these accuracies which suggest some element of overfitting. Figure 5 and 6 show the accuracy and the loss associated with the model ran.

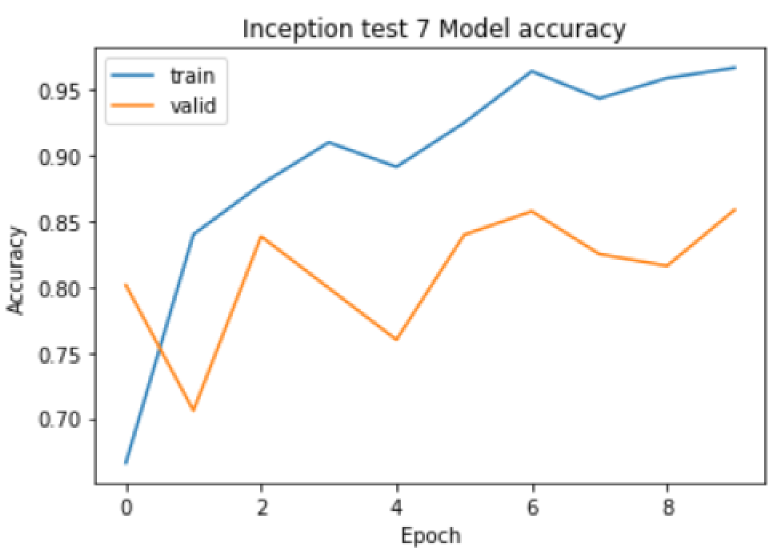

Figure 5: Tune 7 accuracies

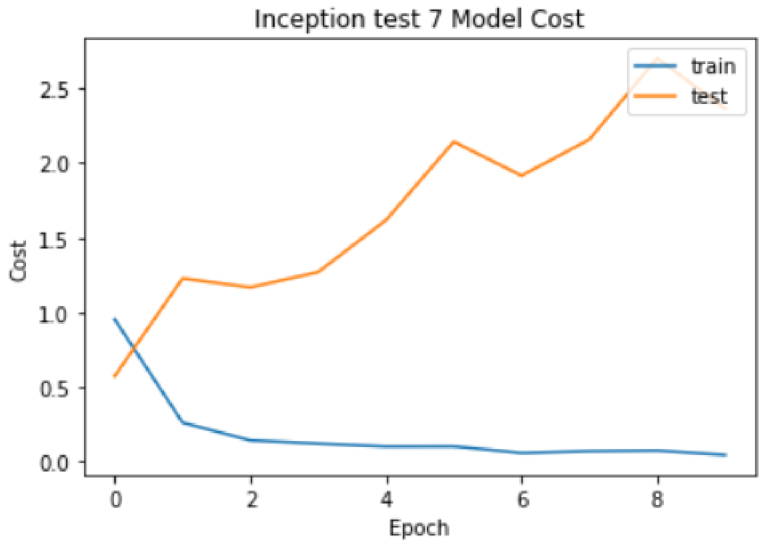

Figure 6: Tune 7 cost

The model cost reduced throughout when training the model, however it increased during the testing from 0.58 to 2.7.

The table below outlines the validation and training accuracies throughout tuning the model. It shows that the first change to the original model produced results that outperformed the benchmark. This required a new objective to be set, the closest the model got to a 100% accuracy using the Inception V3 pre-trained model was 85.9%.

Table 2: Table of results from tuning the Inception model

	Training Accuracy (%)	**Validation Accuracy (%)**
BENCHMARK	98.4	80.1
Base model	76.9	68.6
Tune 1	91.4	84.9
Tune 2	51.3	54.9
Tune 3	82.4	78.9
Tune 4	83.6	81.4
Tune 5	90.9	85.1
Tune 6	87.5	83.8
Tune 7	96.7	85.9

Discussion

The accuracies and results gained from running the models show that there is room for improvement. The challenge of using the worst performing model allows for the next step of using the best original model and improving it to as close to 100% accuracy as possible. The idea that models can be adapted to give a higher accuracy shows that the solution to applying a machine learning technique to real world problems is not a one-size fits all solution and parameters need to be changed and tweaked to achieve the best results.

In regard to the topic of this assignment, P. infestans and A. solani, this is a real-world problem given that the manipulation of traits is not an option in the UK and conventional breeding will take years to get the same standard of immunity. As illustrated in the background section of this report, food security is an outstanding issue as the potato is one of the main food sources for developed and developing countries

(Bailey et al., 2015). As shown by history, blight as a disease has the potential to cause large scale disruption to the supply of potatoes.

With this in mind, our results are not able to be used in a commercial setting (Too et al., 2019). The accuracies are too low for it to be taken forward and used in a field scale trial. With automation comes some ethical concerns around the allowance of an unmanned vehicle spraying potentially dangerous chemicals. For environmental reasons the accuracy for it to be allowed to operate would have to increase to around the 99% mark. There is a large potential for deep learning to be used in conjunction with precision agriculture, as it could have the ability to reduce pesticide use, with the identification of specific areas that require spraying (Melland et al., 2016). It also has the potential to increase the environmental sustainability of the production of potatoes; the correct identification of a disease and then the subsequent treatment being applied to a specific area can prevent over-dosing and reduce run-off of chemical into water systems.

In conclusion the implementation of deep learning into the agricultural sector has the potential for huge increases in environmental and economic sustainability. With the world population increasing food security it paramount to both developed and developing countries and methods of identification using computer vision could revolutionise the industry. It is necessary however for a higher accuracy than the one recorded during this report to minimise misclassification.

References

Bailey, R.L., West Jr, K.P. and Black, R.E., 2015. The epidemiology of global micronutrient deficiencies. Annals of Nutrition and Metabolism, 66(Suppl. 2), pp.22-33.

Ceccarelli, S., 2015. Efficiency of plant breeding. Crop Science, 55(1), pp.87-97.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

Cohen, J.E., 2001, June. World population in 2050: assessing the projections. In Conference Series-Federal Reserve Bank of Boston (Vol. 46, pp. 83-113). Federal Reserve Bank of Boston; 1998.

Denton, E., Hanna, A., Amironesei, R., Smart, A. and Nicole, H., 2021. On the genealogy of machine learning datasets: A critical history of ImageNet. Big Data & Society, 8(2), p.20539517211035955.

European Food Safety Authority (EFSA), 2008. Conclusion regarding the peer review of the pesticide risk assessment of the active substance fluazinam. EFSA Journal, 6(7), p.137r.

Haverkort, A.J., Struik, P.C., Visser, R.G.F. and Jacobsen, E.J.P.R., 2009. Applied biotechnology to combat late blight in potato caused by Phytophthora infestans. Potato Research, 52(3), pp.249-264.

Kaggle. 2021a. Potato leaf disease detection. [Online]. Available from: https://www.kaggle.com/sayannath235/potato-leaf-disease-detection/metadata [Accessed on 01/01/2022].

Kaggle. 2021b. potato disease leaf dataset (PLD). [Online]. Available from: https://www.kaggle.com/rizwan123456789/potato-disease-leaf-datasetpld/metadata [Accessed on 01/01/2022].

Leesutthiphonchai, W., Vu, A.L., Ah-Fong, A.M. and Judelson, H.S., 2018. How does Phytophthora infestans evade control efforts? Modern insight into the late blight disease. Phytopathology, 108(8), pp.916-924.

Melland, A.R., Silburn, D.M., McHugh, A.D., Fillols, E., Rojas-Ponce, S., Baillie, C. and Lewis, S., 2016. Spot spraying reduces herbicide concentrations in runoff. Journal of agricultural and food chemistry, 64(20), pp.4009-4020.

O'Neill, J.R., 2009. Irish potato famine. ABDO.

Rangarajan, A.K. and Purushothaman, R., 2020. Disease classification in eggplant using pre-trained VGG16 and MSVM. Scientific reports, 10(1), pp.1-11.

Schepers, H.T.A.M., Kessel, G.J.T., Lucca, F., Förch, M.G., Van Den Bosch, G.B.M., Topper, C.G. and Evenhuis, A., 2018. Reduced efficacy of fluazinam against Phytophthora infestans in the Netherlands. European journal of plant pathology, 151(4), pp.947-960.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2015. Rethinking the inception architecture for computer vision. 2015. arXiv preprint arXiv:1512.00567.

Yoo, H.J., 2015. Deep convolution neural networks in computer vision: a review. IEIE Transactions on Smart Processing and Computing, 4(1), pp.35-43.