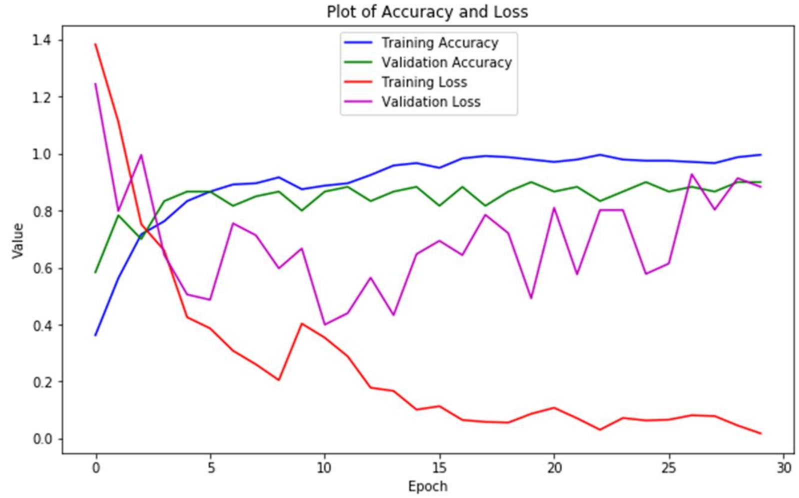
SA52 Team 2 Machine Learning CA – Part2

Team 2 Members: Fun Weng Lup, Ronnie. Aye Phyu Sin. Changying Shao. Willard Toh Hui Kian. Chen Wenquan. Chia Yan Long, Brandon. Hou Lu Chiok Weh Alejandro. Zhao Qi.

Models Specifications											
	Model 1	**Model 2**	**Model 3**	**Model 4**	**Model 5**	**Model 6**	**Model 7**	**Model 8**	**Model 9**	**Model 10**	**Model 11**
Conv2D Layers	Layer 1: 32 Filters, 3x3 Layer 2: 32 Filters, 3x3	Layer 1: 32 Filters, 3x3 Layer 2: 64 Filters, 3x3	Layer 1: 32 Filters, 3x3 Layer 2: 32 Filters, 3x3	Layer 1: 32 Filters, 3x3 Layer 2: 64 Filters, 3x3	5 Layers, each with 32 Filters, 3x3	Layers 1 and 2: 32 Filters each, 3x3 Layers 3 and 4: 64 Filters each, 3x3	Layers 1 and 2: 32 Filters each, 3x3 Layers 3 and 4: 64 Filters each, 3x3	Layers 1 and 2: 32 Filters each, 3x3 Layers 3 and 4: 64 Filters each, 3x3	Layers 1 and 2: 32 Filters each, 3x3 Layers 3 and 4: 64 Filters each, 3x3	Layers 1 and 2: 32 Filters each, 3x3 Layers 3 and 4: 64 Filters each, 3x3	Layers 1 and 2: 32 Filters each, 3x3 Layers 3 and 4: 64 Filters each, 3x3
MaxPooling Layers	One 2x2 Layer	One 2x2 Layer	One 2x2 Layer	One 2x2 Layer	One 2x2 Layer	One 2x2 Layer	One 2x2 Layer	One 2x2 Layer	One 2x2 Layer	One 2x2 Layer	One 2x2 Layer
Dropout Layers	Layer 1: 0.25 Layer 2: 0.5	Layer 1: 0.25 Layer 2: 0.5	Layer 1: 0.25 Layer 2: 0.5	Layer 1: 0.25 Layer 2: 0.5	Layer 1: 0.25 Layer 2: 0.5	Layer 1: 0.5 Layer 2: 0.5	Layer 1: 0.6 Layer 2: 0.6	Layer 1: 0.5 Layer 2: 0.5	Layer 1: 0.5 Layer 2: 0.5	Layer 1: 0.5 Layer 2: 0.5	Layer 1: 0.8 Layer 2: 0.8
Image Size	50 x 50 for each Model										
Batch Size	30 for each Model										
Epochs	10	10	30	30	30	20	20	30	30	40	30
Training Accuracy	0.9292	0.9583	1.0000	0.9958	0.9917	0.9750	0.9292	0.9958	1.0000	1.0000	0.9208
Training Loss	0.1624	0.1238	0.0134	0.0178	0.0173	0.0728	0.1728	0.0179	0.0163	0.0052	0.2150
Training Duration *(Seconds)*	24.8757	25.1749	73.9137	74.0167	73.9785	49.5853	49.9358	73.8875	72.3754	100.36	74.371
Validation Accuracy	0.850	0.8667	0.8667	0.8667	0.8667	0.8333	0.8333	0.9000	0.8667	0.8667	0.8667
Validation Loss	0.5394	0.6229	1.0084	1.0713	1.0790	1.2449	0.9049	0.8834	0.9727	1.0137	0.5347
Padding	'Same'	'Same'	'Same'	'Same'	'Same'	'Same'	'Same'	'Same'	'Valid'	'Valid'	'Same'
Validation Duration *(Seconds)*	0.8433	0.8642	0.8637	0.7991	0.8147	0.8333	0.8661	0.8008	0.8098	0.8287	0.8382

Plot of Loss and Accuracy of the best Training Model (Model 8)



Model with Best Accuracy (Model 8):

We selected Model 8 to be the model with the best accuracy, because it has the highest Validation Accuracy of 0.9. Its Validation Loss of 0.8834 is also lower than all the other Models except for Models 1, 2 and 11.

Although Models 3, 9, 10 have higher Training Accuracies of 1.0 as compared to Model 8, we did not select them to be the more accurate Models. We feel that Validation Accuracy is more important than Training Accuracy when selecting a better model. High Training Accuracy combined with a low Validation Accuracy could be a sign of overfitting. Validation Accuracy is a result of using test data, hence it is more telling of real accuracy.

Training Duration for Model 8 at 74 seconds is longer than the other models which had 10 or 20 Epochs. This is due to Model 8 having 30 Epochs, which takes longer to train. However, Model 8 made up for this longer duration with its better Validation Accuracy.

Reflection of Lessons Learnt in building the network:

We found that by increasing the number of Epochs from 10 in Model 1 to 30 in Model 8, the Training Accuracy was improved. However, there is a downside to this. Increasing the number of Epochs could also increase Validation Loss and potentially lead to overfitting. We observed that Models that had more Epochs all had higher validation loss results. Therefore, care must be taken not to overfit the model when we try to increase the accuracy scores via increasing Epoch count.

We also experimented with adding 2 more Conv2D layers with 64 Filters each in Models 6 and 7 to see if it could improve the Accuracy and Loss of the models. Models 6 and 7 had 20 Epochs each. The modelling results show that adding in 2 more Conv2D layers and having 20 Epochs did not improve the results either.

After this, we gave Model 8 the same parameters as Model 7, except for the number of Epochs which was increased to 30. This resulted in the best Validation Accuracy and comparatively lower Validation Loss. We concluded from this that increasing the number of Conv2D 64 Filter layers, as well as increasing the number of Epochs to 30 was the key to getting the best results from our dataset.

To test the results further, we increased the number of Epochs to 40 in Model 9. This did not get better accuracy scores. Accuracy score increase seemed to plateau at around 30 Epochs.

Next, we found that by changing the padding method from ‘same’ to ‘valid’ in Models 9 and 10, the Validation Scores did not increase. This shows that changing the padding methods did not influence the results to be better.

Lastly, we experimented with increasing the Dropout Rate from 0.5 in Model 8 to 0.8 in Model 11. Dropout increase resulted in lower Validation Loss for Model 11. However, Validation Accuracy for Model 11 was lower than Model 8. In this case study, we think that Dropout Increase could lower Validation Loss and prevent Overfitting. More tests in the future could be done to see the effect of Dropout on modelling results.

In conclusion, we consider that a dual approach of increasing the Epochs to 30, as well as giving it two more Conv2d layers with 64 Filters had the best positive influence on the model’s accuracy.

On a side note, we also noticed from the model predictions that our models were not predicting test images as mixed fruit. This would explain why we were unable to break past 91.6% validation accuracy (minimum 5 mixed fruits out of 60 test images wrong). This demonstrates that It can be difficult to accurately train and predict images with different objects, and that more care needs to be taken for such images.

We also experimented training with cropped fruits based on the XML files. While this model could identify individual fruits (cropped from test images) decently (86% to 88%), it will take a lot more coding and knowledge to apply it to actual test images.