TOMATO GRADING SYSTEM WITH COMPUTER VISION

Say Jin Tee (1), Abdul Rashid bin Mohamed Shariff (2)

[1] Universiti Putra Malaysia,Faculty of Engineering, Department of Biological and Agricultural Engineering, Jalan Universiti 1 Serdang, 43400 Seri Kembangan, Selangor, Malaysia
Email: tsayjin0507@gmail.com, rashidsnml@gmail.com

KEY WORDS: TensorFlow, OpenCV, Surface, Defect, Size

ABSTRACT:

In Malaysia, tomato is one of the few vegetables that are able to penetrate domestic retail sector as well as export. Grading is a key measure to commercialize the produce where surface defects and size are two important parameters to be checked in order to determine its quality. For post-harvest treatments of tomato, surface defect detection and size measuring are carried out independently. This has greatly increased the labour usage, energy and cost. Moreover, the difficulty with the traditional machine learning approach makes surface defects inspection a tedious work. With the great progress of deep learning in the field of computer vision, the application of deep learning in image classification has become more popular. The proposed tomato classification system consists of two main parts: 1) Surface defect detection and 2) Size measuring. For the first part of the model, deep learning approach, specifically Convolutional Neural Network (CNN) is applied as it involves non-linear and complex parameters. For the second part of the model, an algorithm is developed with Open Source Computer Vision (OpenCV) library for tomato size measuring by calculating Euclidean Distance. Both of these models are merged as a robust tomato classifier. Both of these models are developed and tested separately. The accuracy obtained for surface defect detection model developed in this study is 97.61% while the algorithm of tomato size measuring give the Root-Mean-Square-Error (RMSE), normalized RMSE, Index of agreement and Nash-Sutcliffe Model Efficiency (NSE) values of 0.258 mm, 0.00517, 1.000 and 0.999, respectively.

1. INTRODUCTION

Over the past few decades, there has been enormous research carried out to assist in the quality assessment of fruits and vegetables in the industry, including tomato. The efficiency and effectiveness of grading governs the quality standard of the packing lines and the product, which, in turn, determines the marketability of the product (Satpute & Jagdale, 2017). The traditional grading techniques are labour-intensive, subjective, time consuming and affected by inconsistencies of the judgment by different operators. Consequently, there is a growing interest in the use of non-contact techniques i.e. Computer Vision System (CVS) to automate this process. The CVSs are the essential aspect towards improved post-harvest technologies. These systems offer many advantages, such as consistency and uniformity in grading, speed, and accuracy in sorting. Moreover, they have a high success rate and non-destructive methods (Nyalala et al., 2019). At present, the development of CVSs is focused on defining new methods for the evaluation of fruit quality. In the fruit inspection industry, the Support vector machine (SVM), k-nearest Neighbour (KNN), Artificial Neural Network (ANN), and Decision Tree (DT) pattern classification methods are between the most common (Castro et al., 2019).

With the great progress of deep learning in the field of computer vision, the application of deep

learning in image classification has become more popular. The CNN represent a huge breakthrough in image classification and have emerged as the master algorithm in computer vision in recent years. With CNN, the concept of convolution of image and filters is applied to generate invariant features which are passed on to the next layer. The features in next layer are convoluted with different filters to generate more invariant and abstract features and the process continues until the final output is obtained. This eliminates the need for manual feature extraction. Each layer increases the complexity of the learned features.

In addition, OpenCV which was designed for computational efficiency is proposed to be used for several operations in this project such as image pre-processing, edge detection and contour detection. Moreover, in order to accurately compute the size of tomato, the Euclidean Distance is computed between the pre-found sets of midpoints of the target object i.e. tomato.

The principal objective of this study is to develop a tomato grading system in terms of surface defects and size with computer vision. There are several specific objectives to achieve in this project which are:

1. To study and demonstrate the feasibility of using deep convolutional neural networks to classify tomato in terms of surface defects.
2. To develop tomato size measuring algorithms with Open Source Computer Vision (OpenCV) Library.
3. To test the accuracy of the algorithms developed for measuring size of tomatoes.

2. METHODOLOGY

2.1 Introduction

The proposed tomato grading system consists of two parts, namely 1) Tomato surface defect detection and 2) Tomato size measuring. Both of these models were developed and tested separately and merged as a robust classifier. The tomato will be sorted to a particular class if defect is detected on the surface area of tomato. In contrast, the tomato will proceed to size measuring algorithm if it was found in healthy condition. There are a total of 7 grading categories in terms of size according to Malaysian Standard.

2.2 Tomato Surface Defect Detection Model

The tomato surface defect detection model was built by using Transfer Learning Method with deep Convolutional Neural Network. The training data was split into mini batches for efficient training. Each batch of data consists of 32 images. Thus, 32 images have been worked through before updating the internal model parameters.In deep Convolutional Neural Network, the early layers were known to identify simple shapes like lines, edges and corners. Most of the computer vision problems i.e. image classification tasks involve similar low-level patterns in the early layers of a convolutional architecture. Hence, a pre-trained model with convolutional architecture, which has been trained on large data-set was re-purposed by transferring the learned features from the pre-trained model to the proposed model. New layer was then added to the model in order to fit the specialized task that matches the desired data classification problem i.e. defects or healthy. The configuration of the model was shown in Figure 1. As shown in the figure, the total parameters of the model were 1,345,053. By taking the advantage of pre-trained model, the early layers which act as the feature extractor were frozen where 1,343,049 parameters were non-trainable. The last few layers were retrained with the tomato data-set which includes 2,004

parameters.

```
[ ] model =create_model()
    model.summary()

    Building model with: https://tfhub.dev/google/imagenet/mobilenet_v1_050_160/classification/4
    Model: "sequential"
    _________________________________________________________________
    Layer (type)                 Output Shape              Param #
    =================================================================
    keras_layer (KerasLayer)     multiple                  1343049
    _________________________________________________________________
    dense (Dense)                multiple                  2004
    =================================================================
    Total params: 1,345,053
    Trainable params: 2,004
    Non-trainable params: 1,343,049
    _________________________________________________________________
```

Figure 1. Configuration of the proposed model

2.3 Tomato Size Measuring Algorithm

The tomato size measuring algorithm was developed with OpenCV Library on a Python integrated development environment. The input image was converted to grayscale and smoothed with Gaussian Filter. The process was followed by a Canny Edge Detector edge detection. The edge detection was performed along with a dilation and erosion operation to close the gaps in between edges in the edge map.

After pre-processing the image, the contour of the tomato was searched within the image captured using *findContour* function in OpenCV. Each of the individual contours of the input image was looped. The contour was discarded if it was not sufficiently large i.e. not larger than 100. It was presumed to be noise and left over from the edge detection process. The bounding box of the target object was then computed, provided the contour region was large enough. Meanwhile, the outline of the object i.e. tomato was drawn in green, and the vertices of the bounding box rectangle was drawn as small and red circles. The first Euclidean distance was computed between the midpoint of top and bottom line whereas the second Euclidean distance was computed between the midpoint of left and right line. The resultant lines were then drawn and displayed.

3. RESULTS AND DISCUSSION

The tomato surface inspection model and the size measuring algorithm are analysed separately with appropriate assessment indicators. Result of the combined classifier are discussed in terms of its accuracy, precision and speed.

3.1 Tomato Surface Defect Detection Model

Figure 2 depicts the training process of the model. The application of Early Stopping function via call back has avoided over-fitting and under-fitting of the model. The training was kept going until the there is no improvement on the accuracy of validation data set for three epochs continuously. The training stopped at the 13th epochs with a validation accuracy of 97.61%.

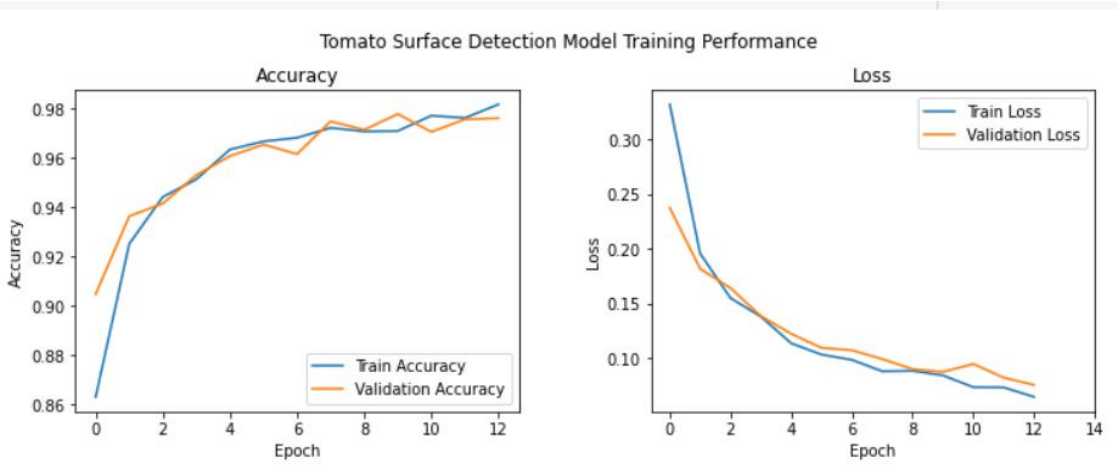

Figure 2. Performance curves (left: Accuracy and right: Loss) of Tomato surface detection model

In addition, the training algorithm is evaluated during each step of model training. Figure 2 depicts the model learning curves in terms of accuracy (left) and loss (right) over number of epochs. These plots suggest that the model has a good fit on tomato surface defect detection as the the training and validation loss have reduced to a stabilization point with a minimal gap between the two final loss values at epoch 13. The minimal gap is referred as gap of generalization. Meanwhile, the accuracy of the model has increased over number of epochs. Continued training of a good fit model is likely lead to an over-fit.

Furthermore, evaluation metrics such as Recall and Precision are computed. The model was assessed on the validation data sets which are not part of training data. Table 1 shows the performance of model for each epoch of training and the final result was tabulated. The final loss, validation accuracy, precision and recall are 0.0761, 97.61%, 0.975 and 0.978, respectively. The classification accuracy of 97.61% on validation data-set indicates the model is reliable. Meanwhile, the validation precision of 0.975 indicates nearly all defects tomato have been correctly classified. Last but not least, the value of Recall indicates the model generated results that correctly predicted positive observations (True Positive) to all observation in the actual class of defect tomato (Actual Positives) with a sensitivity of 0.978.

Table 1. The overall evaluation result of tomato surface defects detection model

Evaluation Metrics	**Model Evaluation Result**
Loss	0.0761
Accuracy	97.61%
Precision	0.975
Recall	0.978

The performance of the tomato surface defect detection model is visualized with confusion matrix as shown in Figure 3.

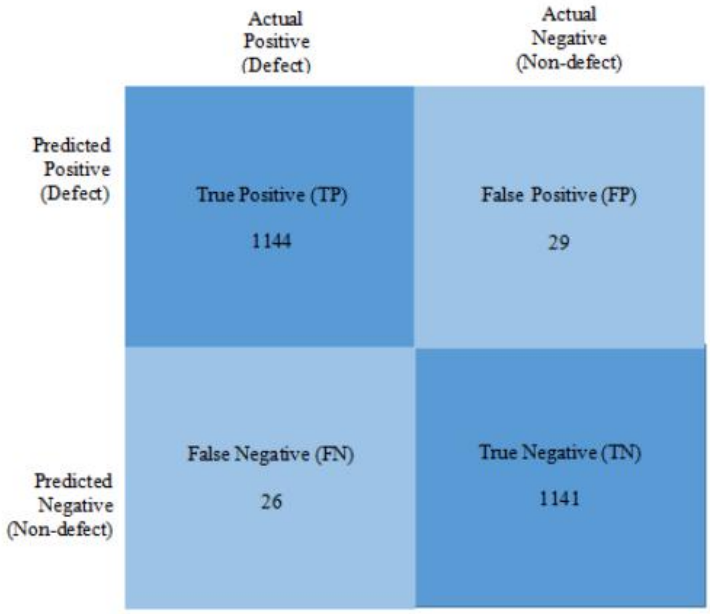

Figure 3. Visualization of the performance of tomato surface defect detection model

As shown in Figure 3, 1144 tomatoes are correctly classified as defect tomatoes and 1141 tomatoes are correctly detected as non-defect tomatoes from the validation data-set. On the other hand, 29 non-defect tomatoes are wrongly classified as defect tomatoes and 26 defect tomatoes are wrongly classified as non-defect tomatoes. This is because some of the defects such as scar is not obvious and less visible compared to other kind of defects.

Figure 4-6 depict some examples of tomato surface defect detection result. The detection result with its confidence level is shown on the top left corner. The threshold was set at 0.5.

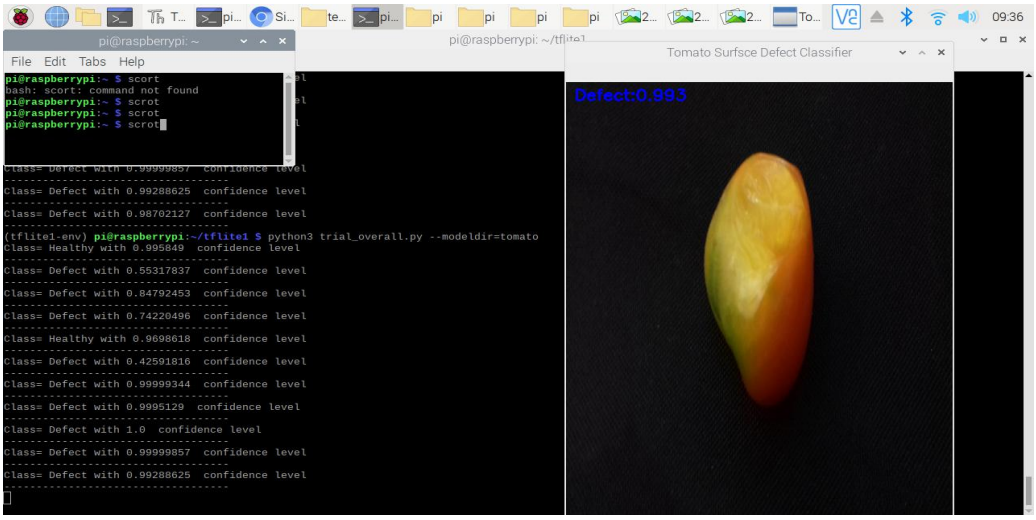

Figure 4. Classification result of tomato surface defect detection model (1)

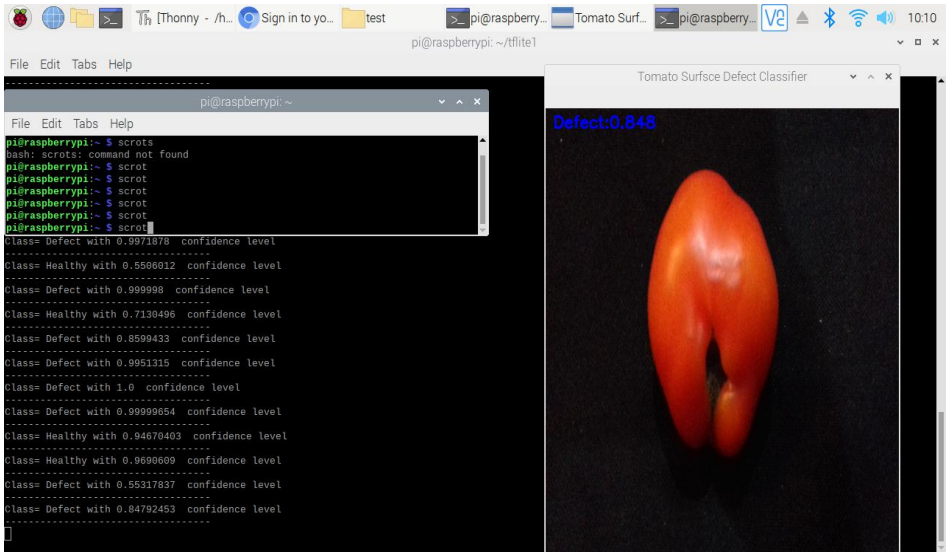

Figure 5. Classification result of tomato surface defect detection model (2)

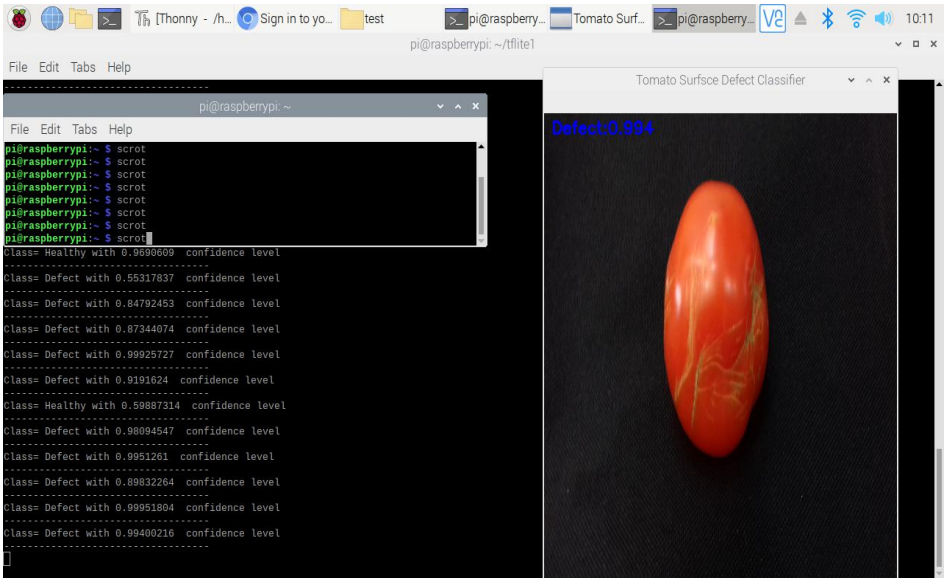

Figure 6. Classification result of tomato surface defect detection model (3)

3.2 Tomato Size Measuring Algorithm

The size of tomato is measured if the tomato is classified as healthy tomato. Length and width of tomato are measured and the major axis of tomato is involved in tomato grading. Figure 7 shows the result of size measuring on the Python development environment, PyCharm. The major axis of the tomato was used to classify the tomato into different category. The bounding box is drawn and the size of tomato is measured by computing the Euclidean distance between the midpoints of the line of bounding box. Time required for size measuring per image is 0.01seconds.

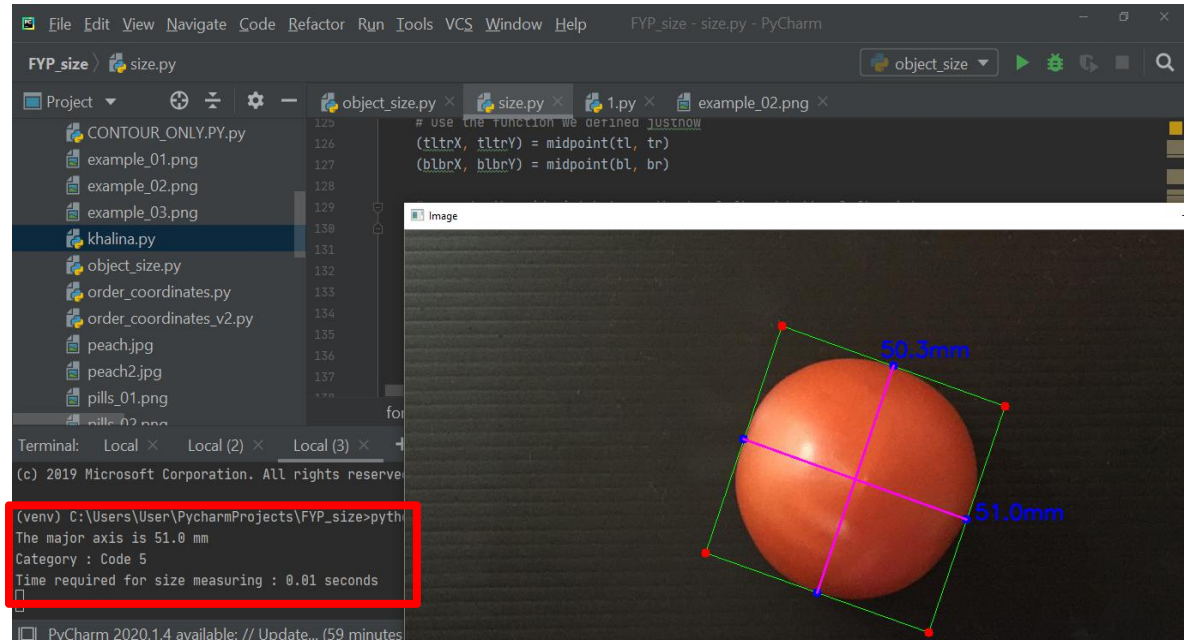

Figure 7.Tomato size measurement with the proposed model

By comparing the actual size to the computed size of tomato, the difference of the actual size and computed size are calculated. The RMSE, nRMSE, Index of agreement and NSE values are calculated as given by Table 2. 30 tomato samples are used for evaluation. The RMSE value of 0.258 mm indicates the model of size measuring fits the data well. The larger the RMSE, the larger the difference between the computed size and the actual size, which means the worse the model fits the data. The RMSE is further normalized to 0.00517. The nRMSE value is scale-free thus it is comparable for simulating parameters with different units.

Secondly, the Index of agreement represents the ratio of the mean square error and the potential error. The agreement value of 1 indicates a perfect match, and 0 indicates no agreement at all. Hence, with the result of 1 indicates the almost all of the tomato samples are measured correctly. In total of 30 samples, 2 tomato samples were measured wrongly by the program with 1 mm difference compared to the actual size. This may due to the round off of measurement. The top down image of tomato with the camera located 90 degree on top of the grading compartment give nearly perfect result. Lastly, NSE which is the coefficient of determination obtained a value of 0.999. A value of 1 would indicate a perfect model while a value of zero indicates performance no better than simply using mean. With this result, the performance of the model is considered good by giving accurate size measurement. Overall, the tomato size measuring algorithm gives a good result by calculating the Euclidean distance between sets of midpoints.

Table 2.The statistical result of RMSE, nRMSE, Index of agreement and NSE

Variable	Healthy Tomato
No. of samples	30
RMSE	0.258 mm
nRMSE	0.00517
Index of agreement	1.000
NSE	0.999

3.3 The Combined Tomato Classifier

The tomato surface defects detection model and size measuring model are merged as a robust

classifier. The classification result on Raspberry Pi is shown in Figure 8 and run on the Raspberry Pi. The result of surface defects detection is displayed on top left corner whereas the length and width of tomato is displayed on their respective location. The time taken for the grading process was about 0.15-0.16 seconds per image on Raspberry Pi. The Raspberry Pi works seamlessly with Python and OpenCV-Python.

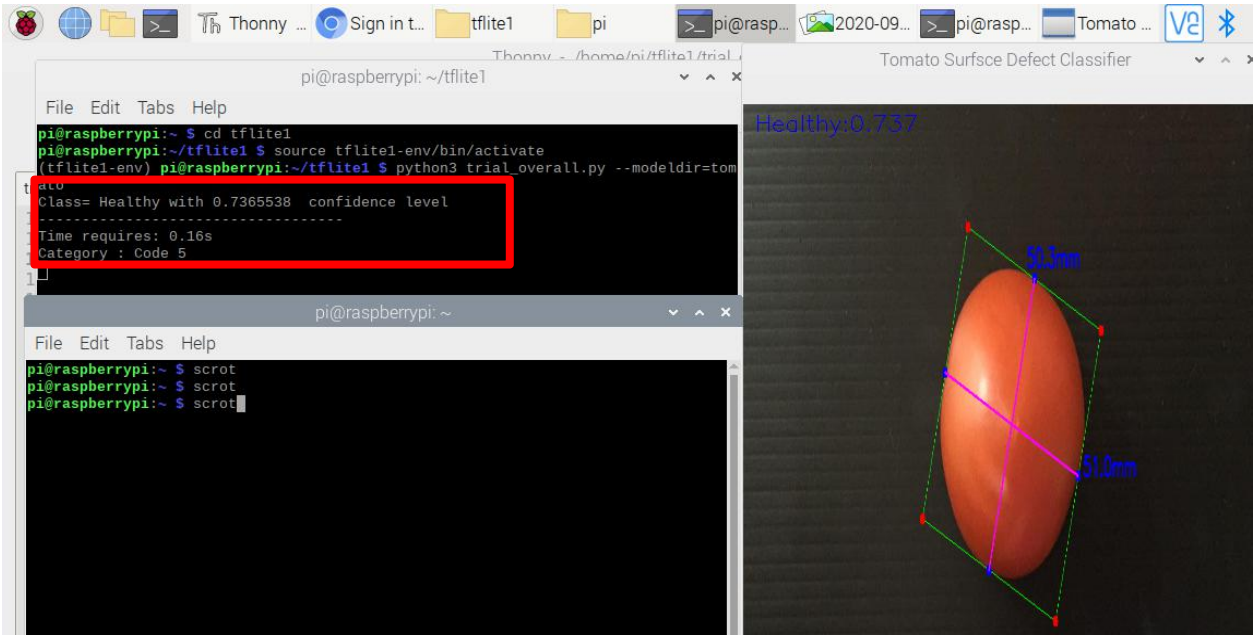

Figure 8.Classification result of the combined classifier

In this project, one camera is used on Raspberry Pi due to the limitation of computational power. Multiple cameras should be used for 360 degree inspection of tomato. Besides that, in practical implementation, the camera can be substituted with a high speed camera for large scale tomato processing factory, depending on the preference of the operator. According to the TensorFlow official documentation, The Version 2 MobileNet can easily run at over 240 FPS. However, the speed of classification for the proposed model is greatly dependent on the model of camera as a clear image is needed for size measuring at second stage. Thus, a high speed camera is suggested to be used for rapid movement of tomato on the conveyor belt.

4. CONCLUSION AND RECOMMENDATION

The objectives of this study have been achieved. A reliable tomato classifier has been successfully built in terms of surface defects and size with computer vision. In the proposed system, the tomato is classified based on its appearance, followed by size measurement.

Grading of multiple fruit at same time is recommended for future work to increase the efficiency of tomato grading. For software, different kind of pre-trained model can be tested or modified to obtain the best model. Further studies should be conducted to improve the algorithm for existing tomato surface defect detection and size measuring.

Last but not least, as image classification involves thousand or more parameters, the increase of training data can potentially increase the accuracy of the model. Hence, a diverse and large data-set is recommended to be used to further improve the performance of the model.

5. REFERENCES

Castro, W., Oblitas, J., De-La-Torre, M., Cotrina, C., Bazan, K., & Avila-George, H., 2019. Classification of Cape Gooseberry Fruit According to its Level of Ripeness Using Machine Learning Techniques and Different Color Spaces. IEEE Access, 7, pp.27389–27400.

Nyalala, I., Okinda, C., Nyalala, L., Makange, N., Chao, Q., Chao, L., Yousaf, K., & Chen, K., 2019. Tomato volume and mass estimation using computer vision and machine learning algorithms: Cherry tomato model. Journal of Food Engineering, 263(July), pp.288–298.

Satpute, M. R., & Jagdale, S. M., 2017. Automatic fruit quality inspection system. Proceedings of the International Conference on Inventive Computation Technologies, ICICT 2016, 1, pp.1–4.