

Bone Belt Monitoring

Bone Belt Monitoring - Stage 2

Project Code
2021-1260

Prepared by
Prof Peng Shi, Dr Qi Wu and Dr Xin
Yuan

Published by
AMPC

Date Submitted
21/03/2022

Date Published
21/03/2022

Contents

Contents	2
1.0 Executive Summary	3
2.0 Introduction	4
2.1 Existing Solutions	4
2.2 Computer Vision	5
2.3 Machine Learning	5
3.0 Project Objectives	7
4.0 Methodology	8
4.1 System Development Overview	8
4.2 Object Detection Model Development	8
4.3 Single Board Computer (SBC) Execution	14
5.0 Project Outcomes	15
5.1 Beef Parts Detection	15
5.2 Lean Meat Detection	16
5.3 Performance Estimation	17
5.4 SBC Execution	18
6.0 Discussion	19
6.1 Beef Parts Detection	19
6.2 Lean Meat Detection	19
6.3 Performance Estimation	19
7.0 Conclusions / Recommendations	20
8.0 Bibliography	21
9.0 Appendices	22

Disclaimer The information contained within this publication has been prepared by a third party commissioned by Australian Meat Processor Corporation Ltd (AMPC). It does not necessarily reflect the opinion or position of AMPC. Care is taken to ensure the accuracy of the information contained in this publication. However, AMPC cannot accept responsibility for the accuracy or completeness of the information or opinions contained in this publication, nor does it endorse or adopt the information contained in this report.

No part of this work may be reproduced, copied, published, communicated or adapted in any form or by any means (electronic or otherwise) without the express written permission of Australian Meat Processor Corporation Ltd. All rights are expressly reserved. Requests for further authorisation should be directed to the Executive Chairman, AMPC, Suite 2, Level 6, 99 Walker Street North Sydney NSW.

1.0 Executive Summary

This project will present a conceptual design to detect the proportions of lean meat left on the bones after the deboning process. This has been attained by applying various computer vision and machine learning concepts. An economical device will be fabricated to perform real-time detection of lean meat. The primary goal of this project is to demonstrate a novel approach using computer vision for a beef bone belt monitoring system that aims to measure the amount of red meat leaving a meat processing plant via the bone belt after the butchering process.

The conceptual system is developed based on 200+ images collected on two bone parts of the beef butt. Detection models are developed based on Mask Region-Based Convolutional Neural Networks. The system is constructed by a two-layer detection model. The first layer detects and extracts the beef part in the frame, and the second layer detects the lean meat left on the detected parts. The system provides a KPI based on the percentage of lean meat left on the beef part. The results show that the beef parts can be detected by the system with over 95% of confidence, and the majority of lean meat on the beef parts can be detected.

This project provides an example of using computer vision technologies to monitor beef processing performance. For future developments, AMPC can coordinate developments of different detection models for various functions. The detection system can be loaded to a small and low-cost device and deployed to the processing sites of AMPC members without any major upgrades or modifications.

2.0 Introduction

2.1 Existing Solutions

No pre-existing lean meat detection solutions have been deployed to a bone belt at present. However, there exist some technologies capable of fat analysis functions.

2.1.1 Fat Analysis Solutions

As of today, there have been several remarkable achievements in developing various cutting-edge solutions for fat analysis on the market. This is due to the increasing consumer demands for a better understanding of different information and nutritional values of the food consumed to promote healthier eating lifestyles. These fat analysis techniques include Near Infrared (NIR) transmission or reflection, microwave and X-ray transmission [7]. However, there are some drawbacks to these solutions which will affect the efficiency and efficacy of analysis.

There are two common installation points for fat analysers in the market, at-line (bench/lab) and in-line (before and after the grinder) [7]. The methods applied for analysis at both points are different, but each has its advantage. The at-line installation implements a surface scanning method using NIR reflection and transmission technologies to analyse fat contents [7]. This method allows more parameters (fat, protein, moisture etc) to be measured compared to the X-ray transmission method, however, it is only limited to a 200g size limit [7]. On the other hand, the in-line installation uses microwave and X-ray technologies to scan through all types of meat whether it is fresh or frozen with no size limitation and is capable of producing a roughly same or higher fat accuracy measurement.

The sections below will further detail several options available in the market today.

2.1.1.1 Eagle's Inline Fat Analysis X-ray Systems

This fat analysis system uses the Dual Energy X-ray Absorptiometry (DEXA) technology to obtain a precise measurement of chemical lean (CL) with a tolerance of +/- 1 CL [6]. The CL value is a numerical value that represents the crude fat content of meat [10]. This system is capable of producing accurate fat measurements, improving contaminant detection as well as mass measurements [6]. Several imaging analysis software has also been developed by Eagle to be executed together with their detectors to simplify calibration routines for a more efficient process.

2.1.1.2 FOSS's Rapid Fat Analysis

The MeatMaster II, a product of FOSS is an in-line installation device using X-ray technology with a capacity of up to 38 tons/hour for various meat analyses [7]. These analyses include fat content, weight and foreign objects detection like metal and bone [7][6]. The MeatMaster II is the only in-line fat analyser device that is installed before the coarse grinder. This is so it could detect raw materials which may contaminate the grinder and remove them before further analyses are done [7].

2.1.1.3 Near-Infrared (NIR) Transmission/Reflection

The NIR fat analysis method involves transmitting NIR light beam through meat samples or reflecting NIR light off a meat sample surface [6]. There exists a limit to the range of NIR wavelengths which affects the amount of information attained from the meat.

2.1.2 Other Existing Solutions

2.1.2.1 Fluorescence Spectroscopy

Fluorescence spectroscopy is a type of electromagnetic spectroscopy that analyses fluorescence from a sample [1]. This technology has been known for its rapid diagnosis ability together with its high sensitivity and specificity rate [1].

It has been applied widely in the medical field for various purposes including the detection of soft-tissue abnormality, the biological activity of flora as well as other microbial activity [1]. It works by having the sample excited with a beam of light which results in the emission of light of lower energy used to interpret results.

When integrated into the real world, fluorescence spectroscopy technology has proved to provide more information as compared to rRNA and DNA bacterial homology grouping as it gives more information about the strain relatedness as well as demonstrates a high level of differentiation quality between strains [1].

2.2 Computer Vision

Computer vision acquires meaningful information from images, videos and other visual inputs. With the growth of computational power and by using various deep learning models and algorithms [11], more useful information and knowledge could be attained from images. It is the computer version of the human vision with our retinas, optic nerves and visual cortex replaced with cameras, data and algorithms [11], yet with higher processing capability.

In recent decades, computer vision combines with a type of machine learning (ML) called deep learning as well as a convolutional neural network (CNN) which uses a large amount of visual data to help it understand what it is looking at. The unique thing about computer vision is when these three essential contributors are combined precisely, it will produce an outcome all by itself that will fascinate mankind. The agent would be able to look at the labelled data provided and subsequently teach and learn by itself with the help of a CNN to ultimately differentiate one image from another.

An example of computer vision in the agriculture industry is the construction of deep learning models to identify individual animals by researchers at Iowa State University [2]. This development aims to free up labour in feedlots and barns in the future to improve production efficiency. These livestock will be evaluated with computer vision algorithms using a camera to ensure enough food are provided and they are acting healthily. It takes lesser time to notice abnormalities in an animal with a machine than a human, therefore this buys the animals some time to get treated before it is too late. Veterinarians will also be able to diagnose observable medical issues of animals remotely with this technology.

The medical industry is one of the many industries that has been extensively applying concepts of computer vision to obtain higher quality medical imagery. A typical example is the Project InnerEye by Microsoft Research Cambridge to democratise AI for medical image analysis as well as to deliver precise medical analysis for better patient outcomes [8]. It employs algorithms such as the latest CNN for the automatic, voxelwise segmentation of medical images [8]. The collaborators of the project are also working on developing their deep learning models for personalised therapy delivery.

2.3 Machine Learning

Machine learning belongs to another sub-branch of AI which uses data and algorithms as a guide to emulating human intelligence by learning from the surrounding environment. The computational process involves inputting a series of data to achieve a specific task without explicitly programming it to do so [5]. A machine learning algorithm shall then adapt itself through repetitive cycles of training to build up its 'experience', so it produces more accurate results over time. Several numerical parameters could also be adjusted to optimise the performance of the algorithm.

There are three main types of machine learning namely supervised learning, unsupervised learning and semi-supervised learning [5]. In this project, we will be focusing on supervised learning in which each training data is paired with its known classification label.

2.3.1 Mask Region-Based Convolutional Neural Networks (R-CNN)

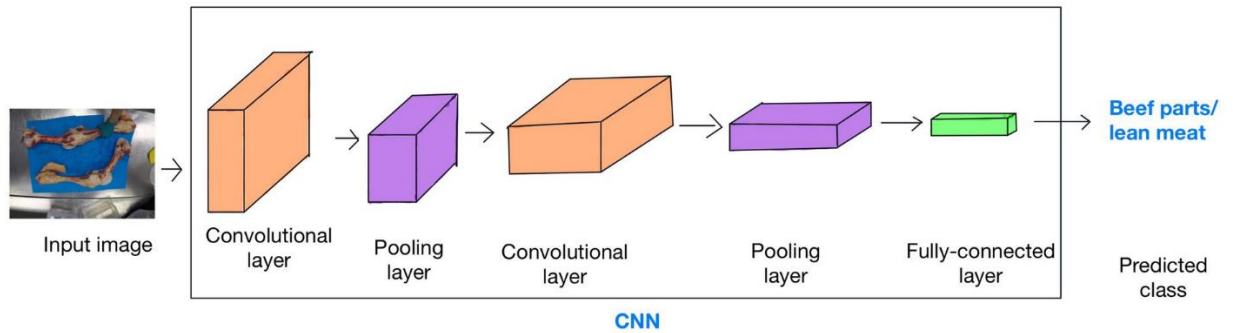


Figure 1 Concept of CNN Architecture [5]

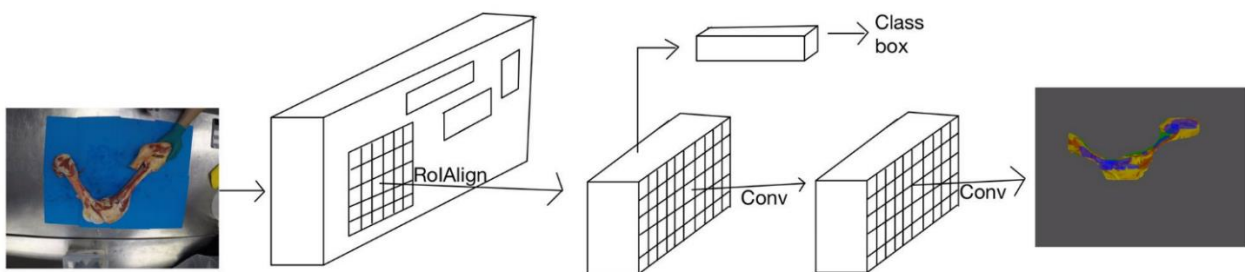


Figure 2 The Mask R-CNN Framework for Instance Segmentation [5]

Mask R-CNN is a CNN that has been widely used for image and instance segmentation. There are three main layers to the CNN, they are the convolutional layer, the pooling layer and the fully connected layer [4]. The convolution layer is responsible for extracting the input image as a feature map with the help of filters and kernels [4]. Moving on to the pooling layer, it works by downsampling the feature maps by encapsulating the presence of patches on them [4]. Lastly, the fully connected layer will connect every neuron from one layer to another [4]. These three layers then combine to form the CNN which helps the neural network in identifying and recognising the objects of interest in an image as shown in Figure 1.

R-CNN builds on the concept of CNN, however, it uses bounding boxes across the object regions, which then assess convolutional networks independently on all the Regions of Interest (RoI) to classify different image regions into the proposed classes. Fast R-CNN is an improved version of R-CNN which involves two stages while Mask R-CNN was developed on top of Fast R-CNN with an additional step of providing a mask output. This project focuses on the Mask R-CNN method to perform lean meat detection.

The first step into Mask R-CNN is the concept of image segmentation which could be divided into semantic segmentation and instance segmentation. This project uses the latter segmentation method where all objects to be detected in the image are precisely labelled. The main element that sets Mask R-CNN apart from Faster R-CNN is the pixel-to-pixel alignment feature. The first stage of Mask R-CNN is the Region Proposal Network (RPN) which portrays the various objects of interest within an image. In the next stage, Mask R-CNN will abstract and classify features found in the candidate box and perform bounding box regression. The final stage will be where Mask R-CNN predict the class and output a binary mask for each RoI as shown in Figure 2. **Error! Reference source not found.**

3.0 Project Objectives

The output of this stage is a detailed design, sufficient to:

- ◆ Demonstrate the working functionality of the concept(s) (on paper and through simulations/animations)
- ◆ Develop indicative targeted \$RRP,
- ◆ Demonstrate feasibilities and performances of the proposed approach, and
- ◆ Draft a plan for the next stage

4.0 Methodology

4.1 System Development Overview

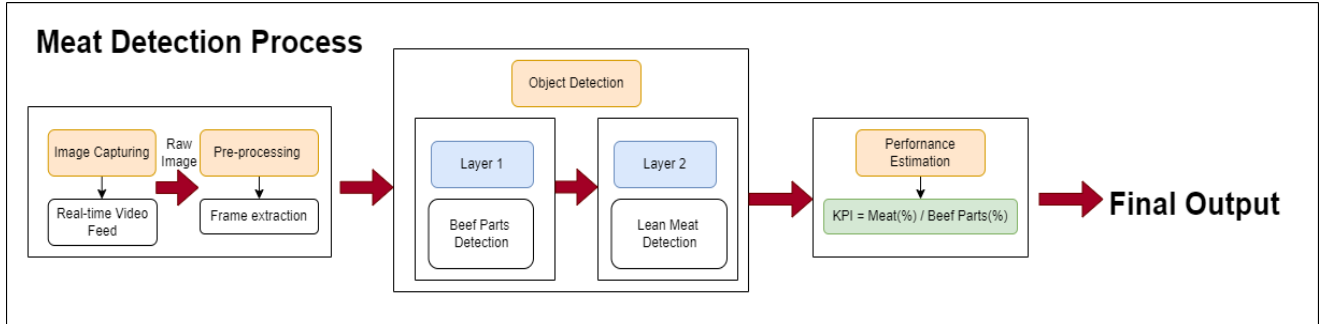


Figure 3 System Structure

Figure 3 shows an overview of the conceptual system structure of the proposed real-time lean meat detection system. The conceptual system is constructed by an image capturing and pre-processing module, an object detection model and a performance estimation module. The image capturing and pre-processing module gets video feeds from the camera, and frames are extracted in real-time. Various adjustments can be performed on the extracted frames depending on the resolution of the camera and extracted frames to deal with poor resolution and overlapping frames. The processed image frames are then passed into the object detection module to detect beef parts and to classify lean meat found on these parts. This is a two-layer detection model, and the development details will be further discussed in section 4.2. Finally, the performance estimation module in section 4.3 estimates the performance of the deboning process based on the percentage of lean meat left on the beef parts.

4.2 Object Detection Model Development

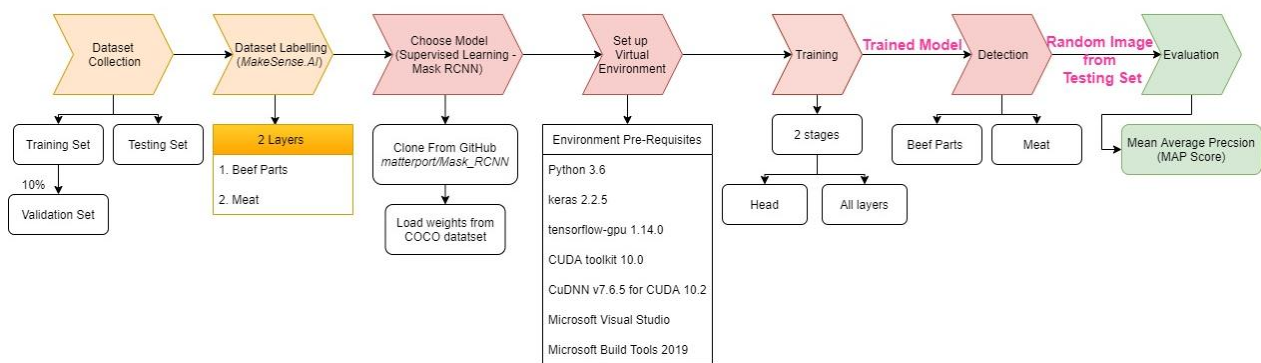


Figure 4 Object detection model development processes

Figure 4 shows the development processes of the object detection model which has been divided into seven primary steps. The development starts from dataset collection (section 4.2.1) and labelling (section 4.2.2), and a suitable model is then chosen (section 4.2.3). Once the model is chosen, an optimum environment is set up (section 4.2.4) to attain the optimal performance of the model. Then, it moves on to the main training process (section 4.2.5) and subsequently into detection (section Detection Stage) and performance estimation (section 4.2.7).

4.2.1 Dataset Collection

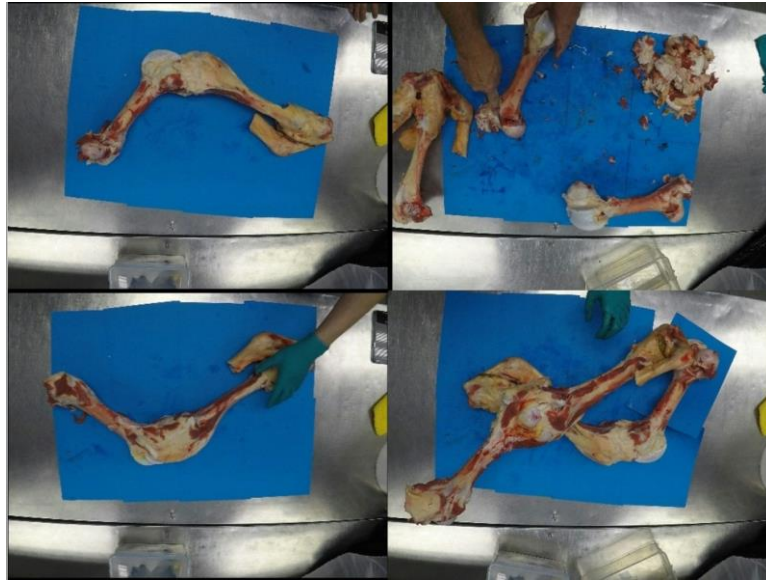


Figure 5 Training Images

For proof of the concept, we only had access to two sets of beef parts images at different angles. Several possible scenarios that may be encountered on a bone belt in the meat processing plant was also taken into consideration like lighting variations, overlapping beef parts as well as having more than one beef part in a single frame as shown in Figure 5. These datasets are then divided into two datasets - the training set and the testing set. A 10% proportion of the images found in the training set has been set apart as the validation set. The validation set provides an unbiased evaluation of a model fit while tuning the model's hyperparameters [2]. The training set is needed to fit the ML model while the testing set is to evaluate the fitted ML model. This part of the process is essential to ensure that models are not tested on trained images that may result in a compromise in model precision.

4.2.2 Dataset Labelling

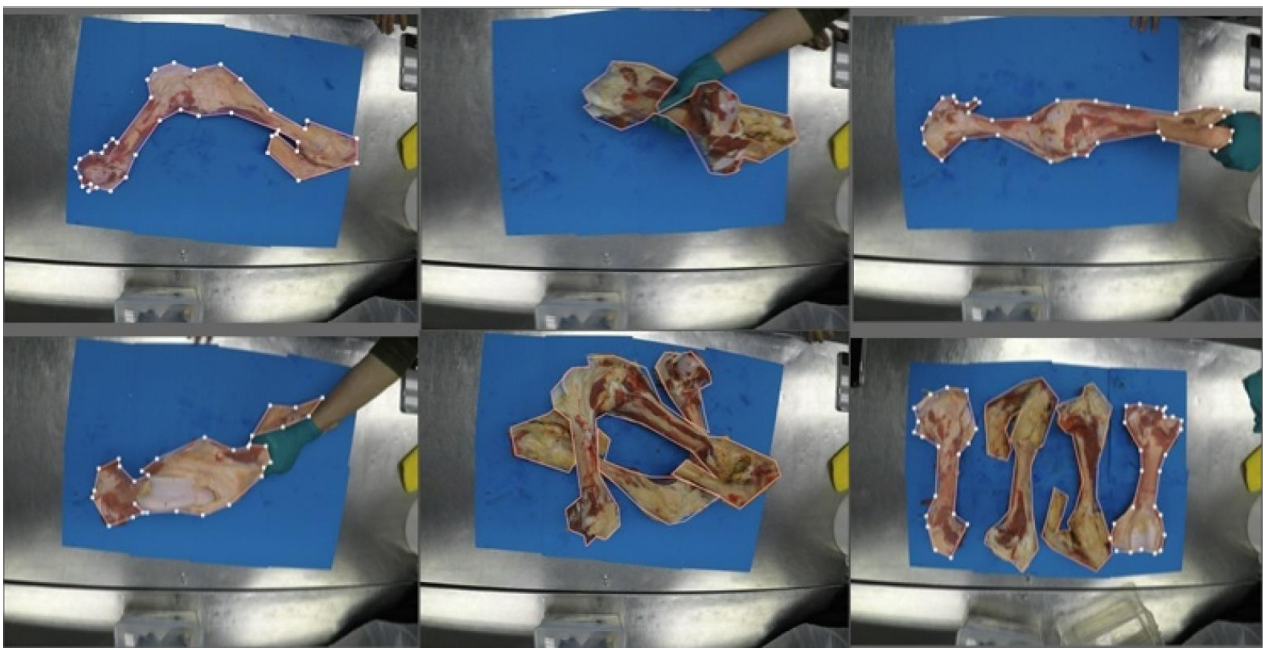


Figure 6 Annotated beef parts images

Before training the Mask-RCNN model for detection, all images in the training dataset shall be labelled and annotated accurately. Makesense.ai has been the primary annotation tool for both stages of detection. Figure 6 shows an example of randomly picked annotated images within the training/validation dataset. A polygon labelling shape has been used to carefully outline the respective beef parts. An outline of the beef parts was labelled for a total of 100 images made up of the training and validation dataset. Labelling of overlapping beef parts has been dealt with by separating the lower layer of beef parts into several sections. Overlapping beef parts were not labelled to avoid inaccuracy assumptions from affecting the training model.

An annotation file for each layer was then generated to be integrated into the training model. Annotating training images is a relatively tedious process, therefore figuring out an efficient way to label an image will produce a better outcome in a shorter time.

4.2.3 Model Choice

A supervised learning method - MaskRCNN has been chosen as it has been commonly known for its powerful capability in image localisation and segmentation. This is considered a supervised learning method as training datasets are accurately labelled with their associated classes. The detection model was not trained from scratch however, it was trained with the trained weights obtained from the MS COCO model. It is a large-scale object detection, segmentation and captioning dataset published by Microsoft. This model was specifically chosen due to its comprehensiveness and its common usage to benchmark different algorithms at most times.

4.2.4 Virtual Environment Setup

The detection model will require a Python 3.6 version environment to perform detection optimally. Various software will be required to ensure that the TensorFlow library runs smoothly. These softwares are listed below together with their respective functions.

- ◆ Anaconda toolkit
 - A data science toolkit consists of a handful of useful tools. However, in this project, we are only using the Jupyter notebook and the Anaconda command prompt.
- ◆ Microsoft Visual Studio 2019
 - Some components of the CUDA toolkit will require the visual studio to work appropriately.
- ◆ NVIDIA CUDA Toolkit 10.2 from CUDA Toolkit Archive
 - A development environment comprising GPU-accelerated libraries, compiler, development tools and the CUDA runtime essential for running the TensorFlow-gpu library. It is an essential toolkit for developing GPU-accelerated applications.
- ◆ NVIDIA CUDA Deep Neural Network (cuDNN)
 - A GPU accelerated library for deep neural networks. It provides highly tuned implementations of routines arising frequently in DNN applications.
- ◆ Tensorflow-gpu 1.14.0
 - An open-source library found in Python for fast numerical computing like machine learning tasks. It was used to create deep learning models.
- ◆ Keras 2.2.5
 - High-level neural network that runs on top of Tensorflow with a focus on deep neural networks.

The cudart64_100.dll file shall be downloaded and placed inside the Anaconda3\\envs\\" *environment name*" path. Three new paths as shown in Appendix A shall be added to the environment of the system to ensure that the model executes smoothly. Relevant links for software installation has also been attached in **Error! Reference source not found.**

4.2.5 Training Process

There were two layers of training involved in the meat detection process. The first layer involves extracting the beef parts from the background and the second layer involved extracting the lean meat from the beef parts.

4.2.5.1 Layer 1 – Beef Parts Extraction

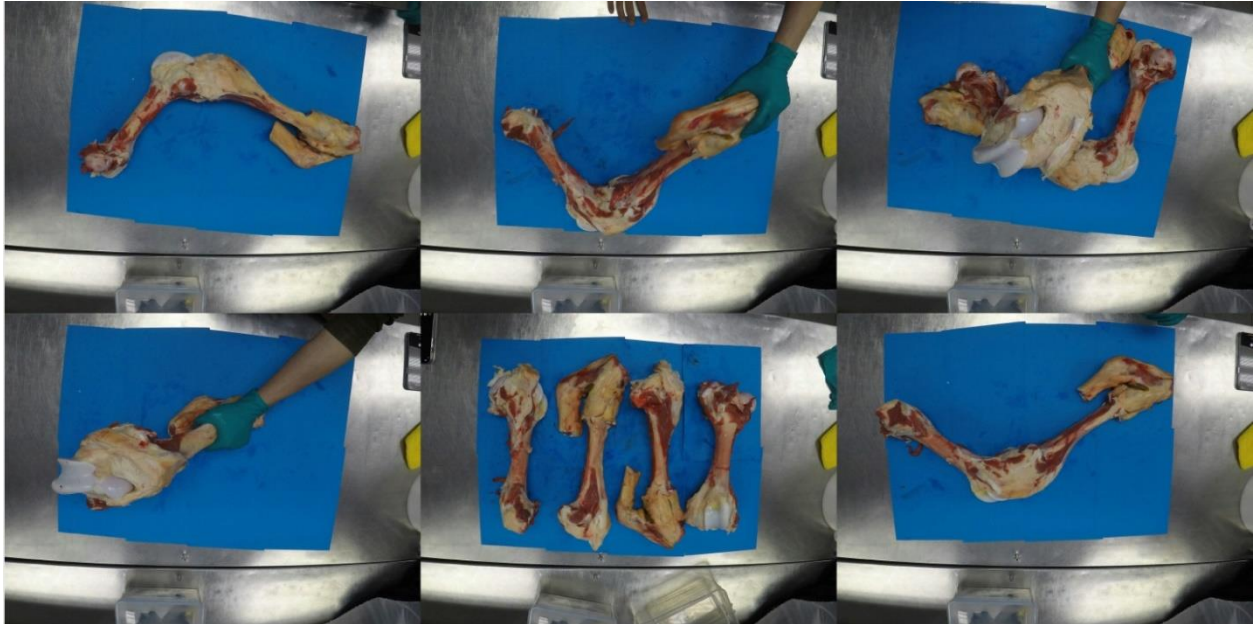


Figure 7 Layer 1 training images

This layer of training was executed with a learning rate of 0.001 in 10 epochs of 100 steps per epoch. Training datasets shown in Figure 7 together with their respective annotations were loaded into the kernel to train the model. Among 100 images that were labelled, 10% of them were randomly picked to set up the validation set which leaves the training dataset with the remaining 90% of images. The MS COCO training model was then loaded to train the model. The head was trained separately prior to training all the other layers to fine-tune the model pre-trained on MS COCO. This also helps to minimise overfitting in the training outcome. Once the head has been trained, the training algorithm moves on to train the remaining layers. The trained model generated after the last epoch will then be used as our detection model in the detection stage subsequently.

The mean average precision (mAP) has been calculated. It is an evaluation metric specifically used in object detection models. The mAP is calculated using the concept of Intersection over Union (IoU) with the formula $IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$, as well as recall [9]. Recall measures how well positives are found and is calculated with the formula $Recall = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$ [9]. True positive is when prediction is as expected, and false negative is when the model fails to detect an object when it was there. The mAP is then the area under the precision-recall curve. An IoU threshold of 0.5 has been set for this model.

4.2.5.2 Layer 2– Lean Meat Extraction

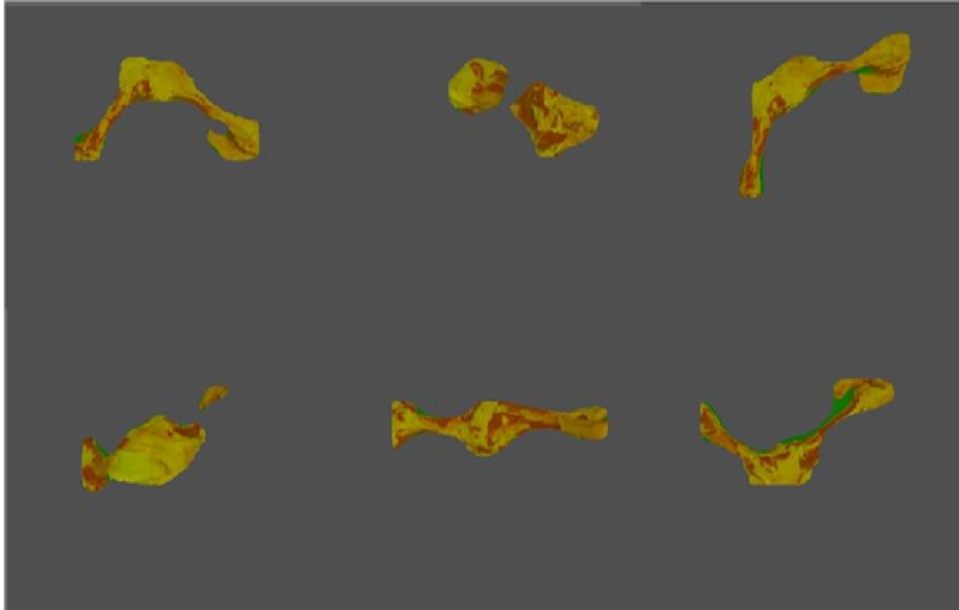


Figure 8 Layer 2 training images

Moving on to the second layer of training, the training dataset will be the results obtained from the first layer. Sample images in the training dataset of this layer have been shown in Figure 8. This image has been obtained by implementing a differencing function between the original image with the detected region to remove the background. The hue channel of the image has also been adjusted to allow for better detection results in the second layer.

All training flow and parameters were the same as the previous layer including the size of the training and validation set. The main difference in this layer is that only lean meat on the beef bones is annotated. It will still go through the same training process to produce a model. The model obtained after training has been completed will also be applied in the detection stage to extract lean meat from the beef bones.

4.2.6 Detection Stage

Random images in the testing set were then fed into this stage for detection purposes. There will also be two layers of detection involved with the second layer as the primary outcome of the project.

4.2.6.1 Layer 1 - Beef Parts Detection

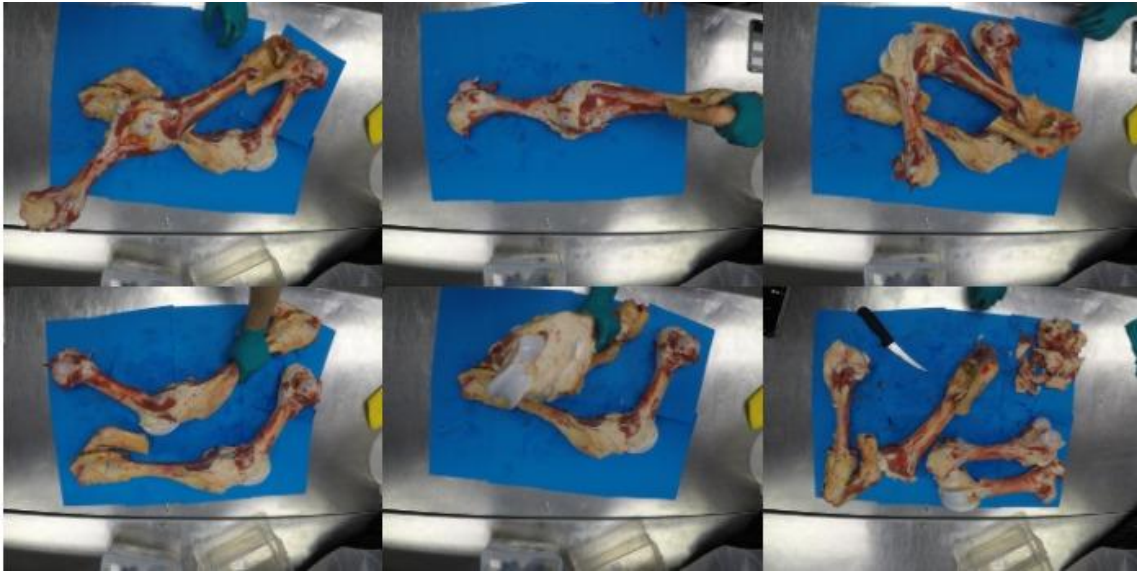


Figure 9 Layer 1 testing images

Once training has been completed, the model will be recreated in the inference model. Trained weights for the beef parts are then loaded and tested on testing images shown in Figure 9. A mask will be drawn over the detected region if beef parts have been detected. This will be illustrated in the project outcome section 5.0 below. A differencing function has been performed on masked images to create training datasets for the second layer.

4.2.6.2 Layer 2 – Lean Meat Detection

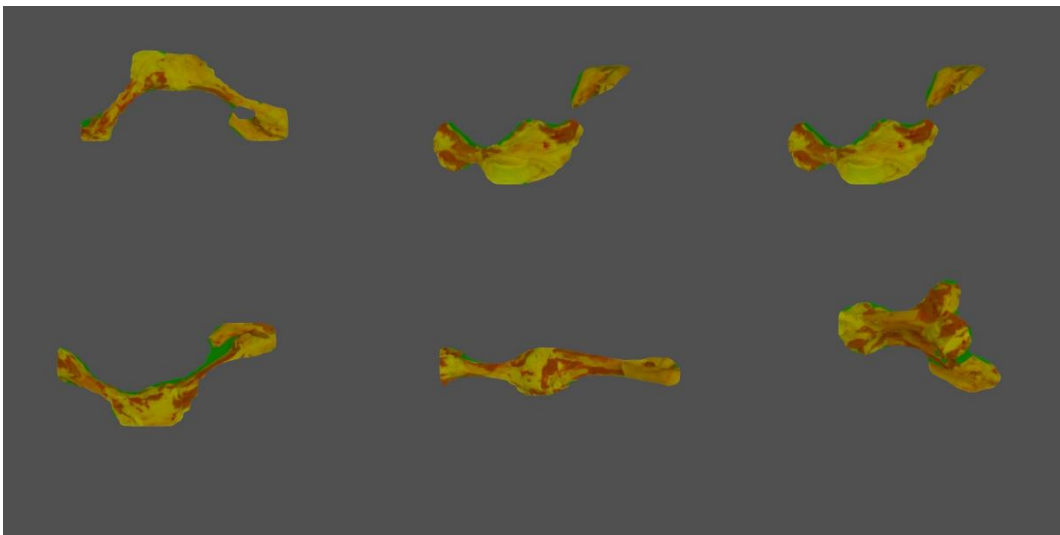


Figure 10 Layer 2 testing images

This layer follows closely the process flow in layer 1 from loading trained models to detection. The differencing function was performed on masked images shown in Figure 10 for the convenience of the performance estimation module in section 4.2.7.

Outcomes from this section will again be illustrated and discussed in the project outcome (section 5.0) below.

4.2.7 Performance Estimation Module

The KPI value shall be estimated with the weight of the beef parts and the bones, however, since this data is not available yet, an alternative estimation parameter – the surface area has been used instead. To express this mathematically,

$$KPI = \frac{\text{surface area of detected lean meat}}{\text{surface area of detected beef parts}}$$

The surface area of the detected object was obtained by comparing the detected regions to the original image. The KPI value will be expressed as a percentage to evaluate how well out of 100% did the butchers performed. This value provides us with information on the amount of lean meat left on the bone after the deboning process. Overlapped detections are only considered once in each frame therefore no assumptions are made for unobserved regions.

4.3 Single Board Computer (SBC) Execution

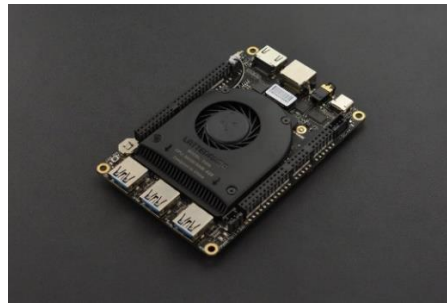


Figure 11 LattePanda Delta

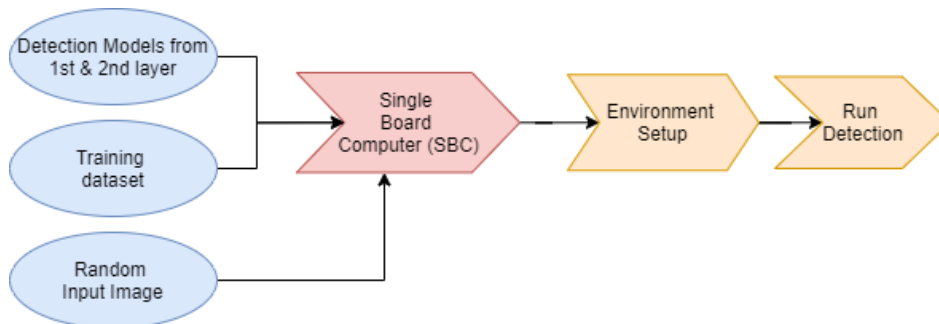


Figure 12 SBC Execution Flow

The SBC used in this project is a LattePanda Delta as shown in Figure 11 **Error! Reference source not found.** This device was chosen merely to simulate a lower computing capability device environment, it does not provide any indication of the device to be implemented in the processing plant. Based on Figure 12, the trained model for both stages, training dataset and random testing images have been loaded to the SBC to test out the duration needed to detect one image. The same environment has been set up in the SBC for optimal performance. The SBC plays a role to simulate an environment that will likely be set up in the meat processing plant. The meat processing plant will not require a powerful machine to run lean meat detection nor does it have space for it. Therefore, executing detection on an SBC saves cost and space, eventually producing an efficient lean meat detection system.

A longer detection time is expected on the SBC due to the low RAM space, however, further improvements shall be made if it takes an unacceptably long duration.

5.0 Project Outcomes

This section presents the project outcome obtained from both stages of the detection models and the final KPI estimation.

Once training has been completed, a trained model will be produced. This training model will then be implemented into the detection module for detection. Random images from the testing set will be used to test the functionality and validity of the trained model.

5.1 Beef Parts Detection

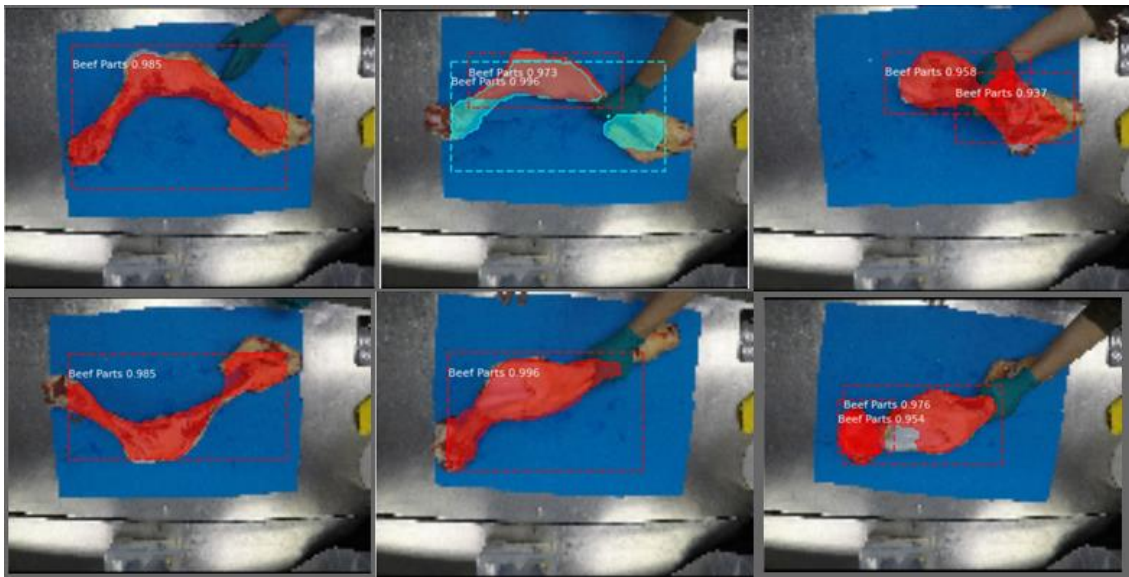


Figure 13 Layer 1 validation set detection outcomes

Figure 13 shows the detection outcome using images in the validation set. The coloured mask represents the area that has been detected and the dotted rectangular lines simply enclose the detected region. The numerical figures associated with the detected class represents the confidence level of detection. This value shows the probability that an anchor box contains an object and the accuracy of bounding boxes.

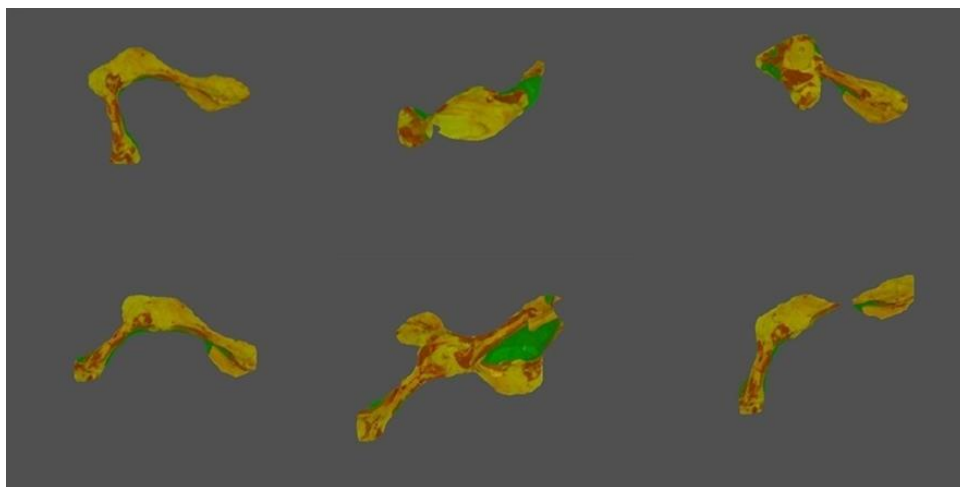


Figure 14 Layer 1 testing set detection outcomes

To evaluate the actual performance of the trained model, random images in the testing set were then passed through the detection algorithm. Figure 14 shows several detection outcomes of the first layer – beef parts. These images appear to be different from the raw image as several image processing operations have been applied to obtain images better suited in the next layer. A differencing operation between the original test image and the masked region has been performed to get the region below the mask. As differencing operation returns a relatively dark image with low contrast, the vuc channel of each image has then been increased by a constant amount to improve the image visibility.

The mAP value of this stage ranges between 0.7-0.8 with the maximum being 1. This is a relatively satisfactory detection outcome with approximately 70-80% precision. Due to the limited accessibility to testing sets, as higher variations are introduced into the testing sets, a lower mAP value is to be expected with the existing detection model.

5.2 Lean Meat Detection

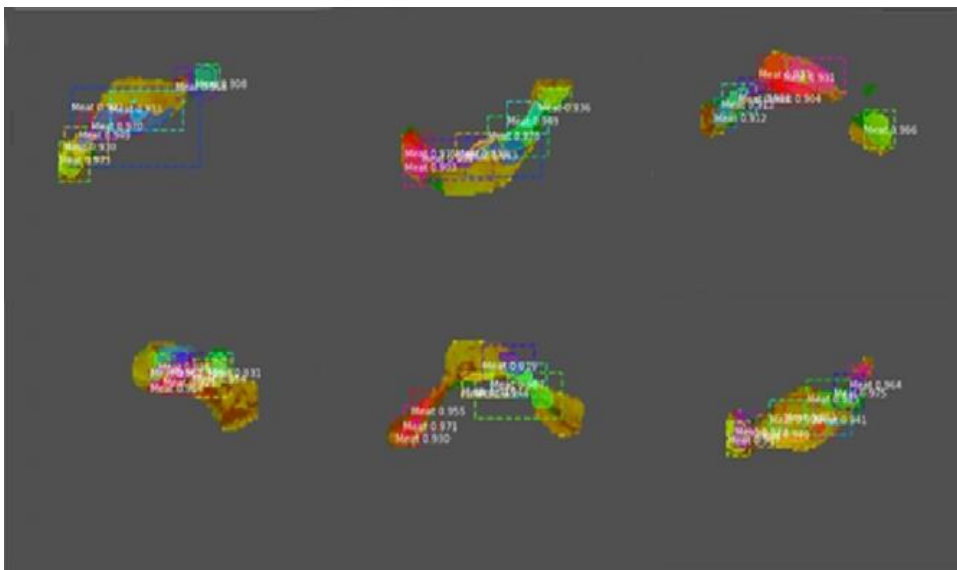


Figure 15 Layer 2 validation set detection outcomes

The second layer of the detection model involves the detection of lean meat. Figure 15 shows the detection outcome from random images found in the validation set. Coloured mask regions represent the detected lean meat while numerical values represent the confidence level. As observed from the outcomes, there is quite a large number of false positives in the image where lean meat was not detected when it was supposed to and non-lean meat was thought of as lean meat. This has resulted in a relatively low mAP value averaging at about 0.05.

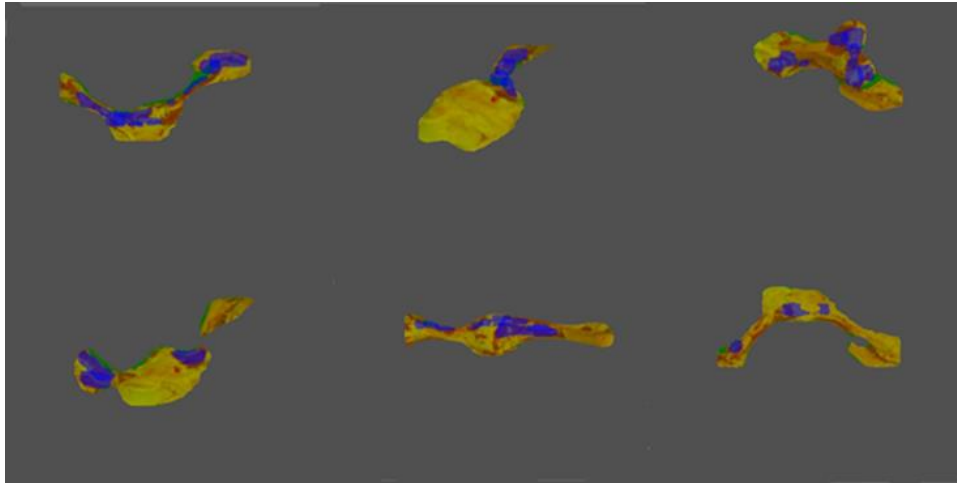


Figure 16 Layer 2 testing set detection outcomes

Figure 16 shows the detection outcome of the lean meat on random images in the testing set after applying beef parts extraction. This is part of the final desired outcome of this project. The blue masked region in the images represents the detected lean meat regions. As expected from the outcomes obtained from the validation set, the majority of the detected regions are false positives. The precision and accuracy of this model is relatively low and will certainly require a better training dataset and training parameters for the model to learn efficiently.

5.3 Performance Estimation

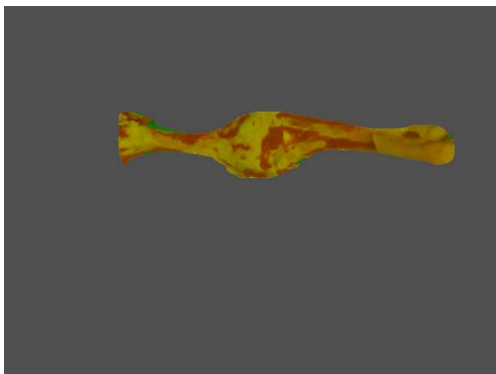


Figure 17 Beef Parts (KPI = 22.97%)

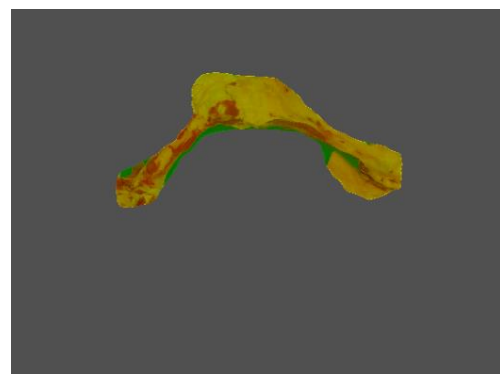


Figure 18 Beef Parts (KPI = 41.60%)

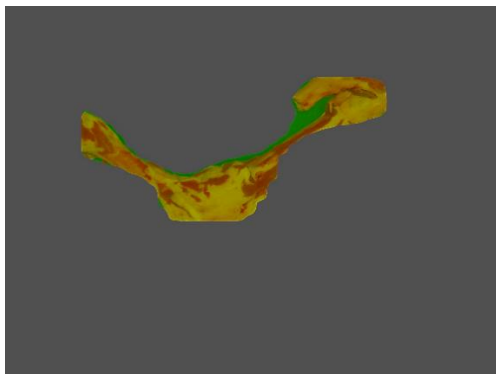


Figure 19 Beef Parts (KPI = 40.11%)

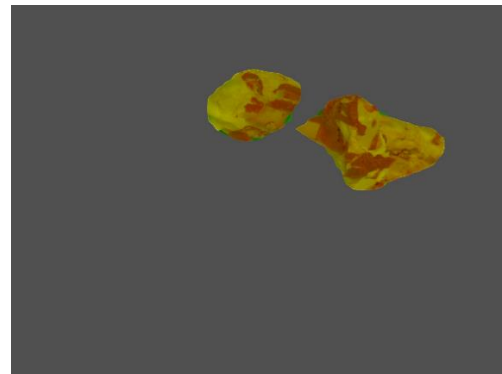


Figure 20 Beef Parts (KPI = 32.03%)

The performance estimation- KPI gives an impression of the amount of meat left on the surface of the beef parts after the deboning process. This estimation acts as a measure of the individual performance of the butchers as well as provide an idea of the amount of wastage from a meat processing plant. With this estimation, meat processing plant managers will evaluate whether it is worth investing in developing new deboning methods to reduce plant wastage.

Figure 17-20 shows several beef parts images with their associated KPI values. By raw observation these estimations seem to be reasonable, however, no further conclusion could be drawn with the limited amount of data available.

5.4 SBC Execution

The SBC was able to produce the same result as the high-performance workstation. However, it is taking approximately 4.5 minutes for the entire process which only involves detecting one image. Imagine having hundreds and thousands of images be processed daily, this speed is not acceptable. Detection should be done almost instantaneously to cater for the high volumes in a meat processing plant. Therefore, methods to improve the detection speed on the SBC shall be explored further before placing this system in the processing plant.

6.0 Discussion

6.1 Beef Parts Detection

Based on the detected regions shown in Figure 14, the trained model appears to be relatively compliant with one of the objectives whereby the majority of the beef parts are being detected and extracted from the raw image. However, there exists a factor – low dataset variability which could potentially affect the reliability of our judgement on the model. This was because, under such conditions, there is no other way than lowering the evaluation benchmark which could potentially compromise the performance of our model. Therefore, while moving into the next stage of the project, not only should we increase the size of the dataset, but we should also introduce a higher variability into it to ensure a reliable evaluation outcome is obtained.

6.2 Lean Meat Detection

As observed from Figure 16, the trained model for lean meat appears to be relatively inaccurate where most of the lean meat regions were not being detected by the model. This could be due to the largely inconsistent characteristics of lean meat which makes it highly challenging to train a model. Subsequently, insufficient training dataset has also made the training process even more taxing as the training algorithm was unable to train itself in identifying such detailed and complex structure with limited information.

6.3 Performance Estimation

While the second layer of detection may not be performing as good as desired, the overall performance estimation – KPI value that was obtained seems to be a reasonable estimation by raw observation. However, further steps are to be taken to appropriately evaluate the reliability of these estimations. One of the steps is to obtain an entirely clean bone – meaning with as little meat as possible on it as a ground truth value to measure the accuracy of the estimation. Also, this estimation value could be compared with the value obtained by the existing technology for verification purposes.

7.0 Conclusions / Recommendations

This is just the initial phase of the project, presenting a conceptual design of using computer vision to detect lean meat on bones. There will still be a series of steps to be taken in the future to improve the project in various aspects.

For the next stage, we propose to build a prototype device that will be installed on the bone belt and assess the quality of deboning processes. In general, the system will,

- ◆ classify different parts going through the bone belt,
- ◆ detect lean meat on the parts,
- ◆ evaluate deboning processes based on the lean meat left, and
- ◆ grade the performance of different deboning processes for further evaluation or improvement.

The following objectives will be achieved in the next stage,

- ◆ demonstrate an example system that can grade the performance of the deboning process,
- ◆ demonstrate the feasibility of using computer vision in meat processing,
- ◆ develop a prototype device to be deployed to the processing site,
- ◆ draft a plan for on-site evaluation, and
- ◆ draft a plan lead to future adoption.

The next stage of the project will cost \$138,500, including \$118,500 professional fee and \$20,000 operating cost, and it will take seven months to complete the next stage development. In order to achieve the proposed objectives, the following milestones will be accomplished.

- ◆ Deposit on signing the contract to enable:
 - System design
 - Development plan design
 - Consultation with AMPC
- ◆ Site visit for data collection, classification and labelling
- ◆ Detection models development
- ◆ Prototype device development
- ◆ Testing and evaluation
- ◆ Demonstration to AMPC, submit Final report and SnapShots and submit a plan for the next stage

8.0 Bibliography

- [1] Shahzad, G. Köhler, M. Knapp, E. Gaubitzer, M. Puchinger and M. Edetsberger, "Emerging applications of fluorescence spectroscopy in medical microbiology field", *Journal of Translational Medicine*, vol. 7, no. 1, 2009. Available: 10.1186/1479-5876-7-99.
- [2] "Animal Monitoring with Computer Vision - Case Study," *viso.ai*, Jan. 21, 2021. <https://viso.ai/applications/animal-monitoring/> (accessed Mar. 11, 2022).
- [3] Brownlee, J., 2022. *What is the Difference Between Test and Validation Datasets?*. [online] Machine Learning Mastery. Available at: <<https://machinelearningmastery.com/difference-test-validation-datasets/>> [Accessed 12 March 2022].
- [4] E. Odemakinde, "Mask R-CNN: A Beginner's Guide", *viso.ai*.
- [5] I. El Naqa and M. J. Murphy, *What is Machine Learning*. Springer International Publishing Switzerland, 2015, pp. 3-11.
- [6] "Inline Fat Analysis Machines | Eagle Product Inspection", *Eagle Product Inspection*, 2022. [Online]. Available: <https://www.eaglepi.com/au/solutions/fat-analysis-machines/>. [Accessed: 09- Feb- 2022].
- [7] K. Nedenskov Jensen and P. Waaben Hansen, "Rapid methods for fat analysis in the meat industry", FOSS, Hilleroed, 2022.
- [8] "Project InnerEye - Democratizing Medical Imaging AI - Microsoft Research", *Microsoft Research*, 2022. [Online]. Available: <https://www.microsoft.com/en-us/research/project/medical-image-analysis/>. [Accessed: 11- Feb- 2022]
- [9] S. Yohanandan, "mAP (mean Average Precision) might confuse you!," *Medium*, Jun. 09, 2020. <https://towardsdatascience.com/map-mean-average-precision-might-confuse-you> [accessed Feb. 15, 2022].
- [10] T. Woolford, "Getting the facts straight with accurate fat analysis", *Food Processing*, 2012. [Online]. Available: <https://www.foodprocessing.com.au/content/processing/article/getting-the-facts-straight-with-accurate-fat-analysis-924200066>. [Accessed: 09- Feb- 2022].
- [11] "What is Computer Vision? | IBM", *ibm.com*, 2022. [Online]. Available: <https://www.ibm.com/au-en/topics/computer-vision>. [Accessed: 11- Feb- 2022].
- [12] Matterport, "Matterport/MASK_RCNN: Mask R-CNN for object detection and instance segmentation on Keras and tensorflow," *GitHub*. [Online]. Available: https://github.com/matterport/Mask_RCNN. [Accessed: 04-Mar-2022].

9.0 Appendices

This appendix shows the relevant commands and links needed to setup the training environment.

1. **Creating a conda virtual environment**

```
Conda create -n myenv python=3.6 pip
```

```
Activate myenv
```

2. **Command to launch Jupyter notebook from the virtual environment**

```
python -m ipykernel install -user -name=myenv
```

3. **NVIDIA CUDA paths to add to environment variables of PC**

- C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2
- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2
- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2

4. <https://universityofadelaide.box.com/s/dbu9y4m7v7i8kaiazbowfzdhc01z7dkf> contains

- Training sets
- Testing sets
- Relevant files needed to run training and detection
- Code for training and detection

5. **Guide to installing CUDA and CuDNN**

<https://towardsdatascience.com/installing-tensorflow-with-cuda-cudnn-and-gpu-support-on-windows-10-60693e46e781>