AMPC
AUSTRALIAN MEAT PROCESSOR CORPORATION

Final Report

Identification of Bagged Racks Using Machine Learning Algorithms

Project Code
2021-1275

Prepared by
MEQ Probe

Date Submitted
12/04/2022

Published by
AMPC

Date Published
12/04/22

Contents

1.0 Executive Summary

1.1 Project objectives

This project concerns the development of algorithms, software, and machine learning models to identify bagged lamb rack primals on an abattoir conveyor belt at chain speed. The automated identification of these primals has the potential to offer significant advantages to processors in terms of product reliability and reduced labor costs, and would additionally represent a strong foundation on which the development of related models and systems could be based for other applications relevant to meat processors.

The scope of the research includes targeted data collection, data labeling, machine learning model development, algorithm design and development, and system evaluation and testing, and was conducted in cooperation with Gundagai Meat Processors (GMP). The specific objectives addressed by this project are:

- To produce a labeled dataset consisting of 30,000 images of bagged lamb racks.
- To develop a machine learning model that can identify a bagged lamb rack on the GMP boning belt with 99.9% accuracy at chain speed.
- To develop a machine learning system that can alert a nearby user if the model is unsure, allowing for manual override.

1.2 Data collection and labeling

To develop a machine learning model to recognize objects, it is necessary to capture raw data (in this case: images of primals) and then label those data with relevant information (in this case: the locations and types of primals in the images) to build a high-quality dataset which can be used for model training, analysis, and evaluation. Data collection was carried out in two phases:

1. In the initial phase, data were collected manually by human operators using both fixed and handheld cameras.
2. In the second phase, with an initial model, software, and fixed camera hardware in place, automated heuristics were used to identify and record relevant data based on identified weaknesses in the in-development model, as part of an active learning regime.

In both phases, the collected raw data were then subject to a four-part labeling process:

1. The in-development machine learning model was used to make provisional object identifications in the images.
2. Using a specialized computer vision annotation platform, human annotators familiar with lamb primals corrected the model’s output.
3. These human-corrected labels were reviewed by a highly experienced member of the team to ensure the accuracy of the object identifications.
4. Finally, some automated processes were carried out on the reviewed labels to render the data suitable for use in training the machine learning model.

1.3 Machine learning model development

The machine learning model architecture chosen for the object identification task was YOLOv4, which is both highly accurate at object detection, and optimized to allow for real-time predictions. The process of model development involved data collection and labeling as described above, data augmentation to increase the effectiveness of training, and hyperparameter optimization for both accuracy and speed of prediction.

1.4 Model output post-processing

Aside from the development of the core machine learning model which can recognize primals within an image, the overall system also incorporates several algorithmic post-processing steps which are able to take advantage of the ability of the system to capture multiple images of a given primal across time and from different angles across several cameras to further increase the overall accuracy of primal identification. The primary post-processing algorithms applied to the raw machine learning predictions are as follows:

1. An object tracking algorithm to track individual primals through time as they cross each camera's field of view.
2. A cross-camera matching algorithm to match detections of the same primal from different angles across multiple cameras at a given moment in time.
3. An overall classification algorithm that uses the outputs of the machine learning model along with the temporal tracking and cross-camera matching information produced by the above algorithms to produce highly robust object classifications which do not depend solely on any single captured image.
4. An object counting algorithm that uses the output of the overall classification algorithm to count the number of each type of primal passing the camera station.

Although the counting of primals represents an additional degree of complexity beyond the formal scope of the project, a counting algorithm was included in the system both because of its utility in system evaluation and as a demonstration of a potential application of the technologies and methods developed and employed in the project.

1.5 System validation and evaluation

The validation method consisted of a human expert in primal identification monitoring the system during operation in the plant in real-time. The evaluations were based on the counts of primals generated by the system in comparison to the counts of primals generated by the expert.

1.6 Results and limitations

For bagged rack primals, the system was evaluated against captured data containing over 1070 individual bagged racks captured over a week of operation, and, the system was shown to have a 100% rate of accuracy in counting the racks in that dataset.

1.7 Findings, recommendations, and conclusions

The success of this project provides strong evidence for the applicability of computer vision and machine learning techniques in the abattoir boning room, and we would recommend that further research and development be undertaken to bring these technologies to bear in further applications which have a direct and specific benefit to processors. In our opinion, these technologies could have immediate value in the following areas:

[REDACTED]

Based on the results of this project, we conclude that machine learning and computer vision technologies can be employed in an abattoir environment to detect bagged lamb rack primals with a high degree of accuracy, and further that these methods show strong promise in terms of applicability to boning room automation, analysis, and quality assurance tasks generally, with the potential to increase the reliability of abattoir processes and reduce labor costs.

2.0 Introduction

In the red meat industry, a consistent product is key. The problem is that there is a multitude of end-point products that can be delivered by a processor. Output from a single plant can be complex and diverse, with product specifications changing day-to-day to meet customer requirements. There is a certain level of skill required in identifying a piece of meat correctly and efficiently on the production line, and ensuring the product is meeting specifications, biosecurity, and quality control measures. As the number of potential options for an object increases, so do the potential errors in classification. This is an issue in the red meat industry, as identification tasks such as packaging products and final checks for boxed meat are often delegated to unskilled labour, with the least experience in the red meat industry. This lack of skill and high rate of labour turnover often results in a high risk of human error occurring, disrupting workflow and having the potential to create quality and biosecurity issues for the processor. If products do not meet specifications, biosecurity requirements, or are incorrectly packaged, there is the potential for reputational damage both domestically and internationally. If the wrong product is in the wrong box, the costs can be very high.

This project aimed to deliver a machine learning based computer vision system specifically designed to identify bagged racks in Australian Abattoir boning rooms, helping to ensure the right product ends up in the right box. This development positively impacts the red meat industry because it proves that artificial intelligence algorithms can be used to successfully identify primals with extremely high accuracy, which reduces the risk of having incorrectly packed products. In addition, the machine learning model built at the GMP abattoir is solid enough that it can then act as a baseline for future models. This is because the model was trained with more than 30,000 images collected during different days and times from a variety of camera angles, which means that adopting this model for other similar processing plants might only require a short development iteration.

A variety of data collection, data labeling, software development, algorithm design, statistical, and machine learning techniques were applied and tested to successfully identify specific cuts of meat. This artificial intelligence system has the capabilities and flexibility to be applied across a wide variety of use cases throughout processing plants in the red meat industry. The ability to capture images that are processed and analysed at chain speed will enable output to be used in real-time decision-making. Unskilled labour requirements will be reduced, and a greater level of quality assurance and biosecurity guarantee will be achieved consistently and efficiently across the variety of end-point products manufactured by processors.

3.0 Project Objectives

The objectives of this project are outlined as follows:

- To produce a labeled dataset consisting of 30,000 images of bagged lamb racks.
- To develop a machine learning model that can identify a bagged lamb rack on the GMP boning belt with 99.9% accuracy at chain speed.
- To develop a machine learning system that can alert a nearby user if the model is unsure, allowing for manual override.

4.0 Methodology

3.1 Data collection

3.1.1 Initial data collection

Using a modular camera system prototype, images were collected in the plant over a series of days. This system undertook initial testing with a simple software application and one **[REDACTED]** camera for the capture of images at two predetermined locations in the abattoir. After assessing several viewpoints, camera settings, space for the mounted system, belt position, focal length, exposure, and lighting conditions, one location on the boning belt was deemed suitable for the collection of bagged lamb racks. These images were collected over a series of days in the plant at the predetermined location on the boning belt, which allowed for different angles and types of bagged racks to be captured. Both a handheld camera and a fixed camera system were used for the collection of images. See *Appendix 1* for a sample of the initial images collected in the abattoir.

3.1.2 Bulk data collection

Using a more advanced version of the modular camera system that included a total of six cameras with different lens types and angles, the rest of the images were captured simultaneously from fixed mounting points over a series of months. See *Appendix 2* for a sample of the new types of frames the new system was able to capture.

In parallel to the bulk data collection step, images were uploaded to an interactive labeling platform and annotated with rectangular bounding boxes that locate and identify primals. These were then submitted for quality review to an experienced member of the team to ensure maximum quality and consistency. Each primal dataset was converted to a CSV file that contains information about the images captured including the objects located in the frame and of its respective coordinates.

3.2 Data labeling

Collected images were manually labeled for primal identification using bounding boxes. These are rectangular boxes used to define the location of the target object. Bounding boxes are determined by the x and y axis coordinates in the upper-left corner and the x and y axis coordinates in the lower-right corner of the rectangle. They are represented by two coordinates (xmin, ymin) and (xmax, ymax). One frame may contain multiple images of primals, referred to as objects in AI terms. Each object has its object-class and bounding box ((xmin, ymin) and (xmax, ymax)), creating a filename that contains multiple, and sometimes overlapping bounding boxes.

A Computer Vision tool was used to annotate images, as it provides an interactive user interface and allows easy tracking and reviewing of bounding box annotations. See *Appendix 3* for sample images with bounding box annotations created in the Computer Vision tool.

Once all the collected images were labeled, a quality review process was initiated. All labels were reviewed and corrected by a single member of the team with expertise in identifying differences between primals. This ensures the dataset is high quality and consistent. All labels were then combined into a final CSV file that contains one row per object identified. Each row has one value for the following columns:

a) Filename. E.g: gmp_20210930_202107.png

b) Frame width and height. Eg: 768 * 768

c) Object: E.g: 'bagged_rack'

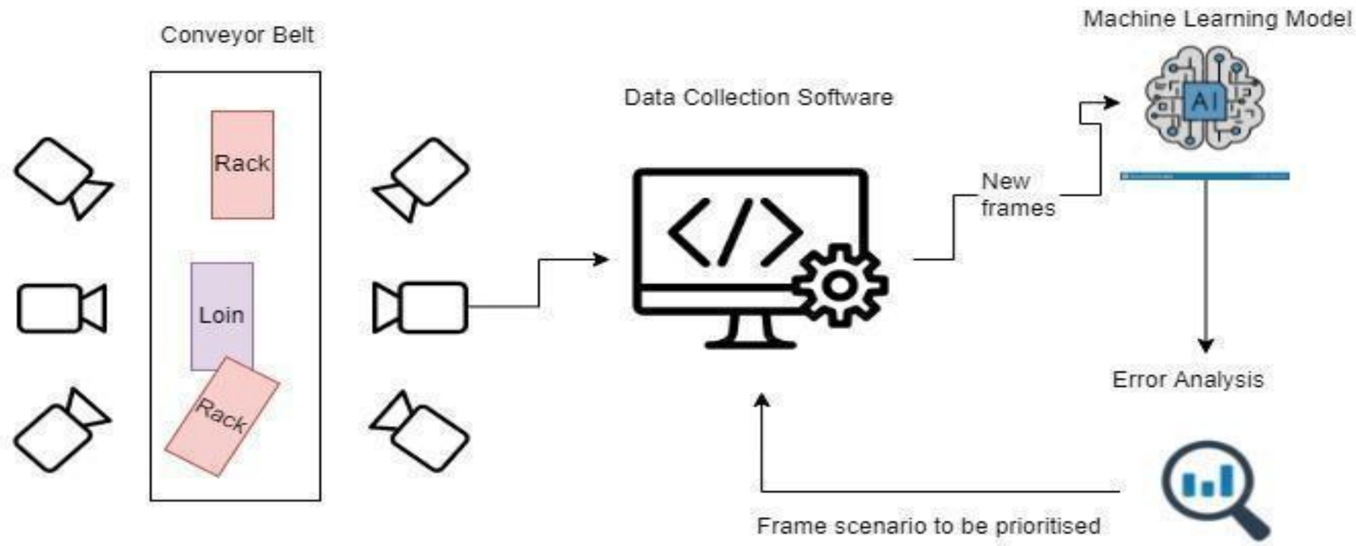
d) Bounding box coordinates as xmin, ymin, xmax, ymax. E.g: 418.35, 131.13, 570.88, 216.76

We are providing the CSV file for bagged lamb racks (bagged_racks_final.csv). The spreadsheet contains a list of 30,000 objects and 30,000 labels. See *Appendix 5* for a sample of the CSV.

3.3 Improving data collection using active learning

Active learning (AL) is a subfield of machine learning that focuses on the study of data sets to obtain performance gains by actively selecting the most useful samples to be labeled. In contrast to active learning, passive learning refers to when all data is given at once to the annotator (Ren *et al.*, 2021). AL has important applications in the machine learning community as it is proven that proper selection of samples for training the model can reduce the probability of mistakenly predicting the response variable for an unknown explanatory variable (Hino 2020).

Although this was not originally planned for the project, the active learning technique was implemented by continuously analysing the model's top errors and creating potential error detection mechanisms for the camera system. This technique allowed us to achieve the project's objectives ahead of schedule because it allowed us to maximise performance with limited human intervention. Data was captured by the camera system, fed into the machine learning model for predictions, and then analysed for error detection. The output of the analysis would suggest a couple of the scenarios where we thought the model was struggling that then would be visualised by a member of the field team and confirmed as being model errors. The camera software system would then be updated t



o capture data in real-time that could simulate a similar scenario.

Figure 1: Active learning cycle implemented for this project.

Figure 1 shows how data was collected in real-time, fed into the machine learning model for predictions, and then to error analysis. One or more frame scenarios were selected as a priority for data that replicated a similar scenario to be collected in the coming weeks.

For example, after performing the first data analysis on the model's errors it became clear that for some scenarios the model detected an object more than once as being in more than one class in a given image. In order to update the model to avoid this type of error, it was necessary to obtain many samples of images for which the model makes such double-detections, annotate them with the correct classifications, and then train a new model version with the resulting image-annotation pairs. To achieve this, an automated heuristic was implemented to identify frames where two bounding boxes are closely aligned with each other (within a small tolerance), and then these frames were stored and prioritized for human annotation.

While not all images gathered using heuristic methods necessarily produce the targeted type of error, careful design of the heuristic algorithms used to identify the images to store and prioritize allowed the automated production of sets of images with a high concentration of error-producing samples, and so by using this automated method of prioritization, we were able to rapidly and automatically target various identified error types for correction, allowing the model to be improved much more rapidly than if the images to be annotated were selected manually.

Table 1 describes the various types of errors we identified during model development, along with a description of the heuristic algorithms used to identify, store and prioritize images for annotation to target each error type.

Table 1: Types of errors identified during model development and heuristics used to prioritize images for annotation to target each error type.

[REDACTED]

3.4 Data cleaning and processing

After the successful collection of images of bagged lamb racks and the production of prioritised labels in the form of bounding boxes, a final dataset for the primal was created. Prior to feeding that dataset into an initial Machine Learning (ML) algorithm, data cleaning and preprocessing was required to remove corrupt and irrelevant records, and to transform data into an appropriate ML format. The steps for data cleaning and processing that were applied programmatically using Python were:

1. Adding empty labels for images where there is no primal.
2. Enforcing consistency in labeling names.
3. Clipping bounding boxes that fall outside the edges of the image
4. Normalizing coordinates to allow easier re-scaling
5. Resizing images and bounding boxes to fit the model architecture.

This process ensured only the highest quality data was used to train the models.

3.5 Machine learning model development:

3.5.1 Model selection

The Computer Vision task required for this project is referred to as Object Detection in Artificial Intelligence terms. The developed model needs to be able to locate the presence of objects with a bounding box and predict the classes (naked rack, naked loin, and bagged rack) of the located objects in an image. Amongst the various models available to solve problems like this, You Only Look Once (YOLO) was selected as the preferred base architecture as it proves to be predicting faster than Region-Based Convolutional Neural Networks models like R-CNN, and more accurately than other single-shot models like Single Shot Detector (SSD) (Kim *et al.*, 2020). Multiple versions of YOLO architectures have been released, but the fourth version (YOLOv4) was specifically designed to produce extremely accurate results in real-time, which is why it was chosen for this project (Bochkovskiy *et al.*, 2020).

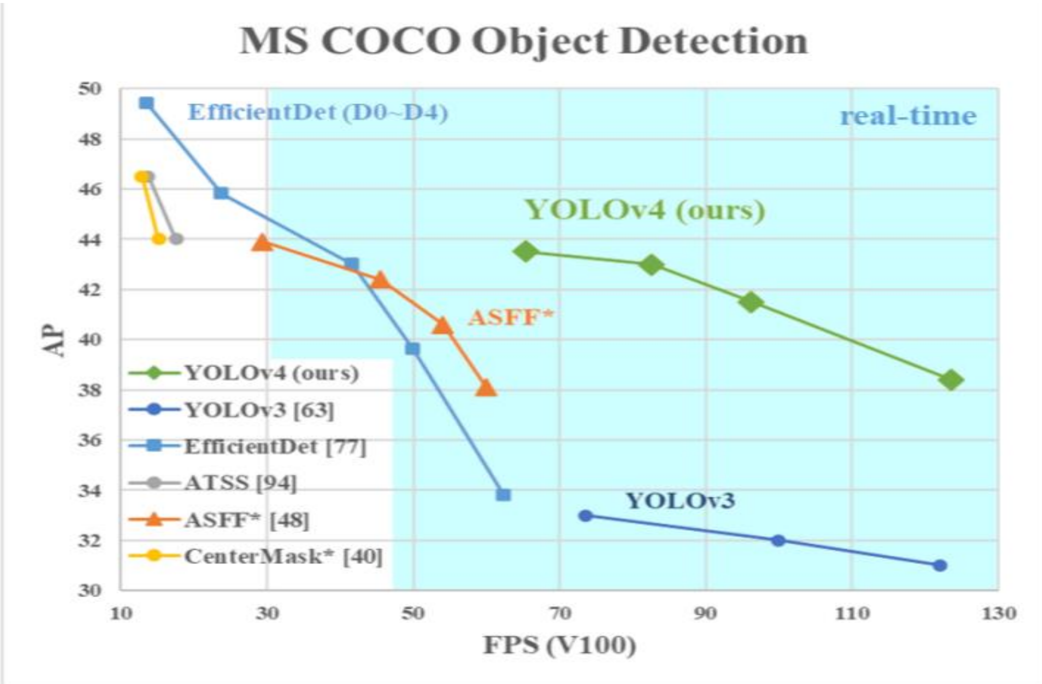
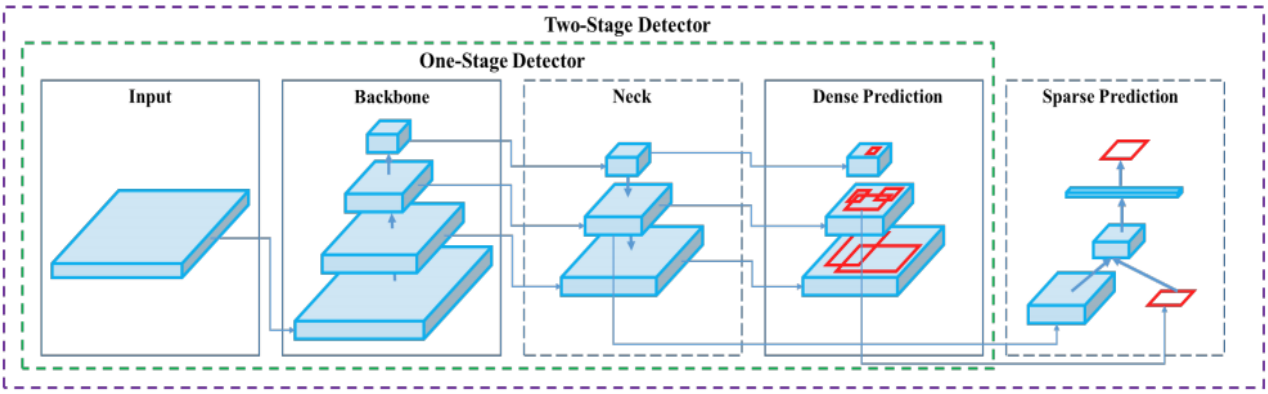


Figure 2: Comparison of the proposed v4 and other state-of-the-art object detectors. YOLOv4 runs twice faster than EfficientDet with comparable performance. Improves YOLOv3's AP and FPS by 10% and 12%, respectively (Bochkovskiy et al., 2020).



Input: { Image, Patches, Image Pyramid, … }

Backbone: { VGG16 [68], ResNet-50 [26], ResNeXt-101 [86], Darknet53 [63], … }

Neck: { FPN [44], PANet [49], Bi-FPN [77], … }

Head:

Dense Prediction: { RPN [64], YOLO [61, 62, 63], SSD [50], RetinaNet [45], FCOS [78], … }

Sparse Prediction: { Faster R-CNN [64], R-FCN [9], … }

Figure 3: YOLOv4's architecture for object detection (Bochkovskiy *et al.*, 2020).

For this project, YOLOv4's architecture had to be modified to fit the requirements and trained from scratch on samples from the dataset delivered for the previous milestone.

3.5.2 Parameter configuration and hyperparameter tuning

Hyperparameters are those parameters whose value controls the learning process of an algorithm. In Machine learning, optimization or tuning is the process of selecting the best combination of hyperparameters that minimizes the predefined loss function on a validation dataset. The hyperparameters that achieved the best performance for the final model are the following ones:

Table 2: Hyperparameters chosen to train the final model.

[REDACTED]

Below there's a summary of what each parameter represents (Redmon, 2016):

- The subdivision parameter controls how many samples will fit into RAM (minibatch/subdivision = samples loaded simultaneously per pass).
- Momentum refers to accumulation of movement, and it controls how much the history affects the further change of weights (optimizer).
- The decay parameter is a weaker updating of the weights for typical features, it eliminates dysbalance in the dataset (optimizer).
- The learning rate is the one used initially.
- Burn_in is the number of iterations that the initial burn_in that will be processed for.
- Max_batches is the number of iterations that the training will be processed for.
- Policy is the method for changing learning rate: constant (by default).
- Steps are the numbers of iterations at which the learning rate will be multiplied by the scale factor.
- Scales is the factor used to multiply learning rate. if policy=steps - f.e. if steps=8000,9000,12000, scales=.1,.1,.1 and the current iteration number is 10000 then current_learning_rate = learning_rate * scales[0] * scales[1] = 0.001 * 0.1 * 0.1 = 0.00001.

3.5.3 Data augmentation

Data augmentation is a common technique used in Machine Learning to increase the size and quality of samples by adding slightly modified copies of already existing data. This approach can help reduce overfitting, when a network learns to model the training data with a high variance, which can fail to fit future observations reliably. Augmentation procedures that were applied to this project include image rotation, shifting, cropping, resizing, blurring,scaling, and modifying saturation levels, exposure, and hue. The image augmentation algorithms that were used in this project to reduce overfitting had to be applied to both the images and its respective bounding boxes, which added another layer of complexity to the task as linear algebra had to be used to ensure bounding boxes were correctly modified. See *Appendix 4* for an example of before and after random cropping transformations used for data augmentation. After successful application of multiple data augmentation techniques, an improvement of up to 6pp was recorded for the model's accuracy.

3.5.4 Model size vs inference speed trade-off

Although developing a model that can achieve 99.9% accuracy is crucial to the success of this project, keeping up with the speed of the chain is as important as the model's quality. Building a very accurate system that is too slow to predict in real-time might not be of value to the plant. For that reason, we performed an experimental analysis to select the ideal model size to maximize the model's accuracy without sacrificing speed. The goal of this analysis was to understand the impact that model size has on algorithm performance. Model version 12 (v12) was trained on an image size of **[REDACTED]** while model version 13(v13) was trained on **[REDACTED]**. Both models had the same train/eval split to ensure the mAP comparison is fair. The results of the experiment can be seen below. Please note that mAP refers to the mean Average Precision (mAP) for all classes of primals on a frame by frame basis, and IoU refers to the Intersection Over Union that describes the extent of overlap between the predicted and the real bounding box. The greater the region of overlap, the greater the IOU. An IoU value of 1 indicates perfect overlap, whereas a 0 indicates no overlap.

Table 3: Results of an experiment to compare object detection performance between two different model sizes.

[REDACTED]

Regarding model performance by size, it is clear that a bigger size model is more accurate, as it is capable of providing improvements in Precision and IoU, and significantly reducing the number of false negatives. On the other hand, the gains in the model's quality are not directly proportional to the losses in speed. Speed was measured in frames per second (FPS), and as *Figure 4* shows, a smaller image size allows the model to predict more frames per second. To find a balance between speed and accuracy, the final model size selected for this project was 608*608*3.

[REDACTED]

Figure 4: Comparison of ML inference processing frame rate with six cameras vs size of model input image.

3.6 Model output post-processing

This section describes the various types of processing that the system uses to enrich and contextualize the output of the core machine learning model to improve overall accuracy and produce additional useful results.

3.6.1 Cross-camera object matching

[REDACTED]

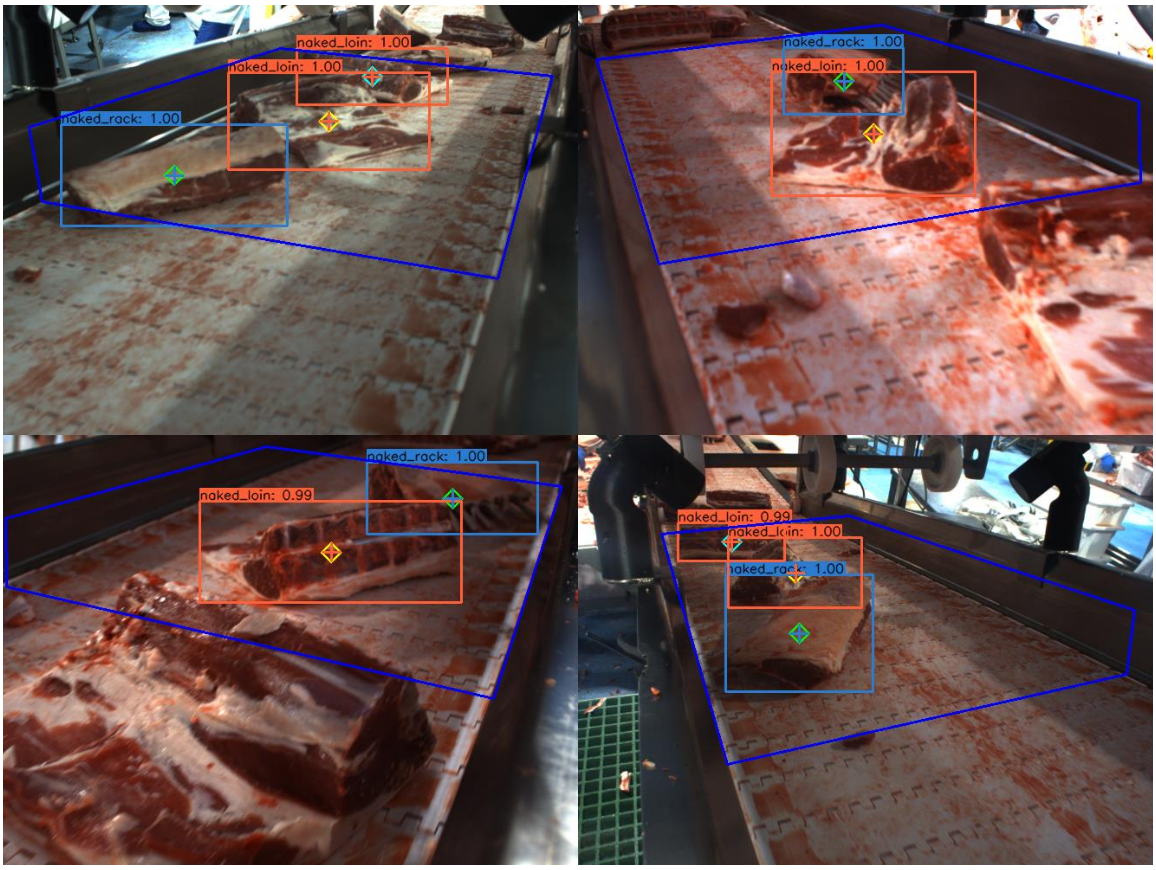


Figure 6: Illustration of cross-frame object matching. The colored diamonds surrounding the center of each object indicate which objects in other cameras it has been matched with: objects with green diamonds are matched with other objects with green diamonds, and similarly for yellow and cyan. Note that objects whose centers are outside the blue region in each frame are not processed by the ML model, and that two of the cameras are pointing in the opposite direction as the other two.

3.6.2 Object tracker

The machine learning algorithm developed for this project can predict a list of bounding boxes from an input image in real-time, but it can't relate one frame to another through time. Although the object tracker was not a project objective, it is clear that the addition of a tracker that could combine predictions from all frames and find associations between objects through time brings a lot of value.

An object tracker was developed to find associations between bagged racks' bounding boxes through time. The goal of this tracker is to identify the path that bagged racks follow through the conveyor belt. The tracker was based on the **[REDACTED].** By combining the input of the YOLOv4 model with these two algorithms and adapting some parameters to fit this specific task, we're able to accurately identify the track that bagged racks follow through the conveyor belt in real-time. This then enables us to count the total number of objects that go through the conveyor belt.

3.6.3 Combined overall classification

While the machine-learning model itself is able to produce highly-accurate classifications of objects using only a single image from a single camera, the accuracy of these classifications can be further improved using the outputs of the object tracking and cross-camera object matching algorithms by combining the model's predicted classifications of the same object across multiple images across time and from multiple angles across different cameras. In this way, a classification error that occurs for a single image or even several images can be corrected by reference to other images of the same object, because the model's single-image classification is correct in the vast majority of cases.

For a given object detection in a given camera, the overall classification of the object using both intertemporal information from the object tracking system and cross-camera matching information from the object matching system is obtained as follows:

[REDACTED]

3.6.4 Counting mechanism

To demonstrate and evaluate the capabilities of the camera system, in addition to producing classifications of objects appearing on the belt, the software was also equipped with the option to attempt to count the number of objects of each class passing by the camera station as the system operates. This adds another layer of complexity to the processing of the model outputs, as it not only requires that objects be correctly identified but also requires that each object is counted exactly once. The counting system comprises an algorithm that attempts to count objects seen by each camera individually, and a second algorithm that processes those single-camera counts to produce an overall count which attempts to correct for situations in which an object is not detected in a particular camera either due to occlusion by another object or due to the angle at which the object is seen.

[REDACTED]

4.0 Results and limitations

4.1 Validation method

The proposed process for validating the system was an in-plant evaluation that consisted of a human expert counting primals in real-time as they passed a certain point on the belt over a one-week period. This validation method was recommended to ensure that the model performs as accurately as reported. This method was not possible to execute for other primals(naked loins and naked racks) due to the high number of primals present at the same time, but it was feasible to utilise for bagged racks. This is because the conveyor belt used to transport bagged racks in the plant carries a smaller number of primals simultaneously, which makes the task of evaluating the system in real-time possible for a human.

An individual expert in primal identification was then able to review a total of 1070 objects over a week period to determine if the number of bagged racks identified by the system was correct. This allowed the team to effectively allocate time and resources to ensure validation was conducted with a high level of accuracy.

4.2 Validation results and limitations

The human expert in primal identification was evaluating real-time runs. There was no difference between the expert human count of 1070, with the model counting 1070 bagged racks passing a certain point on the boning belt. This indicates that, as far as these specific simulation results are concerned, our proposed system is 100% accurate at identifying the number of bagged racks that passed a certain point on the boning belt.

The model has only one limitation in regards to bagged rack identification; green opaque bags that look exactly the same as bags used for other similarly-shaped primals. Figure 9 shows examples of primals bagged in green opaque material where, unless the bags face the translucent side, it’s almost impossible to distinguish one from the other.

Figure 9: Captured frames showing different similarly-shaped primals bagged in opaque green bags.Only the bottom right example shows a translucent bag where it's possible to distinguish the primal from a bagged rack.

The model performance is also limited to the types of bagged racks trained on. If a new type of bagged rack with different characteristics appears, it might not be recognised with high confidence. The solution for this limitation is to gather more samples of this new type and re-train the model.

5.0 Project Outcomes

5.1 Labelled dataset consisting of 30,000 images of bagged lamb racks

For six months, 30,000 images of bagged lamb racks were captured in an abattoir setting. A wide variation of rack images was captured by a handheld camera and a mounted system of six cameras. These images were reviewed, and annotated with bounding boxes, generating a dataset of 30,000 labeled bagged racks. This dataset was formatted to act as an input for a machine learning model. **[REDACTED]**

5.2 Machine learning model that is able to identify a bagged lamb rack on the GMP boning belt with 99.9% accuracy at chain speed

The proposed Machine Learning Model was able to identify the 100% of 1070 bagged racks that passed through the conveyer belt of a processing plant in real-time simulations.

5.3 Machine learning model that can alert a nearby user if the model is unsure, allowing for manual override

While the system is able to produce highly accurate results, it also incorporates functionality to allow the user to directly override the system's outputs. The touchscreen GUI, when run in the mode designed to display the system's detections and overall counts for each class, is able to alert the user to a detection about which the model indicates lower confidence, by highlighting its bounding box in red, and the user is able to override the counts manually using touchscreen controls if a primal is incorrectly classified by the system. An illustrative screen capture of the GUI running in this mode is shown in *Figure 10*.

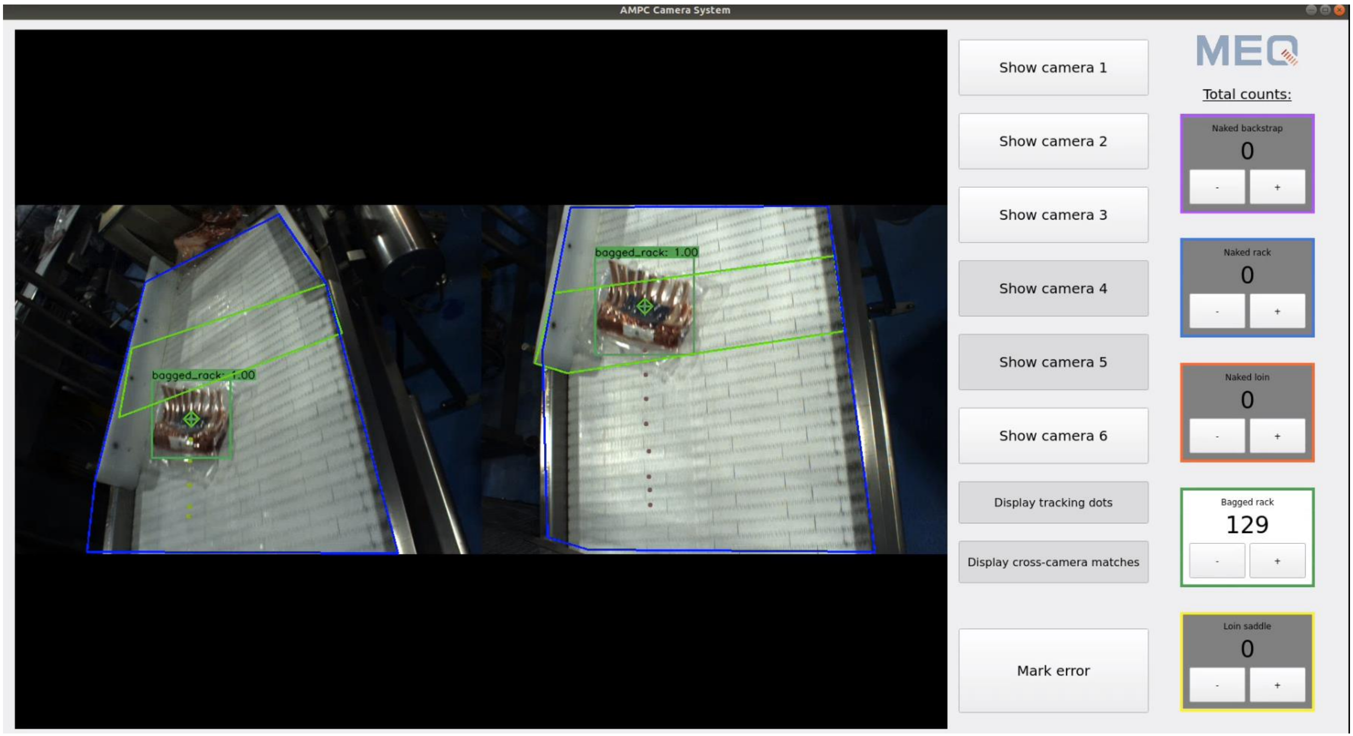


Figure 10: Screen capture of GUI showing bounding boxes and current-session counts for each object class. Note the red bounding box indicating a low-confidence detection (in this case, nevertheless a correct one), and the '+' and '-' buttons below each count, which allow the user to manually override the system's automatic counts.

6.0 Discussion

The validation results described in Section 4 demonstrate that the computer vision system is successful at identifying bagged racks with greater than 100% accuracy at chain speed. There is evidence that the system is more accurate than a human expert trying to identify primals in real-time, as it is a very complex task for a single person to detect multiple types of primals simultaneously at chain speed, and furthermore, the system is able to view the belt from multiple angles simultaneously and cross-reference between those angles in a way that would be impossible for a human.

Moreover, the success of this development corroborates the value that computer vision and machine learning techniques are able to provide to the operations of an abattoir boning room. Processors can benefit from these technologies as they bring objective measurement, alleviate risks, and help reduce labour costs.

7.0 Conclusions / Recommendations,

In conclusion, machine learning and computer vision technologies can be employed in an abattoir environment to detect bagged lamb rack primals with a high degree of accuracy. These methods show strong evidence in terms of applicability to boning room automation, analysis, and quality assurance tasks generally, with the potential to increase the reliability of abattoir processes and reduce labor costs.

We recommend that further research and development be undertaken to bring these technologies to further applications that have a direct and specific benefit to processors. In our opinion, these technologies could have immediate value in the following areas:

[REDACTED]

8.0 Bibliography

[REDACTED]

Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Hino, H., 2020. Active learning: Problem settings and recent developments. arXiv preprint arXiv:2012.04225.

Kim, J.A., Sung, J.Y. and Park, S.H., 2020, November. Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition. In 2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia) (pp. 1-4). IEEE.

Redmon, J. 2016. Darknet: Open Source Neural Networks in C. Available at: http://pjreddie.com/darknet/.

Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Gupta, B.B., Chen, X. and Wang, X., 2021. A survey of deep active learning. ACM Computing Surveys (CSUR), 54(9), pp.1-40.

9.0 Appendices

Appendix 1 - Initial Sample of Images

Figure 11: Six examples of bagged rack images collected in the plant throughout different days on one conveyor belt using different cameras.

Appendix 2 - Sample Images with Bounding Box Annotations

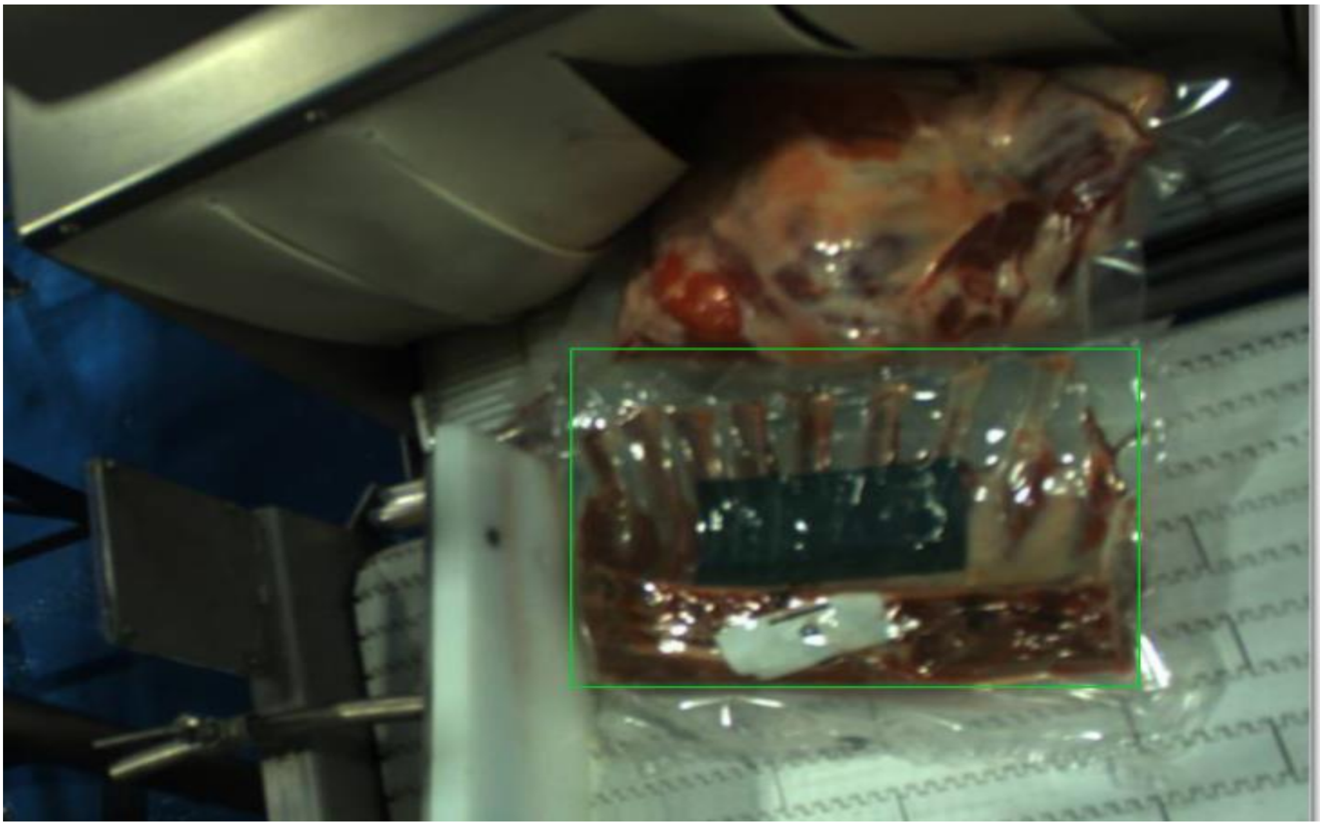


Figure 12: Image **[REDACTED]** *has one object annotated - one bagged_rack (green rectangles).*

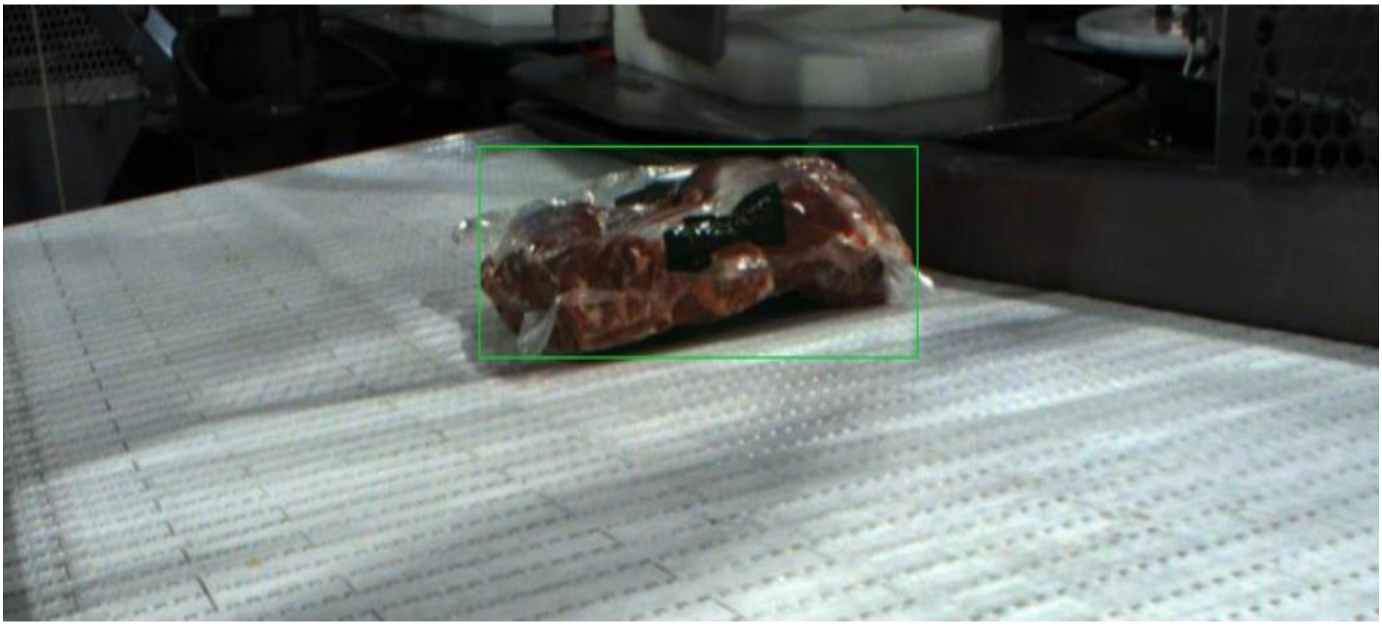


Figure 13: Image **[REDACTED]** *has one bagged rack(green rectangle) annotated.*

Appendix 3 - Sample Images with Bounding Box Annotations

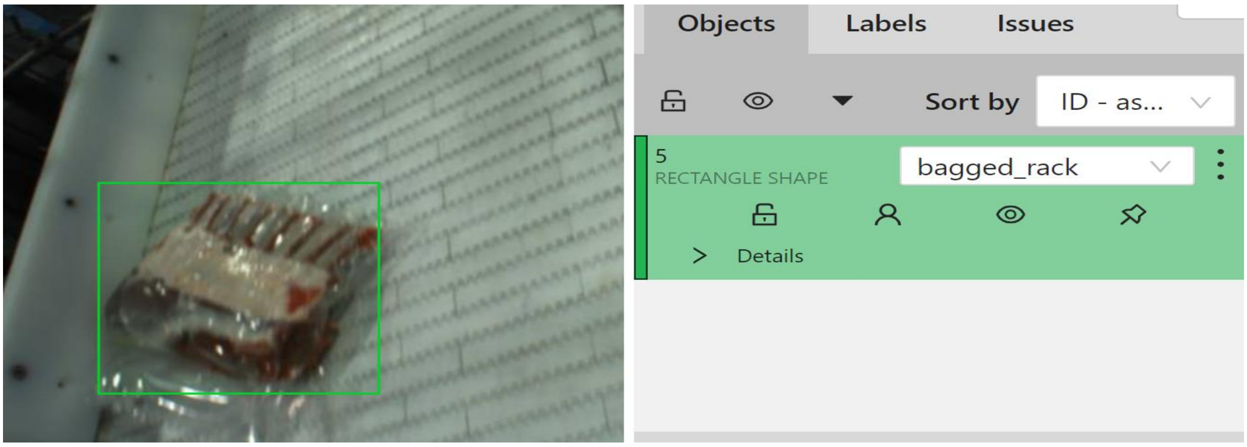


Figure 14: Image shows the graphical user interface used to create bounding boxes for bagged racks.

Appendix 4 - Example of data augmentation method: randomly cropped copies of the same image.

[REDACTED]

Appendix 5 - Sample of csv with labels

[REDACTED]