Running title: Automating Beef Boning Lines

Using Artificial Intelligence to Automate Meat Cut Identification from the *Semimembranosus* Muscle on Beef Boning Lines[1]

Satya Prakash[‡], Donagh P. Berry[‡], Mark Roantree[‡], Oluwadurotimi Onibonoje[‡], Leonardo Gualano[*], Michael Scriney[*] and Andrew McCarren[†, 2]

[*]Insight Centre for Data Analytics, Ireland[§]

[†]UPCOM, Ireland[§]

[‡]VistaMilk SFI Research Centre, Ireland[§]

[§]School of Computing, Dublin City University, Dublin 9, Ireland

[1]The authors gratefully acknowledge the support for this research to Science Foundation Ireland under grant numbers SFI/16/RC/3835, SFI/12/RC/2289-P2 and 20/COV/8436.

[2]**Corresponding author:** Andrew.McCarren@dcu.ie

Abstract

The identification of different meat cuts for labelling and quality control on production lines is still largely a manual process. As a result, it is a labor-intensive exercise with the potential for error but also bacterial cross-contamination. Artificial intelligence is used in many disciplines to identify objects within images but these approaches usually require a considerable volume of images for training and validation. The objective of this study was to identify five different meat cuts from images and weights collected by a trained operator within the working environment of a commercial Irish beef plant. Individual cut images and weights from 7987 meats cuts extracted from *Semimembranosus* muscles (i.e., Topside muscle), post-editing, were available. A variety of classical neural networks and a novel Ensemble machine learning approaches were then tasked with identifying each individual meat cut; performance of the approaches was dictated by accuracy (the percentage of correct predictions); precision (the ratio of correctly predicted objects relative to the number of objects identified as positive), and recall (also known as true positive rate or sensitivity). A novel Ensemble approach outperformed a selection of the classical neural networks including convolutional neural network (CNN) and residual network (ResNET). The accuracy, precision, and recall for the novel Ensemble method were 99.13%, 99.00%, and 98.00%, respectively, while that of the next best method were 98.00%, 98.00%, and 95.00%, respectively. The Ensemble approach, which requires relatively few gold-standard measures, can readily be deployed under normal abattoir conditions; the strategy could also be evaluated in the cuts from other primals or indeed other species.

Keywords: image identification, shelf-life, ensemble method, neural network, boning line

Abbreviations

DATAS, deductive analytics for tomorrows agri sector; MLR, multinomial logistic regression; DT, decision tree; CNN, convolutional neural network; ResNET, residual network; SOL, Start of Line; EOL, End of Line; PW, meat cut weights; PBWI, pre-processed black and white images; CI, colored images

Introduction

Access to a skilled and experienced workforce is fundamental to businesses that depend on human intervention in their production processes. The meat industry is one such sector, and this was highlighted by the levels of absenteeism during the COVID-19 restrictions. Processes such as meat cutting, fat determination, and meat deboning have been partially automated (Bostian et al., 1985; Umino et al., 2011). However, the labelling and identification of meat cuts still require a substantial amount of human intervention and manual handling. This can incur additional labor costs as well as being a source of error and potential microbiological contamination (Choi et al., 2013).

Primal boning lines are a typical example of where multiple operators simultaneously work on a range of meat cuts. Each cut will eventually arrive at a weighing station where a single operator will inspect, identify and weigh the arriving meat cut. The automation of the weighing process on boning lines has traditionally been conducted on single-meat-cut production lines. However, due to spatial restrictions in many meat plants, there is a preference in the beef industry to operate multiple meat cut types simultaneously on a single processing line. This multi-meat-cut processing strategy has made the automation of meat cut identification extremely challenging as there is a high probability of incorrect meat cut identification; any proposed automated system must have a high level of accuracy in order to avoid misclassification and line downtime.

Deep learning such as convolutional neural networks (CNN), a branch of machine learning, has become an increasingly popular method for image identification. In practice, CNN predictions can achieve human-level accuracy for tasks such as face recognition, image classification and real time object detection in an image or video (Du, 2018; Fan and Zhou, 2016; Zeng et al., 2017). CNNs are algorithms which are trained on labelled images (Wei et al., 2015). The training process is implemented by creating features from characteristics such as edges, dots, and lines on each image and then using these as inputs into a traditional neural network classification algorithm (Du, 2018).

The objective of this study was to collect image data and weights of individual meat cuts from the *Semimembranosus* primal, and to develop a methodology to correctly classify meat cuts from an image, resulting in an automated process for the identification of meat cuts. The Ensemble approach was then compared against various classical neural networks. The resulting algorithm enables the removal of a human operator, thus reducing the risk of cross-contamination across samples and potentially improving product shelf life.

Materials and Methods

All animals used as part of the study were reviewed and processed under the approval of the Irish Department of Agriculture following European Union Council Regulation (EC) N° 1099/2009.

Data

The data collected for this project were from beef cuts taken from a Topside (i.e., *Semimembranosus* muscle) trimming line of a major Irish beef processor. The process flow for this line required an operator to weigh the primal topside cut on a start-of-line (SOL) weighing scales. Each cut was then placed on a conveyor belt where a team of operators removed fat, gristle and secondary muscles. The remaining meat cuts were then labelled, weighed, and an image captured by a trained operator on an end-of-line (EOL) weighing scales, where the meat cuts were vacuum packed and labelled.

For this particular study, there were five different meat cuts derived from the Topside primal (Figure 1). The data acquisition required a hardware setup of weighing scales (Machines, 1985), at both the SOL and EOL together with a Vivotek bullet camera (IP8362 - Bullet - Network Cameras :: VIVOTEK ::) at the EOL to capture a photo image of each meat cut. In addition, bespoke data capture software using a node.js platform (Cantelon et al., 2013) was used to acquire the characteristics of each meat cut being weighed in a 4-step process.

1. A manual capture of the carcass identifier number, primal weight and the time of arrival at the SOL scales.
2. The time and the id of the operator validating the meat cut image as well as the meat cut weight, meat cut label, and a photo image at the EOL scales were all captured on bespoke data capture software used as a form of data acquisition in the development of an Agri Data Warehouse (McCarren et al, 2017).
3. The EOL operator identified the meat cut using the data capture interface (shown in Figure 2), ensuring the correct image was stored to disk and linked to the appropriate database entry containing the variables captured at both EOL and SOL points.

4. After each meat cut was removed from the scales, an image of the empty scales was captured. This was done to help remove image noise (discussed later).

The user interface for the data capture software is in Figure 2. A trained operator identified the meat cuts for subsequent categorization; the cuts were categorized as a) Cap Off, Pear Off, PAD, b) Cap Off, Pear On, c) Topside Heart PAD, d) Topside Bullet, or e) Cap Off Non PAD Blue Skin Only. The data collection period lasted 3 weeks and a summary of the data captured is in Table 1.

At the end of the data collection period, an analysis was conducted to determine if there were any outlying weights; this was undertaken by comparing the weights of the primal cut weighed on the SOL scales with the weight of the corresponding generated meat cut on the EOL scales. The ratio of each meat cut weighed on the EOL relative to the primal cut on the SOL is known as the product yield. Boning operators generally have target product yields which are dependent on the product specification. As the beef plant operator had a specification limit of 10.00% for each of the meat cuts used in these experiments, any absolute difference between the actual product yield and the target product yield that exceeded 10.00% was flagged as an outlier and subsequently removed from the dataset (Albertí *et al.*, 2005). As a result, 7,987 records were deemed acceptable for the final dataset (McCarren et al., 2021). Each record in this dataset included an image of the meat cut along with a corresponding weight and the batch number. The weights and images were then used as inputs to classification algorithms.

Image Pre-Processing

When conducting image pre-processing, one generally aims to improve the prediction process by enhancing certain characteristics and/or blurring others (Lancaster et al., 2018). For this study, each meat cut image was accompanied by its associated background image such as that shown in Figure 3. In order to remove distracting or confusing items (e.g., operator hands or small meat blobs), the background image was removed from the meat cut image. This image was then converted to grayscale (Figure 3), and finally, the meat cut was segmented from the scale using Gaussian blur technique. This final set of original and grayscale images were used in the model construction.

The frequency breakdown of the different topside meat cuts is in Table 1; the frequency of meat cut 20002 was disproportionately low as it is not frequently harvested in this plant. Therefore, it was decided to use data augmentation to create artificial training samples for meat cut 20002 in order to improve the imbalanced nature of the dataset. As part of the augmentation process, transformations such as anticlockwise rotation, clockwise rotation, horizontal flip, vertical flip, noise addition and blurring were implemented. These processes created 84 additional images for meat cut 20002 resulting in a final count of 98 images. The pre-processing and the application of deep learning algorithms was implemented using the Python programming language (*Python Release Python 3.6.0*), with the Tensorflow, Keras API, Scikit learn (Géron, 2019), and CV2 (Bradski and Kaehler, 2008) libraries.

Convolutional Neural Network

The CNN algorithm has shown particular success in identifying objects within images (Wallelign et al., 2018) and was therefore considered in the present study. The CNN algorithm processes data by passing images through multiple convolutional and pooling layers and applies non-linear transformations such as the Softmax or rectified linear unit (ReLU) function to obtain the probability-based classes (He and Chen, 2019). The functional form of a convolution layer is described in Eq. 1.

$$X_j^l = g\left(\sum_{i \in N_j} X_i^{l-1} * W_{ij}^l + B_j^l\right) \tag{1}$$

where l is a layer and j is an output, $\boldsymbol{X_j^l}$ is an output vector, $\boldsymbol{W_{ij}^l}$ is the convolution kernel (also known as weights or parameter estimates), $\boldsymbol{X_i^{l-1}}$ is the previous or hidden layer's feature map, $\boldsymbol{B_j^l}$ is an additive bias given to each output map, N_j represents the selection of the input maps, $*$ represents the convolution operation, and i is an element of the training set.

In a neural network, regularization is a technique to prevent overfitting. Overfitting occurs when the model is over-parameterized relative to the volume of data available. A loss function describes the deviation of predictions from the ground truth (Zhao et al., 2016) and is required to calculate the model error. The error for a single pattern can be expressed as in Eq. 2,

$$\epsilon^n = \epsilon^{n-1} + \lambda \sum_{i,j} \left|(\alpha_{ij})\right| \tag{2}$$

where $\in^n$ is the new error calculated after each iteration, $\in^{n-1}$ is the error from previous iteration and is highest for the first iteration, λ is a user-defined parameter that controls the trade-off and α_{ij} are the parameter estimates of the algorithm for a given output from layer i to j.

After each iteration of the CNN, the parameters and learning rates get updated in order to minimize the error (loss) using algorithms such as Adaptive Moment (Adam), which is a first-order gradient-based optimization of the stochastic function and is based on adaptive estimates of lower-order moments (Kingma and Ba, 2014). ReLU, a computationally inexpensive activation function, accelerates the training procedure by avoiding the vanishing gradient problem (He and Chen, 2019). In order to avoid overfitting, a CNN architecture which was originally used to identify numbers in a large handwritten dataset known as MNIST (Garg et al., 2019), was adapted by adding max-pooling and a dropout on each convolution layer (Park and Kwak, 2016).

CNN concatenated with Meat Cut Weights

In order to model the weight of each meat cut along with the cut images, the cut weights were integrated into the flattened layer of the CNN as mentioned above and described in Figure 4. Flattening the final convolution layer converts the images into a 1-dimensional array and transfers it to the fully connected, dense layer. The weight is concatenated with the 1-dimensional array and the last dense layer is used as an output layer which predicts the classes of the meat cut images.

Ensemble Approach with Meat Cut Weights

Theoretically, with CNN algorithms there is no need to engineer features during the classification process, as the mix of the convolution kernels and max pooling automatically creates features that can be inserted into a typical neural network (Liu et al., 2019). However, neural networks are highly non-linear and estimating the choice of initial parameter estimates can be computationally expensive. Creating a simplified set of initial features, such as the object extremities, and using these as inputs to a basket of simpler algorithms or an ensemble of algorithms has been found to be successful in other applications (Wang et al., 2019). In order to identify these object extremities, images were standardized by rotating them so that the longest side was always in a vertical position (Figure 5). From this image, the following hand crafted features were calculated using the CV2 Python library:

- Density: white pixel counts relative to the total number of pixels.
- (X_{min}, $X_{min}Y$): the minimum X and the corresponding Y coordinate.
- (X_{max}, $X_{max}Y$): the maximum X and the corresponding Y coordinate.
- ($Y_{min}X$, Y_{min}): the minimum Y and the corresponding X coordinate.
- ($Y_{max}X$, Y_{max}): the maximum Y and the corresponding X coordinate.

The Ensemble architecture is presented in Figure 6 as a 5-layer structure. At the first layer (Training: Data-Level 1), the handcrafted features, X_{min}, $X_{min}Y$, X_{max}, $X_{max}Y$, $Y_{min}X$, Y_{min}, $Y_{max}X$, $and\ Y_{max}$ were used in conjunction with each meat cut weight, together with a basket of machine learning approaches to identify each meat cut. The three base learners shown at layer 2 were Multinomial Logistic Regression (MLR), Decision Tree (DT) classifier, and CNN.

Multinomial logistic regression can be used for classification of a task with multiple response variables. The general equations of the MLR model are shown in Eq. 3 and 4, where: p_i is the probability of occurrence of each event; θ is the likelihood parameter; p_{k+1} represents the monotonicity of the lower bound iterate; $x = (x_1, \dots, x_m)^T$ is the covariate vector; k is the maximum number of possible outcomes; and θ^i is the parameter vector corresponding to the $i - th$ response category (Böhning, 1992; Li et al., 2010).

$$p_i = \frac{\exp(\theta^{(i)^T} x)}{1 + \sum_{j=1}^{k} \exp(\theta^{(j)^T} x)} \text{ for } i = 1, \dots, k \tag{3}$$

$$p_{k+1} = \frac{1}{1 + \sum_{j=1}^{k} \exp\left(\theta^{(j)^T} x\right)} \tag{4}$$

Decision tree (DT) classifiers are a rapid and useful top-down greedy approach to classify a dataset with a large number of variables (Farid et al., 2014). In general, each DT is a rule set. Researchers have used the ID3 (Iterative Dichotomizer) algorithm widely where objects are classified based on the improvement in information gain given by a proposed split in the tree (Chandra and Varghese, 2009). In the approach used in this study, the handcrafted features were used to calculate the information content and then the classes were subsequently predicted. In addition to the decision tree and MLR classifier, the CNN predictions were also included as part of the input layer to the neural network shown in Figure 6.

The predictions from the base learners comprise layer 3 of the ensemble architecture. The predictions $y_{1i}, y_{2i}, \ldots, y_{4k}, y_{5k}$ are shown in Figure 6, where i(s) are the predictions of MLR, j(s) are the predictions from DT classifier and k(s) are the predictions from CNN. These are then used in conjunction with the meat cut weights (PW) with an additional learner neural network (layer 4) and the final predictions of the meat cuts (20001, 20002, …, 20010), are delivered at layer 5 in the architecture.

Transfer Learning

Transfer Learning approaches such as a ResNET have been found to be successful in classifying images (He et al., 2016; Marsden et al., 2017; Setyono et al., 2018). A ResNET is a CNN with a skip connection, which is also known as an identity shortcut connection. The concept behind the skip connection is to enable gradients to flow between layers as they help to reduce the impact of the vanishing gradient problem in deep learning architectures. The general form is shown in Eq. 5, where a is the activation (outputs) of neurons in layer l, θ is the learning parameter, m is the total number of layers, $i = 1, 2, \ldots, m$ and $j = 0, 1, \ldots, m - 1$.

$$a^{(l+2i)} = g(\theta + a^{(l+2j)}) \tag{5}$$

A 34-layer ResNET architecture was used with and without considering cut weights in the present study. Such architectures are well-balanced and are as accurate as the CNN with relatively low computational power requirements (He et al., 2016).

Experimental Setup and Evaluation

Two broad sets of experiments were carried out in order to better understand the effect of a data transformation step on the predictive performance of the three applied algorithms. In the first set of experiments, the colored input images were transformed to grayscale which has been shown to reduce the noise-to-signal ratio (Vidal and Amigo, 2012), thus reducing the complexity and improving the performance of statistical learning techniques. In the second set of experiments, the color of the input images was retained as it was hypothesized that the color contrasts between the fat and meat components of each cut contained potentially useful information that would inform a better predictive performance. In each experiment, the datasets were split into a training set and a test set using an 80:20 stratified sampling ratio. The training set was further split using a 90:10 ratios for the purpose of implementing a validation strategy. The training data was used to train the model while the validation data was used to examine if the hyperparameters required further tuning. A hyperparameter is a parameter whose values cannot be estimated from the data and are external to the model. The test data was used as an unseen dataset to examine the results of the model.

Evaluation metrics used in image identification are typically accuracy, precision, recall, F1- score and convergence time (Al-Sarayreh et al., 2018; Larsen et al., 2014; Ropodi et al., 2015; Setyono et al., 2018; Wang et al., 2019). Accuracy and F1 scores are described in Eq. 6 and 7 respectively.

$$Overall\ Accuracy = \frac{\sum_{i=1}^{n=5} TP_i}{N} \quad (6)$$

In Eq. 6, TP_i or the true positive is the number of instances predicted correctly for instance i and N is the total number of predictions.

$$F1_i = 2 * \frac{Precision_i * Recall_i}{Precision_i + Recall_i} \quad (7)$$

where,

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (8)$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (9)$$

where FP_i or the false positive, is the number of instances where the true label is negative or of a different class but incorrectly predicted as positive, while FN_i or false negative, is the number of instances where the true label is positive but the class is incorrectly predicted as negative. The weighted-average F1 score was derived from the average F1 score from each classification category weighted by the number of meat cuts in each product group as shown in Eq. 10.

$$F1_{i(wt)} = \frac{\sum_{i=1}^{n} F1_i}{\mathrm{n}} \tag{10}$$

where n is the number of categories. Table 2 demonstrates the value of these metrics along with the time taken to converge for each algorithm.

In order to determine the statistical significance of the results, a beta regression model with a "loglog" link function was implemented in the R programming language using the betareg package (Cribari-Neto and Zeileis, 2010) to model accuracy against the algorithm, dataset and meat cut variables (R Core Team, 2020). Only 2-way interaction terms on combinations of the product, algorithm and image type were examined as the degrees of freedom in this particular analysis was limited to 40. The final beta regression model had pseudo R^2 of 0.98 and the comparison with an identity link was significant (Φ=350.37, z=3.99, p<0.001). A Type III analysis was conducted and interaction effects between algorithm and image type and between algorithm and product were found to be significant (Algorithm*Image Type $F_{4,26}$ = 3.046 and P =0.016, Algorithm*Product $F_{12,26}$ = 5.082 and P <0.001). From this analysis, a post-hoc analysis on the estimated marginal means with a Tukey correction for multiple comparisons was conducted and is outlined in Table 3.

Results

Accuracy statistics for each model and for both the color and grayscale images are in Table 2 for the training and test datasets. In addition, the convergence times for the color and grayscale images, for each method are also summarized in Table 2. While there was a wide disparity in convergence times, ranging from 1,745 seconds for the ResNET on the pre-processed black and white images to 19,224 seconds for the Ensemble approach with color images, it was not unexpected given the difference in model complexities.

The Ensemble approach with color images was the best-performing algorithm with a test accuracy of 99.13% and a training accuracy of 99.50%. The estimated marginal mean (EMM) for the test accuracy difference on color images was higher for the Ensemble approach compared with either the CNN ((EMM_{CNN}-$EMM_{Ensemble}$) Z_{score}= -4.72 or P <0.001) the ResNET (($EMM_{Ensemble}$-EMM_{ResNET}) Z score= 7.82 or P <0.001) algorithms without incorporating the cut weight information. The same algorithm also performed best for images in grayscale, with a test accuracy score of 95.00% and a train accuracy of the same value. However, the only statistical difference found was between the Ensemble and the ResNET without using cut weight information algorithms (($EMM_{Ensemble}$-EMM_{ResNET}) Z score= 4.42 or P <0.001). With a score of 98.00%, the Ensemble approach also had the highest weighted-average F1 score.

Figure 7 illustrates both the training and validation accuracy as the number of epochs changed for each method, for both the color and grayscale images. All approaches, with the exception of the Ensemble approach, demonstrated varying degrees of percentage difference in accuracy between the training and test accuracy on the grayscale images (CNN 4.80%, CNN with weights 5.80%, ResNET 0.90% and Ensemble 0.00%), implying the algorithms over-fitted the training data. The level of overfitting was reduced for both the CNN and the CNN that also used the cut weight information, albeit, there was a marginal increase in overfitting with the ResNET and Ensemble approaches for the color images (CNN 2.90%, CNN with weights 1.60% and ResNET 1.30% and Ensemble 0.43%).

All five algorithms, CNN, CNN concatenated with weights, ResNET, ResNET concatenated with weights and the Ensemble method performed better with color images, as the EMM difference between algorithms run on color images with those run on grayscale images was statistically significant ((EMM_{Color}-$EMM_{grayscale}$) Z_{ratio} = 13.649, P<0.001) as shown in Table 3.

The inclusion of product weights in the model demonstrated a beneficial effect when detecting meat cuts from images, as the CNN and Ensemble approaches when including weights out-performed the same algorithms when excluding the weights (($EMM_{CNN\ with\ Weights}$-EMM_{CNN}) Z_{ratio} = 3.527, P<0.015, (($EMM_{CNN\ with\ Weights}$-EMM_{ResNET}) Z_{ratio} = 5.37, P<0.001, (($EMM_{Ensemble}$-EMM_{CNN}) Z_{ratio} = 3.211, P<0.043, (($EMM_{Ensemble}$-EMM_{ResNET}) Z_{ratio} = 5.095, P<0.001) as shown in Table 3.

Figure 8 shows the F1 score for each model for each individual meat cut. In all cases, the highest F1 score was achieved for the Ensemble method with colored images (CI); while meat cut 20004 had the highest F1 score (100.00%) using the Ensemble method. Meat cut 20002, had the fewest number of images and correspondingly had the smallest F1 scores. However, using the Ensemble method with CI, meat cut 20002 did have the highest F1 score (97.00%).

Discussion

The primary aim of this study was to create an automated meat cut identification strategy for beef boning lines that simultaneously process multiple beef cuts; the present study focused solely on the cuts from the *Semimembranosus* muscle. In order to do this, a number of classical neural network that perform image detection, and a novel Ensemble strategy were applied to a dataset (McCarren *et al.*, 2021) consisting of 7,987 product cut images and their corresponding weights. A series of eight experiments were conducted on both color and preprocessed grayscale images and the novel Ensemble approach developed in this study performed best for each individual cut and that using color images outperformed those that used grayscale while availing of product weights also improved the accuracy of categorization. These results demonstrated findings relating to AI and implementation strategies that would be applicable for future commercial deployment strategies.

AI Strategy

Typically, in image detection problems, one highlights image features using a variety of pre-processing techniques to improve the algorithm's performance. However, on the live production environment, where these experiments were conducted, the opposite result was found; accuracy and weighted-average F1 score was 4.00% higher for all models using color images. While this is not typical in object detection problems (Xu *et al.*, 2016), the occurrence in these experiments can be explained by the fact that the background remained relatively constant throughout the experimental period, thus removing it from the images had little or no effect. In addition, grayscaling the images potentially limited the ability of all algorithms to differentiate between the fat and red meat.

In the meat industry, meat cuts are generally extracted from primal cuts, and knowing the weights of these cuts can potentially help in the identification of candidate labels. Results from the present study clearly demonstrate a benefit of knowing the weight of the on-coming cut, as the inclusion of the product weight into the flat layer of both the CNN and ResNET improved the resulting meat cut identification. This is not surprising as it has been shown to be successful in previous research on product identification (Shi et al., 2020). However, in this study a simplified model where product weights alone were used as the only independent variable resulted in an accuracy of 60.12% on the test dataset. This result justifies the importance of the product weights but also demonstrates that the product weights alone are not sufficient for categorizing product cuts.

Transfer learning is one of the more recent evolutions of machine learning and, in particular, the ResNET transfer learning algorithm is considered to be one of the most advanced deep learning architectures in image detection (Marsden et al, 2017). However, in the experiments conducted in the present study, the incorporation of the weight of each meat cut in the final layer and the outputs of the simpler approaches outperformed the ResNET architecture. While this was somewhat surprising, the combined use of multinomial logistic regression, the CNN and the decision tree algorithm in the ensemble approach on the set of artificially created features, was the most consistent with respect to overfitting and suggests that the use of simpler algorithms in the Ensemble approach may have assisted the CNN algorithm in finding a stable solution. While the Ensemble approach with color images took longer to converge, the ability to avoid overfitting is extremely important in a live environment. In a live environment, the convergence time would not be a considerable issue as model fitting would only be implemented in order to calibrate the model in an offline mode. Finding a stable solution can be an issue when using Neural Network algorithms as the level of non-linearity in the cost function can cause overfitting (Nguyen et al., 2011). Using a mixture of simpler algorithms in the early stage of the Ensemble has been shown to outperform more complex methods with regard to accuracy and F1-score (Abdelaal et al., 2018) and to reduce overfitting (Perrone and Cooper, 1992). GC *et al.*, (2021) achieved a maximum test accuracy of 98.57% and a weighted average F1-score of 94.00% on the test dataset of beef cuts using the alternative VGG16 transfer learning model, a state-of-the-art method. The proposed Ensemble method was able to achieve an accuracy up to 99.13% and weighted-average F1-score of 98.00%.

Deployment Strategy

The data capture unit developed in the present study was implemented using the Node.js programming language, and consisted of a DEM weighing scales (Machines, 1985), a DEM terminal and a Vivotek harsh environment camera. In order to truly automate the collection of the cut weight and subsequently identify the products in a live environment, an external harsh environment color camera will need to be integrated into an inline weighing scales. The terminal for this scales will then need a script that runs the Ensemble machine learning models; however, the code used to create the Ensemble approach in the present study can be easily integrated into many diverse operating systems. For each new group of products, the algorithm will need to be trained on images collected from the live production of the corresponding plant. The number of samples required to train the algorithm will be problem specific. However, in previous research studies, researchers have recommended that at least 1000 images of each object should be used during the AI training phase (Cho et al., 2016). This is not a hard rule and in this study the results demonstrated that there was ample data with the exception of product 20002, where the overall accuracy was lower. As mentioned previously, the data collection for this study was implemented on bespoke software. This code can be readily implemented to help create training data for the Ensemble machine learning algorithm during new deployments and makes the implementation in a commercial environment an attractive proposition.

The cost of deployment is not envisaged to be expensive for a live environment as all the software used is open source (Tilkov and Vinoski, 2010; *Python Release Python 3.6.0*). The camera technology is relatively inexpensive as the image processing in the present study was conducted without the use of spectral images which was not the case in other studies (Larsen *et al.*, 2014; Ropodi *et al.*, 2015; Al-Sarayreh *et al.*, 2018; Yu *et al.*, 2018). The advancement in object detection algorithms and the inclusion of the weights seems to have negated the need for infrared spectroscopy infrared and potentially could be used in many other applications in the food industry. The test accuracy with the ensemble algorithm demonstrates the ability of artificial intelligence to replicate the behavior of a human operator.

Applications

In the meat processing industry, the decision to implement automated or robotic processes is usually dictated by the return-on-investment which, in turn, is usually a function of improved product quality, reduced labor costs or a reduction in safety incidents (Purnell 2013). Automation has been introduced in the sector and has been used in applications such as fat and red meat yield prediction (Pabiou *et al.*, 2011) and a limited number cutting procedures. However, beef boning is still predominantly a highly manual process on modern pace boning lines. These operations rely on operators at the end of the line to identify products, check their quality characteristics, and then manually redirect them to the appropriate packing stations. At present, in operations were there are multiple cuts being processed simultaneously, there is generally no facility to monitor yields during the boning process. This is a major weakness in current systems as plant management rely on in line supervision to continually monitor the operator cut decisions of boning operators. By automating the identification of the relevant meat cuts and, in conjunction with automated weighing technology, the yield of the cut relative to the original primal weight can be accurately monitored during production rather than at the end the batch, thus improving the meat yield of the plant.

In addition to potential yield improvement, removing an operator on the line can potentially reduce the possibility for cross contamination from bacteria such as *Staphylococcus* or *Escherichia coli* which are commonly transmitted on food operations by line operators (Coma, 2008). However, the potential for misspecification of the meat cut could potentially rise without the use of a trained human operator. In order to avoid this issue, the system applied in this study could be adapted to remove products onto a separate QC line if it either did not recognize the meat cut or it was outside the weight specification, effectively mimicking the actions of a human operator.

Conclusions

In the present study, an approach to automate the identification of meat cuts was presented using a live beef production line over a three-week period. It was unclear at the outset as to which machine learning model would perform best on these types of images in the live environment and thus a number of computer vision algorithms were evaluated. As is normal with the construction of a new dataset, imbalances in terms of image distribution frequencies can occur but this was offset using different pre-processing methods and data augmentation. The outcome was that an Ensemble approach, with a mixture of CNN, multinomial logistic regression, and decision tree classifiers that incorporated product weights, had the best performing result in terms of accuracy and weighted F1-score. The results also showed that the CNN-multi-inputs converged 33.00% faster than the Ensemble approach, although this model was 1.00% less accurate on the test dataset and showed less promising results when the training and validation loss graphs were examined. This work focuses on constructing a larger dataset with a broader range of primal cuts and the next step is to apply the best performing model on a more challenging dataset to demonstrate if the overall process can be used in a full-scale commercial application.

Conflict of interest

The authors have no conflict of interest to declare.

Literature Cited

Abdelaal, H.M., Elmahdy, A.N., Halawa, A.A., Youness, H.A., 2018. Improve the automatic classification accuracy for Arabic tweets using ensemble methods. J. Electr. Syst. Inf. Technol. 5, 363–370. https://doi.org/10.1016/j.jesit.2018.03.001

Albertí, P., Ripoll, G., Goyache, F., Lahoz, F., Olleta, J.L., Panea, B. and Sañudo, C., 2005. 'Carcass characterisation of seven Spanish beef breeds slaughtered at two commercial weights', Meat Science, 71(3), pp. 514–521. doi:10.1016/j.meatsci.2005.04.033.

Al-Sarayreh, M., M Reis, M., Qi Yan, W., Klette, R., 2018. Detection of red-meat adulteration by deep spectral–spatial features in hyperspectral images. J. Imaging 4, 63. https://doi.org/10.3390/jimaging4050063

Anaconda Individual Edition — Anaconda documentation (no date). Available at: https://docs.anaconda.com/anaconda/ (Accessed: 18 October 2021).

Böhning, D., 1992. Multinomial logistic regression algorithm. Ann. Inst. Stat. Math. 44, 197–200. https://doi.org/10.1007/BF00048682

Bostian, M.L., Fish, D.L., Webb, N.B., Arey, J.J., 1985. Automated methods for determination of fat and moisture in meat and poultry meat cuts: collaborative study. J. Assoc. Off. Anal. Chem. 68, 876–880. https://doi.org/10.1093/jaoac/68.5.876

Bradski, G. and Kaehler, A. (2008) Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc.

Cantelon, M., Harter, M., Holowaychuk, T.J., Rajlich, N., 2013. Node. Js in Action, Greenwich, CT. USA: Manning Publications Co.

Chandra, B., Varghese, P.P., 2009. Moving towards efficient decision tree construction. Inf. Sci. 179, 1059–1069. https://doi.org/10.1016/j.ins.2008.12.006

Cho, J. et al. (2016) 'How much data is needed to train a medical image deep learning system to achieve necessary high accuracy?', arXiv:1511.06348 [cs]. Available at: http://arxiv.org/abs/1511.06348 (Accessed: 3 September 2021).

Choi, S., Zhang, G., Fuhlbrigge, T., Watson, T., Tallian, R., 2013. Applications and requirements of industrial robots in meat processing, in: 2013 IEEE International Conference on Automation Science and Engineering (CASE). IEEE, pp. 1107–1112. https://doi.org/10.1109/CoASE.2013.6653967

Coma, V., 2008. Bioactive packaging technologies for extended shelf life of meat-based products. Meat Sci., Symposium on Meat safety: From Abattoir to Consumer 78, 90–103. https://doi.org/10.1016/j.meatsci.2007.07.035

Cribari-Neto, F. and Zeileis, A. (2010) 'Beta Regression in R', Journal of Statistical Software, 34, pp. 1–24. doi:10.18637/jss.v034.i02.

Du, J., 2018. Understanding of object detection based on CNN family and YOLO, in: Journal of Physics: Conference Series. IOP Publishing, p. 012029. https://doi.org/10.1088/1742-6596/1004/1/012029

Fan, H., Zhou, E., 2016. Approaching human level facial landmark localization by deep learning. Image Vis. Comput. 47, 27–35. https://doi.org/10.1016/j.imavis.2015.11.004

Farid, D.M., Zhang, L., Rahman, C.M., Hossain, M.A., Strachan, R., 2014. Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks. Expert Syst. Appl. 41, 1937–1946. https://doi.org/10.1016/j.eswa.2013.08.089

Garg, A., Gupta, D., Saxena, S., Sahadev, P.P., 2019. Validation of random dataset using an efficient CNN model trained on MNIST handwritten dataset, in: 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN). IEEE, pp. 602–606. https://doi.org/10.1109/SPIN.2019.8711703

Géron, A. (2019) Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778.

He, X., Chen, Y., 2019. Optimized input for CNN-based hyperspectral image classification using spatial transformer network. IEEE Geosci. Remote Sens. Lett. 16, 1884–1888. https://doi.org/10.1109/LGRS.2019.2911322

IP8362 - Bullet - Network Cameras :: VIVOTEK :: [WWW Document], n.d. URL https://www.vivotek.com/ip8362 (accessed 5.27.21).

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. ArXiv Prepr. ArXiv14126980.

Lancaster, J., Lorenz, R., Leech, R., Cole, J.H., 2018. Bayesian optimization for neuroimaging pre-processing in brain age classification and prediction. Front. Aging Neurosci. 10, 28. https://doi.org/10.3389/fnagi.2018.00028

Larsen, A.B.L., Hviid, M.S., Jørgensen, M.E., Larsen, R., Dahl, A.L., 2014. Vision-based method for tracking meat cuts in slaughterhouses. Meat Sci. 96, 366–372. https://doi.org/10.1016/j.meatsci.2013.07.023

Li, J., Bioucas-Dias, J.M., Plaza, A., 2010. Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. IEEE Trans. Geosci. Remote Sens. 48, 4085–4098. https://doi.org/10.1109/TGRS.2010.2060550

Liu, H., Lang, B., Liu, M., Yan, H., 2019. CNN and RNN based payload classification methods for attack detection. Knowl.-Based Syst. 163, 332–341. https://doi.org/10.1016/j.knosys.2018.08.036

Marsden, M., McGuinness, K., Little, S., O'Connor, N.E., 2017. Resnetcrowd: A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification, in: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, pp. 1–7. https://doi.org/10.1109/AVSS.2017.8078482

Machines, D. (1985) 'Marine Weighing Scale | Mosion Compensating Marine Scale', DEM Machines. Available at: https://demmachines.com/product/marine-weighing-scale/ (Accessed: 27 May 2021).

McCarren, A., McCarthy, S., Sullivan, C.O., Roantree, M., 2017. Anomaly detection in agri warehouse construction, in: Proceedings of the Australasian Computer Science Week Multiconference. pp. 1–10.

McCarren, A., Scriney, M., Roantree, M., Gualano, L., Onibonoje, O., Prakash, S., 2021. Meat Cut Image Dataset (BEEF). https://doi.org/10.5281/zenodo.4704391

Nguyen, H.M., Couckuyt, I., Knockaert, L., Dhaene, T., Gorissen, D., Saeys, Y., 2011. An alternative approach to avoid overfitting for surrogate models, in: Proceedings of the 2011 Winter Simulation Conference (WSC). IEEE, pp. 2760–2771. https://doi.org/10.1109/WSC.2011.6147981

Pabiou, T., Fikse, W.F., Cromie, A.R., Keane, M.G., Näsholm, A. and Berry, D.P., 2011 'Use of digital images to predict carcass cut yields in cattle', Livestock Science, 137(1), pp. 130–140. doi: 10.1016/j.livsci.2010.10.012.

Park, S., Kwak, N., 2016. Analysis on the dropout effect in convolutional neural networks, in: Asian Conference on Computer Vision. Springer, pp. 189–204.

Perrone, M.P., Cooper, L.N., 1992. When networks disagree: Ensemble methods for hybrid neural networks. BROWN UNIV PROVIDENCE RI INST FOR BRAIN AND NEURAL SYSTEMS.

Purnell, G., 2013. Robotics and automation in meat processing, in: Robotics and Automation in the Food Industry. Elsevier, pp. 304–328.

Python Release Python 3.6.0 [WWW Document], n.d. . Python.org. URL https://www.python.org/downloads/release/python-360/ (accessed 8.5.21).

R Core Team (2020) R: A language and environment for statistical computing. Available at: https://www.r-project.org/ (Accessed: 3 September 2021).

Ropodi, A.I., Pavlidis, D.E., Mohareb, F., Panagou, E.Z., Nychas, G.-J., 2015. Multispectral image analysis approach to detect adulteration of beef and pork in raw meats. Food Res. Int. 67, 12–18. https://doi.org/10.1016/j.foodres.2014.10.032

Setyono, N.F.P., Chahyati, D., Fanany, M.I., 2018. Betawi Traditional Food Image Detection using ResNet and DenseNet, in: 2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS). IEEE, pp. 441–445. https://doi.org/10.1109/ICACSIS.2018.8618175

Shi, H., Qin, C., Xiao, D., Zhao, L., Liu, C., 2020. Automated heartbeat classification based on deep neural network with multiple input layers. Knowl.-Based Syst. 188, 105036. https://doi.org/10.1016/j.knosys.2019.105036

Tilkov, S., Vinoski, S., 2010. Node. js: Using JavaScript to build high-performance network programs. IEEE Internet Comput. 14, 80–83.

Umino, T., Uyama, T., Takahashi, T., Goto, O., Kozu, S., 2011. Automatic deboning method and apparatus for deboning bone laden meat. Automatic deboning method and apparatus for deboning bone laden meat. U.S. Patent 8,070,567.

Vidal, M., Amigo, J.M., 2012. Pre-processing of hyperspectral images. Essential steps before image analysis. Chemom. Intell. Lab. Syst. 117, 138–148.

https://doi.org/10.1016/j.chemolab.2012.05.009

Wallelign, S., Polceanu, M., Buche, C., 2018. Soybean plant disease identification using convolutional neural network, in: The Thirty-First International Flairs Conference.

Wang, R., Li, W., Zhang, L., 2019. Blur image identification with ensemble convolution neural networks. Signal Process. 155, 73–82. https://doi.org/doi.org/10.1016/j.sigpro.2018.09.027

Wei, Y., Xia, W., Lin, M., Huang, J., Ni, B., Dong, J., Zhao, Y., Yan, S., 2015. HCP: A flexible CNN framework for multi-label image classification. IEEE Trans. Pattern Anal. Mach. Intell. 38, 1901–1907. https://doi.org/10.1109/TPAMI.2015.2491929

Xu, Y., Zhang, Z., Lu, G. and Yang, J., 2016 'Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification', Pattern Recognition, 54, pp. 68–82. doi:10.1016/j.patcog.2015.12.017.

Yu, X., Tang, L., Wu, X. and Lu, H., 2018 'Nondestructive freshness discriminating of shrimp using visible/near-infrared hyperspectral imaging technique and deep learning algorithm', Food analytical methods, 11(3), pp. 768–780. doi:10.1007/s12161-017-1050-8.

Zeng, T., Wu, B., Ji, S., 2017. DeepEM3D: approaching human-level performance on 3D anisotropic EM image segmentation. Bioinformatics 33, 2555–2562. https://doi.org/10.1093/bioinformatics/btx188

Zhao, H., Gallo, O., Frosio, I., Kautz, J., 2016. Loss functions for image restoration with neural networks. IEEE Trans. Comput. Imaging 3, 47–57. https://doi.org/10.1109/TCI.2016.2644865

List of Figures

Figure 1:

Title: Topside cuts

Caption: 5 meat cut variations. (a) Cap Off Pear Off, PAD topside muscle (20001); (b) Cap off, Pear on topside muscle (20002); (c) Topside Heart muscle (20003); (d) Topside Bullet muscle (20004); and (e) Cap Off, Non PAD, Blue Skin Only topside muscle (20010).

Figure 2:

Title: End of Line (EOL)

Caption: A user interface for data collection

Figure 3:

Title: Images at various stages of pre-processing

Caption: a) is the background image reflecting the scale on which the meat cuts were placed, b) shows the scale with a meat cut on it, c) is the difference between image a, and b, d) is the grayscale conversion of image c and e) represents the segmented meat cut.

Figure 4:

Title: Convolutional neural network with meat cut weight

Caption: Architecture where the weight is concatenated with the image in the flattened layer.

Figure 5:

Title: Handcrafted features

Caption: These features are created from, the co-ordinates of the virtual box surrounding the meat cut.

Figure 6:

Title: Ensemble architecture

Caption: Integrating the Multinomial Logistic Regression (MLR), Decision Tree classifier (DTC) and Convolutional Neural Network Learners, where the handcrafted features X_{min}, $X_{min}Y$, X_{max}, $X_{max}Y$, $Y_{min}X$, Y_{min}, $Y_{max}X$, Y_{max} and images are used as inputs. The outputs y_{1i}.. y_{5k} are the predictions of each product cut from the MLR, DTC and CNN algorithms, which are then fed to a standard Neural Network(NN), whose outputs correspond to prediction of product cuts, 20001, 20002, 20003, 20004, and 20010.

Figure 7:

Title: Training and Validation Loss Graphs

Caption: (a), (c), (g), and (h) show the overfitting as there is a significant difference between the train and the valid curves. In (b), (d), (e), and (f), there is no overfitting as the two lines are almost overlapping showing very minimal or no differences between train and valid results.

Figure 8:

Title: F1 score:

Caption: It shows the F1 score for all five meat cuts with different models on both the pre-processed black and white (PBWI) and the colored images (CI).

Tables

Table 1: Dataset summary statistics

Meat cut ID	N	Meat cut description	$\underline{X} \pm S$	Cut yield (%)
20001	1060	Cap Off, Pear Off, PAD	6.47 ± 1.17	55.11
20002	14	Cap Off, PAD On	8.87 ± 0.98	68.18
20003	2132	Topside Heart PAD	5.87 ± 1.10	44.00
20004	2085	Topside Bullet	1.40 ± 0.29	9.45
20010	2696	Cap Off Non PAD Blue Skin Only	7.82 ± 1.59	61.55

N is the frequency of the images, and $\bar{X}$ and S are mean and standard deviation of weights, respectively.

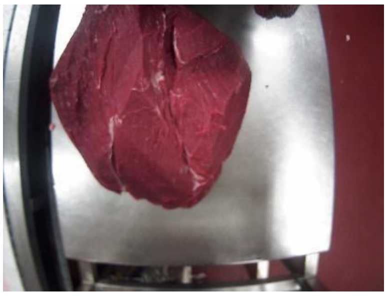
Table 2: Comparative performances for all 3 Models: The accuracy for the training and test datasets, and the weighted-average F1 score for the test dataset are shown in the columns Train Accuracy, Test Accuracy, and Test Weighted F1 Score respectively. For each of the models there are two rows representing the pre-processed black and white images (PBWI) and the colored images (CI). The time (s) column displays the time, in seconds, to train the model.

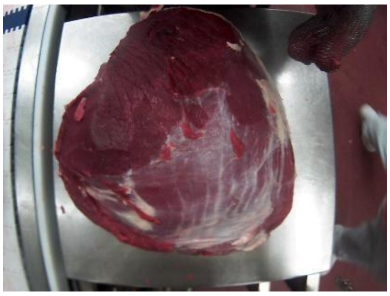
Model	**Image Type**	**Train Accuracy**	**Test Accuracy**	**Average Precision (Test)**	**Average Recall (Test)**	**Weighted F1 Score (Test)**	**Time(s)**
CNN	PBWI	96.80%	92.00%	86.00%	82.00%	84.00%	6675
CNN	CI	98.90%	96.00%	96.00%	92.00%	92.00%	3093
CNN with weights	PBWI	98.80%	93.00%	91.00%	83.00%	86.00%	6059
CNN with weights	CI	99.60%	98.00%	98.00%	95.00%	96.00%	11251
ResNET	PBWI	91.80%	90.90%	90.90%	90.80%	90.80%	1745
ResNET	CI	96.80%	96.50%	96.50%	96.00%	96.00%	12500
ResNET with weights	PBWI	95.20%	92.00%	90.00%	78.00%	81.00%	8345
ResNET with weights	CI	99.10%	97.00%	97.00%	87.00%	90.00%	9278
Ensemble	PBWI	95.00%	95.00%	92.00%	82.00%	85.00%	18518
Ensemble	CI	99.50%	99.13%	99.00%	98.00%	98.00%	19224

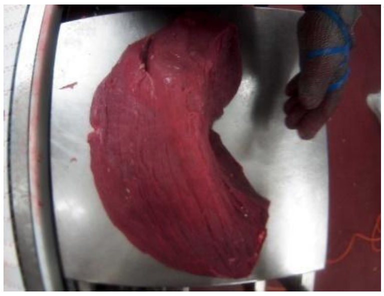
Table 3: Tukey Post hoc contrast analysis of predicted marginal mean difference (SE) between algorithms by image type.

Image Type	**Contrast**	**Marginal Mean Difference (SE)**	**Z ratio**	**Adjusted P value**
Color	CNN - CNN with weights	-0.7573 (0.215)	0.0153	0.0153
Color	CNN - Ensemble	-0.6719 (0.209)	-3.211	0.0433
Color	CNN - ResNET	0.3505 (0.174)	2.017	0.5871
Color	CNN - ResNET with Weight	-0.0801 (0.186)	-0.429	1
Color	CNN with weights - Ensemble	0.0854 (0.237)	0.361	1
Color	CNN with weights - ResNET	1.1078 (0.206)	5.37	<.0001
Color	CNN with weights - ResNET with Weight	0.6773 (0.217)	3.122	0.0566
Color	Ensemble - ResNET	1.0224 (0.201)	5.095	<.0001
Color	Ensemble - ResNET with Weight	0.5919 (0.212)	2.798	0.137
Color	ResNET - ResNET with Weight	-0.4305 (0.177)	-2.437	0.3036
Grayscale	(CNN) - (Ensemble)	-0.4688 (0.175)	-2.687	0.1789
Grayscale	(CNN) - (ResNET)	0.2886 (0.145)	1.996	0.602
Grayscale	(CNN) - (ResNET with Weight)	-0.0423 (0.155)	-0.272	1
Grayscale	(CNN with weights) - (Ensemble)	-0.3269 (0.183)	-1.782	0.7468
Grayscale	(CNN with weights) - (ResNET)	0.4306 (0.155)	2.769	0.147
Grayscale	(CNN with weights) - (ResNET with Weight)	0.0997 (0.166)	0.602	0.9999
Grayscale	(Ensemble) - (ResNET)	0.7575 (0.171)	4.422	0.0004
Grayscale	(Ensemble) - (ResNET with Weight)	0.4266 (0.180)	2.364	0.3477

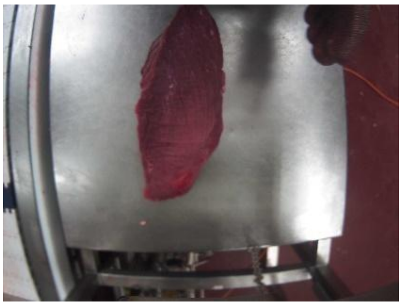
Figures

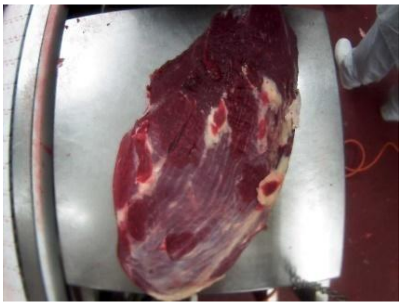






(a) Meat cut 20001 (b) Meat cut 20002 (c) Meat cut 20003





(d) Meat cut 20004 (e) Meat cut 20010

Figure 1: Topside cuts: 5 meat cut variations. (a) Cap Off Pear Off, PAD topside muscle (20001); (b) Cap off, Pear on topside muscle (20002); (c) Topside Heart muscle (20003); (d) Topside Bullet muscle (20004); and (e) Cap Off, Non PAD, Blue Skin Only topside muscle (20010).

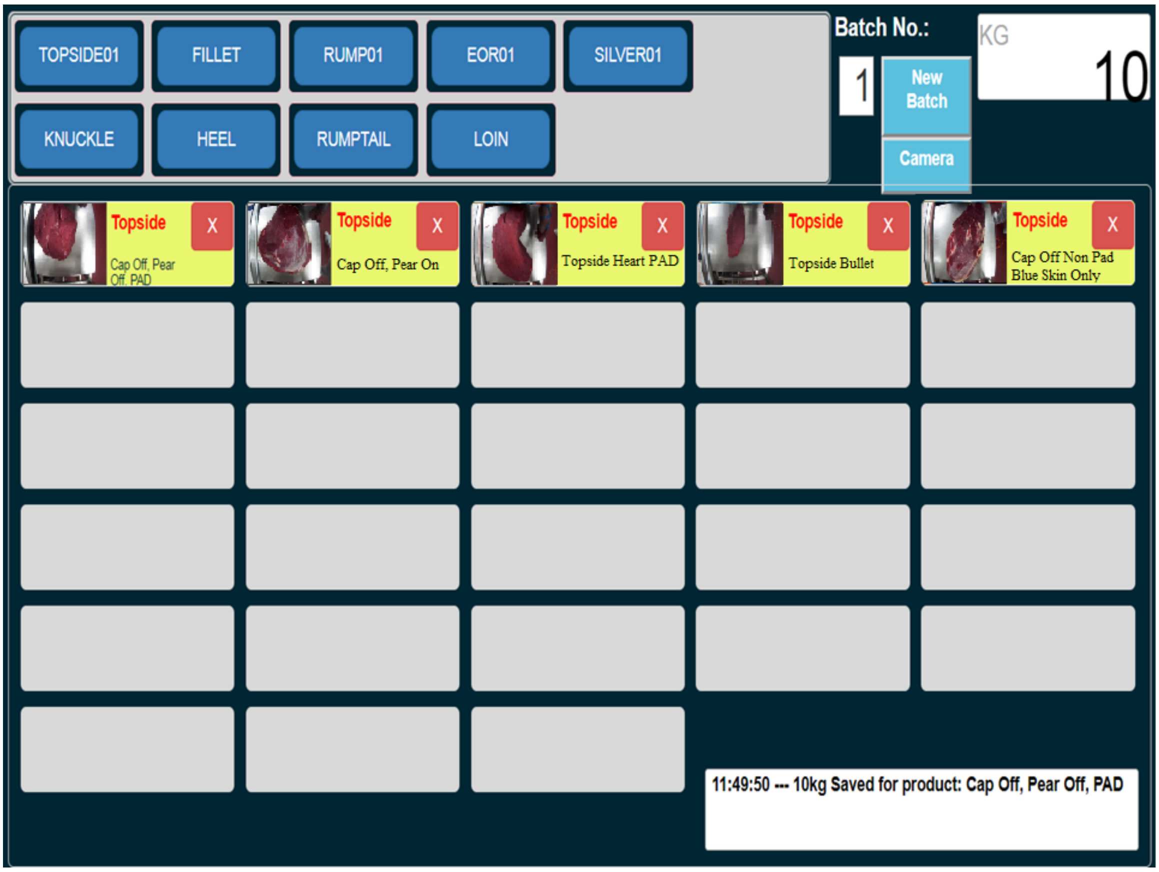
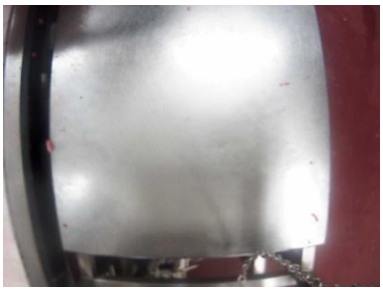
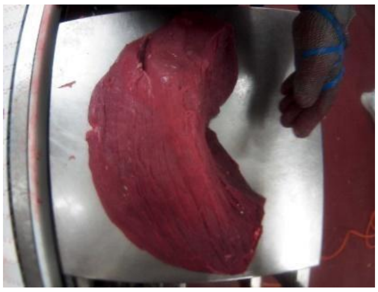


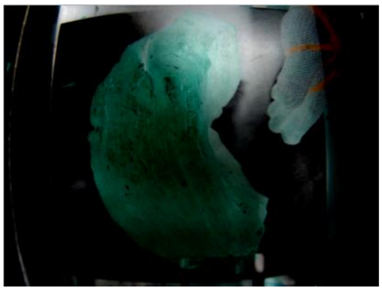
Figure 2: End of Line (EOL): A user interface for data collection



(a) Background image

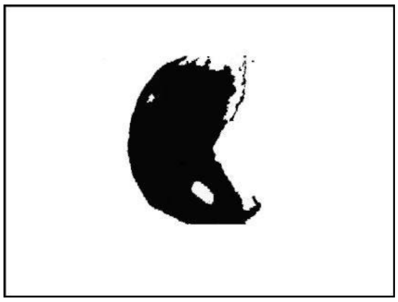


(b) Scale with meat cut and operator's hand



(c) Difference of background and meat cut

(d) Converted grayscale image of the difference



(e) Segmenting only meat cut

Figure 3: Images at various stages of pre-processing: a) is the background image reflecting the scale on which the meat cuts were placed, b) shows the scale with a meat cut on it, c) is the difference between image a, and b, d) is the grayscale conversion of image c and e) represents the segmented meat cut.

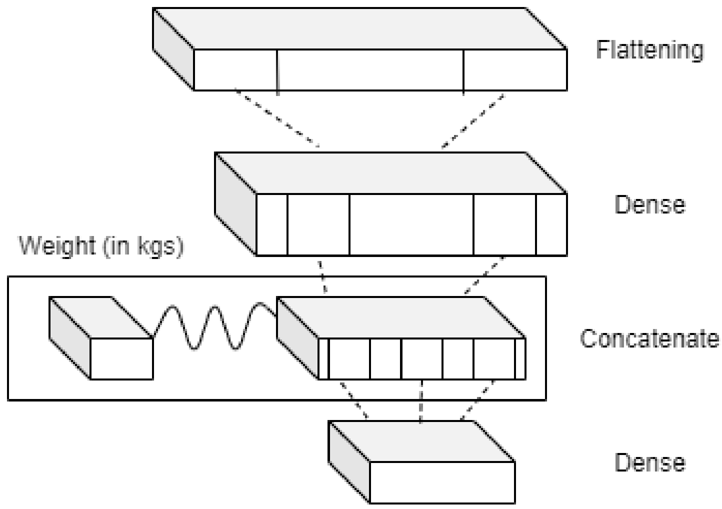


Figure 4: Convolutional neural network with meat cut weight: Architecture where the weight is concatenated with the image in the flattened layer.

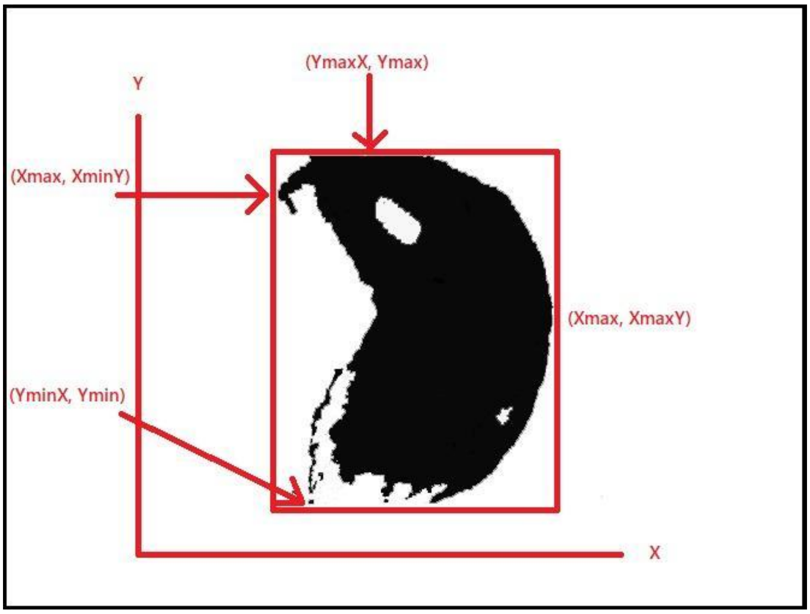


Figure 5: Handcrafted features: These features are created from, the co-ordinates of the virtual box surrounding the meat cut.

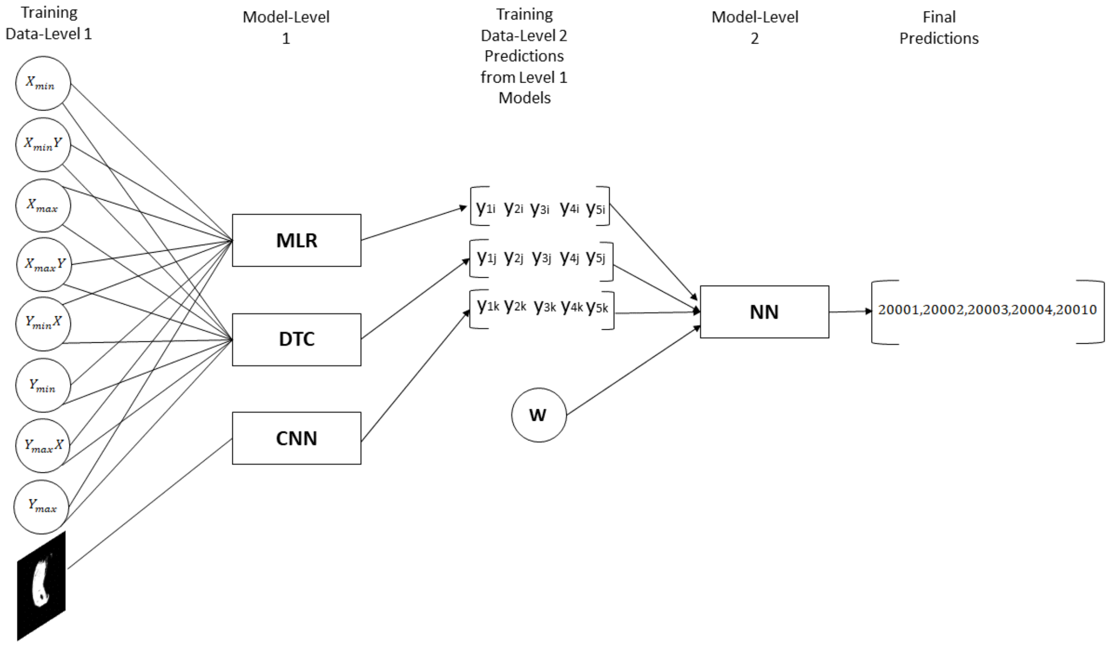
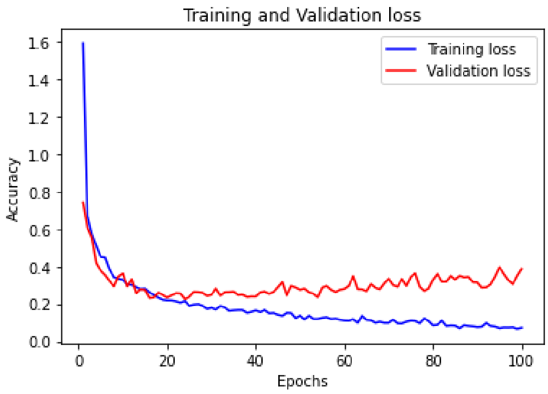
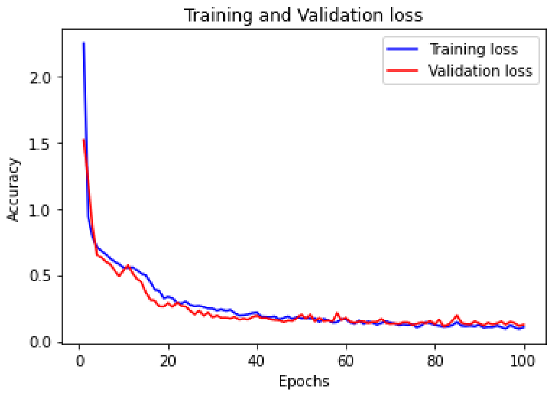


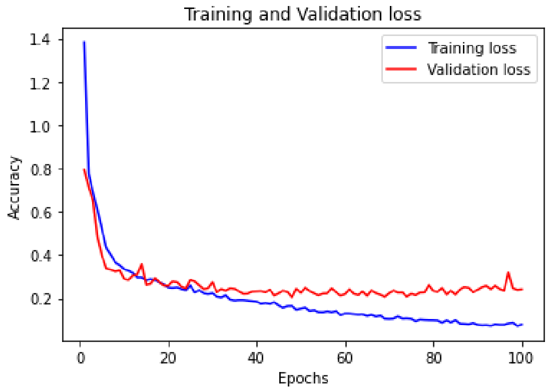
Figure 6: Ensemble architecture: Integrating the Multinomial Logistic Regression (MLR), Decision Tree classifier (DTC) and Convolutional Neural Network Learners, where the handcrafted features X_{min}, $X_{min}Y$, X_{max}, $X_{max}Y$, $Y_{min}X$, Y_{min}, $Y_{max}X$, Y_{max} and images are used as inputs. The outputs y_{1i}.. y_{5k} are the predictions of each product cut from the MLR, DTC and CNN algorithms, which are then fed to a standard Neural Network(NN), whose outputs correspond to prediction of product cuts, 20001, 20002, 20003, 20004, and 20010.



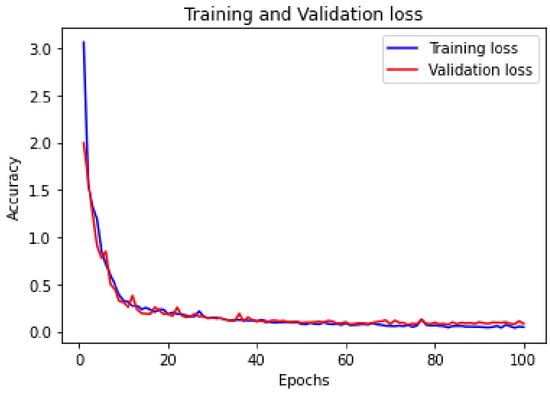
(a) CNN without weights B/W overfit



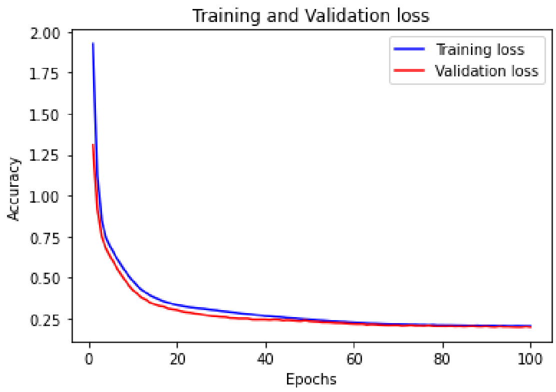
(b) CNN without weights color no overfit



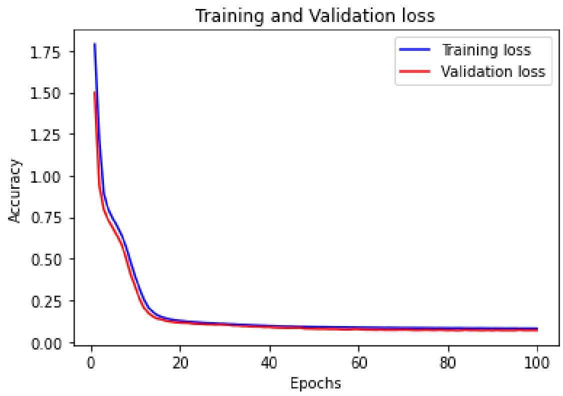
(c) CNN with weights B/W overfit



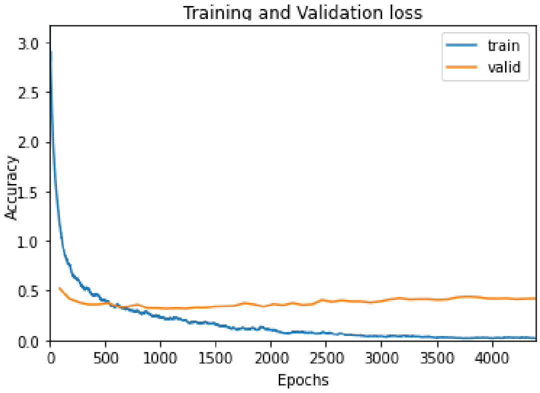
(d) CNN with weights color no overfit



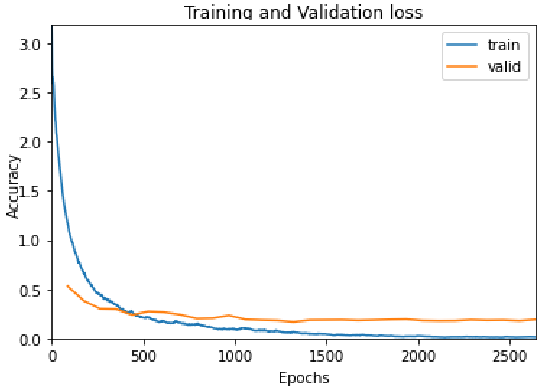
(e) Ensemble B/W no overfit



(f) Ensemble color no overfit



(g) ResNET B/W overfit



(h) ResNET color overfit

Figure 7: Training and Validation Loss Graphs: (a), (c), (g), and (h) show the overfitting as there is a significant difference between the train and the valid curves. In (b), (d), (e), and (f), there is no overfitting as the two lines are almost overlapping showing very minimal or no differences between train and valid results.

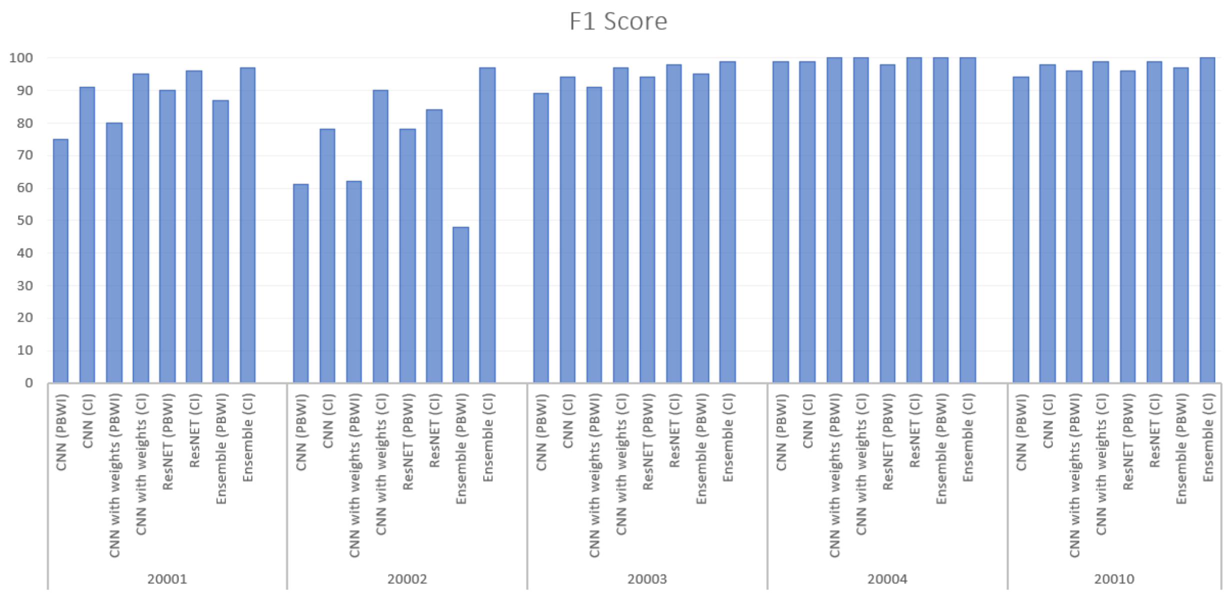


Figure 8: F1 score: It shows the F1 score for all five meat cuts with different models on both the pre-processed black and white (PBWI) and the colored images (CI).