International Journal of Membrane Science and Technology, 2023, Vol. 10, No. 1, pp 363-376

AI-Powered Vision Inspection System for Object Classification Application

Jing Song Ong[1], J. Hossen[2*], Poh Ping Em[3], Thangavel Bhuvaneswari[4], J. Emerson Raja[5], Min Thu Soe[6], Nor Hidayati[7], Gin Chong Lee[8], Sajid Abdullah Alam[9]

[1,2,3,4,5,6,7,9]*Faculty of Engineering and Technology, Multimedia University, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia; E-mail: jakir.hossen@mmu.eu.my*

[8]*Department of Engineering,School of Engineering, Computing and Built Environment, UOW Malaysia Penang University College, 32, Jalan Anson, George Town, 10400 George Town, Pulau Pinang.*

Abstracts: Human operators are often susceptible to eye fatigue due to sleep deprivation and excessive workload, which may negatively impact their consistency and efficiency in performing repetitive and challenging inspection tasks. This paper presents the development of an AI-powered vision inspection system for object sorting applications, utilizing a You-Only-Look-Once (YOLO) version 3 pre-trained model based on Deep Convolutional Neural Network's (DCNN) transfer learning technique. Feature extraction for each data point is performed using Darknet53, which subsequently trains the YOLO v3 model. The dataset is partitioned into a training set and test set at a 90:10 ratio. The trained model achieves a mean average precision (mAP) of 99.146%. Enhancing the precision and recall values of the model can be accomplished by increasing the number of dataset instances used for training.

Keywords: Transfer Learning, You Only Look Once, YOLOv3, Darknet53, Object Detection, Palm Oil Kernel Classification.

1. INTRODUCTION

An "object sorting system" is a method used to systematically label and classify objects based on specific standards and characteristics. Cost optimization, as well as improvements to efficiency and the quality of the final product, are crucial aspects of any production and manufacturing process. This study incorporates both deep learning techniques and more traditional approaches. After extensive research into deep learning methodologies, the AI-Powered vision inspection system described in this article was designed and built. Since the system needs to identify and localize objects within the input images, the approach employed must be related to object detection in some manner. Convolutional neural network techniques for object detection primarily rely on one-stage and two-stage algorithms. However, the main focus of this paper will be on the You Only Look Once algorithm of version 3 (YOLOv3). This literature review will explore the transfer learning technique using the YOLO pre-trained model. By leveraging a pre-trained model, the transfer learning technique allows for improved performance and reduced training time, as it takes advantage of the knowledge gained from solving related tasks.

The YOLOv3 algorithm has attracted significant attention for its noteworthy performance in terms of both speed and accuracy. This characteristic makes it a highly suitable option for real-time object detection and sorting applications, as well as for integration into AI-powered vision inspection systems.

2. LITERATURE REVIEW

2.1. Existing Object Sorting System

The object sorting system has evolved from manual sorting to automatic sorting, significantly improving efficiency. The object sorting system was initially built using shallow learning techniques. For object identification and posture estimation, K. Rahardja [1] and Ata AA [2] employed basic features of complex objects, edge detection technologies, structured light principles, and color attributes, respectively. Machine vision recognition is typically carried out in a multi-step process involving image acquisition, image pre-processing, image segmentation, feature extraction, and recognition classification [3].

2.2. Object Detection

The goal of object detection is to locate and identify individual objects within each image. Unlike image classification, object detection can recognize multiple objects in a single input image. The object detection models are primarily based on two frameworks: the first, which uses two-stage algorithms, classifies each proposal and region proposals into multiple object categories; the second framework presents object detection as a classification or regression problem using single-stage algorithms [4]. This study seeks to leverage CNN models to achieve accurate identification of individual items in an input image and their precise location within said image.

2.3. Convolutional Neutral Network (CNN) Architecture

A convolutional neural network typically consists of several hidden layers in addition to an input layer and an output layer. These hidden layers in CNN [5] are supervised shallow neural networks composed of a series of max-pooling layers, convolutional layers, and so on. These layers are used for feature learning and following them is a fully connected network employed for classification. All these components together form the inputs and outputs, along with the final convolution, are the reasons they arereferred to as hidden layers. During the training phase, CNN can learn comprehensive and advanced feature representations of images. It is widely used because it eliminates complex image pre-processing and allowsfor direct input of original images.

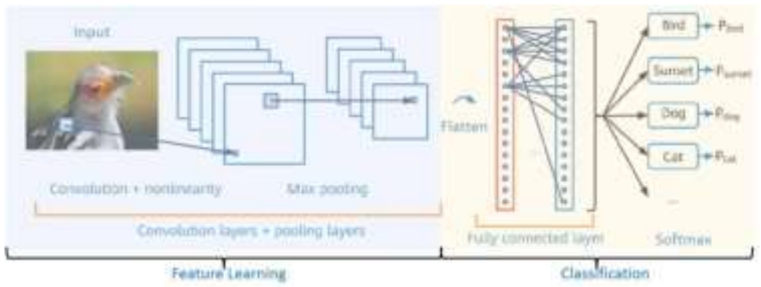


Fig 1: Convolutional Neural Network (CNN) Architecture: Feature Learning and Classification Stages [6]

The first CNN used for recognition was called LeNet5, and it was developed and used for digit recognition [7]. Since then, various CNN architectures have emerged. Other CNN-based object recognition architectures include AlexNet [8], VggNet [9], ResNet [10], and many others.

2.4. You Only Look Once (YOLO)

The single-shot object detection method called "YOLO" was proposed by Redmon.J in 2016 [11]. This method can be used in real-time and is extremely fast. YOLO's core concept is to divide the input image into equally sized grids (s x s). For each grid cell, the model predicts a class and a bounding box forall possible objects present in the grid. The grid cell is responsible for object detection, and it is included ifan object's center or midpoint lies within it.

The number of bounding boxes each grid cell predicts is B, and each box is assigned a confidence score. The confidence score indicates the box's ability to predict the object's boundaries and reveals the model's certainty that the box contains an object. If an object is present in the image, the confidence score is calculated based on the intersection over union (IoU) between the actual image and the predicted box.

Each grid cell also predicts C conditional class probabilities. These probabilities change depending on the presence or absence of an object in a specific grid cell. Each grid cell will only have a single set of predictedclass probabilities, regardless of the total number of boxes B.

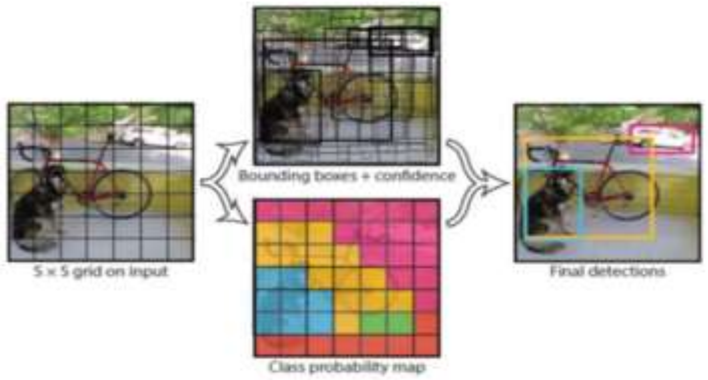


Fig 2: The YOLOv3 Model: Treating Object Detection as a Regression Problem by Dividing Images into S x S Grids and Predicting B Bounding Boxes, Confidence Scores, and C Class Probabilities, Encoded as an S x S x (B *5 + C) Tensor [10].

The Darknet-53 (D53) architecture serves as the basis for the feature extractor used in YOLO v3. D53 comprises 53 layers of convolution. Since D53 has a higher number of convolutional layers than Darknet-19, D53 operates at a slower speed. The performance of D53 is superior to that of ResNet-101, asit is both higher and 1.5 times faster. Both D53 and ResNet-152 offer comparable performance, but D53 does so at double the speed of ResNet-152 [12].

YOLOv3 is a fully convolutional network consisting exclusively of 1x1 and 3x3 size convolution filters. It analyzes data quickly and with an impressive degree of accuracy for objects of any size. YOLOv3'snetwork stride reduces the size of the images by a factor of 32. The initial YOLO version only accepted images with a resolution of 448 x 448 pixels, resulting in an output feature map size of 14 by 14.

In YOLO v3, speed has been sacrificed for increased accuracy due to the growing complexity of the underlying architecture known as Darknet [13].

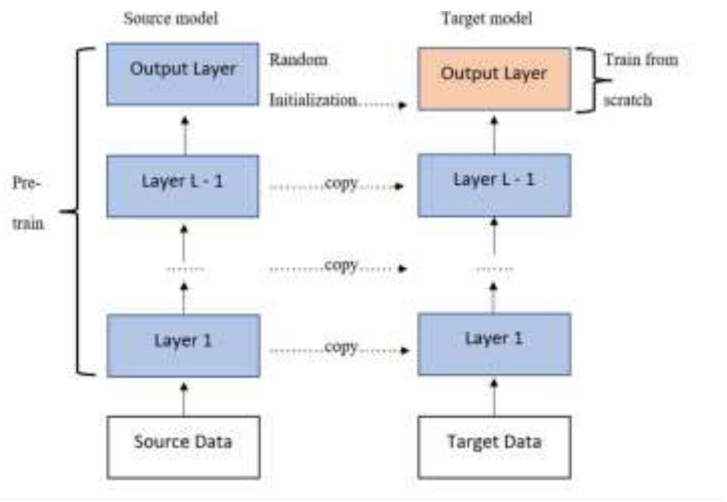
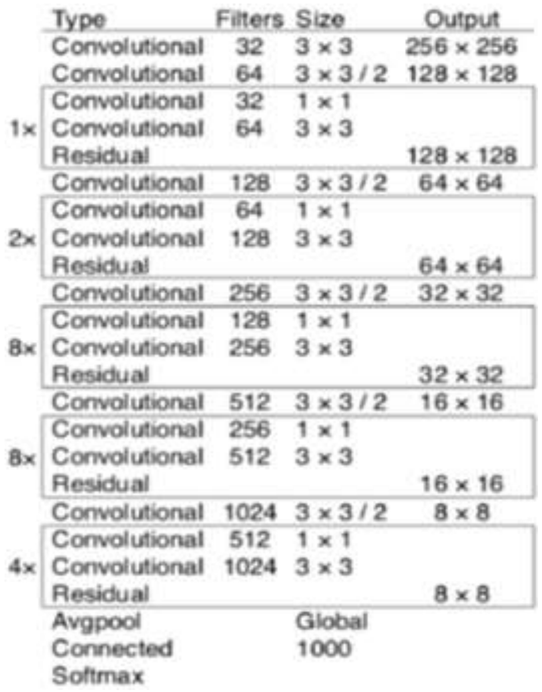


Fig 3: The Darknet-53 Architecture [12].

As shown in Figure 4, the overall performance of YOLO v3 is superior to other approaches when measured with mAP set at a threshold of 50% and the inference time. Compared to other models, YOLO v3's inference time is much more efficient.

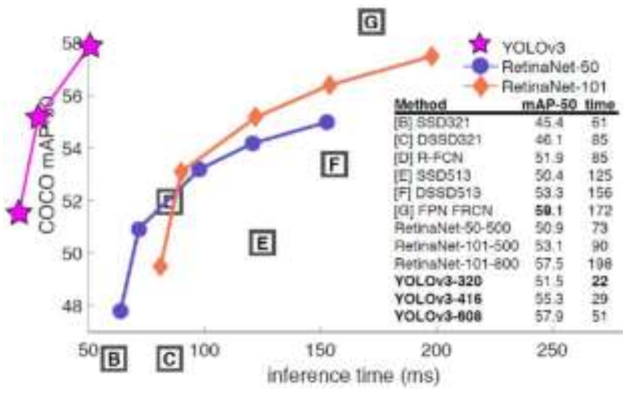


	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig 4: COCO Dataset Testing with mAP@0.5 [12]

2.5. Transfer learning and Pre-trained Model

Transfer learning is the process of leveraging knowledge gained from one task and applying it to address other similar tasks. This approach helps overcome the isolated learning paradigm of traditionally developed conventional machine learning and deep learning algorithms. Pre-trained models are used as feature extractors in popular deep-transfer learning algorithms. The idea is to use one or more layers of an existing deep neural model, excluding the last layer, for a different but related task as a generic feature extractor. Consequently, the newly adapted model will only require shallow training, reducing computation time and resources. During backpropagation, the weighted layers of the pre-trained model will not be updated with the unseen data. It is assumed that the tasks are different, but within the same domain.



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

Fig 5: Utilizing Transfer Learning with Pre-trained Models [3].

2.6. K-Fold Cross Validation

A reliable estimate of the model error can be achieved by performing additional tests with k-fold cross-validation [14]. This approach is particularly useful when there are insufficient data points for hold- out cross-validation. The available dataset for learning is divided into k roughly equal-sized subgroups, where "fold" refers to the total number of derived subgroups. In this particular partitioning, the instances that constitute the learning set are selected at random. The k - 1 subsets that form the training set are combined to create the complete training set. The model is then applied to analyze the final subset, also known as the validation set, and its performance is evaluated. This method is repeated to ensure that no two test sets overlap, and it continues until each of the k subsets has served as a validation set. The cross-validated performance is equal to the average of the k different performance measurements collected from the k different validation sets.

K-fold cross-validation is one technique that can be employed to enhance the generalization capabilities of machine learning models. A training dataset is divided into K folds (K = 5), each of similar size. One fold is randomly selected as the validation set, while the remaining folds are used to train the model. In the K-fold cross-validation method, each piece of data has the opportunity to participate in the learning process.

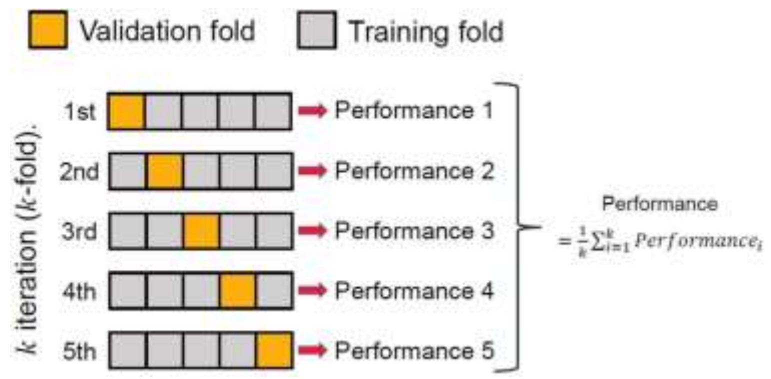


Fig 6: Example of 5-Fold Cross-Validation with k = 5 [6].

To assess the precision accuracy of the models, the correlation can be formulated into two performancemetrics, which are mean square error (MSE) and root mean square error (RMSE).

Mean square error,

$$MSE = \frac{1}{K}\sum_{i-1}^{k} Performance\ i \tag{1}$$

And

Root mean square error,

$$RMSE = \sqrt{\frac{\sum_{i-1}^{N} ||y(i) - \hat{y}(i)||^2}{N}} \tag{2}$$

3. METHODOLOGY

The methodology employed for this work begins with data acquisition. In this context, dealing withimages implies that each data item can be interpreted as an image. Upon completing the dataset collectionproceed to the next step in our process, which involves preliminary tasks such as labelling objects in the data with their respective classes. This is necessary to convert the dataset into the appropriate format for YOLOv3, enabling us to initiate the training process effectively. Once the trained model is developed, it istested using the test dataset, and finally, the YOLOv3 performance is evaluated using specific techniques. The flow is shown through the figure 7.

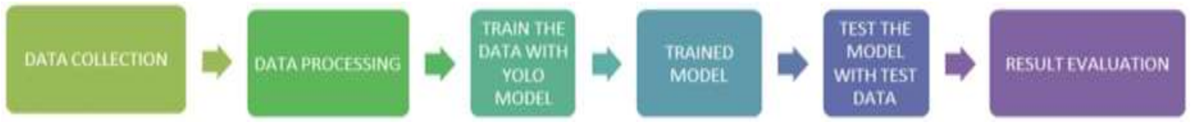


Fig 7: Methodological Approach Employed in the Project

3.1. Data Set

Data is gathered from the plantation farm for the three different stages of palm oil kernel ripeness: unripe, ripe, and over-ripe. To enhance the dataset, data augmentation techniques were applied, which involved making adjustments to the images, such as changing brightness, rotation, flipping, and other operations.

A total of 200 datasets were self-collected from the plantation farm using an RGB sensor, with each image captured at a resolution of 1280 x 720 pixels. The dataset comprised of 70 underripe samples, 70 over-ripe samples, and 60 ripe samples. Before any further processing, data augmentation was performed on the dataset to expand its size and variability.

For the training process, 90 percent of the dataset, including the augmented samples, was used. The remaining 10 percent of the dataset was reserved for performance testing to evaluate the model's effectiveness in detecting palm oil kernel ripeness.

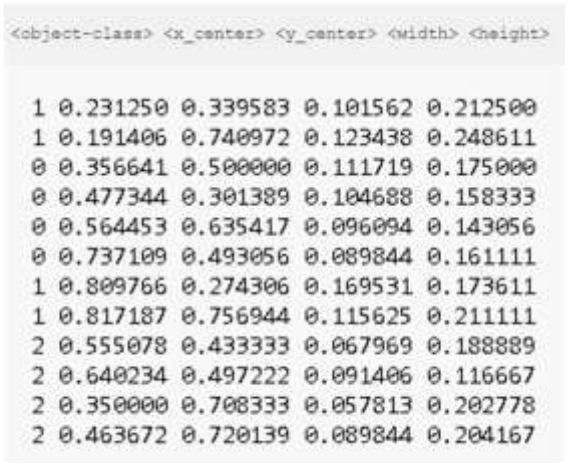
3.2. Labelling

The process of data annotation is essential for assigning descriptive labels to the dataset. In this case, each image was annotated using a tool called LabelImg, and the annotations were saved in a text-format file along with the corresponding images. Data annotation plays a critical role in providing meaningful labels that accurately represent the objects or features present in the dataset.

By annotating the dataset, it becomes possible to train models for object detection, as the annotations provide the necessary information for the model to learn and identify specific objects or regions of interest. Accurate and comprehensive annotations contribute to the quality and effectiveness of the training process, enabling the model to detect and recognize objects with greater precision.

The data annotation process ensures that the dataset is properly labeled, allowing the model to learn from the annotated examples and generalize its understanding to new, unseen data. It establishes a link between the visual information in the images and the corresponding labels, facilitating the training process and enhancing the accuracy of object detection tasks.

In summary, data annotation is a crucial step in preparing a dataset for training models. By accurately assigning labels to the images, the annotation process enables meaningful training and contributes to the effectiveness of object detection systems. The annotation process will save like Figure 8 in the text file.



```
<object-class> <x_center> <y_center> <width> <height>

1 0.231250 0.339583 0.101562 0.212500
1 0.191406 0.740972 0.123438 0.248611
0 0.356641 0.500000 0.111719 0.175000
0 0.477344 0.301389 0.104688 0.158333
0 0.564453 0.635417 0.096094 0.143056
0 0.737109 0.493056 0.089844 0.161111
1 0.809766 0.274306 0.169531 0.173611
1 0.817187 0.756944 0.115625 0.211111
2 0.555078 0.433333 0.067969 0.188889
2 0.640234 0.497222 0.091406 0.116667
2 0.350000 0.708333 0.057813 0.202778
2 0.463672 0.720139 0.089844 0.204167
```

Fig 8: Annotation from Text File using Labelling

3.3. Training

In addition to the primary training process, it is essential to generate two supplementary files to trainthe YOLO algorithm: the names file and the data file. This results in the creation of the classes.name file and the Datav3.data file for the data. The classes.name file enumerates the classes intended for training, while the Datav3.data file contains paths to the train.txt file, test.txt file, and classes.name file, as well as the backup file where the trained model's weights are stored. Once the preparation is complete, the training process can commence. Upon completion of the training process, the trained weights must be forwarded to the object detection algorithms for object detection.

Additionally, a configuration file must be made. It must include a number of parameters that are applied to the YOLOv4. Batch size (64) and subdivisions (16), which regulate how the batch is split for GPU memory efficiency, are two general training process parameters found in the first part. Other settingsinclude momentum, decay, and different colour augmentation factors including saturation, exposure, and hue. Image dimensions (width and height) set to 416x416 pixels, the number of channels (3 for RGB photos),and momentum and decay are also included.

The burn-in parameter specifies the number of early training cycles when the learning rate is progressively raised. The learning rate is set at 0.0013. The training strategy is stated as steps, and the maximum number of training batches is 6000. This means that during training, the learning rate will vary at certain stages (4800 and 5400). The scale variation for each step is defined by the scales parameter.

The mosaic option is set to 1, allowing the mosaic data augmentation method to be used. Mosaic creates a training image from many different training images, increasing the variety of training samples. Following that, a number of convolutional layers with various parameters are defined by the setup. The parameters for each convolutional layer include batch normalisation, the number of filters, the size of the filters, the stride, the padding, and the activation function (mish or leaky).

The "route" layers specify skip connections, allowing the network to use feature maps from earlier layers to enhance the detection process. The "layers" parameter in the route layer indicates which previous layers to concatenate.

Downsample layers are used to reduce the spatial dimensions of the feature maps. They are implemented using convolutional layers with a stride of 2.The configuration also includes the concept of shortcut connections, which are used to bypass several convolutional layers and merge with a later layer. These shortcuts aid in gradient flow and improve the overall performance of the network.

The SPP (Spatial Pyramid Pooling) section introduces a spatial pyramid pooling layer that captures multi-scale information by applying max-pooling at multiple sizes. The SPP layers are followed by convolutional layers with batch normalization and leaky activation.

The subsequent section includes an upsampling layer, which increases the spatial dimensions of the feature maps by a factor of 2. The "route" layers are then used to concatenate feature maps from previous layers.

The remaining layers consist of convolutional layers with different configurations, including batch normalization and leaky activation. The network architecture ends with the final detection layer, which produces bounding box predictions for object detection.

In summary, the provided configuration file outlines the architecture and settings for training a YOLOv4 model, specifying various convolutional layers, skip connections, downsampling, upsampling, and detection layers.

With the configuration file parameters, the darknet53 serve it purposefully. Darknet-53 is a deep neural network architecture that serves as the backbone of the YOLO (You Only Look Once) object detection framework. It is specifically designed to extract meaningful features from input images, enabling accurate and efficient object detection.

The architecture consists of 53 convolutional layers that progressively analyze the input image at different levels of abstraction. Each convolutional layer applies filters to the input data, capturing specific visual patterns such as edges, corners, and textures. By stacking multiple convolutional layers, Darknet-53 can learn increasingly complex features, which are crucial for detecting objects of different shapes and sizes. One notable aspect of Darknet-53 is the use of residual connections, or shortcut connections, which alleviate the vanishing gradient problem. These connections allow the network to bypass certain convolutional layers and directly link earlier layers with later ones. By doing so, the network can propagate gradients more effectively during training, leading to better feature learning and model performance.

In addition to convolutional layers and residual connections, Darknet-53 incorporates downsampling and upsampling operations. Downsampling is performed using convolutional layers with a stride of 2, reducing the spatial dimensions of the feature maps while increasing the number of channels. This downsampling helps capture high-level semantic information from larger receptive fields. Upsampling,on the other hand, is achieved through interpolation techniques that increase the spatial resolution of the feature maps. Upsampling enables the network to recover spatial details and improve localization accuracy.

Darknet-53 also employs route layers, which concatenate feature maps from previous layers. Theseskip connections allow the network to access features from different scales and resolutions, enabling the detection of objects at various levels of detail. By fusing information from multiple stages of the network, Darknet-53 can capture both fine-grained local features and global context, enhancing its ability to detect objects accurately.

Towards the end of the architecture, Darknet-53 typically includes additional convolutional layers followed by fully connected layers or a global average pooling layer. These final layers reduce the spatial dimensions of the feature maps and produce a compact feature representation that summarizes the image information relevant for object detection.

Overall, Darknet-53 plays a crucial role in the YOLO object detection framework by extracting rich and discriminative features from input images. These features are then used by subsequent layers for object detection, bounding box prediction, and class classification, enabling YOLO models to detect objectsin real-time with remarkable speed and accuracy.

The lower layers of the pre-trained model for YOLOv3, which capture low-level features and general patterns, are kept frozen or unaltered during finetuning. This is because these layers have alreadylearned useful representations from a large dataset, such as ImageNet, and can provide valuable features for various tasks.

The later layers of the pre-trained model, which are more specific to the original task the modelwas trained on, are typically modified or replaced during finetuning. These layers are responsible for capturing more task-specific features and patterns. By adjusting these layers, the model can adapt to thenew task and learn task-specific representations.

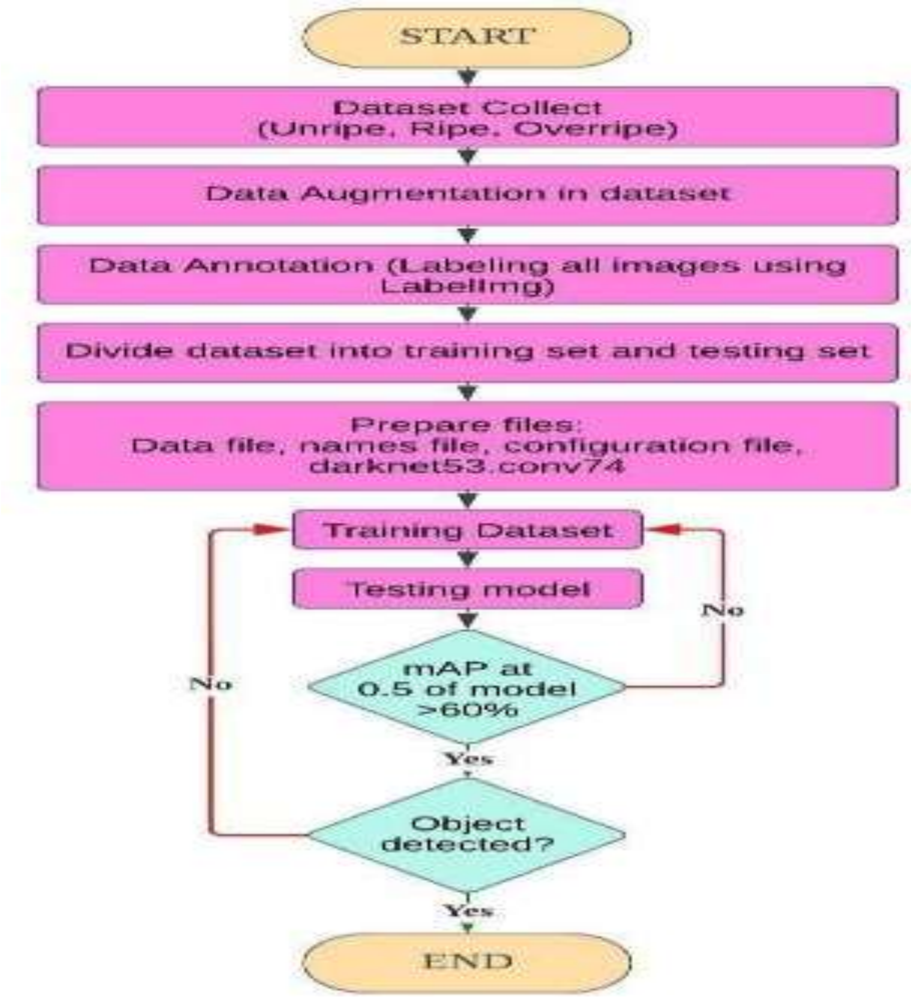

START
Dataset Collect (Unripe, Ripe, Overripe)
Data Augmentation in dataset
Data Annotation (Labeling all images using LabelImg)
Divide dataset into training set and testing set
Prepare files: Data file, names file, configuration file, darknet53.conv74
Training Dataset
Testing model
No
mAP at 0.5 of model >60%
No
Yes
Object detected?
Yes
END

Fig 9: Process Flow Diagram of Training and Testing YOLO v3 model.

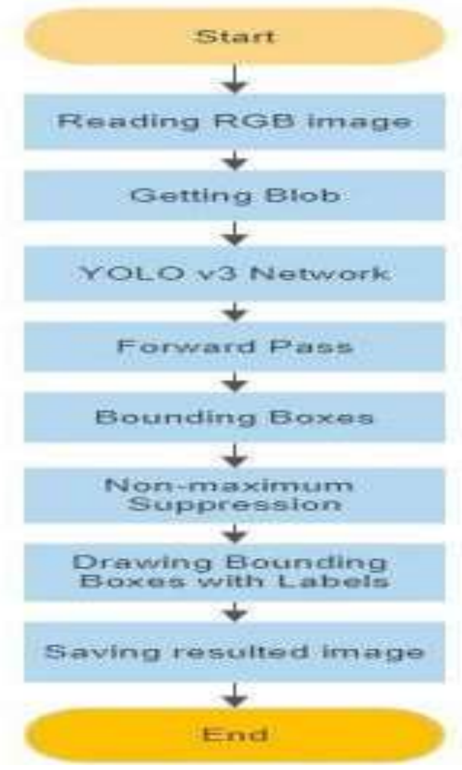


Fig 10: Flow Chart of Object Detection Algorithm

3.4. Performance Evaluation of The Model

Measurements of Recall (R), Precision (P), and Mean Average Precision (mAP) are obtained using the model with trained weights. While mAP is calculated by applying various Intersection over Union (IoU) thresholds, P and R are determined using different confidence levels. Recall (R) represents the proportion of all possible positive outcomes that can be detected, whereas Precision (P) signifies the percentage of accurate predictions. Both P and R are influenced by the values of True Positives (TP), False Positives (FP), and False Negatives (FN). Equations 3 and 4 display the formulas used to compute R and P, respectively:

$$Precision, P = \frac{True\ Positive, TP}{True\ Positive, TP + False\ Positive, FP} \quad (3)$$

$$Recall, R = \frac{True\ Positive, TP}{True\ Positive, TP + False\ Negative, FN} \quad (4)$$

Equation 5 presents the method for calculating Average Precision (AP), which is the mean precision for a set of eleven recall values ranging from 0 to 1, incremented by a step size of 0.1. In this formula, Pr denotes the precision at specific recall values:

$$Average\ Precision, AP = \frac{1}{11}\sum_{r\in(0.0.....1.0)} AP_r \quad (5)$$

Finally, Mean Average Precision (mAP) is the average of AP, as shown in Equation 6:

$$Mean\ Average\ Precision,\ \ mAP = \frac{1}{N}\sum_{i=1}^{N} Average\ Precision, AP_i \qquad (6)$$

4. Experimental Result

The training process generates a chart illustrating the average loss versus iteration to monitor the model's performance which has compiled into Figure 11. In the provided visualizations, the red line represents the mAP, while the blue line corresponds to the loss. The mAP is calculated after the first 1000iterations. Precision (P) and recall (R) are analyzed based on the testing dataset, which consists of 20 distinctimages extracted from the dataset. Furthermore, Table 1 presents the P and R values for all three ripeness stages: unripe, ripe, and overripe. These P and R values serve as indicators of the model's accuracy.

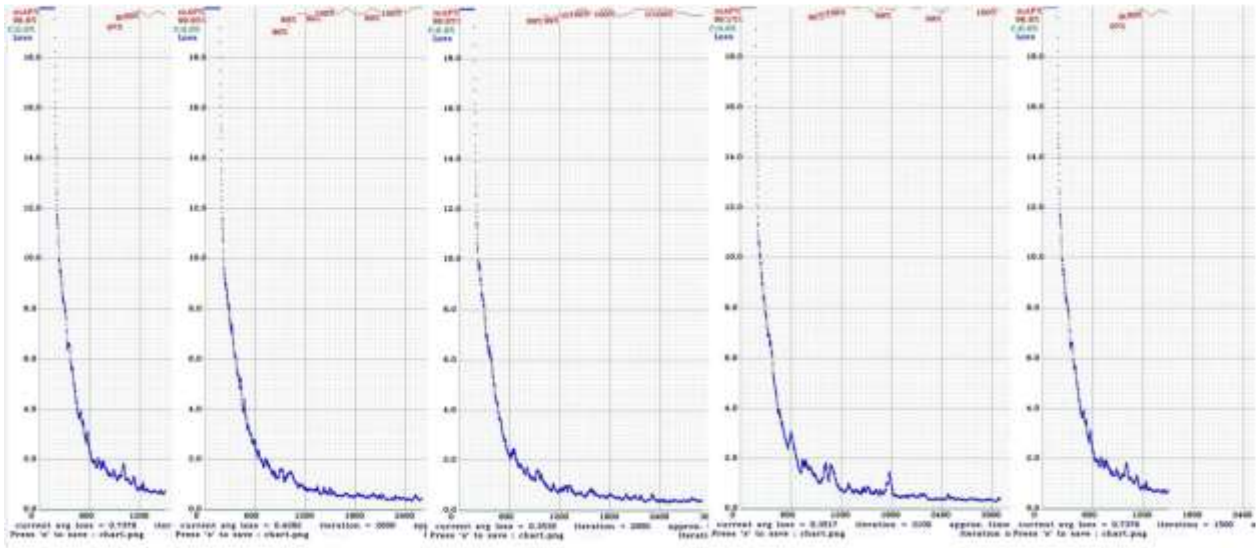
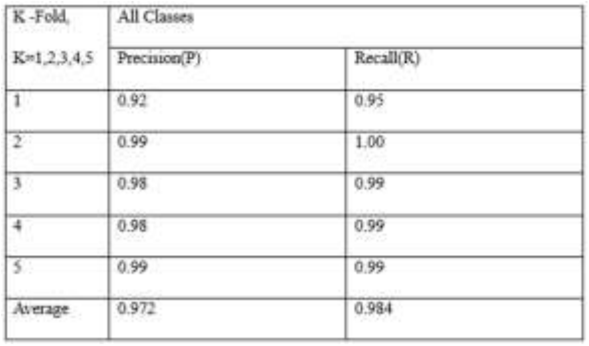


Fig 11: Illustration for Model Performance.

Figure 12 reveals an inverse relationship between Recall (R) and precision for the trained model, indicating that when the proportion of accurate predictions (P) is low, the Recall (R) will be high, capturinga greater proportion of all possible positive outcomes.

The connection between mAP (or AP) and the IoU threshold is investigated, with Table 2 presenting the corresponding results. The IoU performs an analysis to ascertain the dependability of a detected object within a dataset. Figure 13 visually represents the concept and formula of the IoU. AP denotes average precision, while mAP represents the overall mean of the AP values.



K -Fold,	All Classes	
K=1,2,3,4,5	Precision(P)	Recall(R)
1	0.92	0.95
2	0.99	1.00
3	0.98	0.99
4	0.98	0.99
5	0.99	0.99
Average	0.972	0.984

Table 1: Precision and Recall of the Model

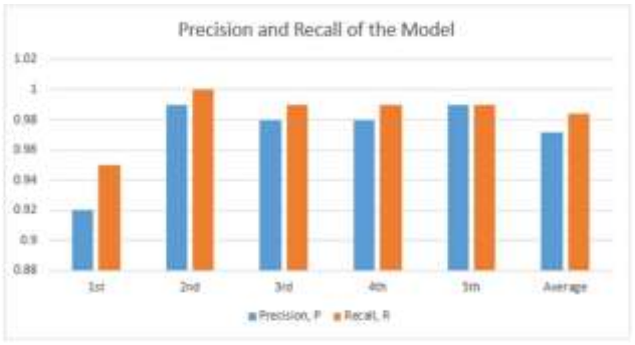
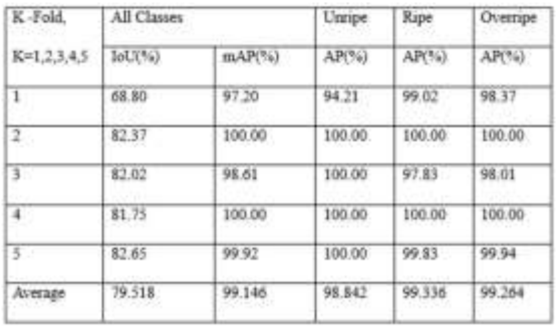


Fig 12: Precision and Recall Chart of the Model

Table 2: Average Precision (AP), Mean Average Precision (mAP) and Intersection over Union Threshold (IoU) of the Model



K -Fold,	All Classes		Unripe	Ripe	Overripe
K=1,2,3,4,5	IoU(%)	mAP(%)	AP(%)	AP(%)	AP(%)
1	68.80	97.20	94.21	99.02	98.37
2	82.37	100.00	100.00	100.00	100.00
3	82.02	98.61	100.00	97.83	98.01
4	81.75	100.00	100.00	100.00	100.00
5	82.65	99.92	100.00	99.83	99.94
Average	79.518	99.146	98.842	99.336	99.264

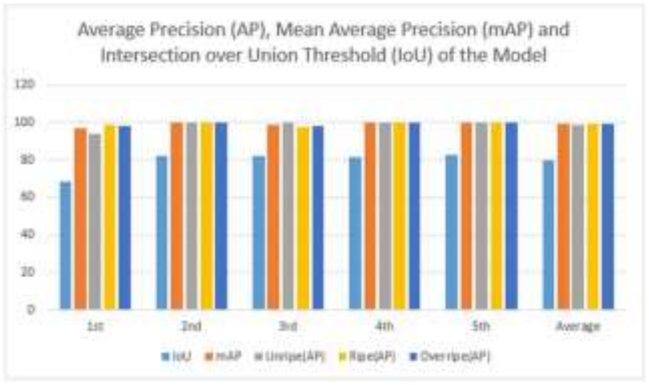


Fig 13: Average Precision (AP), Mean Average Precision (mAP) and Intersection over Union Threshold (IoU) of theModel Chart

The chart demonstrates that although the Intersection Over Union (IoU) threshold is low, after applying certain calculations, the mean average precision (mAP) can still be high at the end of the process.Thus, it can be inferred that the AP is inversely proportional to the IoU. The chart also displays the AveragePrecision (AP) for each class: unripe, ripe, and overripe.

All five folds were accurately categorized, achieving an average of 0.972 correct classifications. Themodels' weights and biases are subsequently applied to the remaining test set, which is then used for evaluation. The remaining test set is reserved for testing purposes. The accuracy of this dataset is determinedto be 0.99.

4.1. K-Fold Cross Validation

The trained model effectively detects all classes, including unripe, ripe, and overripe palm oil fruit, with high accuracy demonstrated by the confidence score and the location of the fruit in the image being encompassed by a bounding box. This AI-powered vision inspection system verifies the successful detection of unripe, ripe, and overripe palm oil fruit.

In this study, the findings are compared to previous research employing an Artificial Neural Network (ANN) to determine the maturity level of palm oil fruit. According to the cited work [14], the data collection procedure closely aligns with the methodology implemented in the present study. The research documented in reference [14] has demonstrated successful classification.

However, the methodology proposed in this study, utilizing a deep learning approach to categorize palm oil kernels, yields promising results due to the employment of YOLO v3 and the Darknet 53 feature extractor for feature extraction. This indicates that using YOLO v3 as a CNN technique for inspection tasksoutperforms the use of an artificial neural network (ANN).

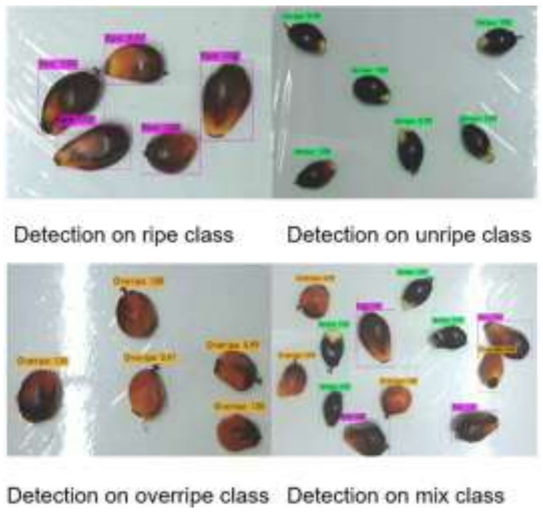


Fig 14: Live Application Showcase - Summary of Palm Oil Kernel Detection Results

Table 4.4 Results comparison with other Researcher

Researcher	Result	Method	Features
[14]	95.48%	Boosting ANN	Raman peaks
This study	99.15%	CNN	Darknet 53

CONCLUSION

The efficacy of the proposed models was evaluated through transfer learning, considering metrics such as recall, precision, mean average precision (mAP), and intersection over union (IoU). The model achieved a recall of 0.984 and a confidence score of 0.98. Moreover, a mAP score of 99.146% and an IoU score of 79.518% were attained, indicating that the developed model exhibits high efficiency and performance for detecting and classifying palm oil kernels in industrial applications. Future research shouldfocus on enhancing the precision and accuracy of the AI-powered vision inspection system by collecting a substantial amount of data and accurately labeling class annotations prior to training the deep learningneural network.

ACKNOWLEDGEMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

[1] K. Rahardja and A. Kosaka, "Vision-based bin-picking: recognition and localization of multiple complex objects using simple visual cues", Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96. Available: 10.1109/iros.1996.569005.

[2] A. Ata, A. Rafeek and H. Yusof, "Sensory-Based Colour Sorting Automated Robotic Cell", Academia.edu, 2022. [Online]. Available: https://www.academia.edu/21204681/Sensory_Based_Colour_Sorting_Automated_Robotic_Cell.

[3] A. Sarda, S. Dixit and A. Bhan, "Object Detection for Autonomous Driving using YOLO [You Only Look Once] algorithm", 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), 2021. Available: 10.1109/icicv50876.2021.9388577.

[4] Zhao, Z.Q.; Zheng, P.; Xu, S.T.;Wu, X. Object Detection with Deep Learning: A Review. IEEE Trans. Neural Netw. Learn. Syst. 2019,30, 3212–3232.

[5] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS ’12), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.

[6] Gin. Chong. Lee, “Artificial Intelligence and Applications”, Faculty of Engineering and Technology, Melaka Campus, 2022.

[7] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] Jam, F. A., Rauf, A. S., Husnain, I., Bilal, H. Z., Yasir, A., & Mashood, M. (2014). Identify factors affecting the management of political behavior among bank staff. African Journal of Business Management, 5(23), 9896-9904.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Adv. Neural Inf. Process. Syst., 2012.

[10] K. Simonyan and A. Zisserman, “VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGESCALE IMAGE RECOGNITION,” 2015.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.

[12] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. Proc. IEEE-CVPR 2016:779–88.

[13] Redmon J, Farhadi A. Yolov3: an incremental improvement. arXiv preprint arXiv: 1804.02767 2018.

[14] Z. Dahirou and M. Zheng, "Motion Detection and Object Detection: Yolo (You Only Look Once)", 2021 7th Annual International Conference on Network and Information Systems for Computers (ICNISC), 2021. Available: 10.1109/icnisc54316.2021.00053.

[15] Bonaccorso, G. (2018). In Machine learning algorithms: Popular algorithms for Data Science and Machine Learning (pp. 404–404). essay, Packt Publishing.

[16] Hong, T. S., Hanim Hashim, F., Raj, T., &amp; Huddin, A. B. (2021). Classification of oil palm fruit ripeness using artificial neural network. 2021 IEEE International Conference on Automatic Control &amp; Intelligent Systems (I2CACIS).

DOI: https://doi.org/10.15379/ijmst.v10i1.1817