IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 October 2023, accepted 5 November 2023, date of publication 13 November 2023,
date of current version 17 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3332215

RESEARCH ARTICLE

Classification of Arecanut X-Ray Images for Quality Assessment Using Adaptive Genetic Algorithm and Deep Learning

PRAVEEN M. NAIK AND BHAWANA RUDRA, (Member, IEEE)
Department of Information Technology, National Institute of Technology Karnataka, Surathkal 575025, India
Corresponding author: Praveen M. Naik (praveen.207it001@nitk.edu.in)

ABSTRACT The traditional approach for analyzing the quality of arecanuts is based on their external appearance. However, using machine learning and deep learning techniques, automated classifications were performed. But the true quality can only be analyzed when the internal structure of the arecanut is examined. Therefore, we use the X-ray imaging technique to determine the internal quality of arecanuts. We prepared a novel dataset of arecanut X-ray images and used a YOLOv5 based deep learning architecture for classification. The present study employs an adaptive genetic algorithm based approach for hyperparameter optimization to enhance the mean average precision (mAP) using a light weight model generated using a ghost network and a feature pyramid network (FPN). We have achieved the highest mAP of 97.84% using our method with a lower model size of 15 MB. Our method has excelled in detecting the arecanut compared to cutting-edge object detection algorithms such as YOLOv3, YOLOv4, Detetron, YOLOv6, YOLOv8, and YOLOX. We also acknowledged the performance enhancement using the adaptive genetic algorithm on the Pascal VOC 2007 image dataset. Despite of significant computational requirements for executing genetic algorithms, we proved that genetic algorithms can boost mAP. Additionally, the methodology developed in this investigation produced multiple models with the best mAP featuring optimized hyperparameters. This methodical strategy is helpful for the design of an automatic, non-destructive, integrated X-ray image based classification system. This system has the potential to revolutionize the quality assessment of arecanuts by offering a more efficient evaluation method.

INDEX TERMS Agriculture, arecanut, deep learning, hyperparameter optimization, nondestructive, X-ray, classification.

I. INTRODUCTION

Arecanut, also known as betel nut or areca catechu, has healing properties that have made it a staple in traditional medicine for centuries. Minerals and vitamins found in abundance in the arecanut include the essential minerals magnesium, potassium, and iron. Arecanut has been used to cure a wide variety of medical conditions, including toothaches, headaches, and stomach problems. The nut's anti-inflammatory and antioxidant properties have also been shown, suggesting it may serve as a remedy for a variety of diseases. High-quality arecanuts are valued for their flavour, aroma, and texture when used as an ingredient or stimulant in food and beverages. A higher market price means more financial gain for producers, suppliers, and merchants. Additionally, the quality of the arecanut impacts the amount of vitamins and minerals it contains. The health benefits and cultural significance of eating arecanuts depend on their quality.

So far, various researchers have proposed deep learning, machine learning, or computer vision methods for grading or classifying arecanuts based on their external appearance [1], [2], [3], [4], [5], [6], [7], [8], [9]. These methods cannot guarantee the true quality of arecanut unless an

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

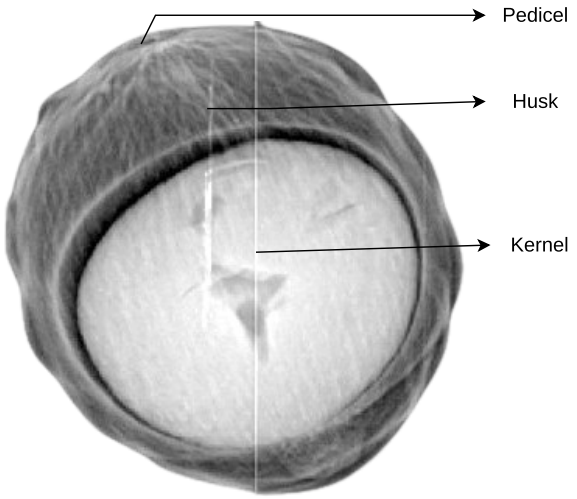


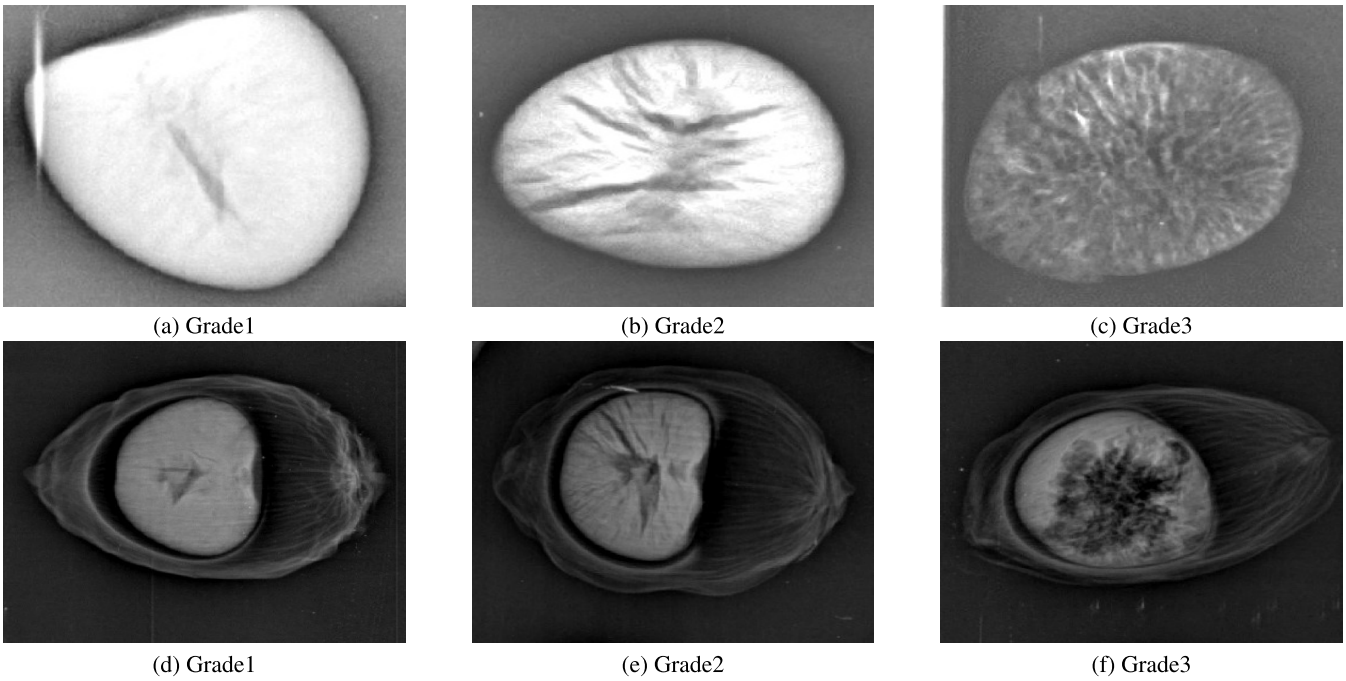
FIGURE 1. Representation of an X-ray image depicting the internal parts of a Grade1 arecanut.

internal examination is performed. The true quality check is determined when an arecanut kernel is broken and internal examination is performed. But the broken arecanut cannot be shelved for longer duration. So, an alternate method is required to analyse the quality of an arecanut using non-destructive approach. Computed tomography (CT), magnetic resonance imaging (MRI), X-ray imaging, infrared thermography, RGB imaging, hyper-spectral imaging, visible and near infrared spectroscopy, Raman spectroscopy, and infrared spectroscopy are some examples of nondestructive approaches that have been recommended for analyzing the quality features of different crops [10], [11], [12], [13]. It has been shown that quite a few of the methods described above are capable of accurately forecasting product quality by analyzing surface characteristics such as shape, size, and flaws. However, the MRI and X-ray imaging techniques are the only ones that can be used to verify the internal properties of an arecanut. However, because of the physical restrictions and the high cost associated with MRI machines, X-ray based approaches seem to have the greatest promise. This is especially the case because of the higher acquisition speed that X-ray techniques provide.

Various studies on X-ray images of crops and seeds for quality examination and classification were performed. Casasent et al. used a neural network to classify pista using the piece-wise quadratic neural network (PQNN) method. With an internal examination, the method was able to classify pistachios into three groups [14]. Irwin Ronaldo Donis-Gonzalez et al. used CT images of chestnuts to examine the internal rot. They classified chestnuts into 5 classes, with class 1 being best and class 5 being completely rotted nuts. The classification was carried out using a quadratic discriminant analysis (QDA) classifier [15]. Irwin Ronaldo Donis-Gonzalez et al. developed an automated, noninvasive, inline CT sorting system to determine asparagus quality [16]. Vani used X-ray images of mangoes to train a multi-layer perceptron neural network and discriminant function analysis to create a prediction model for determining the fruit's internal quality [17]. Ahmed et al. classified watermelon seeds using linear discriminant analysis (LDA) and machine learning [18]. Medeiros et al. used a machine learning classifier with linear discriminant analysis (LDA) to sort X-ray images of jatropha seeds into different groups [19]. Van De Looverbosch et al. suggested using a support vector machine (SVM) in conjunction with a feature extraction technique to effectively identify interior defects in 'Conference' and 'Cepuna' using CT images of pear fruit [20]. With the aid of X-ray images and the application of machine learning, Thomas and Thomas proposed a model for the classification of male and female cocoons [21].

We adopt a non-destructive approach to assess the quality of arecanuts using X-ray imaging. The figure 1 shows the X-ray image of an arecanut together with its components. During the radiography procedure, X-rays pass through the denser part of the arecanut and only a few X-rays reach the X-ray detector. This leads to whiter regions on the X-ray image of the arecanut. Similarly, if the arecanuts have cracks, hollowness, or are porous, the detector will absorb more X-rays, resulting in the development of black patches. Thus, X-ray imaging is a valuable imaging technique employed for assessing the quality of an arecanut. The sample X-ray images of the various arecanut grades with and without husk are represented in the figure 2. The figure 2a and figure 2d represents a Grade1 arecanut. The X-ray images of the Grade1 arecanut appear to have no breakage or may have a hollow at the center due to an air gap. In the case of Grade2 arecanut, due to the presence of gaps and cracks at eccentric region of arecanut, the X-ray image of Grade2 arecanut appears as shown in sample figure 2b and figure 2e. However, Grade3 is the lowest quality among arecanut grades, having porosity at the eccentric region of the arecanut. The X-ray image of a Grade3 arecanut is depicted in figure 2c and figure 2f, the presence of black patches indicates porosity and represents a poor quality arecanut.

In order to perform automatic detection of arecanut, YOLOv5 based deep learning architecture is used [22]. Several researchers have proposed hybrid models of YOLOv5 for their applications in agriculture [23], [24], [25], [26], [27], [28], [29], [30]. However, only few researchers have attempted to explore on hyperparameter based performance enhancement in YOLOv5 architecture. Isa et al. investigated the performance enhancement through hyperparameter optimization through manual selection for the detection of sea fishes [31]. The authors focused their efforts on fine-tuning hyperparameters such as the learning rate and the momentum. Adjusting the learning rate anywhere from 0.1 to 0.000001 and the momentum anywhere from 0.9 to 0.9999. Nevertheless, the manual tuning methodology fails to comprehensively investigate the influence of hyperparameters on the accuracy. Zhipeng Wang et al. explored on genetic algorithm based hyperparameter optimization for



(a) Grade1 (b) Grade2 (c) Grade3

(d) Grade1 (e) Grade2 (f) Grade3

FIGURE 2. The images from a to c represent the X-ray images of sun-dried arecanut kernels. The images labelled d to f depict X-ray scans of sun-dried arecanuts that are enveloped by husks, confirming their effectiveness.

the detection of apple calyx [32]. Mantau et al. optimized the hyperparameters for the detection of human objects using a genetic algorithm [33]. However, both of the authors experimented with a mutation probability of 80% and a variance of 0.04 without investigating the crossover functionality of the genetic algorithm.

The primary objective of our research is to enhance quality control standards and the automatic detection of quality of the arecanuts using a non-destructive method. Among the available techniques, X-ray imaging emerged as the most suitable instrument for this purpose. Our first contribution involves the creation of a unique dataset of X-ray images of arecanuts for the purpose of quality inspection. Our second contribution involves the development of an efficient, lightweight deep learning architecture that incorporates the ghost network and the feature pyramid network. A lightweight model is developed in order to run on a CPU/GPU intensive evolutionary algorithm. Finally, we explore by analyzing mutation-based genetic algorithms versus adaptive genetic algorithms for optimizing hyperparameters using the lightweight model, evaluating their respective effects on detection performance.

This research article is organised into four main sections: introduction, materials and methods, results, conclusion and future work. The introduction provides background information and states the problem and objectives of the study. The materials and methods section describes how the study was conducted. The results section discusses the findings of the study. The conclusion and future work summarize the main findings and implications of the study, followed by future directions for further research.

II. MATERIAL AND METHODS

A. EXPERIMENTAL SETUP

1) ARECANUT SAMPLE COLLECTION

Arecanut samples are collected from a farm at Machchattu, Kundapur, Udupi, Karnataka, India (13° 39' 56.088'' N, 74° 58' 47.82'' E) in the month of February 2023 to create a novel X-ray based image dataset. The final form of the arecanut X-ray image dataset is created after following three steps in sequence: (1) arecanut X-ray image acquisition; (2) annotation of the acquired image; and (3) dataset creation.

2) X-RAY IMAGE DATA COLLECTION

TABLE 1. X-ray projection imaging scanning details (at New Medical Centre, Kundapura, Udupi, Karnataka, India)).

Parameters	Particulars
System	Wipro Ge DX-300
Source Voltage	70 kV
Source Current	40 mA
Cycles	50
Exposure Time	0.05 s

Table 1 shows the setup at which X-ray images of arecanut samples are taken. These numbers decide how much energy is used to get the image. Positioning the arecanuts on the X-ray detector plate is the first step in producing an image. The machine operator starts the image process once all

necessary preparations are completed. The machine's X-ray generator sends out a wave of X-rays, which pass through the arecanut and land on the imaging table. While the lighter area will allow X-rays to pass through the arecanut and reach the detector plate, the denser arecanut will not allow them to do so. The X-ray detector plate is then read by a computed radiography (CR) reader, where the image is processed, displayed, and saved in JPG format. Prior to X-ray imaging of arecanuts, we loaded the detector plate with the highest number of arecanuts possible in an effort to keep the cost of the radiography to minimum. It is important to consider that X-ray images taken at different angles of an arecanut will not exhibit changes in quality of the arecanut. Figure 2 illustrates that the core characteristics of arecanuts can vary between Grade1 to Grade3. This means that Grade1 arecanuts consistently exhibit Grade1 traits in their X-ray images, and the same principle holds for Grade2 and Grade3 arecanuts. In essence, the grading process is based on X-ray images remains reliable and independent of the specific angle from which the images are captured. This ensures that each grade is accurately distinguished based on its inherent quality features.

3) DATASET GENERATION

The X-ray images of arecanuts are split apart and saved separately through a split process. Software like LabelImg is used to label each image of an arecanut adhering to the YOLOv5 format [34]. LabelImg is a user-friendly software application that facilitates the creation of text files with annotations for individual images. Subsequently, these files are employed for the purpose of training deep learning models in order to identify arecanuts. The dataset is divided into two distinct subsets: the training set and the validation set. The training dataset comprises 80% of the images, while the remaining 20% are allocated for validation purposes. We did not use image augmentation in this work. This novel dataset is created using a total of 900 images, with 300 images representing each grade.

B. GENERATING FEATURE RICH LIGHTWEIGHT DETECTION MODEL

In order to achieve automated identification of quality arecanuts, we explore the implementation of YOLOv5, a deep learning framework [22]. YOLOv5 is an object detection algorithm that stands for ''You Only Look Once Version 5''. Since 2020, its object detecting precision and quickness have made it popular. YOLOv5 uses a grid of cells to predict bounding boxes and class probabilities for each cell to find objects in an image. But, YOLOv5 contains significant upgrades that make it more precise and efficient. YOLOv5 has backbone, neck, and head networks. YOLOv5's backbone network employs a modified Darknet-53 design, which is responsible for feature extraction from the input image [35]. The neck network connects the head and backbone networks, which aggregates the features generated from the backbone network. The head network estimates image item class probabilities and bounding boxes. The spatial pyramid pooling (SPP) is a feature fusion module that allows YOLOv5 to recognize objects at variable sizes and helps to improve the detection accuracy [36], [37].

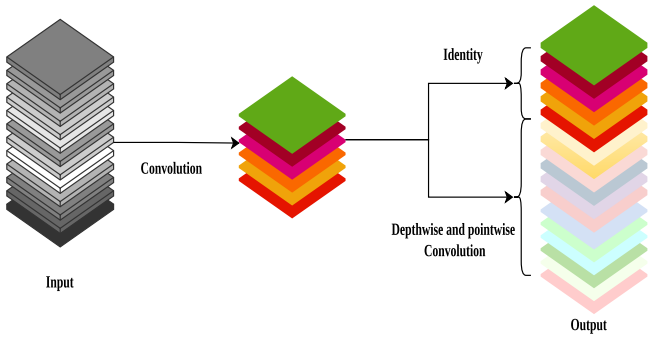


FIGURE 3. Ghost module generating rich features with inexpensive operations and concatenating with convolution.

1) EFFICIENT CONVOLUTION THROUGH GHOST NETWORK

Ghost network is a lightweight deep neural network designed to achieve better accuracy with low computation and memory requirements [38]. The ghost module can be seen as a regular convolutional layer, but instead of creating one feature map for each channel, it creates a smaller number of feature maps through a cheap operation called ghost feature maps. The ghost feature maps are subsequently employed in generating the final output feature maps by means of a blend of linear and nonlinear transformations. This approach reduces the number of parameters while still providing sufficient representational capacity for the network. The ghost network architecture consists of a stem module, several ghost modules, and a classifier. The stem module extracts low-level features from the input image and passes them to the ghost modules. Each ghost module consists of several ghost bottlenecks, which are composed of 1×1 and 3×3 convolutions with ghost modules. The output of the ghost modules is then passed to a classifier module, which consists of global average pooling and a fully connected layer. The generalized representation of ghost network is depicted in figure 3.

The feature map generated through convolution can be represented as:

$$Y = X * f + b \tag{1}$$

where * represents the operation of convolution and bias is represented as b. Given input image $X \in \mathbb{R}^{c \times h \times w}$, where h is the height and w is the width with the channel c. Series of optimal operations generating rich features through inexpensive operations using depth wise and point wise convolution. This generates output feature maps $Y \in \mathbb{R}^{c' \times h' \times w}$ of n channels with h' and w' representing the height and width of the output data. The convolution filter

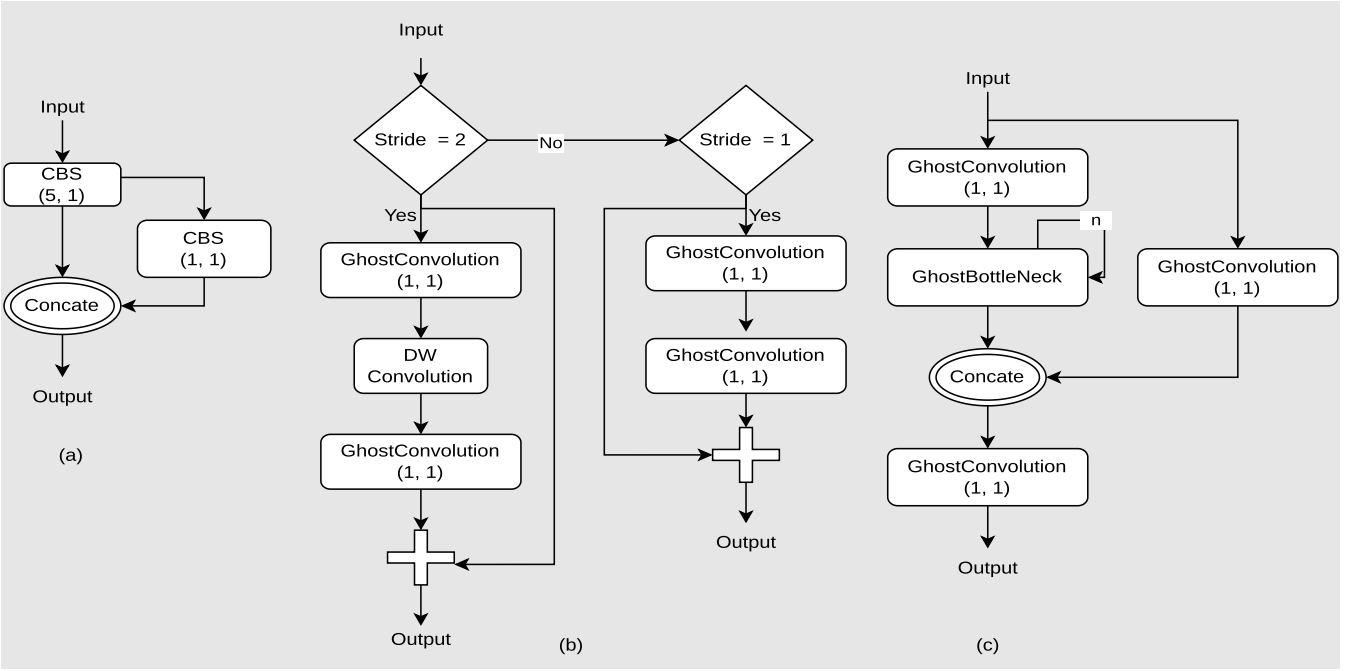


FIGURE 4. Ghost network used for generating rich features with inexpensive operations. a) Ghost Convolution module. b)Ghost Bottleneck module. c) C3Ghost module. Note: '+' indicates addition operation.

$f \in \mathbb{R}^{c\times k\times k\times n}$in convolution layers with $k \times k$ filter size of f.

The process involves a ghost module computation, which is expressed as

$$Y' = X * f' \tag{2}$$

$$Y_{ghost} = Q_j(Y_i), j \in [1, s-1] \tag{3}$$

where Y' represents the features generated through traditional convolution, Y_i' indicates the i^{th} feature map and Q_j is the inexpensive operation generating s-1 ghost feature maps. Thus, we get Y feature maps through concatenation operation, represented as

$$Y = Y_{ghost} + Y' \tag{4}$$

The detailed configuration of the Ghost network architecture is depicted in figure 4. The Ghost convolution module uses two filters of sizes 1×1 and 5×5 as depicted in figure 4a. This is very useful for capturing features at different scales. The 5×5 convolution captures broader contextual information, while the 1×1 convolution focuses on fine details. This combination enables the model to detect objects of varying sizes effectively. Traditional convolutional neural network (CNN) are treated to be hard to run on devices with limited resources, like cell phones or embedded systems, because they require a lot of processing power. The ghost bottleneck module helps to cut down the number of factors and calculations that the network needs to do. This makes the network more efficient and scalable for various platforms. The ghost bottleneck module depicted in figure 4b uses 1×1 and 3×3 convolutions. The $1 \times$ module depicted in figure 4b uses 1×1 and 3×3 convolutions. The $1 \times$ model smaller, but it still has all the important features. The 3×3 convolution is used in the depth-wise(DW) convolution module. The convolutions are typically applied to maintain the spatial resolution of the feature maps when stride is 1, where no down sampling occurs. They help to retain fine-grained spatial information that might be critical for capturing small or detailed features in the input data. For down sampling the spatial dimensions of the feature maps stride is set to 2. This reduces the size of the feature maps so that the receptive field can be increased. The figure 4c uses 3 ghost convolution modules and a ghost bottleneck module which will run for specified figure 4c uses 3 ghost convolution modules and a ghost bottleneck module which will run for specified

2) FEATURE PYRAMID NETWORK (FPN) FOR REDUCTION IN NETWORK LAYERS

We have considered the FPN for feature aggregation, which plays an essential part in enhancing the efficiency of object identification [39]. The fundamental premise of FPN is that they are multi-scale feature pyramids that are generated by extracting feature maps of various spatial resolutions from the backbone of a CNN, and then fusing those feature maps together. This pyramid enables the network to recognize items of varying sizes in an effective manner. This is because smaller things are more accurately represented in feature maps with a higher resolution, while objects of larger size are more accurately represented in feature maps with a lower resolution. We used FPN in YOLOv5 to extract features from multiple layers of the network architecture. These extracted

features are then concatenated together to form a final feature map, which is then used for object detection. Because of this, the network is able to recognize objects of varying sizes and scales with more precision, which ultimately leads to higher detection performance. Referring to figure 5, we use a pyramid with five-scale feature maps (P3, P4, P5, P6, and P7) at stage2. The levels of the pyramid are shown by the subscripts. The backbone network layers from stage1 namely C3, C4, and C5 are used to extract P3, P4, and P5. A top-down convolution is also used to reduce the degradation due to the increase in the depth of the convolutional layers. P6 and P7 are generated with a 3 × 3 convolution with two steps from P5 and P6. This multilevel feature from P3 to P7 shares the information for the detection layers at stage3 to enhance the performance.

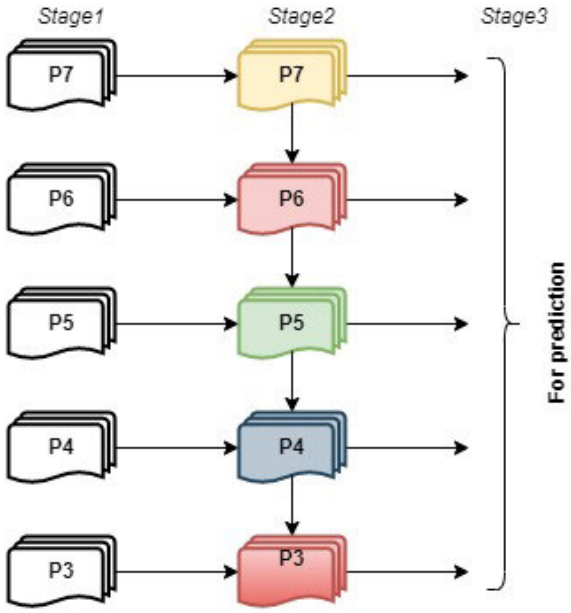


FIGURE 5. The feature pyramid network (FPN) represents the feature aggregation that is generated in the backbone through convolution using the Ghost network in stage1. Stage2 represents the top-down aggregation of features and represents them for detection in stage3.

The layered architecture of lightweight detection model is represented in table 2 which incorporates the Ghost network and the FPN. Module number indicates the number for each module which is from 0 to 19. The modules from 0 to 9 are used for feature extraction using Ghost network, representing the backbone of the YOLOv5 architecture. The modules from 10 to 18 are used for feature aggregation using FPN neck structure as represented in figure 5. The FPN structure uses only top-down approach for feature aggregation making it light structure. Additionally, FPN structure uses Ghost network for convolution process to make the computation less complex with better aggregation of features to be represented for detection or head module 19. Thus, the detection of small, medium, and large-sized objects are recognized at module 19. Referring to table 2, the output of

TABLE 2. The layered architecture used for creating light-weight model structures using the ghost network and the feature pyramid network (FPN).

Module Number	From	N	Parameters	Module Name
0	-1	1	5280	Conv
1	-1	1	22128	GhostConv
2	-1	2	23208	C3Ghost
3	-1	1	85728	GhostConv
4	-1	4	101280	C3Ghost
5	-1	1	337344	GhostConv
6	-1	6	432096	C3Ghost
7	-1	1	1338240	GhostConv
8	-1	2	1346880	C3Ghost
9	-1	1	1476864	SPPF
10	-1	2	1346880	C3Ghost
11	-1	1	0	Upsample
12	[-1, 6]	1	0	Concat
13	-1	1	226752	GhostConv
14	-1	2	341664	C3Ghost
15	-1	1	0	Upsample
16	[-1, 4]	1	0	Concat
17	-1	1	58080	GhostConv
18	-1	2	87888	C3Ghost
19	[18, 14, 10]	1	32328	Detect

one module is used as input for another module, we denote it by specifying the module number. In this notation, "-1" indicates that the module takes into account the input from the module above it. The parameter 'N' represents the number of times a specific module is repeated for use. Finally, the name of the module is represented in the fifth column. The Ghost module is represented by the module names *GhostConv* and *C3Ghost*. Spatial pyramid pooling faster (SPPF) is a variant of spatial pyramid pooling (SPP) designed to reduce the computational expense of the pooling operation. The conventional SPP method computes the pooled feature vectors for each level of the pyramid using a set of maximum pooling layers. However, maximum pooling can be computationally expensive, particularly for large images or maps with high-dimensional feature dimensions. SPPF replaces the initial 5 × 5, 9 × 9, and 13 × 13 maximum pooling layers with three 5 × 5 layers, thereby reducing the number of required pooling operations while maintaining the same level of precision. This modification makes the SPP method more efficient for grading systems by increasing its processing speed.

C. OPTIMIZING HYPERPARAMETERS OF OBJECT DETECTION MODELS USING EVOLUTIONARY ALGORITHMS

The genetic algorithm is a way to solve optimization problems with or without constraints. It is based on natural selection mechanism that drives the evolution of life. Genetic algorithms improve population of potential solutions and finds the best solution in a vast search space. Genetic algorithms start with a population of random search solutions. A common genetic algorithm uses selection, crossover, and mutation on chromosomes. Selection implies choosing fittest parents to have offspring. Fitness functions measure

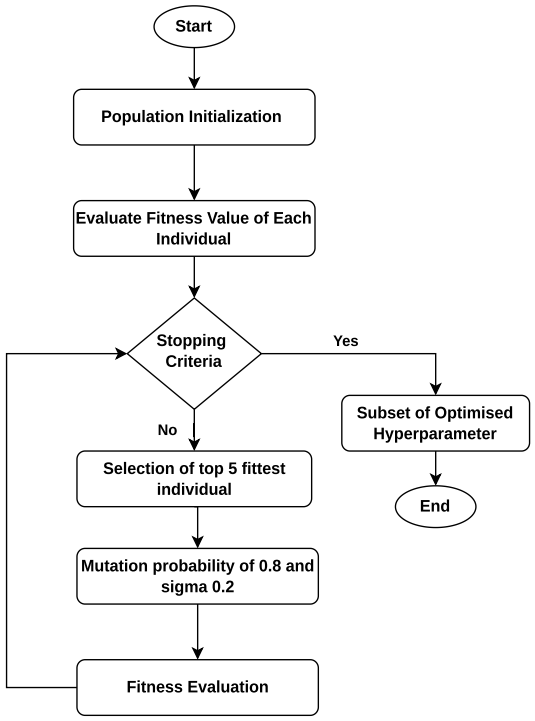


FIGURE 6. Flowchart of the genetic algorithm based approach used in the YOLOv5 architecture.

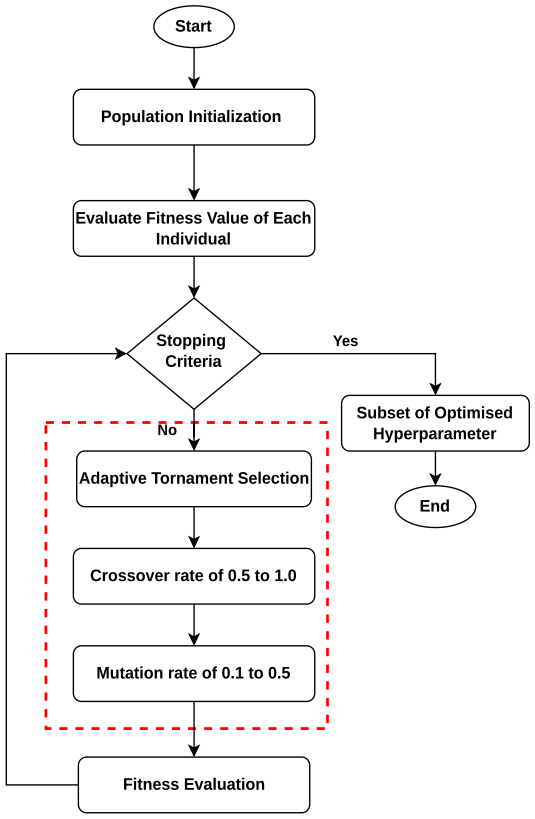


FIGURE 7. Flowchart of the adaptive genetic algorithm-based approach used for hyperparameter optimization. The red dotted rectangle indicates added operations.

a solution's quality. Crossover creates a chromosome from two fittest candidates. The offspring's chromosome has parents' traits. Mutation alters chromosomal values like biological mutation and thus diversity increases. Genetic algorithms are global optimization methods that can find near-optimal answers across the whole search space, unlike grid search and random search, which are local optimization methods. However, genetic algorithms are better than grid search and random search for several reasons [40]. They are great at dealing with large search space, where as grid search becomes impractical and random search might not be the best way due to it's randomness. Their iterative improvement process, balancing exploration and exploitation, allows for the gradual refinement of solutions over generations. We engage the problem of selecting a set of hyperparameter values that results in the best performance as an optimization problem and solve it using the genetic algorithm.

YOLOv5 uses up to 30 hyperparameters and are static during training. However, the process of exploring and determining the optimal combination of hyperparameter values can be challenging and costly due to the vast search space and the unknown inter-dependencies among hyperparameters. By using the genetic algorithm approach, the optimal combination of augmentation hyperparameters for YOLOv5 can be identified, that may enhance object detection performance. Figure 6 illustrates the flowchart of genetic algorithm-based hyperparameter optimization technique using mutation only. In this context, during the evolution process, the top five fittest individuals are selected for mutation with a probability of 0.8 and a variance of 0.04 (or a sigma of 0.2). The evolution process continues until it meets the predetermined stopping criteria, which is a specified number of generations. The final evolved and optimized hyperparameters are considered to have contributed to the enhancement of detection performance (mAP).

1) HYPERPARAMETERS OPTIMIZATION USING ADAPTIVE GENETIC ALGORITHM

Figure 7 illustrates the adaptive genetic algorithm-based hyperparameter optimization technique using both mutation and cross over functionality. The process starts with the creation of a population of candidate solutions, where each solution represents a set of augmentation hyperparameters. The solutions are then evaluated using an objective function, such as mAP, to determine their fitness by choosing 10% of the weight for mAP@0.5 and the remaining 90% of the weight for mAP@0.5:0.95. The fittest candidates are then selected for reproduction, where new candidates are created by combining parts of the fittest candidates. We have considered an adaptive tournament selection algorithm for the selection of the fittest candidate. We utilize mAP@0.5 as a fitness indicator with the objective of maximizing its value. In the context of conventional tournament selection,

Algorithm 1 Adaptive Tournament Selection for Identifying Fittest Individual

procedure AdaptiveTournamentSelection(population, fitness_scores)
    selected_indices ← *emptylist*
    Tmin ← *tournament_size_min*
    Tmax ← *tournament_size_max*
    for *i inrange*(*pop_size* − *elite_size*) **do**
        tournament_size ← max(max(2, *Tmin*), ⌊min(*Tmax*, *pop_size*) − ($\frac{generation}{(\frac{evolve}{10})}$⌋))
        tournament_indices ← *random.sample*(*range*(*pop_size*), *tournament_size*)
        tournament_fitness ← [*fitness_scores*[*j*]*For* j in *tournament_indices*]
        winner_index ← *tournament_indices*[*tournament_fitness.index*(*max*(*tournament_fitness*))]
        selected_indices.append(*winner_index*)
    end for
    elite_indices ← *get_elite_indices*(*fitness_scores*, *elite_size*)
    selected_indices.extend(*elite_indices*)
    Return selected_indices
end procedure

a certain quantity of individuals are chosen at random from the population, and the individual with the most superior fitness value is designated as a parent for the purpose of reproduction. Adaptive tournament selection, on the other hand, adjusts the number of participants in the tournament dynamically based on the characteristics of the population by balancing exploration and exploitation. According to algorithm 1, adaptive tournament selection procedure aims to select individuals from a population for reproduction based on their fitness scores. The procedure takes the population as *population* and the corresponding fitness scores as *fitness_scores* as input and returns the indices of the selected individuals as *selected_individual*. In each iteration of the procedure, a tournament is conducted to choose individuals for reproduction. The tournament size is represented as *tournament_size*, which is adaptively determined based on the current generation, gradually decreasing as the generation progresses. The *tournament_size* is capped by minimum and maximum values, ensuring a balance between exploration and exploitation. For each tournament, a subset of individuals, determined by the *tournament_size*, is randomly selected from the population. Based on the highest *fitness_scores* of the tournament participants, the winner of the tournament is declared. The list of selected indices represented as *selected_indices* is updated with the index of the tournament winner, indicating this individual has been chosen for reproduction. This process will be repeated until the desired number of individuals (*population_size* minus *elite_size*) has been selected. In addition to the tournament winners, the procedure also considers elite individuals. Elite individuals are those with the highest fitness scores in the population. The top elite-size individuals based on their fitness scores are automatically selected, and their indices are added to the list of *selected_indices*. Finally, the procedure returns the list of *selected_indices*, representing the individuals chosen through adaptive tournament selection for reproduction. It's important to consider that the pseudo-code assumes the existence of additional functions, such as *randomly_select_indices* and *get_elite_indices*, to handle the corresponding operations of randomly selecting indices and retrieving the indices of the elite individuals, respectively. These functions are crucial for the execution of the adaptive tournament selection procedure. This technique provides diversity in the population. Crossover and mutation are the two primary processes that operate in genetics. In our work, we considered a crossover rate of 0.5 to 1.0 and mutation rates between 0.1 and 0.5 to produce new offspring based on a mix of the strongest parents from all of the previous generations. The evolution process is continued until the stopping criteria are met. By using this approach, the optimal combination of augmentation hyperparameters for YOLOv5 can be identified, leading to improved object detection performance. Overall technical route of enhancing detection performance using adaptive genetic algorithm on our dataset is depicted in figure 8.

D. EVALUATION METRICS

The study utilizes objective assessment metrics to assess the recognition model's efficiency. These include precision, recall, and mAP (mean average precision). Here are the relevant mathematical formulas:

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{6}$$

Positive samples with the proper classification are known as true positives (TP), false positives (FP), and false negatives (FN), whereas negative samples with the erroneous classification are known as true negatives (TN).

$$\text{AP} = \sum_{n} (R_n - R_{n-1}) P_n \tag{7}$$

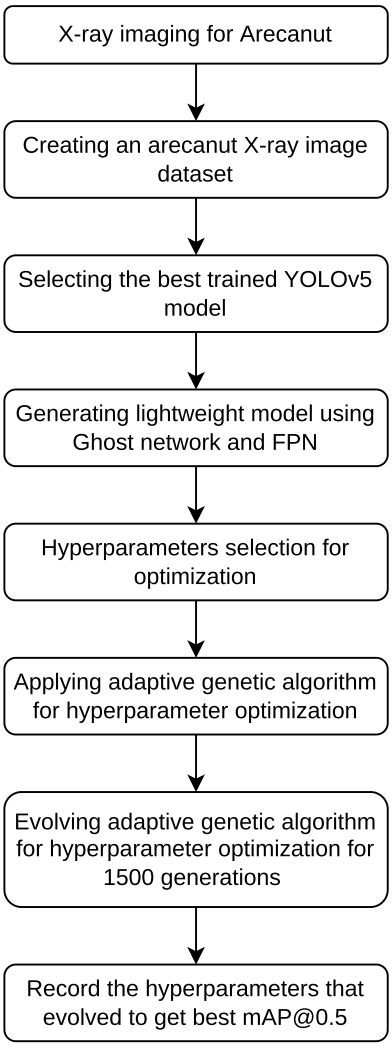


FIGURE 8. Flowchart of the overall structure used to enhance detection performance using an adaptive genetic algorithm on our dataset.

where R_n and P_n are the precision and recall at the n^{th} threshold.

$$\text{mAP} = \frac{1}{n}\sum_{k=1}^{k=n} AP_k \tag{8}$$

where, AP_k = the Average Precision for class k, n = the number of classes. We consider mAP@[0.5:0.95] indicating average mAP over different intersection over union thresholds (IoU) ranging from 0.5 to 0.95 with step size of 0.05. In addition, the layers, the floating point operations (FLOPs), the parameters reflect the computational quantity during training.

Overall, the experiment began by comparing the detection results of various YOLOv5 models, and the model with better accuracy is chosen. Since the evolutionary process of genetic algorithms is GPU-intensive, we generated a lightweight version of a feature rich model using the ghost network and FPN. To ensure that the model achieved its full potential, an adaptive genetic algorithm is introduced. This algorithm is used in optimizing augmentation related hyperparameters to fine-tune the model and achieve even higher accuracy.

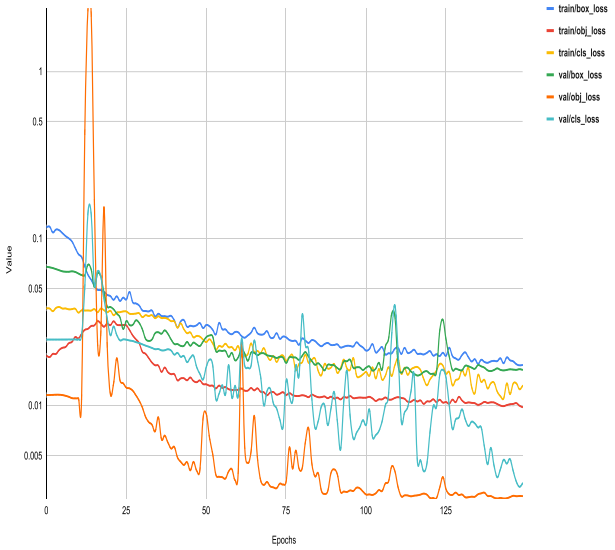


FIGURE 9. The representation of loss chart of light weight model generated during training for 150 epochs.

III. RESULTS

This study utilized the PyTorch library for all experiments and employed two NVIDIA Tesla M40 GPUs with GM200 graphics processors. The configuration used for training in sections III-A and III-B consists of a batch size of 100, training for 150 epochs, an image size of 416 × 416, employing the Sigmoid Linear Unit (SiLU) activation function [41], and utilizing the Stochastic Gradient Descent (SGD) [42] optimizer. However for section III-C, epochs is reduced to 20 and we included evolution that runs for 1500 generations.

A. PERFORMANCE COMPARISON ON USING VARIOUS YOLOV5 MODELS

YOLOv5 defines several models from complex to simple based on the width and depth of multiple. Complex model being YOLOv5l with more number of layers with larger model size, whereas the simple model being YOLOv5n has least number of layers with smallest model size [43]. We compared the detection performance on our dataset for models from complex to simple and recorded the result in table 3. The objective is to identify an optimal model for detecting arecanuts. After conducting experiments, it is found that the models YOLOv5m and YOLOv5l produced the better results, with mAP of 96.14% and 96.02%, respectively. However, by comparing the higher mAP of 96.14% and lower model size of 42.3 MB, the YOLOv5m model is selected over YOLOv5l for further investigation.

TABLE 3. Experimental results of YOLOv5 models trained from scratch on our dataset.

Model	mAP@0.5(%)	mAP@0.5:0.95(%)	Precision(%)	Recall(%)	Layers	Parameters(Million)	FLOPs(Billion)	Model Size(MB)
YOLOv5n	95.68	79.40	91.88	89.47	270	1.76	4.2	3.8
YOLOv5s	94.76	78.77	87.65	94.06	270	7.02	16	14.3
YOLOv5m	96.14	80.35	91.54	91.15	369	20.87	48.3	42.3
YOLOv5l	96.02	78.77	87.65	94.06	468	46.15	108.3	92.8

TABLE 4. Result of ablation experiment performed on implementing lightweight model using Ghost module and Feature Pyramid Network (FPN) to generate lightweight for faster processing in genetic algorithm. M1-YOLOv5m, M2-YOLOv5m+Ghost network, M3-YOLOv5m+Ghost network+FPN.

Model	mAP@0.5(%)	mAP@0.5:0.95(%)	Precision(%)	Recall(%)	Layers	Parameters(Million)	FLOPs(Billion)	Model Size(MB)
M1	96.14	80.35	91.54	91.15	369	20.87	48.3	42.3
M2	95.38	78.34	87.78	92.21	695	8.53	18.5	17.6
M3	95.73	78.08	89.23	90.80	617	7.26	16.6	15.0

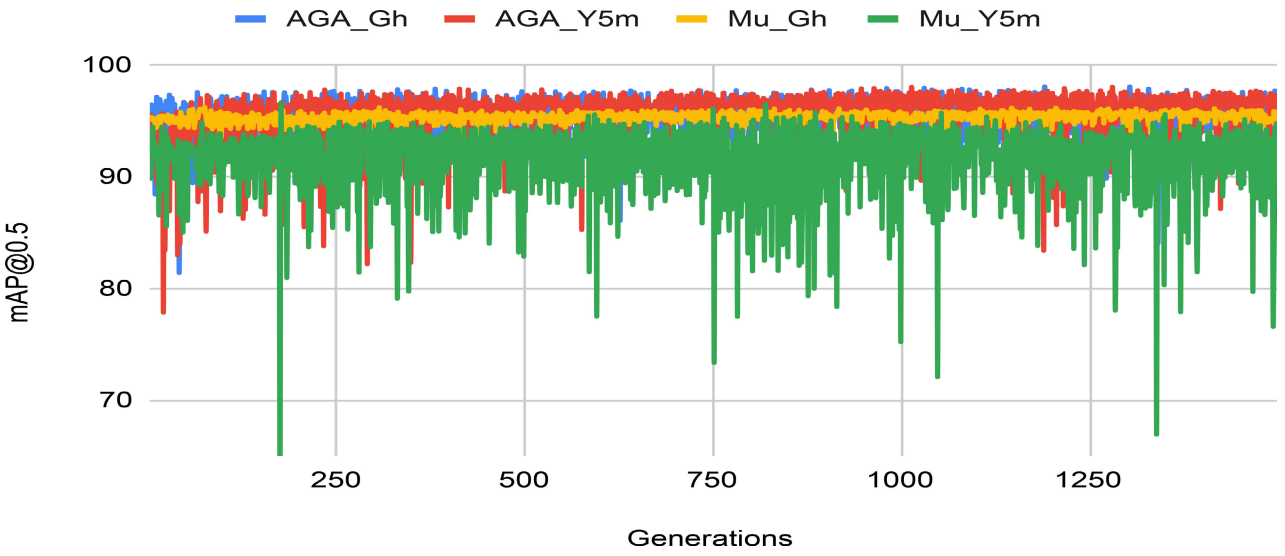


FIGURE 10. The results obtained from conducting tests involving the application of the adaptive genetic algorithm and YOLOv5 based mutation technique over a period of 1500 iterations on our dataset. AGA_Gh- Adaptive Genetic Algorithm using Ghost network and FPN, AGA_Y5m- Adaptive Genetic Algorithm using YOLOv5m, Mu_Gh - Mutation based evolution using Ghost network and FPN, Mu_Y5m- Mutation based evolution using YOLOv5m.

B. PERFORMANCE COMPARISON OF USING LIGHTWEIGHT ARCHITECTURE

Incorporating a ghost network in backbone and implementing FPN in the neck structure of the architecture leads to a notable reduction in the model size. Ghost network is a lightweight CNN architecture that is designed to reduce the model size and improve the efficiency. This happens due to the removal of redundant feature maps and keeping the identity feature maps during feature extraction process. FPN, on the other hand, is a feature aggregation technique, which helps the network to detect objects at multiple scales with halved layers as compared to the path aggregation network (PANET) structure [44]. Thus, replacing PANET with FPN will further reduce the model size. It is noticed that PANET is the path aggregator which is used in YOLOv5 models. After implementing ghost network [38]and FPN [39] on YOLOv5m, the model size gets reduced from 42.3 MB to 15 MB, with a promising mAP of 95.73%. This is a significant improvement over the previous models, and it demonstrates the effectiveness of these techniques for reducing the model size. Table 4 shows the result of ablation studies performed on the YOLOv5m model. Even though using ghost network and FPN reduced the mAP by 0.42%, but there is a significant reduction in the model size by 64.54 % in contrast to YOLOv5m model. This approach is employed to establish a model that enhances a model with rich feature maps and, more significantly, to achieve a smaller size, hence facilitating faster execution of CPU-intensive genetic algorithms, as utilized in the subsequent section. The training and validation loss chart of using lightweight model is represented in figure 9. While working with a smaller sample size, it is noticed that the validation dataset may not entirely represent the overall data distribution, leading to increased fluctuations in the loss curve. However, these fluctuations in the loss curve should not necessarily be a cause for concern. As figure 9 illustrates, the overarching

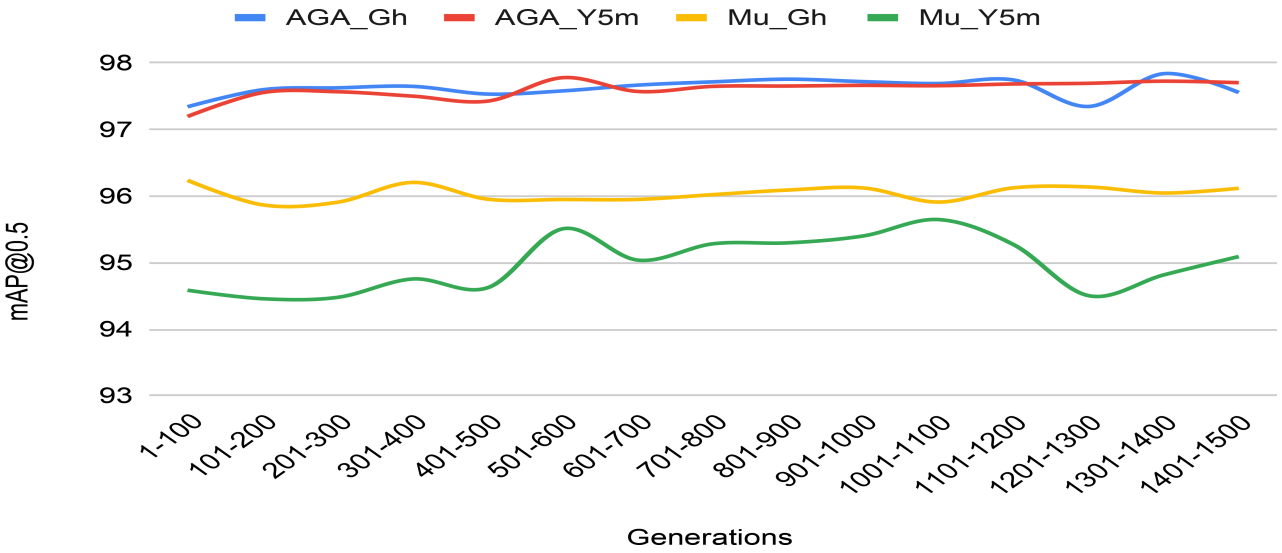


FIGURE 11. Re-representation of the figure 10 in order to enhance its visual clarity. Each data point corresponds to the maximum mean average precision (mAP) achieved during a span of 100 consecutive generations, with a maximum limit of 1500 generations. AGA_Gh- Adaptive Genetic Algorithm using Ghost network and FPN, AGA_Y5m- Adaptive Genetic Algorithm using YOLOv5m, Mu_Gh - Mutation based evolution using Ghost network and FPN, Mu_Y5m- Mutation based evolution using YOLOv5m.

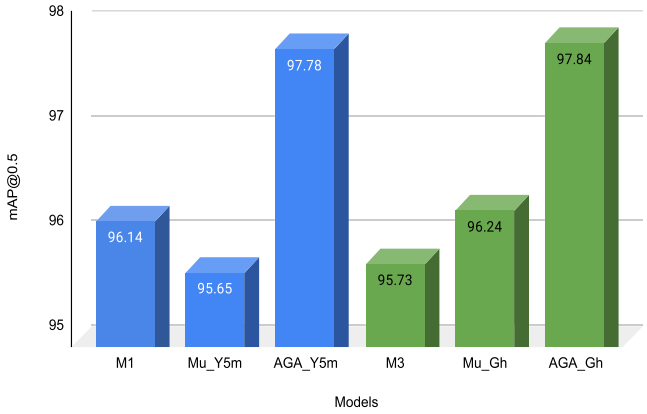


FIGURE 12. The experimental outcomes of the utilized models. M1-YOLOv5m, M3- YOLOv5m+Ghost network+FPN, AGA_Gh- Adaptive Genetic Algorithm using Ghostnetwork and FPN, AGA_Y5m- Adaptive Genetic Algorithm using YOLOv5m, Mu_Gh - Mutation based evolution using Ghostnetwork and FPN, Mu_Y5m- Mutation based evolution using YOLOv5m.

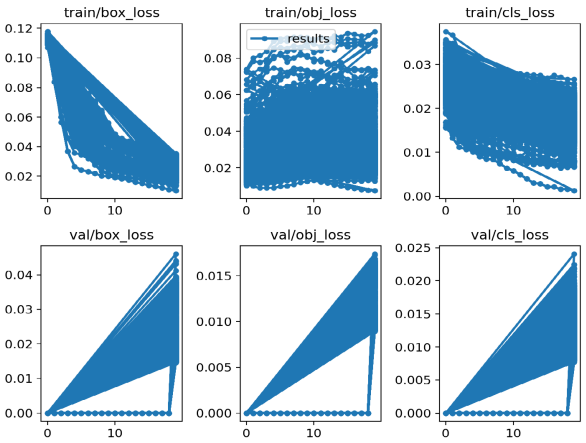


FIGURE 13. The result of losses occurred during training and validation while using the adaptive genetic algorithm.

trend of the validation loss curve is descending, indicating the model's capacity to learn and improve during training. These fluctuations in the loss are not mirrored in the training loss, which can be attributed to the larger number of images in the training dataset.

C. PERFORMANCE COMPARISON USING ADAPTIVE GENETIC ALGORITHM

The evolutionary process of a genetic algorithm is dependent on four essential factors, which we provide as follows. First, selection of trained model having architecture and state dictionary. Second, the selection of hyperparameters that need to be optimized. We consider thirteen augmentation related parameters, and they all began with default values. These default values may be found in the third column of table 5. Third, the number of generations to evolve is set to 1500. Finally, the number of epochs for each generation is set to 20.

The evolution process is applied to models M1 and M3 from the table 4. Figure 10 illustrates the performance comparison between the YOLOv5-based genetic algorithm with mutation-only and the adaptive genetic algorithm-based approach. The figure 11 is re-representation of the figure 10 to enhance its visual clarity. The mutation-only operation, as described by the YOLOv5 architecture, is executed, yielding a mAP of 95.64% for model M1 and 96.24% for model M2. This shows, reduction in mAP by 0.51% in M1 model, but shows improvement in mAP by 0.53%

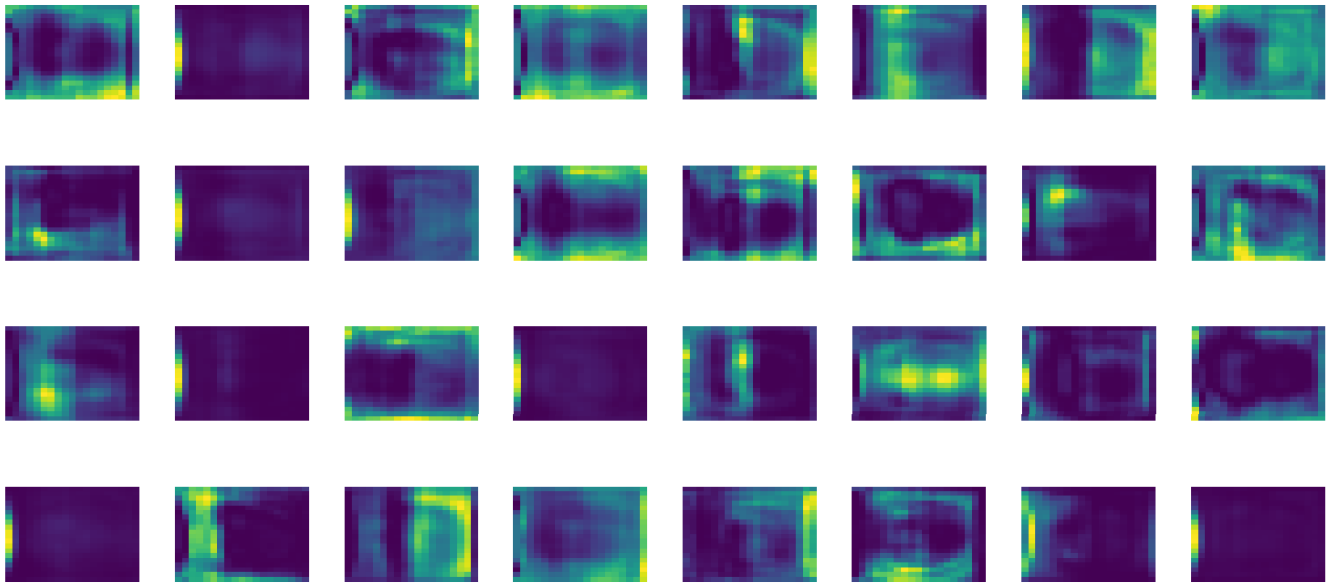


FIGURE 14. The feature map visualization at the final layer of YOLOv5m is generated using an X-ray image of a Grade1 arecanut.

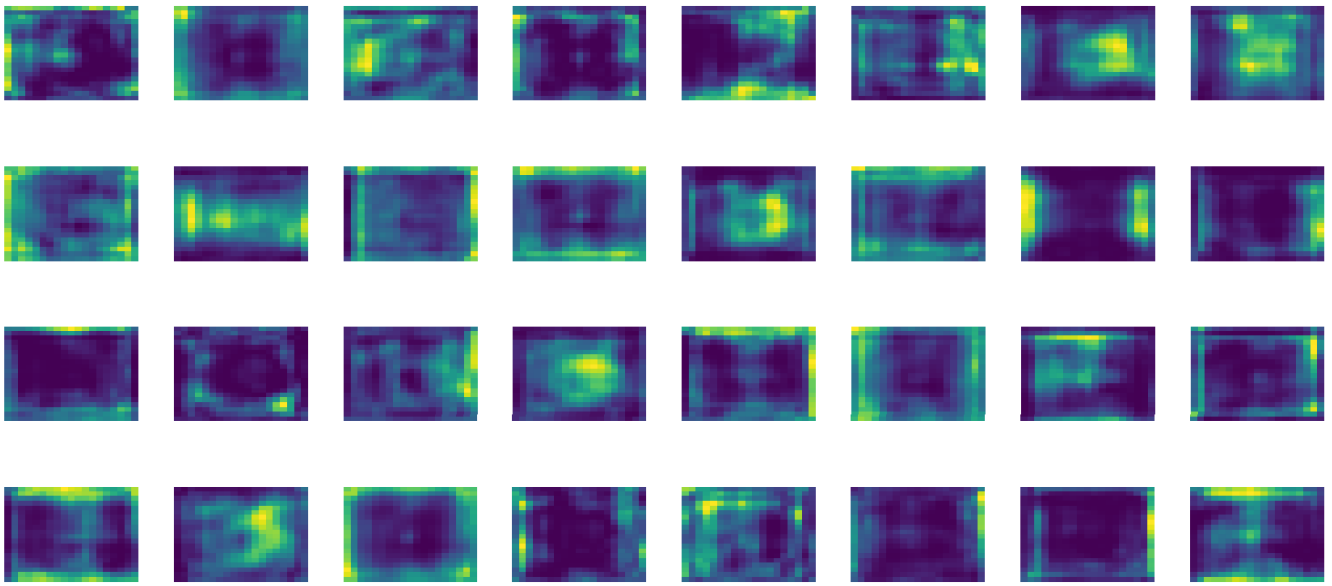


FIGURE 15. The feature map visualization at the final layer of the optimized model is generated using an X-ray image of a Grade1 arecanut.

for model M3. However, when our dataset is experimented using adaptive genetic algorithm on M1 model, the mAP increased from 96.14% to 97.78%, which is 1.70% increase in mAP. In case of using adaptive genetic algorithm on M3 model, the mAP increased from 95.73% to 97.84%, which is 2.20% increase in mAP. From the figure 11 following observations are made. Firstly, the adaptive genetic algorithm has impact on the mAP as compared to YOLOv5 defined mutation operation. Secondly, the introduction of feature rich model 'M3' has shown significant contribution in enhancing the mAP. This is evident from both mutation operation and adaptive genetic algorithm approach depicted as 'AGA_Gh' and 'Mu_Gh' in figure 11. Thirdly, even though the model size of M3 is 15MB and M1 is 42.3MB, the top result is achieved for feature rich model M3, this shows model size may not contribute to the enhancement of mAP. Finally, the mAP for 'AGA_Gh' and 'AGA_Y5m' are overlapping with similar results, M3 model is preferred over M1 model due to its lower model size. A comparative detection performance of various models used is represented in figure 12.

After the evolution process is completed, a plot of fitness (shown on the y-axis) vs hyperparameter values (shown

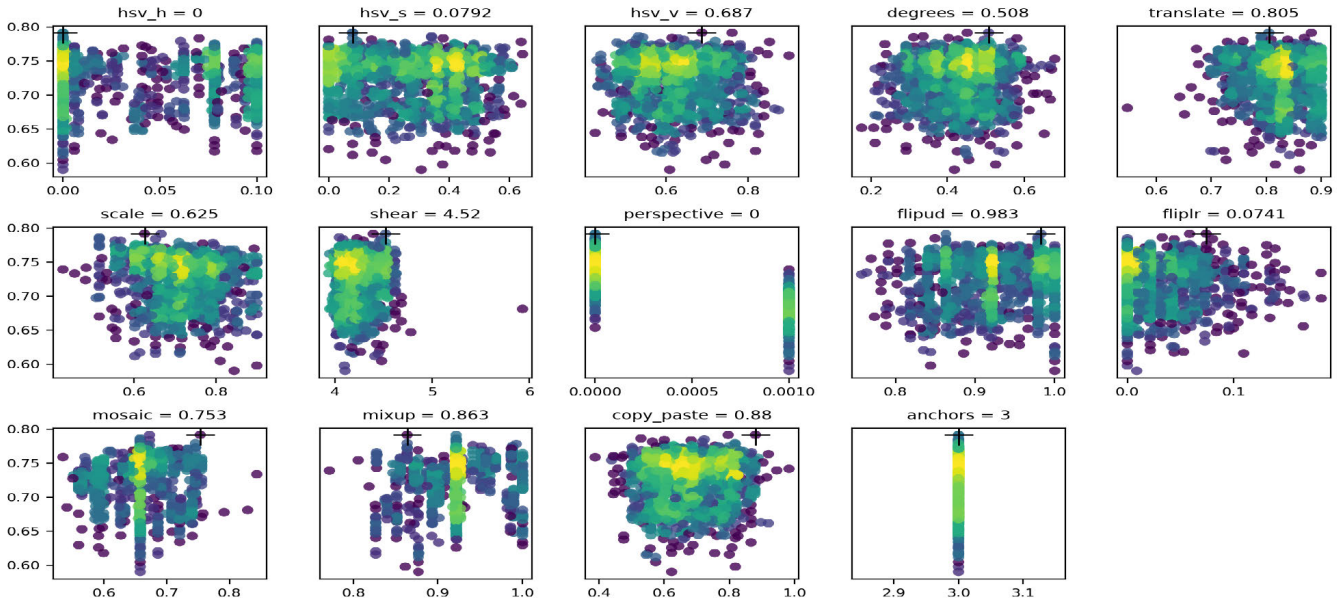


FIGURE 16. Hyperparameter evolution that is completed with one subplot per hyperparameter indicating fitness (y-axis) vs. hyperparameter values (x-axis) that are evolved through an Adaptive Genetic Algorithm for 1500 generations using lightweight model M3. The +' sign indicates the highest fitness value obtained for respective hyperparameter. Yellow indicates higher concentrations.

TABLE 5. The tuned hyperparameter values that are generated through a genetic algorithm using adaptive selection techniques for our dataset recognize arecanuts. Models 1 to 10 represent a series of models that have evolved through various augmented hyperparameters using model M3. Model1 stands out as the top-performing generated model, while the remaining nine models also showcase exceptional performance as some of the best generated models.

Parameter Name	Search Space	Default Value	Model1	Model2	Model3	Model4	Model5	Model6	Model7	Model8	Model9	Model10
Generation	1 to 1500	-	**1303**	880	1142	1190	904	733	1030	640	934	1395
mAP_0.5	0 to 100	-	**97.84 %**	97.76 %	97.75 %	97.74 %	97.72 %	97.72 %	97.69 %	97.67 %	97.66 %	97.65 %
hyp.copy_paste	0 to 1	0	**0.6886**	0.6551	0.6824	0.6012	0.6697	0.6571	0.6014	0.5595	0.5984	0.6342
hyp.degrees	0 to 45	0	**0.4963**	0.2750	0.4469	0.4469	0.3568	0.3816	0.4073	0.2912	0.4469	0.3308
hyp.fliplr	0 to 1	0.5	**0.0000**	0.0583	0.0216	0.0216	0.0000	0.0000	0.0194	0.0088	0.0216	0.0000
hyp.flipud	0 to 1	0	**0.8635**	0.9069	0.9213	0.8635	0.9213	0.9790	0.9213	0.9213	0.9213	0.9789
hyp.hsv_h	0 to 1	0.015	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0110	0.0000	0.0058	0.0000	0.07819
hyp.hsv_s	0 to 0.9	0.7	**0.0000**	0.2681	0.0000	0.0000	0.4185	0.3178	0.1421	0.3598	0.2133	0.41704
hyp.hsv_v	0 to 0.9	0.4	**0.7000**	0.5155	0.5532	0.5532	0.6317	0.5936	0.5532	0.5427	0.5532	0.54269
hyp.mixup	0 to 1	0	**0.9218**	0.9218	0.9218	0.9218	0.9218	0.9218	0.9218	0.9218	0.9218	0.9218
hyp.mosaic	0 to 1	1	**0.6576**	0.6576	0.6576	0.6576	0.6576	0.6576	0.6576	0.6576	0.6576	0.6575
hyp.perspective	0 to 0.001	0	**0.0000**	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
hyp.scale	0 to 0.9	0.5	**0.6660**	0.7128	0.6524	0.6524	0.6877	0.7962	0.6524	0.7128	0.6524	0.7128
hyp.shear	0 to 10	0	**4.4617**	4.1921	4.5462	4.4618	4.2375	4.3076	4.2930	4.1670	4.2086	4.0826
hyp.translate	0 to 0.9	0.1	**0.8391**	0.8087	0.8264	0.8264	0.8264	0.8264	0.8264	0.8087	0.8264	0.8086

on the x-axis) is created as shown in figure 16. The color yellow represents a greater concentration. When a parameter has a vertical distribution, it indicates that the parameter is not allowed to evolve. The losses occurred during evolution process using adaptive genetic algorithm is presented in figure 13. Table 5 represents the top 10 models that have best performed using an adaptive genetic algorithm based approach on model M3. However, the genetic algorithm also derived 271 models with mAP of 97% and above. Referring to table 5, various models have generated different hyperparameter values through Genetic algorithms using adaptive tournament selection techniques. Some of the hyperparameters like hue (*hsv_h*), saturation (*hsv_s*), *perspective* has ideally settled to zero values after evolving for several generations when observed from the top 10 results. Overall, the result demonstrates the significance of hyperparameter tuning in optimizing the performance of deep learning models. The improvements achieved through hyperparameter tuning highlight the potential for further advancements in this field and underscore the importance of careful and thorough optimization to achieve the best possible results. Comparing with the base model YOLOv5m, we achieved a 1.77% increase in model performance using hyperparameter optimization, and the model size is reduced to 63.54%. Comparing to the optimized model (using Ghost network and FPN), the evolved model has significantly improved in mAP by 2.20%. It is important to consider that the genetic algorithm is able to generate multiple optimal

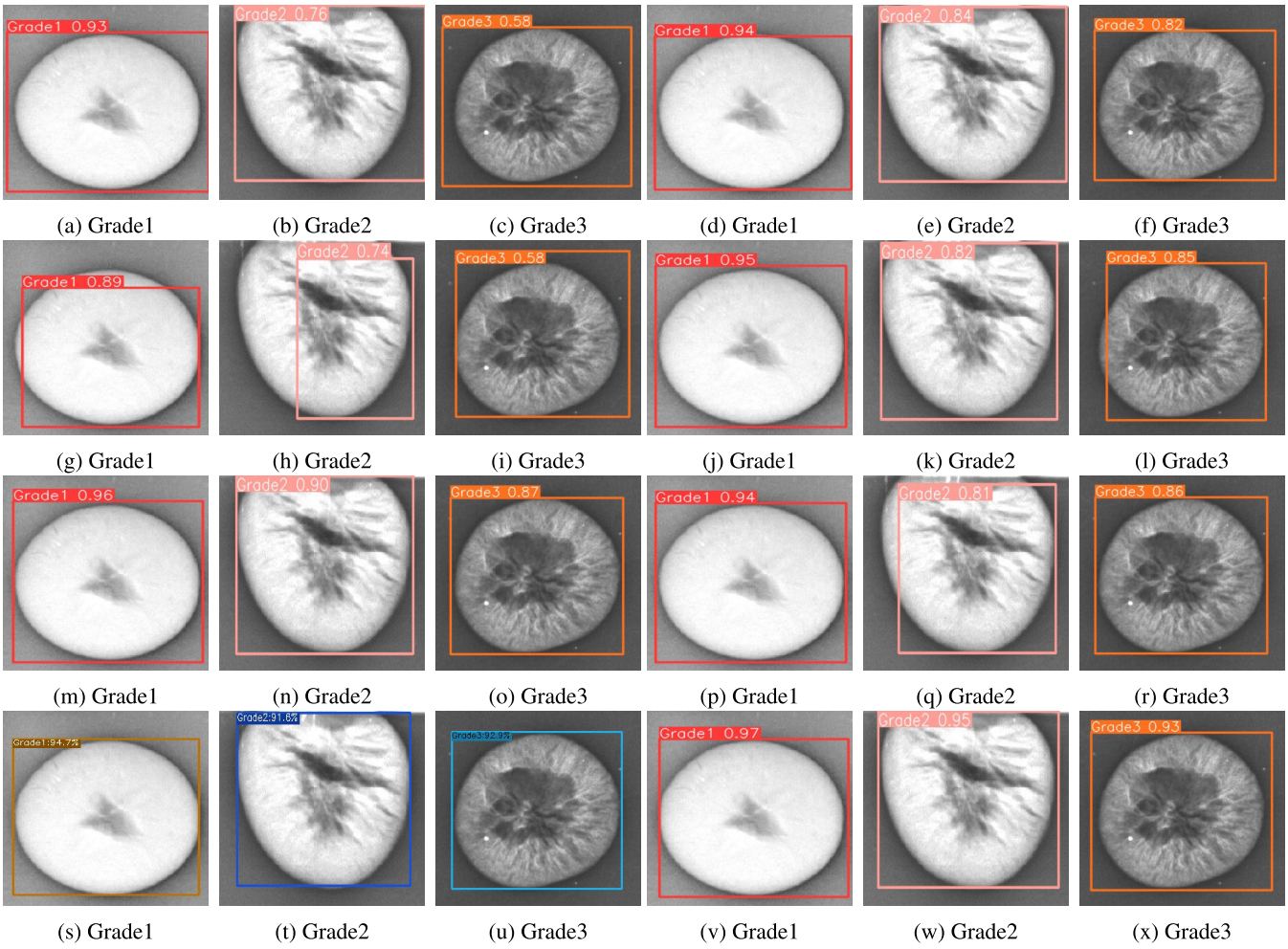


FIGURE 17. Detection results from using cutting-edge object detection algorithms on our dataset. The images from a to c represent the detection result of using the YOLOv3 model. The images from d to f represent the detection result of using the YOLOv4 model. The images from g to i represent the detection result of using the Detetron model. The images from j to l represent the detection result of using model YOLOv5m. The images from m to o represent the detection result of using model YOLOv6. The images from p to r represent the detection result of using the model YOLOv8. The images from s to u represent the detection result of using the model YOLOX. The images from v to x represent our method of detection.

models by exploring various hyperparameters with respect to augmentation. Thus, using adaptive genetic algorithm, the optimization of augmented hyperparameters to the lightweight YOLOv5m model resulted in an improvement in accuracy, with the highest mean average precision (mAP) value of 97.84% when evolved at generation 1303. This is a notable improvement over the previous models, and it demonstrates the effectiveness of hyperparameter tuning in improving the accuracy of the model. Thus, using an adaptive genetic algorithm based approach for hyperparameter optimization has contributed to the enhancement of detection performance (mAP) for the following reasons. First, YOLOv5 models use hyperparameters during training time are static, hence performance enhancement depends only on the neural network architecture. However, our method is trained for hyperparameter optimization and they are dynamically changed during the training phase. This contributes to the further enhancement of model detection performance. Second, our approach strikes a balance between exploration (searching broadly across the search space) and exploitation (focusing on promising configurations). This balance helps to prevent getting stuck in local optima and promotes the discovery of better hyperparameter settings. Third, our approach adapts over time by selecting and recombining the best-performing hyperparameters, leading to a better convergence towards optimal or near-optimal detection performance. This adaptability can be particularly useful when the relationship between hyperparameters and performance is complex and nonlinear. Table 6 represents the detection performance of various state-of-the art object detection algorithms. It can be observed that our genetically evolved models AGA_Y5m and AGA_Gh using adaptive approach has outperformed among all the state-of-the-art models. However, the model AGA_Gh is the best evolved model with mAP of 97.84%. Although, AGA_Y5m model has near best mAP of 97.78%, it exhibits poor performance

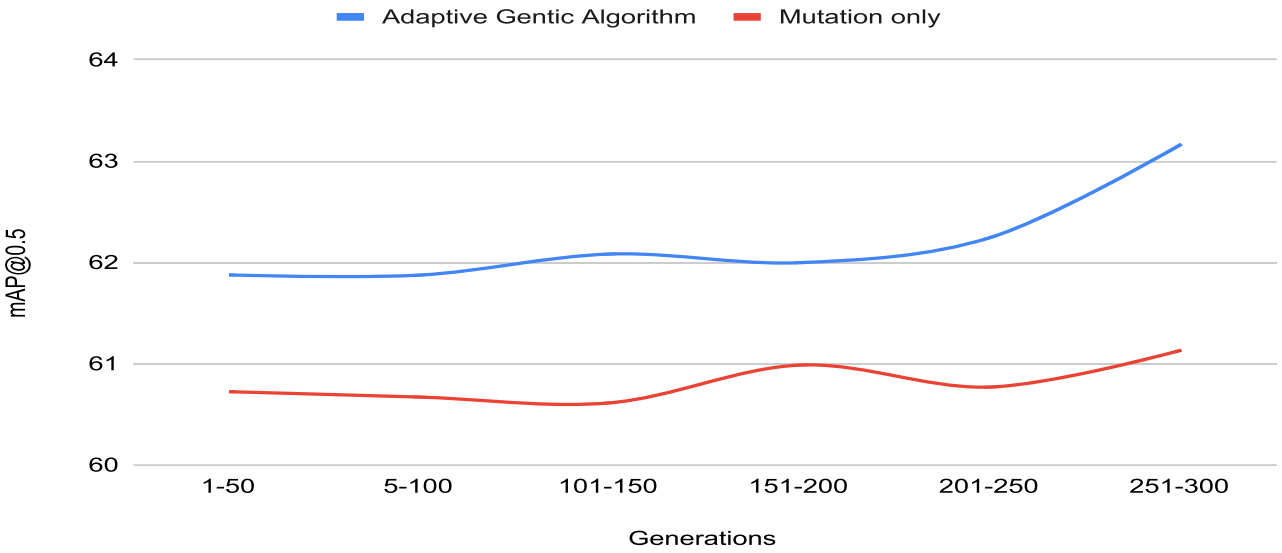


FIGURE 18. Comparative line chart of adaptive genetic algorithm versus mutation based genetic algorithm applied to Pascal VOC2007 image dataset. Each point represents the top mAP for every 50 consecutive generations, up to 300.

TABLE 6. Comparison with other cutting edge object detection algorithms. Values are in percentage. AGA_Gh-Adaptive Genetic Algorithm using Ghost network and FPN, AGA_Y5m-Adaptive Genetic Algorithm using YOLOv5m.

Models	mAP@0.5	mAP@0.5:0.95	Precision	Recall
YOLOv3	93.6	72.62	88.24	91.37
YOLOv4	95.94	78.83	95.38	90.76
YOLOv6	96.58	81.11	88.7	92.17
YOLOv8	95.2	80.01	92.7	90.9
Detetron	90.9	72.6	91.7	91.2
YOLOX	93.48	78.43	93.5	83.2
Our method				
AGA_Y5m	97.78	57.42	95.27	93.93
AGA_Gh	97.84	74.74	93.07	95.87

in mAP@0.5:0.95, achieving only 57.42%, as indicated in table 6. This is due to the fact that, bounding box predictions is lower with IoU thresholds (0.75, 0.8, etc.), resulting in lower mAP@0.5:0.95 performance. This suggests that the AGA_Gh model' has improved localization accuracy and robustness as compared to model AGA_Y5m, leading to the selection of the AGA_Gh model as the superior performing model. The feature map visualization result is represented in figure 14 and result is represented in figure 14 and 15 that are run on using adaptive genetic algorithm and YOLOv5m models respectively, the

D. EVALUATION OF PERFORMANCE AMONG CUTTING-EDGE MODELS

The figure 17 represents the detection result for various experiments performed on the same X-ray image of arecanut grade type. From figure 17 we observe the variations in detection performance on various models used in this experiment. The analysis of the detection performance is conducted on several cutting edge models whose weights are similar to the weights generated in our experiment and results are represented in table 6. The results of the detection performance are illustrated in Figure 17. However, the detection performance for Grade1 arecanut has been demonstrated to exceed 90% for all the models, except for the Detetron model, which exhibited a detection accuracy of 89%. There are only three models, namely YOLOv6, YOLOX, and our model, that have achieved a detection accuracy of above 90% for Grade2 arecanut. Among the several models examined, it has been observed that YOLOX and our model, has demonstrated a detection accuracy over 90% specifically for Grade3 arecanut. However, the performance of detection may exhibit variability between different models and individual images.

E. PERFORMANCE COMPARISON ON PASCAL VOC 2007 DATASET

We expanded our analysis of performance on the Pascal VOC 2007 image recognition dataset [45]. The dataset is initially trained using the YOLOv5m model. Subsequently, our evolutionary-based strategy is employed for 300 generations, incorporating mutation and an adaptive genetic algorithm. The detection performance on the dataset is illustrated in figure 18 using two distinct techniques. The superiority of the adaptive genetic algorithm-based strategy over mutation-based evolutionary strategies and its consistent contribution to the improvement of detection performance can be shown from Figure 18. The mutation strategy yielded a maximum mean average precision (mAP) of 61.13%, whereas the adaptive genetic algorithm approach resulted in a higher mAP of 63.17%. Several studies have performed deep learning based classifications that use X-ray images, which are represented in the table 7.

TABLE 7. A comparison between the proposed method and the related work on internal examination. ML-Machine Learning, DL-Deep Learning, CT-Computed Tomography, CNN-Convolutional Neural Network.

Author(s)	Crop	Image Type	Detection Type	Method	Accuracy
Casasent *et al.* [14]	Pistachios	X-Ray image	Classification based on quality	CNN	89.3
Donis-Gonz *et al.* [15]	Chestnut	CT image	Classification based on decay	CNN	85.9
Donis-Gonz *et al.* [16]	Asparagus	CT image	Classification of fibre	ML	91.2
Ashok [17]	Mango	X-Ray image	Classification of defective and non-defective	ML	91.3
Ahmed *et al.* [18]	Watermelon seed	X-Ray image	Morphology of seed	DL	87.3
Medeiros *et al.* [19]	Jatropha seed	X-Ray image	Germination and seed vigor	ML	89.76
Van De Looverbosch *et al.* [20]	Pear	3-D vision and X-Ray	Defects in pear	ML	95.1
Thomas and Thomas [21]	Silk worm	X-Ray image	Classification based on cocoon gender	ML	96.3
Dael *et al.* [46]	Orange and lemon	X-Ray image	Classification of healthy and damaged tissues in citrus fruit	ML	94.65
Dael *et al.* [47]	Pear	3-D vision and X-Ray	Classification of defective and non-defective	ML	95.2
Medeiros *et al.* [48]	Crambe seed	X-Ray image	Classification of seed vigor and viability	CNN	95
Van De Looverbosch *et al.* [49]	Pear	3-D vision and X-Ray	Classification based on consumption	DL	95.8
Our study	Arecanut	X-Ray image	Quality determination and localization	DL and Optimization	97.84

F. PRACTICAL IMPLICATIONS

The application of deep learning models trained on X-ray images of arecanuts holds significant practical implications for quality determination within the arecanut industry. These models enable automated detection, resulting in improved efficiency and enhanced accuracy. Additionally, they offer real-time quality control, ensuring standardization and reducing costs. The aforementioned models employ X-ray image analysis techniques to categorize arecanuts into different quality classes, hence enhancing efficiency in processing and eliminating defective nuts from the manufacturing line. Deep learning models offer a consistent methodology for evaluating the quality of arecanuts, ensuring the provision of premium products to customers and sustaining market competitiveness. The incorporation of deep learning models into the X-ray equipment utilized for assessing the quality of arecanuts yields several advantages. The real-time assessment of arecanut quality can be achieved by simply embedding models on the devices, eliminating the need for additional research tools. The gadgets use optimized and integrated technology that enhances processing speed and reduces dependence on external resources. Improvements can be made to user interfaces in order to enhance user ability to recognise unambiguous outcomes and visual aids that facilitate decision making. Ultimately, the integration of optimal deep learning models into X-ray equipment enhances the efficiency and quality control procedures for arecanuts. There is currently no equipment available in the arecanut industry that uses a non-destructive approach to quality assurance.

IV. CONCLUSION AND FUTURE WORK

In this study, a non-destructive approach is adopted for the identification of real quality of arecanuts using X-ray images. To automate the process of quality identification, we have created a novel dataset of 900 X-ray images of arecanuts. Initially, the dataset is trained using YOLOv5 models, the YOLOv5m model achieved the highest mAP of 96.14% with a model size of 42.3 MB, outperforming other YOLOv5 models. A light-weight version of model YOLOv5m is generated using the ghost network and FPN to get a feature rich model with a reduction in model size of 15MB and a mAP of 95.73%. Furthermore, in order to enhance the model detection accuracy (mAP), we considered 13 hyperparameters associated with augmentation available in YOLOv5. Hyperparameter optimization is performed using adaptive genetic algorithm and mutation based genetic algorithm by evolving them for 1500 generations on models M1 and M3. Mutation based optimization approach on M1 model has reduced the mAP to 95.65%. However, when adaptive genetic algorithm is used, we got mAP as 97.78%. However, when mutation based optimization is applied to the M3 model, the mAP increased to 96.24%. Furthermore, when the adaptive genetic algorithm is used, we observed enhanced mAP of 97.84% as illustrated in figure 12. The findings of the research indicate that the mutation based optimization approach has shown mixed results for the models M1 and M3. While adaptive genetic based algorithm have shown enhancement in the mAP for the models M1 and M3. Comparing mutation based genetic algorithm and adaptive genetic based algorithm, we have observed enhancement in mAP, as illustrated in figure 11. Even though evolved model M1 (AGA_Y5m) using adaptive genetic algorithm has a near best mAP of 97.78%, we prefer to consider evolved model M3 (AGA_Gh) due to its lower model size of 15 MB, better mAP@0.5:0.95 of 74.54% and better detection performance of 97.84% as illustrated in table 6. This research has demonstrated that deploying a feature-rich lightweight model in conjunction with the adaptive genetic algorithm has yielded a substantial enhancement in detection performance. The detection performance is evaluated for other state of the art detection models such as YOLOv3, YOLOv4, Detetron, YOLOv6, YOLOv8, and YOLOX-s models. However, our model AGA_Gh has performed better in detection, as illustrated in table 6. We also noticed that using an adaptive genetic algorithm on the Pascal VOC 2007 image dataset has improved the detection result. In the future, it is essential to investigate other methods that may be used to carry out non-destructive quality assessments of the arecanuts. It is necessary to investigate several evolutionary-based algorithms, each of which has the potential to provide superior outcomes while simultaneously reducing the total number of generations required for the process of training.

DATA AND MATERIALS AVAILABILITY

X-ray sample images are provided and a complete set of X-ray images is available upon reasonable request at the following GitHub repository:
https://github.com/PraveenMNaik/Xray-Arecanut-Dataset

ACKNOWLEDGMENT

The authors extend their sincere gratitude to the New Medical Centre, Kundapur, Udupi, India, for generously providing them with the facility to capture X-ray images of arecanuts for their research article. Their contribution is instrumental in the progress and success of their study. The availability of their facility enabled them to gather essential data that formed the basis of their analysis and findings. They greatly appreciate their support and recognition of the importance of scientific research in advancing knowledge and understanding. Without their cooperation, their research would not have been possible, and they are truly thankful for their support.

REFERENCES

[1] S. Kusumadhara, M. S. Ravikumar, and P. Raghavendra, "A framework for grading of White Chali Type arecanuts with machine learning algorithms," *Int. J. Recent Technol. Eng.*, vol. 8, no. 6, pp. 2782–2789, Mar. 2020.

[2] M. K. Pushparani, V. Kumar, and A. Gubbi, "Arecanut grade analysis using image processing techniques," *Int. J. Recent Technol. Eng.*, vol. 7, no. 10, pp. 1–6, 2019.

[3] N. K. Bharadwaj and N. V. Kumar, "Classification and grading of arecanut using texture based block-wise local binary patterns," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 11, pp. 575–586, 2021.

[4] H. Chandrashekhara, "Classification of arecanut using neural networks with feed-forward techniques," *Int. J. Res. Advent Technol.*, vol. 7, no. 3, pp. 998–1003, 2019.

[5] A. Danti and Suresha, "Segmentation and classification of raw arecanuts based on three sigma control limits," *Proc. Technol.*, vol. 4, pp. 215–219, Jan. 2012.

[6] K.-Y. Huang, "Detection and classification of areca nuts with machine vision," *Comput. Math. Appl.*, vol. 64, no. 5, pp. 739–746, Sep. 2012.

[7] S. Siddesha, S. K. Niranjan, and V. N. M. Aradhya, "Texture based classification of arecanut," in *Proc. Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Oct. 2015, pp. 688–692.

[8] S. Patil, A. Naik, M. Sequeira, G. Naik, and J. Parab, "An algorithm for pre-processing of areca nut for quality classification," in *Proc. 2nd Int. Conf. Image Process. Capsule Netw.*, J. I.-Z. Chen, J. M. R. S. Tavares, A. M. Iliyasu, and K.-L. Du, Eds. Cham, Switzerland: Springer, 2022, pp. 79–93.

[9] A. Danti and M. Suresha, "Effective multiclassifier for arecanut grading," in *Wireless Networks and Computational Intelligence*. Berlin, Germany: Springer, 2012, pp. 350–359.

[10] A. Dell'Aquila, "Digital imaging information technology applied to seed germination testing. A review," *Agronomy Sustain. Develop.*, vol. 29, no. 1, pp. 213–221, Mar. 2009.

[11] C. B. D. Silva, V. D. J. M. Bianchini, A. D. D. Medeiros, M. H. D. D. Moraes, A. G. Marassi, and A. Tannús, "A novel approach for *Jatropha curcas* seed health analysis based on multispectral and resonance imaging techniques," *Ind. Crops Products*, vol. 161, Mar. 2021, Art. no. 113186.

[12] A. Rahman and B.-K. Cho, "Assessment of seed quality using non-destructive measurement techniques: A review," *Seed Sci. Res.*, vol. 26, no. 4, pp. 285–305, Dec. 2016.

[13] Y. Xia, Y. Xu, J. Li, C. Zhang, and S. Fan, "Recent advances in emerging techniques for non-destructive detection of seed viability: A review," *Artif. Intell. Agricult.*, vol. 1, pp. 35–47, Mar. 2019.

[14] D. A. Casasent, M. A. Sipe, T. F. Schatzki, P. M. Keagy, and L. C. Lee, "Neural net classification of X-ray pistachio nut data," *LWT, Food Sci. Technol.*, vol. 31, no. 2, pp. 122–128, Mar. 1998.

[15] I. R. Donis-González, D. E. Guyer, D. W. Fulbright, and A. Pease, "Postharvest noninvasive assessment of fresh chestnut (*Castanea* spp.) internal decay using computer tomography images," *Postharvest Biol. Technol.*, vol. 94, pp. 14–25, Aug. 2014.

[16] I. R. Donis-González, D. E. Guyer, and A. Pease, "Postharvest noninvasive classification of tough-fibrous asparagus using computed tomography images," *Postharvest Biol. Technol.*, vol. 121, pp. 27–35, Nov. 2016.

[17] V. Ashok, "Combining discriminant analysis and neural networks for detection of internal defects in mangoes using X-ray imaging technique," *Int. J. Innov. Technol. Exploring Eng.*, vol. 9, pp. 188–194, Dec. 2019.

[18] M. R. Ahmed, J. Yasmin, E. Park, G. Kim, M. S. Kim, C. Wakholi, C. Mo, and B.-K. Cho, "Classification of watermelon seeds using morphological patterns of X-ray imaging: A comparison of conventional machine learning and deep learning," *Sensors*, vol. 20, no. 23, p. 6753, Nov. 2020.

[19] A. D. D. Medeiros, D. T. Pinheiro, W. A. Xavier, L. J. D. Silva, and D. C. F. D. S. Dias, "Quality classification of *Jatropha curcas* seeds using radiographic images and machine learning," *Ind. Crops Products*, vol. 146, Apr. 2020, Art. no. 112162.

[20] T. Van De Looverbosch, M. H. R. Bhuiyan, P. Verboven, M. Dierick, D. Van Loo, J. De Beenbouwer, J. Sijbers, and B. Nicolaï, "Nondestructive internal quality inspection of pear fruit by X-ray CT using machine learning," *Food Control*, vol. 113, Jul. 2020, Art. no. 107170.

[21] S. Thomas and J. Thomas, "Non-destructive silkworm pupa gender classification with X-ray images using ensemble learning," *Artif. Intell. Agricult.*, vol. 6, pp. 100–110, 2022.

[22] G. Jocher et al., "Ultralytics/YOLOv5: V7.0—YOLOv5 SOTA realtime instance segmentation," Ultralytics, Los Angeles, CA, USA, Tech. Rep., Nov. 2022. [Online]. Available: https://github.com/ultralytics/yolov5

[23] G. Gao, S. Wang, C. Shuai, Z. Zhang, S. Zhang, and Y. Feng, "Recognition and detection of greenhouse tomatoes in complex environment," *Traitement Signal*, vol. 39, no. 1, pp. 291–298, Feb. 2022.

[24] W. Han, F. Jiang, and Z. Zhu, "Detection of cherry quality using YOLOV5 model based on flood filling algorithm," *Foods*, vol. 11, no. 8, pp. 1–9, 2022.

[25] O. M. Lawal, Z. Huamin, and Z. Fan, "Ablation studies on YOLOFruit detection algorithm for fruit harvesting robot using deep learning," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 922, no. 1, Nov. 2021, Art. no. 012001.

[26] L. Wang, Y. Zhao, Z. Xiong, S. Wang, Y. Li, and Y. Lan, "Fast and precise detection of litchi fruits for yield estimation based on the improved YOLOv5 model," *Frontiers Plant Sci.*, vol. 13, pp. 1–16, Aug. 2022.

[27] S. Lyu, R. Li, Y. Zhao, Z. Li, R. Fan, and S. Liu, "Green citrus detection and counting in orchards based on YOLOv5-CS and AI edge system," *Sensors*, vol. 22, no. 2, pp. 1–20, 2022.

[28] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang, and X. Li, "A real-time detection algorithm for kiwifruit defects based on YOLOv5," *Electronics*, vol. 10, no. 14, p. 1711, Jul. 2021.

[29] B. Yan, P. Fan, X. Lei, Z. Liu, and F. Yang, "A real-time apple targets detection method for picking robot based on improved YOLOv5," *Remote Sens.*, vol. 13, no. 9, p. 1619, Apr. 2021.

[30] Z. Sun, X. Leng, Y. Lei, B. Xiong, K. Ji, and G. Kuang, "BiFA-YOLO: A novel YOLO-based method for arbitrary-oriented ship detection in high-resolution SAR images," *Remote Sens.*, vol. 13, no. 21, p. 4209, Oct. 2021.

[31] I. S. Isa, M. S. A. Rosli, U. K. Yusof, M. I. F. Maruzuki, and S. N. Sulaiman, "Optimizing the hyperparameter tuning of YOLOv5 for underwater detection," *IEEE Access*, vol. 10, pp. 52818–52831, 2022.

[32] Z. Wang, L. Jin, S. Wang, and H. Xu, "Apple stem/calyx real-time recognition using YOLO-v5 algorithm for fruit automatic loading system," *Postharvest Biol. Technol.*, vol. 185, Mar. 2022, Art. no. 111808.

[33] A. J. Mantau, I. W. Widayat, Y. Adhitya, S. W. Prakosa, J.-S. Leu, and M. Köppen, "A GA-based learning strategy applied to YOLOv5 for human object detection in UAV surveillance system," in *Proc. IEEE 17th Int. Conf. Control Autom. (ICCA)*, Jun. 2022, pp. 9–14.

[34] *Labelimg, Free Software: MIT License*, Tzutalin, 2015. Accessed: Mar. 13, 2023. [Online]. Available: https://github.com/

[35] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision—ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 346–361.

[37] B. Xiong, Z. Sun, J. Wang, X. Leng, and K. Ji, "A lightweight model for ship detection and recognition in complex-scene SAR images," *Remote Sens.*, vol. 14, no. 23, p. 6053, Nov. 2022.

[38] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," Tech. Rep., 2019. [Online]. Available: https://arxiv.org/abs/1911.11907
[39] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[40] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for NAS," 2019, *arXiv:1912.06059*.
[41] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Netw.*, vol. 107, pp. 3–11, Nov. 2018.
[42] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.
[43] P. M. Naik and B. Rudra, "Flower phenotype recognition and analysis using YOLOv5 models," in *Proc. 13th Int. Conf. Adv. Comput., Control, Telecommun. Technol. (ACT)*, vol. 8, 2022, pp. 838–848.
[44] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.
[45] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. Accessed: Jun. 23, 2023. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
[46] M. van Dael, S. Lebotsa, E. Herremans, P. Verboven, J. Sijbers, U. L. Opara, P. J. Cronje, and B. M. Nicolaï, "A segmentation and classification algorithm for online detection of internal disorders in citrus using X-ray radiographs," *Postharvest Biol. Technol.*, vol. 112, pp. 205–214, Feb. 2016.
[47] M. van Dael, P. Verboven, J. Dhaene, L. Van Hoorebeke, J. Sijbers, and B. Nicolai, "Multisensor X-ray inspection of internal defects in horticultural products," *Postharvest Biol. Technol.*, vol. 128, pp. 33–43, Jun. 2017.
[48] A. D. de Medeiros, R. C. Bernardes, L. J. da Silva, B. A. L. de Freitas, D. C. F. D. S. Dias, and C. B. da Silva, "Deep learning-based approach using X-ray images for classifying *Crambe abyssinica* seed quality," *Ind. Crops Products*, vol. 164, Jun. 2021, Art. no. 113378.
[49] T. Van De Looverbosch, E. Raeymaekers, P. Verboven, J. Sijbers, and B. Nicolaï, "Non-destructive internal disorder detection of conference pears by semantic segmentation of X-ray CT scans using deep learning," *Expert Syst. Appl.*, vol. 176, Aug. 2021, Art. no. 114925.

PRAVEEN M. NAIK received the B.E. degree in information science and engineering and the M.Tech. degree in computer science and engineering from Visvesvaraya Technological University, Belagavi, Karnataka, India, in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree with the National Institute of Technology Karnataka, Surathkal, India. He was teaching with the Department of Computer Science and Engineering, Visvesvaraya Technological University, till 2020. His research interests include image processing, the IoT, blockchain, machine learning, and model optimization.

BHAWANA RUDRA (Member, IEEE) received the Ph.D. degree in information technology from the Indian Institute of Information Technology Allahabad, in March 2015. She is currently an Assistant Professor with the Department of Information Technology, National Institute of Technology Karnataka, India. Her current research has mostly concentrated on the development and evaluation of security protocols for future internet systems, specifically about service-oriented network architecture (SONATE) within a collaborative environment and middleware. Her research interests include future internet architectures, network protocols, routing security, quality of service, security in wireless networks, loosely coupled protocols and their security, service and composition security, attribute-based authentication, vector-based identification, authorization, confidentiality, integrity, and availability of the resources, the Internet of Things, blockchain security, quantum computing, and quantum cryptography. She is a member of the Board of Studies at the department level in various colleges and a Life Member of various professional societies. In 2017 and 2018, the International Institute of Organized Research (I2OR) and Green Thinkerz—Australia and India named her an Emerging Researcher and a Bright Educator. She is a reviewer of various conferences and journals.