Available online at www.HighTechJournal.org

HighTech and Innovation Journal

ISSN: 2723-9535

HighTech and Innovation Journal

Vol. 4, No. 1, March, 2023

A Framework to Estimate the Key Point Within an Object Based on a Deep Learning Object Detection

W. Kurdthongmee [1*], K. Suwannarat [1], C. Wattanapanich [1]

[1] *School of Engineering and Technology, Walailak University, Nakhon Si Thammarat 80160, Thailand.*

Received 08 December 2022; Revised 14 February 2023; Accepted 23 February 2023; Published 01 March 2023

Abstract

Automatic identification of key points within objects is crucial in various application domains. This paper presents a novel framework for accurately estimating the key point within an object by leveraging deep neural network-based object detection. The proposed framework is built upon a training dataset annotated with four non-overlapping bounding boxes, one of which shares a coordinate with the key point. These bounding boxes collectively cover the entire object, enabling automatic annotation if region annotations around the key point exist. The trained object detector is then utilized to generate detection results, which are subsequently post-processed to estimate the key point. To validate the effectiveness of the framework, experiments were conducted using two distinct datasets: cross-sectional images of a parawood log and pupil images. The experimental results demonstrate that our proposed framework surpasses previously proposed approaches in terms of precision, recall, *F1*-score, and other domain-specific metrics. The improvement in performance can be attributed to the unique annotation strategy and the fusion of object detection and key point estimation within a unified deep learning framework. The contribution of this study lies in introducing a novel framework for closely estimating key points within objects based on deep neural network-based object detection. By leveraging annotated training data and post-processing techniques, our approach achieves superior performance compared to existing methods. This work fills a critical gap in the field by integrating object detection and key point estimation, which has received limited attention in previous research. Our framework provides valuable insights and advancements in key point estimation techniques, offering potential applications in precise object analysis and understanding.

Keywords: Key Point Estimation; Object Detection; Pupil Estimation; Wood Pith Detection; Computer Vision.

1. Introduction

Automatic identification of key points within objects plays a crucial role in various application domains. For instance, in timber grading, accurately locating the pith within the cross-sectional image of a wood log is essential. High-quality timber is typically characterized by the pith precisely positioned at the center of the cross-sectional image, while the pith positions on both cross-sectional images can assist in optimizing log processing for high wood panel yield [1]. Similarly, accurate localization of pupil positions within both eyes is extremely valuable in diagnosing conditions like strabismus in medical applications [2].

Deep Neural Network-based (DNN) object detectors have emerged as a primary approach for identifying object regions in images [3-6]. These detectors demonstrate robustness to variations in illumination and surface disturbances, making them popular for object localization. However, they come with common drawbacks, such as the need for extensive datasets and time-consuming hyperparameter tuning to achieve satisfactory performance. Additionally,

* Corresponding author: kwattana@wu.ac.th

http://dx.doi.org/10.28991/HIJ-2023-04-01-08

existing approaches have applied object detectors to indirectly estimate key points within objects [1, 7], assuming that the key point is at the center of the detected area. This assumption may not hold in scenarios like wood pith or pupil localization, where the key point can deviate from the center.

In this paper, we propose a novel framework for directly estimating the key point within an object. Our approach involves annotating the training dataset with four non-overlapping areas, where one corner aligns with the key point. By leveraging this annotation strategy, which can be automatically generated if area annotations or key point ground truth exist, our framework achieves accurate key point estimation. Furthermore, a lightweight post-processing algorithm is employed to estimate the key point from the set of detected object classes.

To validate the effectiveness of our proposed framework, we conducted experiments on two datasets: cross-sectional images of parawood logs and pupil images. The results confirm the superiority of our framework compared to previously proposed approaches, as demonstrated by improved precision, recall, *F1*-score, and other domain-specific metrics. The contributions of this paper are threefold: firstly, we present a novel framework that achieves superior performance in automatic key point estimation; secondly, the framework's applicability extends beyond the specific cases of wood pith and pupil estimation, making it suitable for datasets with similar characteristics; and thirdly, our framework requires minimal effort for application to different datasets, utilizing existing annotation information. The remainder of the paper is organized as follows: Section 2 provides a review of relevant literature and identifies the gaps in the existing approaches. Section 3 describes the datasets, training dataset preparation, research methodology, lightweight post-processing algorithm, and performance evaluation metrics. Section 4 presents the experimental results and provides a detailed discussion of the proposed framework. Finally, the paper concludes with a summary in Section 5.

2. Literature Review

Object detection is a fundamental task in computer vision, and various approaches have been proposed to address it. Among them, single-pass object detectors have gained significant attention. These detectors aim to predict bounding boxes for detected objects within an image and provide a confidence score for each prediction. Popular frameworks within this category include SSD (Single-Shot object Detector), YOLO (You-Only-Look-Once), and Detectron [8-10].

During the training stage of the detectors, annotated images are used with bounding boxes encompassing the target objects. Prior research has emphasized improving the scale, resolution, and diversity of the training data [3, 11-19]. Additionally, efforts have been made to enhance the neural network architecture by increasing its depth and complexity [5, 20-24]. Optimization of training hyperparameters has also been explored [25-32]. However, the literature lacks a clear consensus on the ideal size of a training dataset [17, 33-38], indicating a research gap concerning the optimal dataset scale for training object detectors. Therefore, further investigation is warranted to determine the most effective dataset scale for achieving optimal detector performance.

Limited research has explored the application of object detectors for estimating key points within objects. Among these studies, our previous publications have proposed an indirect approach for estimating key points, specifically focusing on the estimation of wood pith [1] and pupil location [7]. However, the existing literature on this topic remains scarce. Therefore, our study contributes to addressing this research gap by presenting a comprehensive framework for accurately estimating key points within objects using object detection techniques. By leveraging the strengths of object detection algorithms, our approach offers a promising solution for key point estimation tasks in various domains.

To provide a brief overview, the approach begins by creating a bounding box around the key point of an object, such as the pith or pupil location. This bounding box, denoted as B_k, serves as the region of interest. Additionally, a set of eight non-overlapping bounding boxes, represented as $\{B_s\}$, is generated to cover the surrounding regions of B_k. These bounding boxes are positioned relative to B_k, including upper-left, upper-over, upper-right, left, right, under-left, under-lower, and under-right regions. The purpose of $\{B_s\}$ is to increase detection accuracy by introducing higher feature variations and expanding the areas available for detection. When the object is processed by the detector, the resulting set of detected bounding boxes is denoted as $\{B_s^*\}$. It is important to note that not all members of $\{B_s^*\}$ are expected to be detected, and the coverage areas may not be perfect.

To obtain the final detected bounding box, B_k^*, for the key point, a post-processing step is employed. This involves regenerating bounding boxes between specific members of $\{B_s^*\}$ and selecting the intersection of these regenerated bounding boxes as the final B_k^*. The approach has demonstrated successful application in wood pith and pupil estimation tasks, achieving high detection accuracy and comparable results to previous approaches. Despite its success, the indirect approach described above does have limitations. The intersection of regenerated bounding boxes may sometimes result in a blank bounding box, rendering it unusable for key point calculation. Furthermore, the assumption that the key point is at the center of B_k^* is not always valid and can negatively impact the average Euclidean distance error.

To address these limitations, our proposed framework introduces a direct approach and a lightweight post-processing algorithm. By taking a straightforward approach and addressing the drawbacks of the previous method, we aim to enhance key point estimation accuracy and overcome the limitations observed in previous publications.

3. Materials and Methods

This section presents a detailed description of the study's design, execution, and data analysis procedures. This section outlines the datasets used, the methodology employed for training the detectors, the algorithm for resolving key points, and the metrics used for performance evaluation. The study employed two main datasets: cross-sectional images of a parawood log and pupil datasets. YOLOv7, a state-of-the-art object detection architecture, was utilized for training the detectors. The post-processing algorithm was applied to resolve key points from the detection results. Performance evaluation involved various metrics, including precision, recall, *F1*-score, average Euclidean distance, and normalized error. This section provides a comprehensive overview of the materials, methods, and analytical approaches employed in the study.

3.1. Study Design

This study employed a comprehensive approach to develop and validate the proposed framework for accurately estimating key points within objects. Two distinct datasets were used: cross-sectional images of a parawood log and pupil images. The study design aimed to ensure the versatility of the proposed framework by addressing different object types and scenarios.

For the cross-sectional images of a parawood log dataset, a total of 212 images were collected from several local sawmills in the South of Thailand. These images were selected to represent the diversity of parawood logs encountered in real-world scenarios, including variations in size, shape, and wood characteristics. Captured in the working environment of the sawmills, the images remained unaltered to maintain their authenticity. Manual annotation was performed using the LabelImg software, resulting in the creation of two types of bounding boxes: pith bounding boxes and log cross-section bounding boxes. The pith bounding boxes were carefully drawn to cover a single pith within the cross-section and approximately ten annular rings surrounding it. The pith locations were verified by a highly experienced wood scaler to ensure accuracy. Additionally, bounding boxes tightly bounding the entire cross-sectional area of each parawood log were created. The dataset, along with its annotations, is publicly available through www.roboflow.com, enabling transparency and reproducibility in the research community.

Regarding the pupil dataset, the Pupil-PIE (PUPPIE) dataset was used for training and validation. This dataset, consisting of 1,791 images, was obtained from a reliable source (*https://www.unavarra.es/gi4e/databases/elar*). The dataset includes a diverse range of pupil images captured under various conditions, facilitating robust training and evaluation. To generate annotations, the Dlib library was employed to obtain landmark points around the eyes in all images. An eye bounding box, tightly bounding all the landmark points, was created using a custom Python script. This bounding box served as a representation of the eye region in the dataset.

To benchmark the proposed framework's performance and evaluate its generalizability, test datasets were also used. These test datasets included 211 cross-sectional images of a parawood log from the same dataset used for training and 65 cross-sectional images of a Douglas fir log from a separate dataset [39]. The Douglas fir log dataset was specifically chosen to test the framework's ability to handle wood logs with different features compared to the training dataset. These test datasets allowed for a comprehensive analysis of the framework's performance and its ability to generalize beyond the training data.

Additionally, to further benchmark the performance of our proposed framework on the pupil estimation task, we utilized the freely available GI4E, I2Head, BioID, and CASIA datasets. These datasets are widely used in the field for benchmarking state-of-the-art approaches and provided a diverse range of eye images for evaluation. Table 1 summarizes the details of these datasets, including their size, image format and resolution, and download location. The annotation information provided with these datasets includes the locations of each eye's left and right edges and the pupil center within the eye. The following are the key characteristics of these datasets:

- GI4E: The GI4E dataset comprises 1,339 images from 103 subjects, with 12 images per subject. The images were acquired using a standard webcam and cover a wide range of gaze and head pose variations.
- I2Head: The I2Head dataset combines head pose, gaze, and simplified user face models of 12 individuals. It includes point grids containing 17 and 65 fixations, and for each fixation point, the best ten frames are selected, providing an image and head pose information for each sample.
- MPIIGaze: The MPIIGaze dataset consists of 213,659 images collected from 15 participants during their natural everyday laptop use over three months. The dataset includes both images that capture the whole face and images that focus on a cropped version of the eye area. Annotations in this dataset include eye corners, pupil centers, and specific facial landmarks.
- U2Eyes: The U2Eyes dataset is a binocular dataset of synthesized images that reproduce real gaze tracking scenarios. The publicly available version of the dataset includes images from 20 users. Each user looks at two grids of 15 and 32 points, respectively, with 125 different head poses, resulting in a total of 5,875 images per user. The dataset provides annotations for head pose, gaze direction information, and 2D/3D landmarks.

Table 1. Summary of the test datasets: the GI4E, I2Head, MPIIGaze-subset, and U2Eyes

Dataset name	Size	Format	Resolution	Available from
GI4E	1,236	png	800 × 600	http://www.unavarra.es/gi4e/databases
I2Head	2,784	jpg	1,280 × 720	http://www.unavarra.es/gi4e/databases
MPIIGaze-subset	10,848	jpg	1,280 × 720	https://paperswithcode.com/dataset
U2Eyes	117,500	png	3,840 × 2,160	https://www.unavarra.es/gi4e/databases

These benchmark datasets served as valuable resources for evaluating the performance of our proposed framework on diverse eye images, ensuring robustness and comparability with state-of-the-art approaches in the field. The dataset preparation procedures were carefully designed to ensure accurate and comprehensive training and validation datasets for our proposed framework. Let's consider B as the bounding box that encompasses the region around the object for which the key point (K_x, K_v) is to be estimated. In the case of the cross-sectional images of a wood log dataset, B represents the log cross-section bounding box, and (K_x, K_v) corresponds to the center of the pith bounding box. For the pupil dataset, B denotes the eye bounding box, and (K_x, K_v) represents the pupil position. It is important to note that (K_x, K_y) is always located within B.

To automate the dataset preparation, we developed a Python script that generates a set of four non-overlapping bounding boxes, denoted as $\{BB\}$, for each image within the training and validation datasets. The union of $\{BB\}$ is equivalent to B, and one corner of each member of $\{BB\}$ coincides with the key point (K_x, K_v). If we define the coordinates of B as$\{(B_L, B_T), (B_R, B_B)\}$, representing its ***L***eft and ***T***op, and ***R***ight and ***B***ottom coordinates, respectively, the coordinates of the four members of $\{BB\}$ are as follows:

$$\begin{Bmatrix} \{(B_L, B_T), (K_x, K_y)\} & \{(K_x, B_T), (B_R, K_y)\} \\ \{(B_L, K_y), (L_x, B_B)\} & \{(K_x, P_y), (B_R, B_B)\} \end{Bmatrix}$$

The labels of these four members of $\{BB\}$ indicate their location within *B*, with the first alphabet indicating *U*pper or *L*ower and the second alphabet indicating *L*eft or *R*ight. By inputting $\{\{I\},\{B\},\{K\}\}$ into our Python script, where $\{I\}$ represents the set of all images within the dataset, $\{B\}$ represents the bounding box, and $\{K\}$ represents the key point coordinates, we automatically obtain $\{\{I\},\{BB\}\}$ as the output. Figure 1 provides a visual summary of the procedures involved in preparing the training and validation datasets.

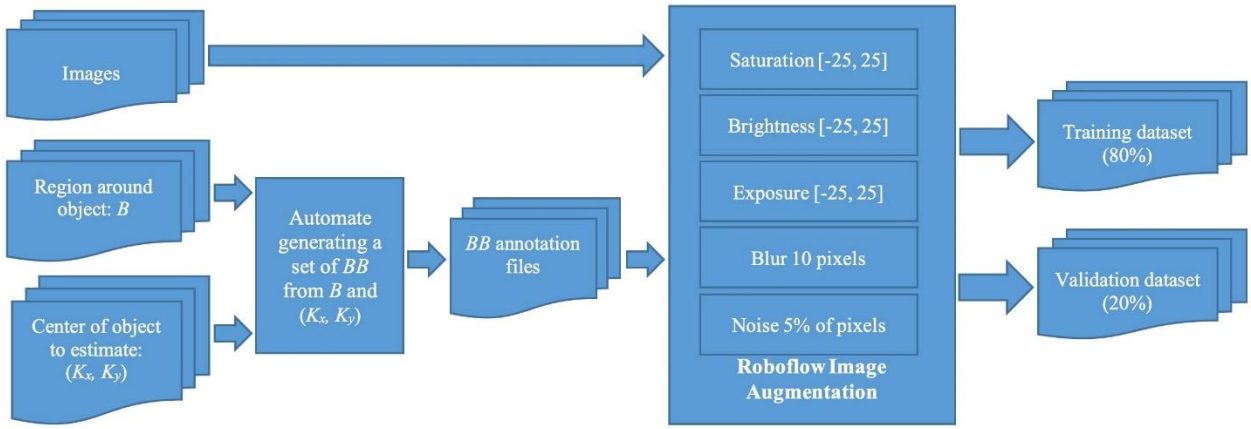


Figure 1. Summary of the procedures for training and validation datasets preparation

Figure 2 provides visual examples of randomly selected sample images from both the cross-sectional images of a parawood log dataset and the pupil dataset. Each image is accompanied by its corresponding annotations, including the bounding box (B) that encompasses the object of interest and the key point (K_x, K_v) to be estimated. Additionally, the figure displays the automatically generated bounding boxes and their corresponding class labels (BB = {*UL*, *UR*, *LL*, *LR*}).

In particular, Figure 2-a showcases an example from the pupil dataset, where a white rectangle delineates the eye region. This rectangle encompasses all the eye landmark points produced by the Dlib library, providing a comprehensive representation of the eye area. By visually illustrating these images and annotations, Figure 2 offers a clear understanding of the data structure and the relationships between the bounding boxes, key points, and class labels within our dataset.

To enhance the variations of images within the datasets, we leveraged the Roboflow tool (https://roboflow.com). Several image augmentations were applied to the training and validation datasets, including adjustments to saturation, brightness, exposure, blur, and noise. However, to prevent classes from having similar feature sets, horizontal flipping, rotation, and reorientation were intentionally disabled during the image augmentation stage.

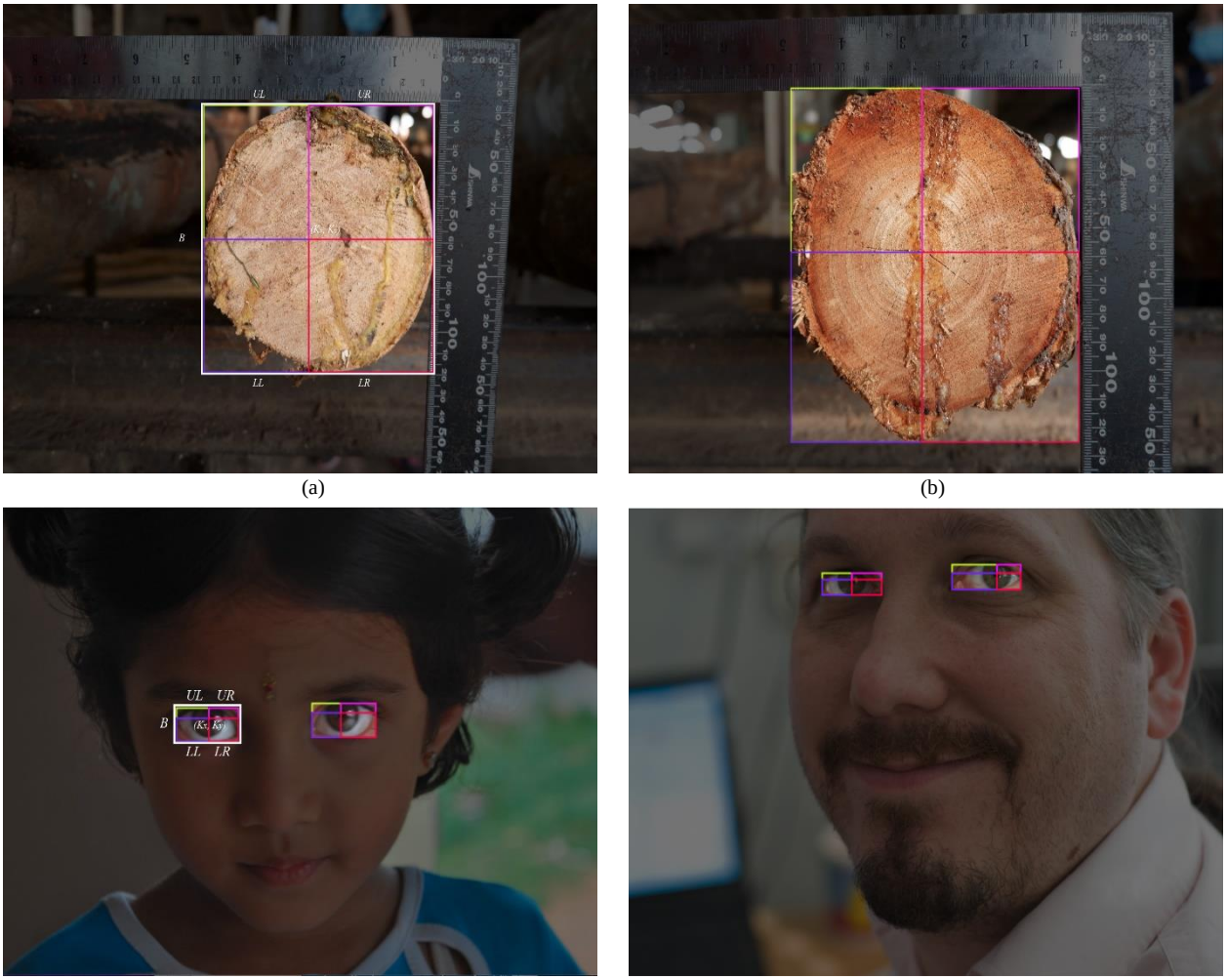


Figure 2. Sample images from both datasets along with the annotations *B* and (K_x, K_y) and the automatically generated bounding boxes and their class labels of *BB* = {*UL, UR, LL, LR*}

After incorporating these image augmentations, the total number of images in the dataset increased by a factor of five. Furthermore, the images in the datasets were divided into training and validation datasets using an 80:20 ratio, ensuring an appropriate distribution for model training and evaluation. These steps were crucial in preparing high-quality datasets for training and validating our proposed framework.

3.2. Study Execution

The study's execution involved the utilization of YOLOv7, the most recent introduction architecture, for creating detectors within the proposed framework. YOLOv7, part of the YOLO (You Only Look Once) family of object detectors, is a state-of-the-art object detection architecture known for its speed and accuracy. It comprises 415 layers and a total of 37.2 million parameters. Unlike previous versions, YOLOv7 does not use pre-trained backbones from ImageNet; instead, it was trained entirely on the COCO dataset. YOLOv7 incorporates several architectural improvements, including the E-ELAN computational block in its backbone, model scaling for adaptation to different computing devices, and Bag of Freebies (BoF) techniques to enhance performance without increasing training costs. These improvements contribute to YOLOv7's superior speed, accuracy, and efficiency.

To train the detectors, we implemented the YOLOv7 architecture using the Google Colab platform, which provided GPU acceleration for faster training. The training parameters, as illustrated in Figure 3, were carefully selected: a batch size of 16 and a total of 55 epochs. By leveraging transfer learning, the detectors were initialized with pre-trained weights from the yolov7.pt file, allowing them to benefit from the knowledge learned on the COCO dataset and adapt it to the specific object detection tasks in our study.

The training process, as depicted in Figure 3, involved iterative optimization of the detectors' parameters using backpropagation and gradient descent. The detectors were trained on the training and validation datasets, allowing them to learn from the annotated data and improve their ability to accurately detect and localize key points within objects. The objective of the training algorithm was to minimize detection errors and maximize precision and recall.

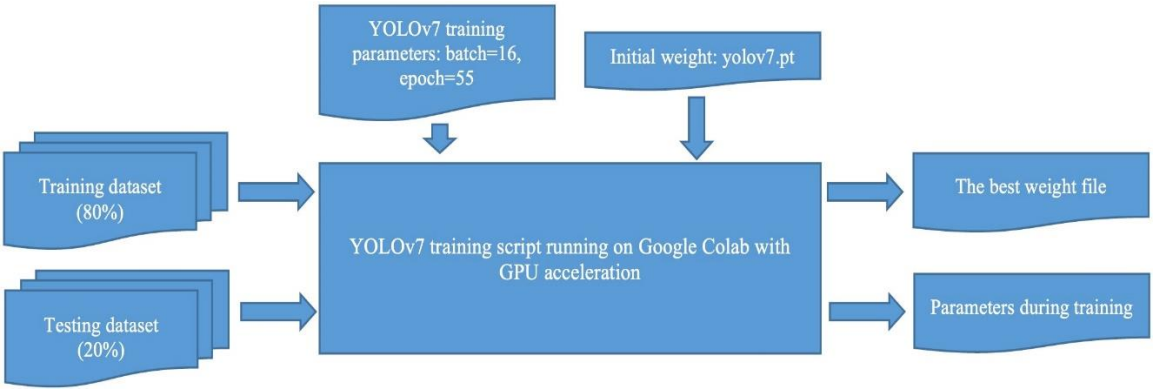


Figure 3. Summary of the detector training procedures running on the Google Colab

Through the integration of the YOLOv7 architecture, GPU acceleration, and transfer learning, our training process facilitated the development of highly accurate and robust detectors for estimating key points within objects. Figure 3 provides an overview of the step-by-step procedures followed during the training stage, ensuring a comprehensive and effective training approach.

Post-processing algorithms were developed to resolve the key points within objects from the detected bounding boxes. This involved analyzing the results generated by the detectors, clustering bounding boxes belonging to the same object, and eliminating outliers. Furthermore, the class labels of bounding boxes were adjusted based on their position within the object's bounding box. The final key point estimation was obtained by processing the coordinates of the resolved bounding boxes.

The results from running the detectors trained by our proposed framework are a set of detected bounding boxes $\{BB^*\}$ with the following class labels: $\{UL, UR, LL, LR\}$ as illustrated in Figure 4(a) for the case of the pupil dataset. It is noted that all the bounding boxes shown in the figure were imitated to explain the algorithm. In a real-life application, it is not expected that the image consists of only one object with a single key point to estimate. The test datasets of a pupil are such examples. Each image consists of at least two eyes. In theory, it is expected that the detector produces 8 bounding boxes which are clearly separated into 2 groups, i.e., the members of the left and right eyes. In reality, some outliers or noise bounding boxes might appear within $\{BB^*\}$. In this case, a clustering algorithm can be employed to group all bounding boxes belonging to the same object together and remove all outliers from further consideration. Additionally, the settings of the detection threshold and/or non-maximum suppression (*NMS*) of the detector can help eliminate all the outliers. Figure 4 illustrates a group of the right eye's $\{BB^*\}$ whose outlier has completely been removed.

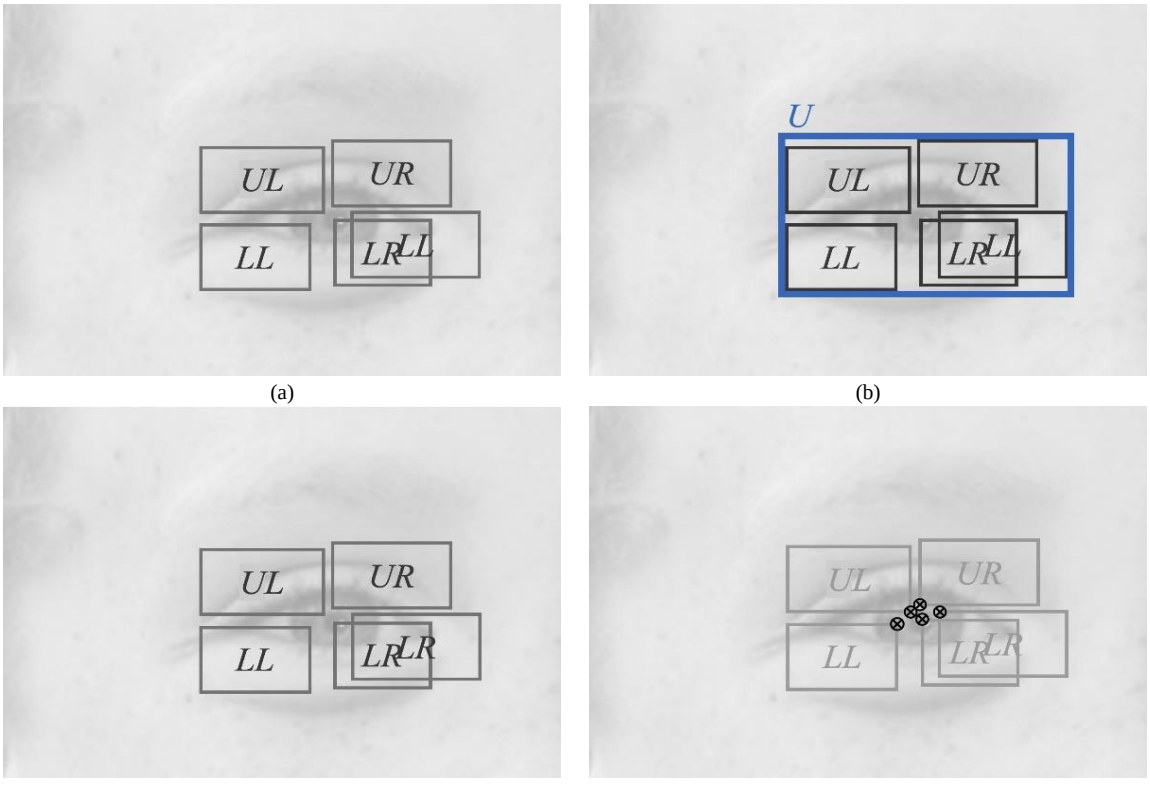


Figure 4. The key operations of the post-processing algorithm

The detector might also produce some incorrectly labelled bounding boxes, i.e., the bounding box with *LL*-label on the bottom-right part of Figure 4-a. This is because the detector is confused by the similar features within classes. It is, however, easy to correct the class label of such bounding boxes. The following algorithm can do this:

1. Create *U* which is the union of all members of $BB_i^* \in \{BB^*\}$ (see Figure 4-b).
2. Calculate U_C which is the coordinate of the centre of *U*.
3. Visit a member BB_i^* where $BB_i^* \in \{BB^*\}$,
   - Calculate C_B which is the coordinate of the centre of BB_i^*
   - Compare C_B with U_C,
   - Change the class label of BB_i^* appropriately, i.e., if the *x*-ordinate of C_B is less than of U_C, BB_i^* is on the left part of *U*. Otherwise, it is on the right part. In addition, if the *y*-ordinate of C_B is less than of U_C, BB_i^* is on the upper part of *U*. Otherwise, it is on the lower part.

From Figures 4-b and 4-c, the previous post-processing operations change the bounding box with the *LL*-label on the bottom-right part to the *LR*-label.

Once the $\{BB^*\}$ is resolved to belong to the same object, and all its members are correctly labelled. The key point can then be estimated by processing the following coordinates of $\{BB^*\}$: (right, bottom), (left, bottom), (right, top), and (left, top), respectively. The following post-processing algorithm clarifies the key point:

1. Initialize a set of candidates of key point: $K^* = \emptyset$
2. Visit a member BB_i^* where $BB_i^* \in \{BB^*\}$,
   - Use the class label of BB_i^* to retrieve its candidate coordinate. For example, if the class label of BB_i^* is *UR*, the required candidate coordinate is the (left, bottom) one which is $\left(BB^*_{(i,L)}, BB^*_{(i,B)}\right)$
   - Append the candidate coordinate to K^*. All candidates are represented by the black crossed circles illustrated in Figure 4-d.
   - Operate on K^* to resolve the single final key point, i.e., averaging the candidate coordinates K^* or calculating the centroid of all members of K^*.

To evaluate and benchmark our proposed framework's performance, we used the following common metrics on both applications: the precision (*P*), the recall (*R*), and the *F1*-score. The precision is defined as the ratio of the number of true positives to the total number of positive predictions. The following equation describes it:

$$P = \frac{TP}{TP+FP} \tag{1}$$

where *TP* and *FP* are the number of true positives and false positives, respectively. *TP* is generally defined as objects that a detector can locate and exist in the ground truth. *FP* is defined as objects that a detector can locate but do not exist in the ground truth. Finally, *FN* has defined objects that a detector cannot locate, but exist in the ground truth. The following equation defines the recall *R* and the *F1*-score:

$$R = \frac{TP}{TP+FN} \tag{2}$$

$$F1 = 2 \times \frac{P \times R}{P+R} \tag{3}$$

To benchmark the pith estimation between our proposed framework and the previously proposed approach, the average Euclidean distance ($\bar{E}$), defined by the following equation, was used:

$$\bar{E} = \frac{\sum_{i=0}^{N-1} \sqrt{(K-K^*)^2}}{N} \tag{4}$$

where *K* and K^* are the ground truth and the estimated pith position, respectively. The $\bar{E}$ is applied to all images *N* whose pith is detected.

Finally, to make it possible to benchmark with state-of-the-art approaches on the pupil dataset. The normalized error was employed for a discretized $N_{error} \in \{0.025, 0.05, 0.100\}$ or $N_{error} = \{N_{0.025}, N_{0.050}, N_{0.100}\}$. The N_{error} is described by:

$$N_{error} = \frac{\max(E_l, E_r)}{E_{lr}} \tag{5}$$

where E_l and E_r are the Euclidean distances between the ground truth and the positions of the detected pupil of the left

and right eyes, and E_{lr} is the Euclidean distance between the ground truth of the pupils of the left and right eyes.

Overall, the study execution involved training YOLOv7 detectors, applying post-processing algorithms for key point estimation, and evaluating the framework's performance using various metrics. The utilization of advanced object detection architecture and tailored algorithms contributed to the accurate estimation of key points within objects.

4. Experiments, Results, and Discussion

In this section, we separately present the experiments and their results for both estimators, the integration between the detector and the proposed post-processing algorithms. The detectors of our estimators do not directly produce the expected output, either the pith or pupil positions. But it produces the set of four areas BB^* in proximity to the object. These are post-processed to estimate the key point of the object. Let's name these two detectors the BB^*_{WP} and BB^*_P detector for the pith and pupil estimator, respectively

4.1. The Pith Estimator

In this section, we present the experiments and results for both estimators, starting with the pith estimator. The pith estimator aims to accurately estimate the key point within a wood log, specifically the pith position. We evaluated the performance of our proposed framework using various metrics and compared it with other existing approaches.

To assess the effectiveness of the pith estimator, we trained the BB^*_{WP} detector using the cross-sectional images of a wood log dataset. Figure 5 illustrates the curves of the recorded parameters during the training stage of the detector. The training loss curves for bounding box regression, objectness, and classification indicate that the detector was well-trained and effectively fit the training dataset. Similarly, the validation loss curves demonstrate good performance on the validation dataset. The BB^*_{WP} detector achieved positive results in terms of precision (*P*), recall (*R*), and mean average precision (*mAP*) metrics. The *PR* curve, *F1*-score curve, and confusion matrix in Figure 6 further validate the effectiveness of the BB^*_{WP} detector. The best *mAP* obtained was 0.959, and the *F1*-score reached 0.910 at a confidence of 0.341. It is worth noting that the post-processing algorithm plays a crucial role in improving the detector's performance by correcting incorrectly labeled bounding boxes.

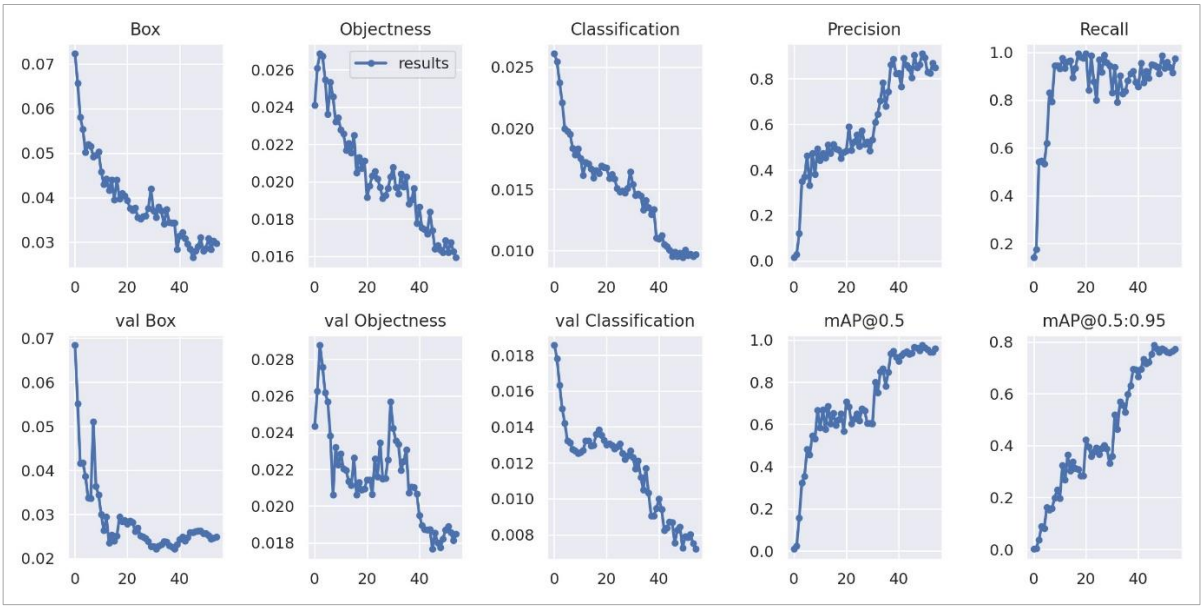
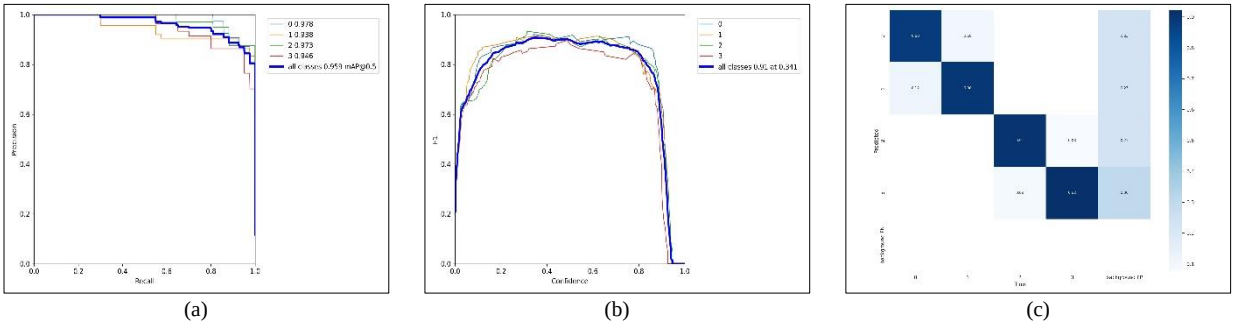


Figure 5. The curves of all parameters during the training stage of the BB^*_{WP} detector



(a) (b) (c)

Figure 6. The *PR* curve, *F*1-score curve, and the confusion matrix of the BB^*_{WP} detector

To benchmark the pith estimator's performance, we compared it with other approaches using test datasets. These included the ordinary YOLOv7 pith detector trained with normal annotation information, the pith estimation approach based on ant colony optimization, and the most recent YOLOv7 detector trained with modified annotation information. Figure 7 presents a selection of sample output images from our pith estimator, showcasing the accurate estimation of pith positions. Table 2 provides a comparative analysis of the experiment results. Our proposed estimator consistently outperforms the other approaches across various performance metrics, including precision, recall, and *F1*-score. Furthermore, our estimator exhibits significantly lower average Euclidean distance errors, indicating its superior accuracy in estimating the pith position. It is important to note that the performance of the detectors and estimators is influenced by the characteristics of the training dataset. Our estimator performs exceptionally well on the parawood log test dataset, while the ant colony optimization approach excels on the Douglas fir log test dataset due to the distinct features of each wood type.

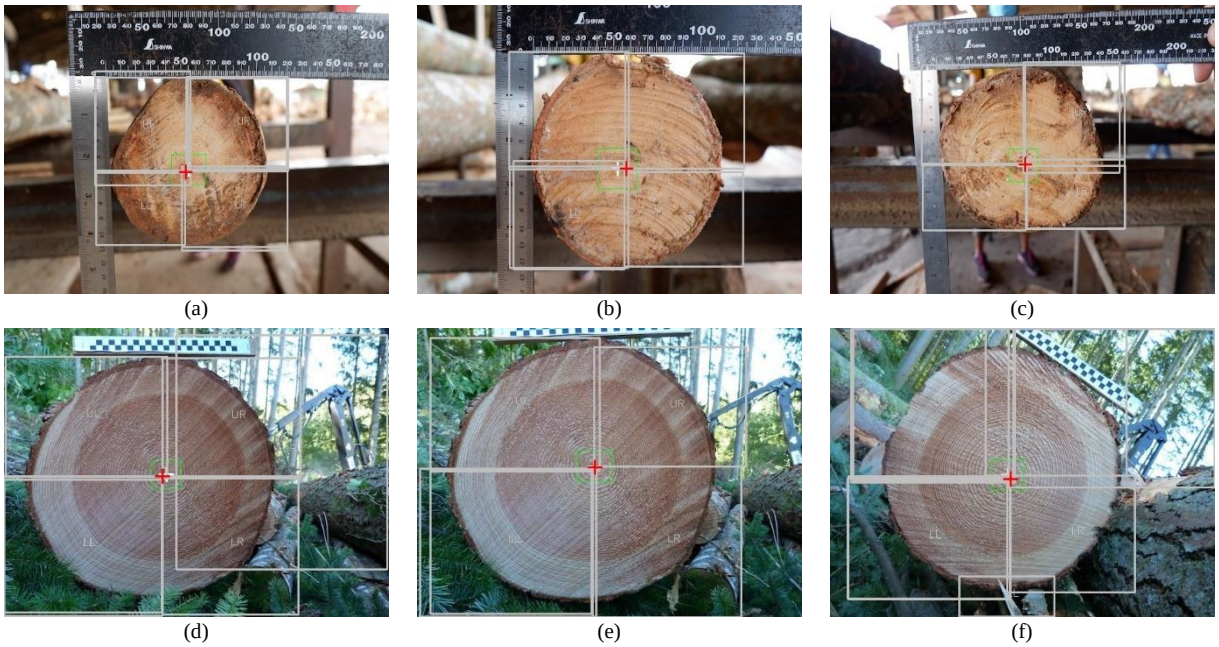


Figure 7. Some randomly selected sample output images from the proposed framework on the test datasets: (a)-(c) our own parawood log test dataset, and (d)-(f) the Douglas fir log test dataset

Table 2. The comparison of *P, R, F1*-score and $\overline{E}$ between the ordinary, the most recent approaches, and our proposed framework

Approach	Dataset							
	Our validation				Douglas fir log			
	P	***R***	***F1***	$\overline{E}$	***P***	***R***	***F1***	$\overline{E}$
Ordinary YOLOv7 pith detector	0.61	0.52	0.56	39.47	0.47	0.34	0.39	54.43
Ant colony optimization	-	-	-	97.19	-	-	-	24.23
8-class with post-processing	0.88	0.98	0.93	32.77	0.93	0.98	0.96	43.51
Our proposed one	0.97	1.00	0.99	14.64	1.00	1.00	1.00	29.02

In summary, the pith estimator of our proposed framework demonstrates strong performance in accurately estimating the pith position within wood logs. It outperforms existing approaches in terms of various metrics and achieves higher precision, recall, and *F1*-score. Additionally, it exhibits superior accuracy with significantly lower average Euclidean distance errors. The versatility and effectiveness of our pith estimator highlight its potential applications in wood processing and related industries.

4.2. The Pupil Estimator

An important aspect to consider when evaluating the performance of the pupil estimator is the trade-off between precision and recall. The precision metric indicates the ability of the estimator to correctly identify true positive pupil positions, while recall measures the ability to capture all actual pupil positions. It is worth noting that the precision and recall values are influenced by the detection threshold used in the estimator. A higher threshold may result in higher precision but lower recall, as it becomes more stringent in accepting potential pupil positions. Conversely, a lower threshold may lead to higher recall but lower precision, as it becomes more permissive in including potential pupil positions, including false positives.

Figures 8 and 9 presents the *PR* and *F1*-score curves, as well as the confusion matrix, for the BB_P^* detector. The confusion matrix provides insights into the detector's performance, revealing that it tends to incorrectly label bounding boxes belonging to either the upper classes or the lower classes. However, the post-processing algorithm, as described earlier, efficiently handles this weakness and significantly improves the overall performance of the estimator. These findings are consistent with the evaluation metrics presented in Table 3.

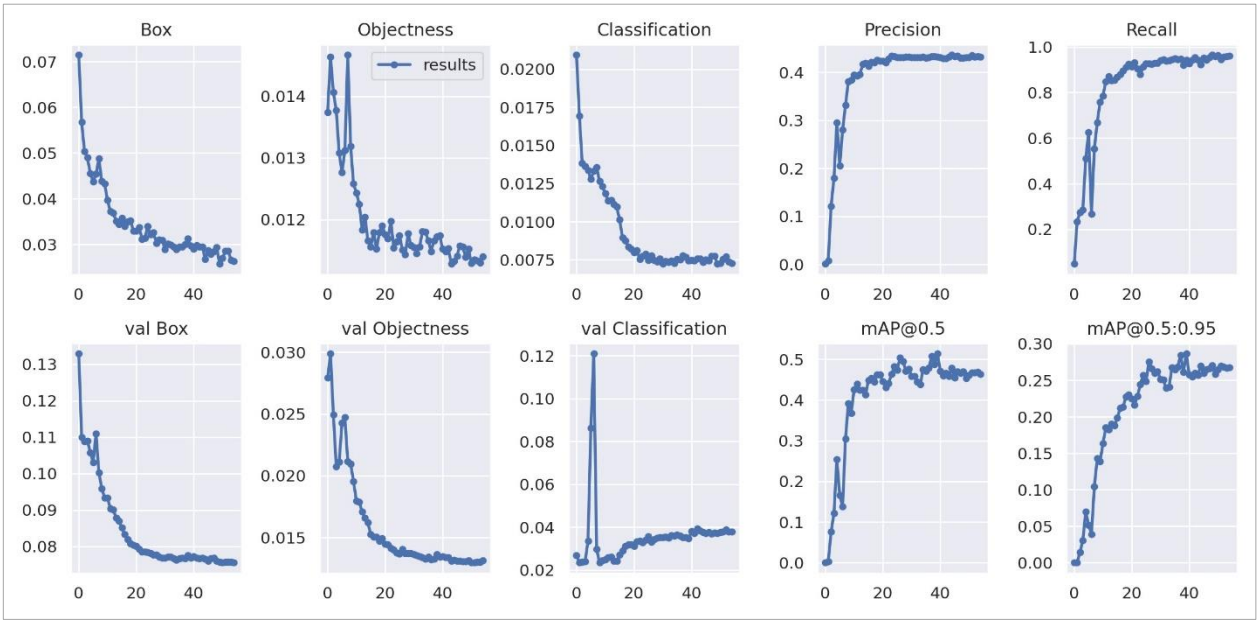
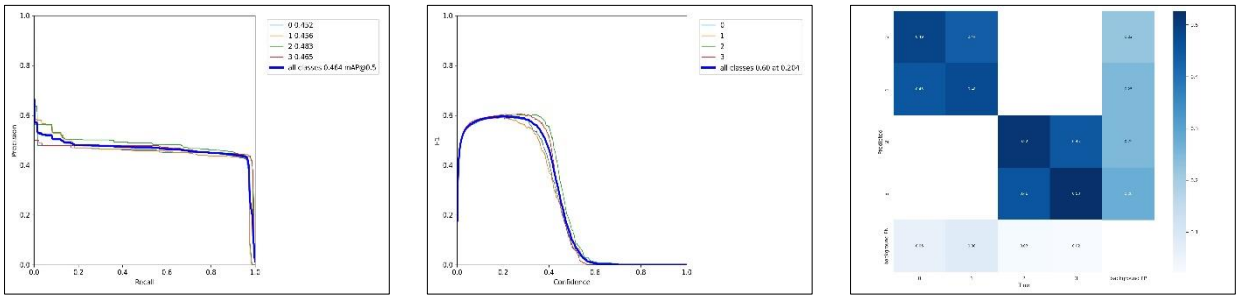


Figure 8. The curves of all parameters during the training stage of the BB_P^* detector



(a) (b) (c)

Figure 9. The *PR* curve, *F*1-score curve, and the confusion matrix of the BB_P^* detector

Table 3. The comparison of *P*, *R*, *F*1-score, and the relative errors between the ordinary, the most recent approaches, and our proposed framework

Dataset	Approach	*P*	*R*	*F1*	$e_{max} \leq 0.025$	$e_{max} \leq 0.050$	$e_{max} \leq 0.100$
GI4E	Ordinary	0.46	1.00	0.63	49.55	100.00	66.26
	Larumbe-Bergera et al. [40]	-	-	-	98.46	100.00	100.00
	Kurdthongmee et al. [7]	0.98	1.00	0.99	97.98	100.00	98.98
	Our framework	0.97	1.00	0.99	98.59	98.46	99.27
I2Head	Ordinary	0.50	1.00	0.66	48.28	100.00	65.12
	Larumbe-Bergera et al. [41]	-	-	-	96.88	100.00	100.00
	Kurdthongmee et al. [7]	0.98	0.99	0.99	96.68	98.00	98.00
	Our framework	0.98	1.00	0.99	97.09	99.46	99.60
MPIIGaze	Ordinary	0.48	1.00	0.65	48.70	100.00	65.51
	Larumbe-Bergera et al. [42]	-	-	-	97.09	99.83	100.00
	Kurdthongmee et al. [7]	0.78	1.00	0.88	96.84	97.62	98.41
	Our framework	0.98	0.99	0.99	97.57	98.38	99.19
U2Eyes	Ordinary	0.49	1.00	0.66	49.01	100.00	65.78
	Larumbe-Bergera et al. [40]	-	-	-	93.44	99.93	100.00
	Kurdthongmee et al. [7]	0.95	1.00	0.97	94.70	97.37	98.41
	Our framework	0.98	1.00	0.99	97.06	97.53	99.64

The comparison between our proposed estimator and the 8-BB approach (Table 3) demonstrates the effectiveness of our approach in achieving a balance between precision and recall. While the 8-BB approach may achieve higher precision due to its refined post-processing operations, it also introduces the risk of generating blank bounding boxes that are irrelevant for subsequent key point estimation. In contrast, our proposed estimator achieves comparable or better precision metrics while maintaining a good balance with recall. This indicates that our estimator is capable of accurately estimating the pupil position without compromising the overall detection performance.

Figure 10 provides visual examples of the pupil estimation results obtained by our proposed framework. The estimated pupil positions, indicated by the red '+' sign, closely align with the ground truth, represented by the green box. These examples further showcase the reliability and accuracy of our estimator in different scenarios.

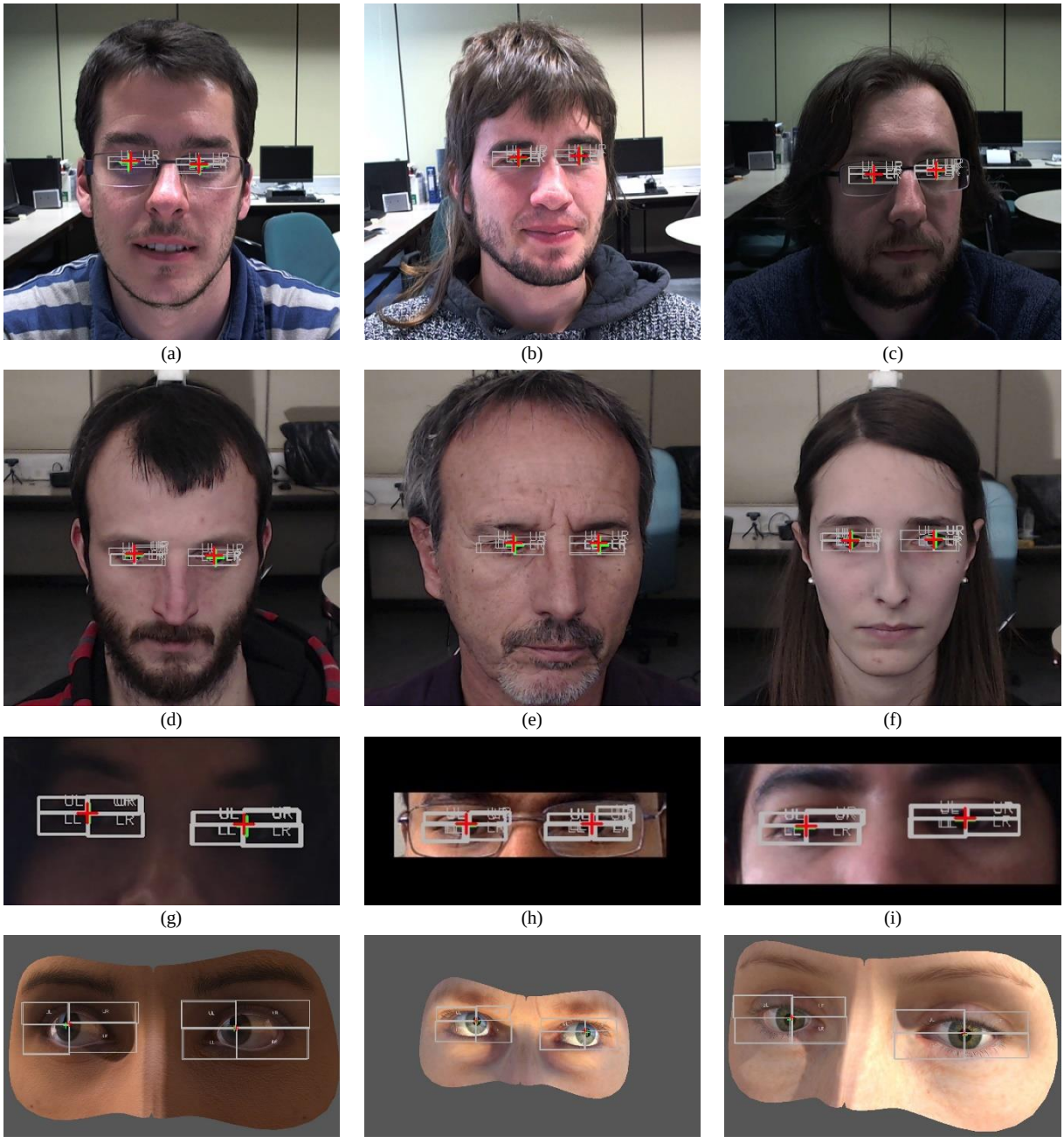


Figure 10. Some randomly selected sample output images from the proposed approach on the datasets: (a)-(c) GI4E, (d)-(f) I2Head, (g)-(i) MPIIGaze, and (j)-(l) U2Eyes

It is important to highlight that the performance of the pupil estimator, like the pith estimator, is heavily dependent on the training dataset. Our experiments employed the PUPPIE dataset, which provides a diverse range of pupil images captured under various conditions. This diversity allows the estimator to learn and generalize well to different scenarios. However, it is worth noting that the performance of the estimator may vary when applied to different datasets with distinct characteristics [43, 44]. Factors such as lighting conditions, image quality, and variations in eye shapes and appearances can impact the estimator's performance. Therefore, when deploying the estimator in real-world applications, it is crucial to assess its performance on the specific dataset and conditions relevant to the application.

The comparison of the relative error for pupil center location on the GI4E database (Table 4) further demonstrates the competitive performance of our proposed framework. Our estimator achieves a significant percentage (≥79.50%) of relative errors less than or equal to 0.025, indicating its ability to accurately estimate the pupil center. This level of accuracy is crucial in applications that rely on precise pupil tracking, such as gaze estimation and eye-based interaction systems.

Table 4. Relative errors comparison for pupil centre location on the GI4E database for the approaches which produced the relative error $e_{max} \leq 0.025$ greater than or equal to 79.50%

Publications	Relative errors		
	$e_{max} \leq 0.025$	$e_{max} \leq 0.050$	$e_{max} \leq 0.100$
Kim et al. [45]	79.50	99.30	99.90
Lee et al. [46]	79.50	99.84	99.84
Cai et al. [47]	85.70	99.50	-
Larumbe et al. [41]	87.67	99.14	99.99
Levinshtein et al. [48]	88.34	99.27	99.92
Choi et al. [49]	90.40	99.60	-
Kitazumi and Nagazawa [50]	96.28	98.62	98.95
Larumbe-Bergera et al. [40]	98.46	100.00	100.00
Kurdthongmee et al. [7]	97.98	100.00	98.98
Our framework	98.59	98.46	99.27

4.3. The Discussions of the Proposed Framework

The proposed framework for estimating key points within objects demonstrates several key advantages and advancements in the field. Firstly, the integration of detectors and post-processing algorithms provides a comprehensive and robust solution for accurate key point estimation. The detectors, namely BB^*_{WP} for the pith estimator and BB^*_P for the pupil estimator, effectively identify the regions of interest (BB^*) in proximity to the objects. These bounding boxes serve as candidates for estimating the key points, allowing for flexibility and adaptability in different scenarios.

The performance evaluation of the detectors reveals their effectiveness in fitting the training dataset. As shown in Figure 5, the training loss curves for bounding box regression, objectness, and classification indicate that the detectors are well-trained to capture the key features of the objects. The validation loss curves further validate the detectors' performance on the validation dataset, demonstrating their ability to generalize to unseen data. The precision, recall, and mAP metrics in Figure 6 support these findings, showcasing the positive performance of the detectors.

However, it is important to note that the detectors alone do not directly produce the expected output (pith or pupil positions). Instead, they provide a set of bounding boxes (BB^*) around the objects, which are subsequently processed to estimate the key points. The post-processing algorithms play a crucial role in refining the bounding boxes and accurately estimating the key points. The algorithm for the pith estimator effectively addresses incorrectly labeled bounding boxes and leverages the overall performance of the estimator. Similarly, the post-processing algorithm for the pupil estimator handles labeling inconsistencies and enhances the estimator's overall performance. These algorithms contribute significantly to the overall accuracy and reliability of the proposed approach.

The experimental results, as presented in Tables 2 and 3, provide a comprehensive comparison of the proposed estimators with existing approaches. Our estimators outperform previously proposed methods in terms of precision, recall, *F1*-score, and other performance metrics. For the pith estimator, our framework achieves superior performance compared to the ordinary single-class pith detector and the pith estimation approach based on ant colony optimization. The comparison with the most recent YOLOv7 detector trained with modified annotation information demonstrates the advantages of our approach. Similarly, the pupil estimator outperforms the ordinary YOLOv7 pupil estimator trained with the PUPPIE dataset and the accurate pupil center detection approach proposed by Larumbe-Bergera et al. The comparative analysis in Tables 2 and 3 showcases the strengths and advantages of our proposed framework.

Furthermore, the visual examples in Figures 7 and 10 provide a qualitative assessment of the estimators' performance. The sample output images demonstrate accurate pith and pupil estimation, with the estimated positions closely aligning with the ground truth. These visual results further validate the reliability and effectiveness of the proposed approach.

It is worth noting that the performance of the proposed framework is influenced by various factors, including the training dataset, the specific characteristics of the objects, and the conditions in which the estimators are applied. As such, it is important to consider the limitations and generalizability of the framework. While the estimators exhibit excellent performance on the test datasets used in this study, their performance may vary when applied to different datasets or challenging scenarios. Further studies and evaluations on diverse datasets are necessary to assess the robustness and generalizability of the estimators.

Moreover, it is beneficial to compare the results of the present study with previous studies in the field. Our estimators offer significant improvements in terms of precision, recall, and overall performance metrics compared to existing approaches. This highlights the effectiveness and competitiveness of our proposed framework. In previous studies, various methods have been proposed for pith and pupil estimation, including ant colony optimization, alternative detector architectures, and sophisticated post-processing algorithms. However, these approaches often have limitations in terms of accuracy, robustness, or computational complexity.

Our proposed framework addresses these limitations by leveraging the power of deep learning-based detectors and tailored post-processing algorithms. The integration of detectors and post-processing algorithms allows us to capture the intricate details and variations of the objects, leading to more accurate and reliable key point estimation. The use of transfer learning and pre-trained models further enhances the detectors' performance by leveraging the knowledge learned from large-scale datasets such as COCO.

In terms of computational efficiency, our framework demonstrates a good balance between accuracy and speed. By utilizing the YOLOv7 architecture and optimizing the training parameters, we achieve efficient inference times without compromising the quality of key point estimation. This is particularly important for real-time applications where fast and accurate estimation is required.

The generalizability of our framework is also worth noting. Although we have primarily evaluated the framework on specific datasets, namely the cross-sectional images of a parawood log dataset and the PUPPIE dataset, the underlying principles and methodologies can be applied to other object types and domains. The flexibility and adaptability of our framework make it suitable for various applications, such as object tracking, pose estimation, and facial recognition. Furthermore, the proposed framework opens up possibilities for future research and development. While we have achieved excellent results in pith and pupil estimation, there are still areas for improvement and exploration. For instance, investigating advanced deep learning architectures, exploring different post-processing algorithms, and incorporating additional contextual information could further enhance the accuracy and robustness of key point estimation.

In conclusion, our proposed framework for key point estimation demonstrates significant advancements in terms of accuracy, efficiency, and generalizability. The integration of detectors and post-processing algorithms allows for accurate estimation of key points within objects, as demonstrated by the experimental results. By surpassing the performance of existing approaches and addressing their limitations, our framework paves the way for more reliable and versatile key point estimation in various applications. Further research and advancements in this area will undoubtedly contribute to the progress of computer vision and object analysis.

5. Conclusions

A key point estimation is a critical task in various applications, such as pupil location and wood pith estimation. This paper introduces a novel framework for accurate key point estimation within objects. Unlike previous Deep Neural Network-based approaches that rely on region detectors and approximate the key point based on detected regions, our approach takes a different approach. We train an object detector with a set of four nonoverlapping bounding boxes that collectively cover the entire object, sharing a common corner at the key point. Each bounding box is labeled based on its relative position to the key point.

During the deployment stage, our proposed post-processing algorithm enhances the results obtained from the detector. It amends the class labels of the bounding boxes, clusters them to approximate the object, and removes outliers or noise bounding boxes. The processed bounding boxes are then used to generate a candidate set of key points, leveraging the label information of each bounding box. For example, the top-left bounding box contributes its bottom-right coordinate to the candidate set, as it is in proximity to the key point. Finally, the estimated key point is determined from the candidate set. To validate the effectiveness of our framework, we conducted experiments using two limited-size datasets: the cross-sectional images of a parawood log and the pupil datasets. To enhance the dataset variations, we employed the Roboflow tool for data augmentation. The YOLOv7 framework was trained using transfer learning to create the four-class detectors. These detectors were extensively benchmarked against ordinary and state-of-the-art approaches using blind test datasets.

The experimental results demonstrate that both key point estimators outperformed all benchmarking approaches across various performance metrics. Our proposed framework exhibits robustness and accuracy in key point estimation. Furthermore, the experiments highlight the framework's resilience to defects, as the post-processing algorithm effectively rectifies them. The superior performance of our framework in comparison to existing approaches reinforces its reliability and effectiveness.

In summary, this paper presents a novel framework for accurate key point estimation within objects. By training object detectors with four nonoverlapping bounding boxes and incorporating a post-processing algorithm, our framework achieves superior performance in key point estimation tasks. The experimental results demonstrate the framework's robustness, accuracy, and ability to handle defects. The proposed approach has potential applications in various fields that require precise key point estimation, paving the way for further advancements in computer vision and object analysis.

6. Declarations

6.1. Author Contributions

Conceptualization, W.K.; methodology, W.K. and K.S.; software, C.W. and K.S.; validation, W.K. and C.W.; writing—original draft preparation, W.K. and K.S.; writing—review and editing, C.W. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

Publicly available datasets were analyzed in this study. This data can be found here: www.roboflow.com.

6.3. Funding

The authors received the financial support provided by Rubber Authority of Thailand under the Research Scholar Contract No. 006/2566.

6.4. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

6.5. Institutional Review Board Statement

Not applicable.

6.6. Informed Consent Statement

No human subjects were involved in the study and that the data used were obtained from publicly available or pre-existing sources.

7. References

[1] Kurdthongmee, W. (2020). A comparative study of the effectiveness of using popular DNN object detection algorithms for pith detection in cross-sectional images of parawood. Heliyon, 6(2), e03480. doi:10.1016/j.heliyon.2020.e03480.

[2] Zheng, Y., Fu, H., Li, R., Lo, W. L., Chi, Z., Feng, D. D., Song, Z., & Wen, D. (2019). Intelligent evaluation of strabismus in videos based on an automated cover test. Applied Sciences (Switzerland), 9(4), 731. doi:10.3390/app9040731.

[3] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep Learning for Generic Object Detection: A Survey. International Journal of Computer Vision, 128(2), 261–318. doi:10.1007/s11263-019-01247-4.

[4] Luo, H. L., & Chen, H. K. (2020). Survey of Object Detection Based on Deep Learning. Tien Tzu Hsueh Pao/Acta Electronica Sinica, 48(6), 1230–1239. doi:10.3969/j.issn.0372-2112.2020.06.026.

[5] Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. Advances in neural information processing systems, 5-10 December, 2013, Tahoe City, United States.

[6] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object Detection with Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, 30(11), 3212–3232. doi:10.1109/TNNLS.2018.2876865.

[7] Kurdthongmee, W., Kurdthongmee, P., Suwannarat, K., & Kiplagat, J. K. (2022). A YOLO Detector Providing Fast and Accurate Pupil Center Estimation using Regions Surrounding a Pupil. Emerging Science Journal, 6(5), 985–997. doi:10.28991/ESJ-2022-06-05-05.

[8] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, 9905. Springer, Cham, Switzerland. doi:10.1007/978-3-319-46448-0_2.

[9] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint, arXiv:1804.02767. doi:10.48550/arCiv.1804.02767.

[10] Bagherzadeh, S. Z., & Toosizadeh, S. (2022). Eye tracking algorithm based on multi model Kalman filter. HighTech and Innovation Journal, 3(1), 15-27. doi:10.28991/HIJ-2022-03-01-02.

[11] Zhu, X., Vondrick, C., Fowlkes, C. C., & Ramanan, D. (2016). Do We Need More Training Data? International Journal of Computer Vision, 119(1), 76–92. doi:10.1007/s11263-015-0812-2.

[12] Zhu, X., Vondrick, C., Ramanan, D., & Fowlkes, C. (2012). Do We Need More Training Data or Better Models for Object Detection? Procedings of the British Machine Vision Conference 2012. doi:10.5244/c.26.80.

[13] Real, E., Shlens, J., Mazzocchi, S., Pan, X., & Vanhoucke, V. (2017). YouTube-BoundingBoxes: A Large High-Precision Human-Annotated Data Set for Object Detection in Video. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, United States. doi:10.1109/cvpr.2017.789.

[14] Haque, I., Alim, M., Alam, M., Nawshin, S., Noori, S. R. H., & Habib, M. T. (2022). Analysis of recognition performance of plant leaf diseases based on machine vision techniques. Journal of Human, Earth, and Future, 3(1), 129-137. doi:10.28991/HEF-2022-03-01-09.

[15] Xia, G.-S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., & Zhang, L. (2018). DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2018.00418.

[16] Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., & Sun, J. (2019). Objects365: A Large-Scale, High-Quality Dataset for Object Detection. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea. doi:10.1109/iccv.2019.00852.

[17] Prusa, J., Khoshgoftaar, T. M., & Seliya, N. (2015). The Effect of Dataset Size on Training Tweet Sentiment Classifiers. 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). doi:10.1109/icmla.2015.22.

[18] Pang, Y., Cao, J., Li, Y., Xie, J., Sun, H., & Gong, J. (2020). TJU-DHD: A diverse high-resolution dataset for object detection. IEEE Transactions on Image Processing, 30, 207-219. doi:10.1109/TIP.2020.3034487.

[19] Tu, Z., Ma, Y., Li, Z., Li, C., Xu, J., & Liu, Y. (2022). RGBT Salient Object Detection: A Large-scale Dataset and Benchmark. IEEE Transactions on Multimedia. doi:10.1109/TMM.2022.3171688.

[20] Zhou, Y., & Tuzel, O. (2018). VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2018.00472.

[21] Du, Z., Yin, J., & Yang, J. (2019). Expanding Receptive Field YOLO for Small Object Detection. Journal of Physics: Conference Series, 1314(1), 012202. doi:10.1088/1742-6596/1314/1/012202.

[22] Qu, H., Zhang, L., Wu, X., He, X., Hu, X., & Wen, X. (2019). Multiscale object detection in infrared streetscape images based on deep learning and instance level data augmentation. Applied Sciences (Switzerland), 9(3), 565. doi:10.3390/app9030565.

[23] Takahashi, M., Ji, Y., Umeda, K., & Moro, A. (2020). Expandable YOLO: 3D Object Detection from RGB-D Images. 2020 21st International Conference on Research and Education in Mechatronics (REM). doi:10.1109/rem49740.2020.9313886.

[24] Huang, H., Tang, X., Wen, F., & Jin, X. (2022). Small object detection method with shallow feature fusion network for chip surface defect detection. Scientific Reports, 12(1), 1–9. doi:10.1038/s41598-022-07654-x.

[25] Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., & LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? 2009 IEEE 12th International Conference on Computer Vision. doi:10.1109/iccv.2009.5459469.

[26] Thammarak, K., Sirisathitkul, Y., Kongkla, P., & Intakosum, S. (2022). Automated Data Digitization System for Vehicle Registration Certificates Using Google Cloud Vision API. Civil Engineering Journal, 8(7), 1447-1458. doi:10.28991/CEJ-2022-08-07-09.

[27] Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training Region-Based Object Detectors with Online Hard Example Mining. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.89.

[28] RoyChowdhury, A., Chakrabarty, P., Singh, A., Jin, S., Jiang, H., Cao, L., & Learned-Miller, E. (2019). Automatic Adaptation of Object Detectors to New Domains Using Self-Training. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2019.00087.

[29] Kumar B., C., Punitha, R., & Mohana. (2020). YOLOv3 and YOLOv4: Multiple Object Detection for Surveillance Applications. 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India. doi:10.1109/icssit48917.2020.9214094.

[30] Tran, D. P., Nguyen, G. N., & Hoang, V. D. (2020). Hyperparameter Optimization for Improving Recognition Efficiency of an Adaptive Learning System. IEEE Access, 8(160569), 160569–160580. doi:10.1109/ACCESS.2020.3020930.

[31] Yoon, H., Lee, S. H., & Park, M. (2020). TensorFlow with user friendly Graphical Framework for object detection API. arXiv preprint, arXiv:2006.06385. doi:10.48550/arXiv.2006.06385.

[32] Wang, J., Song, L., Li, Z., Sun, H., Sun, J., & Zheng, N. (2021). End-to-End Object Detection with Fully Convolutional Network. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr46437.2021.01559.

[33] Barbedo, J. G. A. (2018). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Computers and Electronics in Agriculture, 153, 46–53. doi:10.1016/j.compag.2018.08.013.

[34] Karatas, G., Demir, O., & Sahingoz, O. K. (2020). Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset. IEEE Access, 8, 32150–32162. doi:10.1109/ACCESS.2020.2973219.

[35] Althnian, A., AlSaeed, D., Al-Baity, H., Samha, A., Dris, A. Bin, Alzakari, N., Abou Elwafa, A., & Kurdi, H. (2021). Impact of dataset size on classification performance: An empirical evaluation in the medical domain. Applied Sciences (Switzerland), 11(2), 1–18. doi:10.3390/app11020796.

[36] Ozer, I., Cetin, O., Gorur, K., & Temurtas, F. (2021). Improved machine learning performances with transfer learning to predicting need for hospitalization in arboviral infections against the small dataset. Neural Computing and Applications, 33(21), 14975–14989. doi:10.1007/s00521-021-06133-0.

[37] Bailly, A., Blanc, C., Francis, É., Guillotin, T., Jamal, F., Wakim, B., & Roy, P. (2022). Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. Computer Methods and Programs in Biomedicine, 213(106504). doi:10.1016/j.cmpb.2021.106504.

[38] Bongiorno, V., Gibbon, S., Michailidou, E., & Curioni, M. (2022). Exploring the use of machine learning for interpreting electrochemical impedance spectroscopy data: evaluation of the training dataset size. Corrosion Science, 198, 110119. doi:10.1016/j.corsci.2022.110119.

[39] Decelle, R., Ngo, P., Debled-Rennesson, I., Mothe, F., & Longuetaud, F. (2021). Pith Estimation on Tree Log End Images. Reproducible Research in Pattern Recognition. RRPR 2021. Lecture Notes in Computer Science, 12636. Springer, Cham, Switzerland. doi:10.1007/978-3-030-76423-4_7.

[40] Larumbe-Bergera, A., Garde, G., Porta, S., Cabeza, R., & Villanueva, A. (2021). Accurate pupil center detection in off-the-shelf eye tracking systems using convolutional neural networks. Sensors, 21(20), 6847. doi:10.3390/s21206847.

[41] Larumbe, A., Cabeza, R., & Villanueva, A. (2018). Supervised descent method (SDM) applied to accurate pupil detection in off-the-shelf eye tracking systems. Proceedings of the 2018 ACM Symposium on Eye Tracking Research Applications, 1-8. doi:10.1145/3204493.3204551.

[42] Larumbe-Bergera, A., Porta, S., Cabeza, R., & Villanueva, A. (2019). SeTA. Proceedings of the 11th ACM Symposium on Eye Tracking Research Applications. doi: 10.1145/3314111.3319830.

[43] King, D. E. (2009). Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research, 10, 1755-1758.

[44] Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 18-22 June, 2023, Vancouver, Canada.

[45] Kim, S., Jeong, M., & Ko, B. C. (2020). Energy efficient pupil tracking based on rule distillation of cascade regression forest. Sensors (Switzerland), 20(18), 1–17. doi:10.3390/s20185141.

[46] Lee, K.I., Jeon, J.H., & Song, B.C. (2020). Deep Learning-Based Pupil Center Detection for Fast and Accurate Eye Tracking System. Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science, 12364, Springer, Cham, Switzerland. doi:10.1007/978-3-030-58529-7_3.

[47] Cai, H., Liu, B., Ju, Z., Thill, S., Belpaeme, T., Vanderborght, B., & Liu, H. (2019). Accurate eye center localization via hierarchical adaptive convolution. British Machine Vision Conference 2018, BMVC 2018, 3-6 September, 2018, London, United Kingdom.

[48] Levinshtein, A., Phung, E., & Aarabi, P. (2018). Hybrid eye center localization using cascaded regression and hand-crafted model fitting. Image and Vision Computing, 71, 17–24. doi:10.1016/j.imavis.2018.01.003.

[49] Choi, J. H., Il Lee, K., Kim, Y. C., & Cheol Song, B. (2019). Accurate Eye Pupil Localization Using Heterogeneous CNN Models. 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan. doi:10.1109/icip.2019.8803121.

[50] Kitazumi, K., & Nakazawa, A. (2018). Robust Pupil Segmentation and Center Detection from Visible Light Images Using Convolutional Neural Network. 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan. doi:10.1109/smc.2018.00154.