Jurnal Teknik Informatika C.I.T Medicom, 17 (5) (2025) 225-241
Published by: Institute of Computer Science (IOCS)
Jurnal Teknik Informatika C.I.T Medicom
Journal homepage: www.medikom.iocspublisher.org

Real-Time Pill Counting on Low-Power Device: A YOLOv5 Pipeline with Confidence Thresholding and NMS

Galih Prakoso Rizky A[1], Rifka Widyastuti[2]
1),2) Fakultas Ilmu Komputer, Universitas Pembangunan Nasional Veteran Jakarta, Indonesia

Article Info

Article history

Received : Sep 10, 2025
Revised : Nov 06, 2025
Accepted : Nov 26, 2025

Keywords:

Computer vision;
Deep learning;
Object detection;
Pill counter;
Yolov5.

Abstract

Manual pill counting is still commonly performed in healthcare facilities and pharmacies, but this method is vulnerable to human error and requires significant processing time. This study develops an automatic pill counting pipeline using the YOLOv5 deep learning model, optimized for low-power devices such as Raspberry Pi, Orange Pi, and Jetson Nano. Unlike earlier techniques that depend on conventional retrieval or machine-learning approaches, this pipeline integrates real-time object detection with customized confidence thresholding and Non-Maximum Suppression (NMS), enabling high accuracy and fast performance on edge hardware with limited resources. The development process includes collecting and annotating a dataset of pill images with variations in shape, color, and orientation, followed by training YOLOv5 using optimized parameters. A simple webcam is used as the input device, and system performance is evaluated under different lighting and background conditions. Experimental results show that the model achieves 98% precision, 88% recall, 95% mAP@0.5, and 67% mAP@0.5:0.95, with an average inference speed of around 15 milliseconds per image. Tests on ten pill-counting scenarios under optimal lighting demonstrate strong performance, with only minor discrepancies in dense cases involving 50 and 127 pills, producing accuracies of 98% and 99.21%. These results indicate that the optimized YOLOv5 pipeline provides fast and accurate real-time pill counting on low-power devices. Future work will enhance robustness to lighting variations, validate using external datasets, and incorporate color and shape feature analysis to improve performance in challenging scenarios.

Corresponding Author:

Galih Prakoso Rizky A,
Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jalan Rumah Sakit Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, Indonesia, 12450
galihprizky@upnvj.ac.id

1. Introduction

Manual pill counting is still a common method used in various healthcare facilities and pharmacies. However, dispensing errors in pharmacy continue to present a significant global challenge. According to systematic reviews and institutional reports, medication dispensing errors contribute to patient safety issues, with studies indicating error rates ranging from 0.5% to 6.5% in various settings [1][2]. These errors not only result in inaccurate dosages but also cause service delays and can have serious consequences for patient health [3][4][5]. Therefore, an automated system capable of rapidly, accurately, and consistently counting pills, while integrating with inventory management, is urgently required.

The development of computer vision and deep learning technologies has provided promising solutions in the field of object recognition, including drug detection and classification [6][7]. Various studies have shown that Convolutional Neural Network (CNN) has the ability to accurately recognize the shape, color, and size of pills, even under varying lighting and background conditions [8] [9]. One of the popular and effective object detection algorithms is You Only Look Once version 5 (YOLOv5), which is known for its balance between accuracy and inference speed, making it suitable for real-time applications [10] [11]. Previous research has proposed pill identification systems using deep learning models to verify the type and amount of drugs, which developed CNN-based pill detection systems with high accuracy [12][9]. However, most studies still focus on pill type identification, not fully integrating real-time pill count detection.

However, most prior work emphasizes pill type identification rather than real-time counting and often assumes ideal imaging conditions or high-performance hardware. Some recent works have adapted YOLO for automated pill counting, reporting promising accuracy and efficiency [13][14]. Nevertheless, detection performance can still be impaired by challenging conditions such as poor illumination, background clutter, pill occlusion, or class imbalance within datasets [13][15].

Despite this progress, two substantial gaps remain. First, most YOLO-based pill counting approaches offer limited robustness under low-light scenarios or in the presence of overlapping pills, which frequently occur in real-world pharmacy environments [13][15]. Second, there is limited research reporting comprehensive benchmarking of pill counting solutions specifically on resource-constrained edge devices, such as Raspberry Pi, Orange Pi, or Jetson Nano, with metrics for both speed and accuracy. Furthermore, existing works rarely integrate additional features like adaptive thresholding, model pruning, or color and shape feature classification into their pipelines.

To address these gaps, this study develops and evaluates an automatic pill counting system based on an optimized YOLOv5 architecture, designed for deployment on low-power edge devices. Our main contributions are**:** (a) a robust real-time pill counting pipeline with confidence-thresholding and efficient NMS, (b) comprehensive benchmarking under various lighting, background, and occlusion scenarios with edge hardware, and (c) the groundwork for integrating advanced feature classification (color and shape) in future research.

2. Methods

This research aims to develop an automatic pill counter system using deep learning-based YOLOv5 algorithm. YOLOv5 (You Only Look Once version 5) is a popular and widely used deep learning-based object detection algorithm due to its ability to detect objects in real-time with high accuracy. YOLOv5 is a development of the previous YOLO series that combines efficiency and speed of inference with a modular architecture design.

The YOLOv5 architecture consists of three main parts:

(i) **Backbone**: This part is responsible for extracting features from the input image using convolutional layers. YOLOv5 uses CSPDarknet architecture as the backbone which optimizes the efficiency of feature extraction by reducing duplication of information and enhancing the representation capability.
(ii) **Neck**: This section is responsible for combining features from multiple resolution levels using Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) techniques, so that the model can detect objects of various sizes, including small objects such as pills.
(iii) **Head**: This section predicts object classes and bounding boxes at specific locations in the image. YOLOv5 uses anchor-based detection and a loss function that optimizes detection precision.

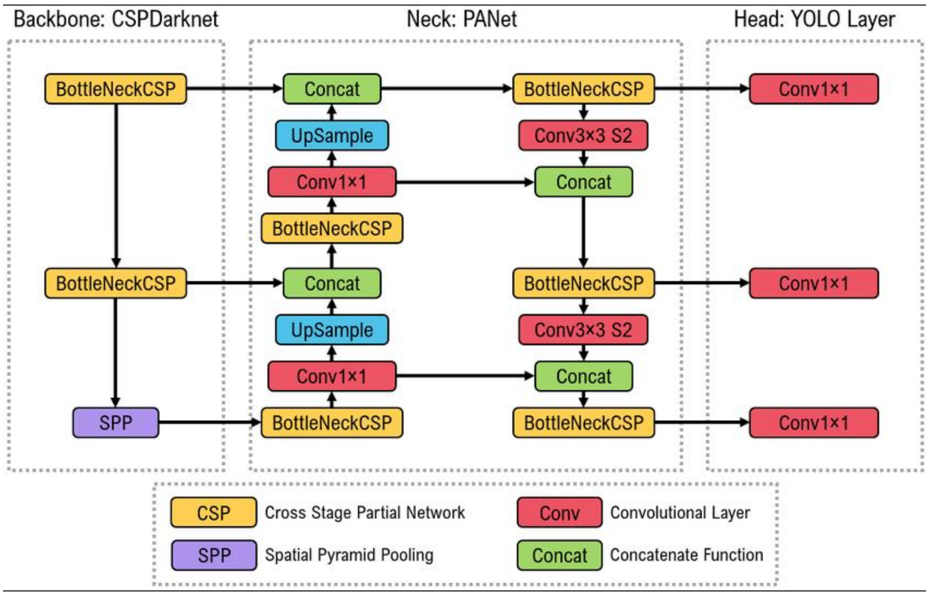


Figure 1. Yolov5 Architecture.

YOLOv5 is known for its high inference speed, making it suitable for real-time applications on devices with limited resources, such as automated pill counter systems in pharmacies. The research method consists of several main stages as follows:

Data Collection and Preparation

In the data preparation process, all images were manually annotated using the Roboflow platform, which is used specifically to facilitate the labeling process. With Roboflow, each pill in the image is labeled with a bounding box and a class according to the type of pill in the image. This annotation process is done carefully so that the data used for training is of high quality and can strengthen the model's performance in recognizing pill objects.

After the labeling process is complete, the dataset is divided into two parts, 1,000 images for training and 120 images for validation. The division process is done randomly so that the distribution of pill variations and environmental conditions is maintained in both parts of the data. With this method, the resulting model is expected to be able to detect and count the number of pills effectively, both on training data and test data that have never been identified before.

This research also refers to standard practices in dataset management for object detection in the pharmaceutical field, as described in the following journals. One of them is the research by W. Sun et al. (2022) who discussed the importance of data variation (type, color, size, and luminance) to improve the generalization of pill detection models [13]. In addition, the use of platform-based labeling tools such as Roboflow has become a common method in the process of manually annotating datasets for the training needs of deep learning models. By following such methods, this research ensures optimal data quality and representation for the development of automated pill counter systems.

Preprocessing Data

The data preprocessing stage is an important part of the training process for deep learning-based object detection models. In this research, all collected images are first processed by resizing to 640x640 pixels to fit the input requirements of the YOLOv5 architecture. Image size uniformity aims to maintain input consistency and speed up the model training process. Various data augmentation techniques such as rotation, flipping, and brightness and contrast adjustment were then applied to the images to increase the diversity of the training data and strengthen the generalization ability of the model to diverse real conditions. Specifically, for the rotation augmentation, each image was rotated by 90 degrees three times, generating three additional variants per original image, in order to cover a wide range of pill orientations commonly encountered in practical scenarios. For flipping, horizontal flip was applied to the initial image to produce a mirrored version, thereby enhancing the dataset's representational completeness regarding spatial orientation. In addition, brightness augmentation was complemented by the utilization of a 700-lumen LED lamp during image acquisition, ensuring consistent and stable illumination across all samples. This controlled lighting setup was implemented to minimize brightness variability and reduce the potential negative impact of uncontrolled environmental factors on model training.

Data augmentation has been shown to be effective in reducing the risk of overfitting and improving the performance of object detection models, especially in tasks involving high visual variation such as pharmaceutical pill recognition. [16]. The study by W. Sun et al. (2022) also emphasized the importance of combining augmentation techniques such as rotation, light intensity change, and flipping to produce models that are more robust in detecting objects under various lighting and background conditions [13].

Below is an example of an image used as a training dataset, which has undergone the preprocessing and augmentation processes described above.

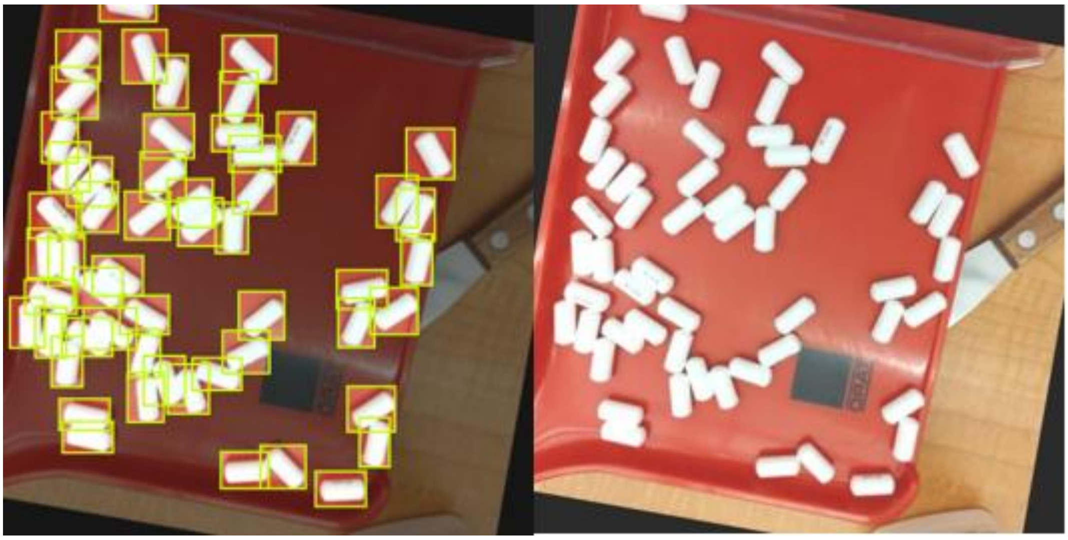


Figure 2: Training data for the model.

YOLOv5 Model Training

The model training stage is the core of the development of an automatic pill detection and counting system. In this research, the "small" version of the YOLOv5 model (yolov5s) is used, which is known for its lightweight structure and high computational efficiency, making it suitable for applications on devices with limited resources. The YOLOv5s model was trained using a pre-processed and annotated dataset, with the following configurations, the image size was set to 640x640 pixels according to the YOLOv5 input standard, the batch size used was 16, and the training was conducted for 150 epochs to ensure the model obtained optimal learning from the available data. Furthermore, an ablation study was performed by systematically varying the batch size (8, 16, and 32) and the number of epochs (50, 70, and 150), as summarized in Table 1. The results demonstrate that the configuration of batch size 16 with 150 epochs consistently achieved the best balance between detection accuracy (mAP@0.5: 0.88), generalizability (mAP@0.5:0.95: 0.68), and robustness in both precision (0.96) and recall (0.88). These improvements can also be visually observed in the training and validation metric graphs presented in Figure 3, where the chosen configuration exhibited stable convergence and superior performance compared to other settings. This systematic evaluation confirms that a batch size of 16 and 150 training epochs is optimal within the context of this dataset and task, supporting both quantitative and qualitative improvements shown by the ablation results.

Table 1.
Comparison of batch and epoch changes in training models

Batch Size	Epoch	mAP@0.5	mAP@0.5:0.95	Precision	Recall
8	50	~0.83	~0.61	~0.97	~0.88
8	70	~0.65	~0.48	~0.72	~0.62
8	150	~0.64	~0.52	~0.82	~0.69
16	150	~0.88	~0.68	~0.96	~0.88
32	150	~0.88	~0.67	~0.97	~0.84

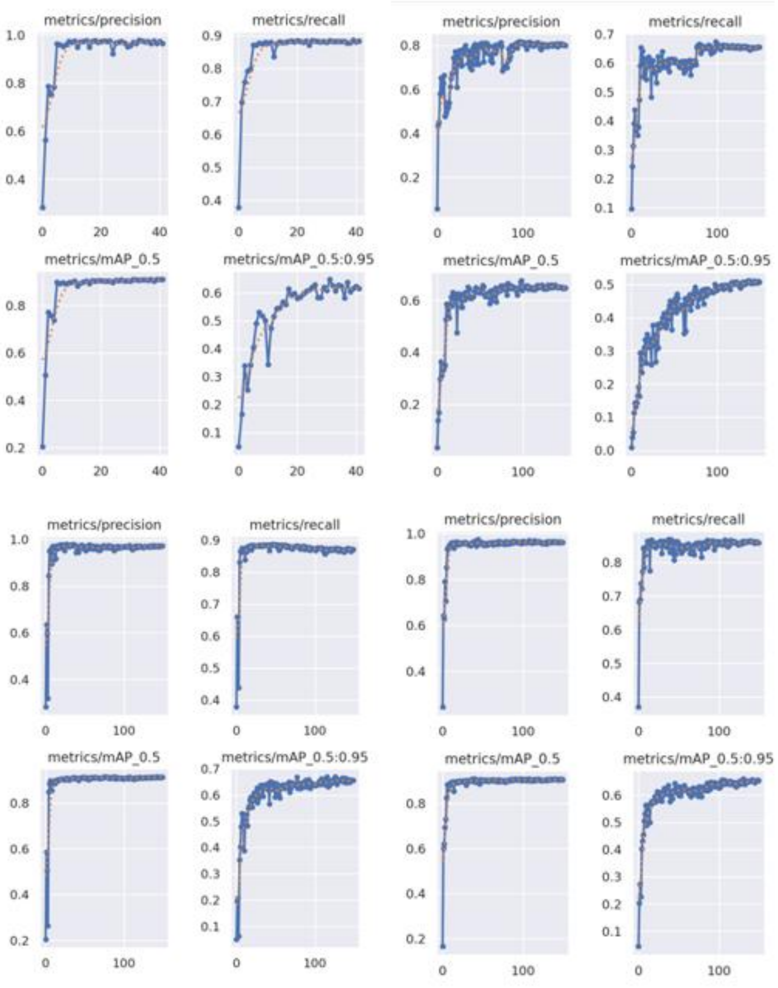


Figure 3. Model training graph changes.

In addition, the model training utilized pretrained weights from yolov5s.pt to speed up the convergence process and improve detection performance at an early stage. The dataset is configured through the data.yaml configuration file, which contains information on the location of the training and validation data, as well as the object classes to be detected. During the training process, the dataset is also cached to speed up the iteration process and maximize the utilization of computational resources.

The main goal of the training process is to minimize the loss value of the detection function, so that the resulting model is able to recognize and count pills with a high degree of accuracy. The choice of model training configuration, including the use of YOLOv5s and transfer learning techniques, has proven effective in various object detection studies, especially in cases with limited data and deployment requirements on edge devices [17]. For example, research by Q. Huang et al. (2024) showed that the use of YOLOv5s with pretrained weights and training configuration optimization can produce fast and accurate models for image-based object detection tasks, including pharmaceutical and healthcare applications [17].

Evaluasi Model

The model performance evaluation was conducted thoroughly on the validation dataset using standard object detection metrics, namely mean Average Precision (mAP), precision, recall, and F1-score. The mAP metric is used to measure the average precision of object detection at various Intersection over Union (IoU) threshold values. Precision and recall are used to assess the model's ability to produce correct predictions and avoid detection errors, both in the form of false positives and false negatives.

F1-score is the harmonic mean of precision and recall, used to assess the balance between the two metrics, especially in data that has an unbalanced class proportion. The F1-score formula is expressed in the following formula:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{1}$$

By using a combination of these metrics, the evaluation aims to obtain a comprehensive assessment of the model's performance, both in terms of detection accuracy and its ability to reduce object misidentification. The use of these metrics has proven effective in various deep learning-based object detection studies, especially in pharmaceutical and healthcare applications. For example, a study by J. Zhou et al. showed that mAP, precision, recall, and F1-score are highly relevant indicators to evaluate the YOLOv5 model in the task of detection and classification of small objects such as pills [18].

Pill Counter System Implementation

In the implementation stage, the trained YOLOv5 model is integrated into a Python-based application that is capable of performing real-time pill detection and counting using input from a webcam. The system utilizes the OpenCV library for direct image capture from the camera, while the inference process is done by loading the YOLOv5 model through the PyTorch framework.
This app design:

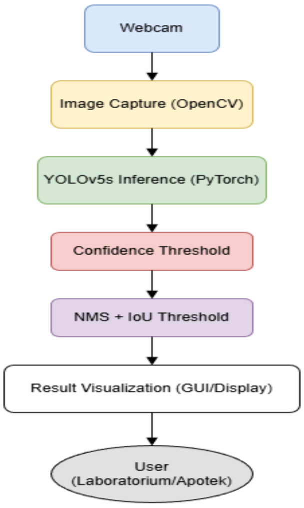


Figure 4. pill counter architecture.

This system design aims to prove that the developed model is not only algorithmically accurate, but also practically applicable in real environments, such as laboratories or pharmacies. Detailed aspects of implementation such as library usage, webcam configuration, and application performance optimization will be discussed further in the discussion chapter.

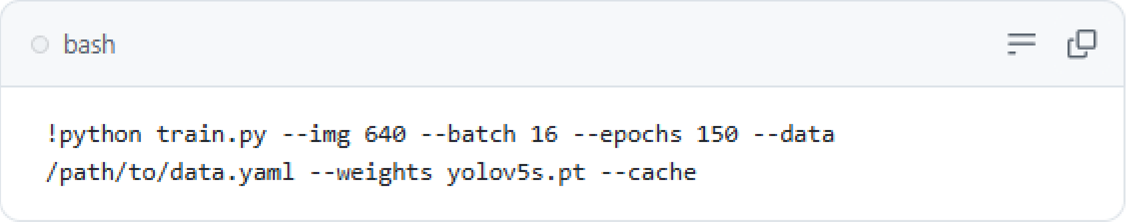
In research related to object detection, the Intersection over Union (IoU) threshold value is often used to assess how accurate the system is in distinguishing between the predicted and actual position of the object in the image. Recent studies have shown that choosing the right IoU threshold determines the quality of detection results. The higher the threshold, the more selective the detection system will be in identifying pills that actually match the original position, thus increasing precision and reducing detection errors [19][20]. In addition, the development of IoU-based evaluation methods and functions continues so that the process of calculating objects, such as pills, can run more efficiently and accurately in real implementation [21]. Therefore, implementing an optimal IoU threshold in a pill counter system is crucial to ensure the number of pills detected is close to the actual number and to minimize double detection and miss detection. The basic concept of the IoU threshold is as follows:

$$IoU = \frac{Area\ (Prediction\ \cap GroundTruth)}{Area\ (Prediction\ \cup GroundTruth)} \tag{2}$$

This approach of integrating object detection models into real-time applications has been widely used in recent research in the pharmaceutical and healthcare fields, where the combination of deep learning frameworks, computer vision, and camera devices has been shown to improve the efficiency and accuracy of the monitoring process [22].

3. Results and Discussion

In this section, various results obtained from the research stages will be described, starting from the evaluation of model performance to the implementation of real-time pill counter applications. The presentation of research results is carried out to assess the level of success of the methods used in answering the problem formulation and achieving research objectives.



```bash
!python train.py --img 640 --batch 16 --epochs 150 --data
/path/to/data.yaml --weights yolov5s.pt --cache
```

Figure 3. YOLOv5 model training command.

In this study, the YOLOv5 model training process was carried out by running the command as shown in Figure 3, using the train.py script with the parameters --img 640 --batch 16 -epochs 150 --data /path/to/data.yaml --weights yolov5s.pt --cache. The command is used to set the image size, number of batches, number of epochs, data location, initial model weights, and speed up the training process with cache.

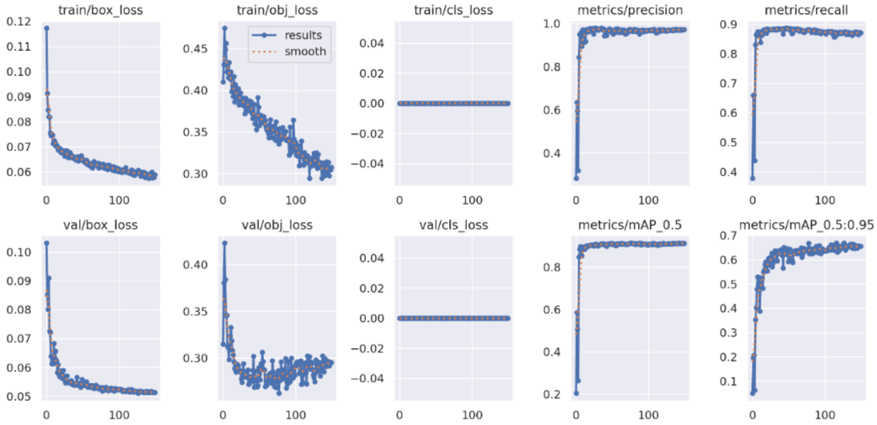


Figure 5. Model training results with YOLOv5.

Based on the YOLOv5 model training results obtained, as shown in Figure 5, it can be observed that the learning process is optimized. The train/box_loss and train/obj_loss graphs show a consistent decreasing trend over the 150 training epochs, indicating that the model successfully minimizes the error in predicting the location and presence of pill objects in the image. The box_loss value in the training data decreased from about 0.12 at the beginning of the epoch to about 0.06 at the end of training, while obj_loss decreased from about 0.45 to about 0.32. A similar decrease in loss values is also seen in the validation data, which indicates that the model undergoes a good generalization process and does not show symptoms of overfitting. Research by Wang et al. (2024) reviewed that during the YOLOv5 training process, the box_loss and obj_loss graphs usually show a consistent downward trend as the number of epochs increases. This decrease indicates that the model has succeeded in minimizing the prediction error of the location and presence of objects in the image, thus improving the detection accuracy [23].

The classification loss (cls_loss) in the training and validation data shows a very small value and tends to be constant at zero throughout the training process. It can be interpreted that the class classification process on the dataset runs well, possibly due to the limited number of classes so that the classification complexity is low.

Furthermore, the model performance evaluation results based on the precision, recall, and mean average precision (mAP) metrics show quite high achievements. The precision and recall graphs show significant spikes at the beginning of the training epoch, then stabilize at high values, which are above 0.9 for precision and above 0.85 for recall. This indicates that the model has a very good level of accuracy and completeness in pill object detection on the validation data. The mAP value at threshold 0.5 reaches more than 0.65 and the mAP at threshold 0.5:0.95 is also at a fairly optimal value, indicating the ability of the model to detect objects of various sizes and positions.

Overall, the stable graph trend at the end of the training process and the absence of loss increase in the validation data indicate that the YOLOv5 model has successfully performed effective learning without overfitting. These results indicate that the developed model has a good level of accuracy and generalization, making it feasible to be implemented in a real-time pill detection system. This achievement also indicates that the selection of training parameters, the use of preprocessed data, and the utilization of pretrained weights contribute positively to the final performance of the model.

This finding is in line with the results of a study conducted by Wang et al. (2023), where the use of YOLOv5 for an automatic pill detection and identification system showed improved accuracy and efficiency in deep learning-based pharmaceutical object recognition [24]. In addition, a recent study by B. Dang et al. (2024) also proved that YOLOv5 is capable of being implemented in mobile applications for real-time pill identification, with excellent performance on real data and special needs user environments [25]. Thus, the model training results in this study are not only consistent with the latest literature, but also reinforce YOLOv5's position as one of the leading object detection methods in the field of pharmacy.

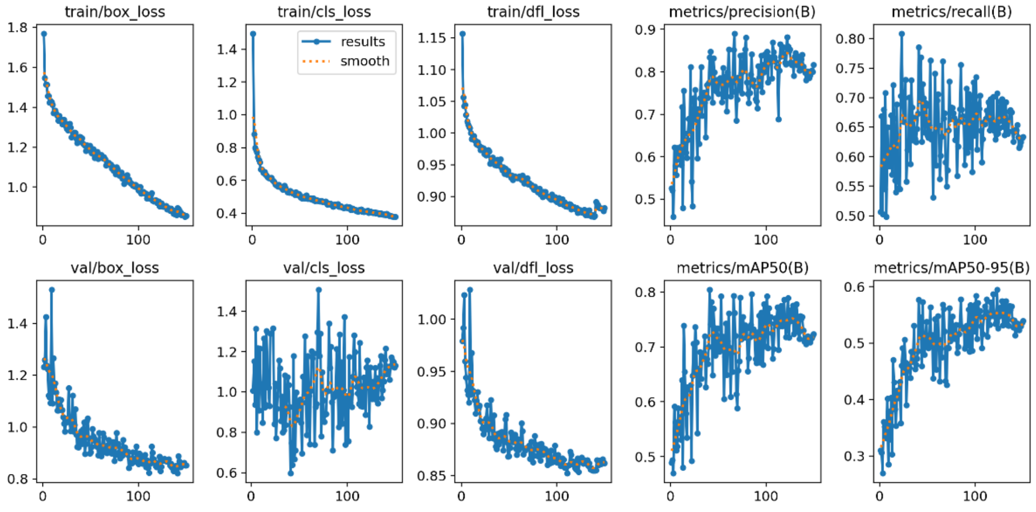


Figure 6. Model training results with YOLOv8.

In addition to evaluating the YOLOv5 model, this study also conducted a separate training experiment using YOLOv8 with the same dataset and parameters. The training results in Figure 6 show that YOLOv5 has more stable optimization performance than YOLOv8 when applied to the same dataset. The downward trend of train loss and validation loss in YOLOv5 is consistent and results in lower final values, especially in the box_loss and obj_loss metrics. This resulted in higher precision and recall, with precision stabilizing above 0.9 and recall above 0.85. This learning stability indicates that the anchor-based YOLOv5 architecture is still very adaptive to the characteristics of pill datasets that tend to be homogeneous and small in size.

In contrast, YOLOv8 using the anchor-free approach shows a more fluctuating training pattern, especially in validation metrics such as precision, recall, and mAP. Although the loss value of YOLOv8 tends to be low and the decline is progressive, the performance of the detection metrics at the end of training is slightly below that of YOLOv5. YOLOv8 achieves a precision of around 0.80-0.85, recall of around 0.60-0.70, and mAP50-95 of around 0.55. This performance variation indicates that YOLOv8 optimization requires further hyperparameter adjustments to accommodate pill datasets that are sensitive to small shape variations. Thus, overall YOLOv5 showed superior performance under the same test conditions.

Table 2.
Comparison of YOLOv5 vs YOLOv8 models.

Aspect	YOLOv5	YOLOv8
Architecture Type	Anchor-based	Anchor-free
Loss Stability	Very stable, consistently decreasing	Stable but with notable fluctuations
train/box_loss	~0.06 (low and stable)	~0.38–0.42
train/cls_loss	Near zero	~0.40–0.50
train/dfl_loss	Not applicable	~0.88 (decreasing steadily)
val/box_loss	~0.85 (stable)	~0.90–1.00 (more variable)
Precision	> 0.90	~0.80–0.85
Recall	> 0.85g	~0.60–0.70
mAP50	> 0.65	~0.75 (but fluctuating)

Aspect	YOLOv5	YOLOv8
mAP50–95	~0.50–0.55	~0.55 (but unstable)
Curve Smoothness	Very smooth, minimal noise	Significant fluctuations in validation curves
Dataset Suitability	Highly compatible for small-object datasets	Requires additional tuning
Overall Performance	More stable and higher-performing	Promising but less consistent

System Implementation

In the implementation stage, a real-time pill detection and counting system was developed based on the pre-trained YOLOv5 model. For the image acquisition process, the system still uses a webcam camera device with 1080p 2MP resolution as the source of image input. This camera was chosen because it is easily accessible and sufficient enough to capture the details of the pill object to be detected.

System Information

Current Date/Time: 22 August 2025, 13:28:03
Computer Name: DESKTOP-AD4UKJU
Operating System: Windows 10 Pro 64-bit (10.0, Build 19045)
Language: English (Regional Setting: English)
System Manufacturer: LENOVO
System Model: 81B1
BIOS: 3ECN37WW(V3.04)
Processor: AMD Ryzen 3 2200U with Radeon Vega Mobile Gfx (4 CPUs), ~2.5GHz
Memory: 8192MB RAM
Page file: 12233MB used, 12243MB available
DirectX Version: DirectX 12

Figure 7. Laptop device specifications.

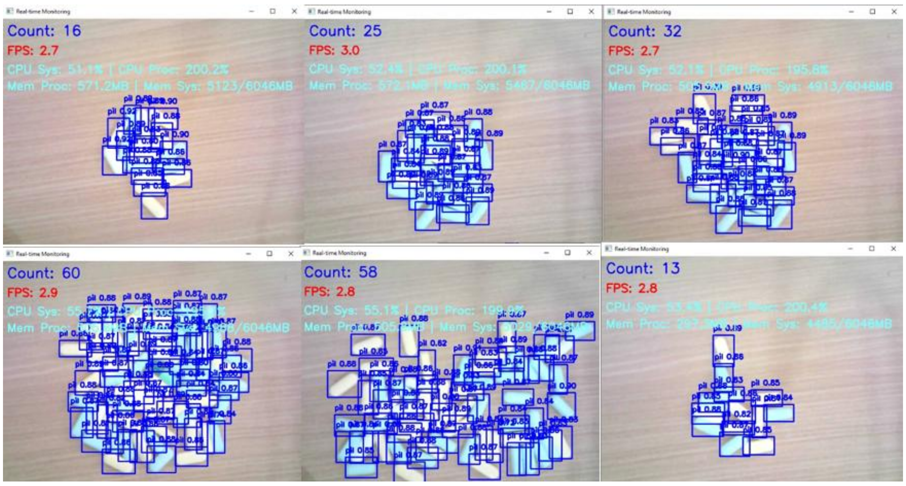
All pill detection and counting processes were performed on a personal laptop device with specifications as shown in Figure 7, without using any special devices or external servers. The system is built using the Python programming language by utilizing various supporting libraries such as OpenCV for image acquisition and processing, as well as the YOLOv5 library for the model inference process.

The script developed in this research is capable of acquiring images directly from the webcam, then running the pill object detection process in real-time. Each frame taken from the camera will be processed through the inference stage using the YOLOv5 model, so that the pill object can be detected continuously and interactively. The system automatically assigns bounding boxes and labels to each detected pill object, and counts the number of pills in each frame, which is then displayed interactively on the video interface so that users can monitor the detection and counting results directly.

To complement the tests conducted on a personal laptop device, this research also includes an evaluation of the model's performance on an edge device, namely a Raspberry Pi 4, to test the model's inference capabilities in a limited computing environment. This test was conducted by running YOLOv5 and YOLOv8 using the same dataset and processing configuration, so that the performance comparison can be observed objectively. The test results show that YOLOv5 has more optimal performance on Raspberry Pi 4 devices, both in terms of speed, processing load, and real-time inference stability. The complete comparison is shown in the following table and Figure 8-9:

Table 3.
Performance comparison of YOLOv5 vs YOLOv8.

Aspect	YOLOv5	YOLOv8	Results
FPS (Real-Time)	2.7–3.0 FPS	1.9–2.0 FPS	YOLOv5
CPU Processor Load	195–203%	233–293%	YOLOv5
CPU System Load	52–57%	77–89%	YOLOv5
Memory Usage	297–572 MB (dynamic)	468–480 MB (stable but heavier)	Draw
Stable for multiple objects	Very stable	Less stable	YOLOv5
Edge device inference capabilities	More optimal	Heavier and slower	YOLOv5



```
Count: 16
FPS: 2.7
CPU Sys: 51.1% | CPU Proc: 200.2%
Mem Proc: 571.2MB | Mem Sys: 5123/6046MB
```

```
Count: 25
FPS: 3.0
CPU Sys: 52.4% | CPU Proc: 200.1%
Mem Proc: 572.1MB | Mem Sys: 5467/6046MB
```

```
Count: 32
FPS: 2.7
CPU Sys: 52.1% | CPU Proc: 195.8%
```

```
Count: 60
FPS: 2.9
```

```
Count: 58
FPS: 2.8
CPU Sys: 55.1% | CPU Proc: 199.9%
```

```
Count: 13
FPS: 2.8
CPU Sys: 53.4% | CPU Proc: 200.4%
```

Figure 8. YOLOv5 performs on edge devices.

```
FPS: 1.9
Count: 13
CPU Sys: 85.5% | CPU Proc: 281.2%
Mem Proc: 468.2MB | Mem Sys: 4860/6046MB
```

```
FPS: 2.0
Count: 35
CPU Sys: 89.2% | CPU Proc: 233.9%
Mem Proc: 468.5MB | Mem Sys: 4884/6046MB
```

```
FPS: 1.9
Count: 60
CPU Sys: 77.3% | CPU Proc: 284.6%
Mem Proc: 470.7MB | Mem Sys: 4841/6046MB
```

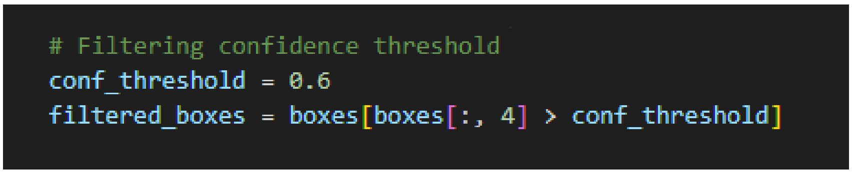
```
FPS: 2.0
Count: 30
CPU Sys: 79.1% | CPU Proc: 293.8%
Mem Proc: 471.2MB | Mem Sys: 4838/6046MB
```

```
FPS: 1.9
Count: 43
CPU Sys: 85.4% | CPU Proc: 282.8%
Mem Proc: 471.7MB | Mem Sys: 5023/6046MB
```

```
FPS: 2.0
Count: 14
CPU Sys: 81.3% | CPU Proc: 287.7%
Mem Proc: 471.7MB | Mem Sys: 4704/6046MB
```

Figure 9. YOLOv8 performs on edge devices.

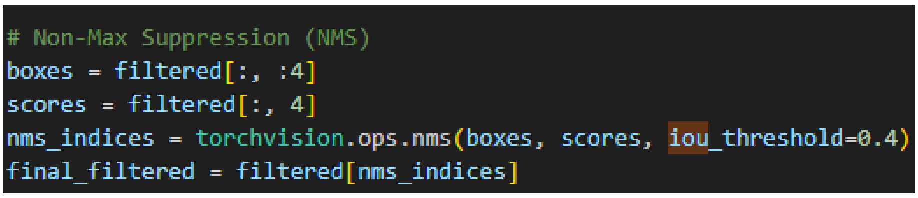
This finding confirms that, under limited computing conditions such as on a Raspberry Pi 4, YOLOv5 provides more efficient and responsive performance, especially in real-time object detection scenarios with a large number of objects. This suggests that model selection should consider the compatibility of the architecture with the computing platform, especially when applied to edge devices.



```
# Filtering confidence threshold
conf_threshold = 0.6
filtered_boxes = boxes[boxes[:, 4] > conf_threshold]
```

Figure 10. Confidence threshold assignment.

This implementation involves several main steps, such as receiving video or image input from the webcam, performing inference on each frame, using a confidence threshold as shown in Figure 10, to ensure that only detections with a high confidence level are displayed, and applying Non-Max Suppression (NMS) to avoid duplicate detections of the same object by giving the iou threshold parameter of 0.4 so that if $\text{IoU} \geq 0.4$, then the box with a lower score will be removed. According to Zhang et al. and Gilg et al. Non-Maximum Suppression (NMS) is applied with a certain IoU threshold to eliminate duplicate detection, where boxes with lower scores will be removed if the IoU value against other higher boxes exceeds the set threshold [26],[27], The information of the number of pills and the detected bounding box are displayed in real-time on the video interface, so that users can know the number of pills easily and conveniently.



```
# Non-Max Suppression (NMS)
boxes = filtered[:, :4]
scores = filtered[:, 4]
nms_indices = torchvision.ops.nms(boxes, scores, iou_threshold=0.4)
final_filtered = filtered[nms_indices]
```

Figure 11. NMS script to avoid duplication.

The implementation results show that the YOLOv5-based pill detection and counting system can run efficiently and responsively despite using only simple hardware such as a personal laptop and a standard webcam. This proves that the developed solution has the potential to be widely adopted, especially in laboratory environments or small-scale pharmaceutical industries that do not yet have specialized devices for pill detection and counting automation. A similar approach has also been applied in other studies, where the use of YOLOv5 and Python with webcam input proved to be capable of real-time object detection and counting with a high level of accuracy and ease of implementation on standard computing devices [28][29].

Experiment Results

This experiment aims to test the effectiveness and accuracy of a real-time pill count detection and counting system using the YOLO model and image input from a webcam. The developed system is expected to be able to detect pill objects in each video frame, display bounding boxes and labels, and calculate the number of pills automatically and interactively, so that it can be used in laboratory or pharmaceutical industry applications to facilitate inspection and quality control processes.

An experimental scenario was conducted by randomly placing a number of pills in front of a webcam camera and running the detection system in real-time. Each process of image capture, detection, and pill count was observed and evaluated, both in terms of response speed and result accuracy. In the test, the parameters and configuration of the system were adjusted.

Main system parameters or configuration:

(i) Confidence threshold: 0.6 (only detections with confidence level > 60% are taken)
(ii) Non-Max Suppression (NMS): default IoU Threshold parameter value of 0.4 is given
(iii) Input frame size: default webcam (1920x1080 pixels)
(iv) The YOLOv5 model was trained with a dataset of 1000 pill images and 120 validation images.
(v) Tests were conducted on a wide variety of pill counts and specific lighting conditions.

Then the system testing process is carried out with the following steps. First, the YOLOv5-based object detection model is loaded from a local file using the PyTorch library. Data acquisition is performed in real-time via video input from a webcam, where each captured frame will be directly processed by the detection model. On each frame, an inference process is performed to detect the object pill.

Next, the detection results are filtered based on the confidence threshold value, where only bounding boxes with a confidence level of more than 0.6 are processed further. After that, the Non-Max Suppression (NMS) method is applied with an IoU threshold of 0.4 to eliminate multiple detections of the same object. Bounding boxes resulting from filtering and NMS are then drawn on the frame along with the label and confidence value of each object.

The number of pills detected in each frame is calculated based on the number of bounding boxes that pass the filtering and NMS processes. The pill count information is displayed in real-time on the monitoring result window, allowing the user to monitor the number of pills detected at any time. The experiment is interactive, and can be stopped by pressing the ‘q’ key on the keyboard. The results of the experiment with good lighting can be seen in table 4.

Tabel 4.
Test results with good lighting.

No	Actual Number	Detected	Accuracy	Description
1	4	4	100%	All pills detected
2	12	12	100%	All pills detected
3	24	24	100%	All pills detected
4	30	30	100%	All pills detected
5	40	40	100%	All pills detected
6	50	49	98%	There are undetectable pills due to pill density
7	57	57	100%	All pills detected
8	84	84	100%	All pills detected
9	100	100	100%	All pills detected
10	127	126	99.21%	There is pill overlap

Based on a series of tests conducted on ten pill-count scenarios under optimal lighting conditions, the detection system showed excellent performance. In the majority of trials, the number of detected objects matched the actual number, indicating that the model was able to operate reliably under clear visual conditions. However, two trials showed discrepancies between the actual count and the detection results, namely in the tests with 50 pills and 127 pills, resulting in 98% and 99.21% accuracy, respectively. This inaccuracy was mainly due to the level of object density and occlusion, which hindered the system from distinguishing the inter-pill boundary consistently.

These findings show that while good illumination is a major contributing factor in improving detection quality, spatial distribution and object density remain critical aspects that can affect model performance. Thus, achieving stable detection accuracy requires an arrangement of objects that minimizes the possibility of occlusion. An example of detection results with good illumination is shown in Figure 12.

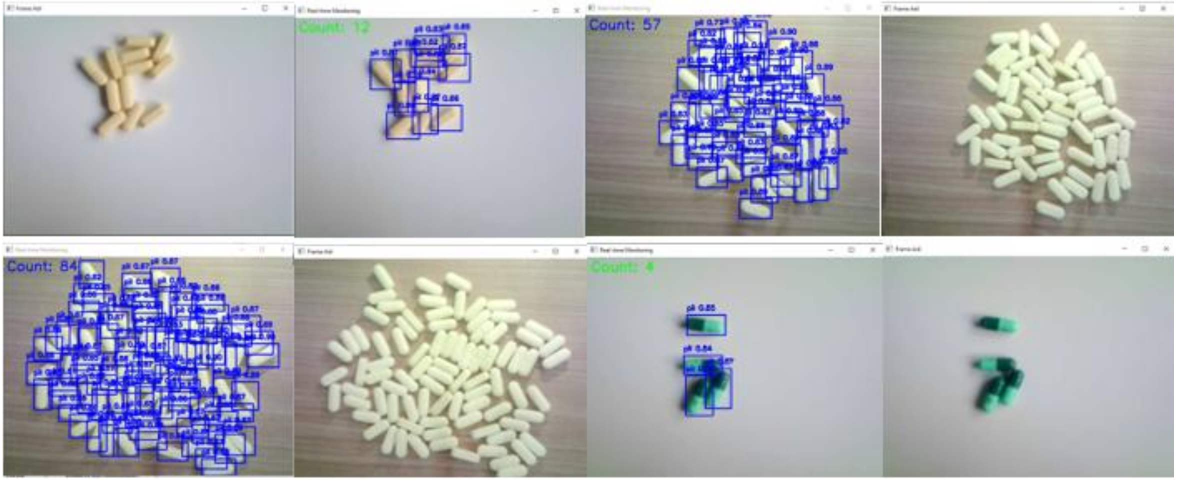


Figure 12. Detection results with good lighting.

Furthermore, several tests were carried out using the same configuration and the same threshold parameters, but with less lighting conditions and the results are shown in table 5.

Tabel 5.
Test results with low light condition.

No	Actual Number	Detected	Accuracy	Description
1	24	24	100%	All pills detected
2	30	30	100%	All pills detected
3	47	46	97.8%	There are pills that are not detected due to pill density.
4	56	57	98.15%	There are pills that are not detected due to pill density
5	78	78	100%	All pills detected

In low light conditions, the system was still able to detect all pills with 100% accuracy at a relatively small number of pills. However, as the number of pills increased and the density between them increased, some pills were not detected, resulting in a decrease in accuracy, such as in the experiments with 47 pills (97.8%) and 54 pills (98.15%). However, with a larger number of pills and a relatively high density, the results showed a 100% accuracy rate in the experiment with 78 pills (100%), as shown in Figure 13. This shows that low light and object density can affect system performance, especially in detecting pills that are close together.

Figure 13. Detection results with poor lighting.

In addition to the influence of lighting and pill density, the test results are also affected by the limited resolution of the camera used, which is only 2MP. A low-resolution camera can cause the resulting image details to be less sharp, so that pill objects that are close together or overlapping become more difficult to detect accurately by the system. Even though the system has worked optimally in poor lighting conditions. Thus, camera quality is one of the important factors affecting the detection performance of the pill counter system. According to research by Liu et al. (2023), detection of small objects in low-resolution images tends to be less accurate because the details of the resulting image are not sharp, so objects that are close together or overlapping become more difficult for the system to recognize accurately [30].

From the two experiments that have been carried out, pill detection with good lighting conditions and with poor lighting conditions, this pill counter system can be implemented in health facilities or pharmacies, but it is necessary to pay attention to the lighting in the detection area and improve the quality of the camera to get better results.

4. Conclusion

The results of this study demonstrate that the YOLOv5-based pill counting system can detect and count pills in real time with a high level of accuracy, particularly under optimal lighting conditions and moderate object density. The model training process indicates effective learning, marked by decreasing loss values and strong precision–recall performance, reinforcing the importance of visual clarity and object separability in small-object detection tasks. Across ten experimental scenarios, the system successfully matched the actual pill counts in most cases, with minor errors occurring only when pills were densely clustered or overlapping, where occlusion limited boundary distinguishability. This finding reflects the inherent limitations of anchor-based detectors when spatial separation between objects decreases. Several constraints affect the interpretation of these results. The dataset remains relatively small and lacks sufficient variation in pill characteristics, potentially limiting generalizability. Detection performance also shows sensitivity to lighting conditions and object distribution, indicating possible degradation in more complex environments. Furthermore, hardware limitations primarily the use of a laptop and Raspberry Pi 4 restricted evaluation on more capable edge-AI platforms. Despite these limitations, the study provides empirical insight into the behavior of lightweight object detection models on constrained hardware and offers implications for other small-object counting tasks such as quality inspection or grain counting. The evaluation contributes to the limited body of research assessing YOLOv5 performance for real-time counting on edge devices and identifies key environmental and hardware factors influencing accuracy. The developed system shows strong potential for further refinement. Future research should expand the dataset, incorporate domain adaptation methods, utilize higher-resolution imaging, and evaluate performance across a wider range of edge-AI platforms to improve robustness and generalizability.

References

[1] I. S. Um, A. Clough, and E. C. K. Tan, "Dispensing error rates in pharmacy: A systematic review and meta-analysis," *Res. Soc. Adm. Pharm.*, vol. 20, no. 1, pp. 1–9, 2024, doi: https://doi.org/10.1016/j.sapharm.2023.10.003.

[2] J. Bonkowski *et al.*, "Effect of barcode-assisted medication administration on emergency department medication errors," *Acad. Emerg. Med.*, vol. 20, no. 8, pp. 801–806, 2013, doi: https://doi.org/10.1111/acem.12189.

[3] Y. Zheng *et al.*, "Designing human-centered AI to prevent medication dispensing errors: focus group study with pharmacists," *JMIR Form. Res.*, vol. 7, no. 1, p. e51921, 2023, doi: https://doi.org/10.2196/51921.

[4] B. Orkaby, E. Kerner, M. Saban, and C. Levin, "Bridging generational gaps in medication safety: insights from nurses, students, and generative AI models," *BMC Nurs.*, vol. 24, no. 1, p. 382, 2025, doi: https://doi.org/10.1186/s12912-025-03034-8.

[5] O. Tchijevitch *et al.*, "Methodological Approaches for Analyzing Medication Error Reports in Patient Safety Reporting Systems: A Scoping Review," *Jt. Comm. J. Qual. Patient Saf.*, vol. 51, no. 1, pp. 46–73, 2025, doi: https://doi.org/10.1016/j.jcjq.2024.10.005.

[6] H.-J. Kwon, H.-G. Kim, and S.-H. Lee, "Pill detection model for medicine inspection based on deep learning," *Chemosensors*, vol. 10, no. 1, p. 4, 2021, doi: https://doi.org/10.3390/chemosensors10010004.

[7] D. Upadhyay, M. Manwal, V. Kukreja, and R. Sharma, "A Fine-Tuned YOLOv5 and Exception Model for Oral Cancer Detection," in *2024 5th International Conference for Emerging Technology (INCET)*, IEEE, 2024, pp. 1–5. doi: https://doi.org/10.1109/INCET61516.2024.10592942.

[8] K. Al-Hussaeni, I. Karamitsos, E. Adewumi, and R. M. Amawi, "CNN-based pill image recognition for retrieval systems," *Appl. Sci.*, vol. 13, no. 8, p. 5050, 2023, doi: https://doi.org/10.3390/app13085050.

[9] A. D. Nguyen, H. H. Pham, H. T. Trung, Q. V. H. Nguyen, T. N. Truong, and P. Le Nguyen, "High accurate and explainable multi-pill detection framework with graph neural network-assisted multimodal data fusion," *PLoS One*, vol. 18, no. 9, p. e0291865, 2023, doi: https://doi.org/10.1371/journal.pone.0291865.

[10] ultralytics, "Ultralytics YOLO Docs," ultralytics.

[11] S. Feng, H. Qian, H. Wang, and W. Wang, "Real-time object detection method based on YOLOv5 and efficient mobile network," *J. Real-Time Image Process.*, vol. 21, no. 2, p. 56, 2024, doi: https://doi.org/10.1007/s11554-024-01433-9.

[12] S. Akshaya, A. C. Uthaman, and S. Sridhar, "Detection and Identification of Pills using Machine Learning Models," in *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, IEEE, 2023, pp. 1–6. doi: https://doi.org/10.1109/ViTECoN58111.2023.10157873.

[13] W. Sun, X. Niu, Z. Wu, and Z. Guo, "Lightweight Detection Counting Method for Pill Boxes Based on Improved YOLOv8n.," *Electron.*, vol. 13, no. 24, 2024, doi: https://doi.org/10.3390/electronics13244953.

[14] S. J. Kim and D. S. Cho, "Medical-Pills Detection Using YOLOv11: A Proof of Concept Study for Pharmaceutical Automation," vol. 2, no. 4, p. 1, 2025, doi: https://doi.org/10.21203/rs.3.rs-6337589/v2.

[15] L. Tan, T. Huangfu, L. Wu, and W. Chen, "Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification," *BMC Med. Inform. Decis. Mak.*, vol. 21, no. 1, p. 324, 2021, doi: https://doi.org/10.1186/s12911-021-01691-8.

[16] S.-H. Lee, D.-M. Son, and S.-H. Lee, "Enhanced Multi-Pill Detection and Recognition Using VFI Augmentation and Auto-Labeling for Limited Single-Pill Data," in *IEEE Access*, IEEE, 2025, pp. 60859–60878. doi: https://doi.org/10.1109/ACCESS.2025.3557569.

[17] Q. Huang, Y. Zhou, T. Yang, K. Yang, L. Cao, and Y. Xia, "A lightweight transfer learning model with pruned and distilled YOLOv5s to identify arc magnet surface defects," *Appl. Sci.*, vol. 13, no. 4, p. 2078, 2023, doi: https://doi.org/10.3390/app13042078.

[18] J. Zhou, T. Su, K. Li, and J. Dai, "Small target-YOLOv5: enhancing the algorithm for small object detection in drone aerial imagery based on YOLOv5," *Sensors*, vol. 24, no. 1, p. 134, 2023, doi: https://doi.org/10.3390/s24010134.

[19] C. Liu, K. Wang, Q. Li, F. Zhao, K. Zhao, and H. Ma, “Powerful-IoU: More straightforward and faster bounding box regression loss with a nonmonotonic focusing mechanism,” *Neural Networks*, vol. 170, no. 2, pp. 276–284, 2024, doi: https://doi.org/10.1016/j.neunet.2023.11.041.

[20] L. Yang, K. Zhang, J. Liu, and C. Bi, “Location IoU: A New Evaluation and Loss for Bounding Box Regression in Object Detection,” in *2024 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2024, pp. 1–8. doi: https://doi.org/10.1109/IJCNN60899.2024.10649985.

[21] H. Xu *et al.*, “Rethinking boundary discontinuity problem for oriented object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17406–17415. doi: https://doi.org/10.48550/arXiv.2305.10061.

[22] Y. J. Hwang, G. H. Kim, M. J. Kim, and K. W. Nam, “Deep learning-based monitoring technique for real-time intravenous medication bag status,” *Biomed. Eng. Lett.*, vol. 13, no. 4, pp. 705–714, 2023, doi: https://doi.org/10.1007/s13534-023-00292-w.

[23] R. Khanam and M. Hussain, “What is YOLOv5: A deep look into the internal features of the popular object detector,” *arXiv Prepr. arXiv2407.20892*, vol. 128, no. 7, p. 4012, 2024, doi: https://doi.org/10.48550/arXiv.2407.20892.

[24] J. Heo, Y. Kang, S. Lee, D.-H. Jeong, and K.-M. Kim, “An accurate deep learning–based system for automatic pill identification: Model development and validation,” *J. Med. Internet Res.*, vol. 25, no. 7, p. e41043, 2023, doi: https://doi.org/10.2196/41043.

[25] B. Dang, W. Zhao, Y. Li, D. Ma, Q. Yu, and E. Y. Zhu, “Real-time pill identification for the visually impaired using deep learning,” in *2024 6th International Conference on Communications, Information System and Computer Engineering (CISCE)*, IEEE, 2024, pp. 552–555. doi: https://doi.org/10.1109/CISCE62493.2024.10653353.

[26] J. Gilg, T. Teepe, F. Herzog, P. Wolters, and G. Rigoll, “Do we still need non-maximum suppression? Accurate confidence estimates and implicit duplication modeling with IoU-aware calibration,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 4850–4859. doi: https://doi.org/10.48550/arXiv.2309.03110.

[27] K.-S. Si *et al.*, “Accelerating Non-Maximum Suppression: A Graph Theory Perspective,” *Adv. Neural Inf. Process. Syst.*, vol. 37, no. 11, pp. 121992–122028, 2024, doi: https://doi.org/10.48550/arXiv.2409.20520.

[28] S. K. Jaiswal and R. Agrawal, “A comprehensive review of YOLOv5: advances in real-time object detection,” *Int. J. Innov. Res. Comput. Sci. Technol*, vol. 12, no. 3, pp. 75–80, 2024, doi: https://doi.org/10.55524/ijircst.2024.12.3.12.

[29] X. Song and W. Gu, “Multi-objective real-time vehicle detection method based on yolov5,” in *2021 International Symposium on Artificial Intelligence and its Application on Media (ISAIAM)*, IEEE, 2021, pp. 142–145. doi: https://doi.org/10.1109/ISAIAM53259.2021.00037.

[30] R. Jing, W. Zhang, Y. Liu, W. Li, Y. Li, and C. Liu, “An effective method for small object detection in low-resolution images,” *Eng. Appl. Artif. Intell.*, vol. 127, no. 1, p. 107206, 2024, doi: https://doi.org/10.1016/j.engappai.2023.107206.