A MACHINE LEARNING ALGORITHMS FOR DRUG IDENTIFICATION: ENHANCING MEDICATION SAFTEY IN THAILAND

Sorayus Kittisorayut

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Information Technology
Graduate Studies
Thai-Nichi Institute of Technology
Academic Year 2024

Thesis Topic	A Machine Learning Algorithms for Drug identification: Enhancing Medication Safety in Thailand
By	Sorayus Kittisorayut
Field of Study	Information Technology
Thesis Advisor	Acting Sub Lt. Dr. Pichitchai Kamin

The Graduate Studies of Thai-Nichi Institute of Technology has been approved and accepted as partial fulfillment of the requirement for the Master's Degree

…….Vice President for Academic Affairs

(Assoc. Prof. Dr. Warakorn Srichavengsup)

Month…… Date…… Year……..……

Thesis Committees

……..…….Chairman

(Assoc. Prof. Dr. Annop Munsakul)

……..…….Committee

(Dr. Pramuk Boonsieng)

……..…….Committee

(Dr. Sarayut Nonsiri)

……..…….Advisor

(Acting Sub Lt. Dr. Pichitchai Kamin)

SORAYUS KITTISORAYUT: A MACHINE LEARNING ALGORITHMS FOR DRUG IDENTIFICATION: ENHANCING MEDICATION SAFTEY IN THAILAND. ADVISOR: ACTING SUB LT. DR. PICHITCHAI KAMIN, 48 PP.

Medication errors are among the leading causes of patient harm within the healthcare system. To address this critical issue, a variety of tools and solutions have emerged to minimize the occurrence of these errors, particularly those stemming from Look-Alike, Sound-Alike (LASA) drugs. Notable among these solutions are automated dispensing systems that utilize barcode scanning and RFID technology to enhance accuracy in medication administration.

This study investigates the challenge of drug identification through a machine learning approach. Initially, a dataset comprising commonly used medications in Thailand was manually curated. Subsequently, various versions of the YOLO (You Only Look Once) object detection models were trained and evaluated to assess their efficacy in drug identification. The primary goal of this research is to develop and compare the performance of these different YOLO model variants.

Evaluation metrics for this study include precision, recall, F1-score, and mean average precision (mAP). The results demonstrate that the YOLOv8-nano model achieves the highest accuracy in drug identification, recording a mAP score of 99.5%, a recall of 99.7%, and an F1 score of 99.4%, outperforming other versions.

In conclusion, this research highlights the significant potential of machine learning as a critical component in efforts to mitigate medication errors in the future. By improving drug identification processes, these advanced technologies can enhance patient safety and contribute to more reliable healthcare practices.

Graduate Studies Student's Signature……..……...……

Field of Study Information Technology Advisor's signature……...…..……

Academic Year 2024

Acknowledgements

I would like to extend my heartfelt gratitude to Acting Sub Lt. Dr. Pichitchai Kamin, my advisor, whose exceptional guidance and unwavering support were crucial to the completion of this dissertation. His expertise and insights not only enriched my research but also inspired me to pursue my academic goals with confidence and rigor. Without his assistance, this work would not have been possible, and I am profoundly grateful for his mentorship throughout this journey.

I also wish to express my sincere appreciation to my research committee members—Dr. Annop Munsakul, Dr. Pramook Boonsieng, and Dr. Sarayut Nonsiri. Their valuable feedback and constructive criticism played a pivotal role in shaping this dissertation into its final form. Each of them brought unique perspectives that challenged me to think critically and creatively, which greatly enhanced the quality of my work. I am truly thankful for their encouragement and support during this process.

A special thank you goes to my family, whose unwavering love and support have been a constant source of strength for me. To my parents and siblings, your belief in my abilities has motivated me to push through the challenges I faced during my research. I am incredibly grateful for the sacrifices you have made and for always being there to encourage me.

Furthermore, I would like to acknowledge my friends, who provided both emotional support and companionship during this challenging journey. Your encouragement and camaraderie made the long hours of research and writing much more enjoyable.

Lastly, I would like to express my gratitude to the Thai-Nichi Institute of Technology, which played an instrumental role in my academic achievements. The opportunities provided by the institute allowed me to expand my knowledge and explore new horizons beyond my original field of study. It has been an incredible journey, and I am deeply thankful for everything I have gained along the way.

Sorayus Kittisorayut

Table of Contents

Table of Contents (Continued)

List of Tables

List of Figures

List of Figures (Continued)

Chapter 1
Introduction

1.1 Statement of the problem

Medication errors are globally a leading cause of patient harm. Statistically, it was found that there are approximately 237 million errors over a year which leads to an estimated annual cost of 4-21 billion euros in Europe [1]. Medication errors are defined as events that can lead to inappropriate medication use or patient harm while in the control of healthcare professionals or patients [2]. These errors could occur in any stage, from prescription, transcription, preparation, dispensing, administration, and monitoring. The most common medication errors include incorrect dosage and wrong administration speed, which are mostly attributed to human factors due to poor work conditions such as heavy workload or staff shortages.

Look-alike, Sound-alike (LASA) drugs are also leading to error at the level of pharmacists and physicians, which are responsible for 6.2-14.7% of all medication errors [3]. Look-alike medicines appear visually the same with respect to packaging, shape, color, or size, while sound-alike medicines are similar in phonetics of their names, doses, or strengths. LASA errors can occur at any stage of medication use. For example, during prescription, errors might happen from poorly legible handwritten prescriptions, verbal orders, use of error-prone abbreviations and selection in computerized prescriber order entry forms. Similarly, during administration, errors might occur due to unclear instructions for administration, failure to double-check the order and failure to monitor the patient after administration. The impact of LASA errors is tremendous, especially when involving high-alert medicine that can cause severe patient harm or death.

The evidence on medication errors shows that 50-70.2% could be prevented through comprehensive systematic approaches to patient safety [4], including changes in policy, drug names, similar packaging, or the use of “Tall man” lettering in confusing names.

Nowadays, to reduce the occurrence rate of medication errors that might be caused by Look-alike, Sound-alike (LASA) drugs, there are many tools and solutions

available. For example, Automated Dispensing Cabinet (ADC) technologies represent a solution for automatically dispensing drugs, utilizing barcodes for drug identification or devices that employ radio-frequency identification (RFID) to locate the positions of drugs. However, while these technologies might reduce the occurrence rate of medication errors, they also have their own limitations. In the case of Barcode, it might be more secure and lower expense than their counterpart, but they can't scan multiple items at the same time. In contrast, RFID systems can be more efficient for scanning large numbers of items, but they can be more expensive and more prone to malicious attack or electromagnetic interference.

This study focuses on the problem of drug identification in look-alike drugs. The first step, a dataset is manually collected of commonly used look-alike drugs in Thailand. Our datasets would consist of 3 generic drugs, which include acetaminophen, ibuprofen, naproxen. And each class also collected data from 3 different manufacturers in Thailand. Which concludes 9 classes in total. Then the object detection models are trained and tested based on the different version of pretrained YOLO model to achieve drug identification. The objective of this paper is to develop and compare the performance of YOLO models capable of accurately identifying look-alike drugs and reducing the occurrence rate of medication events associated with such drugs.

1.2 Objective of the Study

1.2.1 To research and develop a model for drug identification.

1.2.2 To compare the accuracy and performance of each version of YOLO.

1.3 Scope of Work

1.3.1 Datasets consist of 3 generic drugs and 3 different manufacturers from each drug, resulting in 9 classes as follows.

1. Ibuprofen
    i. Coprofen
    ii. Adavil
    iii. Cefen

2. Naproxen
    i. Annoxen
    ii. Soproxen
    iii. Synflex
3. Acetaminophen
    i. Cemol Central
    ii. Sara
    iii. Tylenol

1.3.2 The object detection algorithms: YOLOv8, YOLOv9 and YOLO-NAS have been compared for analysis.

1.4 Expected Benefits

1.4.1 Decrease the number of events in medication errors from LASA problems.

1.4.2 Create datasets of general drugs for further studies.

Chapter 2
Literature reviews

Since this study's goal is to create an object detection model for safety drug use in Thailand. The literature review provided the related technology, tools, and research, is as follows.

2.1 Related Theories

2.1.1 Artificial Intelligence

2.1.2 Deep Learning Architecture

2.1.3 Transfer Learning

2.1.4 You Only Look Once (YOLO) algorithms

2.1.5 Confusion metrics

2.2 Related Research

2.1 Related Theories

2.1.1 Artificial Intelligence

Artificial Intelligence (AI) is a broader term that describes the capabilities of machines to learn and think like humans. AI encompasses a broad range of techniques and approaches, including machine learning, natural language processing, computer vision and more.

Machine learning is a subset of AI that provide the ability to learn with different algorithms and improve from experience without being programmed [5]. Machine learning can be categorized by training datasets into 3 categories.

1) Supervised learning: by using labeled datasets as an input to train model. This category is commonly used for classification and regression. Algorithm that falls in this category such as Naive Bayes, Support Vector Machine, K-nearest neighbor and Logistics regression
2) Unsupervised learning: as the name suggests, by not providing any features to the program allowing machine to search for patterns independently. K-means clustering, DBSCAN, Mean-shift, Singular Value Decomposition fall in this category.

3) Reinforcement learning: using positive or negative reinforcement signals to guide the appropriate action for the machine which useful for games or robots. Reinforcement learning includes Q-Learning, Genetic algorithm, DQN.

Deep learning is a subset of machine learning that inspired by the nature of human brain [6]. By using an artificial neural network to imitate the neuron of human brain which make it has advantages over machine learning in term of pre-processing process. When compare with machine learning algorithms such as Naive Bayes, SVM etc. are require pre-processing and feature extraction phase but in deep learning models are capable of classifying data into different classes and categories themselves without the need of feature extraction steps.

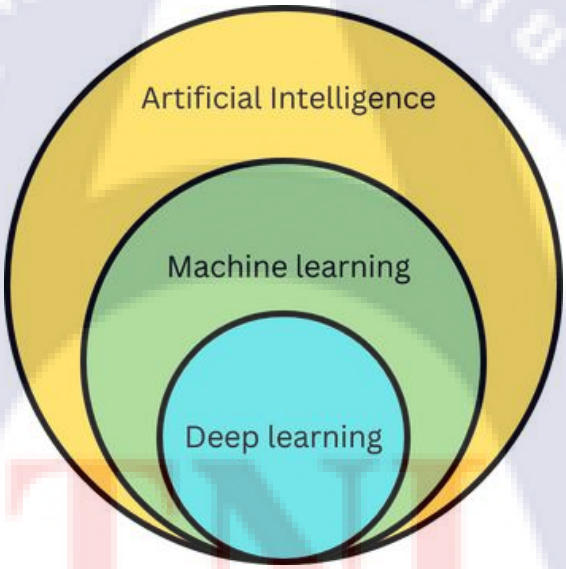

Figure 2.1 Overview of AI, ML and DL

2.1.2 Deep Learning Architecture

Deep learning techniques are also categorized into three major categories: Discriminative (Supervised), Generative (Unsupervised) and others.

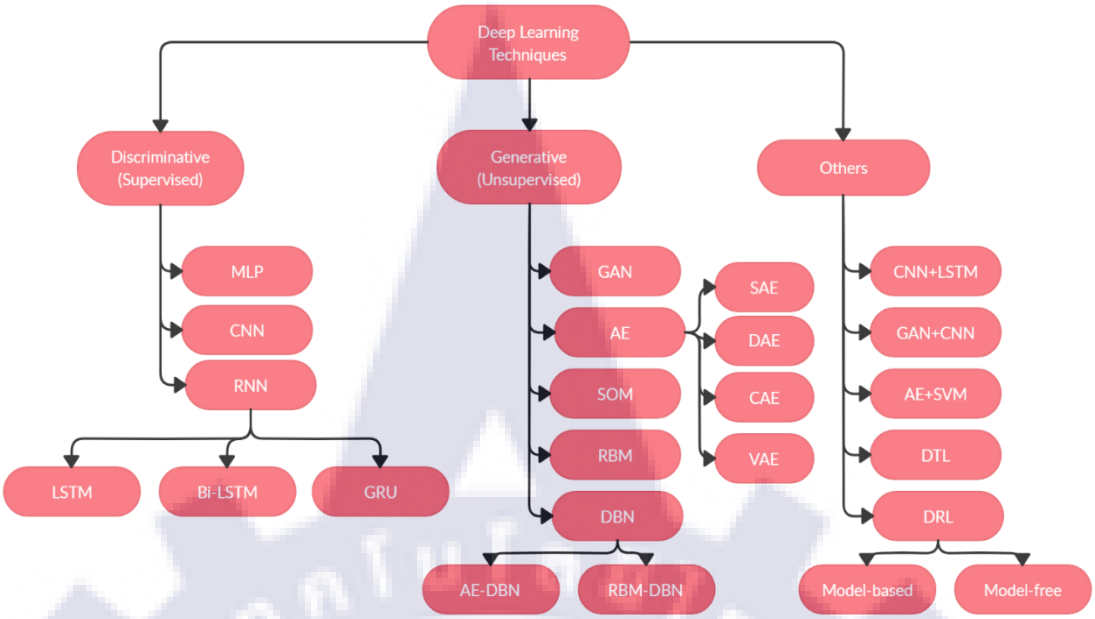

Figure 2.2 Categories of Deep learning techniques [7]

One of popular discriminative deep learning architecture is the convolutional neural network (CNN), which usually consists of three layers type: convolutional layers, pooling layers, fully connected layers.

Convolutional layers act like feature detectors, scanning the input data with filters to identify specific patterns and extract low-level features like edges, corners, and textures.

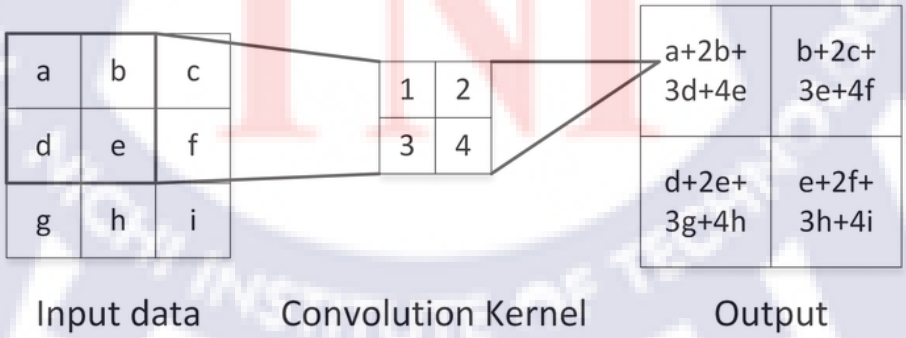

Figure 2.3 Diagram of convolutional operation [8]

Pooling layers by feature mapping, reducing the spatial size of the data while preserving key features. Different pooling techniques, like max pooling and average pooling, was developed to help achieve this summarization.

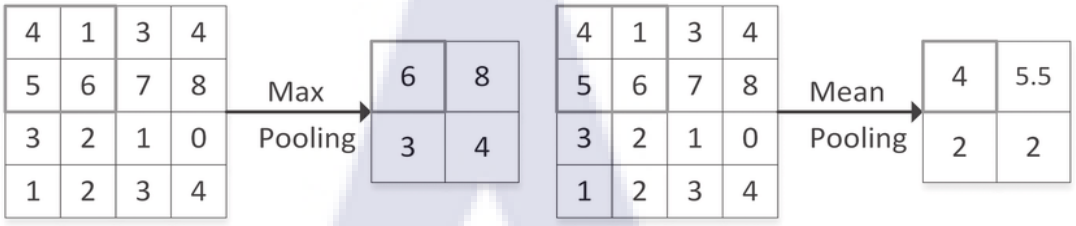

Figure 2.4 Diagram of max pooling and mean pooling [8]

Fully connected layers, connect each neuron forwarding to every neuron in the next level. Which allows the model to analyze the extracted features from previous layers and learn complex relationships within the data to perform tasks like image classification, speech recognition, or other desired predictions.

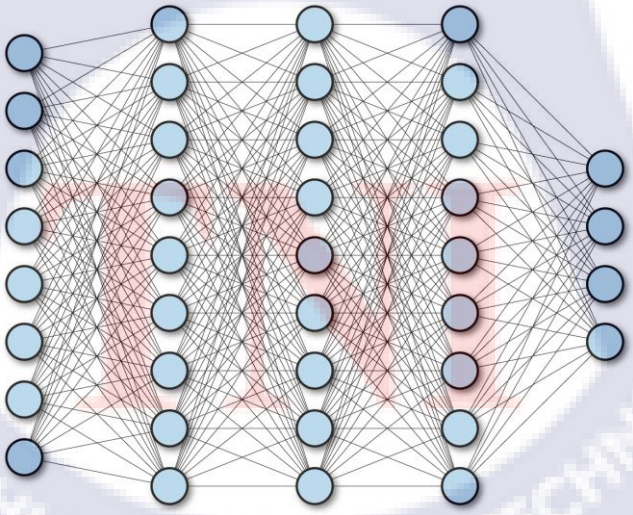

Figure 2.5 A multilayer fully connected network [9]

Through these layers of convolution and pooling, the extracted features from signals such as image and sounds are treated as inputs and then forwarded to fully

connected layers to classify the relationship between pattern recognition and understanding data.

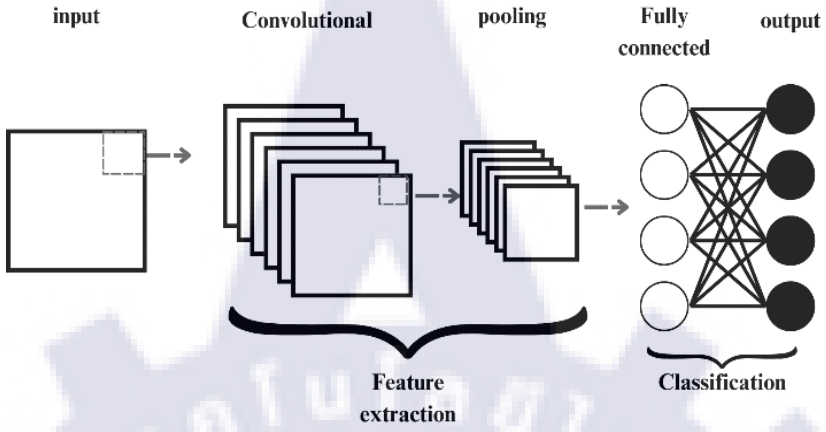

Figure 2.6 Deep Learning Architecture

2.1.3 Transfer Learning

Transfer learning is a machine learning method that we use the existing machine learning model that was built for one specific task to be a starting point for a model on a new task which make the new model can progress rapidly when compared to building from scratch and achieve significantly higher performance than training with only small amount of data, especially when data availability for the new task is limited. The essence of transfer learning typically means using an existing model which has been trained for other purposes. then fine-tuning by retraining only the few last layers specific to the new task.

Traditionally, transfer learning is categorized into three groups depend on task domain and amount of labeled or unlabeled data presents:

1. Inductive transfer learning: in inductive transfer learning, the source and target domain are required to be the same. And by using inductive biases to train model for different specific tasks.
2. Transductive transfer learning: Transductive learning is helpful when the domain of source and target tasks are different. By using the labeled data

from source domain to predict the unlabeled target domain from similarity between those tasks.

3. Unsupervised transfer learning: This method was used when both source and target domain are unlabeled to acquire features from the source that can be transferred to use in target domain.

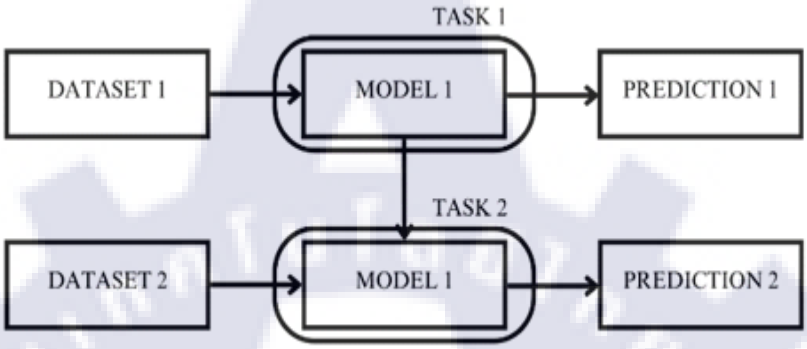

Figure 2.7 Concept of Transfer Learning

2.1.4 You Only Look Once (YOLO) algorithm

By using deep learning methods, image-based solutions have been developed. Using preprocessing images of the drug to obtain the correct viewing angle and characteristics of the medicine are established. Then implement drug identification in a framework based on deep convolutional networks (DCN) to achieve image recognition model. One of the most popular object detection frameworks is YOLO (You Only Look Once)

YOLO is a real time object detection algorithm developed by Joseph Redmon and Ali Farhadi in 2015 and has been developed into several version and multiple variants, such as v1-v8 with an enhanced performance on every updated versions The principal behind YOLO is to divide the input image into a grid of cells and each cell is predicting a set of bounding box and then class label into object with highest probabilities The advantage of YOLO is due to its processes entire image in one time which make process more faster and efficient than R-CNN [10]

YOLOv2: by adding the anchor boxes, allows this version to handle objects of different shapes and sizes better. And due to multi-scale approach, allows this version to extract features at different scales from same image [11]

YOLOv3: this is the third version of You Only Look Once (YOLO) object detection algorithm developed by J. Redmon and A. Farhadi [12]. with the introduction of a technique called "feature pyramid network", make this version able to extract features from images at different scales. Along with three different sizes of detection kernels: 13x13, 26x26, and 52x52. This significantly improved detection accuracy for objects of different sizes.

YOLOv4: in this version, with the addition of several new features include Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross Mini-Batch Normalization (CmBN), Self-adversarial-training (SAT), Mish-activation, Mosaic data augmentation, DropBlock regularization, CIoU loss. And "bag of freebies" techniques, the model's performance has been improved in speed and accuracy [13].

YOLOv5: by changing to EfficientDet architecture and use of anchor-free detection make this model achieve higher accuracy and flexibility [14]. This version is also capable of using transfer learning, which allows it to be pre-trained on a large dataset and fine-tuned on a smaller dataset.

YOLOv6: this model introduces several new enhancements including the implementation of a Bi-directional Concatenation (BiC) module, an anchor-aided training (AAT) strategy, and an improved backbone and neck design, make it able to run faster with fewer computational resources without compromising efficiency [15].
YOLOv7: this version introduces several key features that focus on the optimization of the training process and improve the accuracy of object detection without increasing the inference cost [16]. including a new label assignment method, extended and compound scaling methods etc.

YOLOv8: With an adoption of advanced backbone and neck architectures and anchor-free head, resulting in improved feature extraction and object detection performance [17]. YOLOv8 also offers a wide range of models for different specific tasks such as YOLOv8 for object detection or YOLOv8-seg for segmentation, making this version suitable for a variety of applications.

YOLO-NAS: Stands for “You Only Look Once-Neural Architecture Search”, this version was developed by Deci.ai and made a major leap in object detection [18]. By utilizing Deci’s AutoNAC NAS technology, this model can employ optimized algorithms to discover the most suitable architecture for a given task. YOLO-NAS also employs quantization-aware blocks and selective quantization for optimal performance make this model display an outstanding capability in object detection.

YOLOv9: The latest iterations in YOLO families, with the introduction of techniques such as Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN), ensuring that the essential information required is retained and improve overall performance compared with previous versions [19]

This study employs various versions of YOLOv8, YOLOv9 and YOLO-NAS to train models and subsequently compares the results, showcasing the advancements in object detection achieved through the evolution of YOLO models.

Table 2.1 Comparison between each version [19]

Version	**Date**	**Anchor**	**Framework**	**Backbone**	**AP (%)**
YOLO	2015	No	Darknet	Darknet24	63.4
YOLOv2	2016	Yes	Darknet	Darknet24	78.6
YOLOv3	2018	Yes	Darknet	Darknet53	33.0
YOLOv4	2020	Yes	Darknet	CSPDarknet53	43.5
YOLOv5	2020	Yes	Pytorch	YOLOv5CSPDarknet	55.8
PP-YOLO	2020	Yes	PaddlePaddle	ResNet50-vd	45.9
Scaled-YOLOv4	2021	Yes	Pytorch	CSPDarknet	56.0
PP-YOLOv2	2021	Yes	PaddlePaddle	ResNet101-vd	50.3
YOLOR	2021	Yes	Pytorch	CSPDarknet	55.4
YOLOX	2021	No	Pytorch	YOLOXCSPDarknet	51.2
PP-YOLOE	2022	No	PaddlePaddle	CSPRepResNet	54.7

Table 2.1 Comparison between each version (Cont.) [19]

Version	Date	Anchor	Framework	Backbone	AP (%)
YOLOv6	2022	No	Pytorch	EfficientRep	52.5
YOLOv7	2022	No	Pytorch	YOLOv7Backbone	56.8
DAMO-YOLO	2022	No	Pytorch	MAE-NAS	50.0
YOLOv8	2023	No	Pytorch	YOLOv8CSPDarknet	53.9
YOLO-NAS	2023	No	Pytorch	NAS	52.2
YOLOv9	2024	No	Pytorch	ELAN	55.6

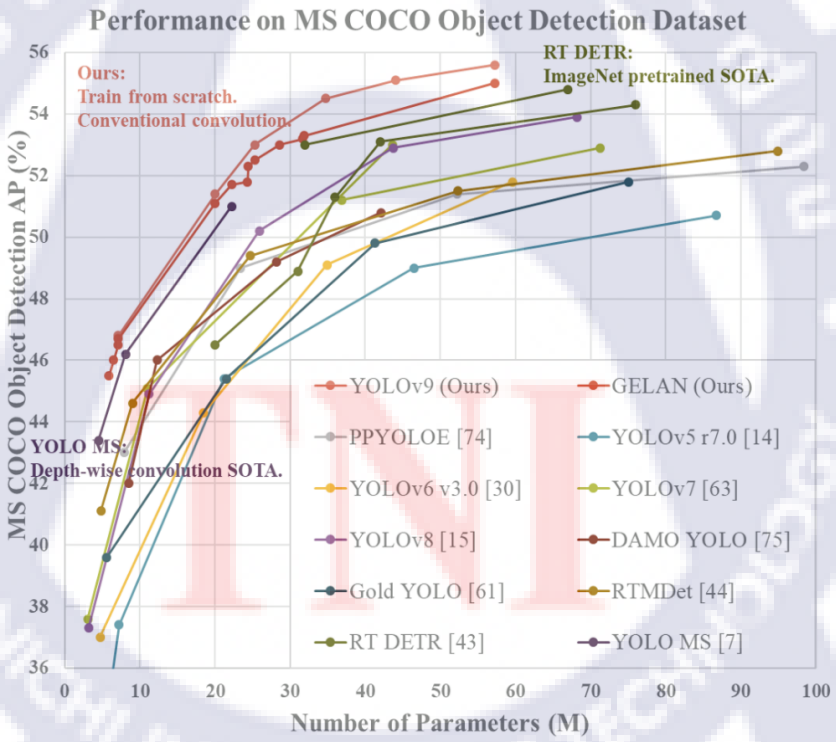

Figure 2.8 Performance of each version [20]

Table 2.2 Table of metrics from YOLOv8, YOLOv9 and YOLO-NAS [20][21]

Model	APval50:95 (%)	APval50 (%)	Parameters (M)	FLOPs (G)
YOLOv8-N	37.3	52.6	3.2	8.7
YOLOv8-S	44.9	61.8	11.2	28.6
YOLOv8-M	50.2	67.2	25.9	78.9
YOLOv8-L	52.9	69.8	43.7	165.2
YOLOv8-X	53.9	71.0	68.2	257.8
YOLO-NAS S	47.5	-	19.0	-
YOLO-NAS M	51.55	-	51.1	-
YOLO-NAS L	52.22	-	66.9	-
GELAN-S	46.7	63.0	7.2	26.7
GELAN-M	51.1	67.9	20.1	76.8
GELAN-C	52.5	69.5	25.5	102.8
GELAN-E	55.0	71.9	58.1	192.5
YOLOv9-S	46.8	63.4	7.2	26.7
YOLOv9-M	51.4	68.1	20.1	76.8
YOLOv9-C	53.0	70.2	25.5	102.8
YOLOv9-E	55.6	72.8	58.1	192.5

2.1.5 Confusion Matrix

Confusion matrix is a matrix that summarizes the performance of a machine learning model on a set of test data to measure the performance of the classification model.

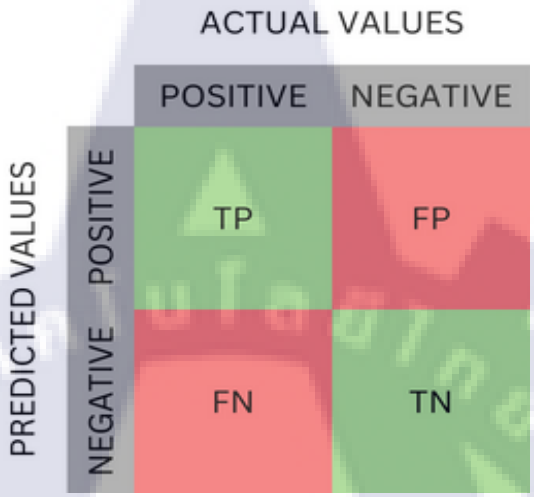

Figure 2.9 Confusion Matrix

- True Positive (TP) refers to a sample belonging to the positive class being classified correctly.
- True Negative (TN) refers to a sample belonging to the negative class being classified correctly.
- False Positive (FP) refers to a sample belonging to the negative class but being classified wrongly as belonging to the positive class.
- False Negative (FN) refers to a sample belonging to the positive class but being classified wrongly as belonging to the negative class.

And by analyzing the result in the form of recall, precision, accuracy or F1-score. The model's performance can be evaluated.

2.1.5.1 Recall

Recall is the ratio of number of true positive to the sum of true positive and false negative, to measure the effectiveness of a model.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

2.1.5.2 Precision

Precision is the ratio of number of true positive to the sum of true positive and false positive, to measure accuracy of model's positive prediction.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$$

2.1.5.3 Accuracy

Accuracy is the ratio of total correct case to total case, to measure the performance of model.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}}$$

2.1.5.4 Specificity

Specificity is the ratio of number of true negative to the sum of true negative and false positive, to measure the ability of model to identify true negative.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}}$$

2.1.5.5 F-1 score

By using both precision and recall, F1-score can evaluate the overall performance of classification model.

$$\text{F1 Score} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}}$$

2.1.5.6 Mean Average Precision (mAP)

mAP is a metric used to measure the performance of object detection models such as YOLO, Fast R-CNN etc. By using the ground-truth bounding box compare with detected box and return a result, to evaluate the accuracy of model detection.

$$\text{mAP} = \frac{1}{n}\sum_{k=1}^{k=n} \text{AP}_k$$

n = number of classes

AP_k= average precision of class k

2.2 Related Research

In the healthcare system, medication errors are one of the most problematic problems which can be reduced by developing a reliable system. In the case of this study, we will focus on drug identification systems, which have also been explored in diverse methodologies with each advantage and disadvantage. K. A. Hussaeni et al. [22] suggest the comparison between three neural networks architectures: Convolutional Neural Network and Support Vector Machines (CNN+SVM), Convolutional Neural Network and K-Nearest Neighbor (CNN+kNN) and ResNet-50 network. By performing in datasets obtained from National Library of Medicine (NLM). The CNN+kNN architecture returned with the best result of 90.8% accuracy. However, the generalizability still remains a concern, since their model was trained by NLM library which has controlled environment and lighting. So, the result might not be the same under different environments. And with the same data source (NLM) but with different approaches. By combining YOLOv5 with ResNet-32, J. Heo et al. [23] Received an accuracy of 85.6% and 78% from controlled environment and non-controlled respectively. Lastly, MobileDeepPill [24], a proposed CNN architecture that integrates pill color, gradients, and shape measurements to compare between consumer and reference images, also introduced which got 73.7% Top-1 accuracy and 95.6% Top-5 accuracy in two-side pill recognition in this same data source. In addition to model trained by NLM datasets, P. Chupawa and K. Kanjanawanishkul [25] proposed

a technique to extract imprint from prepared image more reliably and get the accuracy of 94.4% from his proposed model. Likewise, some researchers tried different approaches by identifying drug from label [26] or packaging [27] using the neural network which results in 88% of precision and 95.99% accuracy respectively. These are the examples of the past research conducted in this area for drug identification. Furthermore, the exploration of related applications, as found in referenced literature [28,29] shows the advancement and practical implementation in pill identification systems.

Table 2.3 Related research

No.	Topics	Objectives	Techniques	Datasets	Results
1	CNN-Based Pill Image Recognition for Retrieval Systems	To investigate the challenging problem of image retrieval and develop an efficient image retrieval system based on deep learning and the k-Nearest Neighbor (k-NN) classifier	propose an image retrieval approach based on two steps: (1) a preprocessing phase with features extraction and (2) classification using k-Nearest Neighbors (k-NN), Support Vector Machine (SVM) and ResNet.	Using pill images from National Library of Medicine (NLM) dataset, included 7000 pill images from 1000 unique pills.	Their model is capable of deriving an accuracy of 90.8%. and improve the pill image retrieval accuracy by 10% compared to existing models.
2	An Accurate Deep Learning–Based System for Automatic Pill Identification: Model Development and Validation	Proposed a deep learning-based system to train with pill image and imprinted characters.	A pill image is fed into the object detection model YOLOv5 and image classification model ResNet-32. Then used classified feature as inputs for the RNN-based character-level language model	2 data sets of pill images provided by MFDS and NLM. Totals of 24404 pills images	Result show accuracy of 85.6% from pill database and 78% from consumer-granted images

Table 2.3 Related research (Cont.)

No.	Topics	Objectives	Techniques	Datasets	Results
3	DLI-IT: a deep learning approach to drug label identification through image and text embedding	to accelerate the processing time by using recognized text for retrieval instead of traditional input from a keyboard.	By training a Connectionist Text Proposal Network (CTPN) to crop the raw image into sub-images based on the text. The texts are recognized through the Tesseract OCR Engine as one document for each raw image. Then transform these documents into vectors and find the most similar reference images to the test image	image samples were collected from Daily-Med; Totals of 4134 label images (749 opioid and 2365 non-opioid drug label images.)	Their model achieves up to 88% of precision in drug label identification
4	Pill Identification with Imprints Using a Neural Network	focused on using deep learning method to identify pills that identical in color and shape through imprints on pills	The algorithm consisted of three main stages. In the first stage, images were preprocessed into binary image. Second stage, imprint feature were extracted and then used as input vector of the feedforward neural network with supervised learning	Prepared images of 6 pill types, each type has 60 images, resulting in 360 images in total	accuracy of 94.4% from their proposed model

Table 2.3 Related research (Cont.)

No.	Topics	Objectives	Techniques	Datasets	Results
5	A drug identification model developed using deep learning technologies: experience of a medical center in Taiwan	to illustrate how 'look-alike' error can be captured and appropriate solution to extract more detailed nuance differences can be utilized in distinguishing look-alike objects.	Using deep learning framework of You Only Look Once (YOLO) for implementation of the proposed deep learning. The F1 score, was used as the performance criterion.	images of 250 types of blister-packaged drug from the Out-Patient Department of a medical center. 72 images were taken for each side of each type of drug resulting in a total of 36,000 images.	outcome consists of accuracy 95.99% from back-side model and 93.72% from front-side model
6	MobileDeepPill: A Small-Footprint Mobile Deep Learning System for Recognizing Unconstrained Pill Images	To introduce the first mobile vision system that achieves real-world applicable performance for recognizing unconstrained pill images.	The system architecture of MobileDeepPill consists of 2 stages. (1) Training stage: By localize and segments the pill of every image. Then perform data augmentation to generate augmented images to increase training samples. Then use	Using the official NLM Pill Image dataset. Contain two categories of pill images: (1) Reference Images: contains 2000 reference images of the 1000 pills,	MobileDeepPill achieves pill image recognition performance with 52.7% Top-1 accuracy and 81.7% Top-5 accuracy in one-side pill recognition scheme as well

Table 2.3 Related research (Cont.)

No.	Topics	Objectives	Techniques	Datasets	Results
			every image as input in three independent CNNs then import as Teacher CNNs (2) Inference stage: By segments the pill images. The segmented image then fed into the Student CNNs to extract CNN features. These features will be compared with the CNN features of all the reference images. Finally, the ranking based on the similarity Is generated. Based on the ranking, the top pill candidates are returned to the user.	(2) Consumer Images: contains 5000 consumer images of the 1000 pills.	as 73.7% Top-1 accuracy and 95.6% Top-5 accuracy in two-side pill recognition scheme.

Chapter 3
Research Methodology

3.1 Proposed method

To achieve our goal, The algorithm in this study was consisted of three main stages. In the first stage, by preparing our datasets from collecting, preprocessing and localizing each pill into library. Next stage, we divided our dataset into 3 subsets: training, validating and testing. Then using both training subsets and validating subsets as an input in CNN network to train model. Lastly, with the best model, we used the testing datasets to measure the outcome.

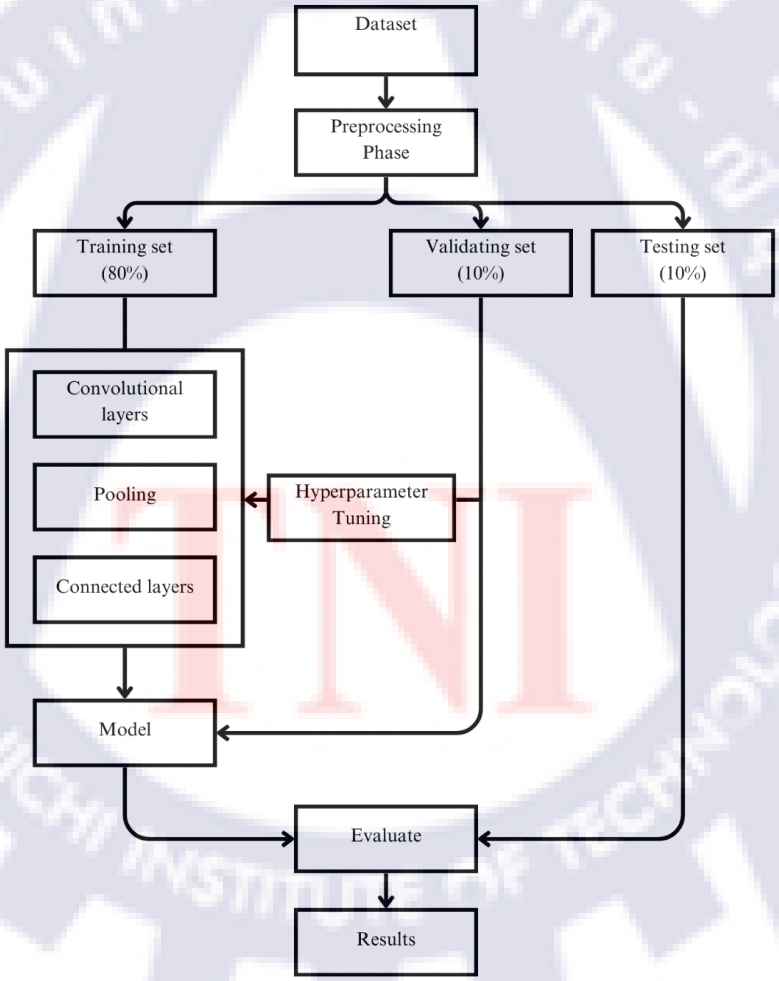

Figure 3.1 Proposed method

3.2 Datasets

This study collected samples image by photograph from consist of 3 generic drugs, which include acetaminophen, ibuprofen, naproxen. And each class was also collected data from 3 different manufacturers in Thailand. Which concludes 9 classes in total. in each class, a photo was taken in different angles with the same background for approximately 500-550 images per class, resulting in a total of 4710 images.

Figure 3.2 Sample of Coprofen

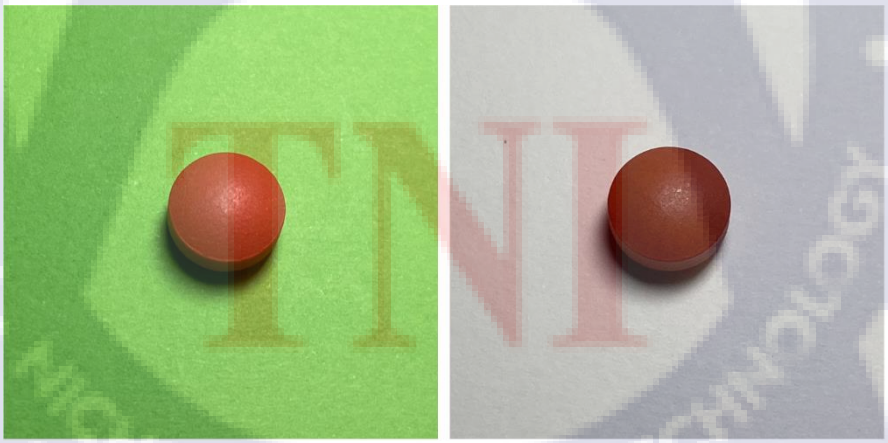

Figure 3.3 Sample of Adavil

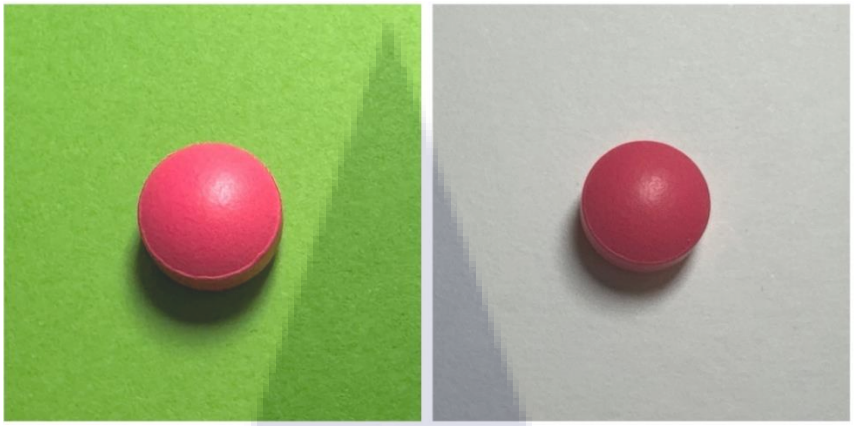

Figure 3.4 Sample of Cefen

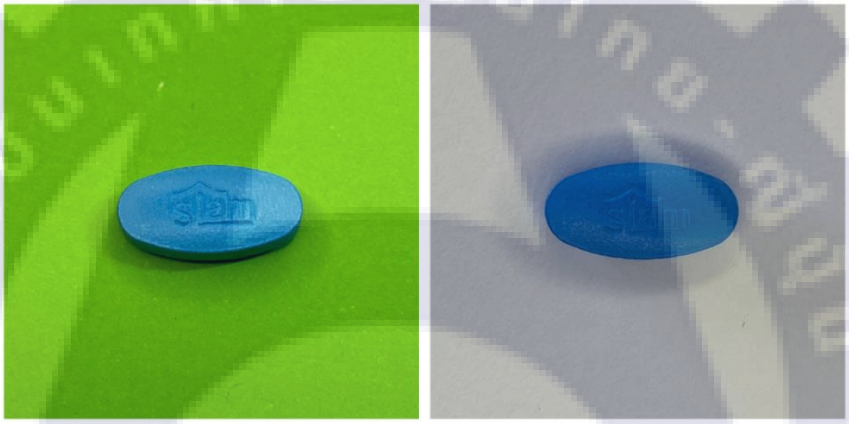

Figure 3.5 Sample of Annoxen

Figure 3.6 Sample of Soproxen

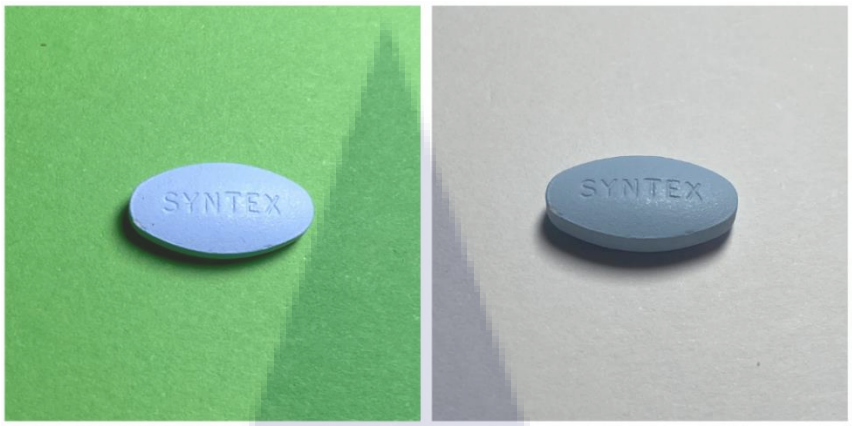

Figure 3.7 Sample of Synflex

Figure 3.8 Sample of Cemol Central

Figure 3.9 Sample of Sara

Figure 3.10 Sample of Tylenol

Then all 4710 images were allocated into three distinct subsets: training, validation, and testing data. By dividing into a predetermined ratio approximately 80% , 10% and 10% for training, validating, and testing respectively.

Table 3.1 Data Splitting Details

Class	Generic name	Train	Valid	Test	Total
Coprofen	Ibuprofen	468	58	58	584
Adavil	Ibuprofen	400	50	49	499
Cefen	Ibuprofen	400	50	49	499
Annoxen	Naproxen	455	57	57	569
Soproxen	Naproxen	400	50	49	499
Synflex	Naproxen	400	50	49	499
Cemol Central	Acetaminophen	451	56	56	563
Sara	Acetaminophen	400	50	49	499
Tylenol	Acetaminophen	400	50	49	499

Table 3.2 The Amount of Data used for Testing and Evaluating models

Total Data	Train set		Test set
	Training	Validating	
4710	3774	471	465

3.3 Model Architecture

The dataset was used as an input to train CNN network. In this study, You Only Look Once (YOLO) algorithm was used as the solution framework for deep learning. Starting by inputting prepared image of size 640*640 pixels into the CNN model. Then, the extracted features are fed into the classification layers and get output as a predicted class. The hyperparameters settings are listed in Table V.

For training model, the dataset was separated into training and validating sets. The training set trained the CNN network to create the model and validating set was used to evaluate the performance of trained model. The best model was evaluated using test dataset.

In this study, the system was configured with Python version 3.10.12, PyTorch version 2.1.0, and CUDA version 12.1 (running on NVIDIA GPU A100-SXM4-40GB, VRAM 40514 MB). The Operational System is Linux Ubuntu 22.04.3 LTS (Jammy Jellyfish), running on Google Colab Pro+. Moreover, CPU specifications include Intel (R) Xeon (R) CPU @ 2.00GHz., 8-core CPU, 51 GB of RAM, and 202 GB of data storage capacity.

Table 3.3 Hyperparameters

Parameters	**Value**
Optimizer	Adam
Number of Classes	9
Color	RGB
Input Image size	640*640
Batch size	16

3.4 Outcome measurement

For the outcome measurement, confusion matrices were used to record the predicted results. Then, the training time, number of epochs, precision, recall and F1 score were collected as a result. The best performance model would be recognized by the highest F1 and mAP score. In this section, we have used our last datasets: testing datasets to test our best model and measure an outcome.

Chapter 4

Results

In this experiment, the evaluation of the different YOLO algorithms such as YOLOv8-nano, YOLOv8-small, YOLOv8-medium, YOLOv8-large, YOLOv8-extra-large, YOLOv9-c, YOLOv9-e, GELAN-c, GELAN-e, YOLO-NAS-small, YOLO-NAS-medium, and YOLO-NAS-large. These algorithms are compared based on the evaluation metrics including precision, recall, F1 score and mean average precision

Table 4.1 Experimental Results

Model	Precision	Recall	F1 Score	mAP50
YOLOv8-n	0.991	0.997	0.994	0.995
YOLOv8-s	0.971	0.968	0.969	0.991
YOLOv8-m	0.983	0.987	0.985	0.994
YOLOv8-l	0.971	0.973	0.972	0.993
YOLOv8-x	0.968	0.955	0.961	0.995
YOLO-NAS-s	0.165	0.955	0.258	0.857
YOLO-NAS-m	0.182	0.903	0.279	0.851
YOLO-NAS-l	0.314	0.937	0.431	0.849
GELAN-c	0.973	0.973	0.973	0.987
GELAN-e	0.954	0.974	0.964	0.991
YOLOv9-c	0.983	0.996	0.989	0.995
YOLOv9-e	0.965	0.964	0.964	0.995

4.1 YOLOv8-n

Model trained with YOLOv8-n has shown the results of 99.1% in precision, 99.7% in recall, 99.4% in F-1 score and 99.5% in mean average precision. As shown in the figure below.

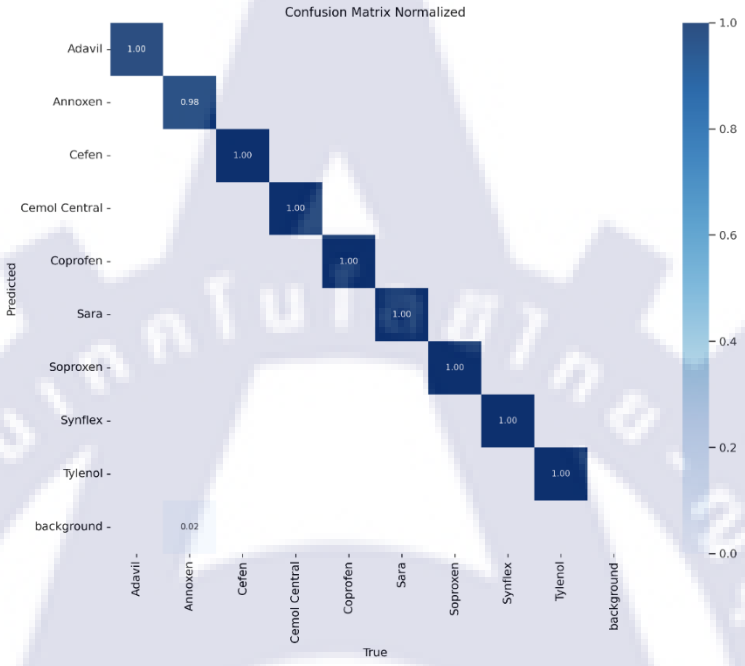

Figure 4.1 Confusion Matrix of YOLOv8-n model

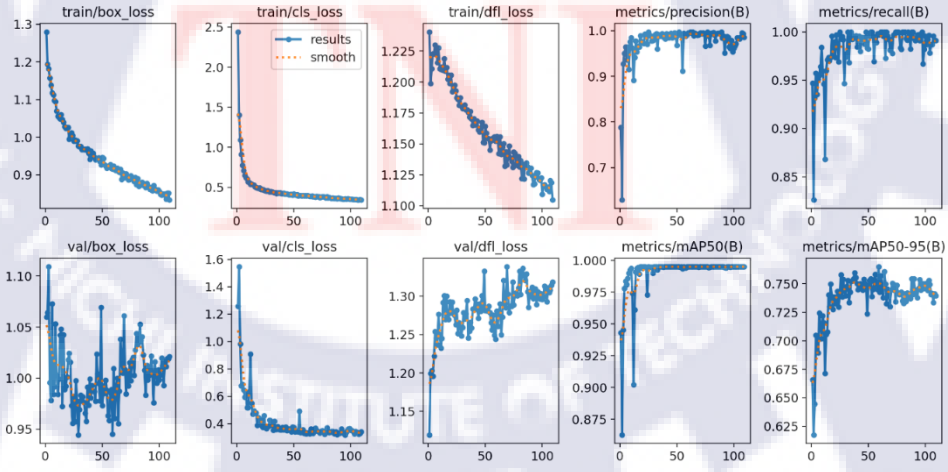

Figure 4.2 Experimental Results of YOLOv8-n model

4.2 YOLOv8-s

Model trained with YOLOv8-s has shown the results of 97.1% in precision, 96.8% in recall, 96.9% in F-1 score and 99.1% in mean average precision. As shown in the figure below.

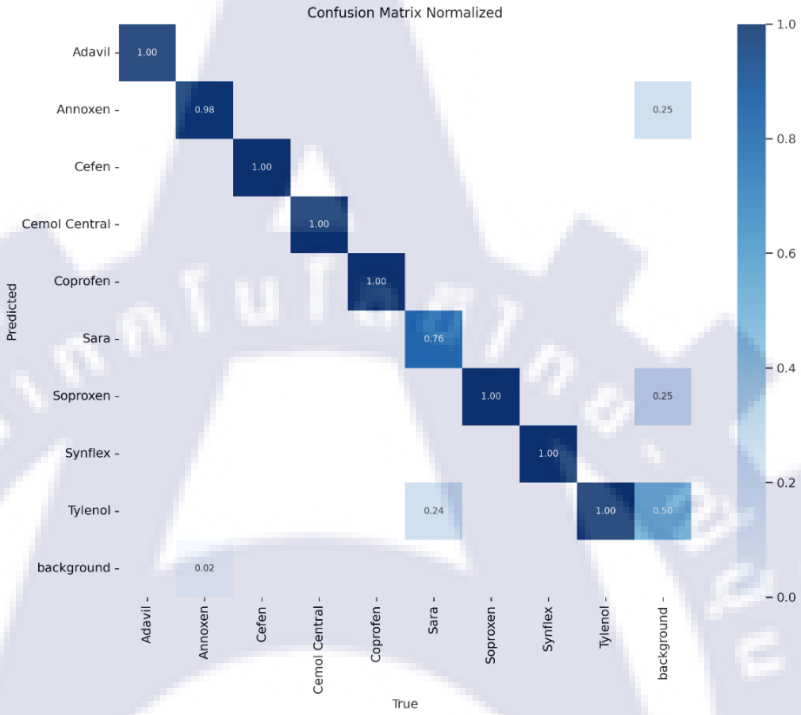

Figure 4.3 Confusion Matrix of YOLOv8-s model

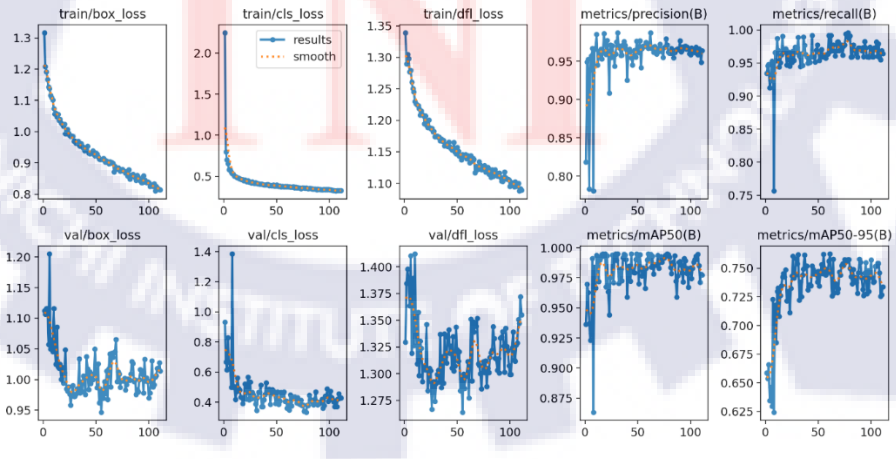

Figure 4.4 Experimental Results of YOLOv8-s model

4.3 YOLOv8-m

Model trained with YOLOv8-m has shown the results of 98.3% in precision, 98.7% in recall, 98.5% in F-1 score and 99.4% in mean average precision. As shown in the figure below.

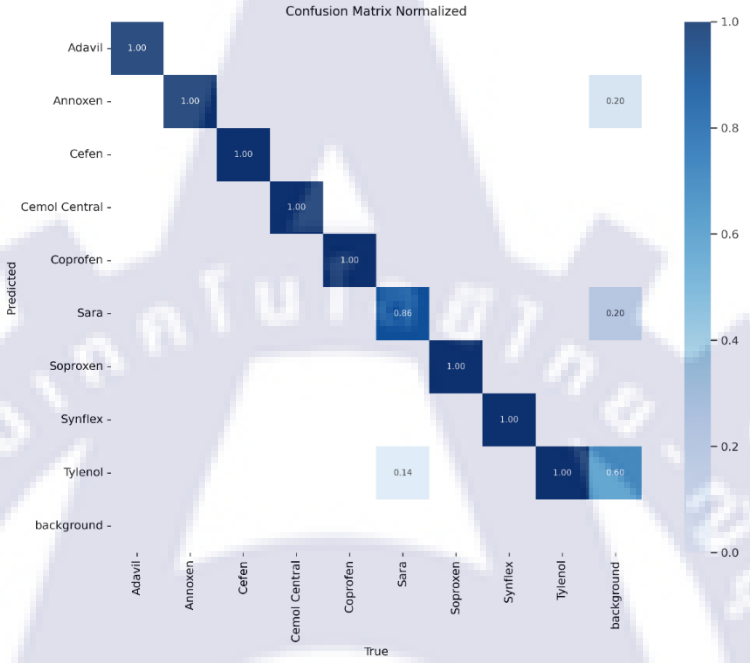

Figure 4.5 Confusion Matrix of YOLOv8-m model

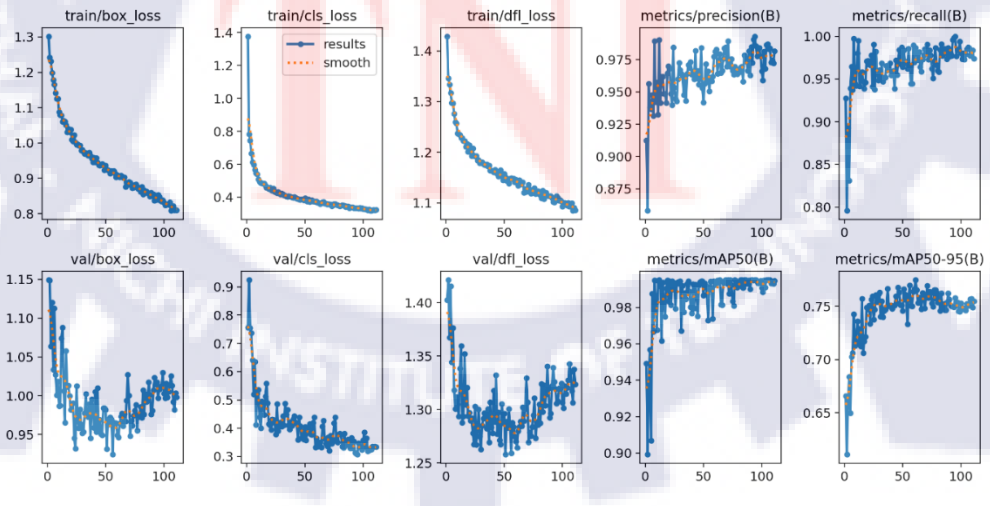

Figure 4.6 Experimental Results of YOLOv8-m model

4.4 YOLOv8-l

Model trained with YOLOv8-l has shown the results of 97.1% in precision, 97.3% in recall, 97.2% in F-1 score and 99.3% in mean average precision. As shown in the figure below.

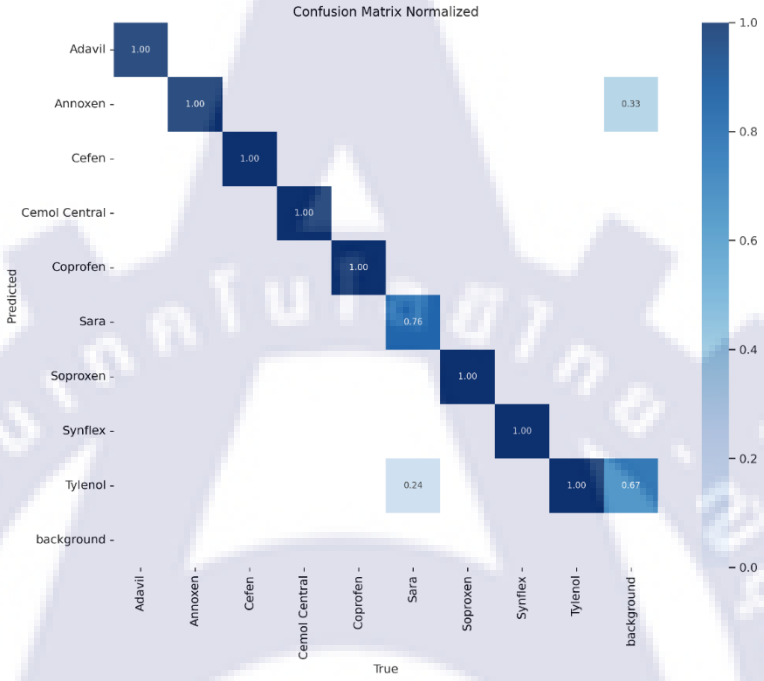

Figure 4.7 Confusion Matrix of YOLOv8-l model

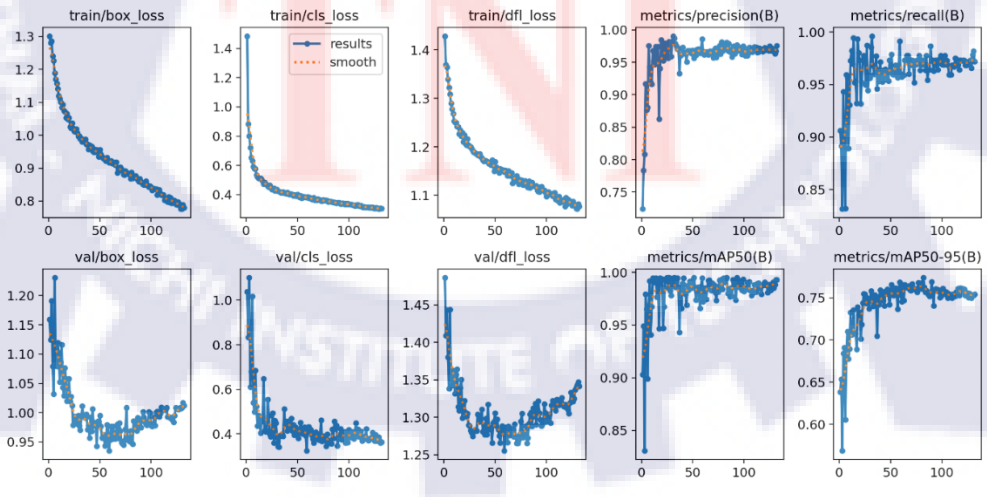

Figure 4.8 Experimental Results of YOLOv8-l model

4.5 YOLOv8-x

Model trained with YOLOv8-x has shown the results of 96.8% in precision, 95.5% in recall, 96.1% in F-1 score and 99.5% in mean average precision. As shown in the figure below.

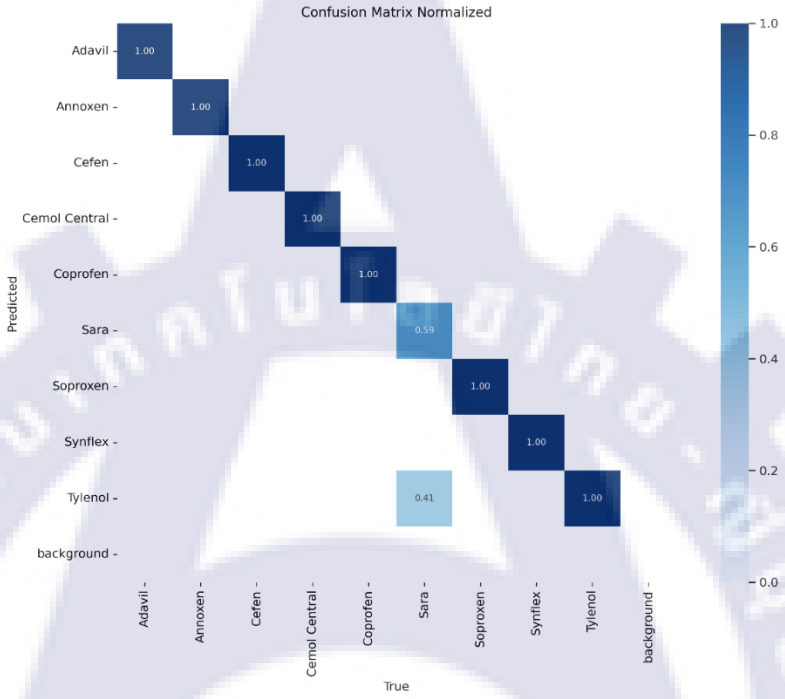

Figure 4.9 Confusion Matrix of YOLOv8-x model

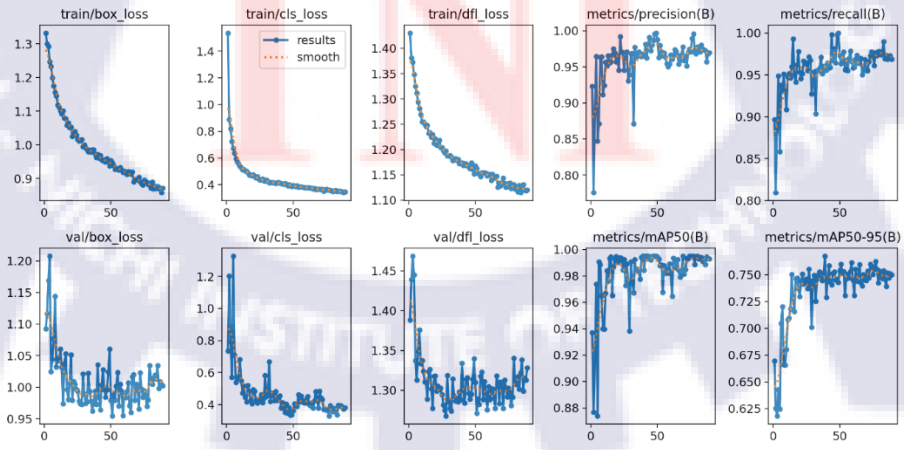

Figure 4.10 Experimental Results of YOLOv8-x model

4.6 GELAN-c

Model trained with GELAN-c has shown results of 97.3% in precision, 97.3% in recall, 97.3% in F-1 score and 98.7% in mean average precision. As shown in the figure below.

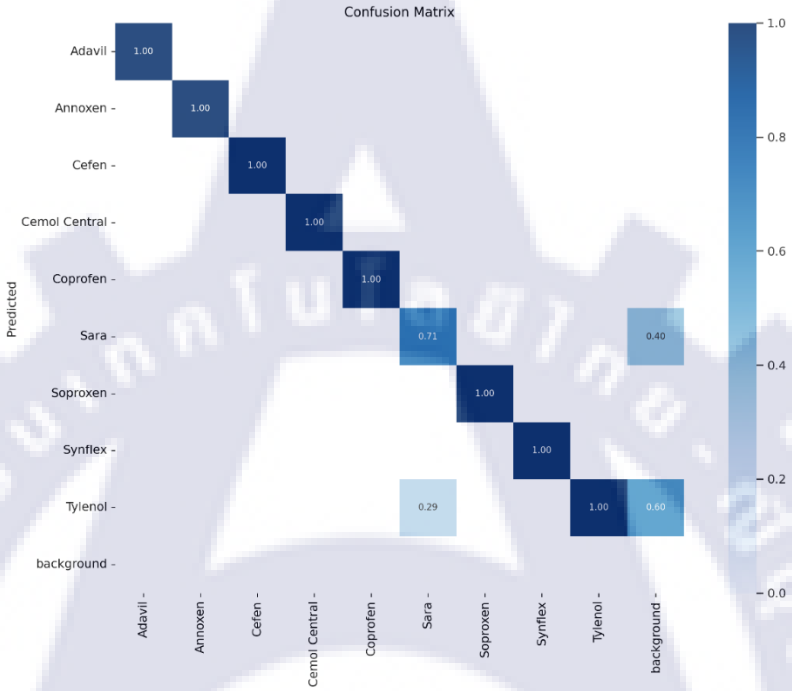

Figure 4.11 Confusion Matrix of GELAN-c model

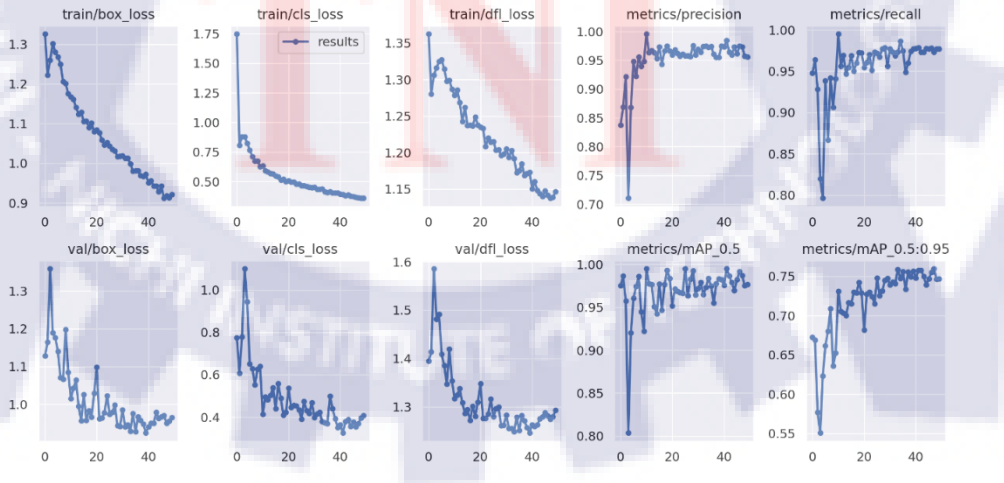

Figure 4.12 Experimental Results of GELAN-c model

4.7 GELAN-e

Model trained with GELAN-e has shown the results of 95.4% in precision, 97.4% in recall, 96.4% in F-1 score and 99.1% in mean average precision. As shown in the figure below.

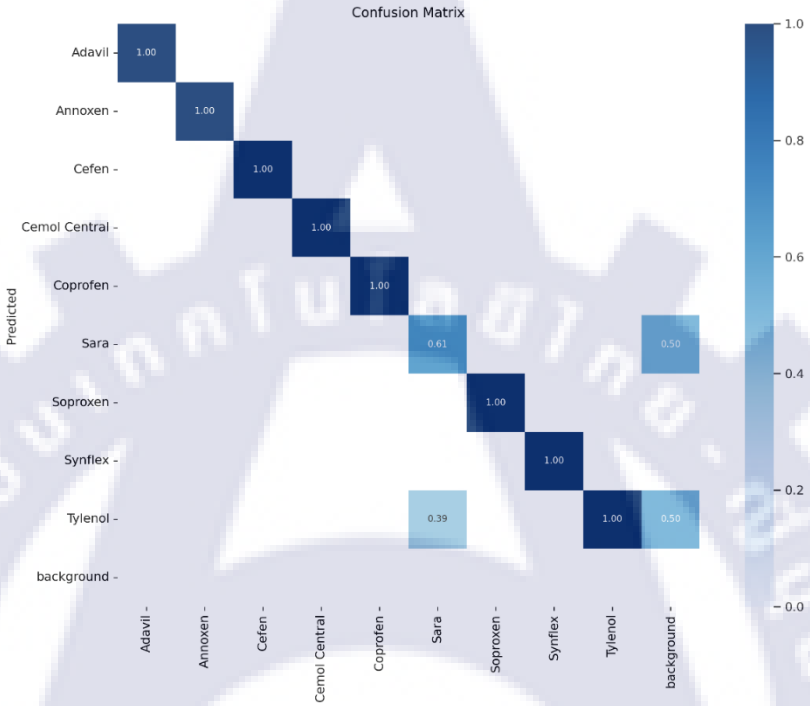

Figure 4.13 Confusion Matrix of GELAN-e model

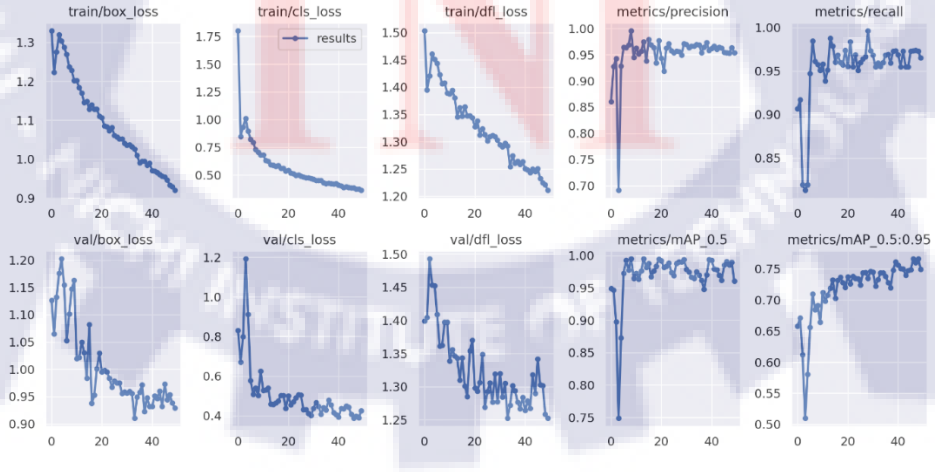

Figure 4.14 Experimental Results of GELAN-e model

4.8 YOLOv9-c

Model trained with YOLOv9-c has shown the results of 98.3% in precision, 99.6% in recall, 98.9% in F-1 score and 99.5% in mean average precision. As shown in the figure below.

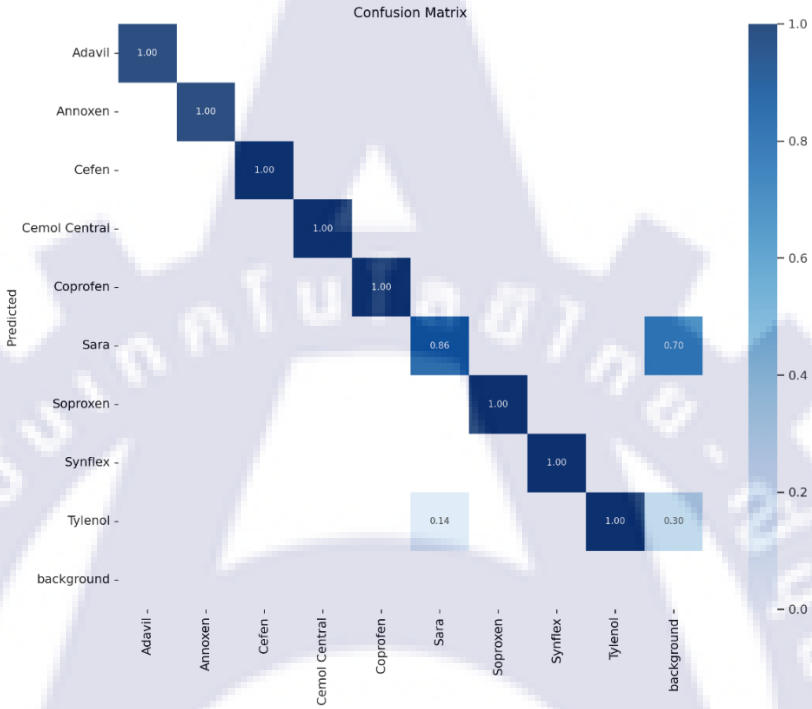

Figure 4.15 Confusion Matrix of YOLOv9-c model

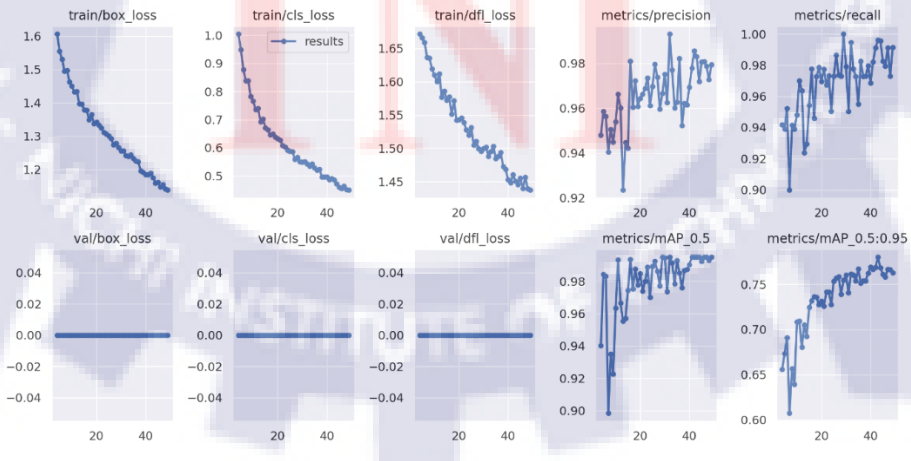

Figure 4.16 Experimental Results of YOLOv9-c model

4.9 YOLOv9-e

Model trained with YOLOv9-e has shown the results of 96.5% in precision, 96.4% in recall, 96.4% in F-1 score and 99.5% in mean average precision. As shown in the figure below.

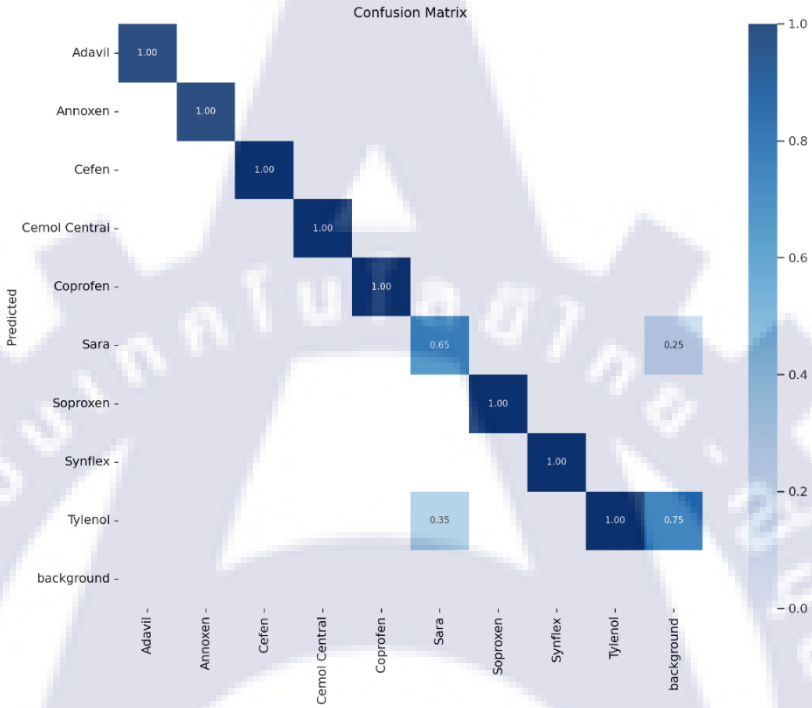

Figure 4.17 Confusion Matrix of YOLOv9-e model

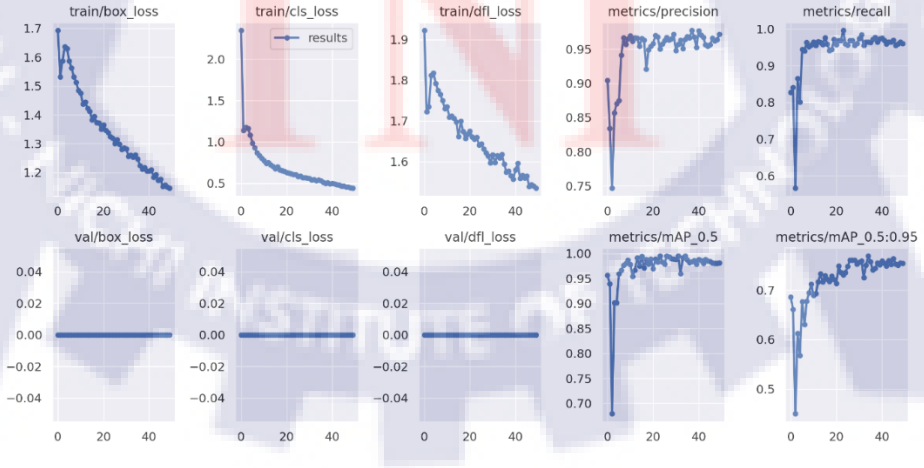

Figure 4.18 Experimental Results of YOLOv9-e model

4.10 YOLO-NAS-s

Model trained with YOLO-NAS-s has shown the results of 16.5% in precision, 95.5% in recall, 25.8% in F-1 score and 85.7% in mean average precision. As shown in the figure below.

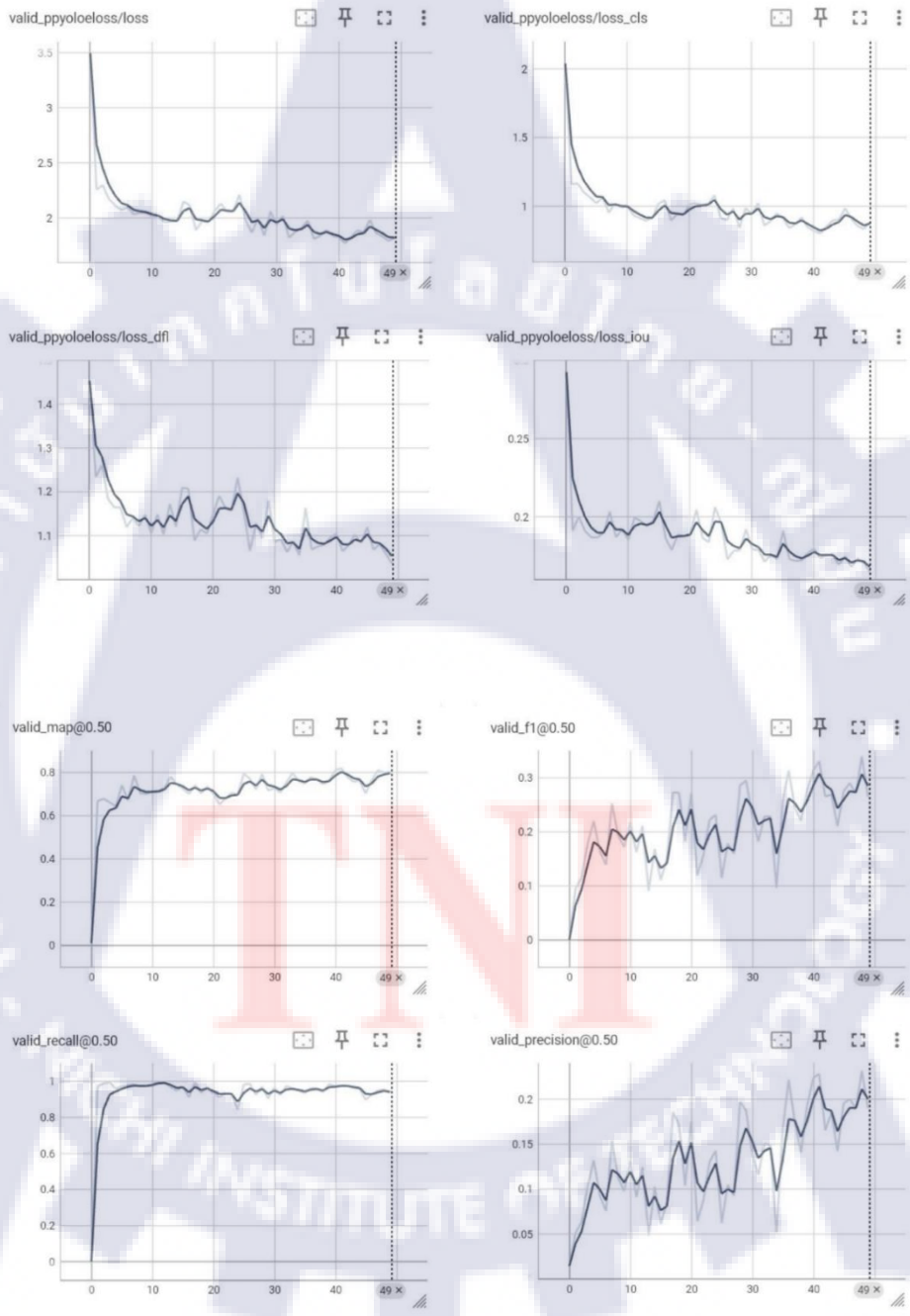

Figure 4.19 Experimental Results of YOLO-NAS-s model

4.11 YOLO-NAS-m

Model trained with YOLO-NAS-m has shown the results of 18.2% in precision, 90.3% in recall, 27.9% in F-1 score and 85.1% in mean average precision. As shown in the figure below.

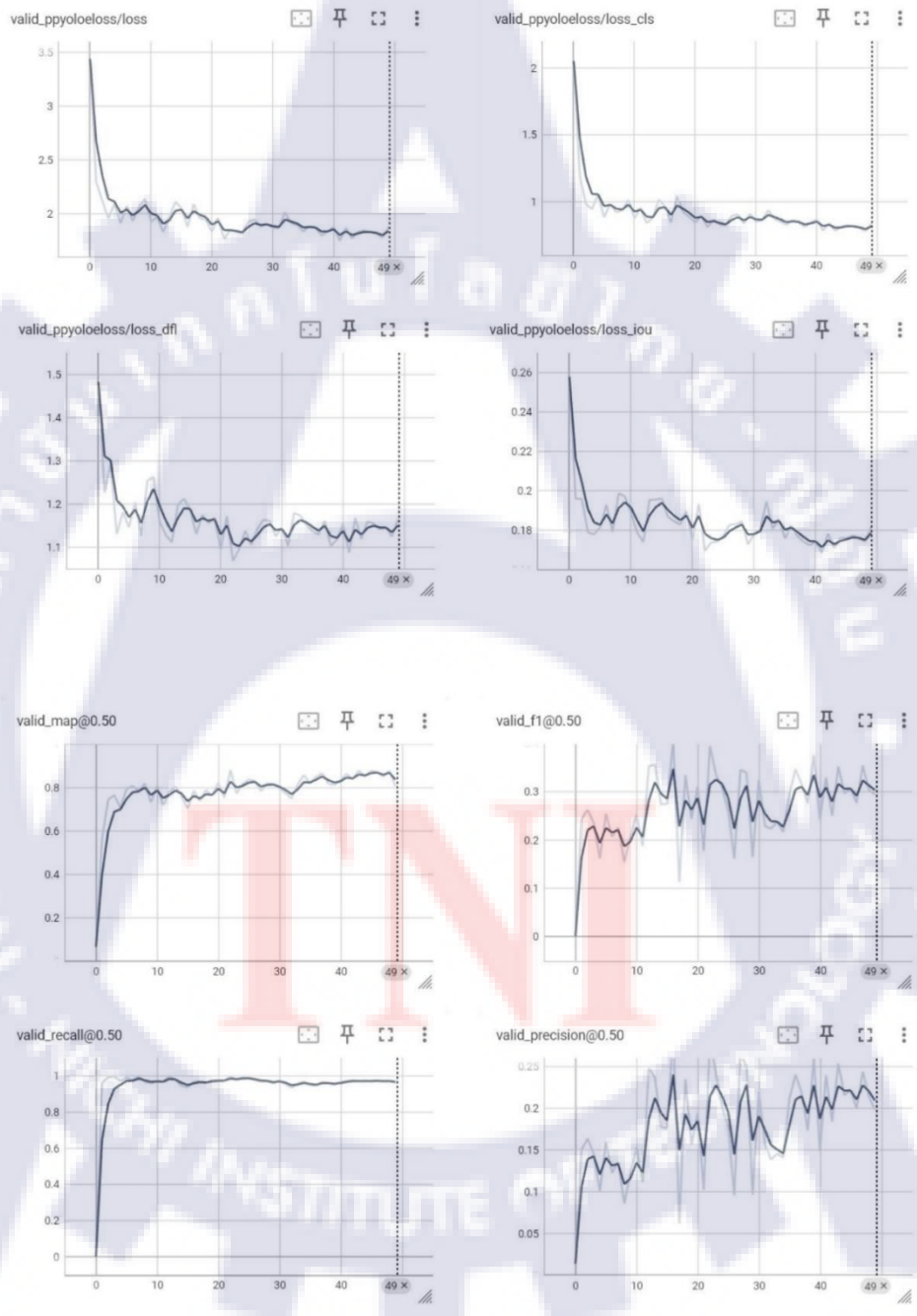

Figure 4.20 Experimental Results of YOLO-NAS-m model

4.12 YOLO-NAS-l

Model trained with YOLO-NAS-l has shown the results of 31.4% in precision, 93.7% in recall, 43.1% in F-1 score and 84.9% in mean average precision. As shown in the figure below.

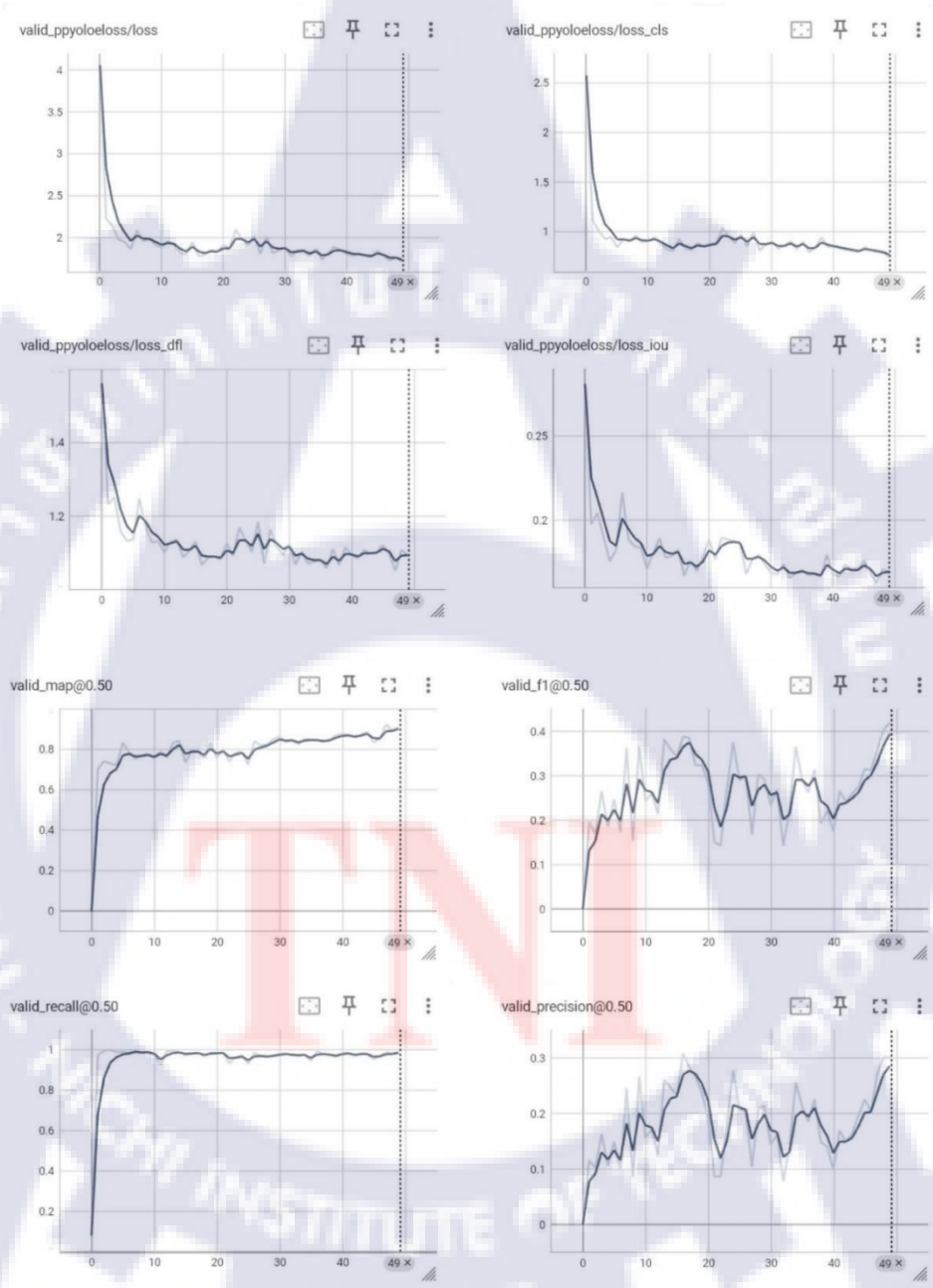

Figure 4.21 Experimental Results of YOLO-NAS-l model

Chapter 5
Conclusion and Future Works

5.1 Conclusion

In this study, researchers applied various versions of YOLO algorithms to the prepared datasets and compare the performance of each version by using standard metrics including mean average precision, precision, recall and F-1 scores. in order to find the best algorithm to train model for drug identification tasks. From the same dataset, the YOLOv8 version shows the superior result compared with YOLOv9 and YOLO-NAS version. Specifically, YOLOv8-nano has shown the best performance compared to other versions, with a result of 99.4% F1 score and 99.5% mAP score. In contrast, all YOLO-NAS and YOLOv9 versions underperform YOLOv8 across all metrics. In summary, according to the experimental results in performance and efficiency, YOLOv8-nano shows the most outstanding outcome for drug identification and require the least computational resources among others.

The reason that we thought that might explain this outcome are due to overfitting and data mismatch. In large transfer learning models are trained on massive datasets like ImageNet, which might not perfectly represents our specific task and resulting to trained model capture irrelevant features from pre-trained data that dominate on smaller custom dataset instead. Additionally, the pre-trained data might have significantly different in format and content from our custom dataset leading to poor performance in larger model.

5.2 Limitations

One major limitation of this study is the limited amount of data and dependency on high-quality images in the training process. Limited or biased datasets can significantly affect the model's ability to identify the drugs or to generalize to new pills in the future. Additionally, in real world environment such as lighting condition, shadow, or damaged pills could lead to variability in pill appearance leading to challenge for model to accurately identify pills in practical. Next, the training of these models requires heavy computational resource and time-consuming which limited our

experiment in number of epochs or configuration of parameter in trained model. Furthermore, the interpretability of these deep learning models remains an issue due to their property as a black box, making it difficult to understand their decision-making processes and trust their prediction fully. Addressing these limitations is crucial for enhancing the reliability of pill identification systems in practical applications in the future.

5.3 Future works

For future work, we suggest to focus on several directions. Firstly, we suggest to train with more diverse dataset including the change in pill types, try to take pictures from different angles and different lightning conditions or even use a picture from lower quality camera, might help to imitate the real world conditions better and might more practical. Secondly, exploring other architectures beyond YOLO variant such as EfficientDet, DETR or transformers-based models may yield better performance. In this study, researchers took advantage of transfer learning by using COCO datasets but experiment this techniques with models pre-trained on larger, more diverse datasets might offer the enhancement of trained model. Optimizing these models for real-time deployment on mobile or embedded devices is also worth to experiment since it might ensure the efficiency of our models. Integrating multimodal approaches that combine image data with other modalities could further improve accuracy and reliability as well. Lastly, integrating these models into practical applications is also essential to fulfill our goals in the future.

References

References

[1] A. Hodkinson et al., "Preventable medication harm across health care settings: A systematic review and meta-analysis," *BMC Med*, vol. 18, no. 1, pp. 313, November 2020, doi: 10.1186/s12916-020-01774-9.

[2] Center for Drug Evaluation and Research, "*Medication Errors Related to CDER-Regulated Drug Products,*" FDA. [Online]. Available: https://www.fda.gov/drugs/drug-safety-and-availability/medication-errors-related-cder-regulated-drug-products. [Accessed: October 16, 2024].

[3] R. Bryan et al., "The problem of look-alike, sound-alike name errors: Drivers and solutions," *Br J Clin Pharmacol*, vol. 87, no. 2, pp. 386–394, February 2021, doi: 10.1111/bcp.14285.

[4] World Health Organization, "*Medication Without Harm,*" [Online]. Available: https://www.who.int/publications/i/item/WHO-HIS-SDS-2017.6. [Accessed: October 16, 2024].

[5] G. Wiederhold and J. McCarthy, "Arthur samuel: Pioneer in machine learning," *IBM Journal of Research and Development*, vol. 36, no. 3, pp. 329–331, June 1992, doi: 10.1147/rd.363.0329.

[6] I. Goodfellow et al., "*Deep Learning*," [Online]. Available: https://www.deeplearningbook.org/. [Accessed: October 16, 2024].

[7] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN COMPUT. SCI.*, vol. 2, no. 6, pp. 420, August 2021, doi: 10.1007/s42979-021-00815-1.

[8] Z. Zhang et al., "Deep learning in omics: A survey and guideline," *Briefings in Functional Genomics*, vol. 18, no. 1, pp. 41–57, February 2019, doi: 10.1093/bfgp/ely030.

[9] "*4. Fully Connected Deep Networks - TensorFlow for Deep Learning,*" [Online].Available: https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html. [Accessed: October 16, 2024].

[10] J. Redmon et al., "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, June 27-30, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[11] J. Redmon and A. Farhadi, "*YOLO9000: Better, Faster, Stronger*," [Online]. Available: https://arXiv.1612.08242, doi: 10.48550/ [Accessed: December 15, 2023].

[12] J. Redmon and A. Farhadi, "*YOLOv3: An Incremental Improvement*," [Online]. Available: https://arXiv.1804.02767, doi: 10.48550. [Accessed: December 15, 2023].

[13] A. Bochkovskiy et al., "*YOLOv4: Optimal Speed and Accuracy of Object Detection*," [Online]. Available: https://arXiv.2004.10934, doi: 10.48550/ [Accessed: December 15, 2023].

[14] G. Jocher et al., "*Ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation - NASA/ADS*," [Online]. Available: https://ui.adsabs.harvard.edu/abs/2022zndo...7347926J/abstract. [Accessed: October 17, 2024].

[15] C. Li et al., "*YOLOv6 v3.0: A Full-Scale Reloading*," [Online]. Available: https://arXiv.2301.05586, doi: 10.48550/ [Accessed: December 14, 2023].

[16] C. Y. Wang et al., "*YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-time Object Detectors*," [Online]. Available: https://arXiv.2207.02696, doi: 10.48550/ [Accessed: December 13, 2023].

[17] G. Jocher et al., "*YOLOv8*," Ultralytics [Online]. Available: https://docs.ultralytics.com/ models/yolov8. [Accessed: October 16, 2024].

[18] S. Aharon et al., "*Deci-AI/Super-Gradients*," Github [Online]. Available: https://github.com/Deci-AI/super-gradients. [Accessed: October 16, 2024].

[19] J. Terven and D. M. C. Esparza, "*A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond*," [Online]. Available: https://arXiv.2304.00501, doi: 10.48550/ [Accessed: December 16, 2023].

[20] C. Y. Wang et al., "*YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information*," [Online]. Available: https://arXiv.2402.13616, doi: 10.48550/ [Accessed: December 13, 2023].

[21] V. Singh, "*YOLO-NAS: New Object Detection Model Beats YOLOv6 & YOLOv8*," [Online]. Available: https://learnopencv.com/yolo-nas/. [Accessed: October 16, 2024].

[22] K. A. Hussaeni et al., “CNN-based pill image recognition for retrieval systems,” *Applied Sciences*, vol. 13, no. 8, pp. 5050, April 18, 2023, doi: 10.3390/ app13085050.

[23] J. Heo et al., “An accurate deep learning-based system for automatic pill identification: Model development and validation (Preprint),” *Journal of Medical Internet Research*, vol. 25, no.4, pp. 1-13, July 2022, doi: 10.2196/ 41043.

[24] X. Zeng et al., “Mobile deep pill: A small-footprint mobile deep learning system for recognizing unconstrained pill images,” in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, in MobiSys’17*. New York, NY, USA, June 19-23, 2017, pp. 56–67. doi: 10.1145/3081333.3081336.

[25] P. Chupawa and K. Kanjanawanishkul, “Pill identification with imprints using a neural network,” *Mahasarakham International Journal of Engineering Technology*, vol. 1, no. 1, pp. 3035, January 2015, doi: 10.14456/MIJET. 2015.7.

[26] X. Liu et al., “DLI-IT: A deep learning approach to drug label identification through image and text embedding,” *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, pp. 68, April 2020, doi: 10.1186/s12911-020-1078-3.

[27] H. W. Ting et al., “A drug identification model developed using deep learning technologies: Experience of a medical center in Taiwan,” *BMC Health Services Research*, vol. 20, no. 1, pp. 312, April 2020, doi:10.1186/s12913-020-05166-w.

[28] A. Cunha et al, “HelpmePills: A mobile pill recognition tool for elderly persons,” *Procedia Technology*, vol. 16, no. 5, pp. 1523–1532, January 2014, doi: 10.1016/j.protcy.2014.10.174.

[29] N. Pornbunruang et al., “Drugtionary: drug pill image detection and recognition based on deep learning,” in *Proceedings of the 18th International Conference on Computing and Information Technology*, IC2IT 2022, Kanchanaburi, May 19-20, 2022, pp. 43–52. doi: 10.1007/978-3-030-99948-3_5.

Biography

Name Mr. Sorayus Kittisorayut

Date of Birth January 30, 1997

Address 44 soi Prawit Lae Puen 1/2, Udomsuk
Bangchak, Bangkok, 10260
Tel: 063-695-6254
E-mail: Ki.sorayus_st@tni.ac.th

Education Background

2024	Master of Science in Information Technology Thai-Nichi Institute of Technology
2016	Doctor of Medicine Srinakharinwirot University

Work Experience

June 2024 – present	Resident Ramathibodi Hospital
June 2024 – present	Physicians Ramathibodi Hospital
June 2021 – May 2024	Physicians AngThong Hospital