Concealed Object Detection: Depth-wise and Depth-wise Separable YOLO Models

DISSERTATION REPORT SUBMITTED
FOR THE AWARD OF DEGREE OF

M. Sc. (DATA SCIENCE)

BY

LAKSHMY SANTHOSH
Roll No: 222828

UNDER THE GUIDANCE OF

PROF. SINGARA SINGH

COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
CENTRAL UNIVERSITY OF HARYANA
MAHENDERGARH, HARYANA, INDIA-123031
JUNE 2024

Certificate

I hereby declare that the work being presented in this thesis, in fulfillment of the requirements for the award of degree of **M. Sc. (Data Science)** submitted in Department of Computer Science and Information Technology, Central University of Haryana, is an authentic record of my own work carried out under the supervision of Dr. Singara Singh, Professor, Department of Computer Science and Information Technology and refers other researcher works which are duly listed in the reference section. The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(**Lakshmy Santhosh**)
Registration No. 222828

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge and belief.

(**Dr. Singara Singh**)
Professor
Computer Science and Information Technology
Central University of Haryana, Mahendergarh
Supervisor

Acknowledgement

I extend heartfelt gratitude to my supervisor, **Dr. Singara Singh**, for their unwavering support, steadfast guidance, and invaluable mentorship during the pre-dissertation phase. Their profound expertise and continuous encouragement have significantly shaped my research endeavors. I would also like to convey my appreciation to the Head of the Department (HOD), **Dr. Keshav Singh Rawat**, for providing me with the transformative opportunity to embark on this academic journey. Their unwavering belief in my capabilities has been a powerful and motivating force throughout this process. Additionally, I am grateful for the myriad of online resources that have expanded my knowledge, facilitating seamless access to a wealth of information. The collective support from these individuals and resources has played a pivotal role in the profound development of my pre-dissertation work, and I am sincerely thankful for their contributions to my academic journey.

(**Lakshmy Santhosh**)

Abstract

Terahertz rays, due to their harmless nature and ability to penetrate clothing, paper, plastic, and many other materials, make them suitable for detecting concealed objects. In the current scenario where there is still a lack of research and technology in detecting concealed objects, terahertz imaging turns out to be one of the best and non-harmful methods. Terahertz imaging has good potential in security surveillance cameras due to its ability to view concealed objects. Unlike other imaging techniques, like X-rays which emit radiation, terahertz imaging does not emit any harmful radiation. The challenge faced while using terahertz video for concealed object detection is the poor video quality and less accurate detection. If we accelerate the training speed, it will be a positive enhancement to the detection system considering the traditional object detection models which require significant training time. A comparative study is conducted using YOLOv5m and YOLOv8m, well-known object detection algorithms, to address challenges in concealed object detection using terahertz data. The study involved replacing the convolutional blocks(Conv) within the models with Depth-wise convolutions(DWConv) and Depth-wise separable convolutions(DWSConv). Remarkably, this modification resulted in significant improvements in training speed while maintaining a minimal decrease in accuracy.

Abbreviations

YOLO	You Only Look Once
PANet	Path Aggregation Network
AMMW	Active Millimeter Wave
RGB	Red, Green and Blue
M16	Military rifle model 16
CNN	Convolutional Neural Network
AK	Avtomat Kalashnikova(gun)
TT	Tula Tokarev (pistol)
F1	F-1 granade
RGD5	Ruchnaya Granata Distantsionnaya (Hand Granade Remote)
P	Precision
R	Recall
mAP	mean average precision
DW	Depth-wise
DWS	Depth-wise Separable

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the realm of modern security surveillance, the ability to accurately and efficiently detect objects in real-time video streams is important. Traditional methods often face challenges such as computational inefficiency, limited accuracy, and susceptibility to environmental factors. Consequently, there is a growing demand for advanced object detection models capable of addressing these limitations and providing reliable performance under diverse conditions.

One promising approach in the field of object detection is the utilization of You Only Look Once (YOLO) models, renowned for their speed and accuracy in real-time applications. Among these models, YOLOv5 and YOLOv8 stand out as models sharing comparable architectural traits. The YOLOv8 model's architecture is similar and a modified version of the YOLOv5 model. With an aim on gaining better results, Depth-wise and Depth-wise separable convolutions are integrated into these YOLOv5 and YOLOv8 frameworks.

Depth-wise convolution techniques have garnered significant attention in recent years for their ability to improve model efficiency without compromising accuracy [1]. By replacing traditional convolutional layers with Depth-wise convolution, we aim to enhance the feature extraction process within the YOLO framework, thereby potentially improving its performance in object detection tasks. Depth-wise separable convolution decomposes the convolution operation into Depth-wise and point-wise convolutions, which reduces computational complexity [2].

Our study focuses on comparing the performance of distinct models derived from YOLOv5m

and YOLOv8m architectures, each edited with Depth-wise convolution and Depth-wise separable convolution layers. Here, we utilize a terahertz video dataset converted into images, obtained from a dedicated website [3]. Terahertz imaging offers unique advantages over traditional optical methods, including enhanced penetration capabilities and reduced sensitivity to environmental factors such as lighting conditions and occlusions [4].

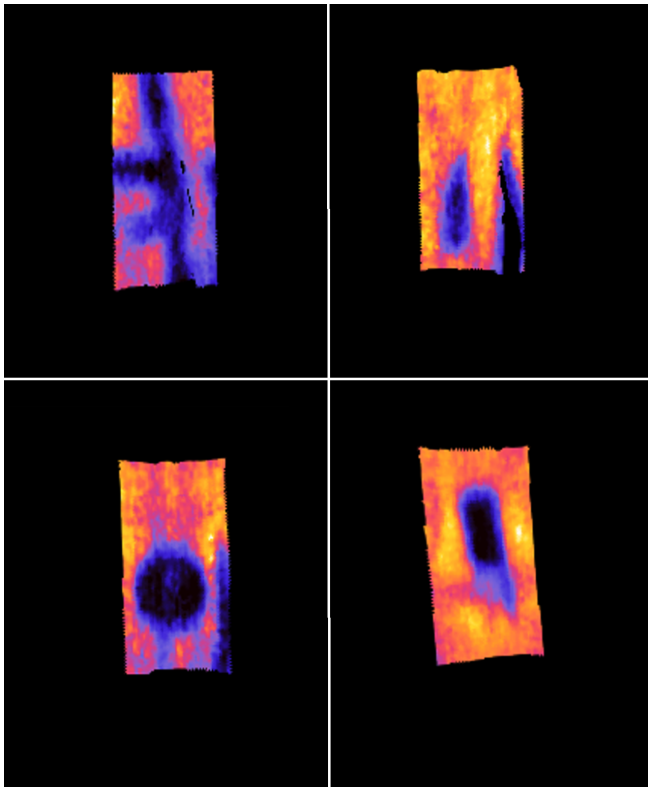

Figure 1.1: Terahertz images

Terahertz images of AK, bottle, saucepanLid, meatKnife hidden under clothing respectively. In the image, the blue and black part shows the hidden object whereas the red and yellow part indicates the body of the person. The difference in frequency of body and the objects in the image helps in detecting the images in terahertz imaging.

There are two types of terahertz imaging - Active terahertz imaging and passive terahertz imaging. Active terahertz imaging is done by capturing terahertz rays reflected by the object whereas passive terahertz imaging is done by capturing radiations of the object. Active terahertz imaging produces non-ionizing radiation and is considered to be safe for humans whereas X-rays produce radiations that cause many side effects and problems in humans. Here, the dataset used is created using passive terahertz imaging which does not produce any radiation and is therefore completely safe for human beings.

This research aims to contribute to the advancement of object detection techniques in security surveillance applications by using enhancements in YOLO algorithms with the help of Depth-wise convolutions and Depth-wise separable convolutions. The aim is to create a model that trains faster and gives better results as compared to the existing YOLO models on the terahertz dataset for security surveillance.

Chapter 2

Related Works

In today's world, ensuring public safety mandates the efficient detection of concealed hazardous objects, such as weapons and explosives. This task is particularly important in densely populated areas like airports, train stations, and public gatherings.

However, detecting these hidden items presents a formidable challenge due to their small size and varied shapes. Traditional object detection methods, including conventional image processing and computer vision techniques, exhibit limitations in terms of both accuracy and speed. Therefore, there is a critical necessity to devise methodologies that enhance detection accuracy and robustness for bolstering public safety. Even though several studies have been done on concealed object detection, there is still a lack of fast and accurate method for detecting concealed objects in public without emission of any harmful rays. This can be done if we utilize terahertz imaging for concealed object detection. In this study, passive terahertz imaging is utilized for detecting concealed objects beneath clothing.

Terahertz rays, situated between microwave and infrared radiation in the electromagnetic spectrum, exploits low levels of non-ionizing radiation to discern hidden objects within clothing and common packaging materials. Additionally, terahertz radiation has the capability to detect metal, plastic, paper and other substances, enabling the detection of plastic explosives and other materials that remain undetectable through X-ray or alternative radiation sources.

Even though researches are being done on this field, there is still a lack of security surveillance

system which can easily identify dangerous objects accurately and fast for ensuring public safety. As of now, in most of the like airports, metros, railway stations, etc., security surveillance is done in the entry and exit gates using X-ray machines for baggage, metal detectors for detecting metals in humans and physical pat downs in certain cases. These methods are efficient in detecting metals, but there are dangerous objects which can cause threat to public which are not made up of metal as well. These systems fail in detecting those objects. If better models using terahertz imaging in security surveillance is developed, it can decrease the chances of security breaches from happening and ensure public safety. Researches in this area and development of better models can make public safety and crowd monitoring process much easier thereby ensuring safety of people in public places.

2.0.1 Concealed Object Detection

In the realm of concealed object detection, Terahertz security screening cameras (TSSCs) offer a balance between low radiation exposure and efficient detection, albeit often providing only approximate object locations. Yang et al.[5] introduces a convolutional neural network (CNN) methodology for automatic object detection and recognition within THz security image sequences. Leveraging sparse and low-rank decomposition (SLD) for initial detection, this approach refines object locations through shape knowledge and morphological processing. Detailed recognition is facilitated through supervised training with the Faster R-CNN model, enhancing efficiency via a narrow-band approach. Extensive experimental validation underscores the high-performance outcomes in terms of accuracy and efficiency, suggesting notable advancements in terahertz security screening applications.

A concealed object detection model with self paced feature attention fusion network(SPFAFN) is proposed by Wang et al[6]. The features with different scales are fused in a top-down manner to integrate details and global semantics to detect small objects properly. During fusing multi-scale features, a hierarchical pyramid attention mechanism composed of channel and spatial attention is developed to perceive the object. Moreover, boosting self-paced learning is exploited to guide the model to learn hard samples that are difficultly detected. The work is done on an active millimeter wave(AMMW) dataset and a passive millimeter wave(PMMW) dataset.

In their another work,Wang et al.[7] introduces a normalized accumulation map-based training

mechanism for concealed object detection in millimeter-wave images. The proposed mechanism's results on millimeter-wave security image dataset demonstrate a 4.43% improvement in mean average precision when applying this approach to the YOLOv2 object detection network. This innovative training strategy holds promise for enhancing the accuracy of concealed object detection systems in millimeter-wave imaging for security applications.

Luo et al.[8] introduced a Cross-Feature Fusion Transformer YOLO (CFT-YOLO) for terahertz images, utilizing an active terahertz image dataset. M. Kowalski [9] conducted a comparison of concealed object detection, encompassing various types of clothing, using passive imagers in the terahertz range and mid-wavelength infrared range. The utilized algorithms included YOLOv3 and R-FCN (region-based fully convolutional networks). Pang et al.[10] presented a real-time detection method for identifying concealed metallic weapons on the human body using passive millimeter-wave (PMMW) imaging based on YOLOv3. Comparative analysis of YOLOv3-13, YOLOv3-53, and Single Shot Multibox Detector algorithm SSD-VGG16 revealed the superior real-time detection efficacy of the YOLOv3-53 model.

Danso et al.[11] has made hidden object detection model on terahrtz images. The dataset they used comprised of objects like knife, blade, screwdriver, etc. placed in bags and other packagings. The model they used is an improved version of YOLOv5 with BiFPN at the neck of YOLOv5 to improve low resolution. They also used transfer learning by fine-tuning the pre-training weight of backbone for migration learning.

Jayachitra et al.[12] proposed a YOLOv5 model integrated with a novel mutation-enabled swalp swarm algorithm (MESSA) for parameter optimization and model fine-tuning. The method purportedly yielded favorable results with high mean average precision at a threshold of 0.5.

Ge et al.[13] focused on preprocessing terahertz images before object detection. Two methods are used in their paper. Non-local mean(NLM) filtering and histogram equalization(HE). After preprocessing YOLOv7 algorithm is used for object detection. The method using non-local mean(NLM) filtering obtained highest accuracy.

In this paper[14], a linear array passive THz imaging system is used for efficient push-broom imaging, but the Thz images aquired are contaminated by severe striping and random noise. There-

fore, a dedicated deep learning network is devoloped for the detection of concealed objects from these THz images. It proposes three key advancements: integrating a bilateral filter into a CNN, designing a space-range grid for multi-scale filtering, and refining the YOLOv5 detector.

2.0.2 Depth-wise and Depth-wise Separable Convolution

These are two modified hence simpler versions of convolution layers, the number of parameters and complexity of a model can be reduced without much loss in accuracy by usig these layers instead of standard convolution layers. Depth-wise separable convolution can be used in variety of models in place of convolution[15]. Panigrahi et al. [16] in their work modified YOLOv2 using Depth-wise separable convolution module and inception depth wise modules. The proposed model is called DSM-IDM-YOLO. The proposed framework is computationally less expensive owing to its convolution design and a moderate number of layers. It aims to improve performance with minimal computational overhead. The work is done on pedestrian dataset.

Qin et al.[17] in their paper has put forward a method of combining classification model and detection model for fire detection. First Depth-wise separable convolution is used to classify fire images, then YOLOv3 target regression function is used to output the fire position information for the images whose classification result is fire. This avoids the problem that the accuracy of detection cannot be guaranteed by using YOLOv3 for target classification and position regression. Training was done on a network public dataset in which result of 98% detection accuracy and 38fps detection rate were obtained.

Tao Liu et al. [18] has proposed a sea surface object detection algorithm based on YOLOv4, by applying Reverse Depth-wise Separable Convolution(RDSC) to the backbone network and feature fusion network of YOLOv4. The number of weights of the algorithm decreased by 40%, detection speed increased more than 20%, and mAP also slightly increased in both the datasets used.

2.1 Limitations

Even though numerous researches have been done in the fields of concealed object detection using algorithms like YOLO, CNN and other models build on CNN, there are certain limitations.

- **Lack of training data :** As we all know, deep learning models using neural networks needs a lot of training data to work on in order to produce better results. The unavailability or limited availability of datasets like terahertz datasets, active millimeter wave datasets is one of the major limitations.
- **Complexity and Computational Demands:** many deep learning approaches have high computational complexity, limiting real-time detection in resource-constrained environments.
- **Data quality and enhancement :** Data in the form of terahertz or infrared images have low quality. Improving the quality of these images used in concealed object detection is challenging, with existing methods having limitations.
- **Challenges with Small Object Detection :** Deep learning struggles to accurately detect small concealed objects, affecting overall detection performance.
- **Difficulty in determining precise object location :** Achieving precise object localization, especially due to noise and clutter in images like terahertz, it is a critical challenge for detection algorithms to detect the precise location of objects.

2.2 Motivation

Concealed object detection is a major requirement in security surveillance as safety is a major concern in public places. Fast and accurate detection of concealed objects can help a lot in ensuring public safety especially in places like airports, railway stations and other crowded places. Terahertz(THz) wave due to its frequency range which lies between infrared and microwave(0.1-10 THz(1THz=10^{12}Hz)) has a unique ability to penetrate opaque objects such as clothing or packaging and detect objects of different frequency like metals, glass and even things like plastic,paper, etc.

Even though terahertz imaging can play a major role in ensuring public security, limited research has been done in this area.

YOLO(You Only Look Once) algorithms are currently showing state-of-the-art performance in object detection and image segmentation tasks. These algorithms are having a very good speed and accuracy as compared to traditional object detection models. Since speed and accuracy are very important in security object detection, algorithms like YOLO can be a good choice when it comes to security surveillance.

Due to the potential posed by terahertz imaging and the performance of YOLO algorithms, integrating them and trying to optimize the results by doing architectural modifications on the algorithm can ne a beneficial approach. Such integration efforts hold promise for advancing the capabilities of security surveillance systems, thereby bolstering public safety initiatives in critical locations. By harnessing the strengths of both terahertz imaging and YOLO algorithms, security systems can be enhance their ability to detect security threats effectively, ultimately creating safer environments for all.

2.3 Contribution

Using terahertz dataset for security surveillance has good potential. Terahertz imaging is one of the best method when it comes to concealed object detection due to its properties to detect concealed objects through clothing and packaging. Since YOLO algorithms perform well on images of different frequencies like terahertz, millimeter waves, etc., We use YOLO algorithms for creating a faster concealed object detection model. For this we used two verisons of YOLO (YOLOV5 and YOLOv8). Using the medium versions of both these algorithms and replacing convolutional layers by Depth-wise and Depth-wise separable convolutional layers, we are able to get good results in terms of mAP, precision and recall.

Chapter 3

Proposed Techniques

3.1 YOLO

YOLO, or ”You Only Look Once,” is a popular object detection and segmentation model known for its high speed and accuracy. As of now, there are 9 versions of YOLO released. The initial model was devoloped by Joseph Redmon and Ali Farhadi in 2015. YOLO quickly gained popularity due to its performance. YOLO enables rapid and accurate prediction of bounding boxes and class probabilities for multiple objects within an image in a single pass. This innovative methodology eliminates the need for complex region proposal networks, resulting in real-time inference and making YOLO well-suited for applications like surveillance systems and autonomous vehicles. In YOLO, matrices like Precision, Recall, mean average precision(mAP), etc are used for comparison of differnt models and model evaluation.

3.1.1 YOLOv5

YOLOv5 is a successor of previous versions of YOLO algorithms.It is build by making improvements on previous versions by introducing improvements in architecture and performance. The architecture of YOLOv5 is a single-stage object detection pipeline that can give real-time inference on various devices. The YOLOv5 has different version like s, m, l, etc. depending upon the speed and number of parameters. Here, we use the model YOLOV5m architecture.

The architecture of YOLOv5 contains backbone, neck and head sections.

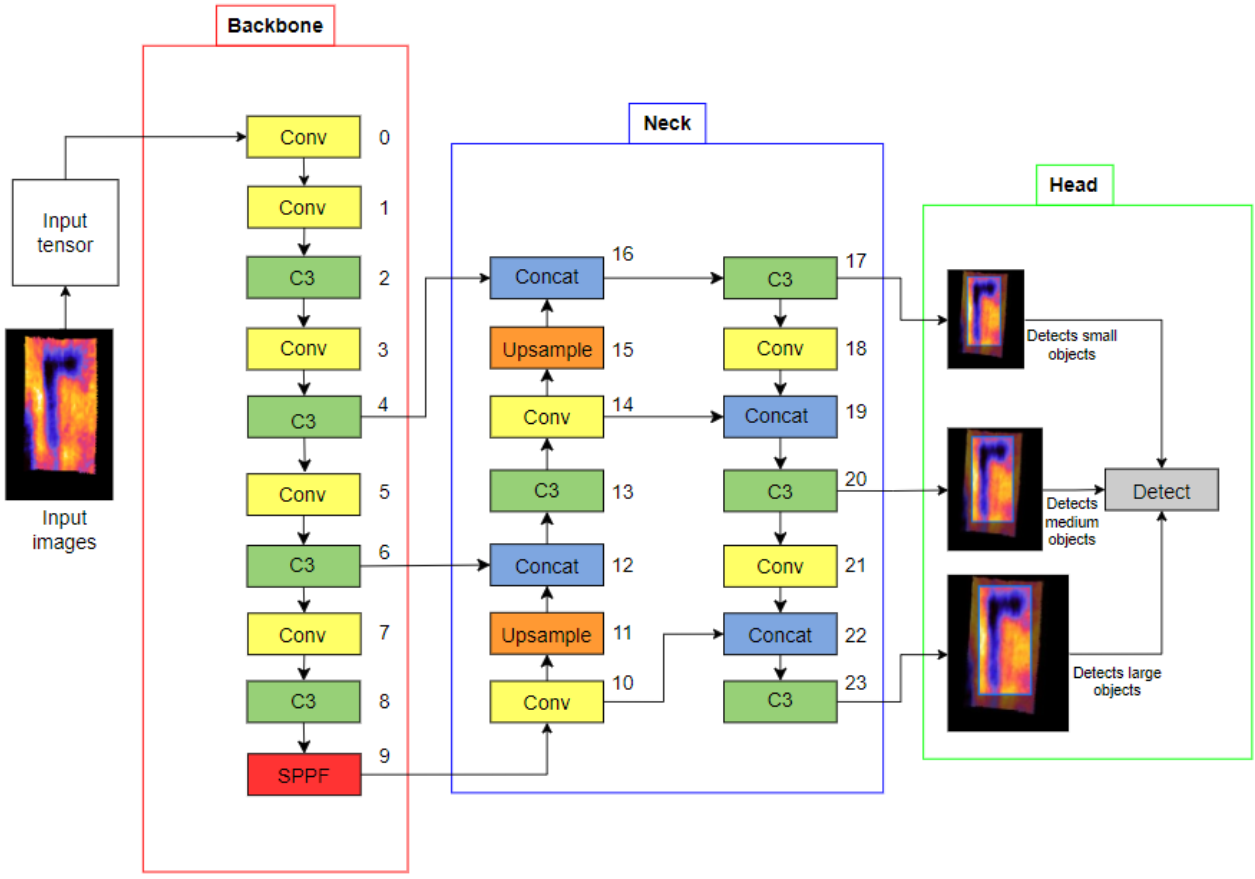

Figure 3.1: YOLOv5m Architecture

The diagram shows the architecture of YOLOv5m, the blocks present here are Conv, C3, Concat, Upsample and Detect. The detect block taking input from the C3 block(17) detects smaller objects, whereas the C3 block(20) detects medium sized objects and the C3 block(23) detects larger objects.

- The **backbone** is responsible for feature extraction. Image is converted into tensors and then sent to the backbone for feature extraction. The backbone is made up of CSPDarknet53. It has multiple CBS and C3 modules and finally one SPPF module. CSB(Conv2d+BatchNormalization+SiLU) layer is shown as 'Conv' in Fig. 3.1. The CBS module assists C3 Fig. 3.2 module for feature extraction and SPPF module Fig. 3.3 enhances the feature expression of the backbone.

- The **neck** is made up of Path Aggregation Network(PANet).PANet concatenates features from the backbone and PANet output. These combined features undergo convolution layers to reduce computational complexity. PANet includes convolutional layers, BatchNormalization, and SiLU activation, with upsampling and downsampling. It enhances accuracy by preserving

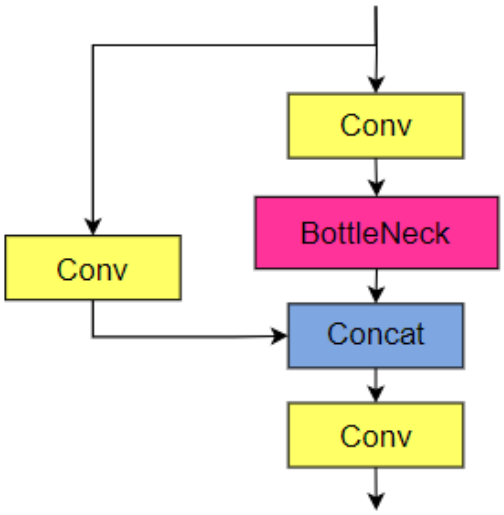

Figure 3.2: Structure of C3 module
The C3 module has Conv block, Bottleneck block and Concat block which concatenates the features. The number of bottleneck blocks are different in different C3 layers.

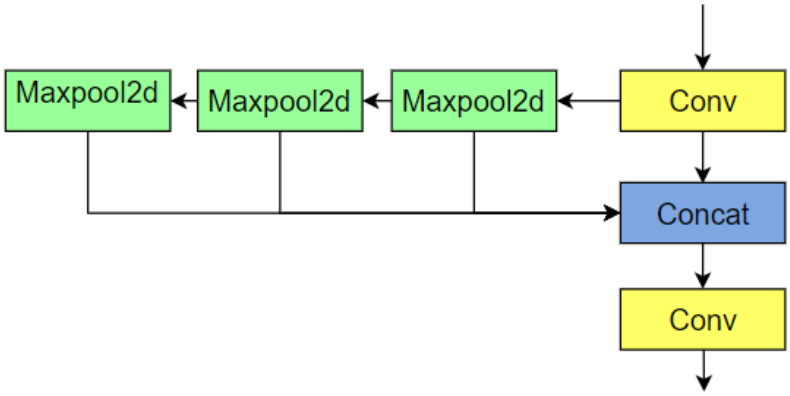

Figure 3.3: Structure of SPPF module
The Spatial Pyramid Pooling Fast (SPPF) layer enhances feature expression ability of the backbone. It has Maxpooling layers, Conv layers and Concat.

fine details and spatial structure through operations like bilinear interpolation and max pooling. This configuration contributes to improved accuracy in concealed object detection.

- The head part or the detection head of YOLO is same as that of YOLOv3 and YOLOv4. It consists of 3 convolutional layers that predicts the bounding box locations, the scores and object classes. The quations to compute target coordinates are:

$$b_x = (2 \cdot \sigma(t_x) - 0.5) + c_x \tag{3.1}$$

$$b_y = (2 \cdot \sigma(t_y) - 0.5) + c_y \tag{3.2}$$

$$b_w = p_w \cdot (2 \cdot \sigma(t_w))^2 \tag{3.3}$$

$$b_h = p_h \cdot (2 \cdot \sigma(t_h))^2 \tag{3.4}$$

3.1.2 YOLOv8

YOLOv8 introduced by ultralytics is build on previous existing YOLO models. It outperforms the previous versions by introducing modifications like spacial attention, feature fusion and context aggregation modules. These improvements result in faster and more accurate detection of objects making YOLOv8 superior to the older versions of YOLO. Some of the key features of YOLOv8 are:

- Imporved accuracy as compared to its predecessors.
- Faster inference speed.
- Multiple backbones like EfficientNet, ResNet and CSPDarknet, giving users the flexibility to choose the best one according to their needs.
- Adaptive training to optimize the learning rate and balance the loss function.

Table 3.1: YOLOv8 variants

Model Variant	d (depth_multiple)	w (width_multiple)	mc (max_channels)
n	0.33	0.25	1024
s	0.33	0.50	1024
m	0.67	0.75	768
l	1.00	1.00	512
xl	1.00	1.25	512

The architecture of YOLOv8 is build upon previous YOLO algorithms. The YOLOv8 algorithm mainly has n, s, m, l and xl variants. Each of these variants have different depth_multiple, width_multiple and max_channels. The depth_multiple determines how many bottlenecks are in C2F block, the width_multiple and max_channels parameters determine the output channel. The values are

given in the table 3.1 . Different variants have different speed and accuracy, the YOLOv8n model has less complexity and hence less number of parameters, so it has high speed and is less accurate compared to the other YOLOv8 models. The larger models YOLOv8l, YOLOv8xl, etc are more complex and therefore slower in training and requires more computational resources, but they give more accurate results.

The YOLOv8 model is made up of a convolutional neural network that can be divided into backbone, neck and head.

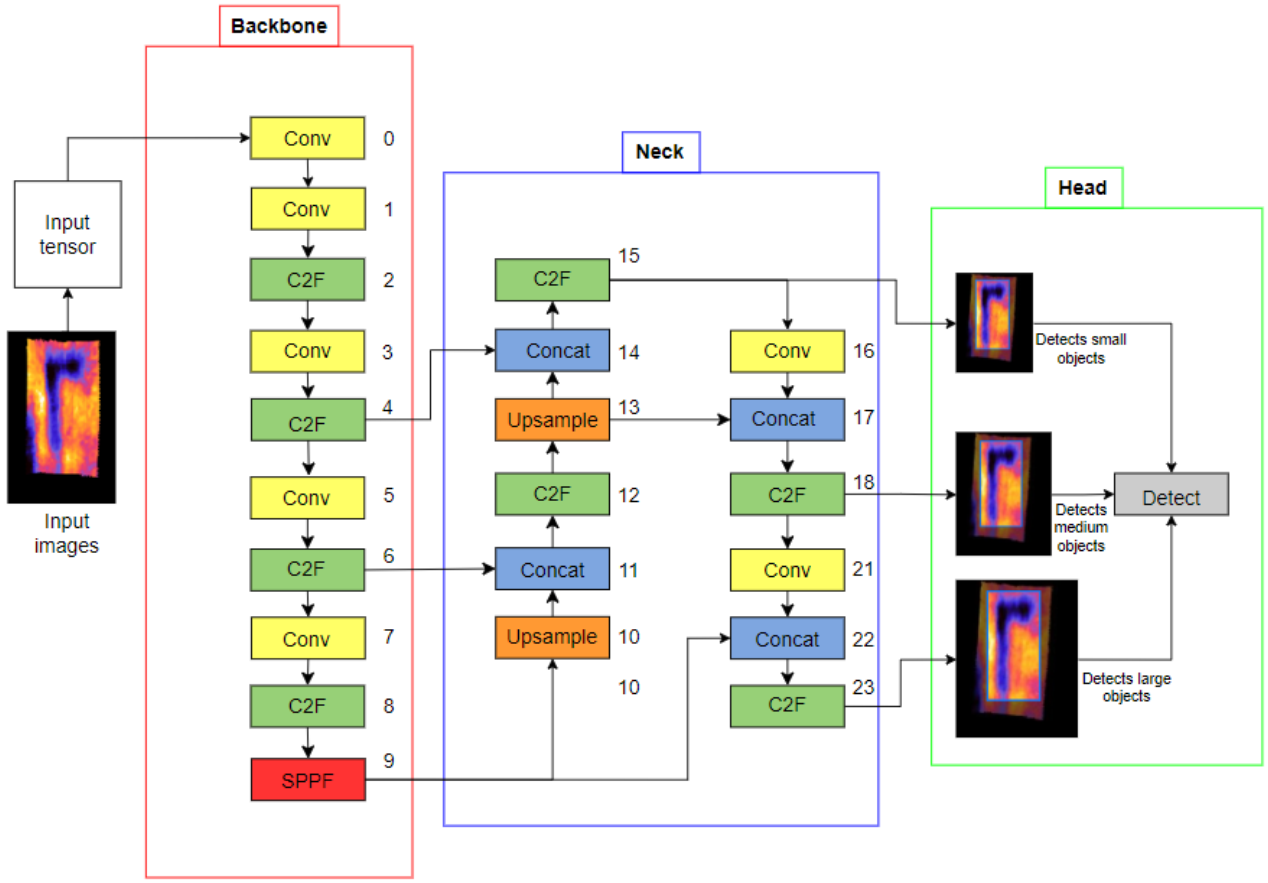

Figure 3.4: YOLOv8m architecture
In YOLOv8, we have Conv blocks, C2F blocks, Concat, SPPF, Upsample and Detect blocks. the detect block is same as that of YOLOv5 and detects objects of various sizes. The architecture shown here corresponds to the YOLOv8m model.

- The **backbone** of YOLOv8 does the feature extraction. The backbone is made up of several convolution layers that extract the features of various levels. The C2F block consists of multiple bottleneck layers depending upon the model variant(s,l,m,etc). The layers in C2F block is

shown in the Fig 3.5. The SPPF(Spacial Pyramid Pooling Fast) block has the same structure as that of the SPPF block in YOLOv5 architecture. It is present at the end of backbone. The main feature of SPPF is to generate representation of objects of various sizes in an image without resizing or causing spacial information loss in the image.

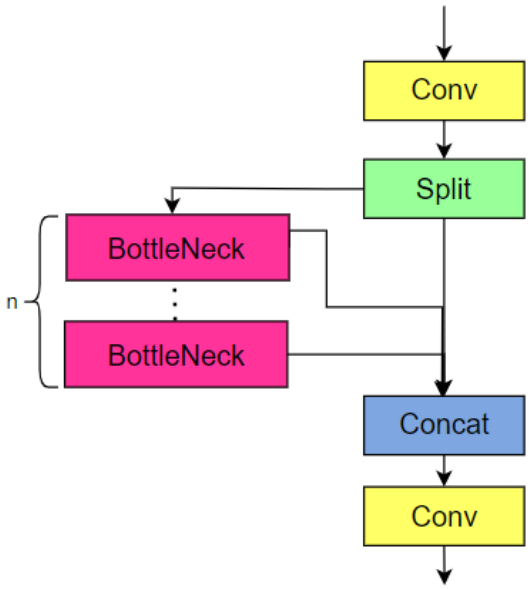

Figure 3.5: C2F block
The C2F block multiple bottleneck layers, Conv layer and Concat, the architecture of this block has some similarity with that of C3 block in YOLOv5. The Split splits the input feature map and the concat combines the output feature maps.

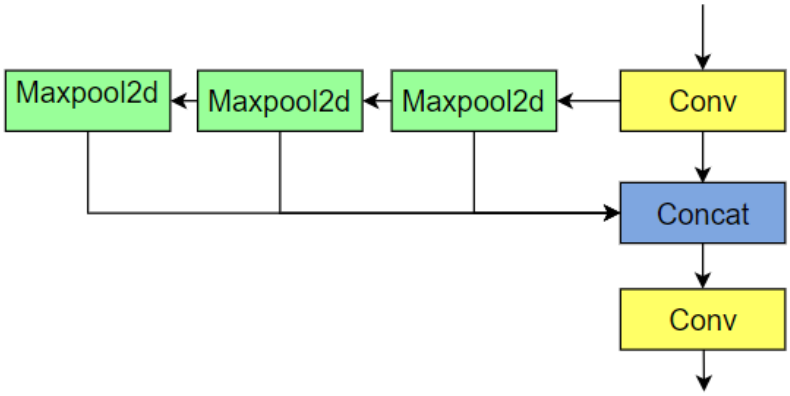

Figure 3.6: SPPF block
The SPPF block with multiple Maxpooling layers, Conv blocks and Concat.It enhances feature expression ability of the backbone.

- The **neck** of YOLOv8 combines the features aquired from various layers of the backbone

model. In the neck, the upsample layers which is at the starting is used to increase the feature map resolution of the previous layer to match with the feature map of the C2F block to which the output of upsample layer is concatenated. The 'Conv' blocks contain Conv2d, BatchNormalization and SiLU activation.

- The **head** predicts the classes and bounding box regions which is the final output of the object detection model. In this layer, detection is done in from 3 different sized feature maps to extract small, medium and large scale features from the image. The bounding box predictions are done in YOLOv8 in the same way as that of YOLOv5, the equations are:

$$b_x = (2 \cdot \sigma(t_x) - 0.5) + c_x \tag{3.5}$$

$$b_y = (2 \cdot \sigma(t_y) - 0.5) + c_y \tag{3.6}$$

$$b_w = p_w \cdot (2 \cdot \sigma(t_w))^2 \tag{3.7}$$

$$b_h = p_h \cdot (2 \cdot \sigma(t_h))^2 \tag{3.8}$$

3.2 Convolution

A convolution is a type of matrix operation, consisting of a kernel, a small matrix of weights, that slides over input data performing element-wise multiplication with the part of the input it is on, then summing the results into an output. The figure 3.7 shows an example of convolution operation with a single channel input ($6 \times 6 \times 1$) and a single kernel of 3×3. In the input image, the 6×6 indicates the height and width of the image and 1 indicates the number of channels. If we consider an RGB image, it will have 3 channels. So, and image of size 6×6 with 3 channels will be represented as $6 \times 6 \times 3$. In this case we consider that the number of kerals is also 1. So the keral size is considered as $6 \times 6 \times 1$. In the fig3.7, each element in the blue square of input image is multiplied with the respective elements in the filter and then added to get the first output element and so on.

If we consider an RGB image of size $10 \times 10 \times 3$ as input and a kernel of $3 \times 3 \times 3$. Then in the same method as above, in a 3D point of view we multiply all the respective elements of the first $3 \times 3 \times 3$ block of the input with the $3 \times 3 \times 3$ kernel and obtain the first output element and repeat

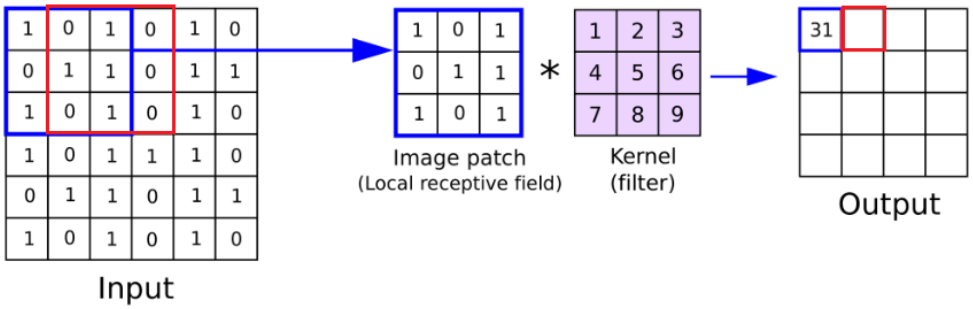

Figure 3.7: Basic Convolution operation
Element-wise multiplication is performed between the elements of the blue square from the input matrix and the elements of the filter respectively and then the results are added. This gives the first element of output matrix. The multiplication and addition processes are carried on until the output matrix is ready.

the same for all other elements.

Here, an input image of width and height 10 pixels and 3 channels(RGB), passed through a kernel of size $3 \times 3 \times 3$, we get an output of $8 \times 8 \times 1$, ie, image of width and height 8 pixels and 1 channel.

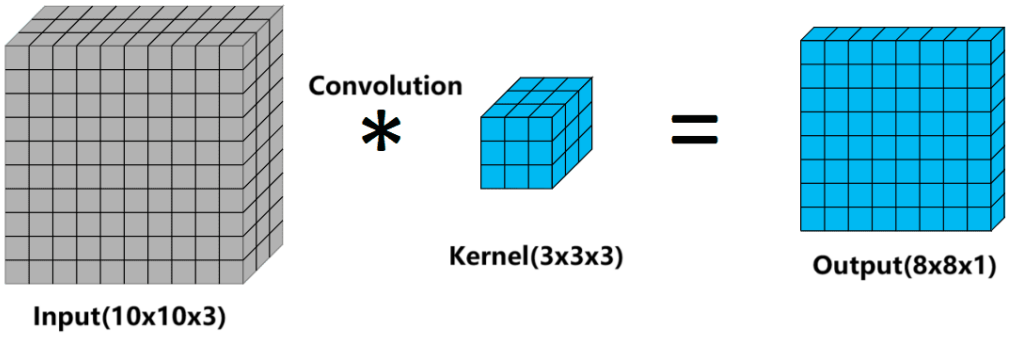

Figure 3.8: Convolution operation
The image represents the change in dimensions through convolution operation. The * indicates convolution operation. Here, the input image is of size 10x10 with 3 input channels(10x10x3) and the kernal is of size 3x3x3, the output we are getting here is 8x8x1. 10x10x3 input * 3x3x3 kernel = 8x8x1 output

Input Image size	**Kernel**	**Output image size**
$D_f \times D_f \times M$	N number of $D_k \times D_k \times M$	$D_p \times D_p \times N$

In general, if we consider an input image of of size $D_f \times D_f \times M$ and N kernels of size $D_k \times D_k \times M$, where D_f represents dimension of input image,M is the number of channels of the image(input

channels), D_k is the dimension of the kernel and N is the number of output channels, the dimesions of the output image is given by,

$$D_p = D_f - D_k + 1 \tag{3.9}$$

where D_p is the dimension of output image. The output image will be of size $D_p \times D_p \times N$

3.3 Depth-wise Separable Convolution

A Depth-wise separable convolution is made up of two type of convolution operations. They are:

- Depth-wise Convolution
- point-wise Convolution

The benefits of using Depth-wise separable convolutions are that they have lesser number of parameters to adjust as compared to the standard CNN's, which reduces overfitting. Also they are computationally cheaper as compared to the normal convolutional layers, which can help the model to be simpler and faster. The famous networks like mobilenet [2] and xception [1] used Depth-wise separable convolution.

3.3.1 Depth-wise Convolution

A Depth-wise convolution is a convolution along only one spatial dimension of the image whereas normal convolution is applied across all spatial dimensions of the image. Here since one convolutional filter is applied to each input channel, the output image will have same number of channels as that of input image.

Here we have a $10 \times 10 \times 3$ input image and 3 kernels of size $3 \times 3 \times 1$. These 3 kernels do convolution operation on the 3 input channels respectively. In the figure, the red coloured channel in input performs normal convolution with one of the kernel(represented by red color) and the result in output is also represented in red. In this method, the number of filters will be the same as the number of input channels and hence the output image will also have the same number of channels.

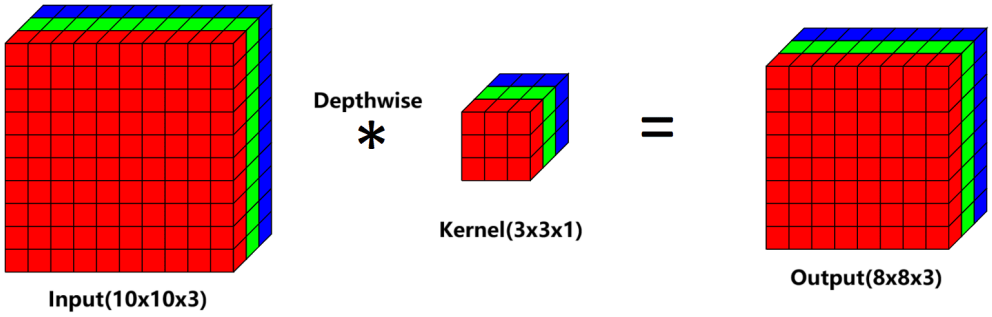

Figure 3.9: Depth-wise convolution operation
Here, the input image is of size 10x10x3 and kernel is of size 3x3x1. Since the number of input channels are 3, there are three 3x3x1 kernals. One kernel does the convolution operation with one input channel respectively and give one output channel. So, in the depthwise convolution the number of input and output channels are same. 10x10x3 input * 3x3x1 kernel = 8x8x3 output

Input Image size	**Kernel**	**Output image size**
$D_f \times D_f \times M$	M number of $D_k \times D_k \times 1$	$D_p \times D_p \times M$

Here an input image of $D_f \times D_f \times M$ after applying Depth-wise operation using M kernels of size $D_k \times D_k$ results in an output of size $D_p \times D_p \times M$. Each channel of the input layer is applied normal convolution operation using a kernel of size $D_k \times D_k \times 1$ and results in a channel of output of size $D_p \times D_p \times 1$. The value of D_p is,

$$D_p = D_f - D_k + 1 \tag{3.10}$$

where D_f is input dimension and D_k is dimension of filter.

3.3.2 point-wise Convolution

In a Depth-wise separable convolution, point-wise convolution is applied after Depth-wise convolution. A point-wise convolution has a $1 \times 1 \times M$ kernel. Where M is number of input channels. See figure 3.10. In the image, the output of Depth-wise convolution layer with a size $8 \times 8 \times 3$ and a kernel of size 1×1 is giving an output of $8 \times 8 \times 1$.

Here since the output of the Depth-wise layer is taken as input of the point-wise convolution, the input image will have a size $D_p \times D_p \times M$. Kernels of size $1 \times 1 \times M$ is used to multiply with each

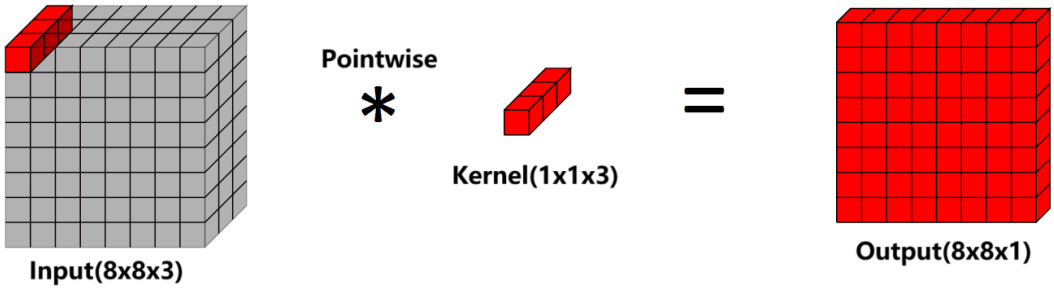

Figure 3.10: point-wise convolution operation
Here the input image has a size 8x8x3 since it is the output of the depthwise operation. The kernal of size 1x1x3 does convolution operation with each of the 1x1x3 blocks of the input and results in a 8x8x1 output. 8x8x3 input * 1x1x3 kernel = 8x8x1 output

Input Image size	**Kernel size**	**Output image size**
$D_p \times D_p \times M$	$1 \times 1 \times M$	$D_p \times D_p \times 1$

element in each of the $1 \times 1 \times M$ block of the input as shown in fig 3.10 to get the elements of the output respectively. Here, the output image will have same size as that of input image of point-wise convolution and 1 channel.

A Depth-wise separable convolution has lesser number of parameters and lesser number of multiplication and addition operation as compared to normal convolution. So Depth-wise separable convolution runs faster than normal convolution. There might be a slight decrease in accuracy in Depth-wise separable convolution.

The number of multiplication operations in normal convolution layer and Depth-wise separable convolution layer are given in fig 3.2.

Table 3.2: Number of Multiplications in different convolutions

	No. of multiplications in 1 conv operation	Total number of multiplication
Normal Convolution	$D_k^2 \times M$	$N \times D_p^2 \times D_k^2 \times M$
Depth-wise Convolution	D_k^2	$N \times D_p^2 \times D_k^2$
point-wise Convolution	M	$N \times D_p^2 \times M$
Depth-wise Separable (Depth-wise + point-wise) Convolution	$D_k^2 + M$	$N \times D_p^2 \times (D_k^2 + M)$

3.4 Models

3.4.1 DW YOLOv5

Depth-wise YOLOv5 is made by replacing the Conv block in the model with DWConv. The DWConv consists of a Conv2d layer with groups as the greatest common divisor of input channels and output channels of the layer, BatchNormalization and SiLU activation function. The modified YOLOv5m architecture is given in Fig 3.11

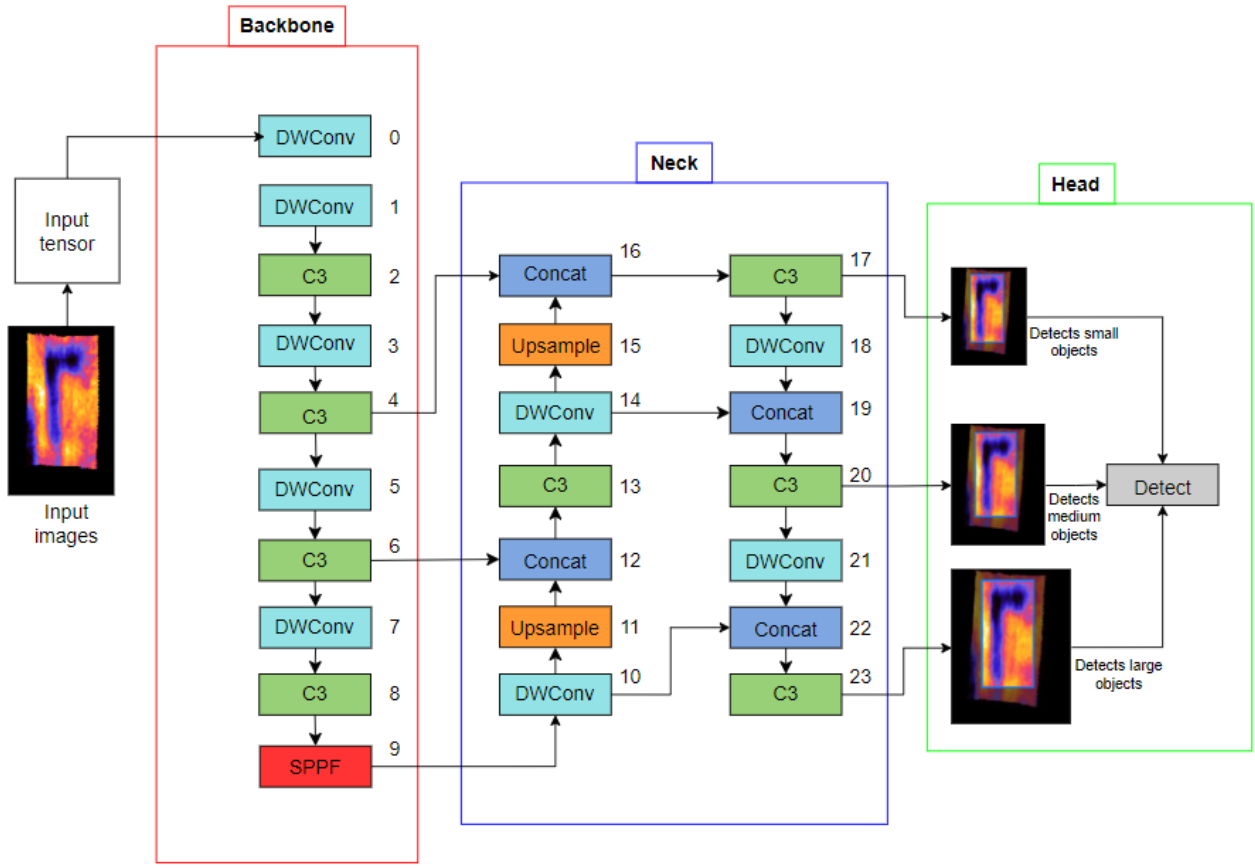

Figure 3.11: DW YOLOv5m Architecture

In this modified YOLOv5m architecture, the Conv blocks are replaced by DWConv blocks which is a slightly modified version of Conv block with the group parameter of the Conv2d in Conv is replaced by greatest common divisor of input channels and output channels(number of filters)

3.4.2 DWS YOLOv5

Depth-wise Separable YOLOv5 is a custom layer made by introducing Depth-wise and point-wise convolutions in place of Conv2d in Conv block. The new block contains a Depth-wise convolution layer, point-wise convolution layer, BatchNormalization and SiLU activation function. The modified YOLOv5m architecture is given in Fig 3.12

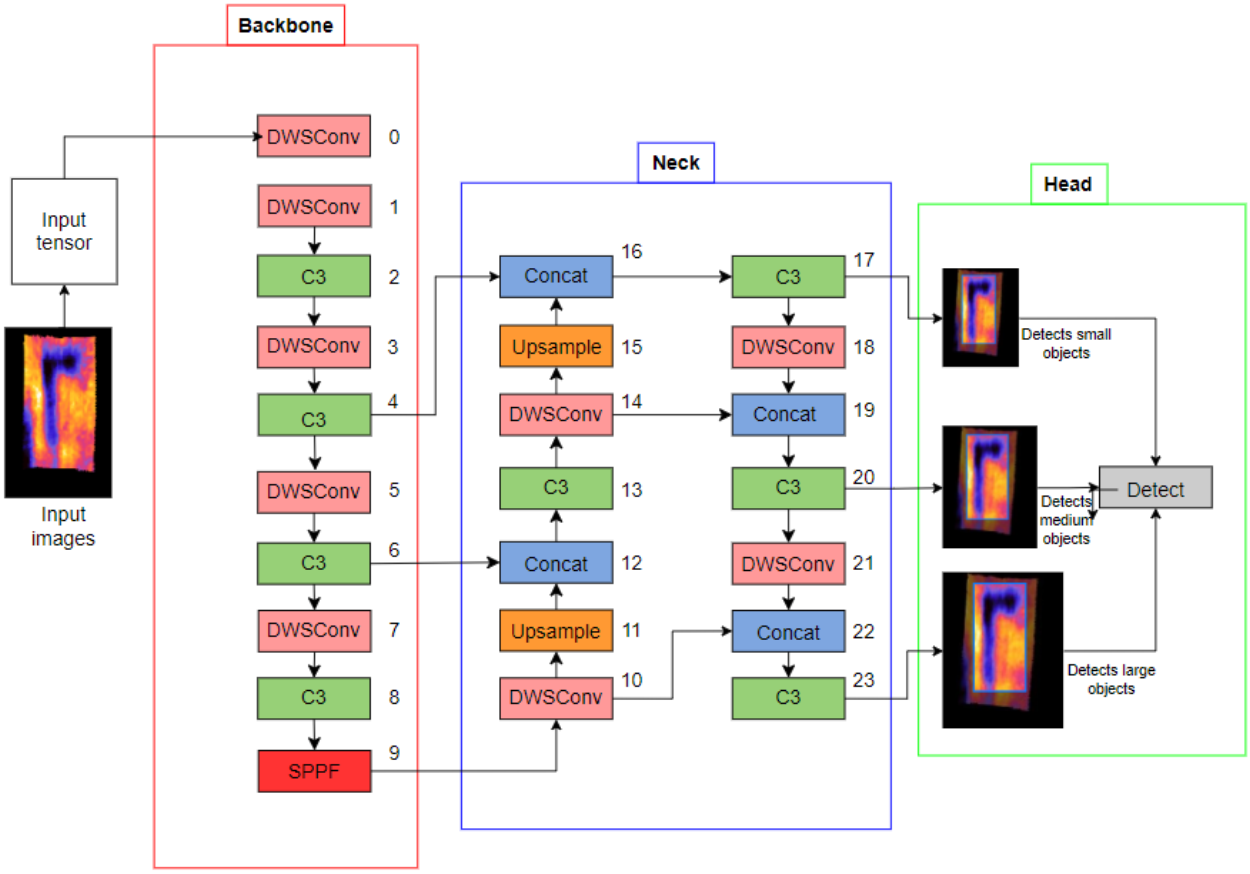

Figure 3.12: DWS YOLOv5m Architecture
In the DWS YOLOv5m, the Conv blocks in the architecture is replaced by DWS convolutional layers which is a combination of Depth-wise and point-wise convolution layers.

3.4.3 DW YOLOv8

In Depth-wise YOLOv8 we replace the Conv block in the model with DWConv. Similar to the Conv block, DWConv block also has a Conv2d layer, BatchNormalization and SiLU activation function, but in the DWConv, the Conv2d layer with groups as the greatest common divisor of input channels and

output channels replaced the normal Conv layer. This helps in decreasing the number of parameters of the model, hence making the model run faster. The modified YOLOv8m architecture is given in Fig 3.13

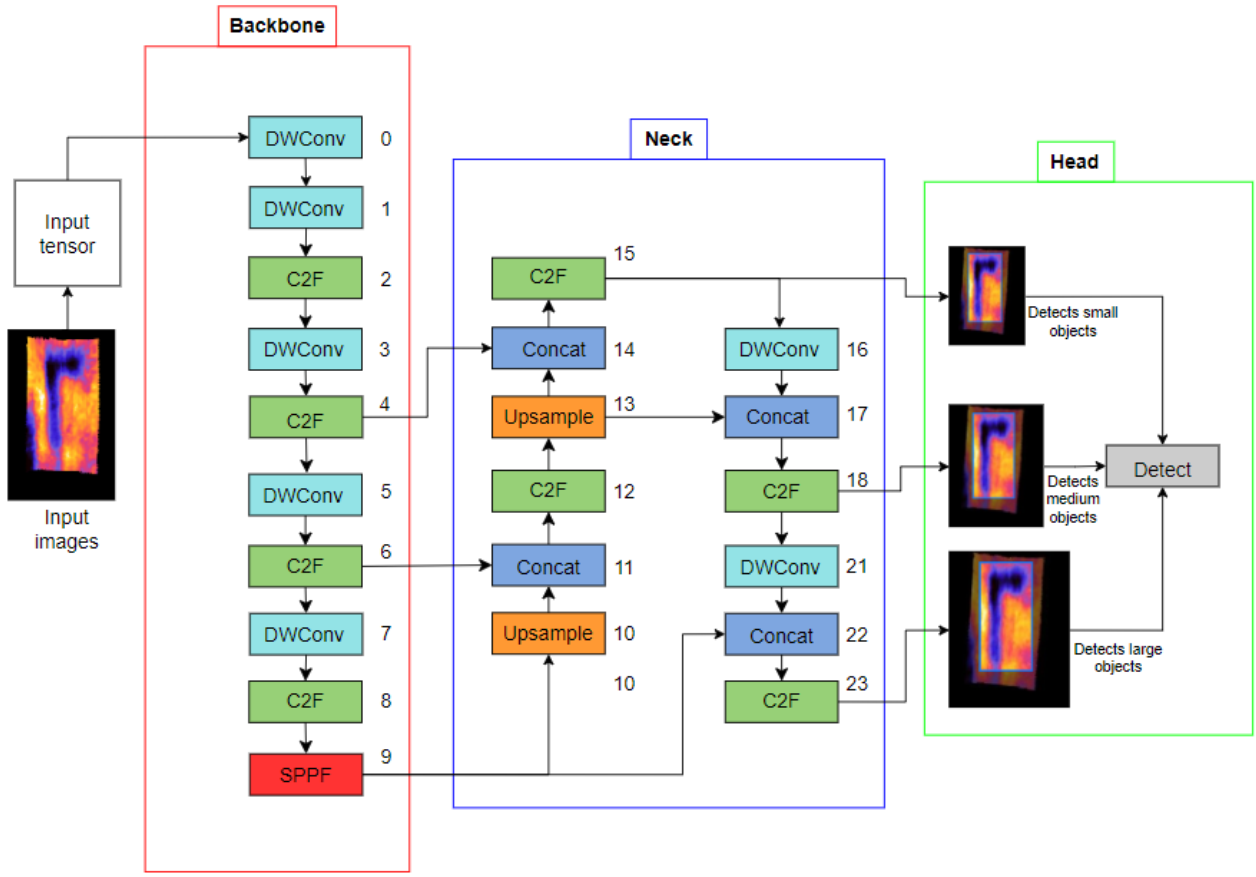

Figure 3.13: DW YOLOv8m Architecture
In this modified YOLOv8m architecture, the Conv blocks are replaced by DWConv blocks which is a slightly modified version of Conv block with the group parameter of the Conv2d in Conv is replaced by greatest common divisor of input channels and output channels(number of filters)

3.4.4 DWS YOLOv5

The Depth-wise Separable YOLOv5 is a custom block created by introducing Depth-wise and point-wise convolutions in place of Conv2d in Conv block. The DWSConv block contains a Depth-wise separable convolution layer(Depth-wise convolution layer + point-wise convolution layer), BatchNormalization and SiLU activation function. The modified YOLOv5m architecture is given in Fig 3.14

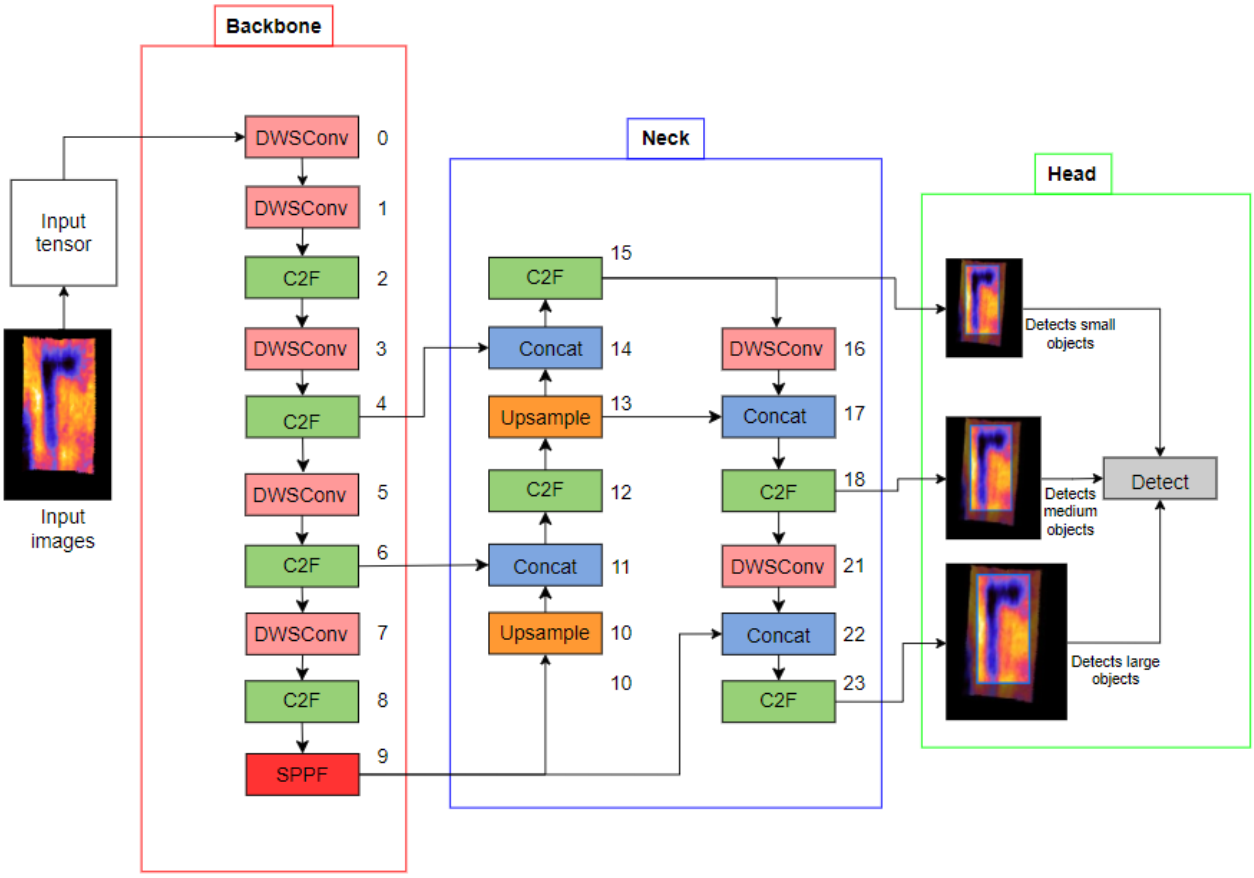

Figure 3.14: DWS YOLOv8m Architecture
In the DWS YOLOv8m, the Conv blocks in the architecture is replaced by DWS convolutional layers which is a combination of Depth-wise and point-wise convolution layers.

Chapter 4

Experimental Results and Analysis

4.1 Dataset Description And Preprocessing

A terahertz video dataset is used for concealed object detection and comparison of results. The dataset is taken from the publicly available terahertz video dataset[3]. The dataset construction involved the utilization of Multimedia_04_3D_normalizer, used in extracting terahertz image frames from the diverse set of 32 videos. The videos included images of objects like pistol, grenade, knife, M16, etc. By consolidating videos depicting the same object in various anatomical locations into a unified class to ensure a coherent representation, the number of classes in the dataset reduced to 22 classes. These images are then annotated and prepared to meet the requirements for YOLO models in Roboflow[19] and then data augmentation is also done in the same platform.

Roboflow provides many datasets on its own, and also it acts as a platform to annotate images by clicking and dragging and drawing bounding boxes. It supports various dataset formats, after annotating our images, data augmentations, resizing, etc. can be done and the datset can be exported in various different formats like COCO JSON format, YOLOv5 Pytorch format, Pascal VOC XML format, etc.

Our dataset comprises a total of 7200 images comprising of 22 different classes. These classes include dangerous objects like knife, pistol, hand grenade which possess threat when carried in public and other non-dangerous objects like A4paper and USBDisk. The dataset also includes images with

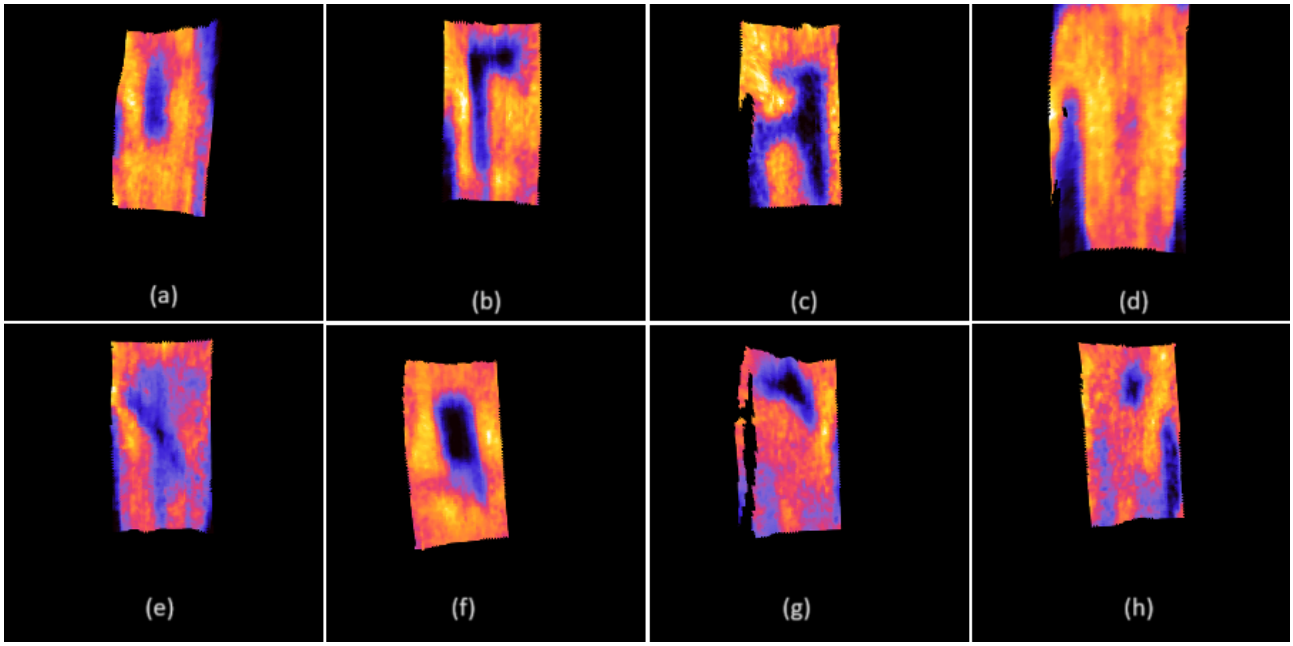

Figure 4.1: Dataset sample
(a) bottle **(b)** axe **(c)** AK **(d)** NULL(no object)
(e) knife **(f)**meatKnife **(g)**pistol **(h)**cigaretteBox

no object present. Since the terahertz images are extracted from terahertz videos, most of the images contain object in the same position, but captured from slightly different angles. The class details is given in table 4.1.

The 7200 images in the dataset is divided into the training set containing 5594 images, the validation set containing 794 images, and the test set containing 812 images. After augmentations the training set has 11188 images. The dataset was prepared by resizing to 640x640 and applying data augumentations on a dataset containing 12794 total images

Table 4.2: Data Augumentations

Transformation	**Range**
Flip	Horizontal and Vertical
Rotation	Between $-45°$ and $+45°$
Brightness	Between -15% and $+15\%$

Table 4.1: Object Classes in the Terahertz Video Dataset

Class	Object Type	Description
1	A4paper	paper
2	AK	gun
3	AK_noMagazine	gun
4	M16	gun
5	Tin	container
6	axe	-
7	beltholster	pistol
8	bottle	container
9	candyboxLid	-
10	cigaretteBox	-
11	fomka	-
12	glassjar	container
13	hammerAndSickle	-
14	handGranade	granade
15	knife	-
16	meatKnife	-
17	phoneNokia	phone
18	phoneXiaomi	phone
19	pistol	pistol
20	saucepanLid	-
21	shoulderholster	pistol
22	usbDisk	-

4.2 Performance Metrices

Various performance matrices used for the YOLO model evalutaion are:

- **Precision(P) :** The precision of a model is defined as the number of true positives divided by the number of true positives plus false positives.It measures the accuracy of positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.1}$$

  where TP is True Positive abd FP is False Positive.

- **Recall(R) :** The recall measures the ability of a model to correctly identify all relevant instances of a class. It quantifies the proportion of true positive predictions out of all actual positive instances in the dataset, indicating how well the model captures the positive cases without missing any.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.2}$$

where TP is True Positive and FN is False Negative.

- **Mean Average Precision(mAP):** The mAP metric takes into account both the precision and recall of the model and is calculated as the mean of the average precision for each class. A higher mAP value indicates a better performance of the model.

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{4.3}$$

where N is the number of classes, AP_i ia the average precision of class i and i ranges from 1 to N. YOLO evaluation metrics include mAP50 and mAP50-95. mAP50 measures mean average precision at an intersection of union(IoU)threshold of 0.5 whereas map50-95 measures the mean average precision across IoU thresholds ranging from 0.5 to 0.95.

4.3 Results

4.3.1 YOLOv5 and its modifications

The YOLOv5 models we trained our terahrtz dataset on are YOLOv5m, DW YOLOv5 and DWS YOLOv5m After tarining 25 epochs across all models, the DWS YOLOv5 exhibited the highest mAP50-95 score, achieving 72.7%, followed closely by DW YOLOv5 with a score of 72.3%. In comparison, the original YOLOv5m model attained an mAP50-95 of 72%. Additionally, both DW YOLOv5 and DWS YOLOv5 showcased a precision that surpassed the normal YOLOv5 by 0.1%.

Considering the runtime, DW YOLOv5 completed the 25 epochs in 1.579 hours, while DWS YOLOv5 finished in 1.738 hours. In contrast, the normal YOLOv5 model required 1.871 hours for the same number of epochs. This translates to DW YOLOv5 and DWS YOLOv5 running 0.292 hours and 0.133 hours faster, respectively, than the original YOLOv5 model. There can be difference in training time due to the fluctuations in speed of the internet.

In summary, the experiments reveal that both DW YOLOv5 and DWS YOLOv5 outperform the traditional YOLOv5 model in terms of mAP50-95 score and precision while also demonstrating im-

Table 4.3: Results of YOLOv5 models model training done on terahertz dataset

Model	Time taken	Precision	Recall	maP50	mAP50-95	layers	parameters
YOLOv5m	1.871	0.992	1	0.995	0.72	291	20956179
DW YOLOv5m	1.579	0.993	0.999	0.995	0.723	291	15419379
DWS YOLOv5m	1.738	0.993	0.999	0.995	0.727	301	16356063

proved efficiency in terms of training time. The table 4.4 shows detailed results.

4.3.2 YOLOv8 and its modifications

We have trained our dataset on three different YOLOv8m models. The original YOLOv8m provided by ultralytics, DW YOLOv8m and DWS YOLOv8m two modified versions of YOLOV8m. After training each of the models for 30 epochs, the recall value of DWS YOLOv8m is 0.001% greater than the recall of YOLOV8m and DW YOLOv8m. Other matrices remains same, but the DW YOLOv8m and DWS YOLOv8m runs faster than the normal YOLOv8m. The layers and number of parameters for model training is the least for DW YOLOv8m model and then the DWS YOLOv8m and the highest for normal YOLOv8m.

Table 4.4: Results of YOLOv8 models training done on terahertz dataset

Model	Time taken	Precision	Recall	maP50	mAP50-95	layers	parameters
YOLOv8m	1.426	0.994	0.999	0.995	0.766	295	25869058
DW YOLOv8m	1.393	0.994	0.999	0.995	0.766	295	21369346
DWS YOLOv8m	1.427	0.994	1	0.995	0.765	303	21861421

Differently from YOLOV5m model, here we can see that the training time do not have much difference. But the metrices like precision and recall of DW YOLOv8m and DWS YOLOv8m has increased, but the mAP50-95 of the models are less than that of the original model. The number of parameters of the model has decreased a lot as compared to the original model and is supposed to make training faster, the training time may vary under different internet speed and processing units.

Chapter 5

Conclusions and Future Scope

5.1 Conculsions

Using the YOLOv5 and YOLOv8 models and modifying them with Depth-wise and Depth-wise separable convolutions have given better results compared to the original models. Precision, Recall and mAP has increased in most of the cases while in some cases, some of the matrices remain same. By using these modified versions of YOLO, we are able to see a notable decrease in training parameters which in turn can speed up the model training process. So, we can say that the lighter versions of YOLO, devoloped using Depth-wise Convolution and Depth-wise Separable Convolution gives slightly better results than that of the original models and run faster.

5.2 Future Scope

The proposed models that we developed here have certain limitations like:

- Limited data : The data source we utilized for this project[3] is made up video files containing different objects. Most of the object categories in the dataset has only 1 video, that is all the images in a certain class are images in which the object is at the same position but taken from slightly different angles. This can possess a problem of overfitting or the model not performing well on diverse datasets.

- Dataset quality : Since the passive terahertz images are captured from the radiation emitted by the subject, the clarity of the images are less as compared to other type of images, this makes it difficult for the algorithms to perform well on these datasets.

- Computational resources : Since models like YOLOv5 and YOLOv8 need good computational resources to run faster, training these models for a large number of epochs or for a large batch size is time consuming and not possible in certain systems.

If we can overcome these limitations and develop a proper terahertz dataset with diverse images and classes with a good number of images. It will really enhance the accuracy of concealed object detection models which depend on the terahertz dataset. If we are able to increase the image quality of terahertz images using some method and then incorporate these methods with the currently existing methods for terahertz detection, it can result in very good results. Also, using newer algorithms like YOLOv9, YOLO NAS, etc, might improve the model performance as compared to the models we used in our research.

References

[1] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[2] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[3] Alexei A Morozov and Olga S Sushkova. Development of a publicly available terahertz video dataset and a software platform for experimenting with the intelligent terahertz visual surveillance. In *Proceedings of International Conference on Frontiers in Computing and Systems: COMSYS 2020*, pages 105–113. Springer, 2021.

[4] Michael C Kemp, PF Taday, Bryan E Cole, JA Cluff, Anthony J Fitzgerald, and William R Tribe. Security applications of terahertz technology. In *Terahertz for military and security applications*, volume 5070, pages 44–52. SPIE, 2003.

[5] Xi Yang, Tan Wu, Lei Zhang, Dong Yang, Nannan Wang, Bin Song, and Xinbo Gao. Cnn with spatio-temporal information for fast suspicious object detection and recognition in thz security images. *Signal Processing*, 160:202–214, 2019.

[6] Xinlin Wang, Shuiping Gou, Jichao Li, Yinghai Zhao, Zhen Liu, Changzhe Jiao, and Shasha Mao. Self-paced feature attention fusion network for concealed object detection in millimeter-wave image. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1):224–239, 2021.

[7] Chen Wang, Jun Shi, Zenan Zhou, Liang Li, Yuanyuan Zhou, and Xiaqing Yang. Concealed object detection for millimeter-wave images with normalized accumulation map. *IEEE Sensors Journal*, 21(5):6468–6475, 2020.

[8] Shaojuan Luo, Heng Wu, Meiyun Chen, Genping Zhao, Chunhua He, et al. Concealed hazardous object detection for terahertz images with cross-feature fusion transformer.

[9] M Kowalski. Hidden object detection and recognition in passive terahertz and mid-wavelength infrared. *Journal of Infrared, Millimeter, and Terahertz Waves*, 40(11):1074–1091, 2019.

[10] Lei Pang, Hui Liu, Yang Chen, and Jungang Miao. Real-time concealed object detection from passive millimeter wave images based on the yolov3 algorithm. *Sensors*, 20(6):1678, 2020.

[11] Samuel Akwasi Danso, Liping Shang, Deng Hu, Justice Odoom, Quancheng Liu, and Benedicta Nana Esi Nyarko. Hidden dangerous object recognition in terahertz images using deep learning methods. *Applied Sciences*, 12(15):7354, 2022.

[12] J Jayachitra, K Suganya Devi, SV Manisekaran, and Satish Kumar Satti. Terahertz video-based hidden object detection using yolov5m and mutation-enabled salp swarm algorithm for enhanced accuracy and faster recognition. *The Journal of Supercomputing*, pages 1–26, 2023.

[13] Zihao Ge, Yuan Zhang, Xuyang Wu, Zhiyuan Jia, Heng Wang, and Keke Jia. Deep-learning-based method for concealed object detection in terahertz (thz) images. In *Advanced Fiber Laser Conference (AFL2023)*, volume 13104, pages 268–274. SPIE, 2024.

[14] Fan Xu, Xuyang Huang, Qihui Wu, Xiaofei Zhang, Zhigao Shang, and Yijia Zhang. Yolo-msfg: toward real-time detection of concealed objects in passive terahertz images. *IEEE Sensors Journal*, 22(1):520–534, 2021.

[15] Tonghao Wang, Shijiao Gao, Yukang Huo, Piercarlo Cattani, and Shuli Mei. Depthwise separable axial asymmetric wavelet convolutional neural networks. *Available at SSRN 4760183*.

[16] Sweta Panigrahi and USN Raju. Dsm-idm-yolo: Depth-wise separable module and inception

depth-wise module based yolo for pedestrian detection. *International Journal on Artificial Intelligence Tools*, 32(04):2350011, 2023.

[17] Yue-Yan Qin, Jiang-Tao Cao, and Xiao-Fei Ji. Fire detection method based on depthwise separable convolution and yolov3. *International Journal of Automation and Computing*, 18(2):300–310, 2021.

[18] Tao Liu, Bo Pang, Lei Zhang, Wei Yang, and Xiaoqiang Sun. Sea surface object detection algorithm based on yolo v4 fused with reverse depthwise separable convolution (rdsc) for usv. *Journal of Marine Science and Engineering*, 9(7):753, 2021.

[19] Brad Dwyer and Joseph Nelson. Roboflow. Roboflow (Version 1.0) [Software], 2022. Computer Vision.