Centre for Artificial Intelligence and Robotics (CAIR)

Defence Research Development Organization (DRDO), Bengaluru

Duration of Summer Internship : 24th July- 8th Dec, 2023

Guided by: Mr. Biswajit Sharma, Scientist "F"

Machine Learning for SAR Satellite Images

Internship Report to be Submitted to

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & ML

HINDUTHAN COLLEGE OF ENGINEERING AND TECHNOLOGY

Submitted by: SANJAY.R

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING, 2020-24

Student ID(CAIR): 1148

Reg No: 20115046

Acknowledgement

I want to express my sincere and profound gratitude to the Director, CAIR, DRDO, Bengaluru for allowing me to undergo the project internship of 9 weeks. In regard to the internship, I bear immense pleasure in expressing my special gratitude to Mrs. Biswajit Sharma, scientist ”F” of the Intelligent Systems and Robotics Division (ISRD) as my project guide and thanks to all the employees of CAIR for their constant guidance and suggestions throughout the internship duration.

Abstract

Computer Vision plays a crucial role in perceiving and understanding the environment using various sensors and conventional and non-conventional methods and algorithms. Deep Learning algorithms have proven to be highly effective in this domain, and among them, YOLO (You Look Only Once) stands out as a prominent architecture capable of real-time object detection with superior speed compared to other methods like Faster RCNN, RCNN, and SSD. Machine learning (ML) is a powerful tool that can be used to analyse and interpret SAR images. SAR images are complex and can be difficult to interpret, but ML can be used to extract meaningful information from them. Object detection in SAR images is a challenging task due to the inherent complexity of SAR data. SAR images are affected by speckle noise, which is a granular pattern caused by the coherent nature of SAR signals. Speckle noise can obscure the underlying features of objects in SAR images, making it difficult to detect them.

Contents

Introduction

Computer vision is a field of study that focuses on enabling computers to interpret and understand visual information from the world. It involves the development of algorithms and models that allow machines to process, analyse, and make sense of images and videos. We can categorize the computer vision tasks into five broad categories. They are:

1. **Image Classification:** Image classification is the task of assigning a label or category to an input image. It involves training a model to recognize and categorize images into predefined classes. Through the process of supervised learning, the model learns to identify patterns and features within images to make accurate predictions about their content. Image classification has wide-ranging applications, from identifying objects in photographs to classifying medical images for diagnosis.

2. **Object Detection and Localization:** Object detection is a more complex task that involves not only identifying objects in an image but also precisely localizing them. This is achieved by drawing bounding boxes around the detected objects. Object detection is particularly useful in scenarios where multiple objects need to be identified within an image or video frame. It finds applications in autonomous vehicles, surveillance systems, and object tracking tasks.

3. **Image Segmentation:** Image segmentation aims to divide an image into regions or segments to identify the boundaries of objects. Unlike object detection, image segmentation provides a pixel-level mask that classifies each pixel into different object categories. This fine-grained understanding

of object boundaries helps in various applications such as medical image analysis, where segmenting organs or tumours is essential.

4. **Image Generation and Synthesis:** Image generation tasks involve using generative models like Generative Adversarial Networks (GANs) or Variational Autoencoders (VAEs) to create new images based on learned patterns and styles. These models can generate realistic-looking images that resemble the training data. Image generation has applications in art generation, image-to-image translation, and data augmentation for training computer vision models.

5. **Image Understanding and Scene Analysis:** Image understanding goes beyond recognizing individual objects and focuses on analysing the overall content and context of a scene. This category involves understanding the relationships between objects, their spatial layout, and the interactions between different elements. Scene analysis tasks include recognizing activities, scenes, and visual narratives from images and videos.

These five categories represent a comprehensive set of computer vision tasks, each contributing to the development of intelligent systems that can perceive and interpret visual information. The advancements in machine learning and deep learning have significantly improved the performance of computer vision models, opening up exciting possibilities for real-world applications in diverse domains.

Machine Learning in Computer Vision: Machine learning algorithms play a fundamental role in computer vision. They are trained on large datasets, where images and labels are used to learn patterns and relationships. Common machine learning techniques in computer vision include:

- Support Vector Machines (SVM): SVM is used for image categorization and classification tasks.
- Random Forests: Ensemble learning for image classification and object detection.
- K-Nearest Neighbours (KNN): Simple algorithm for image recognition tasks.

Deep Learning in Computer Vision: Deep learning, a subset of machine learning, has revolutionized computer vision. It uses Convolutional Neural Networks (CNNs) to automatically learn hierarchical features from visual data. Prominent deep learning approaches in computer vision are:

- Convolutional Neural Networks (CNNs): Used for image recognition, object detection, and segmentation tasks.
- Recurrent Neural Networks (RNNs): Applied to sequential data like videos for action recognition and video analysis.
- Generative Adversarial Networks (GANs): Used for image generation and synthesis tasks.

Applications of Computer Vision: Computer vision using machine learning and deep learning finds application in various fields, including:

- Autonomous Vehicles: For self-driving cars to perceive and navigate surroundings.
- Healthcare: In medical imaging for tumour detection, pathology analysis, and disease diagnosis.
- Robotics: For robot perception, object manipulation, and navigation.

- Surveillance and Security: In video surveillance, facial recognition, and behaviour analysis for enhanced security.

Computer vision using machine learning and deep learning continues to advance rapidly, opening new possibilities and transforming numerous industries. The combination of these approaches has led to significant breakthroughs, making machines more capable of understanding and interacting with the visual world.

2.1 Synthetic Aperture Radar (SAR)

Synthetic Aperture Radar (SAR) is a remote sensing technology that uses radar to capture high-resolution images of the Earth's surface. Unlike traditional optical satellite imagery, SAR is not affected by weather conditions or sunlight, making it particularly useful for all-weather and day-and-night observations.

TYPES OF SAR

- **Passive microwave sensing :**

Passive microwave sensing is similar in concept to thermal remote sensing. All objects emit microwave energy of some magnitude, but the amounts are generally very small. A passive microwave sensor detects the naturally emitted microwave energy within its field of view.

- **Active microwave sensing :**

Active microwave sensors provide their own source of microwave radiation to illuminate the target.

2.2 Working Of SAR

SAR operates by transmitting microwave pulses from a radar system on the satellite towards the Earth's surface. These pulses bounce off the surface features and return to the satellite, where they are recorded by a receiving antenna. SAR employs the concept of coherent processing and uses the motion of the satellite to synthesize a much larger antenna (synthetic aperture) than what is physically present on the satellite. This synthetic aperture allows SAR to achieve high spatial resolution and image quality.

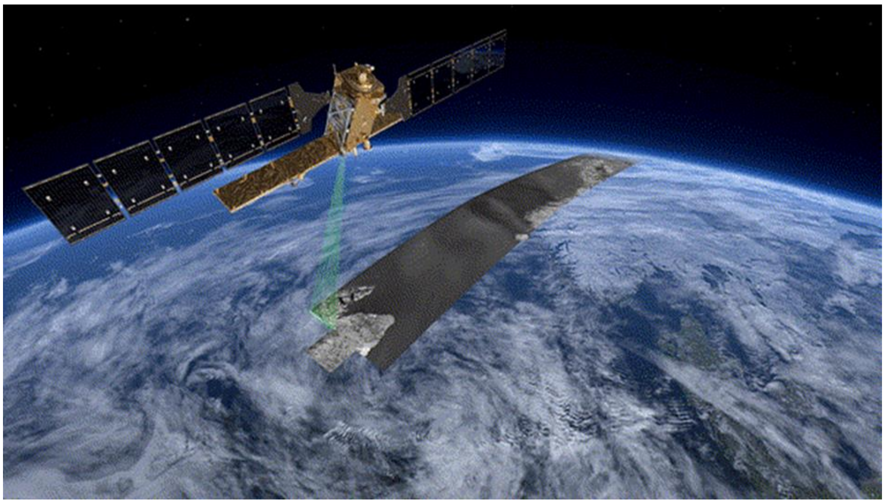


Figure 2.1 SAR Image Processing

2.3 Electromagnetic Spectrum

Synthetic aperture radar is a way of creating an image using radio waves. The radio waves used in SAR typically range from approximately 3 cm up to a few meters in wavelength, which is much longer than the wavelength of visible light, used in making optical images. These wavelengths fall within the microwave part of the spectrum in the figure below.

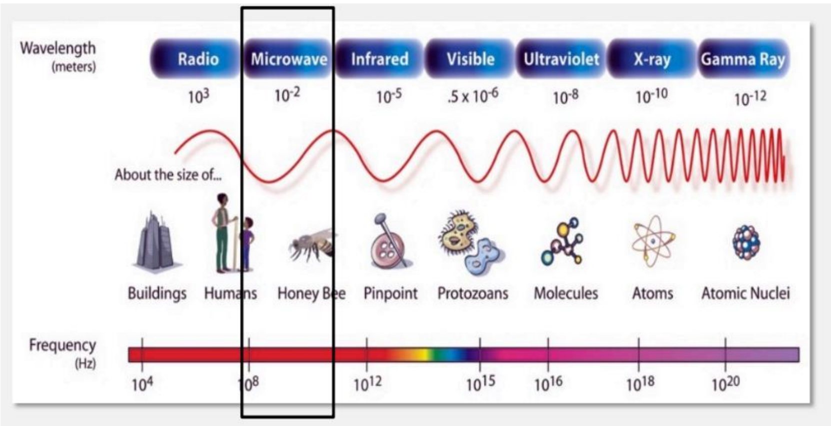


Figure 2.2 Electromagnetic Spectrum

2.4 Radar Image Distortions

As with all remote sensing systems, the viewing geometry of a radar results in certain geometric distortions on the resultant imagery.

❖ Fforeshortening

When the radar beam reaches the base of a tall feature tilted towards the radar (e.g., a mountain) before it reaches the top foreshortening will occur.

Figure 2.3 Foreshortening

❖ Layover

Layover occurs when the radar beam reaches the top of a tall feature (B) before it reaches the base (A). The return signal from the top of the feature will be received before the signal from the bottom. As a result, the top of the feature is displaced towards the radar from its true position on the ground, and "lays over" the base of the feature (B' to A').

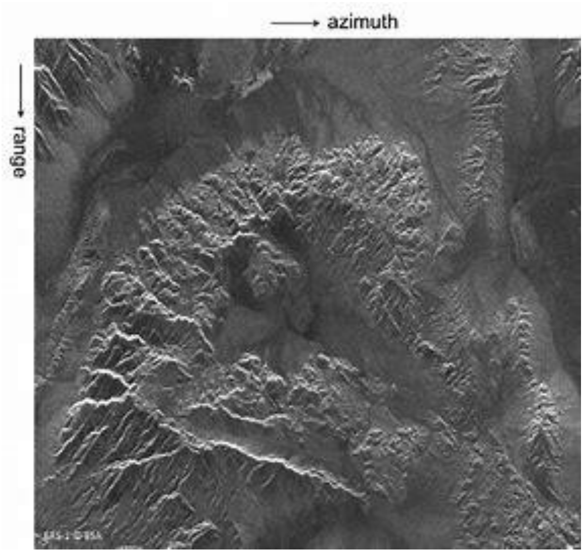


Figure 2.3 Layover

Calibration Device

Calibration is a process which ensures that the radar system and the signals that is measures are as consistent and as accurate as possible prior to analysis, most radar images will require relative calibration.

❖ **Transponders**

These devices receive the incoming radar signal, amplify it, and transmit a return signal of known strength back to the radar. By knowing the actual strength of this return signal in the image, the responses from other features can be referenced to it.

Figure 2.4 Transponders

❖ Corner Reflectors

Artificial radar reflectors, that is, corner reflectors or transponders, are commonly used for: (a) Synthetic Aperture Radar (SAR) sensor calibration and validation.

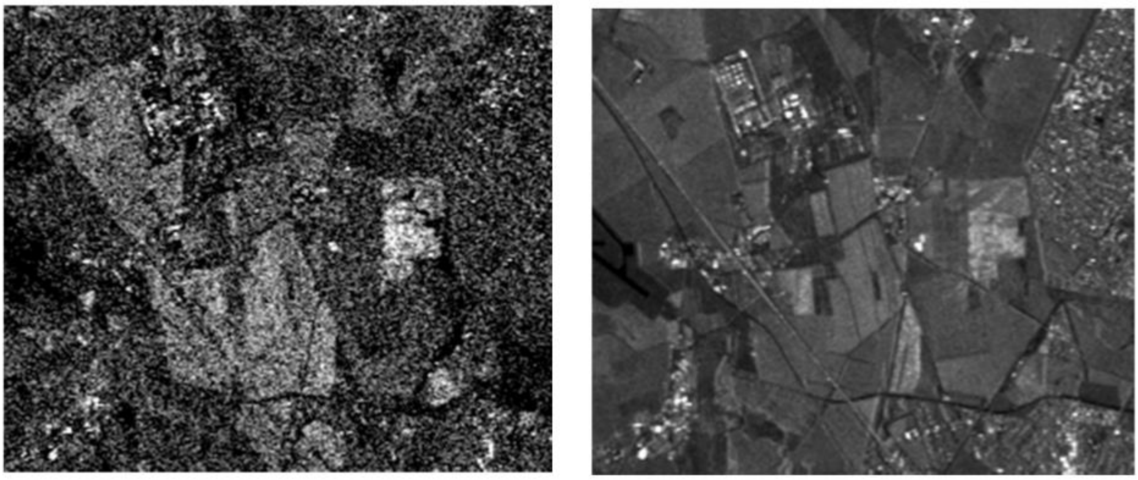
Figure 2.5 Corner Reflectors

Single Look Processing

Single-look processing (SLP) is the simplest and most common method for processing SAR images. In SLP, the SAR image is processed once, using all of the available data. This results in an image that has good azimuth resolution, but it also has high speckle noise. Speckle noise is a random pattern of high-frequency noise that is caused by the interference of multiple radar echoes from different scatterers in the scene.

Multi-look processing

(MLP) is a more sophisticated method for processing SAR images that can reduce speckle noise and improve image clarity. In MLP, the SAR image is processed multiple times, using different subsets of the data. This combines the data from different looks to reduce the effects of speckle noise. The number of looks that can be used in an MLP depends on the amount of data available and the desired level of speckle reduction.



5 looks
20 m x 20 m resolution

320 looks (average of 64 images)
20 m x 20 m ground resolution

Figure 2.6 Single Look and Multi-look processing

❖ Radar Parameters

- **Wavelength :**

Wavelength is the distance between two consecutive peaks of an electromagnetic wave. It is measured in meters and is inversely proportional to frequency. Wavelength is an important parameter in radar systems because it determines the interaction of the radar signal with the target. Shorter wavelengths (higher frequencies) are more sensitive to smaller targets, but they also penetrate less deeply into materials. Longer wavelengths (lower frequencies) are less sensitive to smaller targets, but they can penetrate deeper into materials.

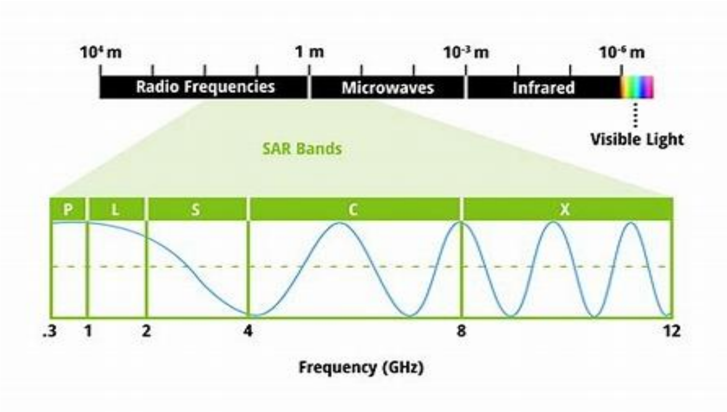


Figure 2.7 Wavelength

- **Polarization :**

Polarization is the property of an electromagnetic wave that describes the orientation of its electric field. Radar systems can transmit and receive waves with different polarizations. The most common polarizations are horizontal polarization (HH) and vertical polarization (VV). HH polarization means that the electric field of the wave is oriented horizontally, and VV polarization means that the electric field of the wave is oriented vertically.

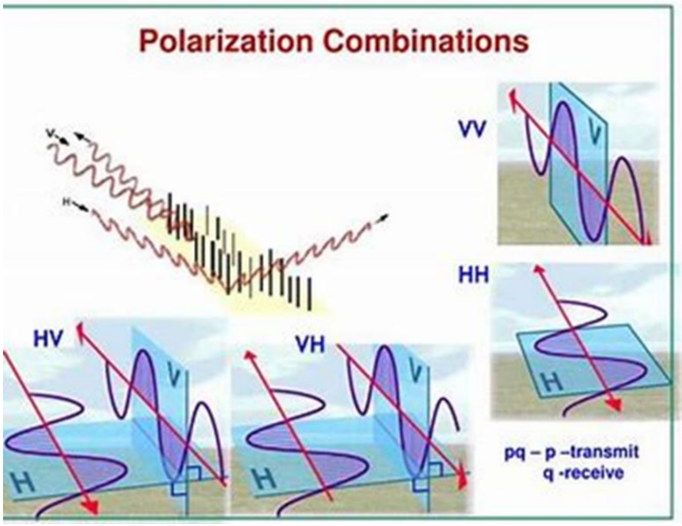


Figure 2.7 Polarization

- **Incidence Angle :**

The incident angle is the angle between the radar beam and the surface of the target. The incident angle affects the amount of radar energy that is scattered back to the radar system. A larger incident angle means that the radar beam is more likely to penetrate the target, and a smaller incident angle means that the radar beam is more likely to be reflected off the surface of the target.

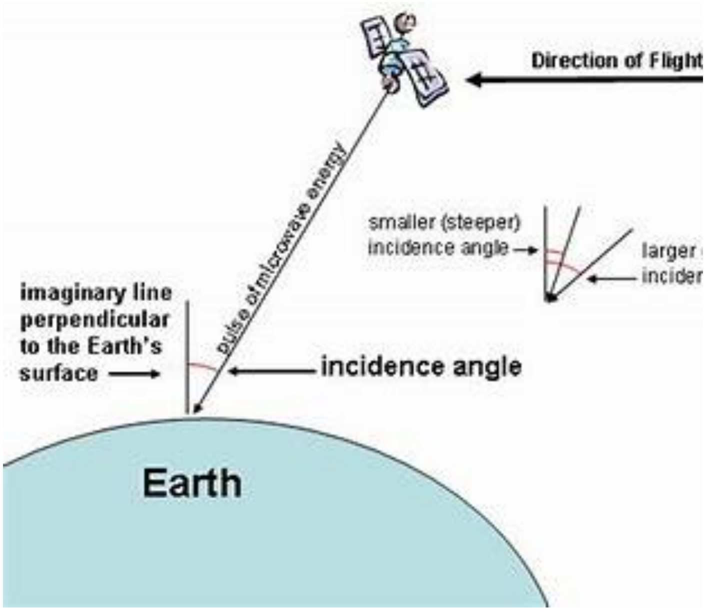


Figure 2.8 Incident Angle

2.2 YOLO (You Only Look Once)

2.2.1 History

YOLO was introduced by Joseph Redmon and Ali Farhadi in 2015. It was the first real-time object detection algorithm that could detect objects in a single pass through the image. Since its initial release, YOLO has undergone several iterations to improve accuracy and speed.

- **YOLOv1 (Version 1):** The first iteration of YOLO introduced the concept of single-pass object detection and real-time processing. Despite its simplicity, YOLOv1 achieved state-of-the-art accuracy on object detection benchmarks at that time.
- **YOLOv2 (YOLO9000):** Released in 2017, YOLOv2 improved accuracy and introduced the use of anchor boxes for better bounding box predictions. It also expanded to detect a broader range of objects, resulting in the name YOLO9000.
- **YOLOv3:** Released in 2018, YOLOv3 further enhanced accuracy by incorporating features like Darknet-53, multi-scale detection, and more anchor boxes. It became a major improvement over its predecessors.
- **YOLOv4:** Released in 2020, YOLOv4 featured even better accuracy and speed with the introduction of CSPDarknet53, PANet, SAM, and other optimizations.
- **YOLOv5:** YOLOv5 is the latest version of the YOLO algorithm, released in 2020. It is known for its improved accuracy and speed and is widely used in real-time object detection tasks.

2.2.2 Architecture

The YOLO architecture is a convolutional neural network (CNN) trained on a large dataset of images with object annotations. It divides the input image into a grid of cells, with each cell responsible for detecting objects within its region. The key components of the YOLO architecture are as follows:

- Each grid cell predicts bounding boxes (usually 2 or more) and the corresponding confidence scores.
- YOLO predicts class probabilities for each object detected in the bounding boxes.

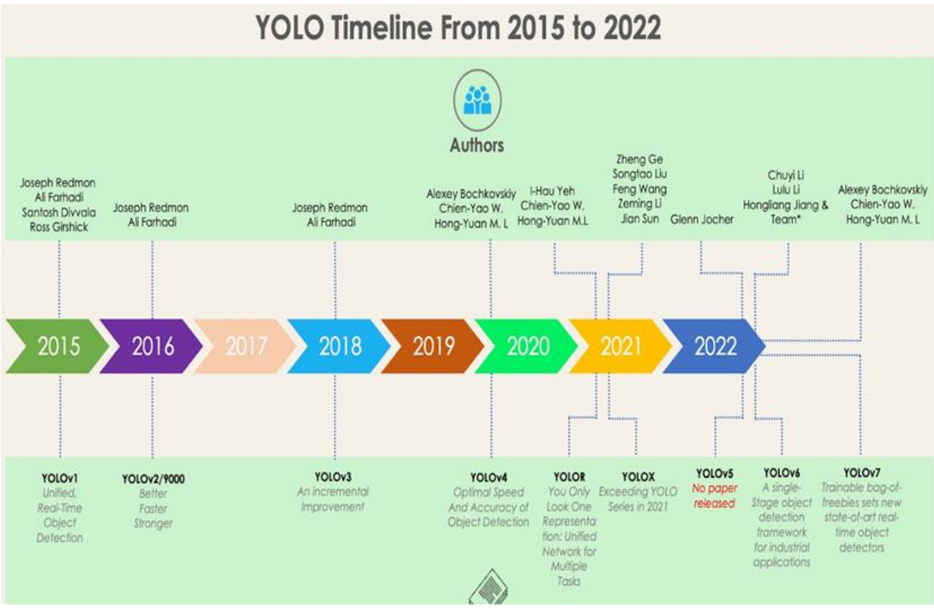


Figure 2.1: YOLO Timeframe

- The model is trained on labelled data with bounding box annotations and class labels.
- YOLO uses anchor boxes to handle different object sizes and aspect ratios, aiding accurate bounding box predictions.

- Non-Maximum Suppression (NMS) is applied to remove duplicate and low confidence detections, retaining the most confident predictions.

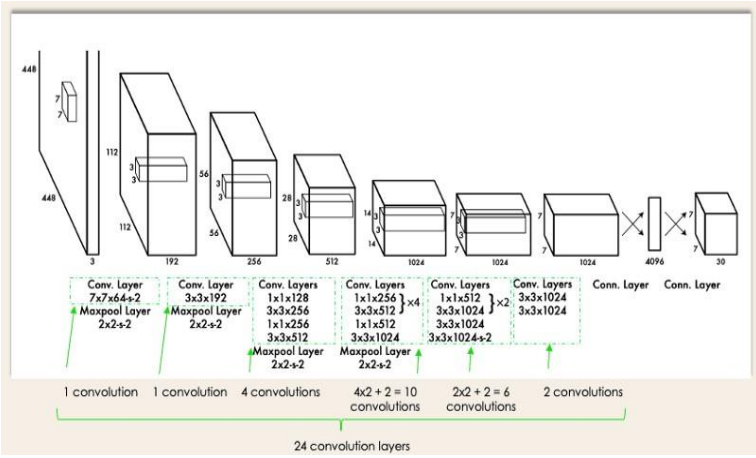


Figure 2.2: YOLO Architecture from the original paper

2.2.3 Evaluation Metrics

The evaluation of YOLO and other object detection models relies on several key metrics:

- **Mean Average Precision (mAP):** mAP is a common evaluation metric for object detection models like YOLO. It measures the precision and recall of object detections at different intersection over union (IoU) thresholds. IoU is the ratio of the area of overlap between predicted and ground-truth bounding boxes to the area of their union. mAP provides a comprehensive evaluation of YOLO's accuracy across different IoU thresholds.

- **Precision and Recall:** Precision measures the accuracy of predicted bounding boxes, while recall measures the coverage of true positives. Both metrics help assess the quality of the object detection model's predictions.

- **Speed and Model Size:** Besides accuracy, the speed and model size of YOLO are crucial factors, especially for real-time applications. YOLO has been optimized over different versions to achieve a balance between accuracy and efficiency.

YOLO's hierarchical history, architecture, and evaluation metrics have made it a widely used and influential object detection framework in computer vision. Its ability to achieve real-time performance while maintaining high accuracy has made it popular in various domains, including autonomous vehicles, surveillance systems, and robotics.

2.4 Dataset

SpaceNet6 is an open-source dataset of half-meter Synthetic Aperture Radar (SAR) imagery from Capella Space and half-meter electro-optical (EO) imagery from Maxar. The dataset was released in 2020 for the SpaceNet 6 Multi-Sensor All-Weather Mapping challenge, which tasked participants with developing algorithms to automatically extract building footprints from the SAR and EO imagery.

The SAR imagery in SpaceNet6 is provided in two formats:

- Single Look Complex (SLC): This is the raw radar data, which is complex-valued and has not been processed to remove artifacts or improve the image quality.
- Pauli decomposition: This is a pre-processed version of the SAR data that has been decomposed into four channels, each of which represents a different polarization of the radar signal.

The EO imagery in SpaceNet6 is provided in RGB format.

The SpaceNet6 dataset includes 202 SAR image strips, each of which covers an area of approximately 20 km x 20 km. The image strips are located in a variety of urban, suburban, and rural environments.

The SpaceNet6 dataset is a valuable resource for researchers and developers who are working on the development of SAR and EO image processing algorithms. The dataset can be used to train and test algorithms for a variety of tasks, such as building extraction, land cover classification, and change detection.

AIM

To develop a machine learning (ML) model using YOLOv5 for the automatic detection and localization of objects in SAR images.

Objectives:

1. To train a YOLOv5 model on a dataset of SAR images with corresponding object annotations.
2. To evaluate the performance of the trained model on a held-out test set.
3. To compare the performance of the YOLOv5 model to other state-of-the-art object detection methods for SAR images.
4. To investigate the potential applications of the YOLOv5 model for SAR image analysis.

Resources:

The project will require the following resources:

- A powerful computer with a GPU
- The YOLOv5 object detection framework
- A dataset of SAR images with corresponding object annotations

Perception through Camera – YOLOv5 on Custom Data

YOLO (You Only Look Once) object detection algorithm. YOLOv5 was developed by Ultralytics and released in 2020. It introduced several improvements over its predecessor, YOLOv4. Here are some key features of YOLOv5:

1. **Model Architecture:** YOLOv5 utilizes a CSPNet (Cross-Stage Partial Network) backbone and PANet (Path Aggregation Network) neck. The model architecture is designed to balance speed and accuracy.
2. **Performance:** YOLOv5 claimed to achieve competitive or superior performance to other object detection algorithms while being computationally efficient. It aimed to provide real-time performance on a variety of hardware.
3. **Flexibility:** YOLOv5 comes in various model sizes (e.g., YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x), allowing users to choose a model based on their specific requirements, considering factors like speed and accuracy.

4. **Training Enhancements:** YOLOv5 introduced features like automatic mixed precision training and a larger default input size to improve training efficiency and accuracy.
5. **Object Detection and Classification:** YOLOv5 can detect and classify objects in an image, providing bounding box coordinates and class probabilities for each detected object.
6. **Ease of Use:** The YOLOv5 codebase is designed to be user-friendly and accessible, with a focus on ease of use and deployment.
7. **Open Source:** YOLOv5 is an open-source project, and its code is available on GitHub, allowing the community to contribute, modify, and use the algorithm for various applications.

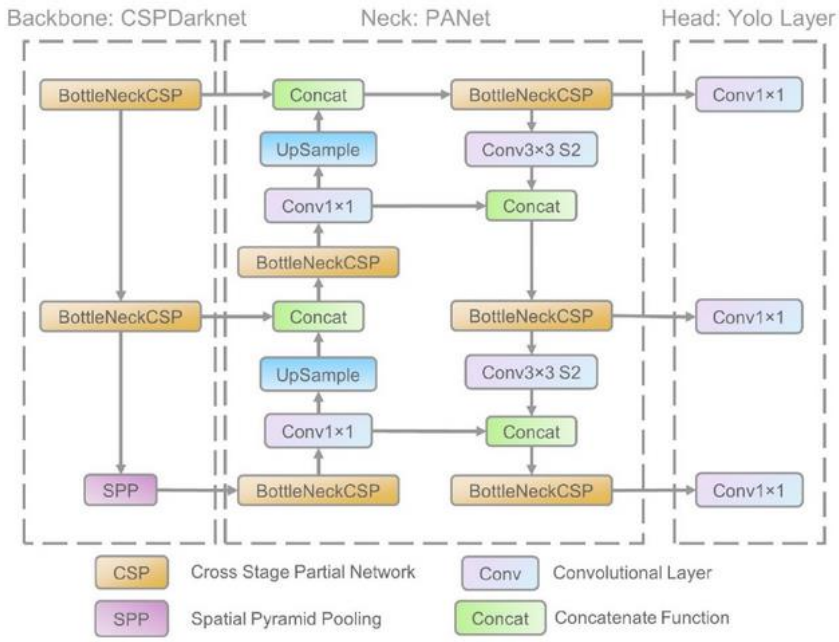


Figure Network architecture for YOLO v5

4.3 CUSTOM DATASET & TRAINING

SpaceNet6 is an open-source dataset of half-meter Synthetic Aperture Radar (SAR) imagery from Capella Space and half-meter electro-optical (EO) imagery from Maxar. The dataset was released in 2020 for the SpaceNet 6 Multi-Sensor All-Weather Mapping challenge, which tasked participants with developing algorithms to automatically extract building footprints from the SAR and EO imagery.

4.3.1 SAR Image Processing

First Step is to separate 4 – Bands from the TIFF image. We can use Python code to separate 4 – Bands from the Image.

```
from osgeo import gdal
# Open the TIFF image
dataset = gdal.Open('image.tif')
# Get the number of bands
num_bands = dataset.RasterCount
# Iterate through each band and save it as a separate image
for i in range(1, num_bands + 1):
    # Get the band data
    band = dataset.GetRasterBand(i)
    band_data = band.ReadAsArray()
    # Create a new GeoTIFF image
    driver = gdal.GetDriverByName('GTiff')
    output_dataset = driver.Create('band_{i}.tif', dataset.RasterXSize,
dataset.RasterYSize, 1, gdal.GDT_Float32)
    output_dataset.SetGeoTransform(dataset.GetGeoTransform())
    output_dataset.SetProjection(dataset.GetProjection())
```

```
# Write the band data to the new image
output_band = output_dataset.GetRasterBand(1)
output_band.WriteArray(band_data)
# Flush the output dataset
output_dataset.FlushCache()
```

This code will also read the TIFF image image.tif and save each of its four bands as a separate TIFF image named band_1.tif, band_2.tif, band_3.tif, and band_4.tif.

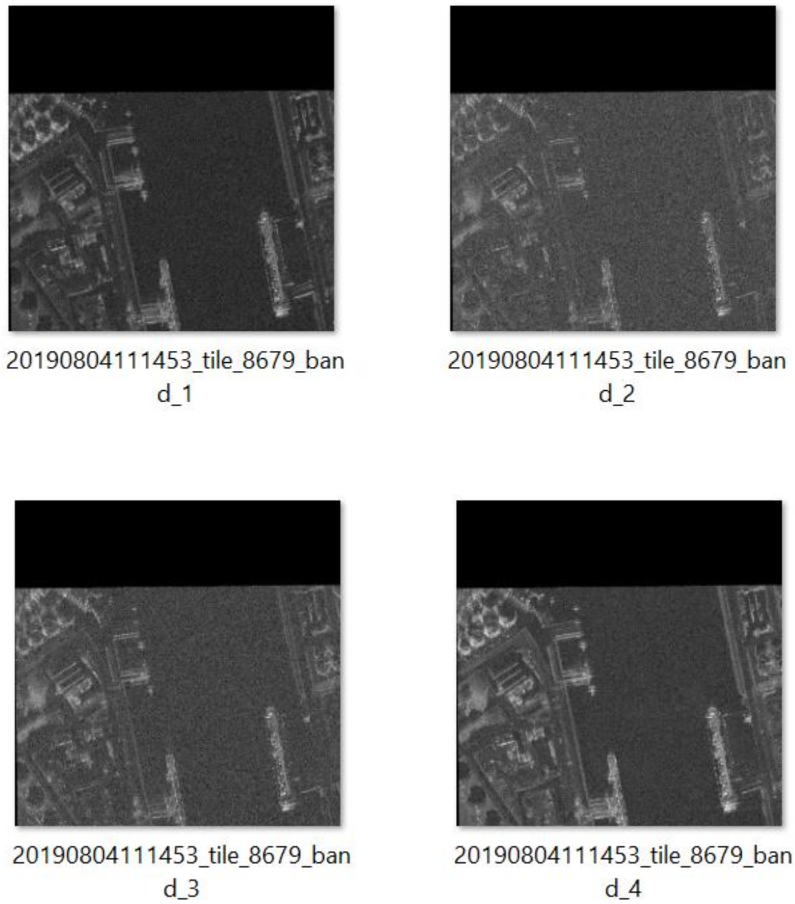


Figure 4.3.1 4-Bands of TIFF Image

4.3.1 Create Annotations with RoboLabelImg

LabelImg provides a graphical interface where users can load images, draw bounding boxes around objects, and assign class labels.

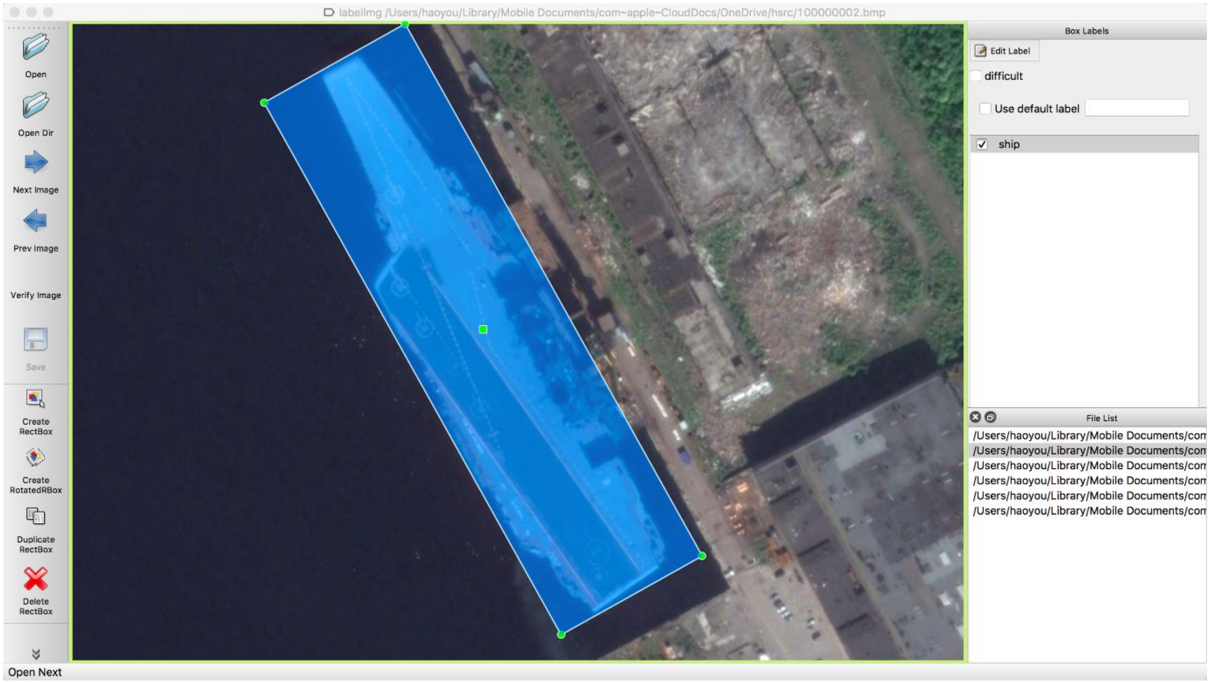


Figure 4.3.1 graphical interface

As per the requirements and specifications given a custom data set was prepared in YOLO format, and the hyperparameters are tuned accordingly and are trained on YOLOv7 architecture and YOLOv8 architecture to find the mAP, recall, precision and other metrics and to find out which is better for actual deployment and usage in real life applications for Integration.

YOLO XML file format :

</width></height></cx-centre></cy-centre></angle>

4.3.2 Types of Bounding Box's :

❖ OBB

An OBB is a rectangle that is aligned with the object's principal axes. The principal axes are the two axes of the object that have the largest and smallest extents. The OBB is defined by its centre coordinates, width, height, and angle of rotation. The angle of rotation is the angle between the OBB's longest side and the horizontal axis.

❖ HBB

An HBB is a rectangle that is aligned with the horizontal and vertical axes of the image or video frame. The HBB is defined by its centre coordinates, width, and height.

Conversion between OBB and HBB

There are two main methods for converting between OBB and HBB:

- Minimum enclosing HBB: The minimum enclosing HBB is the smallest HBB that can be drawn around the OBB. This method is simple and efficient, but it can result in an HBB that is larger than the actual object.

- Rotated HBB: The rotated HBB is an HBB that is aligned with the OBB's principal axes. This method is more accurate than the minimum enclosing HBB, but it is more complex.

4.3.2 Conversion of XML to TXT

- YOLOv5-Utils It's a collection of utility tools for YOLOv5, an object detection model. voc2yolo5_obb.py script is used to convert XML annotation data into a format that can be used with YOLOv5.

- BboxToolkit It's a tiny library of special bounding boxes.

The main features of BboxToolkit include:

Various types of bounding boxes: The library defines three different types of bounding boxes: horizontal bounding boxes (HBB), oriented bounding boxes (OBB), and 4 point polygon (POLY). Each type of box can convert to others easily.

Convenience for usage: The functions in BboxToolkit will decide the box type according to the input shape.

The project is used to support the oriented detection benchmark OBBDetection. They are currently developing BboxToolkit v2.0 intended to support the new OBBDetection based on MMdetection v2.10.

Results and Discussions

YOLOv5, similar to YOLOv7, is designed to be a lightweight and efficient model, making it suitable for deployment scenarios where real-time processing and power efficiency are crucial12. It is known for its ease of use and versatility, being able to run efficiently on both CPU and GPU. YOLOv5 employs various data augmentation techniques to improve the model's ability to generalize and reduce overfitting2. These techniques include Mosaic Augmentation, an image processing technique that combines four training images into one in ways that encourage object detection models to better handle various object scales and translations

Future Scope

The future scope for the integration of YOLO-based architectures in Computer Vision applications is promising and expansive. As technology continues to advance, the following areas offer significant opportunities for further exploration and improvement:

1. **Real-time Applications**: YOLO's speed and efficiency make it ideal for real-time applications such as autonomous vehicles, robotics, and surveillance systems, enabling faster and more accurate object detection and tracking.
2. **Small Object Detection**: As technology evolves, there will be a growing need for detecting smaller and more intricate objects. Future research will focus on optimizing YOLO architectures to handle challenging scenarios involving tiny objects, such as detecting small animals or tiny debris.

In conclusion, the future of YOLO-based architectures in Computer Vision is bright and multifaceted. With ongoing research and innovation, YOLO will continue to shape the landscape of object detection, enabling a wide range of applications and advancements in the field.

References

1. You Only Look Once: Unified, Real-Time Object Detection by Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, Submitted on 8 Jun 2015 (v1), last revised 9 May 2016 (v5).