Deploy an Object Detection Algorithm on Nvidia Drive PX2 Hardware and Benchmark its Performance

A project report

submitted by

Mahesh Ravindra Chaudhari

ED18B018

in partial fulfilment of requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY IN ENGINEERING DESIGN AND MASTER OF TECHNOLOGY IN AUTOMOTIVE ENGINEERING

DEPARTMENT OF ENGINEERING DESIGN

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

24th May 2023

Certificate

This is to certify that the project titled **"Deploy an Object Detection Algorithm on Nvidia Drive PX2 Hardware and Benchmark its Performance"** submitted by **Mahesh Ravindra Chaudhari**, to the Indian Institute of Technology Madras, Chennai for the award of the degree of **BACHELOR OF TECHNOLOGY IN ENGINEERING DESIGN** and **MASTER OF TECHNOLOGY IN AUTOMOTIVE ENGINEERING**, is a Bonafede record of the research work done by him under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Jayaganthan R

Project Advisor

Professor

Department of Engineering Design

Indian Institute of Technology Madras

Chennai 600036.

Signature

Place: Chennai

Date: 24th May 2023

Acknowledgements

It is my sublime duty to express my deepest sense of gratitude and veneration to my respected supervisor Prof. R. Jayaganthan, Department of Engineering Design, Indian Institute of Technology Madras for his sincere exhortation, indelible inspiration, constant encouragement, constructive criticism, sustained interest, immense patience and supporting attitude throughout the investigation of the present research problem and preparation of this thesis.

I am also thankful for the teaching and official staff (technical and non-technical) of the Engineering Design Department for providing me with a nice learning environment. I sought inspiration and I owe a great deal to my brother Somesh Chaudhari, my mother Smt. Amita Chaudhari and my father Shri. Ravindra Chaudhari for their unfailing love and support throughout the journey of my research work. I also would like to thank all my colleagues at Bosch Global Software Technologies who supported me in my work, especially Mr. Raghavendra Muliya who helped me grow both personally and professionally.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last, but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Abstract

Object detection algorithms have been widely used in many real-world applications, such as autonomous driving, security surveillance, and robotic vision. As the demand for real-time and efficient object detection continues to grow, system-on-chips (SoCs) have become an attractive option for deploying object detection algorithms in resource-constrained environments. Through this study, I aim to deploy an object detection algorithm on Nvidia Drive PX2 hardware and benchmark its performance in terms of accuracy and speed. The Nvidia Drive PX2 hardware offers a powerful computing platform with integrated GPUs, making it an excellent choice for deploying and evaluating object detection algorithms.

I wanted to evaluate and compare the performance of the algorithm when implemented on Nvidia PX2 hardware versus Laptop Personal Computer. In this project I use pretrained YOLO object detection model on COCO dataset, to perform object detection. The findings will provide valuable insights into the algorithm's performance and inform future enhancements and optimizations for object detection in autonomous vehicles.

Table of Contents

List of Figures

Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
CNN	Convolutional Neural Network
YOLO	You Only Look Once
CV	Computer Vision
IoU	Intersection over Union
mAP	Mean Average Precision
CPU	Central Processing Unit
GPU	Graphics Processing Unit
SOC	System on Chip
UART	Universal Asynchronous Receiver – Transmitter
CAN	Controller Area Network
LIN	Local Interconnect Network

Introduction

Autonomous driving is an emerging technology that has the potential to revolutionize transportation by making driving safer and more efficient. One of the key components of autonomous driving is object detection, which allows the vehicle to recognize and respond to objects in its environment, such as pedestrians, vehicles, and obstacles.

In this project, I will be focusing on deploying an object detection algorithm called YOLO (You Only Look Once) for autonomous driving. YOLO is a state-of-the-art deep learning algorithm that can detect objects in real-time with high accuracy and speed.

The project will involve several steps, including data collection, preprocessing, training, and deployment. I will use a popular open-source platform like OpenCV to implement and test the YOLO algorithm on a dataset of images and video clips. Finally, I'll define what next steps can be taken to complete this project and what is the future scope.

Overall, the goal of this project is to gain hands-on experience in applying deep learning techniques for detecting and classifying objects in images and videos. The knowledge of this project can be extended to develop production level algorithms which can be used in the Near Range Camera System of Bosch.

Artificial Intelligence and Machine Learning

Artificial Intelligence encompasses a wide range of computer science disciplines focused on the development of "intelligent agents." These agents possess the ability to perceive their environment and take actions to enhance their chances of achieving a specific goal. Essentially, Artificial Intelligence enables robots and machines to emulate human-like cognitive capabilities, including learning, planning, reasoning, perception, and problem-solving.

Within the realm of Artificial Intelligence, machine learning serves as a subset. Machine learning refers to the use of algorithms that can learn from data and generate data-driven predictions. The process involves mathematical optimization and computational statistics to derive these predictions. Rather than being explicitly programmed, a machine learning program is trained using data. During the training phase, the algorithm identifies statistical structures or patterns in the data and creates rules to automate the process. Continuous feedback signals are utilized to measure the error between the current output and the true output, facilitating ongoing adaptation and adjustment of the algorithm during the learning process.

- **History of Artificial Intelligence**

  Artificial Intelligence is a term that has been known in academic circles for quite some time, indicating that it is not a new concept or technology. In fact, its roots can be traced back to much earlier times than one might expect. Even in the mythologies of Ancient Greece and Egypt, there exist stories and legends featuring the concept of mechanical men. These narratives highlight the long-standing fascination with the idea of creating intelligent beings. The journey from the early stages of AI to its current development can be marked by several significant milestones, which have played a crucial role in shaping the field's evolution.

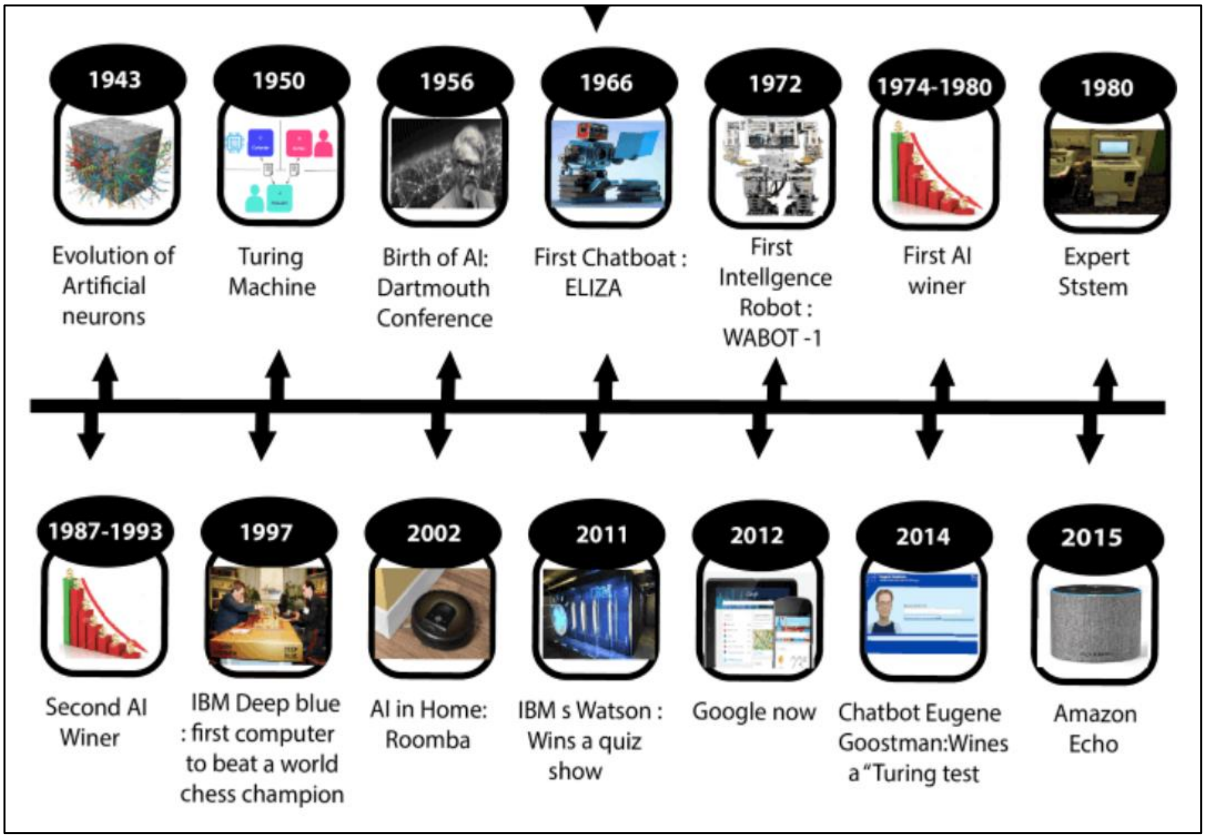


Fig. (1) History of AI and ML

- **History of Machine Learning**

Machine learning (ML) plays a fundamental role in harnessing the power of artificial intelligence (AI) technology. While machine learning is often referred to as AI due to its ability to learn and make decisions, it is, in fact, a subset of AI. Initially, machine learning was a part of AI's development until the late 1970s when it branched out and started evolving independently. Today, machine learning has become a vital tool in various domains, including cloud computing, eCommerce, and other cutting-edge technologies.

In the modern business and research landscape, machine learning has become indispensable for many organizations. It leverages algorithms and neural network models to enable computers to enhance their performance over time. Machine learning algorithms autonomously create mathematical models using sample data, commonly known as "training data," to make decisions without requiring explicit programming for each decision-making scenario. This iterative learning process allows machines to continuously improve their decision-making abilities.

Literature Survey

- Title: Hardware Acceleration and Implementation of YOLOX-s for On-Orbit FPGA. (Year published: 2022)

  Authors: Ling Wang 1,2 , Hai Zhou 1, Chunjiang Bian 1, Kangning Jiang 1,2 and Xiaolei Cheng

  The rapid advancement of remote sensing technology has led to a significant increase in the volume of remote sensing picture data. However, processing such large amounts of data poses challenges due to the limited hardware, size, and power consumption of satellites. Traditional techniques for remote sensing image processing are not effective and robust enough to cope with this growing demand. Furthermore, the task of satellite-to-ground target recognition requires higher speed and accuracy as the availability of remote sensing data continues to rise. To address these issues, this study proposes an efficient and reliable acceleration design for the forward inference of the YOLOX detection network, employing an on-orbit FPGA.

  Given the constraints of onboard resources, the design technique incorporates parallel loop unrolling of the input and output channels. This approach creates the largest DSP computing array, ensuring the optimal utilization of scarce computing resources while reducing the overall inference delay of the network. Additionally, a small-scale cascaded pooling array and a three-path cache queue are implemented to maximize the reuse of on-chip cache data. This effectively reduces the bandwidth barrier of the external memory, enhancing the efficiency of the computing array.

  According to experimental findings, the FPGA acceleration achieves an overall inference performance of 399.62 GOPS and a peak performance of 408.4 GOPS, operating at a frequency of 200 MHz on the VC709 platform. The overall computing efficiency of the DSP array reaches an impressive 97.56%. Compared to previous work, our architecture design significantly enhances computing performance.

- Title: Hardware Accelerator for Object Detection using Tiny YOLO-v3

  Authors: Manan Sharma, Rahul R, Madhusudan S, Deepu S.P, Sumam David S.

  This research paper introduces a specialized hardware accelerator designed to enable real-time object detection using the cutting-edge Tiny YOLO-v3 algorithm. The proposed architecture strikes a reasonable balance between computation speed, measured in frames per second (FPS), and the necessary hardware resources. To achieve this, the design incorporates pipelining and parameterization for each convolutional neural network (CNN) layer, ensuring adaptability and reconfigurability.

  The proposed hardware accelerator was synthesized using the SCL (Semi-Conductor Laboratory, India) 180 nm CMOS process, as well as the Vivado Xilinx software, with the Virtex Ultrascale+ FPGA serving as the targeted device. The implementation of a pipelined architecture, alongside other architectural innovations, yielded remarkable results. The proposed solution achieved a higher frame rate of 32.1 FPS and a performance of 166.4 GOPS at a clock frequency of 200 MHz.

- Title: You Only Look Once: Unified, Real-Time Object Detection

  Authors: Joseph Redmon_, Santosh Divvala, Ross Girshick, Ali Farhadi

  In this paper, we introduce YOLO, a novel approach to object detection that differs from previous methods which repurpose classifiers for this task. Instead, we formulate object detection as a regression problem, predicting spatially separated bounding boxes and associated class probabilities.

Our unified architecture utilizes a single neural network to directly predict bounding boxes and class probabilities from full images in one evaluation. This allows for end-to-end optimization, directly improving detection performance.

Notably, our YOLO model exhibits exceptional speed. The base model achieves real-time processing, handling images at an impressive rate of 45 frames per second. Furthermore, a smaller variant of the network called Fast YOLO achieves an astonishing processing speed of 155 frames per second, while still achieving double the mean average precision (mAP) of other real-time detectors. Although YOLO may exhibit more localization errors compared to state-of-the-art detection systems, it demonstrates a lower tendency to produce false positives on background. Additionally, YOLO excels in learning highly generalized object representations. It outperforms other detection methods, such as DPM and R-CNN, when applied to diverse domains, including artwork, showcasing its versatility and effectiveness.

- Title: Real Time Object Detection with Yolo

  Authors: Geethapriya. S, N. Duraimurugan, S.P. Chokkalingam

The objective of this study is to utilize the You Only Look Once (YOLO) approach for object detection. YOLO offers numerous advantages compared to other existing object detection algorithms. Unlike Convolutional Neural Network (CNN) or Fast-Convolutional Neural Network (Fast-CNN) algorithms, YOLO comprehensively analyzes the entire image. It achieves this by leveraging a convolutional network to predict bounding boxes and associated class probabilities. As a result, YOLO exhibits faster object detection capabilities compared to alternative algorithms.

YOLO's ability to consider the entire image grants it a unique advantage. Other algorithms may not fully capture the entire visual context, limiting their detection accuracy. In contrast, YOLO takes a holistic approach to image analysis, enabling more accurate and precise object detection. One of the key strengths of YOLO lies in its speed. By processing the entire image in one evaluation, YOLO achieves efficient and real-time object detection. This capability sets YOLO apart from traditional algorithms, making it highly suitable for applications requiring fast and accurate object recognition.

In summary, YOLO's approach to object detection sets it apart from other algorithms. By examining the complete image and leveraging convolutional networks to predict bounding boxes and class probabilities, YOLO achieves faster and more comprehensive object detection. Its efficiency and accuracy make it a highly promising choice for a range of applications.

- Title: Object Detection through Modified YOLO Neural Network

  Authors: Tanvir Ahmad , Yinglong Ma , Muhammad Yahya, Belal Ahmad, Shah Nazir and Amin ul Haq.

  Object detection has seen significant progress in recent times, but accurately and rapidly detecting and identifying objects remains a challenging task. While humans effortlessly recognize objects regardless of their appearance, computers struggle with this task. This paper presents a modified YOLOv1-based neural network for object detection, aiming to address these challenges. The proposed model introduces improvements in multiple aspects. Firstly, the loss function of the YOLOv1 network is modified, replacing the margin style with a more flexible and reasonable proportion style. This modification enhances the optimization of network error. Secondly, a spatial pyramid pooling layer is incorporated to improve performance.

Additionally, an inception model with a 1x1 convolution kernel is introduced, reducing the number of weight parameters in the layers. Extensive experiments conducted on the Pascal VOC datasets from 2007/2012 validate the effectiveness of the proposed method, demonstrating better performance compared to existing approaches. Through these modifications, the proposed modified YOLOv1-based neural network addresses the challenges associated with accurate and efficient object detection. The improvements in the loss function enhance the network's ability to optimize network error. The spatial pyramid pooling layer and inception model contribute to performance improvement while reducing the number of weight parameters. The experimental results on Pascal VOC datasets confirm the superior performance achieved by the proposed method. Overall, this paper presents a modified YOLOv1-based neural network for object detection, addressing the challenges faced in accurately and efficiently detecting and identifying objects.

- Title: 3D Object Detection for Autonomous Driving: A Survey

  Authors: Rui Qian , Xin Lai , Xirong Li

Autonomous driving has emerged as a promising solution to mitigate severe accidents, with 3D object detection playing a crucial role in the perception stack for tasks like path planning, motion prediction, and collision avoidance. Despite notable advancements, challenges persist in areas such as recovering visual appearance without depth information, learning representations from partially occluded point clouds, and achieving semantic alignments across different modalities. While numerous studies have investigated 3D object detection for autonomous driving, there is a lack of comprehensive surveys that organize and synthesize this growing body of knowledge.

In this paper, we aim to bridge this gap by presenting a comprehensive survey that covers essential aspects such as sensors, datasets, performance metrics, and the latest state-of-the-art detection methods, including their respective strengths and weaknesses. Additionally, we offer quantitative comparisons with the current state of the art. To provide a practical perspective, we conduct a case study involving fifteen representative methods, encompassing runtime analysis, error analysis, and robustness assessment. Finally, after a thorough analysis of the surveyed literature, we provide concluding remarks and outline promising avenues for future research.

- Title: Quantitative Analysis of Object Detectors for Autonomous Driving and Autonomous Parking

  Author: Pritesh kumar Gohil, Bonn-Rhein-Sieg, Paul G. Plöger, André Hinkenjann

  Current state-of-the-art (SOTA) object detectors are typically evaluated on popular object detection challenge datasets. However, there is a lack of comprehensive quantitative analysis specific to the automotive domain, leading to incomplete evaluations. Furthermore, the evaluation of detectors predominantly relies on the average precision (AP) metric, which fails to capture the shape of the precision-recall curve.

  This paper aims to address these limitations by evaluating the most widely used SOTA detectors on datasets specifically designed for autonomous driving and autonomous parking. To overcome the shortcomings of the AP metric, the paper introduces the Localization-Recall-Precision (LRP) metric, which provides a more detailed understanding of a detector's performance. Additionally, the paper introduces the optimal AP (oAP) metric, which enables a fair comparison of detectors and was previously overlooked.

The proposed oAP metric can be easily utilized with any object detector, enhancing the evaluation process. Moreover, the paper presents a comprehensive object detection pipeline that covers all stages from data collection to deployment on embedded devices. To provide a deeper analysis of the detectors, the experimental results are presented using visual representations, allowing for a different perspective on detector performance.

Deep Learning for Object Detection and Classification

Deep learning is a branch of machine learning that uses artificial neural networks to learn and recognize patterns in data. One of the most popular applications of deep learning is in the field of computer vision, where it is used for tasks such as object detection and classification.

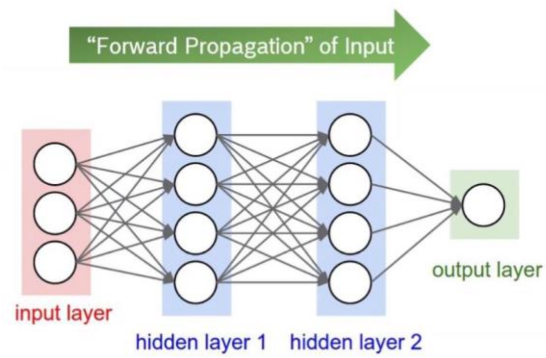


Fig. (2) Neural Network Architecture

Object detection and classification are important problems in computer vision, and they are used in a wide range of applications, from self-driving cars to security cameras. Object detection refers to the process of identifying the location and size of objects in an image or video stream, while object classification refers to the process of assigning a label to each object based on its category.

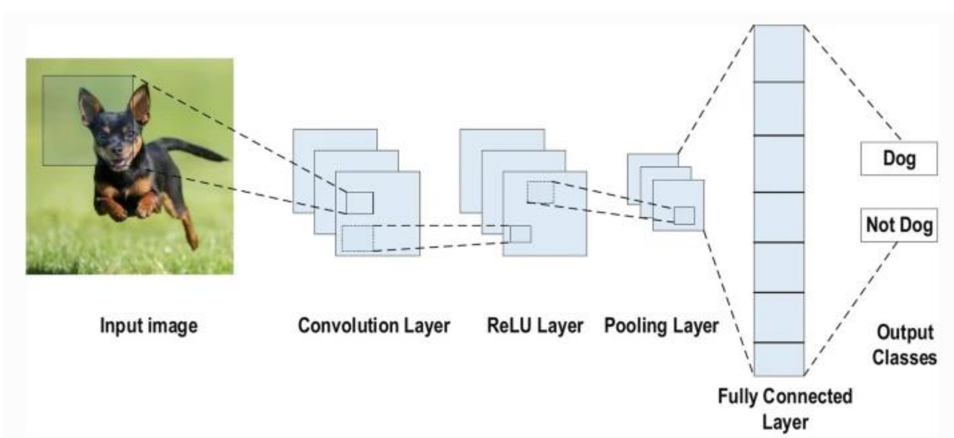


Fig. (3) Image Classification Architecture

Deep learning has revolutionized object detection and classification by enabling the creation of highly accurate and efficient models that can handle complex visual data. The most popular deep learning architectures used for object detection and classification are convolutional neural networks (CNNs).

Convolutional Neural Networks

CNNs are composed of several layers of interconnected nodes, each of which performs a specific operation on the input data. The first layer is typically a convolutional layer, which applies a set of filters to the input image to extract features such as edges, corners, and textures. The output of the convolutional layer is then passed through several additional layers, such as pooling layers and fully connected layers, which help to reduce the dimensionality of the data and extract higher-level features.

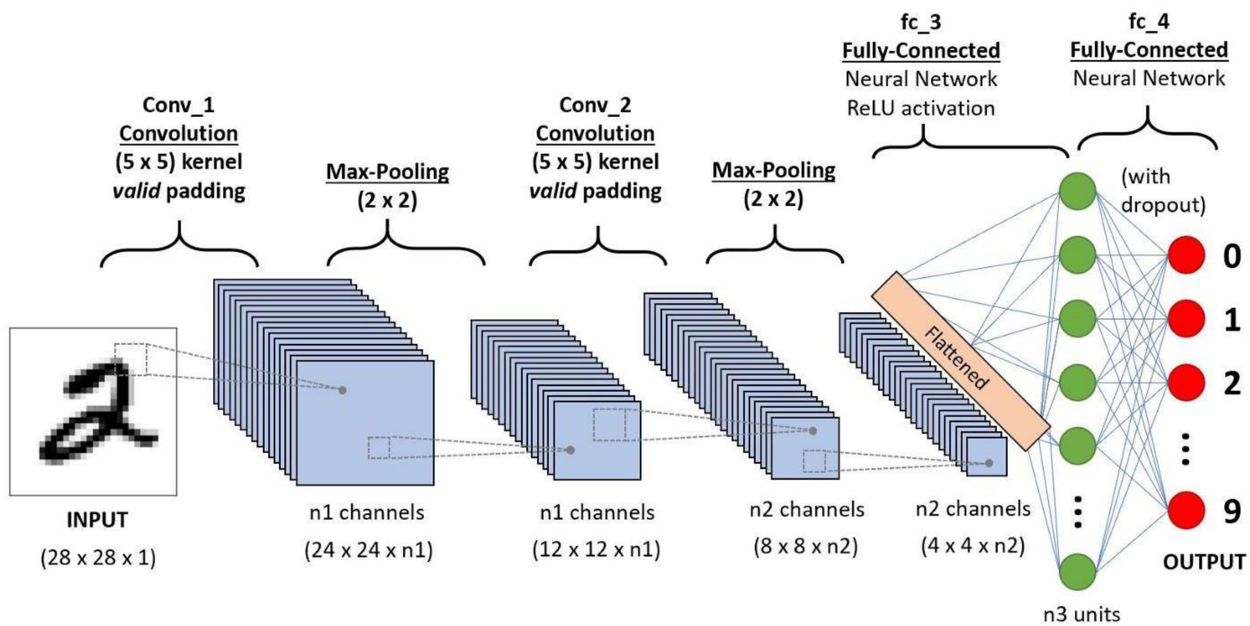


Fig. (4) Fully Connected Convolutional Neural Network

There are several popular architectures for object detection and classification that use CNNs, including YOLO (You Only Look Once), Faster R-CNN (Region-based Convolutional Neural Networks), and SSD (Single Shot Detector). These architectures differ in their approach to object detection and classification, but they all use CNNs to extract features from the input data and make predictions about the location and category of objects.

YOLO is a popular architecture for real-time object detection, which can detect objects in an image or video stream in a single pass. YOLO divides the input image into a grid of cells and predicts the probability of each cell containing an object and the location and size of the object within the cell. This approach is faster than other architectures that use multiple passes over the input data, but it may not be as accurate in some cases.

Faster R-CNN is another popular architecture for object detection, which uses a region proposal network (RPN) to generate candidate regions of interest in the input image. The RPN produces a set of bounding boxes around potential objects in the image, which are then passed through a CNN for classification and refinement. This approach is slower than YOLO, but it is more accurate and can handle a wider range of object sizes and shapes.

In conclusion, deep learning has revolutionized object detection and classification by enabling the creation of highly accurate and efficient models that can handle complex visual data. CNNs are the most popular deep learning architectures used for object detection and classification, and they have been used to create several popular object detection and classification models, including YOLO, Faster R-CNN, and SSD. These architectures differ in their approach to object detection and classification, but they all use CNNs to extract features from the input data and make predictions about the location and category of objects.

Object Detection Algorithms

There are several object detection algorithms available that have been developed to address the task of detecting objects in images and videos. Some of the commonly used object detection algorithms include:

1. R-CNN (Region-based Convolutional Neural Network)
2. Fast R-CNN (Fast Region-based Convolutional Neural Network)
3. Faster R-CNN (Faster Region-based Convolutional Neural Network)
4. SSD (Single Shot MultiBox Detector)
5. YOLO (You Only Look Once)
6. RetinaNet
7. Mask R-CNN
8. Cascade R-CNN
9. EfficientDet
10. CenterNet

Each of these algorithms utilizes different techniques and architectures to detect objects in varying levels of accuracy, speed, and complexity. The choice of algorithm depends on factors such as the specific requirements of the application, available computational resources, and the trade-off between accuracy and speed.

Why YOLO

YOLO (You Only Look Once) stands out as a popular choice among object detection algorithms for several reasons. Here are some key advantages of YOLO over other object detection algorithms:

- **Real-time performance**

  YOLO is known for its impressive real-time processing capabilities. It can detect objects in an image or video frame in a single pass, making it significantly faster compared to many other algorithms that rely on region proposal methods. This real-time performance is particularly crucial for applications such as autonomous vehicles, where timely object detection is critical for safe and efficient operation.

- **Simplicity and efficiency**

  YOLO adopts a straightforward architecture that combines object detection and classification into a single neural network. This simplicity not only makes YOLO easier to implement and understand but also enables efficient computations. The streamlined design eliminates the need for complex post-processing steps, resulting in faster inference times and efficient resource utilization.

- **Accuracy across object scales**

  YOLO performs well across a wide range of object scales. It divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell. This grid-based approach allows YOLO to handle objects of different sizes and aspect ratios effectively.

While other algorithms struggle with detecting small objects or objects at varying scales, YOLO excels in maintaining accuracy across different object sizes.

- **Contextual information**

  YOLO considers global context information during object detection. By incorporating features from multiple spatial scales and using multiple convolutional layers, YOLO can capture contextual details that aid in accurate object localization and recognition. This contextual understanding contributes to robust detection performance, especially in complex scenes with overlapping objects or occlusions.

- **Versatility**

  YOLO is versatile and can be applied to various object detection tasks, including general object detection, pedestrian detection, and even real-time face detection. Its flexibility makes it suitable for a wide range of applications beyond autonomous vehicles, such as surveillance systems, robotics, and object recognition in images and videos.

While YOLO offers these advantages, it's important to note that the choice of object detection algorithm depends on specific requirements and constraints. Different algorithms may excel in different scenarios or have varying trade-offs between speed, accuracy, and computational resources. Evaluating the needs of the application and considering the specific challenges will help determine the most suitable object detection algorithm for a given task.

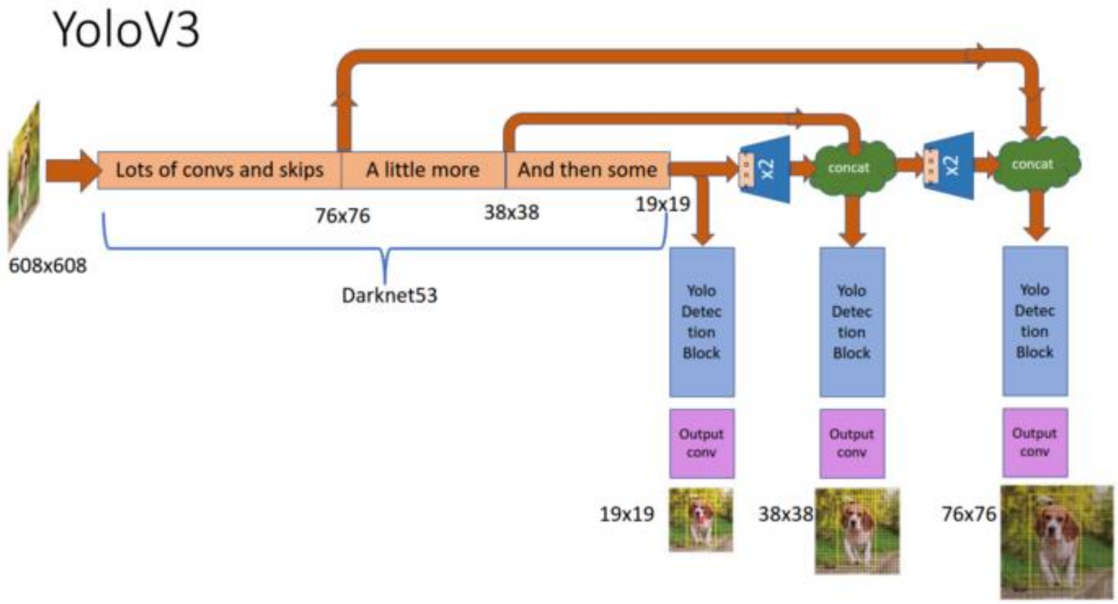
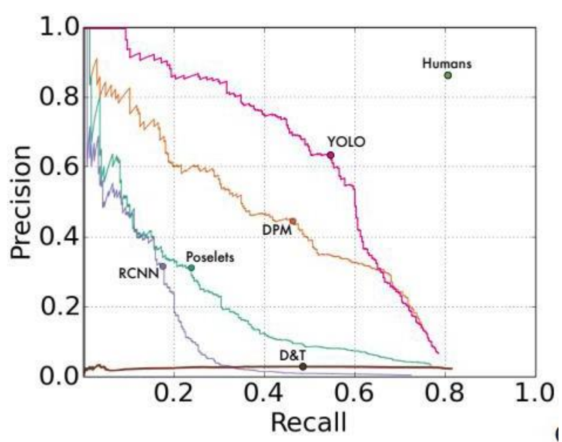


Fig. (5) Architecture of YOLOV3 Algorithm

It is researched and tested that YOLO algorithm has high accuracy and overall speed of processing as compared to other object detectors. YOLOv3-tiny is naturally suited for embedded computer vision. It has high precision values than many other algorithms, can be seen the image below.



(a) Picasso Dataset precision-recall curves.

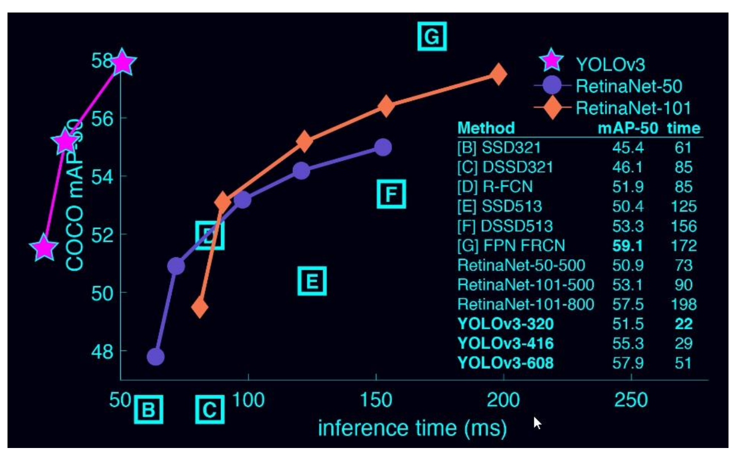


Fig. (6) Precision - recall curve for object detectors Fig. (7) mAP vs Inference time curve for object detectors

Approach

- Studied basics of object detection and classification using YOLO algorithm
- Developed a python code for classifying objects in images and videos using state-of-the-art **yolov3 algorithm** pretrained on COCO dataset
- Executed the training of the model on custom dataset, used ClearML platform for visualization purpose
- Converted the code into C++ so that it can be deployed on hardware since python code cannot be implemented on the hardware available

Tools & Packages used

- Python
- OpenCV
- C++
- NumPy
- Deep Learning
- YOLO Algorithm
- ClearML Platform

Working of YOLO algorithm

The YOLO algorithm is a real-time object detection system that processes images and predicts the location and class of objects in those images. It's an end-to-end neural network-based approach that uses a single convolutional network to predict both the object class probabilities and the bounding box coordinates. Here are the main steps in the YOLO algorithm:

1. Divide the input image into a grid: The input image is divided into a grid of cells. Each cell is responsible for predicting a certain number of bounding boxes.

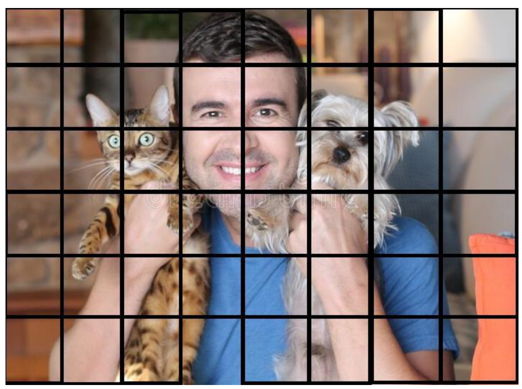
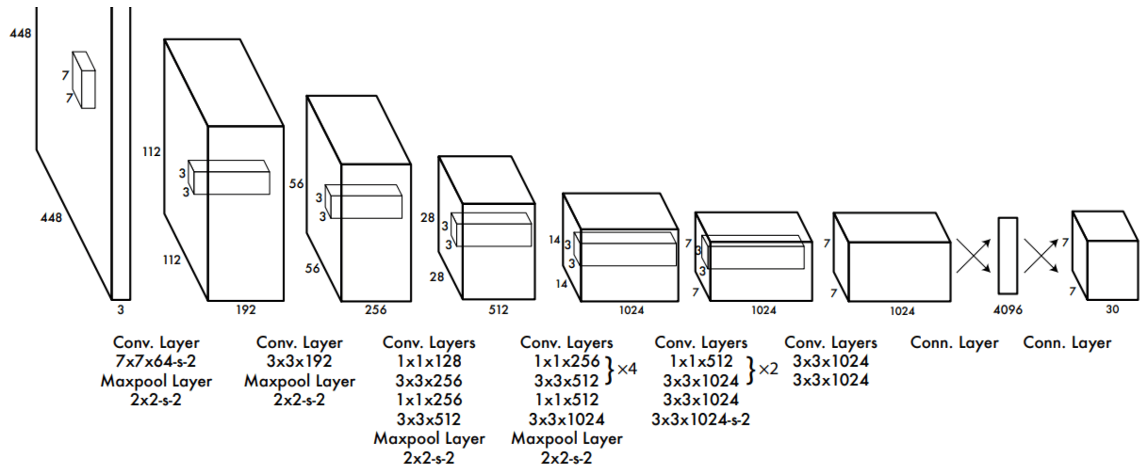


Fig. (8) Image Divided into an nxn grid

2. Predict the bounding boxes and class probabilities: For each cell in the grid, the YOLO algorithm predicts a fixed number of bounding boxes along with their corresponding class probabilities. Each bounding box has five attributes: x, y, width, height, and confidence. The (x,y) coordinates represent the center of the box relative to the cell, and the width and height represent the size of the box. The confidence score represents the probability that the box contains an object, and the class probabilities represent the probabilities that the object belongs to each class.



3. Calculate the confidence score for each bounding box: The YOLO algorithm calculates a confidence score for each bounding box using the object-ness score and the intersection over union (IoU) score. The object-ness score is the probability that the bounding box contains an object, and the IoU score measures the overlap between the predicted bounding box and the ground truth bounding box.

4. Apply non-max suppression: The YOLO algorithm applies non-max suppression to remove duplicate detections. This involves selecting the bounding box with the highest confidence score and removing any other boxes with high IoU scores with it.

5. Output the final detections: The YOLO algorithm outputs the final detections, which include the class label, the confidence score, and the bounding box coordinates.

X	Y	W	H	Confidence	Class scores of 80 classes

Fig. (9) Output vector of each cell in the image grid

Limitations of YOLO

YOLO introduces strict spatial limitations for bounding box predictions, as each grid cell is restricted to predicting only two boxes of a single class. This constraint hampers the model's ability to predict numerous nearby objects. Consequently, this model encounters difficulties in accurately detecting small objects that appear in clusters, such as group of cars.

The model struggles with generalization arises from its training to predict bounding boxes solely based on available data. It faces challenges when presented with objects exhibiting uncommon aspect ratios or configurations. Additionally, the model employs relatively coarse features for bounding box predictions due to the presence of multiple down sampling layers in its architecture, which reduce the resolution of the input image.

Furthermore, although the model is trained using a loss function that approximates detection performance, the treatment of errors is uniform for both small and large bounding boxes. While a small error in a large box is typically insignificant, a small error in a small box can significantly impact Intersection over Union (IOU) scores. The primary source of error in the model lies in incorrect localizations during object detection.

Hardware Implementation

The Nvidia Drive PX2 is a computer platform designed to facilitate the efficient development of automated and autonomous vehicles. With its integrated and discrete Pascal architecture GPUs, the Drive PX2 platform offers accelerated processing power, making it highly suitable for tasks involving deep neural networks (DNN), as well as sensor data fusion and processing. Its impressive computational performance of 8 TFLOPS or 24 DL TOPS contributes to its capability. The platform incorporates two Tegra X2 SoCs, each equipped with ARM v8 architecture CPUs featuring a total of 12 cores, comprising four ARM-A57 and two Denver cores. It also includes 8 GB LPDDR4 128-bit memory and an integrated GPU with Pascal architecture. Furthermore, the platform boasts an array of connectivity options, such as UART, CAN, LIN, FlexRay, USB 3.0 and 2.0, 1 and 10 Gbit Ethernet, HDMI, and 12 GSML camera ports, catering to the diverse needs of autonomous vehicle development.

Fig. (10) Nvidia Drive PX2 Hardware which was used for implementing algorithm

- **Hardware Specifications**

Total Chips	2 x Tegra X2 SoCs, 2 x Pascal MXM GPUs
CPU	12 core ARM CPU (in total)
CPU architecture	8 x A57, 4 x Denver (in total)
GPU architecture	Pascal (256 cores)
Total computational power	24 DL TOPS or 8 TFLOPS (32bit)
System memory	8 GB LPDDR4 for each Tegra X2 (50+ GB/s)
Graphics Memory	4 GB GDDR5 for each discrete GPU (80+ GB/s)
Total dissipated power (TDP)	80W

Reference: https://www.edi.lv/en/equipment/nvidia-drive-px2-computer-platform/

Results

Image 1

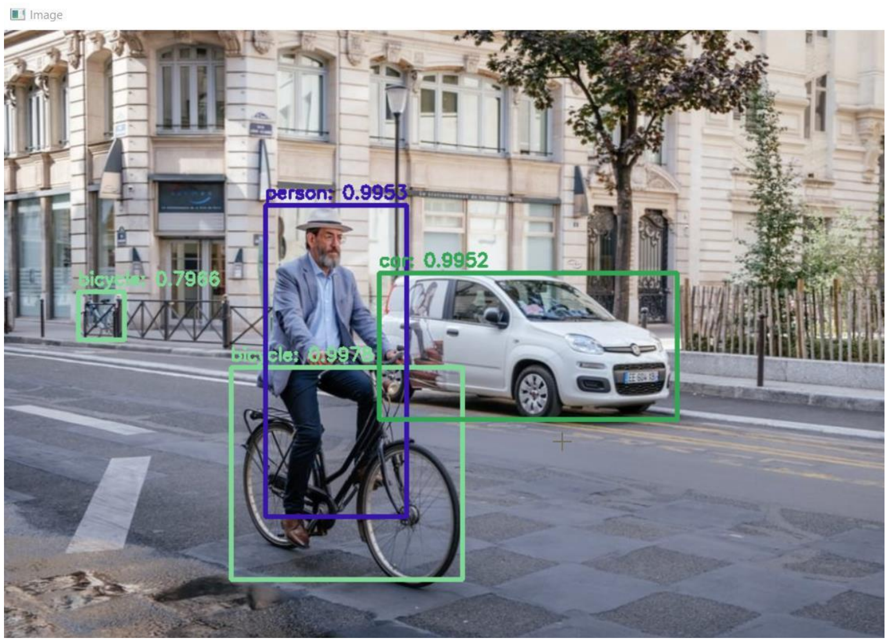
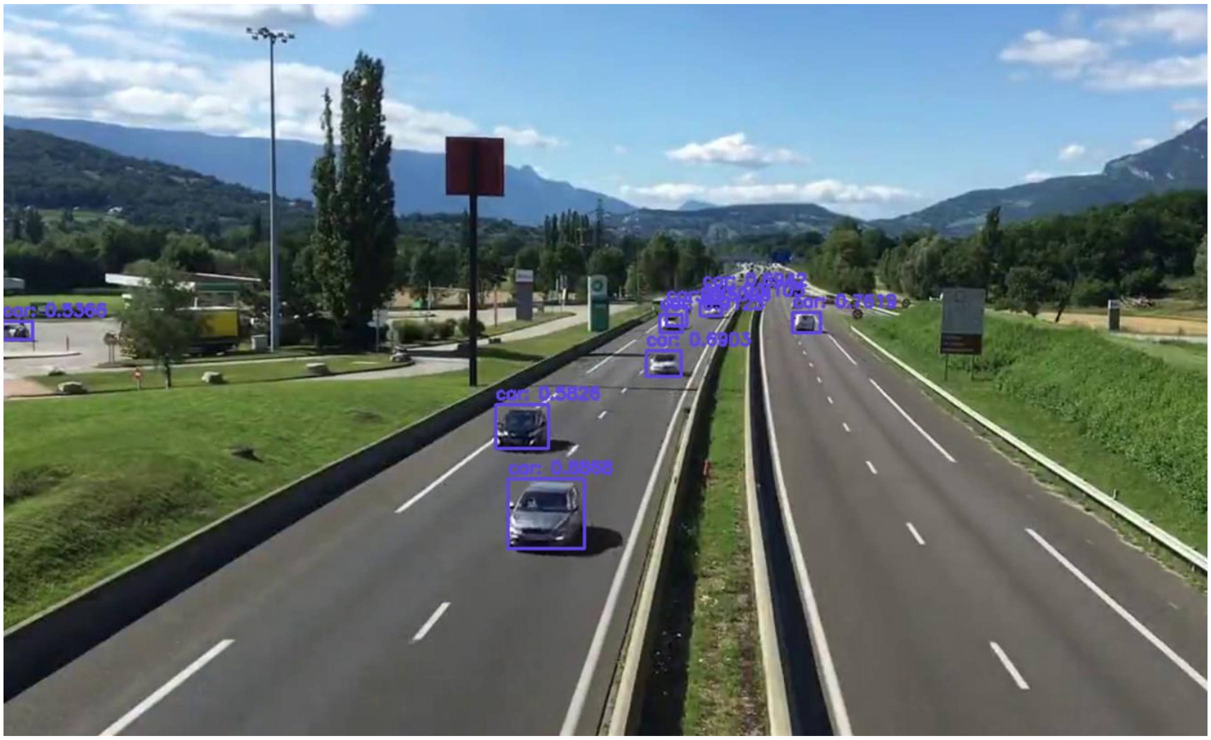
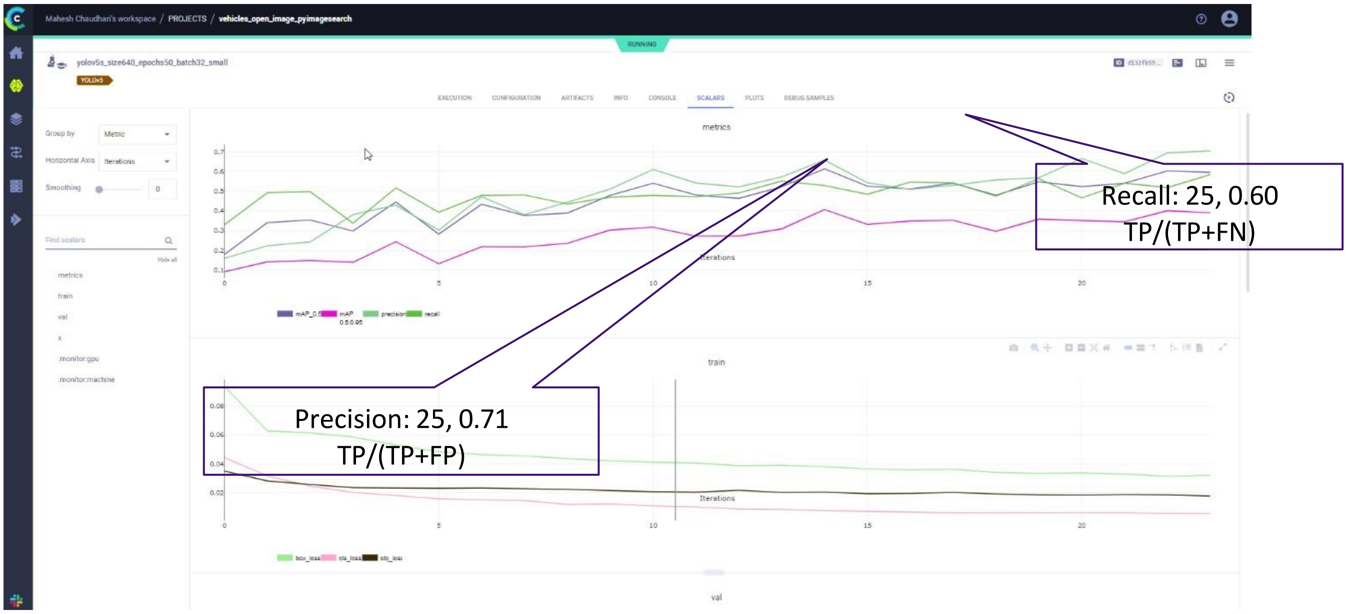


Image 2

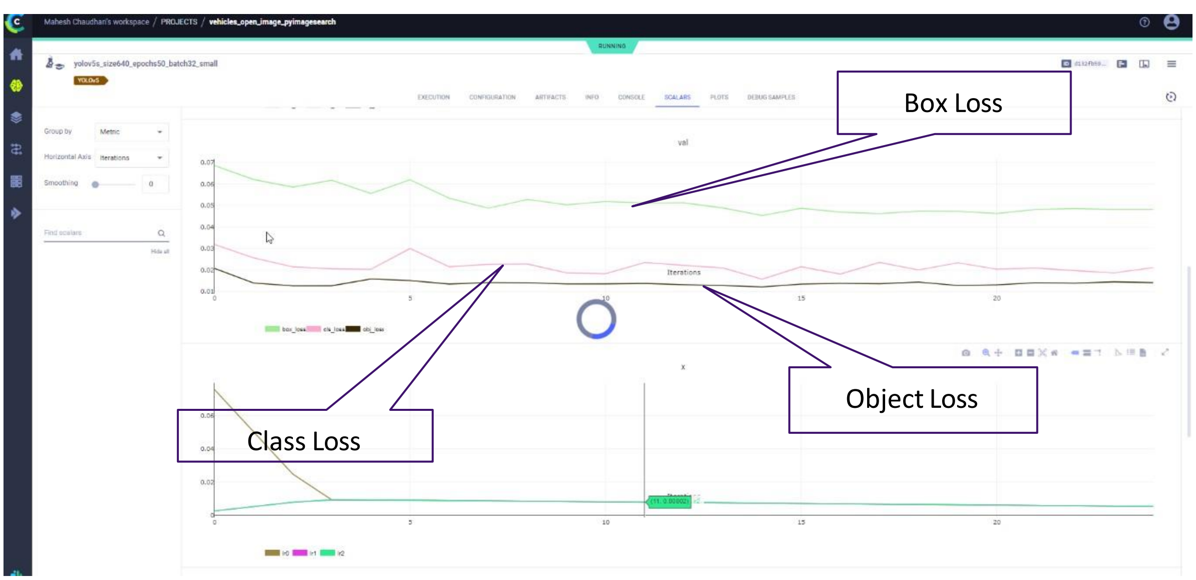
Video 1



Video 2



1. Ran 25 iterations to train the model
2. Achieved values of classification evaluation metrics' as precision = 71% and recall = 60%
3. Measured 3 loss metrics on validation dataset
4. Reduced the losses to following values: Box loss: 4.8%, Class Loss: 2%, Object Loss: 1.5%



Processing Time to execute the codes (Sec)	Laptop	Nvidia Drive PX2
Object Detection on a single image (Python)	1.3	0.52
Object Detection on a single image (CPP)	1.85	0.91
Object Detection on a video (Python) Video Length: 180 Seconds	679.3	551.8

Observations from the table above:

1. CPU on hardware takes less time to run the code than Laptop's CPU
2. Python code took less time than the CPP code to run
3. Frame wise video processing was comparatively slower than image processing for YOLO algorithm

Future Scope

The future scope for object detection in self-driving cars holds immense potential for advancements and enhancements. As technology progresses, we can expect several exciting developments in this field. Firstly, there will be a continuous focus on improving the accuracy and reliability of object detection algorithms. By leveraging deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), researchers and engineers will strive to achieve even higher precision in identifying and classifying objects in real-time scenarios. Additionally, the integration of multiple sensor modalities, such as cameras, LiDAR, radar, and ultrasonic sensors, will further enhance object detection capabilities. By combining the strengths of these sensors and employing sensor fusion techniques, self-driving cars can gather comprehensive and detailed information about the surrounding environment, leading to more robust and reliable object detection.

Another exciting area of future development is the exploration of real-time 3D object detection. By utilizing advanced techniques like point cloud processing and volumetric representations, self-driving cars will be able to perceive and understand the spatial aspects of the detected objects. This advancement can significantly improve the vehicle's ability to accurately estimate object distance, size, and orientation, thereby enhancing decision-making and overall safety. Furthermore, the advent of edge computing and onboard AI processing will enable faster and more efficient object detection in self-driving cars. By performing complex computations directly within the vehicle, latency can be minimized, allowing for quicker response times in critical situations. This decentralized approach also reduces the dependence on cloud computing, ensuring continuous object detection capabilities even in areas with limited or no internet connectivity.

Lastly, the future of object detection in self-driving cars will involve continuous refinement and adaptation through machine learning. As self-driving car fleets gather vast amounts of real-world data, this data can be leveraged to improve and fine-tune object detection algorithms. Continuous learning and updates will enable the detection system to adapt to new objects, scenarios, and environmental conditions, ultimately enhancing the safety and reliability of self-driving cars. In summary, the future scope for object detection in self-driving cars encompasses advancements in accuracy, sensor fusion, real-time 3D detection, edge computing, and continuous learning. These developments will contribute to safer and more capable autonomous vehicles, paving the way for a future where self-driving cars can navigate complex environments with precision and confidence.

Summary and Conclusion

The project aimed to deploy the YOLOv3 (You Only Look Once) object detection algorithm on Nvidia Drive PX2 hardware and evaluate its performance. The YOLOv3 algorithm is known for its real-time object detection capabilities, making it suitable for various applications, including autonomous vehicles and surveillance systems. To accomplish the goal, the project followed a systematic approach. It involved setting up the Nvidia Drive PX2 hardware, installing the necessary software libraries, and integrating the YOLOv3 algorithm into the system. The project team then conducted extensive testing and benchmarking to measure the algorithm's performance in terms of detection accuracy, speed, and resource utilization. The deployed YOLOv3 algorithm demonstrated good performance in object detection tasks, accurately identifying and localizing various objects in real-time scenarios. Additionally, the algorithm efficiently utilized the computational resources of the Nvidia Drive PX2 hardware, further affirming its effectiveness for demanding visual processing tasks.

In conclusion, the project successfully deployed the YOLOv3 object detection algorithm on Nvidia Drive PX2 hardware and assessed its performance. The combination of the algorithm's real-time capabilities and the computational power of the hardware proved to be a formidable solution for object detection tasks. The project's findings contribute to the field of computer vision and pave the way for the integration of advanced object detection systems into various industries, such as autonomous driving and surveillance. Further optimization and customization of the algorithm can unlock even more potential, leading to improved performance and expanded applications in the future.

References

1. Wang, L., Zhou, H., Bian, C., Jiang, K., & Cheng, X. (2022). Hardware Acceleration and Implementation of YOLOX-s for On-Orbit FPGA. *Electronics*, *11*(21), 3473.
2. Sharma, M., Rahul, R., Madhusudan, S., Deepu, S. P., & Sumam, D. S. (2021, December). Hardware accelerator for object detection using tiny YOLO-v3. In *2021 IEEE 18th India Council International Conference (INDICON)* (pp. 1-6). IEEE.
3. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
4. Geethapriya, S., Duraimurugan, N., & Chokkalingam, S. P. (2019). Real-time object detection with Yolo. *International Journal of Engineering and Advanced Technology (IJEAT)*, *8*(3S).
5. Ahmad, T., Ma, Y., Yahya, M., Ahmad, B., Nazir, S., & Haq, A. U. (2020). Object detection through modified YOLO neural network. *Scientific Programming*, *2020*, 1-10.
6. Qian, R., Lai, X., & Li, X. (2022). 3d object detection for autonomous driving: a survey. *Pattern Recognition*, *130*, 108796.
7. Gohil, P., Plöger, P. G., & Hinkenjann, A. (2022, August). Quantitative Analysis of Object Detectors for Autonomous Driving and Autonomous Parking. In *2022 26th International Conference on Pattern Recognition (ICPR)* (pp. 3652-3659). IEEE.