People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University Echahid Echeikh Larbi Tebessi – Tébessa
Faculty of Exact Sciences and Natural and Life Sciences
Department: Mathematics and Computer Science

Final dissertation for obtaining the MASTER degree
Field: Mathematics and Computer Science
Stream: Computer Science
Option: System Information

Improved Vehicles Detection in Aerial Images for 3D City Modeling

Theme Submitted by:
Chergui Kheir Eddine

Before the jury:

Dr. Bennour Akram	*University of Larbi Tébessa*	*President*
Dr. Douadi Kheir Eddine	*University of Larbi Tébessa*	*Examiner*
Pr. Bendjenna Hakim	*University of Larbi Tébessa*	*Supervisor*
Pr. Meraoumia Abdallah	*University of Larbi Tébessa*	*Co-Supervisor*

Date of defense: 06/ 06/2023

Abstract

Improved Vehicle Detection in Aerial Images for 3D City Modeling

CHERGUI Kheir Eddine

High-quality 3D city models serve as fundamental infrastructure for smart cities and various applications. However, the presence of moving objects, especially Vehicles, poses a significant challenge to the automated generation of these models. Moving targets introduce instability in density matching and aerial triangulation processes, which can adversely affect the overall quality of the final models.

To address this challenge and faithfully represent the dynamic environment of cities using discrete still captures, we propose a pre-processing procedure for optical imagery. This procedure focuses on detecting problematic objects, specifically vehicles, to pass them later to the elimination phase and to ensures the generation of accurate and precise 3D city models without the inconveniences and distortions caused by these objects.

This research contributes to the fields of Image Processing and Computer Vision by addressing the Object Detection key aspect. We design a modern, fresh and more flexible Deep Learning-based method to detect moving vehicles that may disrupt stereo-vision during the 3D extrusion process. The detection models showed a promising result of 98% mAP and 94% mAP, this may not seem impressive but the flexibility and new features of these models make up for the relatively average accuracy.

By mitigating the impact of moving vehicles on the 3D city generation process, our approach enhances the overall accuracy and realism of the resulting models.

Key words: 3D City Modeling, Deep Learning, Aerial triangulation process, Vehicle Detection, Photogrammetry, mAP.

ملخص

كشف المركبات المحسن في الصور الجوية لبناء المدن ثلاثية الأبعاد

شرقي خير الدين

تعتبر نماذج المدن ثلاثية الأبعاد عالية الجودة أساسًا للبنية التحتية الأساسية للمدن الذكية والتطبيقات المتنوعة. ومع ذلك، يشكل وجود الأجسام المتحركة، وخاصة المركبات، تحديًا كبيرًا يعيق توليد هذه النماذج بشكل تلقائي. تضيف الأهداف المتحركة عاملًا غير ثابت في عمليات مطابقة الكثافة والتثليث الجوي، مما يؤثر سلبًا على جودة النماذج النهائية.

للتغلب على هذا التحدي وتمثيل البيئة الديناميكية للمدن باستخدام لقطات ثابتة فصلية، نقترح إجراءات مسبقة لمعالجة الصور البصرية. تركز هذه الإجراءات على اكتشاف الأجسام الغير مرغوب فيها، وبالتحديد المركبات، ومن ثم إزالتها لضمان إنشاء نماذج مدن ثلاثية الأبعاد دقيقة ومتقنة دون الإزعاج والتشويش الناتج عن هذه الأجسام.

يساهم هذا البحث في مجال معالجة الصور ورؤية الحاسوب من خلال التركيز على جانب الكشف عن الأجسام الغير مرغوب فيها. نقوم بتصميم طريقة حديثة قائمة على تقنيات التعلم العميق لاكتشاف المركبات المتحركة التي قد تؤثر على الرؤية الاستريو أثناء عملية الانبثاق ثلاثية الأبعاد. أظهرت نماذج الكشف نتائج واعدة بمعدل دقة متوسط يصل إلى 98% و 94% لنموذجينا، حيث توازنت الدقة المتوسطة بالمرونة والميزات الجديدة لهذه النماذج. من خلال تخفيف تأثير السيارات المتحركة على عملية إنشاء مدن ثلاثية الأبعاد، يعزز نهجنا الدقة العامة والواقعية للنماذج الناتجة.

الكلمات المفتاحية: نماذج ثلاثية الأبعاد، التعلم العميق، التثليث الجوي، كشف المركبات، المسح التصويري، متوسط الدقة.

Résumé

Amélioration de la détection des véhicules dans les images aériennes pour la modélisation 3D des villes

CHERGUI Kheir Eddine

Les modèles urbains 3D de haute qualité servent d'infrastructure fondamentale pour les villes intelligentes et diverses applications. Cependant, la présence d'objets en mouvement, en particulier les véhicules, constitue un défi majeur pour la génération automatisée de ces modèles. Les cibles mobiles introduisent une instabilité dans les processus d'appariement de densité et de triangulation aérienne, ce qui peut affecter négativement la qualité globale des modèles finaux.

Pour relever ce défi et représenter fidèlement l'environnement dynamique des villes à partir de captures d'images fixes, nous proposons une procédure de prétraitement pour les images optiques. Cette procédure se concentre sur la détection d'objets problématiques, en particulier les véhicules, afin de les éliminer ultérieurement et d'assurer la génération de modèles urbains 3D précis, sans les inconvénients et les distorsions causés par ces objets.

Cette recherche contribue aux domaines du traitement d'images et de la vision par ordinateur en abordant l'aspect clé de la détection d'objets. Nous concevons une méthode moderne et flexible basée sur l'apprentissage profond pour détecter les véhicules en mouvement pouvant perturber la vision stéréo lors du processus d'extrusion 3D. Les modèles de détection ont montré un résultat prometteur de 98% mAP et 94% mAP, ce qui peut ne pas sembler impressionnant, mais la flexibilité et les nouvelles fonctionnalités de ces modèles compensent la précision relativement moyenne.

En atténuant l'impact des voitures en mouvement sur le processus de génération de villes 3D, notre approche améliore la précision globale et le réalisme des modèles obtenus.

Mots clés : Modélisation 3D des ville, Apprentissage profond, Processus de triangulation aérienne, Détection de véhicules, Photogrammétrie, mAP.

Acknowledgment

In the name of Allah, our Lord and Creator, Who has blessed us with the gift of reasoning and the pursuit of knowledge, we begin by expressing our gratitude.

I would like to extend my heartfelt appreciation to my supervisors, Pr. BENDJENNA Hakim and Pr. MERAOUMIA Abdallah for their unwavering trust and patience. Their insightful guidance, advice, and commitment were instrumental in shaping this thesis.

I also wish to express my deep gratitude to the esteemed members of the jury Dr. BENNOUR Akram and Dr. DAOUADI Kheir Eddine for agreeing to evaluate and judge this humble work.

I am indebted to CHERGUI Abdelmalek, my brother, who was the first to encourage me to explore this field.

Finally, I must express my profound gratitude to my family. Their unwavering support, continuous encouragement, and countless sacrifices throughout our academic journey made this accomplishment possible.

“To my beloved parents, your encouragement is the

only thing keeps me going.

To my two sisters and my bigger brother

Thanks for always being there for me.

To all my friends, the ones near my heart but far from my tongue.

To all those who had impact in my carrier.

Thank you, I dedicate this work to you”

Table of Content

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
DL	Deep Learning
DRL	Deep Reinforcement Learning
FPN	Feature Pyramid Network
GAN	Generative Adversarial Network
GIS	Geographic Information System
GPT	Generative Pre-trained Transformer
GRU	Gated Recurrent Unit
IoU	Intersection over Union
LiDAR	Light Detection and Ranging
LSTM	Long Short-Term Memory
mAP	Mean Average Precision
MS COCO	Microsoft Common Object Context
MVS	Multi-View Stereo
PPO	Proximal Policy Optimization
R-CNN	Region based Convolutional Neural Network
ReLU	Rectified Linear Unit
R-FCN	Region based Full Convolutional Network
RNN	Recurrent Neural Network
SAME	Structure from Motion
SSD	Single-Shot Detector
VEDAI	Vehicle Detection in Aerial Images
VGG	Visual Geometry Group
VOC	Volatile Organic Compounds
YOLO	You Only Look Once

Introduction and Research Background

3D city modeling is an important field of research that has gained significant attention in recent years. It involves the creation of three-dimensional digital models of urban areas using various data. Accurate 3D city models can help in visualizing and analyzing urban environments, identifying potential risks and hazards, and planning future development projects. Additionally, 3D city models can provide a realistic representation of urban areas that can be used in video games, movies, and virtual reality applications. Therefore, the development of accurate and efficient methods for 3D city modeling is crucial to support decision-making processes and improve the quality of life in urban areas[1].

However, obtaining accurate 3D models of urban areas is a challenging task due to various factors such as the complexity of urban environments, occlusions caused by buildings and other structures, and the presence of dynamic objects such as vehicles and pedestrians.

One of the most widely used data sources for 3D city modeling is aerial imagery, which is obtained by capturing images of urban areas from airplanes, drones and satellites. Aerial images provide a top-down view of urban areas, making them a valuable source of data for 3D city modeling. However, using aerial images for 3D city modeling poses several challenges[2].

Despite these challenges, aerial images remain an essential data source for 3D city modeling due to their availability, affordability, and high-resolution capabilities. Advancements in computer vision and machine learning techniques have also enabled researchers to develop more efficient and accurate methods for 3D city modeling using aerial images.

Deep into aerial images, the presence of vehicles in these images stay in the way of creating accurate and detailed three-dimensional representations of urban environments and can significantly impact the quality and reliability of the resulting models.

Vehicles present several challenges in the context of 3D city modeling. Firstly, vehicles often obstruct the view of important urban features such as buildings, roads, and infrastructure. This occlusion can lead to incomplete or inaccurate representations of the city's geometry and layout. Furthermore, vehicles can introduce noise, shadows, and artifacts into the aerial images, which amplifies the complexity of the modeling process[3].

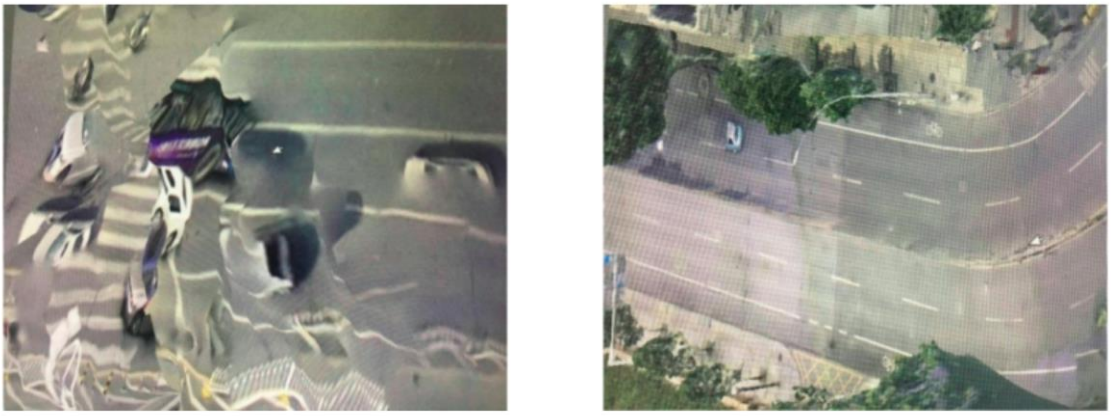

Figure 1 : Rendering Degradation on the model texture

Figure 2 : Distortion on the Geometric models

Moreover, the accurate removal of vehicles from these images is crucial for generating clean and precise 3D city models. Vehicle removal techniques are necessary to eliminate the unwanted visual artifacts and occlusions caused by vehicles, allowing for a clearer view of the underlying urban structures[3].

Addressing the problem of vehicles in aerial images for 3D city modeling requires the development of robust vehicle detection and removal algorithms. These algorithms need to be capable of accurately identifying vehicles, estimating their pose and dimensions, and effectively removing them from the images while preserving the integrity of the remaining urban elements.

By solving the problem of vehicle detection and removal in aerial images, we can enhance the quality, completeness, and accuracy of 3D city models. This, in turn, enables various applications such as urban planning, architecture, environmental simulations, and transportation analysis to benefit from more reliable and detailed representations of the urban environment.

In various situations, it is relatively straightforward to create individual building models and generate a generic street image by manually adding other objects. However, when it comes to modeling an entire city, a significant dilemma arises. The process becomes exceedingly monotonous, repetitive, and demands substantial time and financial resources. Consequently, an automated procedural approach to 3D modeling becomes the only viable solution[4].

Our Objective

We can present our objectives on the following notes:

- Train an effective vehicle detection algorithm for accurately identifying and localizing vehicles in aerial images.
- Evaluate the proposed vehicle detection methods using benchmark datasets and compare their performance against state-of-the-art approaches.
- Assess the impact of vehicle detection on the quality and accuracy of 3D city models.

- Investigate the practical applications and potential benefits of improved vehicle detection techniques in the context of 3D city modeling.

On this thesis we present 3 main chapters;

The first will contain a review of literature on 3D city modeling as well as its techniques and how important it is in some applications, then concluding by identifying the gaps and areas that require further investigation.

On chapter 2 we will start by checking an overview of object detection with deep learning, seeing the principles, applications, and types of models. State the challenges in training deep learning models and techniques and how to overcome them. And finally, the integration of deep learning in our project (vehicle detection in aerial images.)

Chapter 3 talks about Vehicle Detection in Aerial Images where we discuss the methodology for detecting vehicles in aerial images. Present the dataset and preprocessing steps as well as the deep learning model architecture we used. Additionally, the training process and evaluation metrics. Then comparing the experimental results with the state of the art. Last but not least we will discuss the limitations and areas of improvements.

In the conclusion we will summarize the main findings. Contributions of the thesis to 3D city modeling and deep learning and limitations of the thesis and suggestions for future research.

Chapter I : 3D City Modeling

Introduction

Before bringing out the source of 3D city models distortions and miss-rendering issue, we will try in this chapter to place the problem in its context by giving an overview of urban 3D modelling technologies while showing some of their pros and cons.

1. What are 3D city models

3D city modeling involves creating digital representations of urban environments in three dimensions. It aims to provide accurate and detailed virtual cityscapes for visualization, analysis, and simulation purposes.

It integrates various data sources and technologies like aerial imagery, LiDAR data, and photogrammetry to reconstruct the geometry and appearance of buildings and other urban features. The resulting 3D models enable immersive exploration and analysis of the urban environment.

3D city modeling is valuable for urban planning, architecture, transportation, environmental studies, and more, allowing stakeholders to visualize, assess, and make informed decisions about urban development and infrastructure projects[5], [6].

2. Modeling Methods Categorization

The categorization of 3D modeling methods can be based on various criteria, including the application domain, level of abstraction, representation mode, or visualization effects. However, in many cases, 3D modeling methods are primarily categorized either based on their automation level or the data input techniques used[7].

2.1. Automation basis

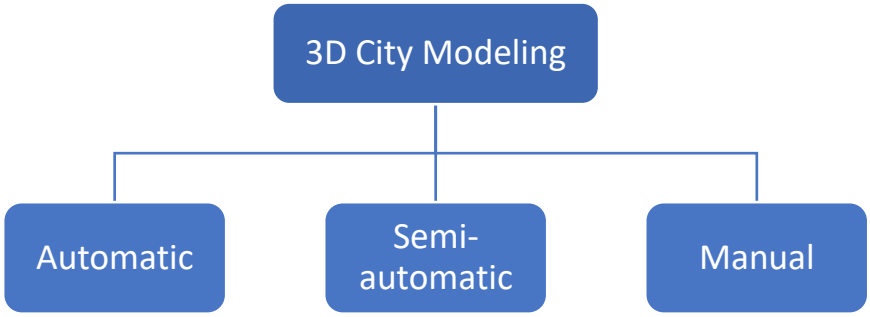

Figure I-1: Automation Basis

- Automatic methods rely heavily on computational algorithms and techniques to generate 3D models without significant human intervention. They are efficient and can handle large datasets, but may have limitations in capturing intricate details.
- Semi-automatic methods combine automated algorithms with user interaction, allowing users to provide input or guidance during the modeling process. These methods strike a balance between automation and user control.
- Manual methods involve extensive human involvement and expertise, with skilled artists or designers manually creating 3D models. Manual methods offer high levels of customization and attention to detail but are time-consuming.

2.2. Data input techniques basis

Data input technique-based categorization focuses on the primary data sources used for 3D modeling. Image-based methods use 2D images, often obtained from aerial or street-level photography, as the main data source. These methods leverage computer vision algorithms to extract 3D information from overlapping images. LiDAR-based methods utilize data captured by LiDAR sensors, which measure the distance to objects using laser pulses. LiDAR data enables the creation of highly accurate point clouds for 3D modeling. GIS-based methods integrate Geographic Information System (GIS) data with other data sources such as satellite imagery or LiDAR to generate 3D models. GIS data provides geospatial information such as elevation, land use, and infrastructure data.

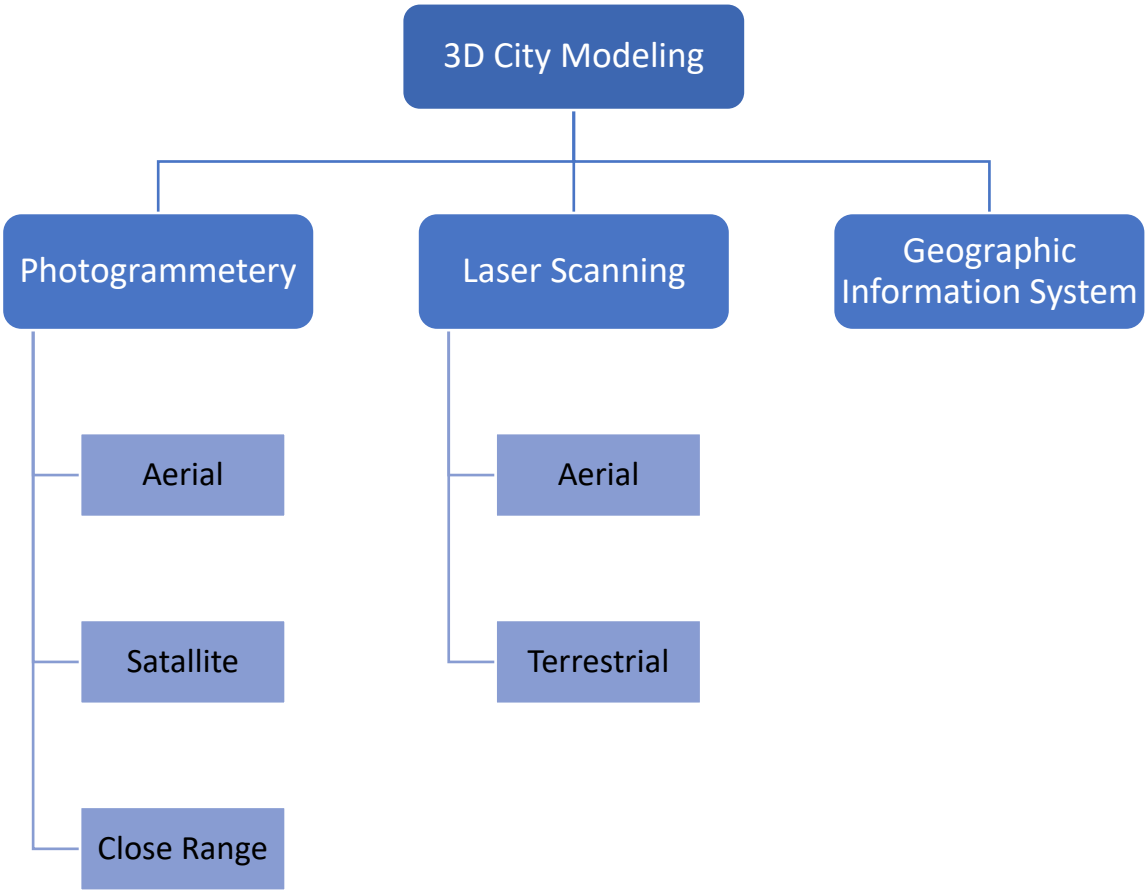

Figure I-2: Data Input Techniques Basis

2.3. 3D Modeling Techniques

A comprehensive review of the literature on 3D city modeling reveals various approaches and techniques used in the field. These approaches can be categorized into several key areas:

a. Photogrammetry-based Approaches:

- Structure from Motion (Same): This technique uses multiple overlapping aerial images to reconstruct the 3D geometry of the city by identifying corresponding features and estimating camera positions.
- Multi-View Stereo (MVS): MVS algorithms combine information from multiple images to generate dense 3D point clouds, which are then used to create detailed 3D models.

b. **LiDAR-based Approaches:** Light Detection and Ranging (LiDAR): LiDAR sensors emit laser pulses and measure their return time to calculate accurate 3D point cloud representations of the city. LiDAR data is often used in combination with aerial imagery for 3D city modeling.
c. **Procedural Modeling:** Procedural modeling techniques use algorithms and rule-based systems to generate 3D city models automatically. These approaches define rules for the generation of buildings, roads, and other urban elements, enabling the creation of large-scale city models efficiently.
d. **Image-based Modeling:** Image-based modeling techniques rely on a collection of images from different viewpoints to reconstruct the 3D geometry of the city. These approaches use feature matching, camera calibration, and bundle adjustment algorithms to estimate the 3D structure.
e. **Point Cloud Processing:** Point cloud processing techniques involve filtering, segmentation, and classification of LiDAR or dense point clouds to extract meaningful urban features. These features can include buildings, vegetation, roads, and other objects.
f. **Semantic Modeling:** Semantic modeling aims to assign meaningful labels or semantic information to the elements in 3D city models. This enables the representation of not only geometric properties but also functional and contextual information of urban objects.
g. **Hybrid Approaches:** Hybrid approaches combine multiple data sources, such as aerial imagery, LiDAR data, and GIS data, to create more accurate and detailed 3D city models. These approaches leverage the complementary strengths of different data sources for improved modeling results.
h. **Real-time and Interactive Modeling:** Real-time and interactive modeling techniques focus on generating 3D city models in real-time or near real-time for applications like virtual reality, augmented reality, and gaming. These approaches often prioritize efficiency and user interactivity.

The literature review should provide an in-depth understanding of the strengths, limitations, and applications of each approach. It should also identify the

gaps in the existing research and highlight the areas that require further investigation for advancing the field of 3D city modeling[6].

2.4. Comparison between modeling techniques

We can summarize the advantages and limitations of the techniques mentioned above on the following table:

Technique	Advantages	Limitations
Photogrammetry-based Approaches	Cost-effective, detailed 3D models, texture information.	Sensitive to image quality, struggles with complex environments, time-consuming manual processes.
LiDAR-based Approaches	High precision, reliable in various conditions, detailed geometric features.	Expensive, varying point cloud density, limited texture information.
Procedural Modeling	Efficient for large-scale models, parametric control, simulation of urban scenarios.	Lack of detail, simplified representations, time-consuming rule creation.
Image-based Modeling	Cost-effective, captures visual appearance, advancements in computer vision.	Sensitivity to image quality, struggles with depth estimation, computationally demanding.
Point Cloud Processing	Detailed analysis, precise geometric properties, integration with other data.	Noise removal, computational intensity, data acquisition limitations.
Semantic Modeling	Contextual information, advanced analysis, enhanced understanding.	Data annotation challenges, standardization, scalability.
Hybrid Approaches	Combined benefits of multiple data sources, accurate and detailed models.	Data fusion challenges, increased cost and resource requirements.
Real-time and Interactive Modeling	Immediate visual feedback, interactivity, immersive experiences.	Sacrifices accuracy for efficiency, hardware requirements, simplified representations.

Table 1: Advantages and Limitations of 3D City Modeling Techniques

3. Applications of 3D City Modeling

Accurate 3D city models contribute to improved planning, decision-making, and analysis in urban environments. As it plays a vital role in various applications as we will explain[8].

a. **Urban Planning and Design:** Provide urban planners, architects, and designers with a realistic representation of the urban landscape. They enable better visualization and understanding of existing structures, infrastructure, and spatial relationships. This information aids in the development of sustainable urban plans, efficient transportation systems, and optimized land use[9].
b. **Disaster Management and Emergency Response:** During natural disasters or emergencies, accurate 3D city models assist in disaster preparedness, response planning, and resource allocation. They facilitate the identification of vulnerable areas, evacuation routes, and potential hazards. These models also support simulations and predictive analysis, helping authorities mitigate risks and improve disaster management strategies.
c. **Environmental Analysis:** They are crucial for assessing environmental impacts and conducting simulations. As they enable the evaluation of sunlight exposure, wind patterns, and energy consumption in urban areas. These models aid in optimizing renewable energy installations, reducing carbon emissions, and enhancing the overall environmental sustainability of cities[10],[11].
d. **Infrastructure Development and Management:** Provide insights into the existing infrastructure network. Which helps to identify areas requiring infrastructure upgrades, optimizing utility networks, and planning for future expansions. These models also assist in asset management, maintenance planning, and coordination among different stakeholders.
e. **Visualization and Virtual Reality:** Enhance visualization and immersive experiences for various applications. They enable virtual tours, augmented reality applications, and interactive simulations. These models are invaluable

for marketing, tourism, education, and cultural heritage preservation, providing a realistic and engaging representation of urban spaces.

f. **Simulation and Gaming:** Accurate 3D city models serve as a foundation for urban simulations and gaming applications. They support traffic simulations, crowd behavior analysis, and urban simulations for research purposes. In the gaming industry, these models form the basis for creating realistic virtual worlds, enabling engaging and interactive gaming experiences.

4. Current research gaps in the field

The existing research on vehicle detection and removal in aerial images for 3D city modeling has made significant progress, but there are still several gaps that require further investigation. One area that requires attention is the development of improved vehicle detection algorithms. Current algorithms exhibit varying performance and struggle with challenging scenarios, such as crowded urban environments and occlusion. Further research is needed to develop more robust and accurate algorithms, potentially utilizing deep learning techniques and advanced feature extraction methods[12].

Another important gap is the enhancement of vehicle removal techniques. While existing methods show promise, there is room for improvement in preserving the integrity and quality of the background scene. Researchers need to focus on developing advanced techniques that effectively remove vehicles while maintaining visual coherence and realism in the aerial images. This involves addressing challenges such as accurate inpainting, texture blending, and handling complex backgrounds.

The availability of diverse and comprehensive datasets is crucial for evaluating and comparing different vehicle detection and removal methods. There is a need for large-scale datasets that encompass various urban environments, vehicle types, and occlusion scenarios. Establishing standardized evaluation metrics and benchmarks would enable fair comparison and facilitate advancements in the field.

Real-time and scalable solutions are highly desirable for practical applications. Further research should focus on developing efficient algorithms and techniques that

can handle large-scale aerial images and process them in real-time or near real-time. This involves exploring parallel processing, optimization strategies, and hardware acceleration to achieve faster and more scalable solutions.

Integrating vehicle detection and removal techniques seamlessly with 3D city modeling frameworks is an important area of research. Investigating methods to automatically update 3D city models based on the detected and removed vehicles would enable dynamic and accurate representations of urban environments. This includes addressing challenges related to data fusion, registration, and maintaining consistency between the 3D models and the underlying aerial images.

Additionally, researchers should consider application-specific considerations. Different applications may have specific requirements and challenges in vehicle detection and removal. Tailored solutions are needed for applications such as urban planning, traffic management, or environmental analysis. Understanding these specific considerations and developing specialized algorithms and techniques will be crucial in meeting the requirements of these applications.

Addressing these research gaps will contribute to the advancement of vehicle detection and removal techniques in aerial images for 3D city modeling. It will lead to more accurate and realistic representations of urban environments, benefiting various fields such as urban planning, disaster management, environmental analysis, and virtual reality applications.

Conclusion

In this chapter, we mentioned the main goal of the research, which is the 3D city modelling, starting by giving definitions and literature. Next, we talked about the main criteria in 3D city modelling categorization while showing the pros and the cons of each one. Furthermore, we mention some of its applications and Research Gaps.

Chapter II : Object Detection with Deep Learning

Introduction

After getting a clear view of 3D city modeling, we will now take our research to the deep learning part where we will understand the principles and applications of DL. As well as its different types and which one we chose to achieve our goal. Additionally, we will further explain object detection and see a brief performance comparison between detection models

1. Overview of deep learning

Deep learning is a subfield of machine learning that focuses on training artificial neural networks to learn and make predictions or decisions. It has gained significant attention and achieved remarkable success in various domains. Deep learning has revolutionized the field of artificial intelligence by enabling models to automatically learn and extract meaningful representations from complex data[13].

1.1. History and Development

Deep learning has its roots in the field of artificial neural networks, which dates back to the 1940s. However, it wasn't until the late 2000s and early 2010s that deep learning gained widespread popularity due to advancements in computational power, availability of large datasets, and algorithmic innovations. Breakthroughs such as the AlexNet architecture in 2012, which won the ImageNet[14] competition, demonstrated the power of deep learning in computer vision and triggered the deep learning revolution.

1.2. Principles of deep learning

The key principle underlying deep learning is the use of deep neural networks with multiple layers. These networks are composed of interconnected nodes, called artificial neurons or units, organized into layers. Each neuron performs a simple computation on its inputs and passes the result to the next layer. DL models learn by adjusting the weights and biases associated with these connections to minimize the difference between predicted and actual outputs. This

process, known as training, is typically performed using large datasets and optimization techniques such as stochastic gradient descent[15].

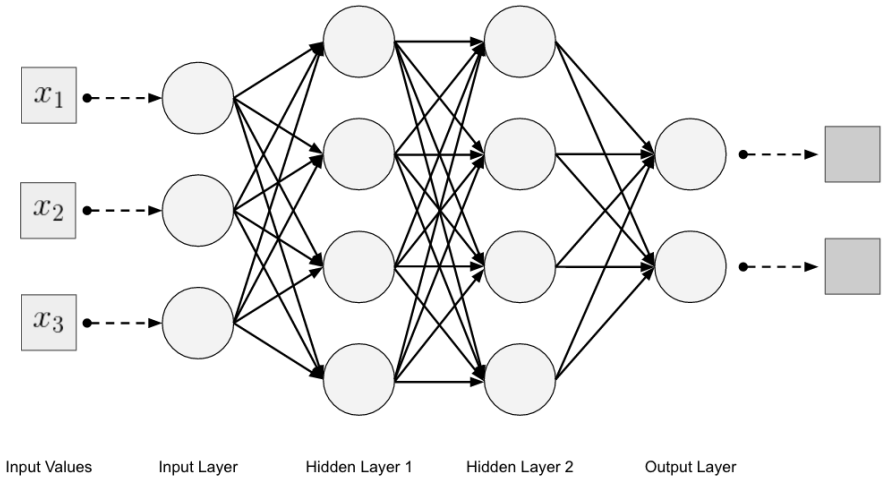

Figure II-1 : Major architecture of deep learning [16]

1.3. Application fields

Deep learning has found applications in various domains. In computer vision, deep learning models have achieved state-of-the-art performance in tasks such as object detection, image classification, and image segmentation. Natural language processing has also benefited from deep learning, with applications like machine translation, sentiment analysis, and text generation. Speech recognition systems, recommender systems, autonomous vehicles, and medical diagnosis are other areas where deep learning has made significant contributions[17].

Deep learning's success stems from its ability to learn intricate patterns, aided by hierarchical representations, and advancements in parallel computing with high-performance GPUs. Large labeled datasets have further fueled its progress. Challenges include the need for substantial computational resources, overfitting, and interpretability. Nevertheless, deep learning holds vast potential for complex problem-solving and driving AI advancements.

2. Deep learning types

There are various types of deep learning models, each designed to tackle different types of problems and data structures.

2.1. Convolutional Neural Networks (CNNs)

CNNs are primarily used for computer vision tasks and excel at image classification, object detection, and image segmentation. They are composed of convolutional layers that apply filters to extract spatial features from input images, followed by pooling layers to downsample the feature maps[18]. CNNs have revolutionized computer vision and achieved remarkable performance in tasks like image recognition. CNN will be the type we are going to use to achieve our desired goal.

2.2. Recurrent Neural Networks (RNNs)

RNNs are suitable for sequential data processing tasks, such as natural language processing and speech recognition. They have recurrent connections that allow information to persist across time steps, making them capable of capturing temporal dependencies. RNNs can process variable-length sequences and have been extended with variants like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) to address the vanishing gradient problem and improve performance on long-term dependencies.

2.3. Generative Adversarial Networks (GANs)

GANs consist of a generator network and a discriminator network that compete against each other[19]. GANs are used for generative modeling, where they learn to generate realistic synthetic samples that resemble the training data. GANs have been successfully applied to tasks like image synthesis, image-to-image translation, and data augmentation.

2.4. Autoencoders

Autoencoders are unsupervised learning models that aim to learn efficient data representations by reconstructing the input from a compressed latent representation. They consist of an encoder network that maps the input to a lower-dimensional latent space and a decoder network that reconstructs the input from the latent representation. Autoencoders have applications in tasks like dimensionality reduction, anomaly detection, and denoising.

2.5. Transformers

Transformers have gained popularity in natural language processing tasks, especially in machine translation and language understanding tasks. Transformers employ a self-attention mechanism that allows them to capture dependencies between different positions in the input sequence. They have shown superior performance in tasks that involve long-range dependencies and have become the backbone of many state-of-the-art language models, such as the Transformer-based models like BERT and GPT.

2.6. Deep Reinforcement Learning (DRL)

DRL combines deep learning with reinforcement learning, where an agent learns to interact with an environment to maximize a reward signal. DRL has been successful in game playing, robotics, and control problems. Deep Q-Networks (DQN) and Proximal Policy Optimization (PPO) are examples of popular DRL algorithms.

These are just a few examples of the different types of deep learning models. There are also hybrid models that combine different architectures to leverage their strengths for specific tasks. The choice of the model depends on the problem at hand, the nature of the data, and the specific requirements of the task.

3. Deep learning and traditional machine learning

Deep learning offers several advantages over traditional machine learning techniques. It can automatically learn features from raw data, handle high-dimensional data effectively, scale well with large datasets, enable end-to-end learning, and generalize to new data. That greatly makes it more efficient and worthy. However, deep learning requires substantial labeled data and computational resources and may be challenging to interpret and explain.

4. Image processing using deep learning

4.1. Overview of Object detection

As mentioned earlier the presence of vehicles in aerial images is considered a significant challenge, as they can cause distortions and rendering issues in the resulting models. In order to address this problem, it is necessary to first detect these vehicles. This detection step plays a crucial role in accurately identifying and localizing the cars for subsequent removal[20].

Image detection, Image classification, 3D-pose Estimation, Object tracking and many more are well known terms in computer vision, however there is still a confusing between these many terms[21].

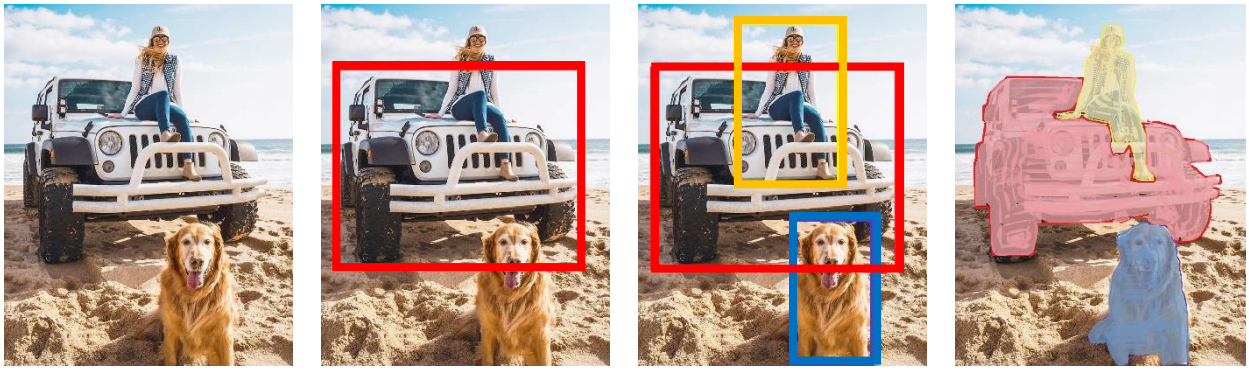

Figure II-2 : Comparison between Deep Learning Tasks

As shown in the Figure II-1, computer vision tasks can be broadly categorized into two main categories based on their focus:

Single object tasks:

- Classification: This involves assigning a single image into one of several predefined categories or classes. The goal is to determine what object or concept the image represents.
- Localization: In this task, the objective is to locate a single object within an image and provide bounding box coordinates around it. The focus is on identifying the object's presence and its spatial position within the image.

Multiple objects tasks:

- Detection: The aim is to detect and locate multiple objects of interest within an image. This involves identifying the presence and location of multiple objects while also classifying them into respective categories.
- Segmentation: This task goes a step further by providing a pixel-level mask for each object present in the image. The goal is to precisely delineate the boundaries of each object and assign a unique label to every pixel belonging to that object.

For our research we will be working on the object detection task.

4.2. Definitions

4.2.1. Object Detection

Object detection refers to the computer vision task of identifying and localizing objects within an image or a video. It involves both classifying the objects into predefined categories or classes and providing their spatial location information in the form of bounding boxes. The goal of object detection is to accurately detect and locate objects of interest, regardless of their size, orientation, or context within the given image or video frame[15].

Given an input image we aim to obtain three primary outputs:

- A list of bounding boxes, or the coordinates for each object in an image;
- A class label associated with each bounding box;
- A confidence score associated with each bounding box and class label.

4.2.2. Object Segmentation

A computer vision task that involves dividing an image into distinct regions or segments and assigning a specific label to each pixel within those regions. The goal of object segmentation is to precisely delineate the boundaries of objects within an image and assign them a unique label, indicating their semantic meaning or class[22].

Unlike object detection, which provides bounding box coordinates around objects, object segmentation aims to provide a pixel-level mask for each object in the image. This pixel-level accuracy allows for a more fine-grained understanding of the image content and enables more detailed analysis and manipulation of objects.

4.3. Applications

4.3.1. Security and Surveillance:

Object detection and segmentation have important applications in security and surveillance systems. They can be used for intrusion detection, perimeter security, crowd monitoring, suspicious object detection, facial recognition and tracking, traffic monitoring and management, and object tracking. These technologies enhance monitoring capabilities, enable proactive threat detection, and ensure public safety in security-sensitive environments.

4.3.2. Visual Search Engines:

Object-based Image Retrieval, Visual Similarity Search, Image Annotation and Tagging, Image Object Localization and Image Content Analysis are the main tasks where deep learning shines in visual search engines by enhancing the functionality and effectiveness of these engines and enabling more precise and targeted search.

4.3.3. Aerial Image Analysis

Object detection and segmentation have numerous applications in aerial image analysis such us urban planning and development, Environmental monitoring, Infrastructure inspection, Object tracking, Geospatial mapping and change detection.

4.3.4. Data Processing

They are used for data annotation, labeling, cleaning, and quality control. These techniques aid in data augmentation, compression, and storage.

4.4. Model training challenges

Training deep learning models poses several challenges that researchers and practitioners need to address. Some of the main challenges include:

4.4.1. Availability of labeled data

Deep learning models often require large amounts of accurately labeled data for effective training. However, obtaining such data can be time-consuming, expensive, and sometimes impractical. Limited labeled data can lead to overfitting or poor generalization of the model. Techniques such as data augmentation, transfer learning, and active learning can be used to overcome this challenge[15].

4.4.2. Computational resources

Deep learning models, particularly those with complex architectures and large-scale datasets, demand substantial computational power and memory. Access to high-performance GPUs and sufficient memory is a must for training deep neural networks efficiently.

4.4.3. Model architecture and hyperparameter selection

Choosing the right model architecture and hyperparameters can significantly affect the model's convergence, generalization, and overall performance. DL models offer various architecture options, including the number and type of layers, connectivity patterns, and activation functions[22].

4.4.4. Gradient vanishing or exploding

Gradients can diminish or explode during backpropagation, affecting the model's optimization process. This can lead to slow convergence or instability. Techniques like careful weight initialization, using appropriate activation functions (e.g., ReLU), employing gradient clipping, or normalization methods (e.g., batch normalization) help alleviate these problems and ensure stable gradient flow.

4.4.5. Overfitting

Overfitting occurs when the model becomes too complex and starts to memorize the training data instead of learning generalizable patterns. Regularization techniques such as dropout, L1/L2 regularization, and early stopping, ensemble learning and dropout regularization are commonly used to mitigate overfitting and improve the model's abilities.

5. Object Detection Approaches

5.1. One-stage object detector

In the regression approach, the whole image will be run through a CNN directly to generate one or more bounding boxes for objects in the images (e.g. SSD[23], YOLO[24], RetinaNet[25], FPN[26]).

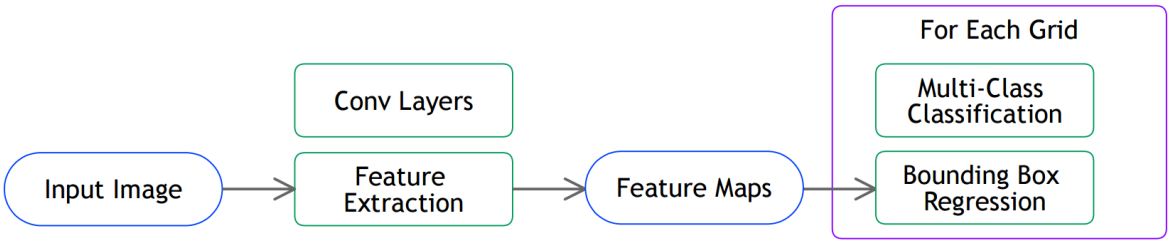

Figure II-3 : One-stage object detectors flowchart

5.2. Two-stage object detector

In the classification (or region-based) approach, the image is divided into small patches, each of which will be run through a classifier to determine whether there are objects in the patch. The bounding boxes will be assigned to patches with

positive classification results (e.g. R-CNN[18], Fast R-CNN[27], Faster R-CNN[18], R-FCN[28], Mask R-CNN[29], Light-Head R-CNN[30] ... etc.).

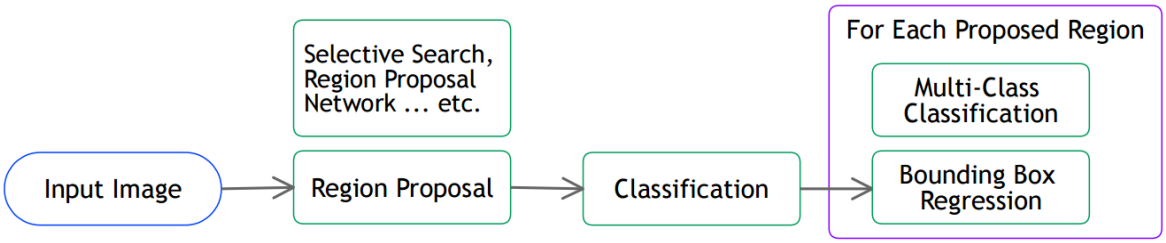

Figure II-4 : Two-stage object detectors flowchart

6. Comparison between Object detection methods

The field of Computer Vision has witnessed the introduction of innovative concepts and techniques in object detection, making it challenging to compare different object detectors and determine the best model. Each year, new systems are proposed, but conducting a fair apples-to-apples comparison becomes difficult due to variations in base feature extractors (e.g., VGG[31], Residual Networks), default image resolutions, and hardware and software platforms.

Instead of searching for the definitive best detector, the more important question to ask is which detector and configurations provide the optimal balance between speed and accuracy for a specific application. The choice should be guided by the requirements of the application at hand. By carefully selecting the detector and fine-tuning its settings, one can achieve the desired trade-off between detection speed and accuracy, catering to the specific needs of the application.

Performance Results

It is unwise to compare results from different papers side-by-side. Those experiments are done in different configurations which are not purposed for apples-to-apples comparisons. But in this section, we summarize the performance reported by the corresponding papers in Table 2:

Detector	VOC07	VOC12	MS COCO
mAP	IoU=0.5	IoU=0.5	IoU=0.5 : 0.95
R-CNN	58.5	-	-
Fast R-CNN	70.0	68.4	19.7
Faster R-CNN	73.2	70.4	42.0
YOLOv1	66.4	57.9	-
SSD	76.8	74.9	31.2
R-FCN	79.5	77.6	29.9
YOLOv3	-	-	33.0
YOLOv5	-	-	50.2
FPN	-	-	53.3
Mask R-CNN	-	-	45.2
RetinaNet	-	-	39.1
YOLOv7[32]	-	-	51.4
YOLOv8	-	-	**53.9**

Table 2 : Performance reported by the corresponding papers

Conclusion

In this chapter, we defined object detection and mentioned some of its applications. We went through the details of object detection approaches and present some of the problems and challenges that may cause issues to object detection. We also talked about the available datasets and the performance metrics for comparison reason. And concluded by showing a table that summarize the performance reported from a variety of detectors.

Chapter III : Research Results and Models Implementation

Introduction

In this chapter, we will check the used datasets for training and testing and give an overview of the selected object detector, namely You Only Look Once version 8, present the reasons which motivated this choice and describe the implementation of the adapted Yolov8 object detector that we have modified to fit our specific problem which is overhead car detection.

1. Proposed Solution

In our research, we trained two models using a RoboFlow[33] dataset of aerial images for vehicle detection. The dataset was split into training, validation, and test sets. We used the Vedai[20] (Vehicle Detection in Aerial Images) dataset for extensive testing.

To improve generalizability, we applied data pre-processing techniques[34], including auto-orientation of pixel data and image resizing. The models were trained with different hyperparameters, one on RoboFlow cloud server and the other on personal computers with limited resources. Our approach achieved accurate vehicle detection, distinguishing between various vehicle types.

The proposed structure is visually represented in this diagram:

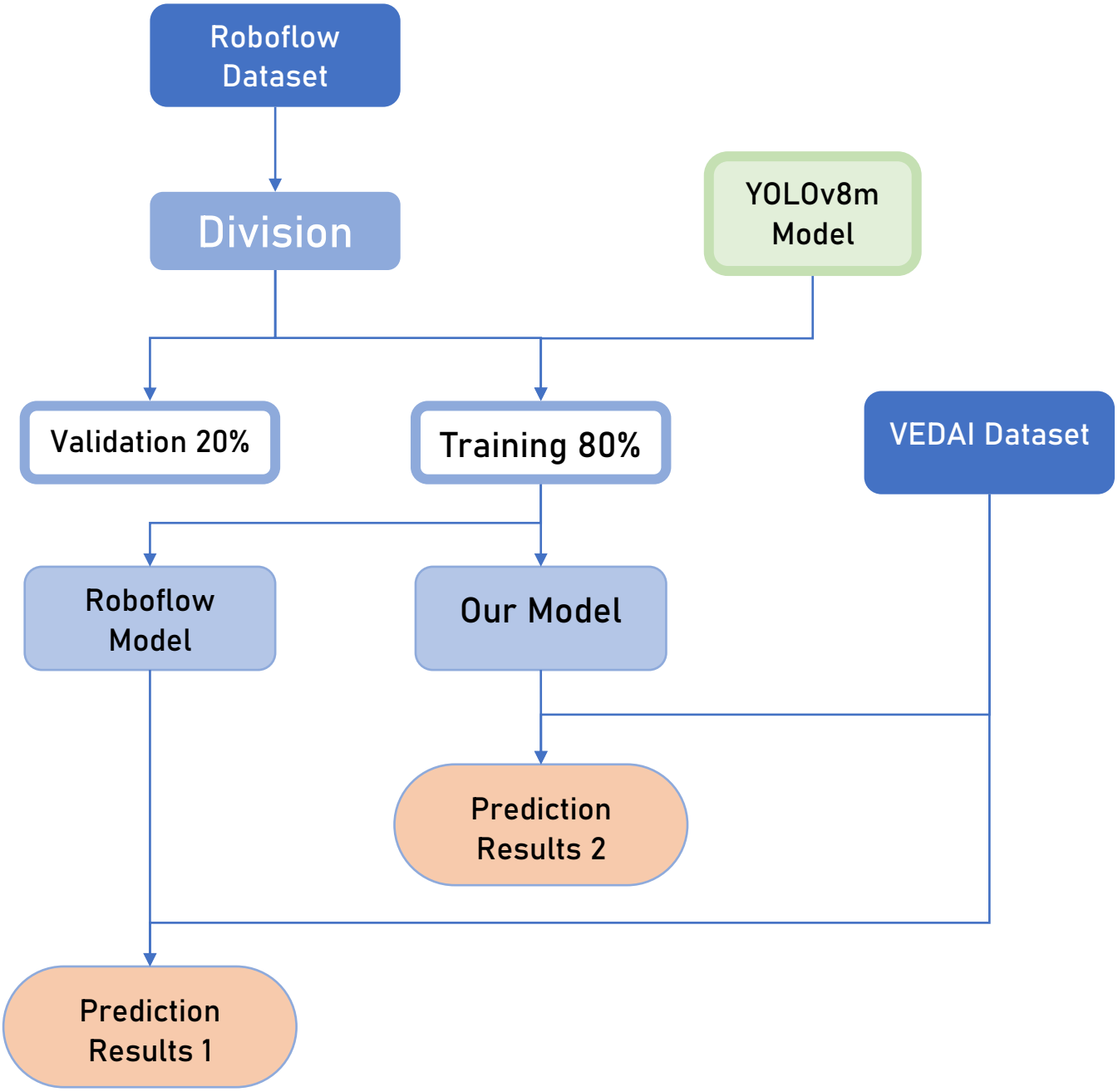

Figure III-1 : Our Research General Structure

2. Datasets and performance metrics

Over the years, numerous datasets have been made available for detection challenges. These datasets are valuable resources that provide standardized benchmarks for evaluating the performance of object detection algorithms.

2.1. Datasets

Datasets play a crucial and often underestimated role in driving research advancements. With the release of each new dataset, researchers have the opportunity to compare and enhance existing models. This iterative process of model development and evaluation fuels progress in the field. Below is a list of the most used datasets in Table 3:

Name	Images	Classes	Updated
MS COCO	330K	80	2021
ImageNet	14M	200	2021
Pascal VOC[35]	11K	20	2012
Cityscapes[36]	25k	30	2020

Table 3 : Most used datasets in artificial intelligence

2.2. Performance metrics

In the context of performance assessment, early research often relied on the term "Accuracy" to evaluate the quality of a model's predictions. This measure is typically computed by comparing the model's predictions to the ground truth information. In the case of Object Detection, the ground truth consists of the bounding box coordinates and the corresponding object class for each object present in the image. Hence, a high level of accuracy indicates that the model is capable of generating bounding boxes that closely align with the ground truth and accurately classifying the objects.

Now, when evaluating object detector performance, we use an evaluation metric called Mean Average Precision (mAP) which is based on the Intersection over Union (IoU) across all classes in our dataset.

3. Used Datasets

The dataset used for training our models is a RoboFlow universe dataset provided by a random anonymous user. It contains 2758 aerial images taken in urban aeras in the city of Columbus, Ohio, USA and the city of Potsdam in Germany. Both cities' images are 256x256 pixels size and are taken using the GeoEye-1 satellite sensor, however Columbus city ones are in grayscale.

Figure III-2 : Sample images of RoboFlow Universe dataset

The second dataset, Vedai is provided by Sebastien Razakarivony and Frederic Jurie. It contains 1999 images at 256x256 resolution. No further information about the dataset was announced by the publishers.

Figure III-3 : Sample images of Vedai dataset

4. Images pre-processing

In the pre-processing step we ran our datasets through 2 stages, auto-orientation of pixel data (with EXIF-orientation stripping) and image resizing.

The auto-orientation process strips images of their EXIF data so that we see images displayed the same way they are stored on disk. This step is obligatory by the Roboflow cloud server training for any dataset, therefore the second model that we trained in our personal computer does not run through this phase.

However, the resizing process touch both our datasets because the algorithm we are going to use is Yolov8 and it automatically resize the images to 640x640 as standard or any other selected dimension.

Then, we divided our Roboflow data to training and validation images as follow:

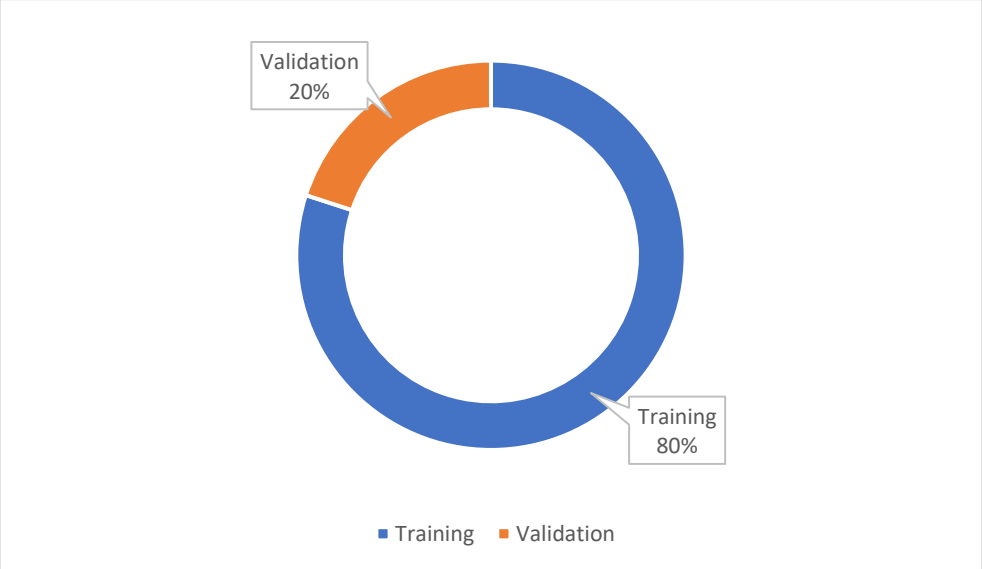

Figure III-4 : Dataset Division

5. Training Models using Yolov8

Based on the detectors comparison we mentioned in chapter II Table 2, we decided to use a Yolov8[37] model and train it for detecting vehicles in aerial images.

The YOLOv8 model represents an innovative advancement developed by the team responsible for the highly influential YOLOv5 architecture[38]. This novel iteration exhibits substantial enhancements in terms of visual perception when compared to existing models. Empirical evidence, as illustrated by performance graphs provided by the Ultralytics team (Figure III-5), demonstrates notable improvements in accuracy across the widely-used MS COCO dataset[21].

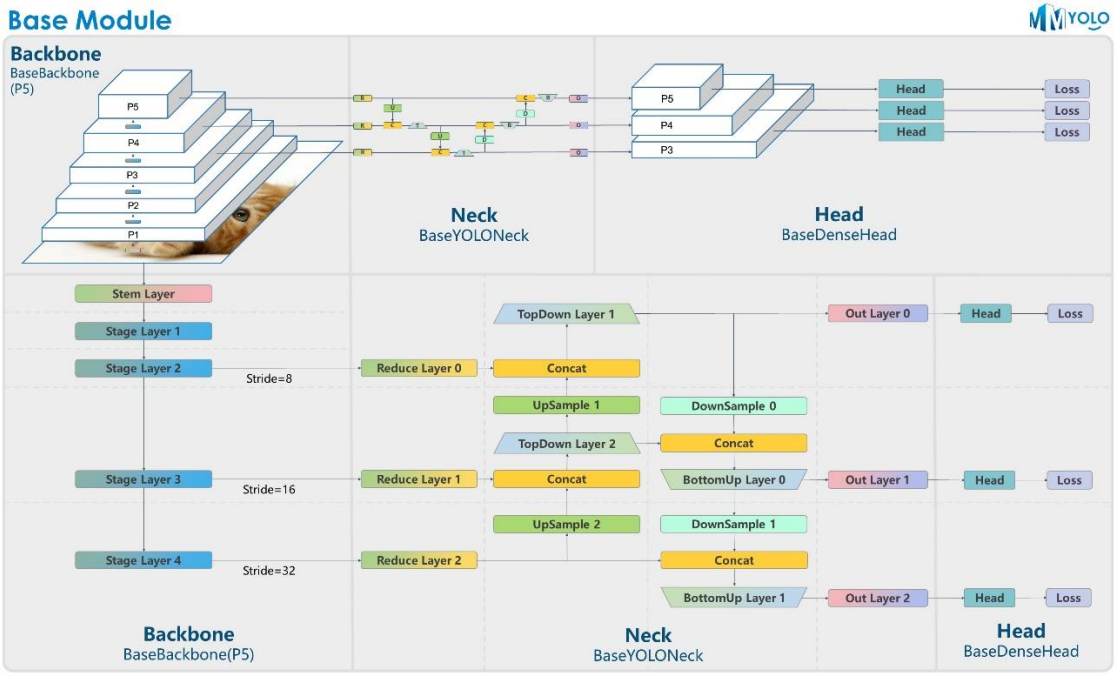

Figure III-5 : Yolov8 Architecture[39]

The YOLOv8 model is based on the YOLOv5 architecture, but it introduces a number of key innovations. These include:

- A new backbone network that is more efficient and accurate.
- A new head network that is better at predicting object bounding boxes and classes.
- A new loss function that is more robust to noise and outliers.

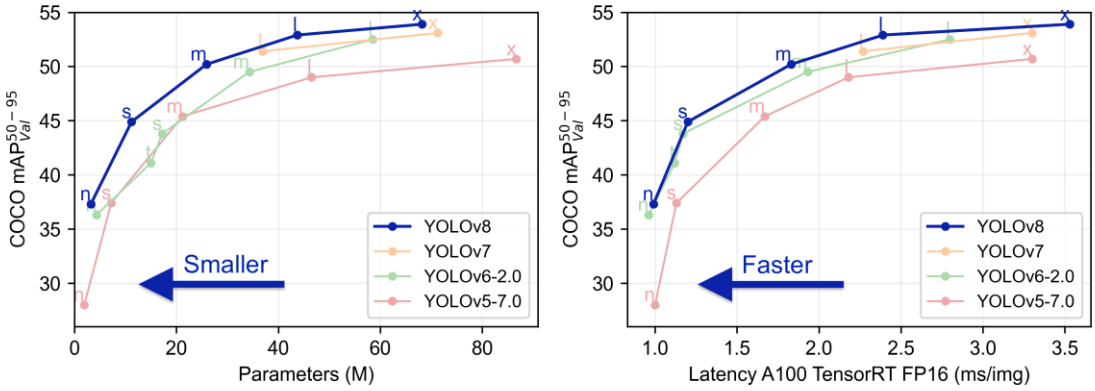

Figure III-6 : Yolov8 Performance compared to other Yolo models[40]

The distinction between the two trained models primarily lies in the hyperparameters employed during the training process. The model trained on the RoboFlow cloud server utilized the yolov8m.pt base model, with 16 batches, 199 epochs, an 8-core CPU, and an NVIDIA V100 GPU.

Conversely, our local training involved the yolov8m.pt base model, with 8 batches, 50 epochs, a 4-core CPU, and an AMD Radeon Vega 8 GPU.

These variations in hyperparameters were selected based on the available resources and computational capacity of each training environment resulting in a quite gap in the overall performance of each model.

6. Training Results

The training of computer vision models became a lot easier with Yolov8 since the developers officially added a YOLO package to the python libraries, however it still has significant limitations and challenges. To get a performant model we changed some of the hyperparameters and configuration data of the main yolov8 medium model (which could never predict vehicles from a top view) like classes, layers and region selection rules. Finally, we had to train the model and validate it.

Since we have the yolo package, all we had to do is run the following code snippets for training and validation

```
yolo task=detect mode=train  data={dataset.location}
/data.yaml epochs=50 model=yolov8m.pt imgsz=640 batch=8


yolo task=detect mode=val model={HOME}/runs/detect
/train/weights/best.pt data={dataset.location}/data.yaml
```

after the training is complete and result graphs are extracted, we used yet another simple code to detect vehicles in our test dataset.

```
yolo predict model=best.pt imgsz=256 conf=0.5
source="D:/Khairou/Studies/Univ/M2 SI/Q4/PFE/Code/
DataSets/vedai-master/images/train/" line_thickness=1
save_txt=true
```

6.1. Roboflow model

Cloud training took an hour to complete and provided us with the following results:

6.1.1. Performance Graphs

Obtained results of this model:

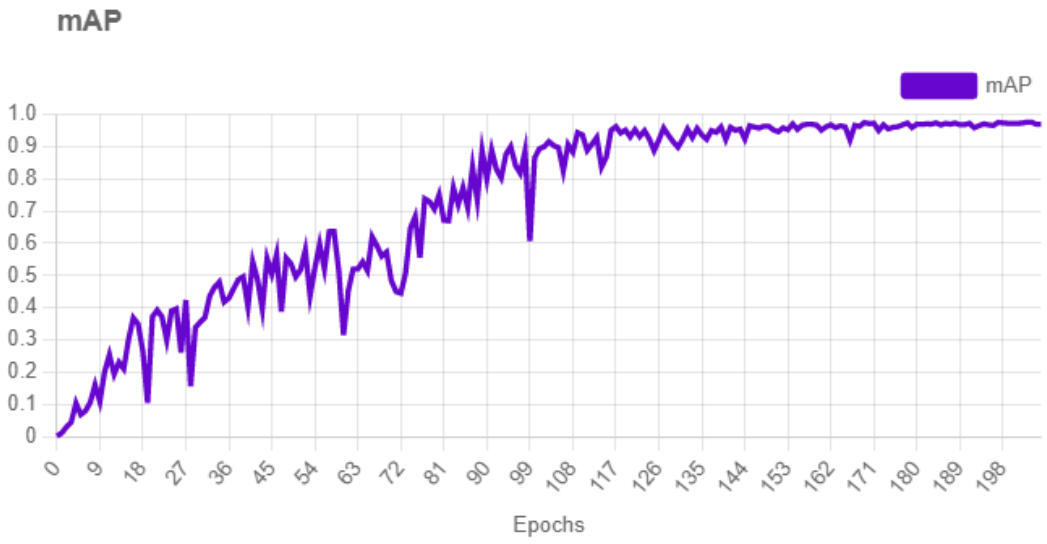

Figure III-7 : Mean Average Precision over epochs

The observed graph demonstrates the training progress of our model in terms of Mean Average Precision (mAP). Initially, the model's mAP started at 0 and gradually increased over the course of several epochs. This consistent improvement indicates that the model was effectively learning and not suffering from overfitting.

However, after approximately 110 epochs, the model's mAP reached a plateau at around 0.96. It appeared that the precision remained relatively stable for the 90 epochs of training. While this may give the impression that the model's development has stagnated, it is a critical stage in achieving the desired refinement. It is this incremental progress that distinguishes our model from other available models, as it represents a mere 2% difference that contributes significantly to its overall performance and pushes to final mAP to 98%.

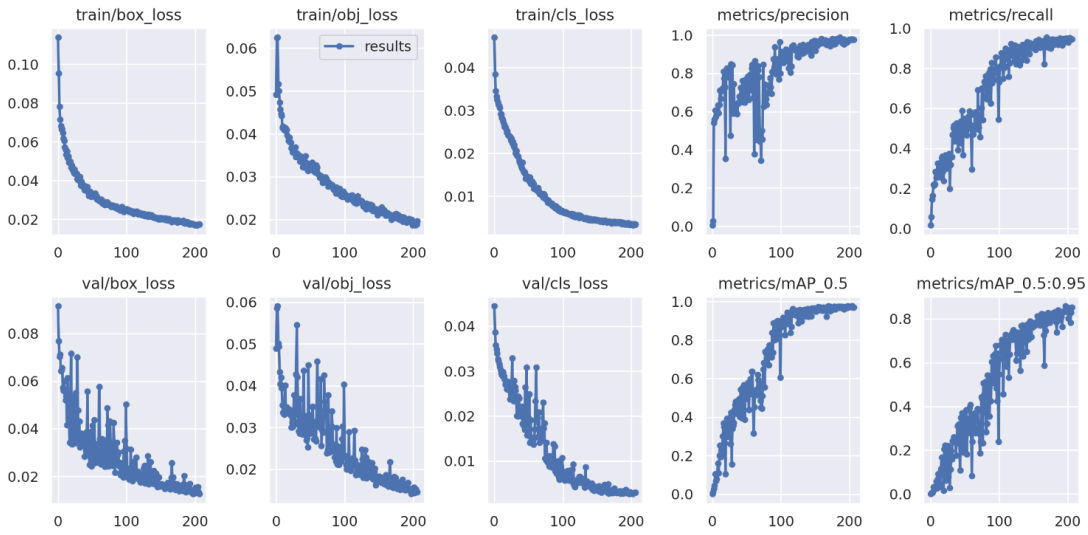

Figure III-8 : Training graphs of the Roboflow model

When utilizing the Roboflow cloud server for training, there are certain limitations to be aware of. Firstly, the model can only be used online, and it is limited to processing a single image at a time. This means that multiple images predictions may not be supported.

6.1.2. Detection Samples

Figure III-9 : Predictions of Roboflow model

6.2. Local model

Our local model completed the training in a very stressful 2 days and 18 hours, proving us with these results.

6.2.1. Performance graphs

Our local model got the following results:

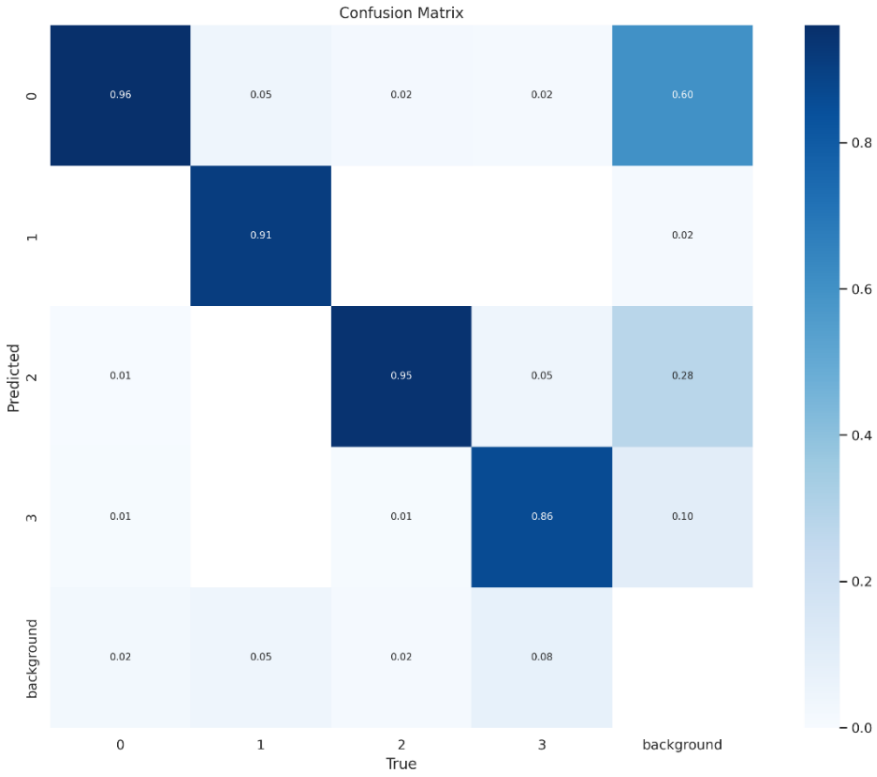

Figure III-10 : Personal model Confusion matrix

The confusion matrix demonstrates promising results for a locally trained model with a limited training duration of 50 epochs. The average accuracy of the model is 92.5%. However, it is important to note that the accuracy of class 3, representing "Other vehicles," is relatively low. Conversely, classes 0, 1, and 2, representing "Hatchbacks," "Pickups," and "Sedans" respectively, exhibit strong performance. The reason behind the low accuracy of class 3 ("Other vehicles") can be attributed to the limited number of labels available during the training phase.

It is worth mentioning that the "background" class is not one of the targeted objects for detection but rather denotes non-object regions. Its inclusion in the model's training was intended to enhance overall accuracy.

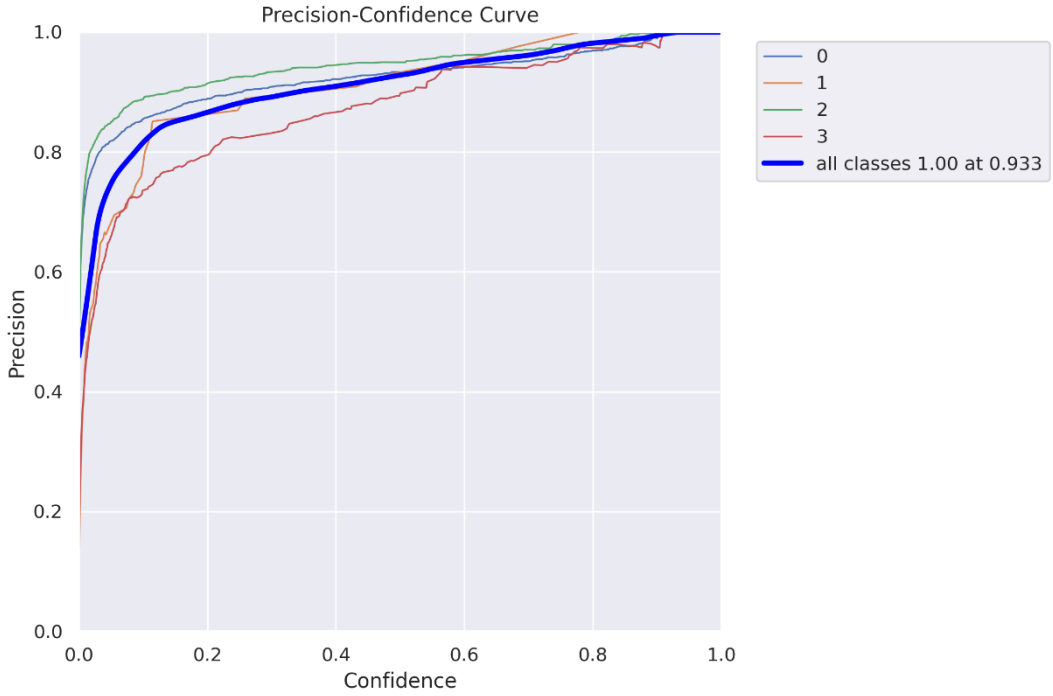

Figure III-11 : Precision-Confidence Curve

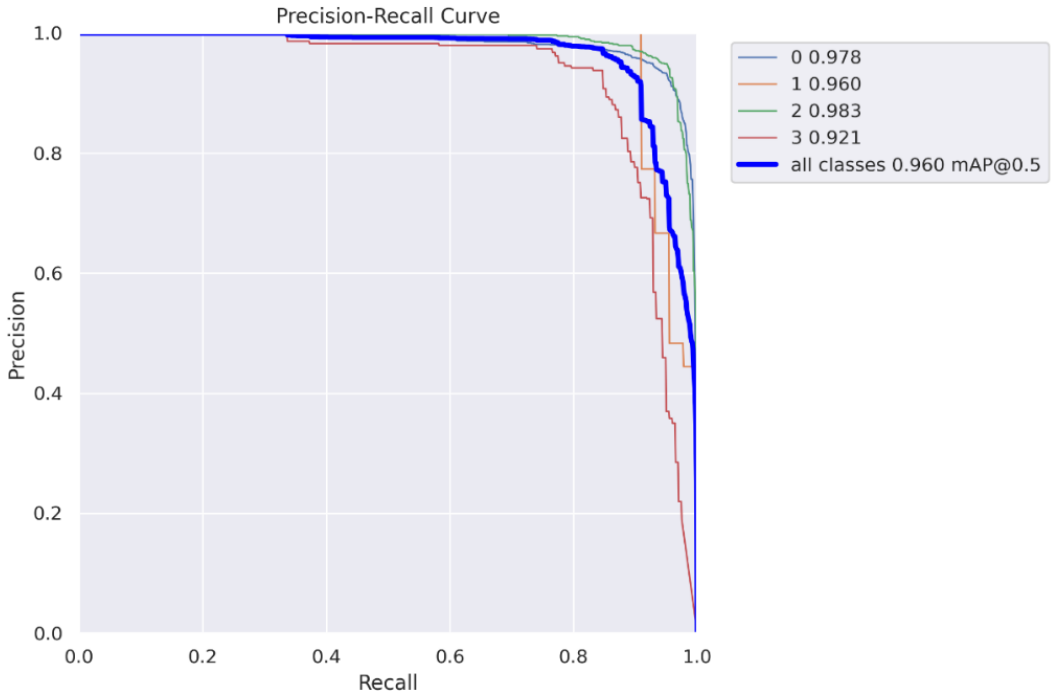

Figure III-12 : Precision Recall Curve

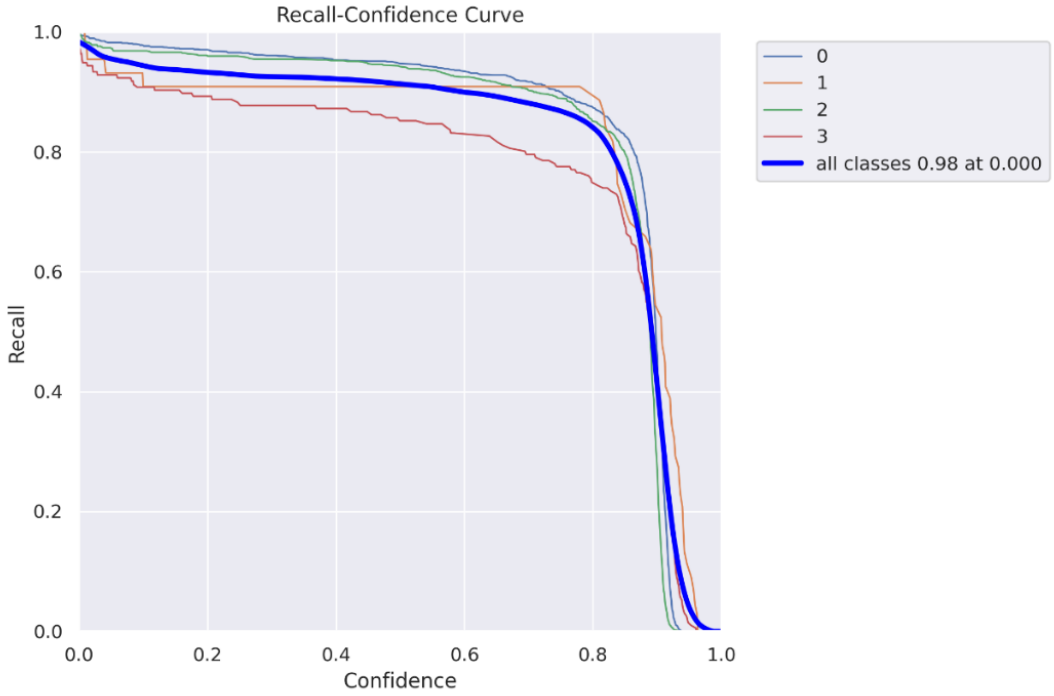

Figure III-13 : Recall-Confidence Curve

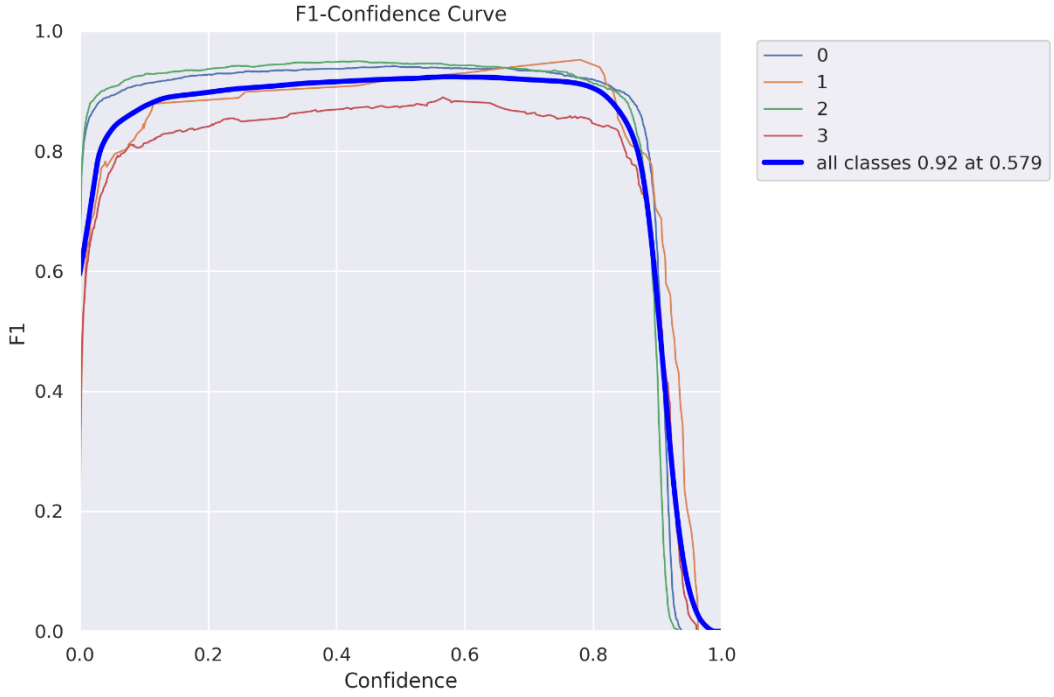

Figure III-14 : F1 Curve

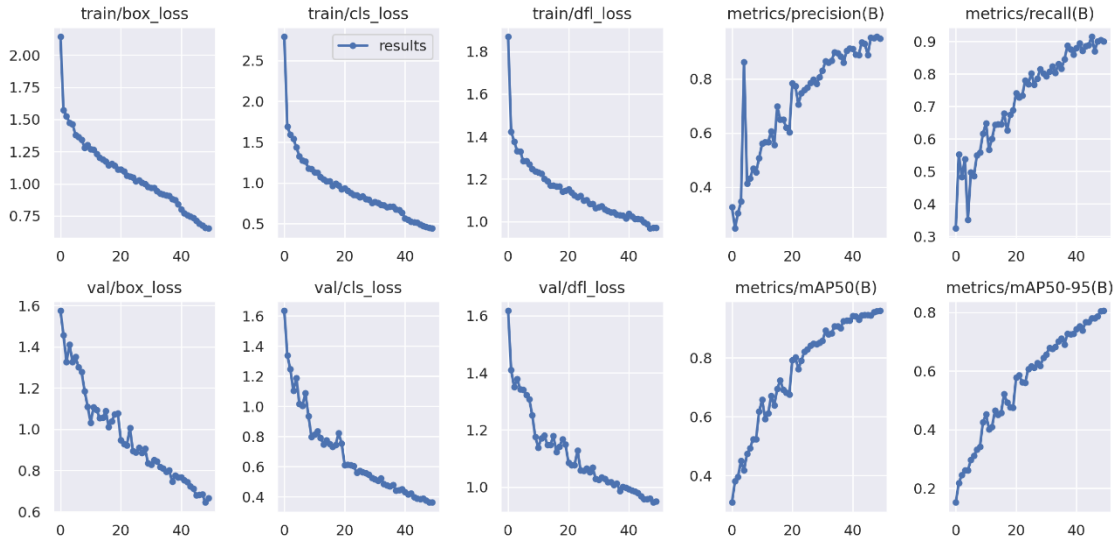

Figure III-15 : Training graphs for the personal model

Figures III-11 to III-15 provides a comprehensive overview of the performance evaluation of our model, complementing the confusion matrix. Notably, the loss values for bounding boxes (box_loss), classes (cls_loss), and Distribution Focal Loss (dfl_loss) show a significant degradation, indicating that our model is undergoing successful training. Additionally, the precision, recall and mAP graphs demonstrate a consistent rise, reaching around 0.95.

These results are particularly remarkable, considering the model's training limited to only 50 epochs. The findings from this evaluation hold significant implications for my master's thesis.

6.2.2. Detection Samples

Figure III-16 : Predictions of Personal Model

6.3. Comparison between the models

After a thorough analysis of the two models, it was observed that the Roboflow model exhibited superior performance in detecting vehicles, achieving an average precision of 0.98. In contrast, the personal model achieved a lower average precision of 0.94 and remarkably missed some vehicles that was either on the edges of the images of partially-hidden by buildings and trees. Based on these results, it can be concluded that longer training time **with more resources** generally lead to better model performance.

However, it is important to note that higher-performance cloud servers often come with limitations, such as in Roboflow case restricted access to training data and no access to the detection model itself. These limitations can impact the ability to fine-tune or modify the model according to specific requirements.

7. Limitations and Area of Improvement

As we have seen detecting objects in general is considered an easy and fast task, but training the detection model requires a lot more time and resources, and that is what stood in the way of further improving our models and fine-tune it. The

restricted access to training data of Roboflow model and the very poor hardware performance of our personal computers forced us to conclude this thesis without reaching the peak of the research.

If not, our work would have extended to Object segmentation for more accurate detection and extracting masks instead of bounding boxes. Then passing to the Object Removal and inpainting phase as the second and final part of eliminating vehicles and offering perfect and reliable aerial images for 3d city modelers.

Conclusion

In this chapter, we have used Yolov8 to create our model that takes aerial images as input and outputs a bounding box for each vehicle instance in the image. These boxes will be used as inputs to other researches which will concentrate on vehicle removal and image inpainting as the second pre-processing phase to get the desired aerial images. Besides, we included the implementation of the chosen detector, a description of data and also the preparation of the dataset used for its training and testing as well as pointing fingers to the limitations and area of improvements of our research.

General Conclusion

In conclusion, this thesis has explored the increasing importance of 3D city models in various aspects of our lives. However, the process of building these models is often challenging, particularly when it comes to the presence of vehicles in the data inputs. To address this issue, an efficient vehicle detection model has been developed as part of this research.

The developed model has demonstrated its capability to accurately detect vehicles in aerial images, paving the way for their removal and the creation of pristine aerial images. This process significantly simplifies the 3D city modeling process by providing high-quality imagery devoid of vehicles.

The results obtained from the developed model show its potential for practical application in various fields, such as urban planning, architecture, transportation management, and virtual simulations. The accurate detection of vehicles enables researchers, city planners, and decision-makers to focus on the core aspects of city modeling without the hindrance of unwanted objects.

Our research not only presents a novel approach to vehicle detection but also emphasizes the importance of automation and efficiency in the development of 3D city models. It opens up opportunities for further research and development in the field, encouraging the exploration of innovative techniques to enhance the quality and realism of 3D city representations.

Because there is no such completed or perfect research, ours still holds a lot of potential for the future. Adding the vehicles removal and inpainting phase is the first step to improve it whether using deep learning methods such as GANs or develop a whole new algorithm to do so. Next we might think about 3D City models, how can we enhance and push them to the near millimeter perfect quality which will prove it’s worth over the few next years.

Last but not least, the developed vehicle detection model offers a valuable contribution to the field of 3D city modeling by providing an effective solution to the challenges posed by vehicles in aerial images. Its implementation holds great promise for advancing the accuracy, efficiency, and realism of 3D city models, ultimately benefiting a wide range of applications and stakeholders.

References

[1] M. Shashi and K. Jain, ‘Use of photogrammetry in 3D modeling and visualization of buildings’, vol. 2, Jan. 2007, ARPN Journal of Engineering and Applied Sciences.

[2] C. Portalés, J. L. Lerma, and S. Navarro, ‘Augmented reality and photogrammetry: A synergy to visualize physical and virtual city environments’, *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 1, pp. 134–142, Jan. 2010, doi: 10.1016/j.isprsjprs.2009.10.001.

[3] C. Frueh, R. Sammon, and A. Zakhor, ‘Automated texture mapping of 3D city models with oblique aerial imagery’, in *Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004.*, Sep. 2004, pp. 396–403. doi: 10.1109/TDPVT.2004.1335266.

[4] N. Paparoditis, O. Bentrah, M. Deveau, O. Tournaire, and L. Pénard, ‘B.2 Modélisation 3D automatique terrestre d’environnements urbains et complexes à très grande échelle’.

[5] Thompson, M. Emine, and Margaret, ‘Sharing 3D city models: an overview’, p. 9, 2009, Northumbria Research Link, Education and Research in Computer Aided Architectural Design in Europe (eCAADe), pp. 261-267. ISBN 978-0954118389.

[6] S. A. Aydar, J. Stoter, H. Ledoux, E. D. Ozbek, and T. Yomralioglu, ‘ESTABLISHING A NATIONAL 3D GEO-DATA MODEL FOR BUILDING DATA COMPLIANT TO CITYGML: CASE OF TURKEY’, Jan. 2016, doi: 10.5194/isprsarchives-XLI-B2-79-2016.

[7] S. P. Singh, K. Jain, and V. R. Mandla, ‘VIRTUAL 3D CITY MODELING: TECHNIQUES AND APPLICATIONS’, *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XL-2-W2, pp. 73–91, Aug. 2013, doi: 10.5194/isprsarchives-XL-2-W2-73-2013.

[8] F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin, ‘Applications of 3D City Models: State of the Art Review’, *ISPRS Int. J. Geo-Inf.*, vol. 4, no. 4, Art. no. 4, Dec. 2015, doi: 10.3390/ijgi4042842.

[9] M. Trapp *et al.*, ‘Colonia 3D Communication of Virtual 3D Reconstructions in Public Spaces’, *Int. J. Herit. Digit. Era*, vol. 1, no. 1, pp. 45–74, Mar. 2012, doi: 10.1260/2047-4970.1.1.45.

[10] M. Vaaraniemi, M. Goerlich, and A. in der Au, ‘Intelligent Prioritization and Filtering of Labels in Navigation Maps’, 2014, Accessed: May 13, 2023. [Online]. Available: http://dspace5.zcu.cz/handle/11025/11895

[11] J. Engel and J. Döllner, ‘Approaches Towards Visual 3D Analysis for Digital Landscapes and Its Applications’, Willkommen am Hasso-Plattner-Intsitut, DE.

[12] T. H. Kolbe, G. Gröger, and L. Plümer, ‘CityGML: Interoperable Access to 3D City Models’, in *Geo-information for Disaster Management*, P. Van Oosterom, S. Zlatanova, and E. M. Fendel, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 883–899. doi: 10.1007/3-540-27468-5_63.

[13] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, ‘Object Detection With Deep Learning: A Review’, *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[14] O. Russakovsky *et al.*, ‘ImageNet Large Scale Visual Recognition Challenge’, *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[15] K. S. Chahal and K. Dey, ‘A Survey of Modern Object Detection Literature using Deep Learning’, *arXiv.org*, Aug. 22, 2018. https://arxiv.org/abs/1808.07256v1 (accessed May 13, 2023).

[16] ‘4. Major Architectures of Deep Networks - Deep Learning [Book]’. https://www.oreilly.com/library/view/deep-learning/9781491924570/ch04.html (accessed May 28, 2023), International Journal of Computer Vision 104, 154-171(2013), Spring Linker.

[17] D. Tang, W. Jin, D. Liu, J. Che, and Y. Yang, ‘Siam Deep Feature KCF Method and Experimental Study for Pedestrian Tracking’, *Sensors*, vol. 23, no. 1, Art. no. 1, Jan. 2023, doi: 10.3390/s23010482.

[18] S. Ren, K. He, R. Girshick, and J. Sun, ‘Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks’, in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2015. Accessed: May 13, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

[19] L. Yu, W. Zhang, J. Wang, and Y. Yu, ‘SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient’, *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, Art. no. 1, Feb. 2017, doi: 10.1609/aaai.v31i1.10804.

[20] S. Razakarivony and F. Jurie, ‘Vehicle detection in aerial imagery : A small target detection benchmark’, *J. Vis. Commun. Image Represent.*, vol. 34, pp. 187–203, Jan. 2016, doi: 10.1016/j.jvcir.2015.11.002.

[21] T.-Y. Lin *et al.*, ‘Microsoft COCO: Common Objects in Context’, in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 740–755. doi: 10.1007/978-3-319-10602-1_48.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, ‘Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation’, presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587. Accessed: May 13, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.html

[23] W. Liu *et al.*, ‘SSD: Single Shot MultiBox Detector’, in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ‘You Only Look Once: Unified, Real-Time Object Detection’, presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788. Accessed: May 13, 2023. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html

[25] M. Zlocha, Q. Dou, and B. Glocker, ‘Improving RetinaNet for CT Lesion Detection with Dense Masks from Weak RECIST Labels’, in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, D. Shen, T. Liu, T. M. Peters, L. H. Staib, C. Essert, S. Zhou, P.-T. Yap, and A. Khan, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 402–410. doi: 10.1007/978-3-030-32226-7_45.

[26] H. Xu, L. Yao, W. Zhang, X. Liang, and Z. Li, ‘Auto-FPN: Automatic Network Architecture Adaptation for Object Detection Beyond Classification’, presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6649–6658. Accessed: May 21, 2023. [Online]. Available: https://openaccess.thecvf.com/content_ICCV_2019/html/Xu_Auto-FPN_Automatic_Network_Architecture_Adaptation_for_Object_Detection_Beyond_Classification_ICCV_2019_paper.html

[27] R. Girshick, ‘Fast R-CNN’, presented at the Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448. Accessed: May 21, 2023. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html

[28] J. Dai, Y. Li, K. He, and J. Sun, ‘R-FCN: Object Detection via Region-based Fully Convolutional Networks’, in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2016. Accessed: May 13, 2023. [Online]. Available: https://proceedings.neurips.cc/paper/2016/hash/577ef1154f3240ad5b9b413aa7346a1e-Abstract.html

[29] ‘matterport/Mask_RCNN: Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow’. https://github.com/matterport/Mask_RCNN (accessed May 13, 2023).

[30] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, ‘Light-Head R-CNN: In Defense of Two-Stage Object Detector’, *arXiv.org*, Nov. 20, 2017. https://arxiv.org/abs/1711.07264v2 (accessed May 13, 2023).

[31] D. A. Dutta, ‘VGG Image Annotator (VIA)’, Seebibyte: Show and Tell Event, 15 June 2017.

[32] ‘WongKinYiu/yolov7: Implementation of paper - YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors’, *GitHub*. https://github.com/WongKinYiu/yolov7 (accessed May 28, 2023).

[33] Q. Lin, G. Ye, J. Wang, and H. Liu, ‘RoboFlow: a Data-centric Workflow Management System for Developing AI-enhanced Robots’, in *Proceedings of the 5th Conference on Robot Learning*, PMLR, Jan. 2022, pp. 1789–1794. Accessed: May 21, 2023. [Online]. Available: https://proceedings.mlr.press/v164/lin22c.html

[34] B. Chitradevi and P. Srimathi, ‘An Overview on Image Processing Techniques’, vol. 2, no. 11, 2007, International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization), ISSN(Online): 2320-9801.

[35] ‘The PASCAL Visual Object Classes Homepage’. http://host.robots.ox.ac.uk/pascal/VOC/ (accessed May 21, 2023), International Journal of Computer Vision, 111, 98-136, Springer Link.

[36] ‘Dataset Overview – Cityscapes Dataset’. https://www.cityscapes-dataset.com/dataset-overview/ (accessed May 21, 2023).

[37] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, ‘A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition’, *Drones*, vol. 7, no. 5, Art. no. 5, May 2023, doi: 10.3390/drones7050304.

[38] G. Jocher *et al.*, ‘ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations’, *Zenodo*, Apr. 2021, doi: 10.5281/zenodo.4679653.

[39] ‘open-mmlab/mmyolo’. OpenMMLab, May 18, 2023. Accessed: May 18, 2023. [Online]. Available: https://github.com/open-mmlab/mmyolo

[40] G. Jocher, A. Chaurasia, and J. Qiu, ‘YOLO by Ultralytics’. Jan. 2023. Accessed: May 21, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics