https://doi.org/10.15388/vu.thesis.519
https://orcid.org/0000-0002-2611-3383

VILNIUS UNIVERSITY

Povilas Gudžius

Automated Machine Learning for Accurate and Low-latency Object Recognition in Optical Satellite Imagery

DOCTORAL DISSERTATION

Technological Sciences,
Informatics Engineering (T 007)

VILNIUS 2023

This dissertation was prepared between 2017 and 2022 at Vilnius University.

Academic Supervisor – Prof. Dr. Olga Kurasova (Vilnius University, Technological Sciences, Informatics Engineering – T 007).

Academic Consultant – Assoc. Prof. Dr. Ernestas Filatovas (Vilnius University, Technological Sciences, Informatics Engineering – T 007).

This doctoral dissertation will be defended in a public meeting of the Dissertation Defence Panel:

Chairman – Prof. Dr. Povilas Treigys (Vilnius University, Technological Sciences, Informatics Engineering – T 007).

Members:
Prof. Dr. Pasi Fränti (East Finland University, Finland, Technological Sciences, Informatics Engineering – T 007),
Prof. Dr. Virginijus Marcinkevičius (Vilnius University, Technological Sciences, Informatics Engineering – T 007),
Prof. Dr. Audris Mockus (Vilnius University, Technological Sciences, Informatics Engineering – T 007),
Prof. Habil. Dr. Rimvydas Simutis (Kaunas University of Technology, Technological Sciences, Informatics Engineering – T 007).

The dissertation shall be defended at a public meeting of the Dissertation Defence Panel at 1 p.m. on the 8th of September 2023, in Room 203 of the Institute of Data Science and Digital Technologies of Vilnius University. Address: Akademijos Street 4, LT-04812, Vilnius, Lithuania.

The text of this dissertation can be accessed at the library of Vilnius University, as well as on the website of Vilnius University: www.vu.lt/lt/naujienos/ivykiu-kalendorius

https://doi.org/10.15388/vu.thesis.519
https://orcid.org/0000-0002-2611-3383

VILNIAUS UNIVERSITETAS

Povilas Gudžius

Automatinis mašininis mokymasis, skirtas tiksliam ir greitam objektų atpažinimui optiniuose palydoviniuose vaizduose

DAKTARO DISERTACIJA

Technologijos mokslai,
Informatikos inžinerija (T 007)

VILNIUS 2023

Disertacija rengta 2017–2022 metais Vilniaus universitete.

Mokslinė vadovė – prof. dr. Olga Kurasova (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – T 007).
Mokslinis konsultantas – doc. dr. Ernestas Filatovas (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – T 007).

Gynimo taryba:

Pirmininkas – prof. dr. Povilas Treigys (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – T 007).

Nariai:
prof. dr. Pasi Fränti (Rytų Suomijos universitetas, Suomija, technologijos mokslai, informatikos inžinerija – T 007),
prof. dr. Virginijus Marcinkevičius (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – T 007),
prof. dr. Audris Mockus (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – T 007),
prof. habil. dr. Rimvydas Simutis (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – T 007).

Disertacija ginama viešame Gynimo tarybos posėdyje 2023 m. rugsėjo 8 d. 13 val. Vilniaus universiteto Duomenų mokslo ir skaitmeninių technologijų instituto 203 auditorijoje. Adresas: Akademijos g. 4, LT-08412, Vilnius, Lietuva.

Disertaciją galima peržiūrėti Vilniaus univeristeto bibliotekoje ir VU interneto svetainėje adresu: www.vu.lt/lt/naujienos/ivykiu-kalendorius

TERMS AND ABBREVIATIONS

AutoML – Automated Machine Learning

CNN – Convolutional Neural Network

CRF – Conditional Random Field

DANet – Dual Attention Network

DeepLab – Subtype of FCN architecture

DNN – Deep Neural Network

FCAU-NET – Fully Convolutional Attention-based UNET

FCN – Fully Convolutional Neural Network

FLOP – Floating-Point Operation

G-FLOP – Giga-Floating Point Operations

GPU – Graphics Processing Unit

HDCN – Hybrid Deep Convolutional Network

HRNet – High-Resolution Network

LEO – Lower Earth Orbit

MACU – Multi-Scale Skip Connected Architecture of UNET

NAS – Neural Architecture Search

NAS-MACU – MACU backbone-based network topology designed by NAS

PSPNet – Pyramid Scene Parsing Network

RCNN – Regions-CNN

RPN – Region Proposal Network

SSD – Single Shot Detectors

TPU – Tensor Processing Unit

UNET – Subtype of FCN architecture

ABSTRACT

Satellite imagery is changing how we understand and predict economic activity worldwide. Advancements in optical satellite hardware and lower costs for orbital rocket launches with satellite payloads increased the demand for geospatial intelligence. Commercial satellite constellations by Airbus Defence and Space, Maxar technologies, resulted in near-real-time, high-resolution images, covering the entire Earth and opening doors for brand new applications of geospatial data. Human annotators cannot manually analyse petabytes of satellite imagery in the current computer vision research; dealing with this problem still lacks 1) accuracy and 2) prediction speed, both significantly important metrics for latency-sensitive industrial applications. In the dissertation, we address both of the aforementioned challenges by proposing improvements to the object recognition model design, training, and complexity regularisation applicable to a range of neural networks.

The dissertation proposes a framework for optimizing a Fully Convolutional Neural Network (FCN) architecture (UNET) designed for accurate and fast object recognition in optical satellite imagery. We show that this FCN exceeds existing networks' performance with state-of-the-art speed over multiple sensors and outperforms other proposed methods in this specific domain of light-vehicle object class recognition in optical satellite imagery. Its computationally light architecture delivers a fivefold improvement in training time and a rapid prediction, essential to real-time applications. To illustrate practical model effectiveness, we analyse it in the context of an algorithmic trading environment.

In addition to improving and adapting the FCN, we also examine the limitations of manually-designed neural networks. The object recognition problem in multi-spectral satellite imagery carries unique intricate spatial structures and dataset properties such as perspective distortion, resolution variability, data spectrality, and other features that make it difficult for a specific human-invented and manually designed neural network to perform well across dispersed datasets. It requires manual recalibration and further configuration-testing to adjust the neural network architecture to the task at hand. The present dissertation evaluates and proposes how Automated Machine Learning (AutoML) based techniques can be employed to solve these limitations.

We then continue to examine the area for AutoML with particular emphasis on Neural Architecture Search and propose the NAS-MACU type

of architecture learning framework that automatically designs and adopts neural network architecture within the MACU backbone. Neural architecture search is conducted at the cell-level which is a building block of reusable neural network modules that perform a specific function such as convolutional or recurrent operation. The constructed NAS-MACU performed exceptionally well in a low information environment compared to manually designed networks.

A proprietary annotated satellite imagery dataset was created, published and open-sourced to contribute to the further development of this research field. Research findings can be readily implemented in other object recognition applications too.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Research context, motivation, and relevance

According to the Committee on Earth Observation Satellites (CEOS), commercial satellite imagery will soon reach the coverage of the entire Earth, with near-real-time frequency and high-resolution [1] [2]. Commercial satellite constellations from Maxar technologies like RADARSAT-2 [3], Pleiades-1 and ICESat-2 [4], Vision-1 from Airbus Defence and Space [5], and Cartosat-3 by IRSO [6] are providing full earth visual coverage of RGB and panchromatic imagery with a resolution close to the maximum legal accuracy of 25 cm per pixel [7].

The growing accessibility and affordability of satellite and aerial imagery have resulted in a significant surge in the utilization of these image types across a wide range of applications. Industries that utilize this data include government, military, agriculture, supply chain and finance. It enables non-profits and governments to leverage these insights for humanitarian purposes, including economic impact assessment of global pandemic (object count of aircraft, lorries in supply chains, container ships), rapid forest wildfire detection [5], time-sensitive flash flood hydraulic modelling [7], [6] precision agriculture, environmental impact prevention for extractive industries and surveillance for disaster relief [8]. In the financial sector, quantamental hedge funds utilize satellite imagery as a source of intelligence for their financial trading algorithms to generate access returns (alpha) [3]. Alpha is a measure of the excess return generated by an investment strategy or portfolio after accounting for risk and expected returns. In the context of quantamental hedge funds, it represents the value added by the investment manager's skill in exploiting market inefficiencies while adopting alternative data such as satellite imagery. Near-real-time satellite imagery combined with computer vision enables investment managers to leverage insights from the “ground-truth” data to predict the price movements of financial securities in the public stock and commodity markets. Examples of practical applications encompass revenue prediction for companies using car count data across parking lots, estimation of manufacturing output by analysing supply chain activity, forecasting agricultural commodity prices through estimated crop yields, and detection of oil supply by monitoring global oil tank lids [4].

As novel real-world use cases arise, they increase the demand for developing high-precision and real-time computer vision techniques [9]. Human-derived analytics and data annotation are no longer economically

viable. Based on recognised standards [10], a professional annotator can annotate approximately 1km^2 to 2km^2 of satellite imagery per day within the light-vehicles object class. Therefore, annotating 100km^2 of satellite imagery would take approximately 50 to 100 days for a single annotator to complete [11]. Even though substantially better than human annotators, the latest computer vision models still require a significant amount of time (over 30 min.) to process approximately 100 km^2 of satellite imagery [12] with an object recognition accuracy level [13] lower than that of professional human annotators (<90%) [14] [15] [16].

In addition, the current body of academic research exhibits a lack of comprehensive methodologies aimed at improving object recognition models specifically tailored to address the intricacies inherent in satellite imagery as a distinct data category [17] [18]. The formidable challenge stems from the unique properties of satellite imagery as a dataset itself, including perspective distortion, resolution variability, data spectrality, and other salient characteristics that render conventional human-invented neural networks ill-suited to excel in the presence of diverse and dispersed scenic elements. Consequently, the observed limitations in both accuracy and prediction speed contribute to an exacerbation of the bottleneck effect, impeding the seamless integration of satellite imagery into real-world, latency-sensitive applications, such as algorithmic trading within the financial securities domain [19].

Satellite images can now be effectively processed using Convolutional Neural Network (CNN) models, which are popular deep learning techniques widely employed for object detection and segmentation tasks. CNN has found extensive application in computer vision tasks such as object segmentation, object tracking, change detection, foreground object detection, optical flow, pose estimation, and semantic segmentation. Among these applications, semantic segmentation emerged as the most promising approach for addressing the challenges posed by the nature of satellite imagery data. Architectures like UNET [14], MACU [20] and similar manually designed Fully Convolutional Network (FCN) architectures have shown satisfactory results in terms of segmentation accuracy, particularly for larger objects.

Manually-designed networks refer to neural networks that are designed by human experts. This process involves manually testing and specifying the network architecture, the hyperparameters, and the training procedure. Manually-designed networks use to be the predominant method to design the neural network architectures that achieved reasonable performance on a

variety of tasks. However, this process is difficult to design, time and resource-intense, and entails multiple limitations.

The most important limitation is that the performance of these architectures tends to be limited due to the narrow investigation of architectural space. Manually designed networks typically examine only a subset of the vast architectural space due to the finite knowledge, creativity and recourses of the researcher. This limitation can prevent the discovery of innovative architectures that may offer improved performance or efficiency.

Additionally, the performance of the manually-designed FCNs and CNNs tends to diminish when applied to unseen or out-of-distribution data. The architectural choices made during manual design may be biased towards the training data, leading to poor performance on new, unseen samples. The performance is also impacted when training datasets are relatively small (known as low-information environments) resulting in continuous manual re-calibration and configuration testing to adapt the neural network architecture accordingly. Manual network design relies heavily on researcher expertise and domain knowledge, requiring a deep understanding of the problem domain, architectural principles, and relevant techniques. This expertise may not be easily transferable, posing challenges for researchers without extensive knowledge of network design to create optimal architectures.

In contrast, Automated Machine Learning (AutoML) and Neural Architecture Search (NAS) techniques systematically research a broader range of architectural configurations. This dissertation addresses the challenges related to object recognition in satellite imagery for the light-vehicles object class, taking into consideration its unique characteristics, the performance limitations of manually-designed FCNs and the need for fast and accurate object recognition across various dataset types. To tackle these challenges, we leverage NAS as part of an AutoML framework. The NAS technique enables us to automatically search for problem-specific CNN architectures that maximize its performance. Through our research, by leveraging the capabilities of AutoML and NAS, we introduce a novel NAS-MACU neural network, surpassing the performance of manually designed networks to date. This novel approach, NAS-MACU, specifically caters to and can address the limitations of manually designed CNNs.

1.2 Object of the dissertation

The scope of the dissertation is object recognition of light-vehicle class in optical satellite imagery using Deep Learning (DL) and Automated Machine Learning (AutoML) techniques.

1.3 Aim of the dissertation

The aim of the present thesis is to provide solutions to accurate and fast object recognition in satellite imagery employing Deep Learning and AutoML techniques.

1.4 Objectives of the dissertation

The following objectives were set:

1. Conduct an in-depth literature review of wide-range of deep learning-based methods for object recognition of satellite imagery;
2. Propose a deep learning-based framework for improved accuracy and accelerated object recognition (object class: light-vehicles) in satellite imagery including image pre-processing and fully convolutional neural networks (FCNs) design;
3. Perform experimental investigation to assess the accuracy and prediction speed of the convolutional neural network;
4. Perform comparative experimental analysis on the most promising neural networks for object recognition;
5. Design an AutoML-based Neural Architecture Search (NAS) technique suitable for object recognition problems in satellite imagery that can outperform the manually-designed neural networks given problem-specific constraints (e.g., low-information training environments and dataset specificities).

1.5 The scientific novelty of the research

- A deep learning-based framework "Sat-Modification" was proposed for improved accuracy and accelerated object recognition of light-vehicle object class in satellite imagery. The framework includes image pre-processing, pixel-frame sequencing, hyperparameters tuning, network complexity assessment, and UNET architecture adjustment techniques;

- An in-depth comparative analysis and experimental investigation of the top-performing FCNs (UNET, FastFCN, DeepLab, MACU) was conducted and the important features and components of the neural network design were investigated enhancing the performance of the segmentation tasks.
- A novel solution (NAS-MACU) was proposed based on automated Neural Architecture Search (NAS) and MACU network backbone that can automatically discover well-performing cell topology optimised for relatively small-size object recognition (e.g. light-vehicle class) in optical satellite imagery.

1.6 Defended statements

1. The proposed fully convolutional neural network modification based on UNET architecture provides lower network-specific prediction latency for object recognition task in satellite imagery for the light-vehicle object class as compared to other FCNs including MACU, DeepLab and FastFCN networks.
2. The proposed novel NAS-MACU provides a more accurate object recognition for light-vehicle object class in a low-information environment as compared to the manually-designed MACU network that was created and published by expert researchers.

1.7 Practical impact

1. This dissertation research process produced and open-sourced a proprietary satellite imagery training set with labelled polygons to enable further development in this research field. A high-quality training set with 80 316 marked objects using QGIS geospatial software was created using professional data annotation techniques. Labelling and polygon coordinate generation was manually completed by multiple professional annotators and quality cross-checked. An extremely limited amount of publicly available high-resolution satellite imagery datasets with labelled “light-vehicle” object class polygons existed at the time of this research.

2. This work addressed two important practical limitations of satellite imagery application in the algorithmic trading domain: high accuracy and speed in a low-information environment. These real-life obstacles can now be easier to solve with practical techniques suggested in the

dissertation, such as how to measure the computational complexity of the network to improve prediction speed; and how to apply NAS techniques to find the network architecture with the most accurate object prediction when the amount of training data is scarce or expensive (i.e. in a low-information environment).

3. The discovery of NAS-MACU techniques has the potential to greatly benefit scientific researchers since it significantly reduces the time needed to find optimal neural networks for object recognition tasks in specific problem domains, even outside of remote sensing or satellite imagery. This translates into substantial research time savings and domain expertise dependency reduction. It also accelerates models' "time-to-publishing" and to production. Moreover, NAS-MACU enhancements can be extended to other latency-sensitive industrial and humanitarian applications.

1.8 Approbation of the research

The results of the dissertation were published in international research journals with a citation index in the Clarivate Analytics Web of Science (CA WoS) database:

- Gudžius, P., Kurasova, O., Darulis, V., & Filatovas, E. (2021). Deep learning-based object recognition in multispectral satellite imagery for real-time applications. *Machine Vision and Applications*, 32(4), 1-14;
- Gudžius, P., Kurasova, O., Darulis, V., & Filatovas, E. (2023). AutoML-based Neural Architecture Search for Object Recognition in Satellite Imagery. *Remote Sensing,* 25(3), 15-31.

The results of the thesis were presented at the following international conferences:

- 2018: International Conference on Control and Computer Vision (ICCCV), November, Singapore;
- 2019: 16th ACS/IEEE International Conference on Computer Systems and Applications, AICCSA, November, Abu Dhabi, UAE;
- 2019: Data Science, E-learning and Information Systems, December, Dubai, UAE;
- 2022: The 8th International Conference on Machine Learning, Optimisation, and Data Science, June, Siena, Italy.

The results of the thesis were presented at the following national conference:

- 2017: 9th International Workshop on Data Analysis Methods for Software Systems, December 2017, Druskininkai, Lithuania.

1.9 Visual representation of holistic dissertation research process

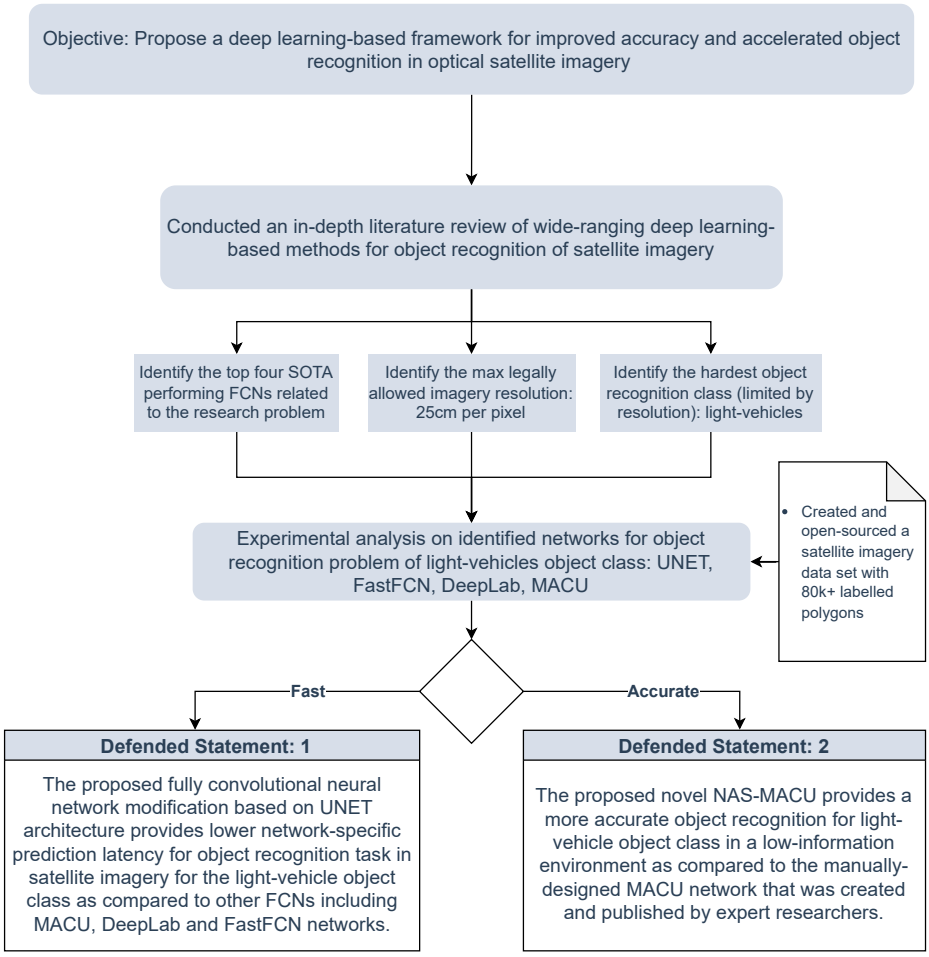


Figure 1.1 Dissertation research process diagram

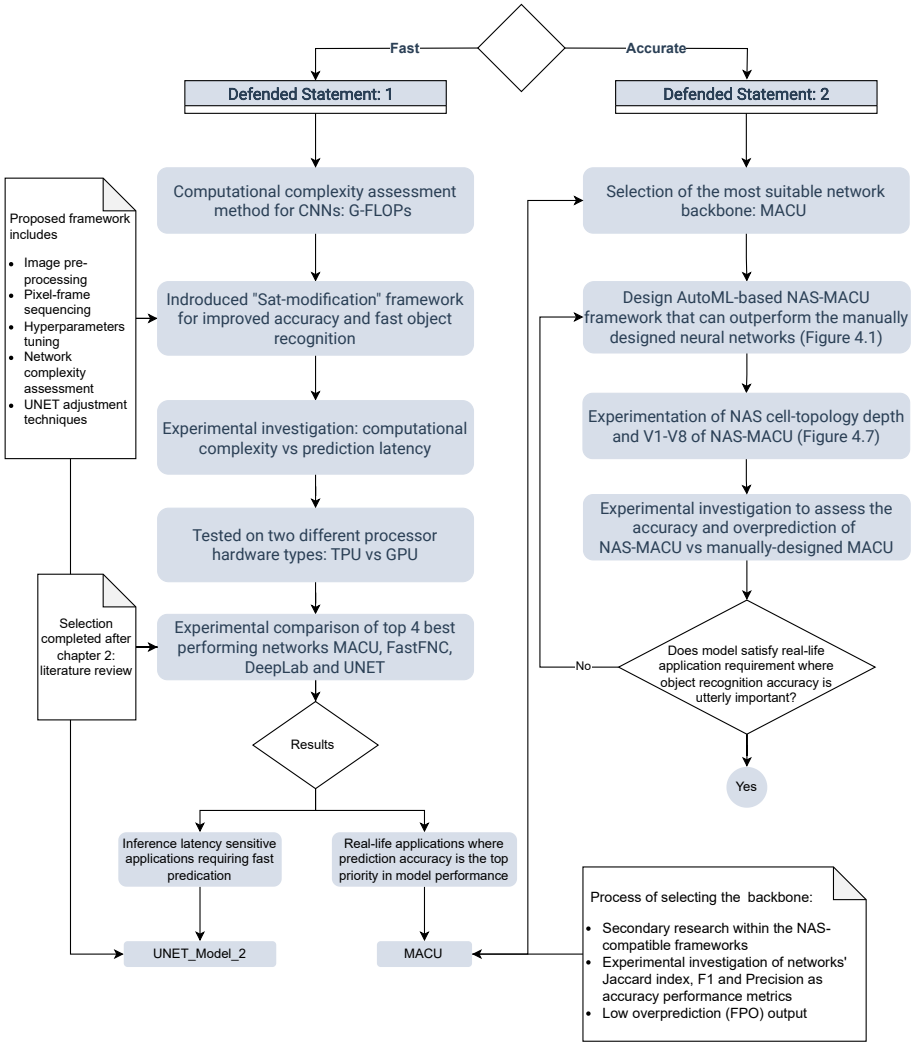


Figure 1.2 Dissertation research process diagram by defended statements

2 LITERATURE REVIEW

This chapter focuses on the research work done in idiosyncrasies of satellite imagery data, CNN, FCN and AutoML models. The chapter also covers practical industrial and humanitarian applications of satellite imagery combined with machine learning technology.

2.1 Satellite imagery

The objects captured in satellite imagery are patches of the earth's surface involving land or ocean. This distinguishes it from terrestrial photos taken by people, surveillance videos taken for safety, the output of automotive sensors used for automatic driving, or medical images used for diagnostics. Rich multispectral aperture, low pixel resolution, and a wide aspect ratio are some of the unique properties of optical satellite imagery [21]. Particularly relevant for machine learning-based object recognition is the spatial resolution of the images. Satellite imagery typically has a lower spatial resolution than on-ground imagery, meaning that each pixel in an image could be almost as large as an object itself. This can make it more challenging to discern small objects such as buildings, cargo ships, trees, or light-vehicles [22].

Another unique aspect of satellite imagery is its multispectral nature. Satellite imagery is often captured in multiple spectral bands, including visible, infrared, and radar. Each band provides information about different scene characteristics, such as the objects' colour, texture, and surface properties. The multispectral nature of satellite imagery can provide additional information for object recognition, yet it also requires specialised pre-processing and feature extraction techniques. Satellite imagery also often covers larger land areas than aerial photography and can be affected by various atmospheric conditions, such as haze and cloud cover, which can impact image quality and make it more challenging to identify objects in the scene. In addition, satellite imagery can be affected by geometric distortions caused by the sensor's position and angle, leading to changes in scale, orientation, and shape of objects in the image. These geometric distortions must be corrected to obtain accurate object recognition results. While satellite imagery presents unique challenges for object recognition, it also offers valuable information that can aid in identifying and classifying objects in the scene photographed from a 600km-800km distance in the Lower Earth Orbit (LEO).

2.2 Semantic segmentation

Image segmentation, like image classification and object detection, is one of the important research areas in the computer vision community. Image segmentation differs from object recognition since object recognition aims to find a bounding box locating the objects, while segmentation tries to find exact boundaries by classifying pixels. The segmentation problem can be divided into two types: semantic segmentation and instance segmentation. Semantic segmentation can be considered a classification problem for each pixel, and it does not distinguish different instances of the same object. On the other hand, instance segmentation also represents a unique label for different instances of the same object [23].

Objects such as light vehicles in satellite imagery are depicted in a relatively small 200-pixel matrix (20 x 10 pixels) in contrast to the millions of pixels processed in more common computer vision datasets like COCO (Common Objects in Context), Pascal VOC (Visual Object Classes) or ImageNet [24]. Given the resolution constraints, we deploy semantic segmentation [21] for the light-vehicle recognition problem to capture these rich multispectral properties. Semantic segmentation assigns a label or category to each pixel in an image. It is used to identify groups of pixels that represent various categories. An autonomous vehicle, for instance, must recognise pedestrians, other cars, traffic signs, pavement, and other road elements. It outputs the semantically interpretable category of each pixel [25] and is more precise than object detection and scene interpretation [26]. Semantic image segmentation techniques originated from the recursive thresholding method [27], spatially constrained k-means approach [28], histogram-based image segmentation, non-parametric clustering, entropic thresholding, and edge detection techniques [29]. These methods are manually calibrated [28] [29], consequently lacking generalisation and scalability [17].

Today, better-performing solutions to the segmentation problem are obtained with deep learning-based solutions compared to the classical Machine Learning (ML) techniques such as Support Vector Machine (SVM) and k-means clustering. While classical methods require feature extraction implemented by the developer, CNN architectures combine feature extraction and classification in the learning phase. One of the first attempts for a deep learning-based semantic segmentation [30] is based on Fully Convolutional Networks (FCNs). The general classification architecture with CNN consists of convolutional and pooling layers to extract features with lower dimensions. In the last layers of these types of networks, fully convolutional layers are

used to make a final decision. On the other hand, in FCN, fully convolutional layers are placed in final dense layers, resulting in the same size output as the input image. Up-sampling is applied to be able to acquire the same resolution frames. Different types of FCN-based architectures were developed since 2015 [31] and some of the covered FCN architectures [32] use pre-trained classification models in the feature extraction stage.

Furthermore, the vehicle recognition problem has received a lot of attention from researchers [33], and CNNs were popularised [34] since they do not require prior feature extraction [17] [35]. A CNN processes data in multiple arrays [36] for example VGG [37] and ResNet [38]; therefore, a multiband satellite imagery dataset is well suited for it by design [39]. Considerable research has been published on implementing semantic segmentation using various CNN architectures [33]. Nguyen et al. [40] presented a five-layer CNN and achieved high object recognition accuracy of 91% for large urban area objects [41]. Later, Chen et al. [42] developed a Hybrid Deep Convolutional Networks (HDCN) architecture for light-vehicle objects and claimed the best performance at the time [42] and significantly surpassed other Hybrid CNN structures such as Hierarchical Robust CNN (HRCNN) using AlexNet as a backbone [35]. It demonstrated that extracting multiscale features is critical to improving the performance of the object detector. HDCN architecture divided the maps of the highest convolutional and max-pooling layer of Deep Neural Network (DNN) into multiple blocks to extract variable-scale features. Building on that work, Yu et al [43] proposed the convolutional capsule network that delivered >90% accuracy, an outstanding result for the vehicle recognition field [43]. A Capsule network consists of capsules made of a group of neurons, unlike conventional CNNs [43].

In addition, Ferdous et al. [44] introduced prediction speed criteria in 2019. They argued that Regions-CNN (RCNN) [45], Fast-RCNN [46], and Faster-RCNN [47] are incompatible with real-time applications due to a slow multistage regional-proposal-based approach. The architectures containing a separate region proposal network required much computational power, and Fast-RCNN reduced the running time of these detection networks. In the Faster-RCNN, the Region Proposal Network (RPN) shares convolutional features with the detection network. Thus, RPN and Fast-RCNN are merged into a single network by sharing the convolutional features. Anchor boxes of multiple scales and aspect ratios are used to classify and regress the bounding boxes in the Faster-RCNN. End-to-end detection-based methods like You

Only Look Once (YOLO) [48] and Single Shot Detectors (SSD) [49] were suggested to increase prediction speed, yet compromising accuracy (only 89.21%) [50].

2.3 Network types and prediction speed

Shelhamer et al. [43] proposed an alternative fully convolutional neural network that combined features from complementary resolution levels (contextual and spatial information). The FCN architecture demonstrated the best precision using semantic segmentation [51] and also improved parameter optimisation and gradient flow, as discussed by Estrada et al. [52]. FCNs applied in satellite images obtained promising results [53]. However, the main issue of FCN is that the resolution of feature outputs is down-sampled with several convolutional and pooling layers and therefore losing contextual data. To eliminate this issue, FCN variants [53] introduce a skip connection from previous layers to enhance the output (for changes in scale) and perform well in remote sensing images. Various more advanced FCN-based approaches, such as SegNet [54], UNET [55], and DeepLab [56], were proposed to address this issue.

DeepLabv1 architecture [57] applies a Fully Connected Conditional Random Field (FCRF) to enhance the poor localisation property of deep networks. Thus, it is more effective to localise segment boundaries than previous methods. DeepLabv2 [56] architecture applies atrous convolution (also named dilated convolution) for upsampling and Atrous Spatial Pyramid Pooling (ASPP) to robustly segment objects at multiple scales. ASPP is a different variant of Spatial Pyramid Pooling (SPP) proposed in the study [58] and aims to improve the accuracy for different object scales. DeepLabv3 [59] augments the ASPP module with image-level features encoding global context and further boosts performance. It improves over previous DeepLab architecture versions and achieves comparable performance with other state-of-art architectures.

The desired output in many visual tasks, particularly in satellite imagery and biomedical image processing, should include localisation; each pixel should be given a class label. Thousands of training images are typically out of reach for biomedical or satellite imagery related research. We rely on the so-called “fully convolutional network”, a more elegant architecture that displays how we modified and expanded this architecture to make it more effective with fewer training images and produce more accurate segmentations. The primary idea is to add additional layers to a typical

convolutional neural network, replacing the pooling operators with upsampling operators. As a result, the output resolution is increased by these layers. High-resolution characteristics from the contracting path are mixed with the output that was upsampled to localize. Based on the knowledge, a subsequent convolution layer can learn to produce a more precise result. We also have many feature channels in the upsampling section, which enables the network to relay context information to higher-resolution layers. As a result, the expansive path produces a U-shaped design because it is roughly symmetric to the contracting path, which the UNET name has been derived from. The segmentation map only comprises the pixels for which the whole context is present in the input image. The network has no fully linked layers and only uses the valid fraction of each convolution. By using an overlap-tile technique, this solution enables the smooth segmentation of arbitrary huge images. The missing context is extrapolated by mirroring the input image to forecast the pixels in the border region of the image [14].

Even with the above-mentioned enhancements, the FCN models, nevertheless, take considerable time (>30 min.) to process ~100km^2 of satellite imagery [12] with accuracy lower or similar to [13] a professional human annotator (~90%) [14] [15] [10] [16]. Also, current academic research is focused on accuracy, and it lacks methods for improving object recognition models suited for more rapid prediction and latency-sensitive use-cases [17] [18]. This gap in research increases the bottleneck for satellite imagery adoption in real-time applications that we are trying to address in this dissertation.

2.4 Algorithmic trading and latency-sensitive applications

One specific latency-sensitive application of satellite imagery combined with machine learning is in algorithmic trading in the financial markets. It is used by hedge fund managers leveraging alternative data, including satellite imagery [3] [19] to generate excess investment returns. Figure 1.3 illustrates data flow in the algorithmic trading system and identifies signal origination latency per data input to accentuate this bottleneck:

- market data (<40 min. delay);
- non-market data (<50 min. delay);
- satellite imagery (>3.5 min. delay);
- research-based metrics (no delay/pre-event).

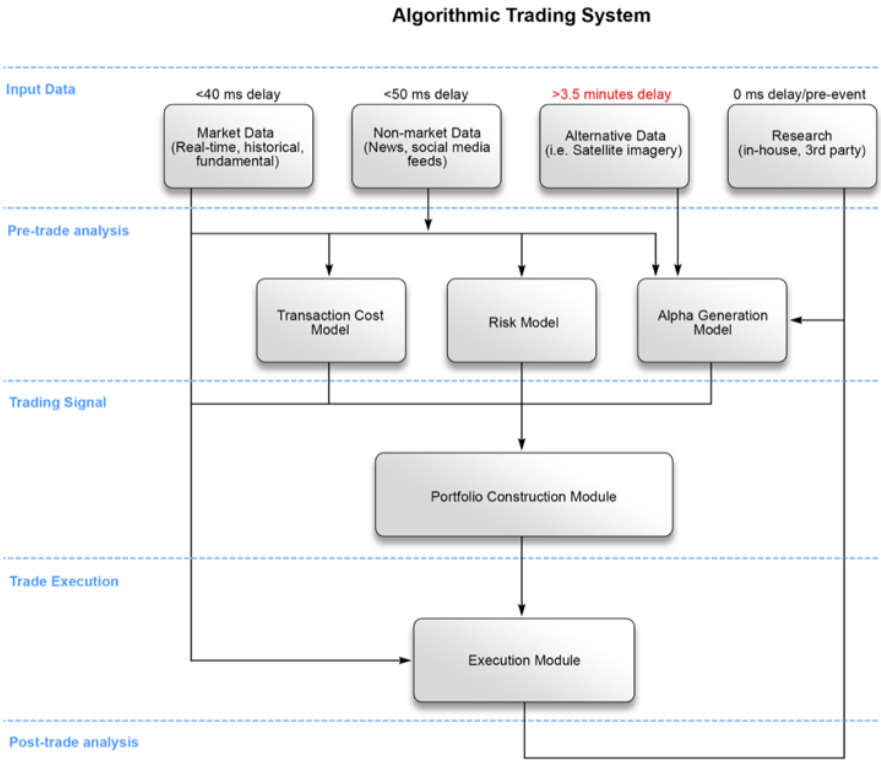


Figure 1.3. Signal generation bottleneck from satellite imagery data in the algorithmic trading system. Modified with an additional layer based on Cliff et al. [19]

The downstream analytics mentioned in the thesis and the examples below refer to various trading activities, including high-frequency and algorithmic trading. The count of cars in an area, obtained through real-time object recognition, provides high-level information that can drive rapid trading decisions. While processing speeds alone may not reach millisecond-level trading requirements, the advantage lies in gaining timely insights for informed decision-making. A 30-second analytical advantage can offer a substantial competitive edge in dynamic financial markets and five examples illustrate where this advantage could be even more significant.

The need for speed in this context is justified by the desire to capitalize on market opportunities before other participants process the same information. Real-time object recognition and vehicle counting enable traders to react swiftly to market events, leading to efficient trading strategies and

potentially significant financial gains. Therefore, the thesis justifies the need for speed in terms of the competitive advantage it provides in rapidly evolving financial markets. Below are specific real-world use cases where accuracy and prediction speed provide this advantage and are worth further investment and research.

- Geopolitical Risk Assessment: Satellite imagery enables real-time monitoring of geopolitical events, such as political unrest, military movements, or infrastructure changes, providing insights into potential risks and opportunities in financial markets. Low-latency object recognition allows for quick identification and counting of relevant objects (e.g. light-vehicles, tanks, military equipment) enabling traders to anticipate market shifts, execute trades faster than competitors, and capitalize on short-term price movements driven by geopolitical factors.
- Production Facility Analysis: Satellite imagery provides a bird's-eye view of production facilities, such as Tesla's Gigafactory, allowing for monitoring car production numbers. Visual cues from satellite imagery, such as the number of vehicles or components in staging areas or storage lots, offer insights into inventory levels, potential production volumes, or shipment activities.
- Commodities Supply and Demand: Real-time counting of vehicles at transportation hubs, such as ports or storage facilities, with the help of satellite imagery, provides insights into the supply and demand dynamics of commodities. Monitoring the total number of vehicles involved in the transportation or storage of specific commodities (e.g., oil tankers, grain transport trucks) enables investors to anticipate shifts in supply or demand. Low-latency vehicle counting allows for informed trading decisions in commodities futures, options, or related stocks, capitalizing on price movements due to changes in supply and demand.
- Natural Disasters and Insurance: High-speed prediction and assessment of natural disasters, such as hurricanes, wildfires, or floods, using satellite imagery aid insurance companies in estimating potential claims and financial impact. Rapid analysis of the extent of damage and affected areas enables informed decisions on risk exposure, claims processing, and pricing, optimizing portfolios for profitability.
- Maritime Trading: Satellite imagery offers real-time information on shipping activities, vessel movements, and cargo flows. Analysing patterns in shipping routes, port activity, and changes in inventory levels identifies opportunities in maritime logistics, international trade, and

commodities tied to shipping, such as iron ore or coal available on CBOE or other liquid exchanges in the form of Futures and Derivatives.

In addition to algorithmic trading, reducing computational complexity and prediction speed in machine learning models offers significant cost and environmental benefits. By reducing computational resource requirements during training and prediction phases, energy consumption is minimized, resulting in lower operational costs and a reduced carbon footprint.

It is worth emphasising that this dissertation aims to reduce neural network-specific inference speed in particular by aiming to measure and reduce the computational complexity of the model. Techniques such as model compression, parameter pruning, computational complexity assessment and efficient network architectures streamline the training process reducing energy consumption. Prioritizing these reductions aligns with the principles of green computing, yielding efficient and sustainable machine learning systems. In addition to the model-architecture-specific computational complexity reduction, other methods include model compression, hardware acceleration, and software optimization:

- Model compression: Quantization is a technique for reducing the precision of the model's parameters, which can significantly reduce the model's size and speed. Pruning is a technique for removing unnecessary connections from the model, which can also reduce the model's size and speed. Knowledge distillation is a technique for training a smaller model to mimic the predictions of a larger model, which can improve the accuracy of the smaller model while also reducing its size;
- Hardware acceleration: Graphics processing units (GPUs) are well-suited for accelerating the inference of deep learning models. Tensor processing units (TPUs) are specialized hardware accelerators designed for deep learning. Field-programmable gate arrays (FPGAs) can be programmed to accelerate the inference of specific models;
- Software optimization: This can include optimizing the model's code by using more efficient programming techniques and libraries. It can also include optimizing the model's runtime environment by using a more efficient runtime environment, such as a high-performance computing (HPC) cluster.

2.4.1 UNET-based models

Segmentation means classifying each pixel in the image. Therefore, the output of the segmentation algorithms is a mask image of the same size as the original image. Ronneberger et al**.** [14] developed FCN called UNET for solving high-level feature extraction in biomedical image segmentation [14] that won a competition at Symposium for Biomedical Imaging [15]. Segmentation architectures consist of two basic stages called encoding and decoding. While the image size is reduced and compressed during the encoding stage, the size increase process is applied to obtain the exact size output from the decoding stage. Biomedical images share similar dimensionality, resolution, and perspective properties with satellite imagery. It was later realised that the overweighting model's higher-level feature extraction (i.e., the object's contours) improves prediction accuracy in both dataset types [14]. Subsequently, UNET was adapted to satellite imagery by Iglovikov et al. [60] and won 3rd place in the Kaggle competition achieving the highest Jaccard coefficient confirming UNET's suitability for this problem [16]. In addition to feature extraction, the network's ability to extract spatial information was researched by Yuan et al. [61]. They discuss the benefits of convolution and deconvolutions similar to the UNET structure. They also introduced a light network structure MobileNet that suggested ideas for light network infrastructures [61].

Besides, neural networks for image segmentation, such as UNET and SegNet**,** roughly consist of encoding and decoding stages. UNET and SegNet architectures transfer the outputs of the encoding layer to the decoding layer by using skip connections. The encoder stage of SegNet consists of 13 convolutional layers from the VGG16 network [37]. The contribution of SegNet is that pooling indices in the max-pooling layers at the encoding stage are transferred to the decoding stage to perform non-linear upsampling. However, UNET transfers the entire feature maps from the encoding layers to the decoding layers so that it uses much memory. Different pre-trained models could be used in the encoding stage of these networks to apply transfer learning. UNET was originally proposed for medical images, however, it also reveals good performance for satellite image segmentation [55]. Different UNET-based architectures are presented in the literature, such as UNET++ [62] and UNET variants like Inception-UNET [63]. UNet++ architecture modifies the skip connections in UNET by adding new layers between the encoder and decoder connection. It uses dense convolutional blocks, and this causes an increase in the number of trainable parameters and floating-point

operations (FLOPs). Inception variants of UNET apply the inception [64] approach differently and enhance the feature extraction stages. Using the Inception approach in each layer increases the computational cost excessively. Therefore, it is much more convenient not to use inception in all layers. INCSA-UNet [65] uses inception block with DropBlock module only in the encoding stage and adds spatial attention modules to prevent overfitting and enhance important features by focusing on key areas, respectively. The INCSA-UNet architecture was evaluated against Inception-based architectures, UNet++, and classic UNET for the problem of building segmentation from aerial images and performs well. UNet3+ [66] has fewer parameters than UNET and applies a hybrid loss function for position and boundary-aware segmentation maps.

Besides, the loss function is also crucial in training machine learning models. It measures how well a machine learning model can predict the expected output. The loss function takes in the predicted output of the model and the ground truth and returns a value that indicates how well the model is performing. Training a machine learning model aims to find the model parameters that minimise the loss function. Loss functions used in image segmentation problems typically fall into one of two categories: pixel-wise loss functions and region-based loss functions. Pixel-wise loss functions operate on a per-pixel basis and generally are used to predict the probability of each pixel belonging to a certain class. Examples of pixel-wise loss functions include binary cross-entropy and mean squared error.

On the other hand, region-based loss functions operate on a per-region basis and are used to evaluate the performance of the segmentation model on a larger scale. Examples of region-based loss functions include the Dice coefficient and the Jaccard index. These loss functions are often used in conjunction with pixel-wise loss functions to provide a complete evaluation of the segmentation model. The Dice coefficient [67] is a measure of the overlap between the predicted segmentation and the ground truth segmentation. It is widely used in the training and evaluation of segmentation models. Inception-UNet and INCSA-UNet use a Dice loss function, UNet3+ proposes a hybrid loss function, Hybrid UNET uses a binary cross-entropy loss function, and FCAU-Net uses a combination of cross-entropy and dice loss functions. Abraham, N., & Khan, N. M. [68] proposed a generalised focal loss based on other Tversky indexes and compared it with the dice loss using UNET and attention UNET architectures. They studied lesion segmentation and aimed to address the data imbalance issue.

Continuing with the UNET variations, UNet3+ also combines multi-scale features by re-designing the interconnection between the encoder and the decoder. UNet3+ is only tested on medical images. Figure 2.1 shows a rough comparison of the three architectures: UNET, UNet++ and UNet3+.

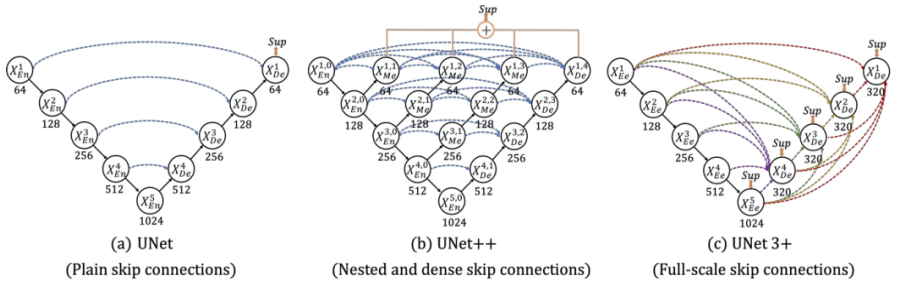


Figure 2.1. Comparison of UNET, UNet++ and UNet3+ [65]

In addition, MANet [69] is a semantic segmentation network containing multi-scale context extraction and adaptive fusion modules. Multi-scale context extraction module has atrous convolutions with different dilation rates in parallel. The adaptive fusion module, inspired by "Squeeze-and-Excitation" blocks [70], contains a channel attention mechanism to determine more valuable features. It aims to handle the problem of different target sizes for remote sensing images. HRNet [71] seeks to eliminate the problem of losing spatial information in the encoding stage. It applies multi-branch parallel convolutions and produces four feature maps at different resolutions. TransUNet [72] and Swin-UNet [73] intend to use the power of transformers for medical image segmentation problems. Transformers have demonstrated great success in Natural Language Processing, and computer vision researchers have used transformers for computer vision problems. Transformers were first proposed for sequence-to-sequence tasks in natural language processing by Vaswani et al. [74] in 2017. Transformers use self-attention mechanisms; inspired by it, researchers have proposed different CNN architectures to implement self-attention within CNN. The full-resolution image is divided into small patches to use transformers in computer vision tasks, and the patch sequence is passed to the transformer model. Vision Transformer [75] achieved state-of-the-art classification accuracy by applying a pure transformer to the sequences of image patches.

Furthermore, TransUNet is based on an attention UNet architecture, and the TransUnet transformer encodes the image patches from a CNN feature map as input for the transformer layer. The decoding part upsamples the encoded features to obtain the final mask. Thus, it indicates that the

transformer can be used as a powerful encoder for medical image segmentation tasks. The Swin-UNet builds on top of TransUNet architecture and uses the Swin Transformer, a type of Vision Transformer, as a backbone to build a pure U-shaped architecture. While transformers have been integrated into the UNET architecture in recent studies, re-designed UNet architecture studies of enhanced skip connections are continuing. Different feature fusion techniques and skip connections are applied to improve the UNET architecture in the literature further. Phan et al., 2021 [76] re-designed the decoder sub-network of UNET and proposed a multitasking architecture. It aims to perform three tasks: lesion segmentation, boundary distance map regression, and contour detection. It also suggests a new skip connection module. Additionally, Lee et al. [77] tested different skip connections between the encoder and decoder sub-networks of UNET architecture for microplastic segmentation. A recent Hybrid UNET [78] proposed a multi-scale skip-connected segmentation network for high-resolution satellite images. In contrast, UNET fuses the features from the same scale between the encoder and decoder, while Hybrid-UNet fuses coarse and fine semantic feature maps from both the decoder and encoder sub-networks. It designs an additional decoder sub-network and fuses features of both decoder sub-networks to obtain a final semantic segmentation mask.

The attention UNET architectures can be considered an augmentation of UNET architecture with attention blocks proposed in SA-Unet [79] and also used in INCSA-Unet. SA-UNet is an UNet-based architecture containing a spatial attention module for feature refinement and dropout blocks to prevent overfitting. The spatial attention module helps to focus on key regions, while channel attention is used to enhance important feature channels. The channel attention module used in the INCSA-Unet (both spatial and channel attention mechanisms) performs well in different architectures like MACU, SENet [80], and DANet [81]. A multiscale UNET study [82] proposes an architecture to merge the low-level and abstract features extracted from the shallow and deep layers. It aims to retain detailed edge information for building segmentation issues.

MACU [20] is another UNET-based architecture using multiscale skip connections and asymmetric convolution blocks. The skip connection in UNET and its variants bridges low-level and high-level features. This approach and multiscale feature extraction make significant performance improvements in the segmentation task. In addition to that, the attention modules with an encoding-decoding structure have been widely used for fine-resolution image segmentation. Multiscale skip connections at MACU

architecture are introduced together with with channel attention blocks and asymmetric convolution blocks built on the UNET backbone. It enhances the standard convolution layers with the asymmetric convolution block involving branches of the square, horizontal, and vertical kernels. The experiments on remote sensing datasets have demonstrated the effectiveness of MACU [20].

An alternative model for capturing spatial object features similar to MACU is the Coordinate Attention (CA) mechanism [83]. In this approach, the spatial and channel information is captured by embedding positional information into channel attention. FCAU-NET [84] uses the advantages of CA in the encoding stage, Asymmetric Convolution Block (ACB) in the decoding stage to enhance the extracted features, and Refinement Fusion Block (RFB) to combine low- and high-level features, however, it did it result in promising performance in the segmentation problems. Experimental results on two remote sensing image datasets reveal that MACU outperforms [20] architectures like FCAU-NET, PSPNet [85], and TransUNET [72], producing a similar performance to DeepLabv3 and FastFCN that were selected for further investigation and most promising network architectures.

2.4.2 Summary of manually designed networks

To conduct a chronological review of how manually-designed CNN, FCN, and UNET network topologies have evolved over time, a summary of the networks, their architecture main feature, and unique approach are provided in Table 2.1 together with their release year.

In Table 2.1, we present a comprehensive breakdown of manually designed neural network architectures specifically developed for semantic segmentation. These architectures were meticulously designed to enhance the accuracy, efficiency, contextual understanding, robustness, and generalization capabilities of semantic segmentation tasks. Notably, these architectures introduce novel design choices such as skip connections, attention mechanisms, and multi-scale context extraction to enhance segmentation accuracy. They also optimize computational efficiency and memory usage. By capturing contextual information effectively, the architectures improve the model's understanding of image context. The development of these architectures represents the progression of research in semantic segmentation, setting the foundations for further advancements in the field.

Table 2.1 Breakdown of manually designed neural networks for semantic segmentation

Architectures	Year	Unique approached deployed
UNET [55]	2015	Use skip connections from down-sampling layers to up-sampling
DeepLabv1 [57]	2016	Use a fully connected Conditional Random Field (CRF)
SegNet [54]	2017	In skip connection, SegNet transfers only pooling indices to use less memory
PSPNet [85]	2017	Use dilated convolutions and pyramid pooling module
DANet [80]	2017	Its position and channel attention modules followed by ResNet feature extraction
UNET++ [62]	2018	Improved skip connections from down-sampling layers to upsampling
DeepLabv2 [56]	2019	Use atrous or dilated convolution and fully connected CRF together
MACU [86]	2019	Has multiscale skip connections and asymmetric convolution blocks
SA-UNET [79]	2020	Applies spatial attention module and structured dropout convolutional blocks within the UNET architecture
UNET3+ [66]	2020	Modifies skip connection and fewer parameters compared to the UNet++. Proposes hybrid loss function
MANet [69]	2020	Proposes adaptive fusion module with channel attention and multi-scale context extraction module for remote sensing images
HRNet [71]	2020	Proposes multi-branch parallel convolutions
DeepLabv3 [59]	2021	Improved atrous spatial pyramid pooling (ASPP)
Inception-UNET [63]	2021	Uses Inception modules instead of standard kernels (wider networks)
Swin-UNet [73]	2021	Build U-shaped segmentation architecture based on the Swin transformer block
TransUNET [72]	2021	Transformers encode the image patches in the encoding stage
FastFCN [87]	2021	Contains fully convolutional network layers
INCSA-UNET [66]	2021	Use drop block inside Inception modules, and also apply attention between encoding and decoding stages

Architectures	Year	Unique approached deployed
Hybrid Unet [78]	2022	Builds a hybrid UNET with additional decoder sub-networks and introduces high-resolution satellite images dataset.
Lee et al. [77]	2022	Different types of skip connections are tested for UNET architecture to evaluate the effect of the skip connections.
FCAU-NET [84]	2022	Contains attentions, asymmetric convolution blocks to enhance the extracted features and refinement fusion block (RFB) in skip connections

2.5 AutoML and Neural Architecture Search

Pre-eminently performing neural network architectures are currently designed by scholars and practitioners. These networks are manually tailored to a certain type of imagery and resolution. Therefore, if the training set topology is vastly different from what the network was based on at inception, the performance drops even after extensive training [18]. Developing CNN architectures and experimentation requires adjusting the CNN hyperparameters, cell topology, and architecture modifications which can take months and, in some cases, years to reach a satisfactory result [19]. The research and architecture design process for semantic segmentation-related case studies is labour-intensive as well as time and resource consuming [88].

In addition to the limitations of human researcher capabilities, another major problem in ML for object recognition, especially in the satellite imagery domain, is the lack of available training and test data. In satellite imagery, this problem arises due to the low number of high-resolution optical imagery satellites operating in the Lower Earth Orbit, high-cost constraints, and therefore the availability of publicly available datasets required for training is considerably limited [21].

Manually built network design includes theoretically pre-select hyperparameters, e.g., the activation function forms, the numbers of network layers and nodes in each layer, and connection manners between different layers, all requiring human expertise, subjective judgment, and experimentation. An effective neural network architecture design often requires substantial knowledge of the particular domain and lengthy manual trialling [89]. The process of network component experimentation can take months and, in some cases, years to reach the required result [90] [91]. Researchers encounter limitations such as the design process being time-

consuming and labour-intensive. This brings great difficulty when building a high-quality machine learning system in practice and therefore limits ML applications [22].

As a part of the AutoML, Neural Architecture Search aims to solve this problem and make the process of purpose-built neural network design accessible to a wide range of domains and a larger quantity of researchers. NAS essentially aims to do tuning a neural network faster and more effectively. Therefore, in recent years NAS has become an active research topic [92]. Specifically, NAS represents a technique for automating the design of artificial neural networks [25] instead of conventional hand-designed ones [26] and has recently obtained gratifying progress [27]. NAS neuron cell-level search space has been looked into for various broader architecture types, including NAS-UNET [93].

The objective of NAS is to remove the manual and high-technical knowledge requirement and do the work of a human manually tuning a neural network significantly faster and more effectively. NAS belongs to a deep learning methods group known as meta-learning. Meta-learning includes an auxiliary search algorithm to design the characteristics of a neural network. These characteristics are inside the neural network, such as activation functions, hyperparameters, or NAS-based search space investigating the cell-level topology.

The NAS search space is used to find the best architecture, while a performance estimation method is used to score the performance of a network. The cell-level search aims to examine combinations of basic building blocks, known as “cells”, to form a larger, more complex neural network. A cell is a small, self-contained neural network typically consisting of several layers, such as convolutional layers, pooling layers, and normalisation layers. Cell-level NAS algorithms work by repeatedly stacking together different combinations of cells to form a neural network. The algorithm then evaluates the performance of each generated architecture on a specific task, such as image classification or object detection.

Furthermore, the algorithm uses this evaluation to guide the search for better architectures. It involves searching for the best combination of cells to use in the network. Cell-level NAS algorithms have the advantage of being more computationally efficient than other NAS methods, as they only need to search through a reduced set of possible combinations of cells. It also allows

the search to be more focused, as it only searches for the optimal combination of cells rather than the entire architecture.

It is worth noting that there are different ways to implement cell-level search. Some methods use a fixed set of cells and search for the best combination, while others generate new cells during the search. Also, the search can be done using different optimisation techniques such as reinforcement learning (RL) [94], evolutionary algorithm (EA) [95], Bayesian optimisation method [96], and gradient-based method [97]. As first attempts, most NAS algorithms were based on RL or EA. A controller produces new architectures in RL-based methods, and the controller is updated with the accuracy of the validation dataset as the reward. However, RL-based methods typically require significantly higher computational resources [98]. The gradient-based methods use the search space as a continuous space and search the architectures based on the gradient information. The gradient-based algorithms are more efficient than the RL-based algorithms. The EA-based algorithms apply evolutionary computation to solve the NAS issue [99].

Hitherto, NAS research has been conducted predominantly on image classification problems [100]. Several papers have proposed methods introducing NAS search space for encoding-decoding-based architectures similar to UNET for medical image segmentation. NAS-UNET [93] selects primitive operation sets within cells by using Differentiable Architecture Search (DARTS) [101], while C2FNAS [102] tries to find the best topology followed by the convolution size within cells by using a topology-similarity-based evolutionary algorithm. Figure 2.3 shows the NAS-UNET architecture with cell-based architecture search space. NAS-UNET uses primitive operations that performed well in the literature. It proposes two cell architectures named DownSC and UpSC.

On the other hand, C2FNAS is proposed for 3D medical image networks, which require a huge amount of memory. C2FNAS has two stages for the search technique: macro-level topology search and cell-based micro-level operations search. Macro-level search defines how the cells are connected and stacked within an broader model architecture. In micro-level search, the primitive operations are selected. In the research [103], the authors first create a configuration pool from advanced classification networks for better cell configuration instead of searching for a cell from scratch. Thus, it prevents overgrowth of the search space caused by searching from scratch while adding well-known methods to the search pool. However, it should be noted that this method largerly depends on the selected network backbone

type. Considering that different network types can give better results in various problems, it can also cause a disadvantage depending on the problem. The combination of both types of micro-level and macro-level search described above is called a Mixed-Block NAS (MB-NAS), and broader model architecture-level search is followed by cell-level search in this method. It uses a search algorithm called Local Search [104], yet is extremely computationally intensive and therefore contains practical implementation limitations.

More effective approach is using cell-level search and for example DARTS uses an efficient strategy over a continuous domain by gradient descent. The limitation of that is that its performance often drops due to overfitting in the search phase. To avoid it, NAS-HRIS [105], GPAS [106], and Auto-RSISC [107], which are based on a gradient descent framework, have been proposed for remote sensing scene classification issues. NAS-HRIS uses the Gumbel-Max trick [108] to improve searching efficiency. NAS-HRIS undergoes evaluation for remote sensing image segmentation problems and demonstrates superior performance compared to existing methods in the literature. Notably, it is recognized as the pioneering NAS study specifically focused on high-resolution remote-sensing image segmentation. NAS-HRIS employs a U-shaped encoder-decoder structure, wherein the encoder architecture is searched within a cell-based search space.

GPAS utilizes a greedy and progressive search strategy to enhance the correlation between the search and evaluation stages. On the other hand, the auto-RSISC algorithm aims to reduce redundancy within the search space by sampling architectures in a proportionate manner. Thus, Auto-RSISC requires fewer computational resources, limiting the model's performance by reducing the architecture diversity. RS-DARTS [109] adds noise to suppress the skip connections and aims to close the gap between training and validation. It applies the same approach as Auto-RSISC to speed up the search processing. RS-DART demonstrates an competitive performance in remote sensing scene classification while reducing computational overload in the search phase, however still falling short from NAS-UNET shaped structures' performance.

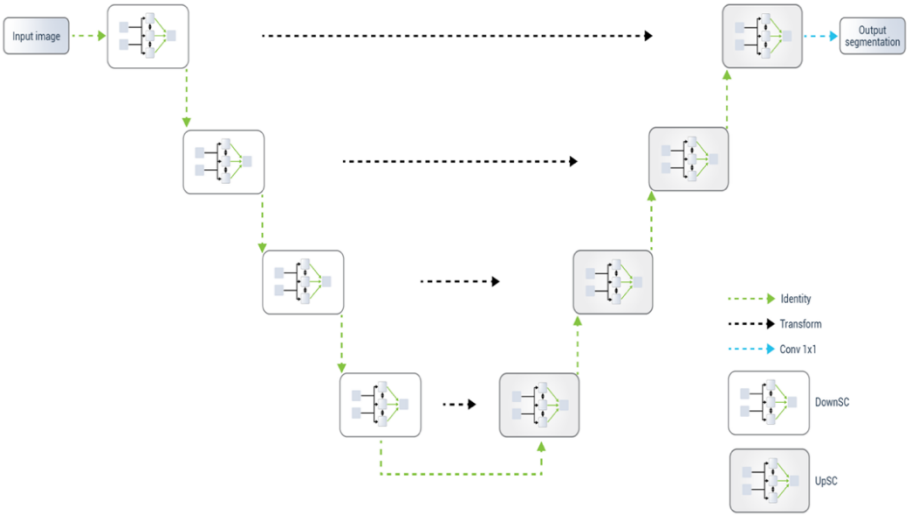


Figure 2.3. NAS-UNET architecture with primitive operation sets of DownSC and UpSC

The research by Weng Y et al [93] captures the recommendations for effective semantic segmentation tasks and develops a NAS-UNET search methodology as an effective NAS for remote sensing as depicted in Figure 2.3. Satellite imagery presents challenges such as topology variations, limited data availability, and the need for specialized architectures. Manual network design requires extensive expertise and time-consuming experimentation. NAS techniques automate architecture design by investigating combinations of self-contained neural network cells. These cells, consisting of convolutional, pooling, and normalization layers, enable the construction of tailored architectures for satellite imagery object recognition.

The reviewed NAS techniques, including NAS-UNET and NAS-HRIS, address these challenges. NAS-UNET employs a differentiable architecture search to select primitive operations within cells, enhancing search efficiency and addressing the main objectives of the present dissertation.

2.6 Outcome of the literature review

In conclusion, the UNET and MACU networks were selected as the most promising architectures for conducting experimental investigations considering the stated objectives of the dissertation. The UNET architecture, originally developed for biomedical image segmentation, showed promising

performance in satellite imagery as well. Its unique design, with skip connections and upsampling operators, allows for the extraction of high-resolution features and effective localization. UNET-based variations, such as UNET++, Inception-UNET, and UNet3+, have further improved upon the original architecture by introducing additional layers and enhancing feature extraction.

On the other hand, the MACU network stands out with its multiscale skip connections, asymmetric convolution blocks, and the integration of attention mechanisms. MACU demonstrated superior performance in remote sensing datasets, surpassing other architectures like FCAU-NET, PSPNet, and TransUNET, while achieving comparable results to DeepLabv3 and FastFCN. The inclusion of channel attention and asymmetric convolution blocks in the UNET backbone enhances the feature extraction process and effectively captures spatial and channel information.

Both UNET and MACU networks offer promising solutions to image segmentation tasks, particularly in satellite imagery and biomedical image processing. Their ability to handle high-level feature extraction, spatial information, and multiscale context makes them suitable for accurate and precise segmentations. By conducting experimental investigations using these architectures, further insights can be gained into their performance and potential improvements, ultimately advancing the field of image segmentation and its applications in various domains.

Additionally, Neural Architecture Search (NAS) is a promising approach in Automated Machine Learning (AutoML) that addresses the limitations of manually designed neural network architectures. NAS automates the process of designing neural networks, making it accessible to a wider range of domains and researchers. By investigating combinations of basic building blocks called "cells," NAS constructs complex neural networks more efficiently. It employs various optimization techniques such as reinforcement learning, evolutionary algorithms, and gradient-based methods. NAS has been successful in image classification and has shown potential in medical and satellite imagery segmentation and remote sensing. Further research in NAS and its application in AutoML holds promise for advancing machine learning systems across diverse domains. NAS and its relative performance compared to manual networks will be explored in more depth in this dissertation.

3 MANUALLY DESIGNED NEURAL NETWORKS FOR OBJECT RECOGNITION IN SATELLITE IMAGERY

This chapter focuses on the satellite imagery data overview, the dataset's unique properties and limitations, and pre-processing and augmentation techniques. Following that, the chapter covers two of this dissertation's three most important research areas: object recognition accuracy and prediction speed and evaluates manually designed neural network architectures such as UNET, MACU, DeepLab and FastFCN.

3.1 Problem definition

In the present thesis, the object recognition problem is being solved. We derive object recognition results using semantic image segmentation metrics. Due to the low-resolution nature of satellite imagery, the semantic segmentation technique is suitable for object recognition in satellite imagery problems because it provides the most granular, pixel-level performance.

The object class selected for empirical investigation is "light vehicle" class. Objects in this class are as small as 200 pixels (20 x 10-pixel matrix compared to millions of pixels in common images sourced from the ImageNet for example), as illustrated in Figure 3.1. Therefore, each pixel should provide valuable information.

Figure 3.1. Semantic pixel-level segmentation of the "light vehicles" object class in satellite imagery. Input image (left) and output image (right)

In Figure 3.1, blue colour pixels represent the “light-vehicle” object class recognised by the segmentation technique; red colour represents the original annotator-marked object polygon, and white colour represents the accurate match for per-pixel prediction. Semantic segmentation can be considered a classification problem for each pixel since we classify it in binary output (object within a class or no object), and it does not distinguish different instances of the same object.

We obtain object recognition metrics from semantic image segmentation results and overlay the segmented “light vehicle” pixels against human annotator-derived masks (polygons) in the data sets (training, validation, and testing datasets). Then, we derive which objects were correctly recognised and which ones where not. There can be significant variability in how an object can appear in different contexts, lighting conditions, angles, etc. A lower threshold can allow for more flexibility in recognizing an object despite these variations as well as manual annotator errors. Therefore, at least 25% of pixels of an object have to identically overlay for an object to be considered correctly recognised.

This threshold was selected to adjust for human annotator labelling inaccuracies in the dataset (as evident in Figure 3.1) as well as the required minimal threshold. Upon empirical investigation (multiple levels between 15% and 40%), we discovered that objects that match 25% of the annotated polygon are sufficient to classify that the object was correctly recognized at the same time generating minimal false positive signals.

Once the object is correctly recognised, it is then counted as a True Positive object (TP) or otherwise appropriately classified as either a False Positive object (FP), False Negative object (FN), or True Negative object (TN). Based on these fundamental numbers, other performance metrics were also derived. Those metrics reflect performance in both semantic segmentation and object recognition. For consistency, both semantic segmentation and object recognition metrics are used to measure neural network performance.

3.2 Metrics

To quantitatively evaluate object recognition results, the following metrics were used and derived: True Positive objects (TP), False Positive objects (FP), True Negative objects (TN), False Negative objects (FN), Jaccard Index, Recall, Precision, Overprediction error (FPO), and F_1 as the overall accuracy metrics. The metrics are categorised into two categories; one is for image segmentation metrics (Jaccard Index), and the second is for derived object recognition metrics as described in Subsection 3.1. Metrics overview:

- TP reflects the number of objects ("light vehicles") correctly detected as compared to the "ground truth" – the actual object in the imagery;
- FP demonstrates the number of objects ("light vehicles") incorrectly detected as compared to the "ground truth";
- TN reflects the number of objects that correctly predicted an absence of the object;
- FN demonstrates the number of objects detected being not an object where there was an object ("light vehicle");

Jaccard index (see Eq. (1)) is a pixel-level segmentation accuracy metric of semantic segmentation:

$$Jaccard\ index_c = \frac{TP_c}{TP_c + FP_c + FN_c}. \quad (1)$$

Where TP_c is the number of "True positive pixels" in class c across the entire data set; FP_c is the number of "False Positives pixels" in c; FN_c – "False Negatives" in c.

Recall (Sensitivity) is the ratio of correctly predicted objects to all observations in the actual class (see Eq. (2)):

$$Recall = \frac{TP}{TP+FN} \quad . \quad (2)$$

Precision (Positive predictive power) is the relation between true positives and all positive predictions (see Eq. (3)):

$$Precision = \frac{TP}{TP+FP} \, . \quad (3)$$

Accuracy (TPO) measures the proportion (in %) of objects correctly detected as compared to the total number of labelled objects within the prediction set (see Eq. (4)):

$$TPO = \frac{TP \times 100}{Number\ of\ labelled\ objects}. \quad (4)$$

Overprediction (FPO) measures an overprediction error, i.e., the percentage of objects recognised by the network, not by the annotator as compared to the total predicted objects by the network (see Eq. (5)):

$$FPO = \frac{FP \times 100}{TP + FP}. \quad (5)$$

F_1 combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is considered the best overall accuracy performance identifier (Eq. 6):

$$F_1 = \frac{2TP}{2TP + FP + FN} = \frac{2\ x\ Precision * Recall}{Precision + Recall}. \quad (6)$$

3.3 "Sat-Modification" framework overview

To fulfil one of the primary objectives of the dissertation, which involves proposing a framework based on deep learning and satellite imagery modification for enhanced accuracy and expedited object recognition, we implemented numerous advancements throughout the entirety of the pre-processing and network design stages (i.e. "Sat-Modification framework"). Collectively, these improvements facilitated the attainment of state-of-the-art network performance results, thereby successfully achieving the aforementioned objective. The entire process, from satellite imagery acquisition (P1) to end-signal generation and delivery (P13), is depicted in Figure 3.2. Components from P1 to P4 and P10 represent the dataset and satellite imagery-specific processes such as data acquisition, pre-processing, augmentation, etc. These components are described in subsections 3.2.1 – 3.2.4 inclusively.

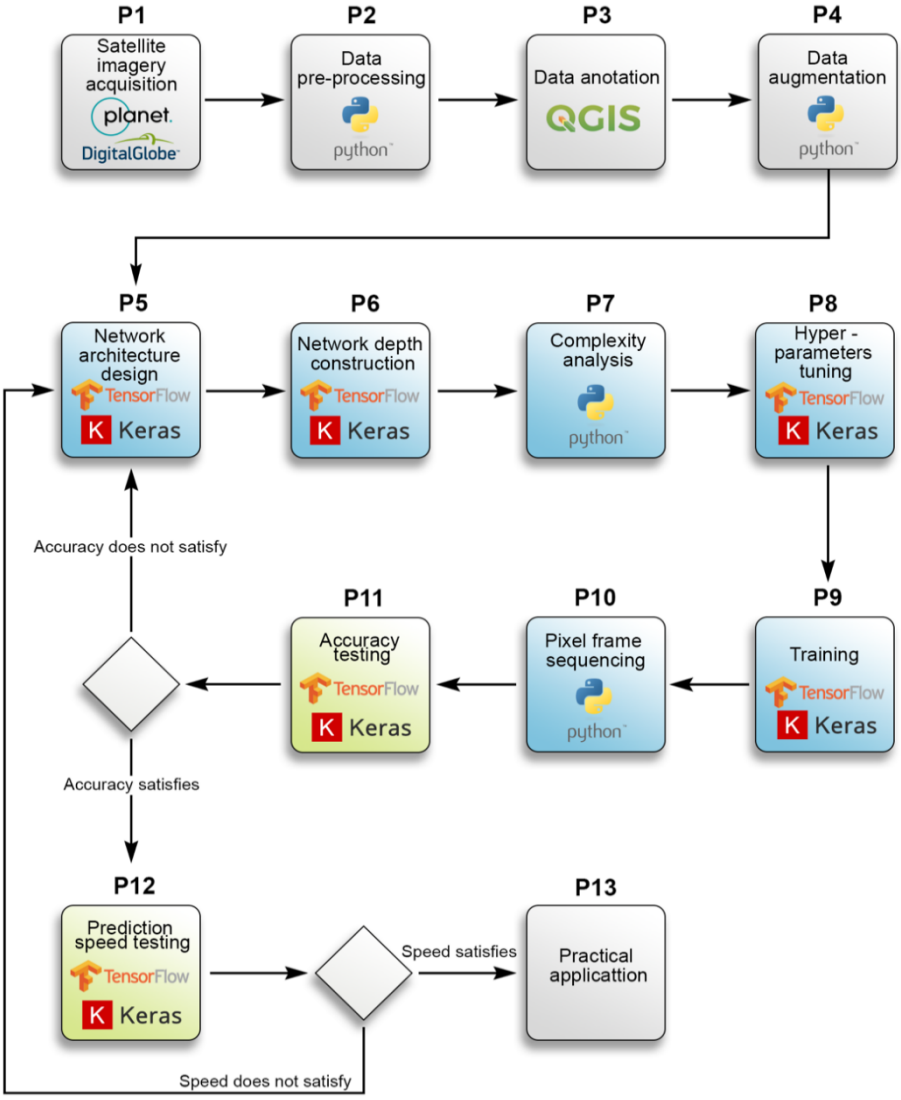


Figure 3.2 Schematic workflow diagram for the entire process of object recognition in satellite imagery. Steps P5 – P10 represent the "Sat-Modification" Framework

Components from P5 to P9 represent the areas of advancement proposed in the thesis and are described in Section 3.2. We discuss two main areas of research: 1) Network depth construction and feature extraction for prediction accuracy and 2) Computational complexity analysis for Prediction Speed.

3.3.1 Raw satellite imagery

This section corresponds to P1 stage in Figure 3.2. The underlying imagery in the dataset was produced by the DigitalGlobe WorldView-3 satellite and is available via an open-source raw satellite imagery database SpaceNet. The SpaceNet database offers a large collection of multi-band high-resolution raw imagery along with validated building footprint and road network annotations [110]. This dataset did not provide "light vehicle" object annotations.

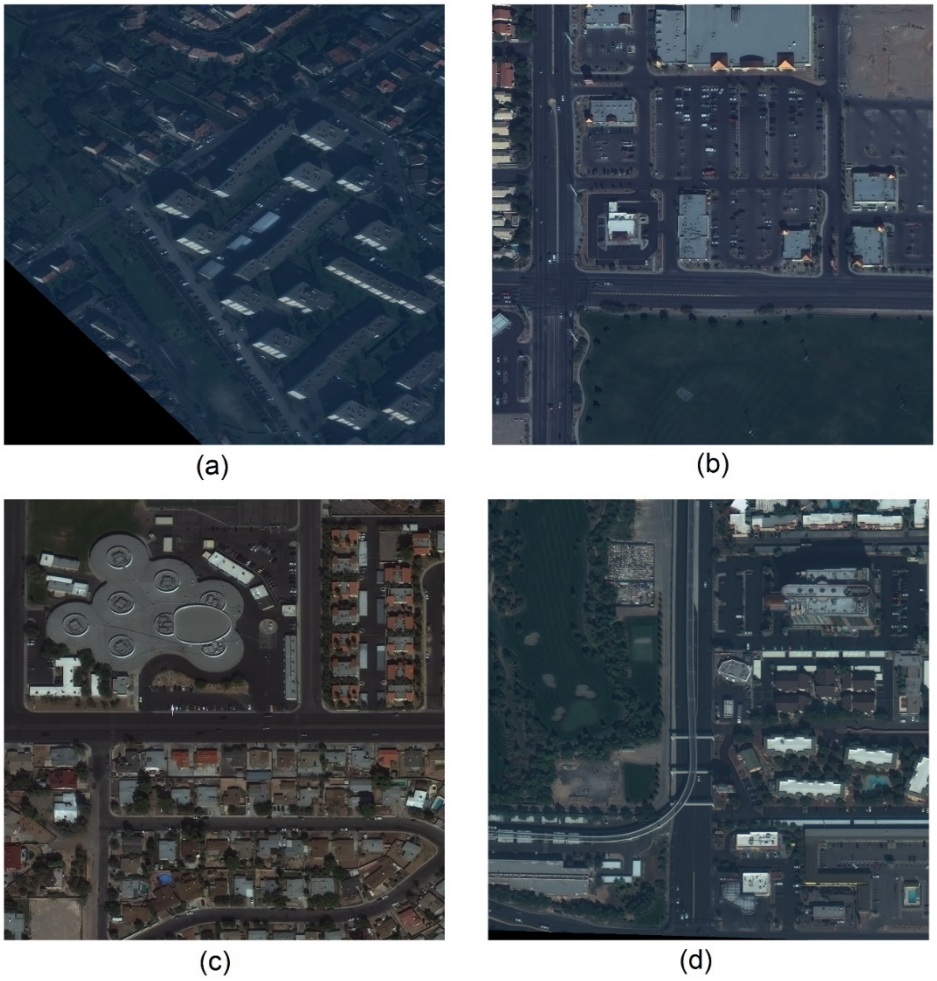


Figure 3.3 Sample images produced by DigitalGlobe WorldView-3 satellite

A total of 250 (125 augmented as discussed in subsection 3.3.3) high-resolution (30cm per pixel) multispectral satellite images, equivalent to 50km^2 AOI of Paris (in Figure 3.3.a), Shanghai (Figure 3.3.b), Las Vegas (Figure 3.3.c), and Khartoum (Figure 3.3.d), were used for training/validation (80%) and testing (20%). The annotated satellite imagery dataset used in the experimentation of the present dissertation research was derived and augmented from the SpaceNet. However, it is important to mention that the imagery used from the SpaceNet was raw and not annotated. The annotation with object polygons marking was done manually in preparation for this research and is described in detail in the next subsection 3.3.2.

3.3.2 Annotated dataset

This section corresponds to P2 stage in Figure 3.2. A total minimum of 350 hours of manual annotation work were conducted to prepare a high-quality training set with 80 316 labelled objects in the light-vehicles object class. Images were annotated using QGIS geospatial imagery software (figures 3.4 and 3.5). Labelling and polygon coordinate generation was manually completed by multiple professional annotators and quality peer-reviewed and cross-checked. No publicly available high-resolution satellite imagery datasets with marked "light-vehicle" object classes existed at the time of this research. Therefore, we open-sourced and published our in-house developed proprietary dataset with marked polygons online to enable further development in this research field [111]. Figures 3.4 and 3.5 demonstrate the snapshot in the process of creating the labelled objects of the "light-vehicle" class using QGIS.

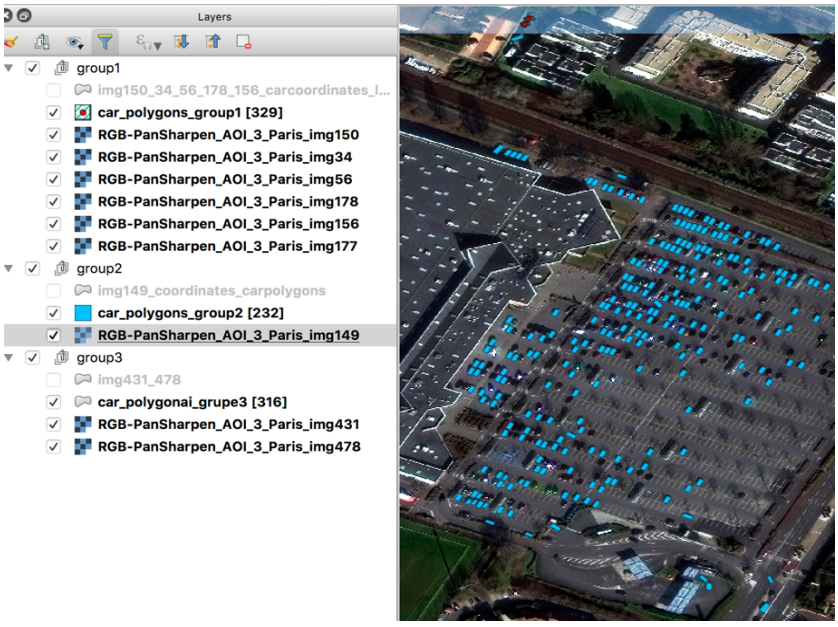


Figure 3.4 Car polygons in the satellite imagery of a parking lot in Paris

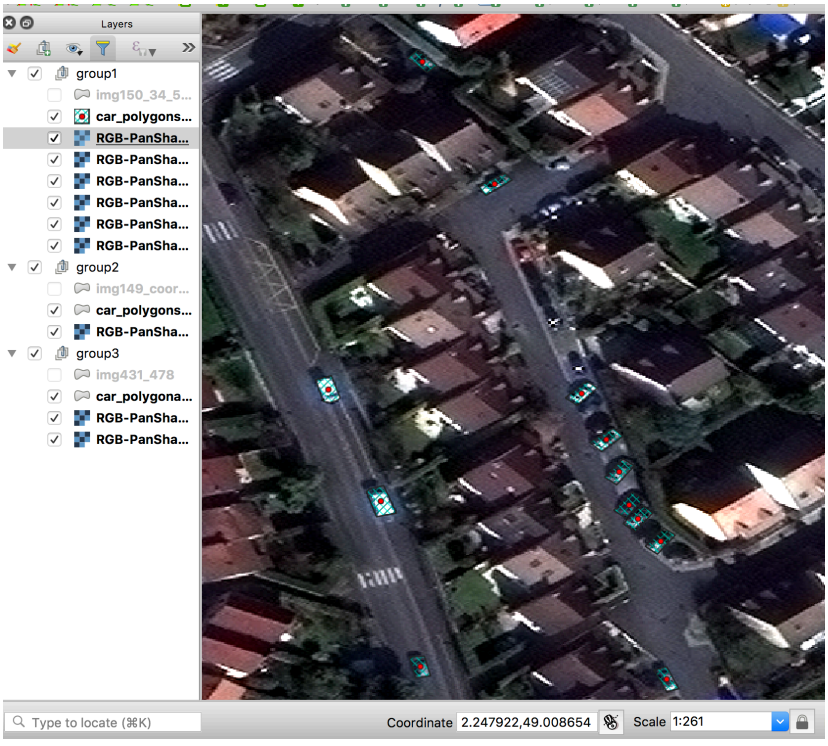


Figure 3.5 Car polygons in the satellite imagery of a residential area in Shanghai

3.3.3 Data Augmentation

This subsection corresponds to P4 stage in Figure 3.2. Augmentation techniques were combined to create a more diverse dataset and make the model more robust to different conditions [112]. From the 8-band spectrum, Coastal (400 – 452 nm) to near-infrared (NIR 866 – 954 nm), a 4-band RGB+P (450 – 630 nm) band was applied. To expose the training to the desired invariance and ensure the model is robust, additional data augmentation was implemented: random brightness (30% of images in training set with random brightness), rotation (10%), perspective distortion (10%) and Gaussian noise addition (30%). Local contrast normalisation and pan-sharpening were applied. An overview of the satellite imagery augmentation techniques applied is provided below:

1. **Rotation:** Involves rotating the images by small angles to increase the diversity of the training data;
2. **Perspective distortion:** Involves applying non-rigid geometric transformations and perspective changes to the images;
3. **Brightness and contrast adjustment:** Involve adjusting the images' brightness and contrast to increase the training data's diversity;
4. **Gaussian noise:** Involves adding Gaussian noise to the images to increase the training data's diversity and make the model more robust to noise in real-world data;
5. **Weather and atmospheric conditions:** Involve adding different weather and atmospheric conditions to the images to increase the training data's diversity and make the model more robust to different lighting conditions.

3.3.4 Data pre-processing

This section corresponds to P9 stage in Figure 3.2. Due to practical GPU/TPU memory limitations, training a neural network using a pixel frame size equivalent to a full raw satellite image would cap the training batch size to a minimum and prevent the network from training effectively. Thus, satellite images with large Areas of Interest (AOI) are cropped into smaller AOIs called pixel frames (also known as pixel frame patches) of the size of (160x160 pixels) and consolidated into mosaics of the required shape and size for training. Smaller pixel frames allow for larger training batches and wider

context variability in each backpropagation cycle. However, a drawback of this approach is that on frame edges it collates mixed landscapes and cropped objects, consequently generating noise that distorts the contextual information in the training set as per Figure 3.6.

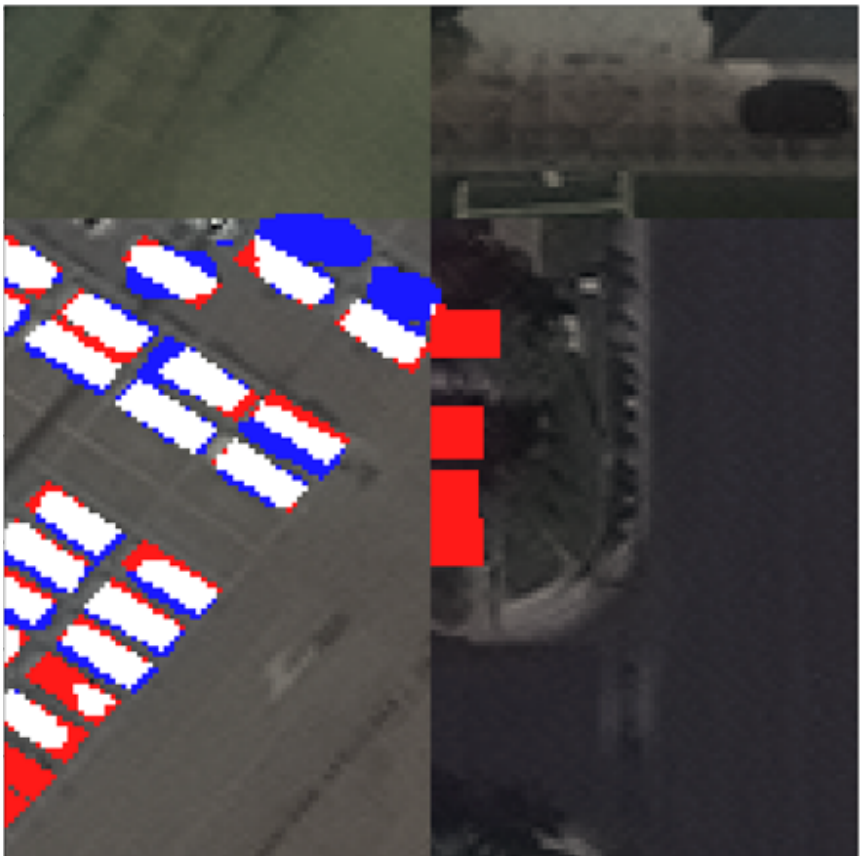


Figure 3.6. Four unrelated scenes are artificially combined in one frame. Used for training, it captures the partially-cut objects and scene shifts as ground truth. Doing so distorts the object-specific and contextual information and generates noise

Hence, to prevent the above drawback, we learned from the cropping techniques proposed approached by Chen et al. [42] and developed a programmatic conditional approach called “pixel frame selection” that feeds the neural network (Figure 3.2, component P9). It is an improvement from the method proposed by Chen et al. [42]. Via this approach, the network is trained on selected pixel frames (small cropped images) that allow intersections for better augmentation yet prevent duplications. We introduce the following

three principles as described in Figure 3.7 to minimize the contextual noise within a single frame and avoid duplicate frames:

1. Selection of 160x160 training frames randomly;
2. Rejection if a particular pixel frame falls across multiple large satellite images;
3. Rejection if pixel frames duplicate entirely.

This approach reduced the number of incorrect object polygons and the contextual noise in the training set, improving training accuracy and prediction precision.

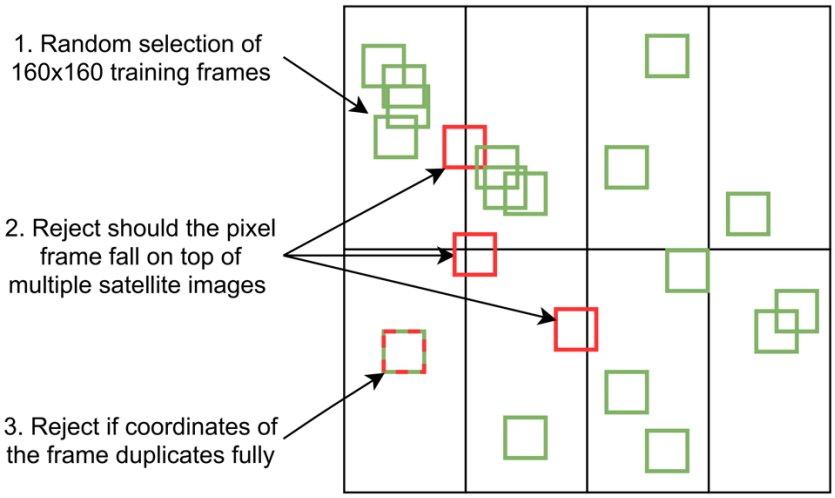


Figure 3.7. Pixel frame selection approach for network training. The green colour depicts valid pixel frames (patches), red colour represents rejected frames that were excluded from the training

In addition to the solution to training, we introduce a technique called "prediction frame sequencing" for improved prediction (Figure 3.2, component P10). It essentially allows for the neural network to broaden the contextualisation of the object it is classifying. Object classification is done given at least two different backgrounds (prediction frames). In the event of classification mismatch, the object is considered positively recognised (i.e., "OR" function), as illustrated in the four steps in Figure 3.8.

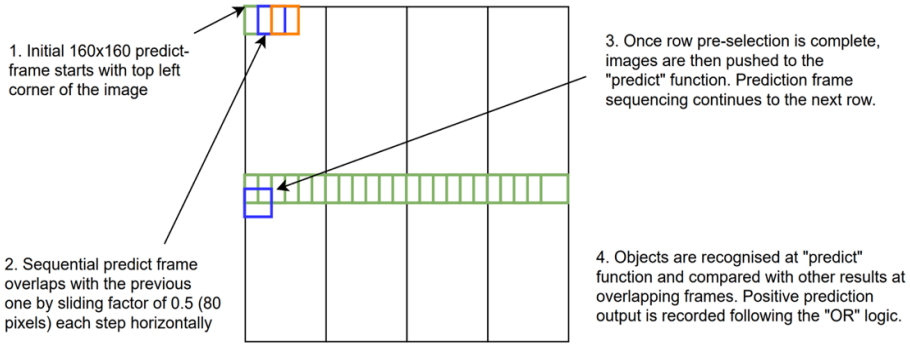


Figure 3.8 Prediction frame sequencing algorithm

To assess the impact of this technique, we trained and tested two identical neural networks on identical datasets. The first network utilised a standard prediction function (single random step, a non-overlapping prediction frame). The second network with implemented “prediction frame sequencing” approach outperformed the first network with a 3.57% higher object recognition accuracy rate.

3.4 Computational considerations

In addition to implementing pixel frame patches to improve contextual variability and taking practical GPU/TPU memory limitations, [55], [113] Ronneberger et al. [14] also suggest that to minimise the overhead and make maximum use of the GPU and TPU memory, we should favour a large input pixel frame over a large batch size and experiment with training batch sizes ranging from 32 to 192. In addition to this rule, the momentum optimisation algorithm – Adam [114], [115] was also implemented. Experiments were conducted on the custom-built Google Cloud Platform (GCP) architecture specifically developed for our research problem. To further experiment with latency reduction, two leading-edge computational machines, GPU NVIDIA Tesla P100 64GB (1 core) and TPU v3-8 128GB (8 cores), were deployed on our GCP system.

3.5 UNET and MACU

This section corresponds to components P5 and P6 in Figure 3.2. The present dissertation aims to address and find methods and models that perform best in accuracy and prediction speed metrics. Based on the literature review

and conclusion made in Chapter 2, we selected a UNET and MACU as the prospect to be the most accurate and/or the fastest network.

3.5.1 UNET architecture advancements

In this subsection, advancements are proposed to the process of UNET design, hyperparameters tuning, training, and complexity optimisation to enhance prediction accuracy and speed. We propose four distinctive architectures to derive an optimal network configuration for solving a prediction speed and accuracy problem. To originate these proposed configurations, we conducted quantitative experiments (see UNET Performance subsection 3.5.3) and visual examination (Figure 3.10) [14], [60].

UNET_Model_1

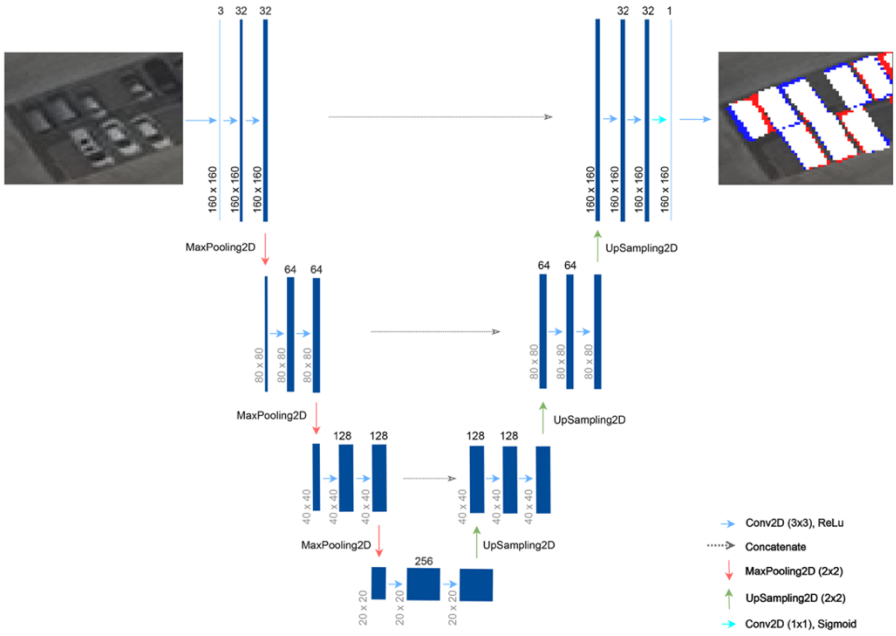


Figure 3.9. UNET_Model_1 design. Input image (left) and output image (right). The blue colour pixels represent the "light-vehicle" object class recognised by the UNET, red represents the original annotator-marked object contours, and white represents the accurate per-pixel prediction in the output image (right)

Each proposed UNET model consists of an even number of layers plus a single fully-connected layer with a Sigmoid activation function generating per-pixel semantic segmentation as an output (Figure 3.6). The breakdown of hyperparameters used across all four derived UNET model topologies is provided in Table 3.1 below.

Table 3.1 Hyperparameters of the UNET backbone implemented for all four UNET models

Hyperparameter	**Value**
Activation function (hidden layers)	ReLU
Activation function (output layer)	Sigmoid
Batch size	128
Epochs	20 - 100
Learning rate initiation	0.001
Optimizer	Adam
Drop-out	0.1
Hardware	NVIDIA Tesla P100 GPU
Memory	64 GB

A range of architectures with total layers from 11 to over 100 layers were considered. However, due to the computational demands involved, a comprehensive analysis encompassing all the available models was not feasible. This limitation was one of the inherent motivations and that prompted the investigation into alternative approaches, leading to the proposition of Neural Architecture Search (NAS) techniques discussed in the following chapters. The selected models within the target range were investigated starting with fifteen convolutions that sequentially (in four groups) were increased by six layers (three in the encoder and three in the decoder part) to a total of 39 layers where any models outside of this range of complexity were underperforming. Here are the configurations selected within the target range:

- UNET _Model_1*: 21 layers in total (15 conv2d)*
- UNET _Model_2: *27 layers in total (19 conv2d)*
- UNET _Model_3: *33 layers in total (23 conv2d)*
- UNET _Model_4: *39 layers in total (27 conv2d)*

During the encoder process, we capture semantic/contextual information, strengthening features extraction of "what" and reducing the "where". Each decoder convolutional block is part of upsampling and contains 2×2 convolutions (up-convolution) that halve the number of feature channels [52]. On the back of these upsampling operations, we recover the spatial information and enable precise localisation, i.e., the "where". A fully-connected layer leverages the corresponding concatenation and outputs the segmentation map of object classes. Rectified Linear Unit (ReLU) was selected as an initial activation function for non-linear mapping. The drop-out regularization technique was deployed with a 0.1 set to avoid overfitting [15] and provided a computationally cheap way to regularise the neural network [116] that increased the learning speed.

Given that there are no empirical methods to investigate how effectively a network performs feature extraction, we deploy the deconvolution-based Lucid visualisation technique [117]. We compare the feature maps from the last layer of the convolution operation of four UNET architectures (Figure 3.) [118]. In general, detectable objects in satellite imagery overall have specific contours (e.g., light-vehicle, truck, ship, or plane), which are consistent due to the perspective invariance of the camera. Specifications of our particular dataset are detailed in the dataset overview section and directory [111].

The Lucid visualisation method allows us to investigate how well the network performed high-level feature extraction tasks, i.e., recognising the object's contours, which is a crucial prediction accuracy driver. The Lucid feature visualisation approach reinforces providing an image that "most engages" a single feature at a time. The classifier is run on it to determine the direction (gradients) to alter the image for a more accurate classification (thereby minimising the loss between the prediction and the actual) starting with a random image. Each layer has different features, and we build an image for each one to maximise its responsiveness when averaged across all spatial positions [117].

U-net_Model_1

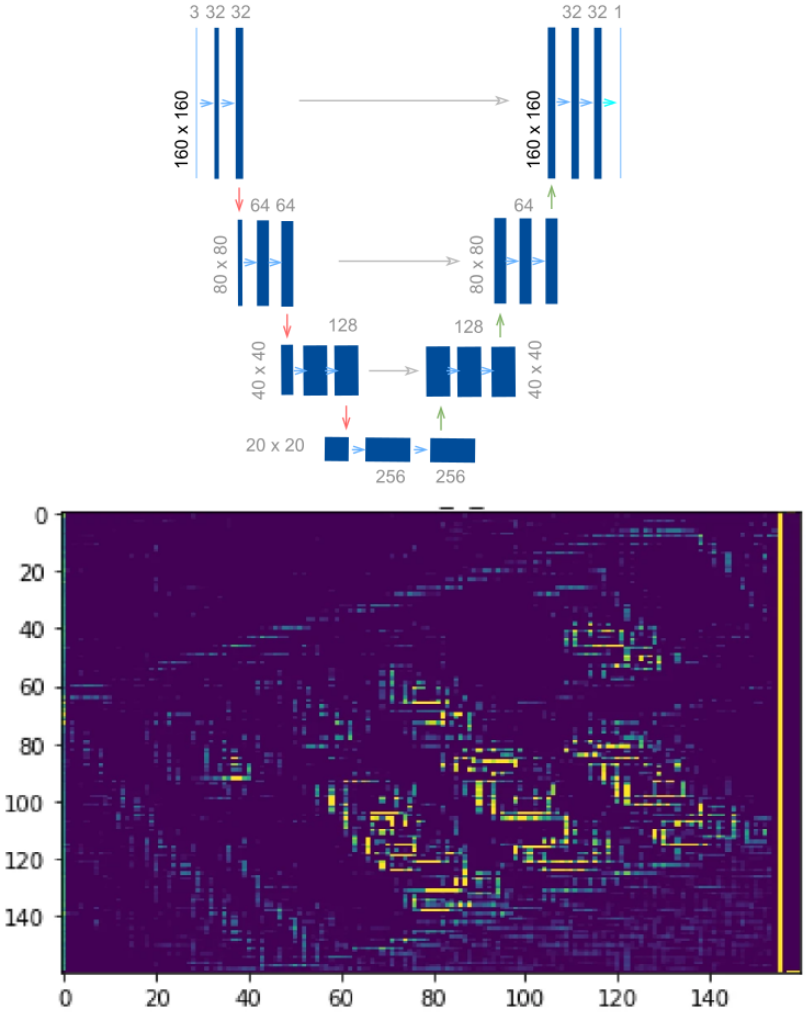


Figure 3.10 (a) Feature extraction capabilities with a different depth (UNET Model_1)

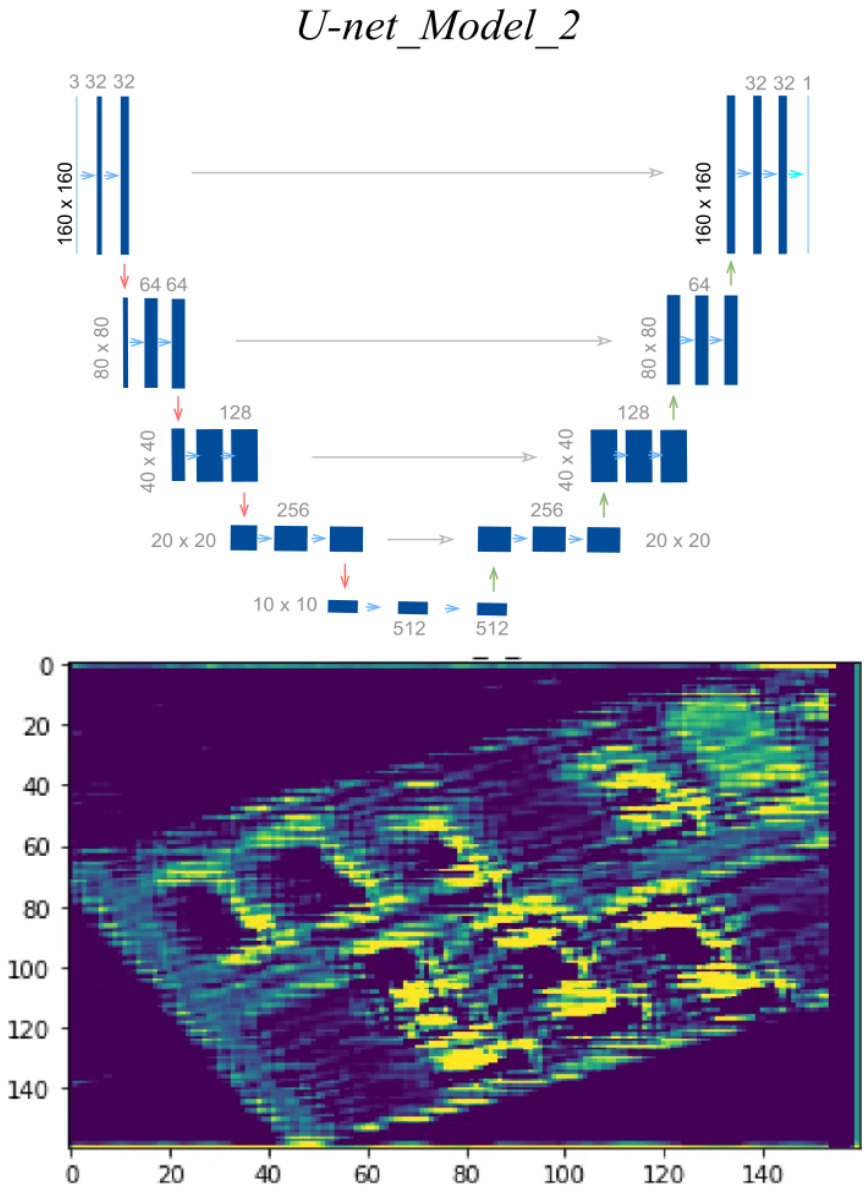


Figure 3.10 (b) Feature extraction capabilities with a different depth (UNET_Model_2)

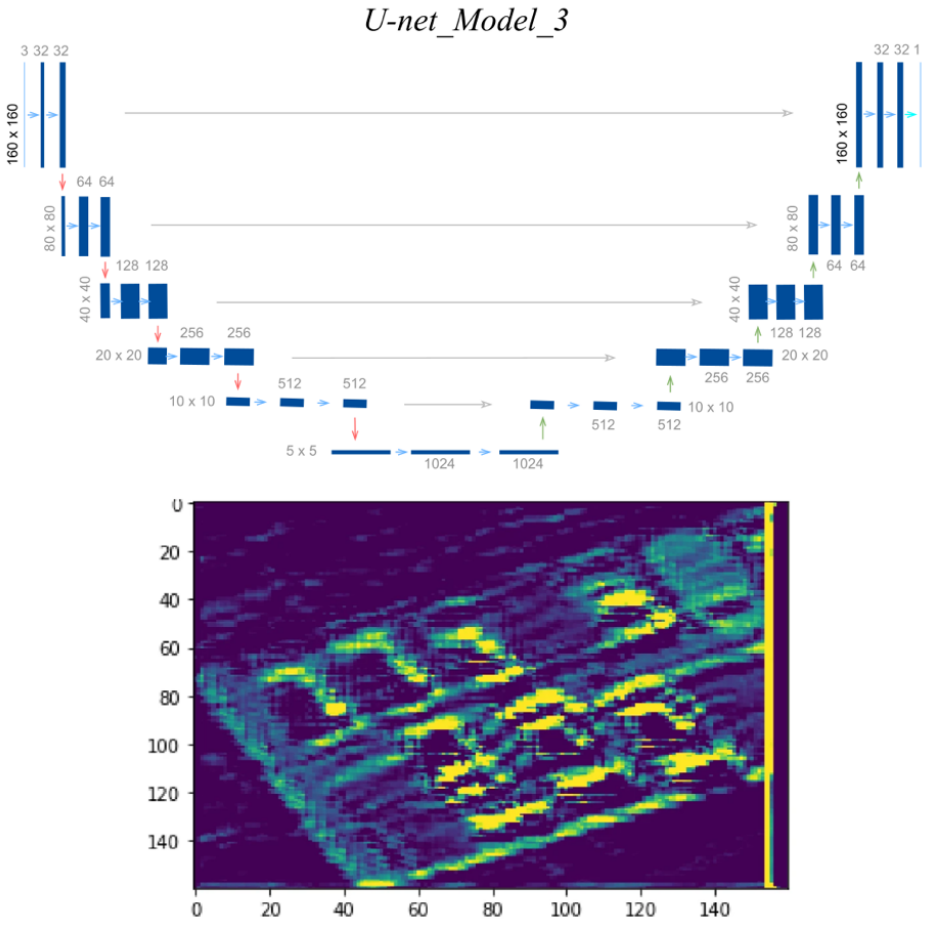


Figure 3.10 (c) Feature extraction capabilities with a different depth (UNET_Model_3)

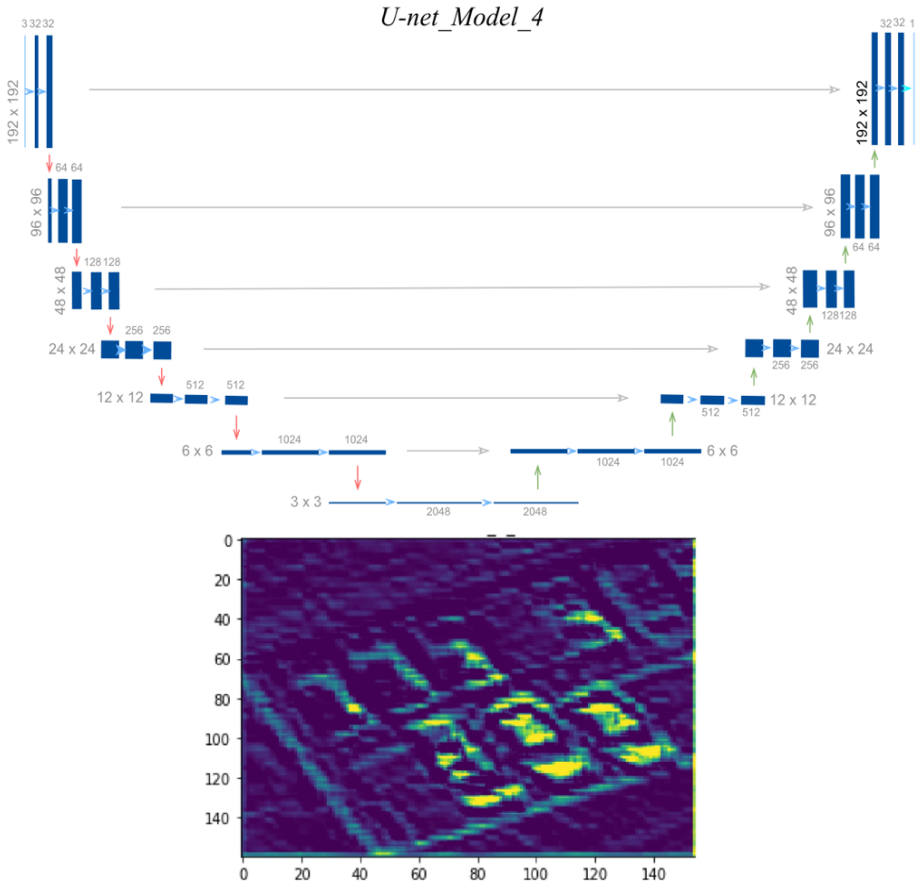


Figure 3.10 (d) Feature extraction capabilities with a different depth (UNET_Model_4)

Based on visual examination, we can suggest that the object's contours (i.e., high-level features) are more defined as the network depth increases. We can see a gradual improvement in feature extraction at each step with the equal depth differential (six convolutional layers) between architectures. The Lucid visual experimentation method can be utilised in other neural networks' performance testing since it allows us to compare performance “inside” the network.

3.5.2 Computational complexity

This section corresponds to component P7 in Figure 3.2. Significant signal latency from satellite imagery is caused by slow object recognition models (as illustrated in Figure 2.1) because complex models take time to "scan", detect and recognise objects in large land Areas of Interest (AOI) (e.g., 10km^2 at a time). Object recognition speed is a factor of computational complexity and power of computing [119]. For the design of efficient models, a detailed analysis of the number of floating-point operations (FLOPs) is required based on matrix operations such as matrix-matrix products (Figure 3.2, component P7). The product of two matrices $A^{m\times n}$ and $C^{n\times l}$ needs mnl FLOPs for multiplication operations and $ml(n-1)$ FLOPs for summation operations [120]. However, to our knowledge, no conventional benchmarks define the computational complexity of the neural network [121].

Research confirms that the number of operations in a network model can effectively estimate inference time [122]. In addition to reducing the prediction latency, model complexity is a critical consideration for other reasons such as:

- Overfitting: Less complex models have less risk of overfitting, which is when a model learns to perform very well on its training data but poorly on unseen data. Overfitting is a common problem in machine learning, especially with very complex models.
- Cost: Less complex models also cost less to run because they require less computational power. This can lead to substantial savings when models are run frequently or need to be trained on large amounts of data
- Efficiency: Less complex models require fewer operations, leading to faster predictions. This is especially important for applications that require real-time or near-real-time prediction.

The number of FLOPs represents how computationally expensive the model is [88]. We customise the FLOPs approach suggested by Sehgal et al. [88] to calculate the computational complexity of a neural network as defined in Equation (7):

$$G\text{-}FLOPs = \left[\underbrace{\sum_{e=1}^{E}\left(2 \times \left(\prod_{d=1}^{D_e} A_{ed}\right) \times F_e \times H_e \times W_e\right)}_{\text{Convolutional (}Conv2D\text{) layers}} + \underbrace{\sum_{b=1}^{B}\left(\prod_{x=1}^{X_b} P_{bx} \times \prod_{z=1}^{Z_b} O_{bz}\right)}_{\text{Max-pooling (}MaxPool\text{) layers}}\right] / 10^9 \quad (7)$$

Model complexity (G-FLOPs) is a sum of FLOPs for every layer, where E is the number of conv2D layers, D_e – number of output dimensions, A_{ed} – the size of the dimension of e layer, F_e – filter in depth parameter of e layer, H_e – filter height parameter of e layer, W_e – filter width parameter of e layer, B – number of Max-pooling layers, X_b – the number of filter dimensions of layer b, P_{bx} – the size of x dimension in layer b, Z_b – number of output dimensions of layer b, O_{bz} – the size of dimension z in layer b. The Conv2D layer count of floating-point operations depends on layer parameters count and layer output size [119]. The MaxPool layer count of floating-point operations depends on the size of the filter area and layer output size. Activation functions, including ReLU operations, can be executed by a single instruction. It was considered as one floating-point operation. Upsampling2D only reads the data from memory and writes to a certain position in the output using indices, and other pixels are filled by 0. The indices array always has the same shape as the input. Concatenation is just a memory copy; hence no floating-point operation will be conducted [123]. Index calculation might be needed depending on the concatenation axis, however, our approach ignores such operations. This calculation allows us to examine the relationship between the computational cost of the network, prediction accuracy, and prediction speed, all further examined in subsections 3.5.3 and 3.5.4.

3.5.3 UNET performance

In this subsection, we compare and contrast the performance of four proposed UNET architectures, starting with Table 3.2.

Table 3.2. Performance results on the test set

	Accuracy (TPO) %	**Overprediction (FPO) %**	**Jaccard index**	**G-FLOPs**
UNET_Model_1	95.33	12.01	0.6402	5.3218
UNET_Model_2	**97.67**	17.83	0.6162	6.9832
UNET_Model_3	97.01	26.45	0.5573	8.6443
UNET_Model_4	96.70	16.60	0.6226	10.3053

We achieved perfect object recognition accuracy (TPO) of 97.67% with UNET_Model_2. This network maintained an FPO level of 17.83% and a 0.6162 Jaccard Index. However, a close second best, UNET_Model_3, provided a significant overprediction (FPO = 26.45%) rate. G-FLOPS metric indicates the computational complexity, and UNET_Model_2 represents relatively light computational complexity with 6.9832 allowing an accelerated prediction as mentioned in the previous subsection 3.5.2. Figure 3.11 compares the accuracy performance between the models as well as their complexity.

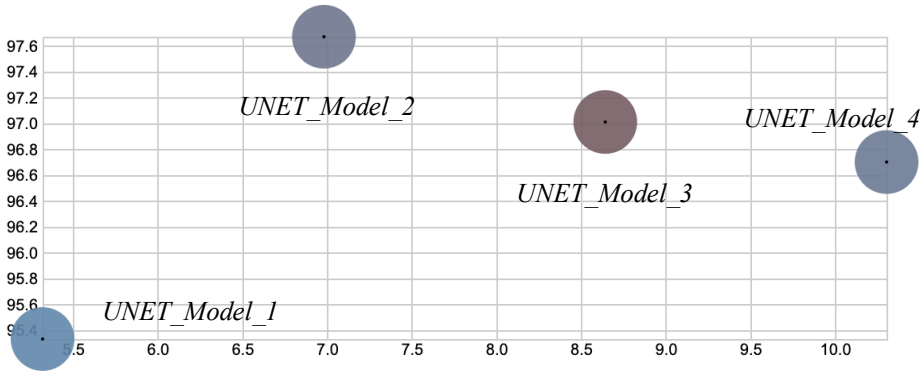


Figure 3.11. Comparison of performance results between UNET models (x-axis: computational complexity (G-Flops); y-axis - prediction accuracy (TPOs); dark red colour - highest, and blue - lowest overprediction (FPOs)

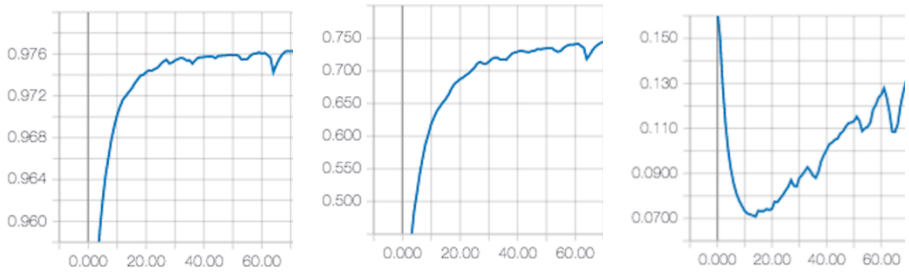
To continue enhancing the UNET_Model_2 performance, we experimented with the following activation functions within the hidden layers and not the output layer expecting an increase in accuracy and quality (see Table 3.3):

Table 3.3. Impact of activation function on prediction results on UNET_Model_2

Activation function	**Accuracy (TPO)**	**Overprediction (FPO)**	**TPO/FPO**	**Jaccard index**
ELU	96.74	18.12	5.34	0.6209
Tanh	90.81	6.09	14.92	0.5999
Softsign	86.62	**5.42**	**15.99**	0.5711
Softplus	93.76	23.17	4.05	0.5574
LeakyRelu	95.72	11.76	8.14	**0.6562**
PreLu	96.74	14.70	6.58	0.6364
ReLU	**97.67**	17.83	5.71	0.6162

Rectified Linear Unit activation (ReLU) provided the best accuracy (TPO) results for UNET [124]. However, the activation function that generates the lowest level of noise (FPO = 6.09%) is a Hyperbolic Tangent (Tanh) activation function that still provides > 90% accuracy and, simultaneously, a high TPO/FPO ratio (14.92).

Furthermore, to optimise the network training time, we monitored the process of the UNET_Model_2 training throughout a 20-100 epoch cycle. Training completeness was measured using three metrics, as illustrated in Figure 3.12 below. UNET reached the peak validation accuracy at epoch 35-40 and started to overfit. The validation loss curve (c) confirms the overfit by reaching a minimum at 15 and rapidly increasing beyond 35 as well as Jaccard Index plateaus beyond epoch 40. The variability of the optimal range did not change after experimenting with other UNET models. Understanding an optimal epoch range (35-40 epochs) minimises computational expense and re-training time. This range (35-40 epochs) just illustrates the most optimal point in the training cycle to avoid overtraining and save on computing costs. Therefore, it is particularly useful in applications where models such as algorithmic trading need to be recalibrated (retrained) frequently.



a) validation accuracy *b) validation Jaccard Index* *c) validation loss*

Figure 3.12. Training and Validation of UNET _Model_2
(x-axis: number of epochs)

We compared the performance of our proposed approach with the latest leading object recognition methods using external datasets to benchmark our proposed approach performance. Table 3.4 provides quantitative evaluations of the performance of our and other competing methods on two high-resolution remote sensing image data sets: OIRDS [125] and VEDAI [126]. Performance metrics of completeness (CPT) (also known as the Precision metric) and correctness (CRT) (also known as the Recall metric)

were adopted from the competing articles to ensure consistency and are calculated as $CPT = \frac{TP}{TP+FN}$, and $CRT = \frac{TP}{TP+FP}$.

Table 3.4. Quantitative evaluation of different leading methods

	Dataset	**Proposed method**	**Y.Yu's method [43]**	**H.Zhouze's method [12]**	**L.Wan's method [127]**
CPT	VEDAI	**0.90**	0.79	0.73	0.64
CPT	OIRDS	**0.89**	**0.89**	0.87	0.82
CRT	VEDAI	**0.57**	0.56	0.47	0.42
CRT	OIRDS	**0.78**	0.70	0.64	0.62
Both	*VEDAI and OIRDS*	***E = 40***	*E = 2000*	*K = 3000*	*K = 3000*

Our proposed architecture achieved the highest accuracy across all external datasets and methods in both CPT and CRT metrics. Furthermore, the number of epochs used to train the proposed UNET_Model_2 architecture was forty (40) epochs as compared to Y. Yu's [43] of 2 000 epochs resulting in a significantly lower computational cost. H. Zhou's and L. Wan's methods used K = 3 000 algorithm iterations. K iterations are the closest comparable metrics to E = Epochs. It can only be used as a rough comparable estimate of the computational resources used for the training stage of these fundamentally different methods.

3.5.4 Prediction speed

To compare its performance in practice, we conducted experiments utilising UNET_Model_2 for time-to-predict on two computational architectures, GPU and TPU. Identical models were used intentionally. These two computational environments and their characteristics are described in subsection 3.4. Consequently, GPU generated faster prediction speed results (see Table 3.5).

Table 3.5. TPU vs. GPU prediction speed for UNET_Model_2

Type	**Frame Size**	**Jaccard Index**	**Time-to-predict** (10k patches, in seconds)
TPU-v8	128 × 128	0.64	**20.45**
TPU-v8	160 × 160	0.64	36.42
TPU-v8	192 × 192	0.63	41.49
GPU-p100	128 × 128	0.64	**6.94**

Type	Frame Size	Jaccard Index	Time-to-predict (10k patches, in seconds)
GPU-p100	160 × 160	**0.65**	12.37
GPU-p100	192 × 192	**0.65**	20.45

One of the reasons why TPU might perform slower at the prediction task is that TPU-v8 is designed for larger complexity computations and longer operations compared to GPU-p100, with a much lower upfront computational load. As confirmed by Wang et al., "TPU speedup over GPU increases with larger CNNs" [128]. UNET_Model_2 architecture works better on GPU due to its light complexity (6.98 G-Flops). Therefore, we selected GPU as a preferred computational engine for UNET's rapid object recognition in real-time applications.

A total of eight UNET configurations with two different pixel frame parameters (128×128 and 160×160) were examined on a GPU machine to test the relationship between three metrics: 1) object recognition accuracy (%), 2) computational complexity (G-Flops), and 3) time it takes to predict a total of 10,000 patches of raw satellite imagery (in milliseconds).

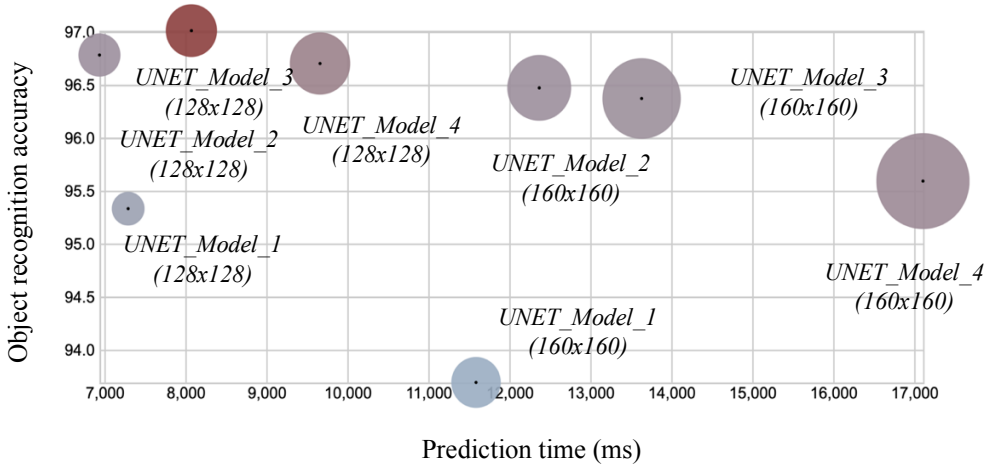


Figure 3.13. Object recognition accuracy vs. prediction speed vs. computational complexity. X-axis - prediction speed (in milliseconds), Y-axis - accuracy (TPO); the size of the circle - computational complexity in G-Flops; colour scale: the red colour indicates the highest overprediction error (FPO), blue - the lowest

Figure 3.13 depicts a direct relationship between the number of G-Flops in the computational architecture and prediction latency. Experiments were conducted with all four models and two-pixel frame sizes each. The higher the complexity, the longer it takes to predict when using an identical computational machine. Furthermore, a larger input frame size increases computational expense (G-Flops) in the network, slowing the prediction and not increasing accuracy in return. We can see that the fastest CNN network is UNET_Model_2 (128x128), which generated low overprediction (FP) and high accuracy (TP), which, as a result, is concluded as an optimal network for this real-time application on the GPU machine.

3.6 Multi-Scale Connected and Asymmetric-Convolution-Based network (MACU)

3.6.1 Prediction accuracy

We have conducted experiments with these four networks, MACU, FastFCN, DeepLabv3 and UNET (*formerly UNET_Model_2 which is now replaced with "UNET" for simplicity*) under different batch size environments to test their sensitivity to the quantity of the training data and also compare their relative performance on accuracy and overprediction. This experimentation would allow us to understand which network would potentially be more promising backbone for further investigative NAS research. We have adapted the networks to the Google Cloud Platform architecture used for the experimental investigation to be compatible with the satellite imagery dataset. We have recorded individual performance using the metrics described below.

Table 3.6 Comparable performance of four neural networks to assess the leading backbone in accuracy metrics

# Images and batch size	**Model**	**Jaccard Index**	**Recall**	**Precision**	***FPO* (%)**	F_1
30000 & 4	MACU	**0.661**	0.948	0.945	5.501	**0.946**
	FastFCN	0.615	**0.958**	0.926	7.383	0.942
	DeepLabv3	0.441	0.820	**0.968**	**3.156**	0.888
	UNET	0.652	0.955	0.923	7.691	0.939
30000 & 6	MACU	**0.667**	0.953	0.933	6.675	**0.943**
	FastFCN	0.506	0.828	**0.972**	**2.833**	0.894
	DeepLabv3	0.538	0.918	0.950	4.993	0.934
	UNET	0.658	**0.960**	0.919	8.099	0.939

During this experimental investigation, we have identified that the MACU network has the best overall performance defined by the F_1 score, which is the balance between Recall and Precision across three different information ratio/training intensity environments. In addition, MACU also performed best in all three environments on the pixel accuracy metric, the Jaccard index. UNET provides the best Recall, particularly useful in use cases where the objective is to recognise the maximum universe of objects within the given satellite imagery. The F_1 score is an improved representation of the network's overall performance, especially when assessing the practical application of the network to real-world problems. Precision allows understanding the targeted accuracy of correctly predicted objects.

Moreover, DeepLabv3 and FastFCN have shown a modest accuracy performance with the lowest number of objects. Yet, it is conservative and has the lowest overprediction error in two of three information intensity scenarios. A visual comparison between the results obtained by the four networks is depicted in Figure 3.14.

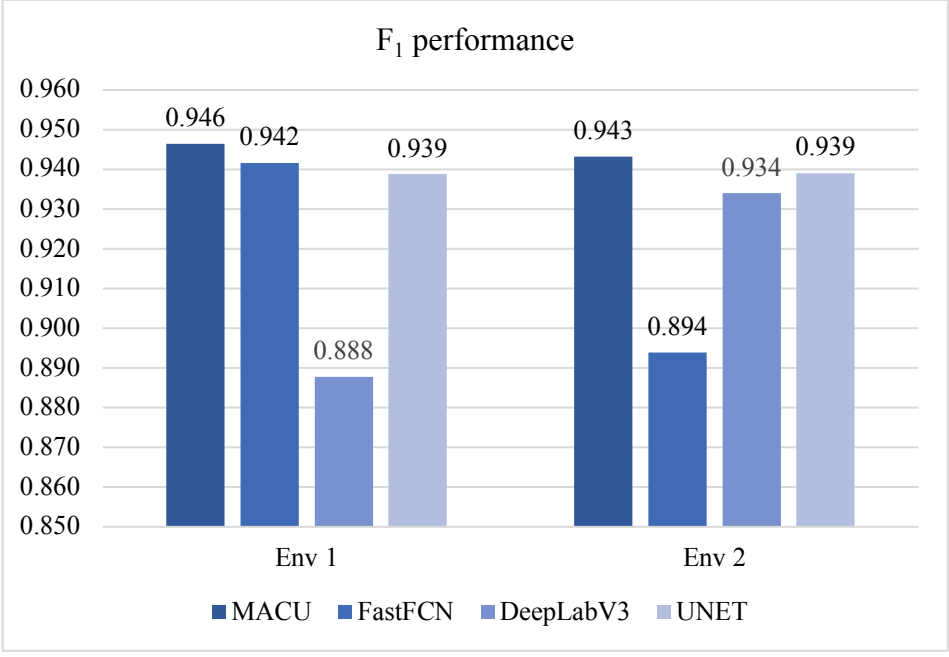


Figure 3.14 Performance (F_1) comparison in two information-intensity environments obtained by the four networks (MACU, FastFCN, DeepLabV3, UNET)

3.6.2 Prediction speed relative to accuracy

When comparing the two best-performing networks for accuracy (UNET and MACU), it is important to check what are the prediction speed performances between the two (see Table 3.7).

Table 3.7 Performance comparison for prediction speed

	Object recognition metrics (derived)		Prediction Speed (in seconds)	Quality to speed ratio
	Jaccard Index	**F_1**	**Time-to-Predict (10k patches)**	**Accuracy/Inference speed**
MACU	**0.667**	**0.943**	121.28	0.08
FastFCN	0.506	0.894	78.00	0.12
DeepLabV3	0.538	0.934	21.61	0.23
UNET	0.656	0.939	**17.53**	**0.54**

Even though MACU provides better performance as compared to UNET in accuracy for segmentation metric (Jaccard Index 0.667 vs 0.656) and F_1 (0.943 vs 0.939), due to its more computationally complex architecture, its prediction speed is 7 times slower (6.92x) as compared to UNET. Also, on a relative basis, UNET has also provided a better Accuracy/Inference speed ratio (0.52 vs 0.08). As a result, we can conclude that the proposed fully convolutional neural network modification based on UNET architecture provides the lowest network-specific prediction latency in satellite imagery as compared to other FCNs including MACU, DeepLab and FastFCN networks due to its computationally efficient architecture which is one of the defended statements. Therefore, in latency-sensitive applications, our manually designed UNET is the preferred choice of the model.

However, as compared to UNET, MACU would be a preferred model of choice when the accuracy of object recognition is of higher importance than inference speed. Also, MACU due to its better F1, Precision and Overprediction performance results evidently is more promising backbone network structure for further investigative research. This research, the purpose of further SOTA accuracy performance improvements will be exploring wider topology configurations, as part of NAS methodology. We investigate this domain in depth in chapter 4.

3.7 Chapter conclusions

1. Four UNET models were proposed and tested for best accuracy performance, computational complexity and prediction speed. The best-performing model was then compared to other state-of-the-art networks for metrics comparison, and the MACU network was selected as the most promising network architecture and the backbone for AutoML and NAS research.
2. Due to its computationally efficient architecture, the proposed fully convolutional neural network modification based on UNET architecture provides lower network-specific prediction latency for object recognition task in satellite imagery for “light-vehicle” object class as compared to other FCNs including MACU, DeepLab and FastFCN networks;
3. Manually-designed neural networks like UNET, MACU, and other manually designed networks require time to build, test and calibrate to certain problems. To address manual network limitations the next chapter focuses on AutoML and NAS techniques to propose novel solutions to improving the manually-designed network performance.

4 AUTOMATED NEURAL ARCHITECTURE SEARCH FOR OBJECT RECONGITION IN SATELLITE IMAGERY

In order to develop NAS for a certain type of network, we re-created and adapted the top-performing convolutional neural networks to date (MACU) (as discussed in section 2) and conducted thorough experimentation of its performance in object recognition via semantic segmentation task on the satellite imagery dataset. In Figure 4.1, we describe how NAS-MACU was constructed, and in the following sections, we explain the essential algorithm for an automated cell-topology design using the MACU backbone.

4.1 AutoML-based Neural Architecture

Automated Machine Learning offers techniques and procedures to make customized machine learning accessible. It allows us to increase its effectiveness and quicken the pace of Machine Learning research. Human-machine learning specialists normally must perform the following tasks to achieve excellent results in object recognition domain-specific tasks:

- Clean up and pre-process the data;
- Choose and construct the proper features;
- Choose the right model category;
- Enhance the model's hyperparameters;
- Create neural network topology;
- Model post-processing in machine learning;
- Critically evaluate the outcomes.

The construction of neural network architecture is automated using Neural Architecture Search. The architecture of the networks, including how to link nodes and which operators to use, is optimised using NAS techniques. Therefore, user-defined optimisation measures may include accuracy, model size, or inference time to determine the best architecture for a given application.

Conventional evolution of reinforcement-based AutoML algorithms tends to be computationally costly due to the vast search space. As a result, this study concentrated on investigating more effective approaches for NAS, especially for object recognition in satellite images. Recent advancements in

gradient-based and multi-fidelity approaches have offered a viable route and accelerated study in these areas [93].

4.2 Proposed NAS-MACU

NAS-MACU (Neural Architecture Search with Multi-level Attention and Cross-level Utilisation) is a neural architecture search (NAS) method that uses a multi-level attention mechanism and cross-level utilisation to improve the efficiency and effectiveness of the search process. The NAS-MACU method [129] consists of three main components:

- A search space: Set of possible architectures that the NAS algorithm will search through. The search space in NAS-MACU is defined by the types of layers (e.g., convolutional, pooling, etc.) and their connections.
- A search strategy: Method used by the NAS algorithm to research into the search space. In NAS-MACU, the search strategy combines reinforcement learning and evolutionary algorithms to guide the search process.
- Evaluation metrics: Used to determine the performance of the architectures found by the NAS algorithm. In NAS-MACU, the evaluation metrics is typically a measure of accuracy on a validation dataset.

The NAS-MACU algorithm starts by randomly generating a population of architectures within the search space. The algorithm then guides the search process and improves the performance of the architectures. Finally, the NAS-MACU algorithm returns the best-performing architecture found during the search process. This architecture can then be used as a starting point for further fine-tuning and training on the target dataset.

NAS can be applied across multiple use cases and have auto-calibration features that allow us to custom-cater for the problem to be considered. Then, we applied and further developed the NAS for auto-customised best-performing MACU network focused on optimising a search space at the cellular level. We produced an optimised and automatically-generated NAS-MACU neural network that was able to generate better accuracy performance. NAS-MACU is a new approach for object recognition in multispectral satellite imagery, which was never used before and opened multiple paths for future research. It is also a beneficial study for the remote sensing field due to the limitations of available training sets.

4.2.1 Proposed NAS-MACU development process

One of the most challenging components in solving real-world object recognition problems is to design a well-performing deep learning architecture catered to tackle remote sensing data-specific challenges such as dispersed scenery, variable satellite imagery resolution (e.g., 25cm per pixel – 5m per pixel); type of the sensors (e.g., Optical vs Synthetic Aperture Radar [SAR]), object class(-es), and other specificities of the training data. Our empirical research revealed that the MACU architecture demonstrates promising performance across various metrics when compared to other architectures such as UNET, FastFCN, and DeepLabv3. These evaluations were conducted on standard publicly available datasets, which were time-consuming to construct and have inherent limitations when applied to real-world scenarios.

Notably, even a manually calibrated network without an optimized cell-level architecture produced top-performing results. This finding suggests that the inherent network backbone architecture itself holds promise and warrants further investigation and experimentation. Therefore, we selected a MACU network as a backbone so that we could further improve performance by optimizing its cellular-level architecture.

No research or empirical study has been conducted to design and test for NAS-MACU to date that would overcome this problem. In this chapter, we design, implement and conduct empirical experimentation on the novel NAS-MACU which automatically adapts to the specificities of the remote sensing problem at hand. In order to deploy an effective NAS-MACU network, we improved a reiterative process illustrated in Figure 4.1.

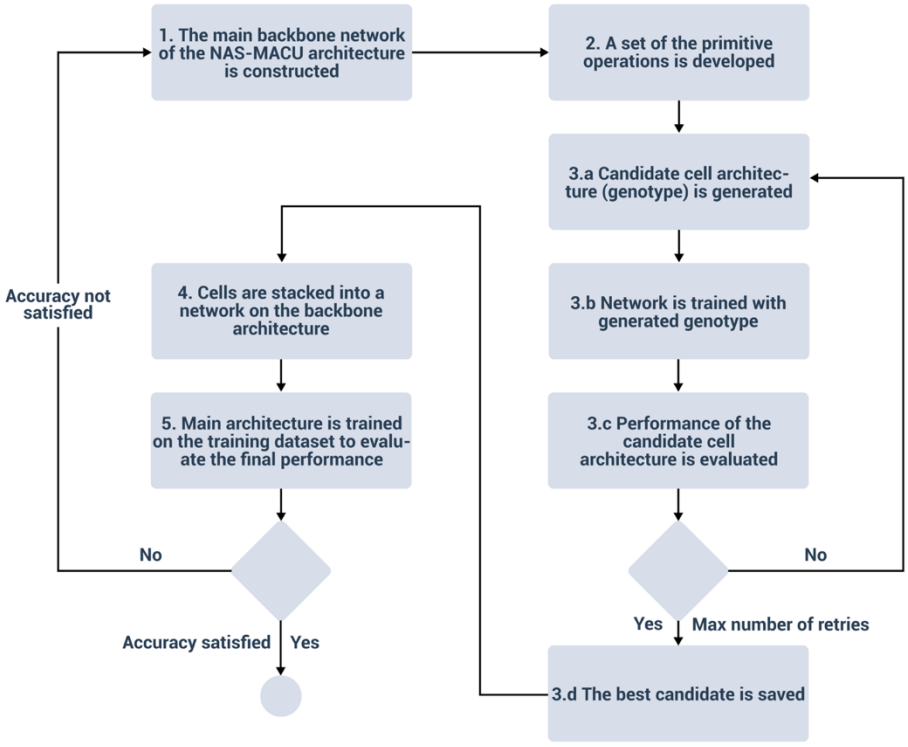


Figure 4.1 NAS-MACU construction process

Figure 4.1 depicts the process of delivering an excellent performance, self-designing-topology NAS-MACU network that adapts to a high dispersity of datasets without human expertise in the problem space or manual intervention. The NAS-MACU topology-design framework follows the iteration cycle until it reaches maximum performance given the constraints. Those constraints are expressed as operations (Step 2 in Figure 4.1) and are further discussed in Subsections 4.2.2 and 4.2.3. The research on NAS focuses on three aspects: search space (Step 3.a and 3.b in Figure 4.1), search strategy (Step 3.a and 3.b in Figure 4.1), and performance estimation strategy (Step 3.c in Figure 4.1).

The search space parameters shape through which architectures can be represented. The search strategy also describes how to investigate the search space. The objective is to find architectures with highly evaluated performance on unseen data. Performance estimation is divided into two parts. Firstly, the performance is evaluated to determine whether the candidate architecture will be kept (or expanded) for the next update. Secondly,

candidate cell architecture is added to a network stacked by the cells, and then the final performance is evaluated on a training dataset.

4.2.2 Cell-level topology search

A directed acyclic graph (DAG) in Figure 4.2 depicts the framework and basic structure for cell topology. Also, the diagram in Figure 4 illustrates the example of the cell architecture searched when the intermediate nodes in the DAG are three. We use three types of operations: down-up, normal, and concatenate operations.

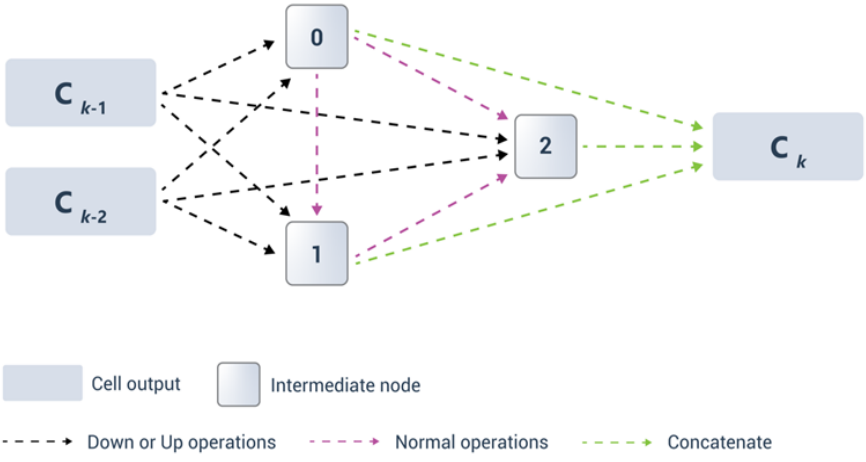


Figure 4.2. A directed acyclic graph diagram for cell architecture. The black arrow demonstrates a down operation; the magenta arrow depicts the normal operation (an operation that does not reduce the dimension of the feature map); the green arrow illustrates a concatenate operation

The input nodes C_{k-1} and C_{k-2} are defined as the cell outputs in the previous two layers. Every unit of intermediate nodes represents an input image or a feature map layer. An edge defines an operation between DAG nodes that the search space algorithm is tasked to find. The entire network shares the resulting framework of the cell. In this research, the DAG generation method was restricted to avoid huge search space and searched only for cell-based architecture. After determining the best cell architecture, the cells are stacked into a deeper network on the backbone MACU architecture. An example of the over-parametrised cell architecture is depicted in Figure 4.3.

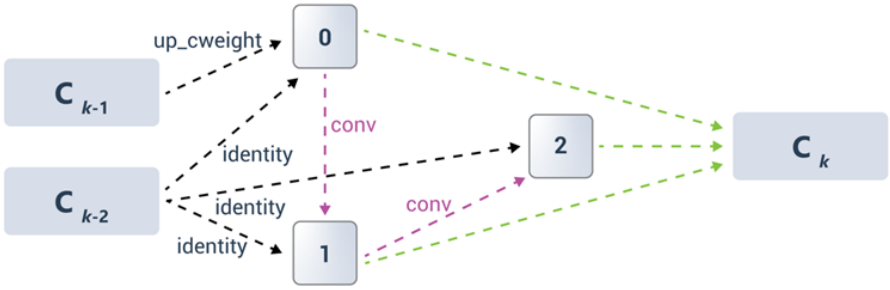


Figure 4.3 DAG diagram of the example of the cell architecture searched

In cell architecture search, we place each edge in DAG as a mixed operation, denoted as MixO. We use N candidate operations, denoted as $O = o_i$, which created N parallel paths. The output of a mixed operation MixO is defined based on weights w_i and operation o_i result in all paths (Eq. (7)):

$$\text{MixO}(x) = \sum_{i=1}^{N} w_i o_i(x) \quad (7)$$

4.2.3 Algorithm for the generation of cell genotype

NAS helps to automatically design two types of cell architectures called down-sampling cells (DownSC) and upsampling cells (UpSC) based on the MACU backbone (Figure 4.4. We improved the NAS-MACU cell genotype algorithm (see Algorithm 1 in Appendix A).

We describe the high-level logic of the underlying algorithm defining the cell topology design and iteration process, where E – total epochs and N – total nodes in a cell. The algorithm corresponds to steps from 3.a to 3.d in Figure 4.1 for the NAS-MACU construction process.

At the start of the algorithm, matrices of path weights (Weight1 and Weight2) are initiated with random values from a normal distribution with mean 0 and variance 1. Pleasae see the Appendix A for the logic tree of the algorithm. Due to the nature of the NAS process, the impact of initial random values is minimal. Weight1 is dedicated to up or down operation edges, and Weight2 stores values for normal operation edges (Figure 4.2). On every step i for N nodes, n paths are sampled, and all the other paths are masked (Mask1 and Mask2). Our method uses n=2, and the two paths are updated at each step. The small value of n reduces the time of searching and requires less computation. The edges1 array is created, which is a sorted array of row

indexes from masked weight matrices (denoted as $W1$) and sorted by row max weight values. Sorting uses the standard python TimSort algorithm.

The same selection process repeats for normal operations, and $gene\_items2$ is appended. When the cycle is finished, the best genotype is formatted from *gene_items*. The process repeats on every epoch while reaching the max number of total epochs (E) or genotype repeats, and constant MAX_PATIENCE is reached. This constant defines maximum iteration times when the best genotype in the last iteration is the same as in the previous iteration and set to 40.

Table 4.1. Primitive operations by type: down, up, and normal operations

Type	**Operations**
down_operations	'avg_pool', 'max_pool', 'down_cweight', 'down_dil_conv', 'down_dep_conv', 'down_conv'
up_operations	'up_cweight', 'up_dep_conv', 'up_conv', 'up_dil_conv'
normal_operations	'identity', 'none', 'cweight', 'dil_conv', 'dep_conv', 'shuffle_conv', 'conv'

4.2.4 MACU and NAS-MACU comparison

Based on UNET and asymmetric convolution block, multiscale features are generated by different layers of UNET. We use a multiscale skip-connected architecture MACU, for semantic segmentation, as illustrated in Figure 4.4. This standard MACU design has the following advantages: (1) the multiscale skip connections combine and realign semantic features that persist in high and low-level feature maps with different scales; (2) the asymmetric convolution block advances the representative capability of a standard convolution layer.

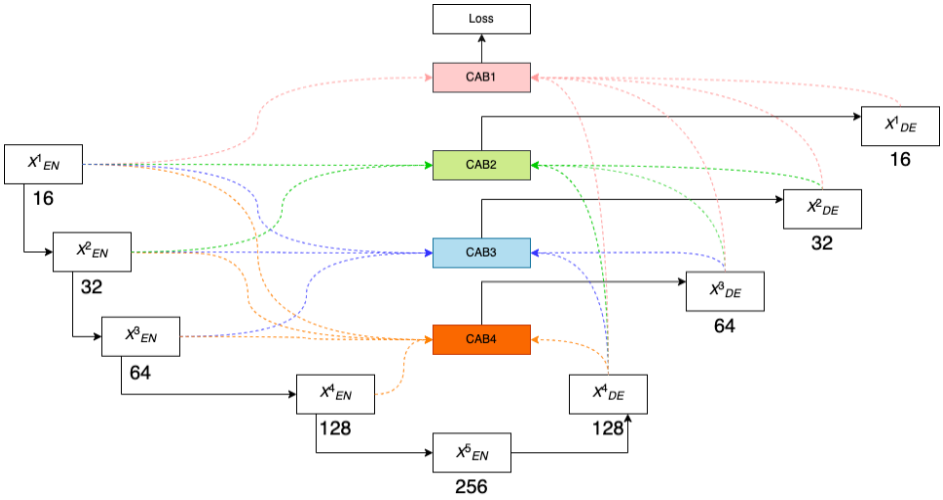


Figure 4.4. MACU network diagram with multiscale connectors (adapted from [20]). The CAB represents channel attention blocks

Channel Attention Blocks (CAB) are used to decrease the enormous number of channels coming from five feature maps of equal size and resolution and to realign channel-wise features. Coloured dotted arrows represent the multiscale skip connectors to each CAB.

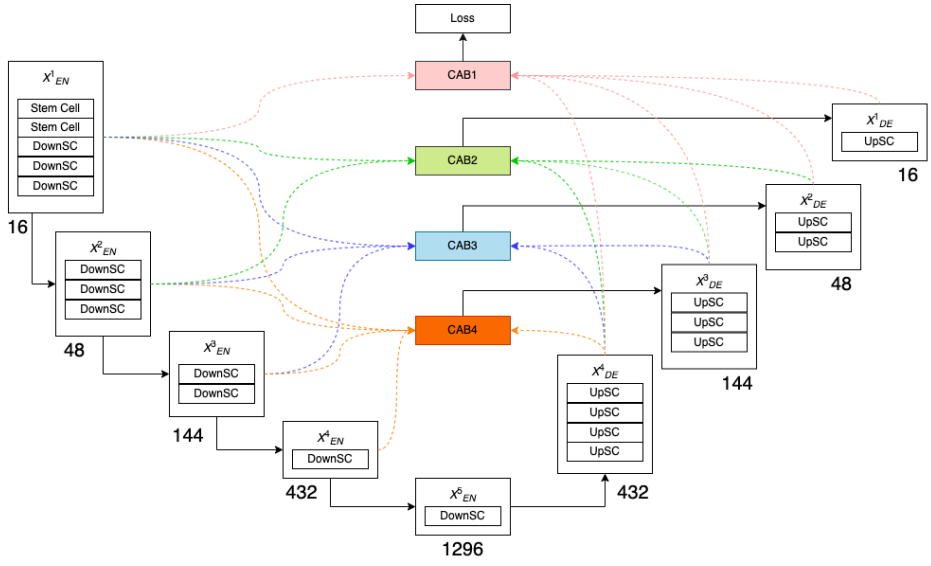


Figure 4.5. Proposed NAS-MACU architecture and cell topology at a high level. Downsampling cells and upsampling cells are stacked into the MACU network

NAS-MACU leverages the backbone of the MACU network described above. Within the cell structure, we implement the following DownSC (down-sampling cell) and UpSC (upsampling cell) cells, as illustrated in Figure 4.5. The difference between NAS-MACU and MACU is inside the cellular level of the network layers where the cell topology is designed automatically using NAS techniques.

4.2.5 NAS-MACU cell genotypes

As a result, we were able to conduct experimental cycles, and different genotypes of NAS-MACU were produced. The search and performance validation space of this experimentation was limited by the depth of search (up to 4 and 5 levels), the number of epochs in training (up to 500), batch size (up to 16), training duration, and several iterations to validate the best network (each winning cell structure was stress-tested up to 500 times that also can be expanded). These hyperparameters and limitations at each genotype are detailed in Table 4.2. In total, eight cycles were concluded, generating new NAS-MACU genotypes each time. The relative performance of each genotype is showcased in Table 4.2.

The best performance was delivered by NAS-MACU-V7 and NAS-MACU-V8. An additional parameter specific to NAS is a cellular level depth which represents the number of layers of operations within the cell.

Genotype Version	**Structure** (Down Operation, Parent Node Number)	**Structure** (Up Operations, Parent Node Number)	**Hyperparameters** (Epochs, Batch size, Cellular level depth, Training, Validation cycle)
NAS-MACU-V1	('down_cweight', 0), ('down_conv', 1), ('down_conv', 1), ('conv', 2), ('down_conv', 0), ('conv', 3)	('cweight', 0), ('up_cweight', 1), ('identity', 0), ('conv', 2), ('shuffle_conv', 2), ('conv', 3)	Epochs 300, batch 4, depth 4, training set 1000, validation set 100, max patience not reached
NAS-MACU-V2	('down_conv', 0), ('down_deep_conv', 1), ('down_conv', 1), ('conv', 2), ('shuffle_conv', 2), ('conv', 3)	('up_cweight', 1), ('identity', 0), (‘'up_conv’, 0), ('conv', 2), ('shuffle_conv', 2), ('conv', 3	Epochs 300, batch 4, depth 4, training set 1000, validation set 100, max patience not reached
NAS-MACU-V3	('down_dep_conv', 0), ('down_conv', 1), ('down_conv', 1), ('conv', 2), ('shuffle_conv', 2), ('conv', 3)	('up_cweight', 1), ('identity', 0), ('identity', 0), ('conv', 2), ('shuffle_conv', 2), ('conv', 3)	Epochs 500, batch 4, depth 4, training set 1000, validation set 100, max patience not reached
NAS-MACU-V4	('down_dil_conv', 0), ('down_conv', 1), ('down_conv', 1), ('conv', 2), ('conv', 3), ('shuffle_conv', 2)	('up_conv', 1), ('identity', 0), ('identity', 0), ('conv', 2), ('shuffle_conv', 2), ('identity', 0)	Epochs 500, batch 8, depth 4, training set 1000, validation set 100, max patience not reached

Genotype Version	**Structure** (Down Operation, Parent Node Number)	**Structure** (Up Operations, Parent Node Number)	**Hyperparameters** (Epochs, Batch size, Cellular level depth, Training, Validation cycle)
NAS-MACU-V5	('down_dep_conv', 0), ('down_conv', 1), ('down_dep_conv', 1), ('conv', 2), ('shuffle_conv', 2), ('conv', 3)	('identity', 0), ('up_conv', 1), ('identity', 0), ('conv', 2), ('shuffle_conv', 2), ('conv', 3)	Epochs 500, batch 32, depth 4, training set 1000, validation set 100, max patience not reached
NAS-MACU-V6	('down_dep_conv', 0), ('down_conv', 1), ('shuffle_conv', 2), ('down_conv', 1), ('cweight', 3), ('down_cweight', 1)	('conv', 0), ('up_conv', 1), ('identity', 0), ('shuffle_conv', 2), ('cweight', 3), ('identity', 0)	Epochs 500, batch 32, depth 4, training set 1000, validation set 100, max patience not reached
NAS-MACU-V7	('down_cweight', 0), ('down_conv', 1), ('down_conv', 1), ('conv', 2), ('down_conv', 0), ('conv', 3)	('up_cweight', 1), ('identity', 0), ('identity', 0), ('conv', 2), ('conv', 3), ('identity', 0)	Epochs 500, batch 16, depth 4, training set 2500, validation 500. Stopped after 204 max patience reached
NAS-MACU-V8	('down_cweight', 0), ('down_conv', 1), ('conv', 2), ('down_conv', 1), ('down_dep_conv', 0), ('max_pool', 1), ('max_pool', 1), ('identity', 3)	('conv', 0), ('up_conv', 1), ('up_conv', 1), ('conv', 2), ('identity', 3), ('conv', 2), ('cweight', 4), ('identity', 3)	Epochs 500, batch 16, depth 5, training set 2500, validation set 500

During experimental evaluations, first, we tested a total of eight different versions of NAS-MACU genotypes to select the best one. The detail of each genotype structure, their hyperparameters, and the performance of object recognition metrics is discussed further in this section. To derive the most optimal NAS-MACU architecture for applications, we conducted experiments with network configuration, complexity, and hyperparameters. Experiments were executed on the custom-built Google Cloud Platform architecture specifically developed for our research problem, and GPU NVIDIA Tesla P100 64 GB (1 core) was deployed on the system.

For the first NAS-MACU-V1, we used the down operations in the following order: (down_cweight, 0), (down_conv, 1), (down_conv, 1), (conv, 2), (down_conv, 0), (conv, 3) and ('conv', 3), where the first value in each operation represents the down operation and the second value represents the parent node number. The order of UP operations is as follows: (cweight, 0), (up_cweight, 1), (identity, 0), (conv, 2), (shuffle_conv, 2), (conv, 3). The depth of this model is set to 4 levels. The model is run for 300 Epochs, with a total of 1100 data set images, where 1000 are for training and 100 are for the validation set 100. It achieved a recall value of 0.949, a precision value of 0.880, and F_1 score of 0.913. The false positive rate for this version is 12.038%. This model performs slightly better than a simple, manually designed MACU model, yet it does not yield the best results.

The structure of the second model NAS-MACU-V2, is based on the following sequence Down operations: (down_conv, 0), ('down_deep_conv, 1), (down_conv, 1), (conv, 2), (shuffle_conv, 2), (conv, 3). The UP operations are in the following order:(up_cweight, 1), ('identity', 0), (up_conv, 0), ('conv', 2), ('shuffle_conv', 2), ('conv', 3). The depth of this model is also 4, and it is run for a total of 300 Epochs, with a similar dataset size as V1. The model, V2, achieved a recall value of 0.939, a precision value of 0.893, and an $F1$ score of 0.915. The false positive rate for this version is 10.704%. It gives better results when compared with NAS-MACU-V1 because of introducing the ‘deep-conv’ and ‘shuffle_conv’ down operations model still does not produce better results yet.

The third assembly of the NAS-MACU-V3 consists of (down_dep_conv, 0), (down_conv, 1), (down_conv, 1), (conv, 2), (shuffle_conv, 2), and (conv, 3) down operations and (up_cweight, 1), (identity', 0), (identity', 0), (conv', 2), (shuffle_conv, 2), and (conv, 3) UP operations. In this version, we train the model for a longer period with 500 Epochs while keeping the same batch size depth and training/validation set.

However, by increasing the depth and tweaking the structure a little bit, we observed a significant improvement compared to manually designed MACU. This V3 attained a recall value of 0.951, a precision value of 0.901, and a false positive rate of 9.865%. The $F1$ score obtained by this version is 0.926.

Thus, the NAS-MACU-V4 structure consists of (down_dep_conv, 0), (down_conv, 1), (down_conv, 1), (conv, 2), (shuffle_conv, 2), (conv, 3), and (shuffle_conv, 2) down operations and (up_conv, 1), (identity, 0), (identity, 0), (conv, 2), (shuffle_conv, 2), (identity, 0) UP operations.). Here, we introduced (down_dil_conv) at parent node 0. Also, in this model, we increased the batch size to 8 while keeping the same number of epochs and dataset (both training and validation) size. This version 4 attained a better false positive percentage of 9.552% with precision and recall values of 0.945 and 0.904. The F_1 score obtained by this version is 0.924.

The sequence of down operation in NAS-MACU-V5 structure is in the following order: (down_dep_conv, 0), (down_conv, 1), (down_dep_conv, 1), (conv, 2), (shuffle_conv, 2) and (conv, 3). The collection of UP operations is in order as (identity, 0), (up_conv, 1), (identity, 0), (conv, 2), (shuffle_conv, 2), and (conv, 3). In this model version, the batch size is further increased to 32 while keeping the rest of the hyperparameters the same as in the previous version. However, the mentioned combination of cellular-level operations and the increase in batch size does not lead to better FPO performance. Instead, the false positive rate increases to 17.626%, and the F_1 score decreases to 0.889. A similar case is with NAS-MACU-V6 where the down operation are: ('down_dep_conv', 0), ('down_conv', 1), ('shuffle_conv', 2), ('down_conv', 1), ('cweight', 3) and ('down_cweight', 1) followed by the UP operations in the sequence (conv, 0), (up_conv, 1), (up_conv, 1), (conv, 2), (identity, 3), (conv, 2), (cweight, 4), (identity, 3). With the same hyperparameters as the V5, this model also performs similarly to V5 by providing a high false positive percentage of 12.835 with an F_1 score of 0.916.

The best-performing versions are NAS-MACU-V7 and NAS-MACU-V8, where the batch size is reduced from 32 to 16. The training and validation set size is increased for these two models, with 2500 samples for training and 500 samples for validation. The cellular level down operations for V7 are (down_cweight, 0), (down_conv, 1), (down_conv, 1), (conv, 2), (down_conv, 0), (conv, 3). The UP operations for V7 are ('up_cweight', 1), ('identity', 0), (identity, 0), (conv', 2), (conv', 3), and ('identity', 0). Similarly, the cellular level down operations for V8 are ('down_cweight', 0), ('down_conv', 1), (conv, 2), (down_conv', 1), (down_dep_conv', 0), (max_pool, 1), (max_pool, 1), and (identity, 3). The UP operations for V8 are

('conv', 0), ('up_conv', 1), ('up_conv', 1), ('conv', 2), (identity, 3), (conv, 2), (cweight, 4), and (identity, 3). Both NAS-MACU-V7 and NAS-MACU-V8 obtained the best False positive percentages with values of 7.616% and 8.544%, respectively. In contrast, the F_1 score of NAS-MACU-V8 is slightly better than the F_1 score of NAS-MACU-V7, with values of 0.934 and 0931, respectively.

After obtaining the best versions, we finalise the NAS-MACU-V8 for further experimentation and comparison with the manually designed MACU model. Segmentation performance evaluation of NAS-MACU-V8 compared with MACU is done using four training set sizes across the major performance parameters. We observed that the NAS-MACU-V8 provides better results even if the training set size is reduced.

With constrained computational capabilities, this empirical study was carried out using the Google Cloud Platform. These capabilities might be expanded on a bigger scale even though they were not the minimum in computation time. We might enhance the cellular depths and broaden the search subspace of the cell architecture, given the increased computing resources available. We might also increase the constraining hyper-parameters like Total Epochs and "max patience" to expect further better performance. Better performing NAS-MACU designs may be discovered after expanding these computing resources, increasing searching capability and diversity.

To illustrate the cell-level topology generated through the cell search process described above, we created figures 4.6 – 4.7 to cover the NAS-MACU cell genotypes NAS-MACU-V7 and NAS-MACU-V8. These figures reflect a graphical representation of NAS-MACU cell genotypes. The cell can be considered a special block where layers are piled like any other model. These cells apply many convolution operations to get feature maps that can be passed over to other cells. C_{k-1} and C_{k-2} represent the output from previous cells. C_k is the output of the present cell. A complete model is made by stacking these cells in a series.

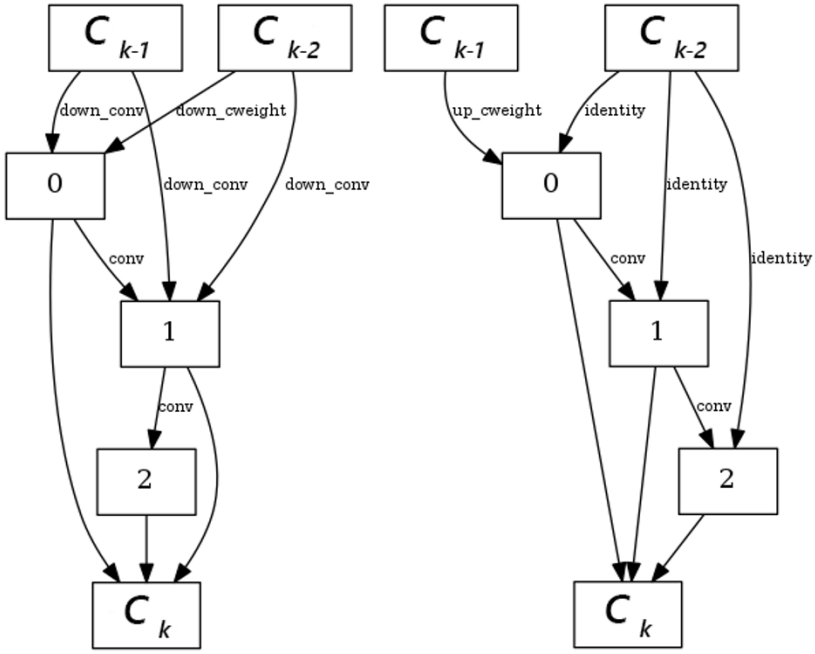


Figure 4.6 NAS-MACU-V7 (DownSC [left] and UpSC [right])

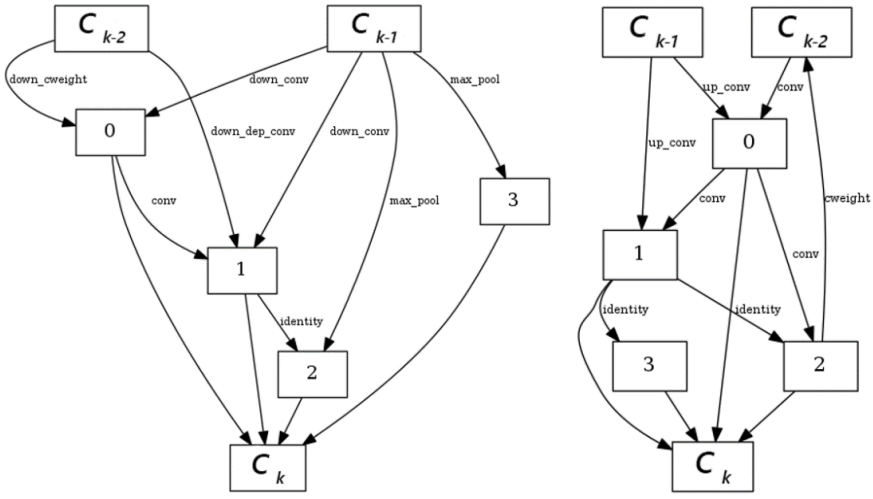


Figure 4.7 NAS-MACU-V8 (DownSC [left] and UpSC[right])

4.3 Experimental results and discussion

4.3.1 NAS-MACU Performance Evaluation

Eight genotypes were generated on the back of the different configurations to evaluate the performance of NAS-MACU on the full dataset, as described in Subsection 4.2.4. Results improved across the spectrum of metrics when comparing the NAS-MACU-V1 to NAS-MACU-V7 and NAS-MACU-V8 (Table 4.3.)

NAS-MACU-V7 and NAS-MACU-V8 showed similar performance. NAS-MACU-V8 achieved the best F_1 score. Also, it is worth mentioning that the NAS-MACU was able to uptrain itself extremely fast compared to manually designed networks with low information intensity for training. Hence, it was beneficial in settings where the training set is hard or expensive to acquire (e.g., high-resolution satellite imagery). Also, our experiments show that it takes only 15-20 epochs to reach top performance. Figure 4.8. illustrates this performance.

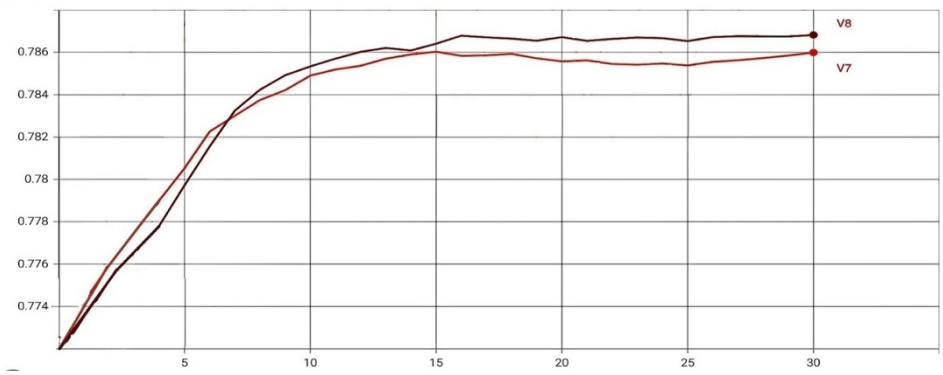


Figure 4.8. NAS-MACU-V7 and NAS-MACU-V8 comparison by validation accuracy. X- axis is the number of epochs; Y-axis Validation

NAS-MACU performance proved to surpass the manually designed MACU for this particular light-vehicle object recognition task and on this dataset. It performed especially well in the low-information intensity environment, as illustrated in Table 4.3.

Table 4.3. Performance comparison by object recognition metrics (Recall, Precision, FPO, F_1) across genotypes

	Object recognition Metrics (derived)			
Genotype Version	**Recall**	**Precision**	**FPO (%)**	***F1***
MACU	**0.969**	0.858	14.16	0.910
NAS-MACU-V1	0.949	0.880	12.038	0.913
NAS-MACU-V2	0.939	0.893	10.704	0.915
NAS-MACU-V3	0.951	0.901	9.865	0.926
NAS-MACU-V4	0.945	0.904	9.552	0.924
NAS-MACU-V5	**0.964**	0.824	17.626	0.889
NAS-MACU-V6	**0.965**	0.872	12.835	0.916
NAS-MACU-V7	**0.957**	**0.924**	**7.616**	**0.931**
NAS-MACU-V8	0.953	**0.920**	**8.544**	**0.934**

Furthermore, a few hours of AutoML work combined with GCP were employed to compute and generate a highly efficient NAS-MACU infrastructure. This stands in stark contrast to the lengthy duration of months typically required by researchers and practitioners to address tasks related to object recognition and semantic segmentation, as well as manually designing neural networks. Additionally, the utilization of these novel AutoML techniques enables the execution and fine-tuning of this process to achieve superior performance across a broad spectrum of problems, encompassing diverse object types, dataset specifications, and resolution constraints.

Table 4.4 illustrates the performance comparison of NAS-MACU-V8 and MACU across the main four performance metrics when trained using four different training set sizes. We can see that the performance of NAS-MACU-V8 compared to MACU increases as the training set size decreases, indicating the superiority of NAS-MACU-V8 over MACU in low-information environments. Information intensity environments can mostly vary by the Training set side (# of patched images) and other parameters like number of Epochs, and the number of images within a single batch per training epoch (Batch size). Information intensity is used to identify the quantity and completion of the training data used for the network's supervised learning. A training environment sufficient for a network to be sufficiently trained (i.e., low validation loss) is considered a high-information intensity environment. In contrast, a low-information intensity environment is defined as conditions where the network training data and the raining-related hyperparameters contain less than 30,000 images as validation loss increases.

Table 4.4. NAS-MACU-V8 vs. MACU in the variable information environments

		Object recognition metrics (derived)			
Training set size	**Network**	**Recall**	**Precision**	**FPO (%)**	F_1
5000	MACU	**0.968**	0.83	16.96	0.894
	NAS-MACU-V8	0.93	**0.893**	**10.67**	**0.911**
10000	MACU	**0.96**	0.87	13.03	0.913
	NAS-MACU-V8	0.938	**0.908**	**9.17**	**0.923**
20000	MACU	**0.969**	0.858	14.16	0.910
	NAS-MACU-V8	0.953	**0.915**	**8.54**	**0.934**
30000	MACU	**0.953**	**0.933**	**6.675**	**0.943**
	NAS-MACU-V8	0.941	0.917	8.321	0.929
40000	MACU	0.945	**0.942**	**5.788**	**0.943**
	NAS-MACU-V8	**0.958**	0.909	9.075	0.933

After the empirical investigation, we can confirm that NAS-MACU-V8 outperforms the MACU network, especially once the information intensity is reduced. The most important metrics to measure are F_1 (overall performance) and Precision. NAS-driven genotype outperformed human-made MACU network in overall accuracy performance (F_1) and Precision metrics in any information constraint environment and with an increasing difference as the training set size is reduced (Figures 4.9 and 4.10). Conducting NAS operation took from 4h to 58h of training and search time across NAS-MACU-V1 – NAS-MACU-V8 genotypes.

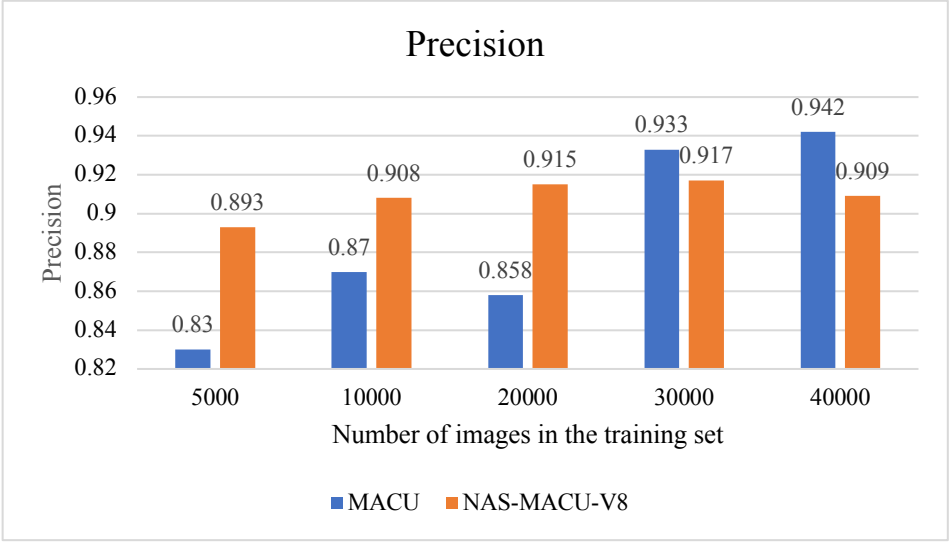


Figure 4.9. Precision of NAS-MACU-V8 vs MACU in five different training set sizes

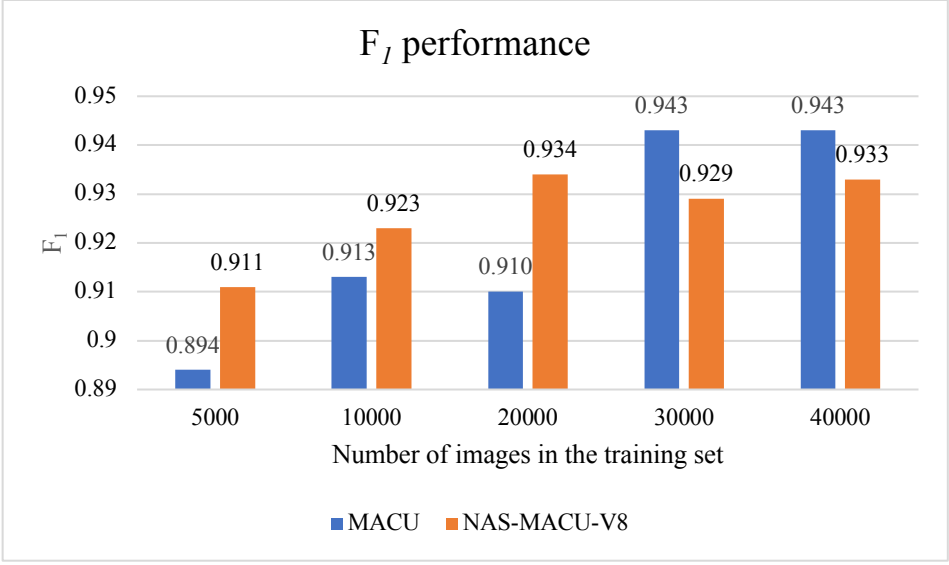


Figure 4.10. F_1 performance of NAS-MACU-V8 vs MACU in five different training set sizes

This was done automatically and without human intervention making this solution applicable at scale and a vast range of real-world applications. Figure 4.11 depicts a visual representation of performance on two example satellite images (A and B): Raw satellite imagery A (RGB, 30cm per pixel,

Shanghai) at the top and Raw satellite imagery B (RGB, 30cm per pixel, Paris) at the bottom. As you can see from these images, NAS-MACU-V8 outperformed MACU particularly well when applied in a low-light scene and when the object was similar to surroundings and darker (i.e., low-information environment).

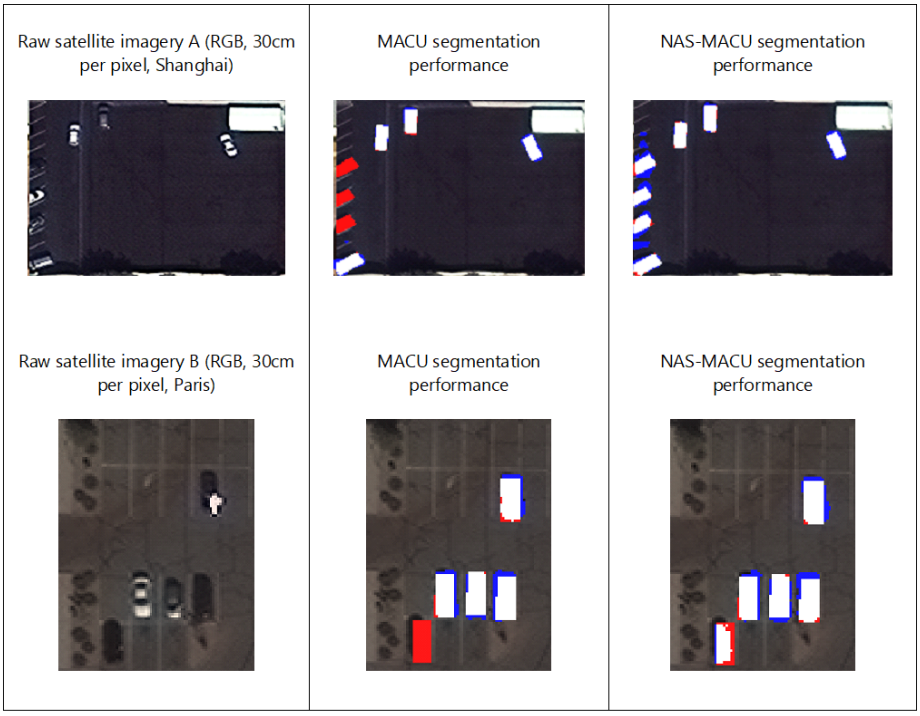


Figure 4.11. Precision performance of MACU vs NAS-MACU-V8 in a visual format. Blue colour pixels represent the "light-vehicle" object class recognised by the MACU or the NAS-MACU-V8, the red colour represents the polygon marked by the original annotator, white represents an accurate per-pixel prediction result

In order to understand the applicability, reproducibility and reliability of the performance results in the given dataset, we looked into the dispersity of the experimentation outcomes for key metrics that were used to indicate model and architecture accuracy performance (F_1 and Precision and Jaccard Index).

For the sample network architectures, the average standard deviation (SD) of F_1 was 0.007, and the SD of the Precision metric was 0.019, whereas Jaccard Index had an SD of 0.026. The standard deviation represents the amount of variation or dispersion in the dataset. In this case, it indicates that

the performance of the sample architectures even after conducting multiple experiments have relatively small standard deviations and are close to the mean. Deviations of 0.007 - 0.026 indicate the reliability of the results on the given dataset and results' reproducibility.

The actual satellite imagery dataset was also heterogeneous, dissimilar, or varied in nature. It contained 4 cities (Paris, Las Vegas, Khartum, Shanghai) and a variety of environments (parking on the street, parking with trees, city centre and urban areas) and atmospheric conditions (dispersed haze, light, perspective distortion datasets) suggesting that results and methods are likely to be applicable and reliable once applied to other satellite imagery datasets too.

4.4 Chapter conclusions

Chapter 4 proposes a novel approach to automated neural architecture search (NAS) for object recognition in light-vehicle class in satellite imagery using a CNN cell-level topology search in the MACU backbone. The NAS-MACU network outperforms other popular manually designed networks for object recognition in satellite imagery. The NAS procedure allows for obtaining a new, well-performing network configuration without human intervention.

The lack of publicly available satellite imagery data is a limitation for effectively researching and applying deep learning models to real-world problems. The constructed NAS-MACU performed exceptionally well in a low-information environment compared to other popular manually designed networks. Several NAS-MACU configurations were obtained that outperformed the MACU network.

In all low-information cases analysed (training set size up to 20,000), the NAS-MACU-V8 network achieved better object recognition performance compared to the MACU network on precision, FPO, and F_1 metrics. NAS-MACU-V8 achieved the best performance according to the F_1 metrics (0.934) when the training set size was 20 000, also having better precision (0.915) and FPO (8.54) than manually-designed MACU. An effective NAS implementation in the MACU network can self-discover the well-performing cell topology and architecture optimized for object recognition in multi-spectral satellite imagery.

5 GENERAL CONCLUSIONS

Conclusion 1: The Sat-modification framework improved the accuracy and speed of object recognition in satellite imagery

Through the incorporation of novel approaches in the Sat-modification framework to enhance neural network capabilities, including feature extraction, network complexity measurement, training process fine-tuning, and prediction speed optimization, the framework yielded substantial improvements in accuracy and efficiency. Notably, the UNET architecture within the framework achieved an accuracy of 97.67% for the "light-vehicle" object class. Additionally, the computationally light UNET_Model_2 architecture demonstrated a remarkable fivefold improvement in training time, enabling real-time applications. These findings demonstrate the efficacy of the Sat-modification framework in enhancing object recognition capabilities in the challenging domain of satellite imagery.

Conclusion 2: UNET is to be selected as the preferred FCN low inference latency use case due to its light computational architecture

The computational complexity of object recognition models has a significant impact on prediction latency and overall performance. The number of floating-point operations (FLOPs) is an effective measure of computational complexity and can be used to estimate inference time. Less complex models not only reduce prediction latency but also mitigate issues such as overfitting, lower costs, and improve efficiency. The UNET_Model_2 demonstrated the best performance in terms of prediction speed, yet with a high overprediction, and relatively light complexity of 6.9832 G-FLOPs. The choice of activation function also plays a role in performance, with ReLU providing the best accuracy and Tanh offering the lowest noise level. Additionally, the optimal epoch range of 35-40 epochs was identified to minimize overfitting and computational expenses for training. The prediction speed experiments demonstrated that the GPU outperformed the TPU for the UNET_Model_2. In direct comparison with MACU, UNET provided a nine times faster prediction than MACU (14.22 seconds vs 112.46 seconds) and an accuracy differential only of 2% (F_1 of 0.935 vs 0.949). These findings allow us to conclude that the UNET_Model_2 process using the Sat-modification framework is to be selected as a preferred network architecture and technique for use cases that are sensitive to inference latency such as algorithmic trading.

Conclusion 3: MACU outperformed other manually designed networks for overall accuracy metrics and was selected as the backbone for NAS

In this research, we conducted experiments to compare the performance of four neural networks, namely MACU, FastFCN, UNET, and DeepLabv3, under different information-intensity environments. The results obtained in terms of segmentation and object recognition metrics were analysed and compared. DeepLabv3 and FastFCN exhibit moderate accuracy with a lower number of objects but demonstrate conservative behaviour with lower overprediction errors. Our findings indicate that the MACU network demonstrates the best overall performance, as measured by the F_1 score, across all three information intensity environments. Based on these findings, the MACU network is selected as the most promising architecture for further research in AutoML and NAS.

Conclusion 4: 1. The proposed novel NAS-MACU provides more accurate object recognition for light-vehicle object class in a low-information environment compared to the manually expert designed MACU network.

The development of the NAS-MACU network, incorporating automated Neural Architecture Search (NAS) techniques, represents a notable contribution to the field of object recognition in satellite imagery. Through NAS, multiple configurations of the NAS-MACU network were obtained, surpassing the performance of the manually designed MACU network, particularly in low-information environments. Notably, NAS-MACU-V8 achieved the best F_1 score of 0.934, demonstrating the effectiveness of NAS in optimizing network architectures for light-vehicle object class recognition in multi-spectral satellite imagery. By automating the discovery of well-performing cell topologies, the NAS implementation in the MACU network eliminates the need for manual intervention and streamlines the architecture optimization process. These findings highlight the potential of NAS techniques to significantly enhance the performance and efficiency of neural networks in the domain of satellite imagery object recognition.

The present dissertation successfully achieved its objectives, providing a comprehensive investigation of the proposed Sat-modification framework, conducting a rigorous comparative analysis of neural network architectures, building a model to assess and evaluate computational complexity and prediction speed and designing an innovative AutoML-based

NAS-MACU network. These contributions advanced the field of AutoML for object recognition in satellite imagery, offering improved accuracy, prediction speed, and automated architectures catered for the unique and dispersed applications of object recognition in optical satellite imagery.

5.1 Future work

The effectiveness of deep learning models in addressing real-world problems is hindered by the scarcity of publicly available satellite imagery data. The experimental investigation in this study was conducted on the Google Cloud Platform, utilizing limited computational resources. Although these experiments required a significant number of computational hours, future endeavours could benefit from increased computational resources. This expansion would facilitate research of a broader search space for cell infrastructure and an increase in cell depth. Moreover, it would allow for the relaxation of constraints imposed by limiting hyperparameters such as "max_patience" and "Total Epochs". In addition to that, the other three methods of improving inference latency discussed in the Introduction chapter (Model Compression, Hardware acceleration and Software optimisation) can be further researched to enhance the low-latency performance.

Considering the limitations associated with optical multispectral satellite imagery, particularly concerning atmospheric and sunlight conditions, it would be advantageous to explore research avenues involving SAR satellite imagery. The imagery enables us to capture data through clouds, during night time, and in the presence of haze. This exploration could potentially lead to the adoption of improved NAS-MACU architectures.

Furthermore, there is potential for further investigation into alternative neural network backbone architectures using NAS. The current research could extend beyond object recognition in satellite imagery and delve into other domains such as medical image analysis (e.g., tumour detection), aerial image processing (e.g., semantic segmentation in UAV imagery), forensics (e.g., handwriting detection), autonomous machinery (e.g., machinery navigation in specific environments), and other relevant fields.

6 REFERENCES

[1] Q. Zhao, Y. Le, Z. Du, D. Peng, P. Hao, Y. Zhang and P. Gong, "An overview of the applications of Earth observation satellite data: impacts and future trends," *Remote Sensing,* vol. 14, no. 8, p. 1863, 2022.

[2] A. S. Belward and J. O. Skøien, "Who launched what, when and why; trends in global land-cover observation capacity from civilian earth observation satellites," *Journal of Photogrammetry and Remote Sensing,* vol. 103, pp. 115-128, 2015.

[3] A. Monk, M. Prins and D. Rook, "Rethinking Alternative Data in Institutional Investment," *The Journal of Financial Data Science,* vol. 1, no. 1, pp. 14-31, 2019.

[4] A. Letizia, S. Marino, U. Ahmad and A. Alvino, "Investigating the global socio-economic benefits of satellite industry and remote sensing applications," *IBIMA Publishing,* vol. 10, no. 3, pp. 475-480, 2019.

[5] T. W. Gillespie, J. Chu, E. Frankenberg and D. Thomas, "Assessment and prediction of natural hazards from satellite imagery," *Progress in Physical Geography,* vol. 31, no. 5, pp. 459-470, 2007.

[6] E. Guirado, S. Tabik, M. L. Rivas, D. Alcaraz-Segura and F. Herrera, "Whale counting in satellite and aerial images with deep learning," *Nature Scientific Reports,* vol. 9, no. 1, pp. 1-12, 2019.

[7] G. Di Baldassarre, G. Schumann and P. Bates, "Near real time satellite imagery to support and verify timely flood modelling," *Hydrological Processes: An International Journal,* vol. 23, no. 5, pp. 799-803, 2009.

[8] S. Voigt, F. Tonolo, J. Lyons, J. Kučera, B. Jones, T. Schneiderhan, Platzeck and Gabriel, "Global trends in satellite-based emergency mapping," *Science,* vol. 353, no. 6296, pp. 247-252, 2016.

[9] D. Franco, C. Kourogiorgas, M. Marchese, A. Panagopoulos and F. Patrone, "Small satellite and CubeSats: Survey of structures, architectures, and protocols," *International Journal of Satellite Communications and Networking,* vol. 4, no. 37, pp. 343-359, 2019.

[10] Z. N. Musa, I. Popescu and M. A, "A review of applications of satellite SAR, optical, altimetry and DEM data for surface water modelling, mapping and parameter estimation," *Hydrology and Earth System Sciences,* vol. 9, no. 19, pp. 3755-3769, 2015.

[11] R. Roberts, J. Goforth, G. Weinert, C. Grant, W. Ray, B. Stinson and A. Duncan, "Lawrence Livermore National Lab.(LLNL)," in *Autmated Annotation of Satellite Imagery using Model-based Projects*, Livermore, CA (United States), 2018.

[12] H. Zhou, L. Wei, C. P. Lim and S. Nahavandi, "Robust vehicle detection in aerial images using bag-of-words and orientation aware scanning," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 12, no. 56, pp. 7074-7085, 2018.

[13] R. A. Marcum, C. H. Davis, G. J. Scott and T. W. Nivin, "Rapid broad area search and detection of Chinese surface-to-air missile sites using deep convolutional neural networks," *Journal of Applied Remote Sensing,* vol. 11, no. 4, 2017.

[14] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[15] H. Li, J. Chen, H. Lu and Z. Chi, "CNN for saliency detection with low-level feature integration," *Neurocomputing,* vol. 226, pp. 212-220, 2017.

[16] S. Ghosh, N. Das, I. Das and U. Maulik, "Understanding deep learning techniques for image segmentation," *ACM Computing Surveys (CSUR),* vol. 4, no. 52, pp. 1-35, 2019.

[17] M. Längkvist, A. Kiselev, M. Alirezaie and A. Loutf, "Classification and segmentation of satellite orthoimagery using convolutional neural networks," *Remote Sensing,* vol. 4, no. 8, p. 329, 2016.

[18] L. Yingyan, C. Sakr, Y. Kim and N. Shanbhag, "PredictiveNet: An energy-efficient convolutional neural network via zero prediction," 2017.

[19] D. Cliff, D. Brown and P. Treleaven, "Technology Trends in the Financial Markets: A 2020 Vision: the Future of Computer Trading in

Financial Markets-Foresight Driver Review-DR 3," *Government Office for Science,* 2010.

[20] L. Rui, C. Duan and S. Zheng, "MACU-Net Semantic Segmentation from High-Resolution Remote Sensing Images," *arXiv preprint arXiv:2007.13083,* 2020.

[21] K. Ose, T. Corpetti and L. Demagistri, "Multispectral satellite image processing," *Optical Remote Sensing of Land Surface,* pp. 57-124, 2016.

[22] X. Li, Y. Yuan and Q. Wang, "Hyperspectral and Multispectral Image Fusion via Nonlocal Low-Rank Tensor Approximation and Sparse Representation," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 59, no. 1, pp. 550-562, 2021.

[23] A. Khoreva, R. Benenson, J. Hosang, M. Hein and B. Schiele, "Simple does it: Weakly supervised instance and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[24] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan and P. Z. L. Dollár, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European ConferenceProceedings, Part V 13, pp. 740-755. Springer International Publishing*, Zurich, Switzerland, September 6-12, 2014.

[25] S. Gupta, R. Girshick, P. Arbeláez and J. Malik, "Learning Rich Features from RGB-D Images for Object Detection and Segmentation.," *Computer Vision (ECCV),* vol. 8695, 2014.

[26] A. Angelova and Z. Shenghuo, "Efficient object detection and segmentation for fine-grained recognition," 2013.

[27] M. Cheriet, J. Said and C. Suen, "A recursive thresholding technique for image segmentation," *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society,* vol. 7, pp. 918-21, 1998.

[28] M. Luo, Y. Ma and H. Zhang, "A spatial constrained K-means approach to image segmentation.," vol. 2, pp. 738 - 742, 2004.

[29] D. Krstinic, A. Kuzmanic Skelin and I. Slapnicar, “Fast two-step histogram-based image segmentation,” *Image Processing,* vol. 5, pp. 63-72, 2011.

[30] J. Long, E. Shelhamer and T. Darrell, “Fully convolutional networks for semantic segmentation,” *Proceedings of the IEEE conference on computer vision and pattern recognition,* p. 3431–3440, 2015.

[31] H. Noh, S. Hong and B. Han, “Learning deconvolution network for semantic segmentation,” *Proceedings of the IEEE international conference on computer vision,,* p. 1520–1528, 2015.

[32] Z. Tong, P. Xu and T. Denœux, “Evidential fully convolutional network for semantic segmentation,” *Applied Intelligence,* vol. 51, no. 9, p. 6376–6399, 2021.

[33] M. Manana, C. Tu and P. A. Owolaw, “A Survey on Vehicle Detection Based on Convolution Neural Networks,” 2017.

[34] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE,* vol. 11, no. 86, p. 2278–2324, 1998.

[35] Y. Zhang, Y. Yuan, F. Yachuang and L. Xiaoqiang, “Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 57, no. 8, pp. 5535-5548, 2019.

[36] Y. LeCun, Y. Bengio and G. Hinton, “Deep learning,” *Nature,* vol. 521, no. 7553, pp. 436-444, 2015.

[37] A. Zisserman and B. Simonyan, “Very deep convolutional networks for large- scale image recognition,” *arXiv preprint arXiv:1409.1556,* 2015.

[38] K. He, X. Zhang, R. Shaoqing and J. Sun, “Deep residual learning for image recognition,” in *IEEE conference on computer vision and pattern recognition*, 2016.

[39] L. Ma and e. al., “Deep learning in remote sensing applications: A meta-analysis and review,” *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 152, pp. 166-177, 2019.

[40] T. Nguyen, J. Han and C. Park, “Satellite image classification using convolutional learning,” 2013.

[41] J. Ball, D. Anderson and C. S. Chan, “A Comprehensive Survey of Deep Learning in Remote Sensing: Theories, Tools and Challenges for the Community,” *Journal of Applied Remote Sensing,* vol. 4, no. 11, p. 042609, 2017.

[42] X. Chen, S. Xiang, C.-L. Liu and C.-H. Pan, “Vehicle detection in satellite images by hybrid deep convolutional neural networks.,” *IEEE Geoscience and remote sensing letters,* vol. 11, no. 10, p. 1797–1801, 2014.

[43] Y. Yu, T. Gu, H. Guan, D. Li and S. Jin, “Vehicle detection from high-resolution remote sensing imagery using convolutional capsule networks,” *IEEE Geoscience and Remote Sensing Letters,* vol. 16, no. 12, pp. 1894-1898, 2019.

[44] S. N. Ferdous, M. Moktari and N. Nasser, “Super resolution-assisted deep aerial vehicle detection,” *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications,* vol. 11006, p. 17, 2019.

[45] R. Girshick, J. Donahue, T. Darrell and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” pp. 580-587, 2014.

[46] R. Girshick, “Fast R-CNN,” in *IEEE international conference on computer vision*, 2015.

[47] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems,* pp. 91-99, 2015.

[48] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You only look once: Unified, real-time object detection,” pp. 779-788, 2016.

[49] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed and A. Berg, “SSD: Single shot multibox detector,” in *European conference on computer vision*, Cham, 2016.

[50] A. Mansour, A. Hassan, W. Hussein and E. Said, “Automated vehicle detection in satellite images using deep learning,” vol. 610, 2019.

[51] E. Shelhamer, J. Long and T. Darrel, “Fully convolutional networks for semantic segmentation,” *IEEE Annals of the History of Computing,* vol. 39, pp. 640-651, 2017.

[52] S. Estrada, S. Conjeti, M. Ahmad, N. Navab and M. Reuter, “Competition vs. concatenation in skip connections of fully convolutional networks,” *International Workshop on Machine Learning in Medical Imaging,* pp. 214-222, 2018.

[53] H. Corentin, S. Azimi and N. Merkle, “Road segmentation in SAR satellite images with deep fully convolutional neural networks,” *IEEE Geoscience and Remote Sensing,* vol. 15, no. 12, p. 1867–1871, 2018.

[54] V. Badrinarayanan, A. Kendall and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE transactions on pattern analysis and machine intelligence,* vol. 39, no. 12, p. 2481–2495, 2017.

[55] P. Gudžius, O. Kurasova, V. Darulis and E. Filatovas, “Deep learning based object recognition in satellite imagery,” *Machine Vision and Applications,* vol. 32, no. 4, pp. 1-41, 1 October 2021.

[56] C. Liang-Chieh, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS,” *IEEE transactions on pattern analysis and machine intelligence,* vol. 40, no. 4, p. 834–848, 2017.

[57] C. Liang-Chieh, “Semantic image segmentation with deep convolutional nets and fully connected CRFS,” *arXiv preprint arXiv:1412.7062,* 2014.

[58] K. He, X. Zhang, S. Ren and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE transactions on pattern analysis and machine intelligence,* vol. 37, no. 9, p. 1904–1916, 2015.

[59] C. Chen and e. al., “Rethinking atrous convolution for semantic image segmentation,” in *IEEE: 27th Signal Processing and Communications Applications Conference (SIU)*, 2017.

[60] V. Iglovikov, S. Mushinskiy and V. Osin, “Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition,” *Preprint at https://arxiv.org/abs/1706.06169 (2017).*

[61] Y. Yuan, X. Zhitong and W. Qi, "VSSA-NET: vertical spatial sequence attention network for traffic sign detection," *IEEE transactions on image processing,* vol. 28, no. 7, pp. 3423-3434, 2019.

[62] Z. Zhou, M. Siddiquee, N. Tajbakhsh and J. Liang, "Unet++: A nested UNET architecture for medical image segmentation," *Deep learning in medical image analysis and multimodal learning for clinical decision support,* pp. 3-11, 2018.

[63] I. Delibasoglu and M. Cetin, "Improved U-Nets with inception blocks for building detection," *Journal of Applied Remote Sensing,* vol. 14, no. 4, 2020.

[64] C. Szegedy, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[65] İ. Delibasoglu, "INCSA-UNET: Spatial Attention Inception UNET for Aerial Images Segmentation," *Computing & Informatics,* vol. 40, no. 6, pp. 1244-1262, 2021.

[66] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto and J. Wu, "Unet 3+: A full-scale connected unet for medical image segmentation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[67] R. Shamir, Y. Duchin, J. Kim, G. Sapiro and N. Harel, "Continuous dice coefficient: a method for evaluating probabilistic segmentations," *arXiv preprint arXiv:1906.11031,* 2019.

[68] N. Abraham and M. Khan, "A Novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation," in *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*, 2019.

[69] R. Shang, J. Zhang, L. Jiao, Y. Li, N. Marturi and R. Stolkin, "Multi-scale adaptive feature fusion network for semantic segmentation in remote sensing images," *Remote Sensing,* 2020.

[70] J. Hu, S. Li and S. Gang, "Squeeze-and-Excitation Networks," in *Proceedings of the IEEE conference on computer vision and pattern recognitio*, 2018.

[71] J. Zhang, S. Lin, L. Ding and L. Bruzzone, "Multi-Scale Context Aggregation for Semantic Segmentation of Remote Sensing Images," *Remote Sensing,* 2020.

[72] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306,* 2021.

[73] H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian and M. Wang, "Swin-unet: Unet-like pure transformer for medical image segmentation," *arXiv preprint arXiv:2105.05537,* 2021.

[74] A. Vaswani and e. al., "Attention is all you need," in *Advances in neural information processing systems*, 2017.

[75] A. Dosovitskiy and e. al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929,* 2010.

[76] T. Phan and e. al., "Skin Lesion Segmentation by U-Net with Adaptive Skip Connection and Structural Awareness," *Applied sciences,* 2021.

[77] K. Lee and e. al., "U-Net skip-connection architectures for the automated counting of microplastics," *Neural Computing and Applications,* 2022.

[78] A. Nagwa, P. Songhao, A. Koubaa, A. Noor and A. Afifi, "HTTU-Net: Hybrid Two Track U-Net for automatic brain tumor segmentation," *IEEE Access ,* vol. 8, pp. 101406-101415, 2020.

[79] C. Guo and e. al, "SA-UNet: Spatial Attention U-Net for Retinal Vessel Segmentation," in *2020 25th international conference on pattern recognition (ICPR)*.

[80] D. Cheng, G. Meng, G. Cheng and C. Pan, "SeNet: Structured edge network for sea–land segmentation," *IEEE Geoscience and Remote Sensing Letters,* vol. 14, no. 2, pp. 247-251, 2016.

[81] J. Fu and e. al., "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.

[82] Y. Wei, X. Liu, J. Lei and L. Feng, "Multiscale feature U-Net for remote sensing image segmentation," *Journal of Applied Remote Sensing,* vol. 16, no. 1, 2022.

[83] Q. Hou, D. Zhou and J. Feng, "Coordinate attention for efficient mobile network design," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* p. 13713–13722, 2021.

[84] X. Niu, Q. Zeng, X. Luo and L. Chen, "CAU-net for the semantic segmentation of fine-resolution remotely sensed images," *Remote Sensing,* vol. 14, no. 1, p. 215, 2022.

[85] H. Zhao, J. Shi, X. Qi, X. Wang and J. Jia, "Pyramid scene parsing network," *Proceedings of the IEEE conference on computer vision and pattern recognition,* p. 2881–2890, 2017.

[86] S. Woo, J. Park, J. Lee and I. Kweon, "Cbam: Convolutional block attention module," *Proceedings of the european conference on computer vision (ECCV),* pp. 3-19, 2018.

[87] H. Wu, J. Zhang, K. Huang, K. Liang and Y. Yu, "Fastfcn: Rethinking dilated convolution in the backbone for semantic segmentation," *arXiv preprint arXiv:1903.11816,* 2019.

[88] A. Sehgal and N. Kehtarnavaz, "Guidelines and benchmarks for deployment of deep learning models on smartphones as real-time applications," *Machine Learning and Knowledge Extraction,* vol. 1, no. 1, pp. 450-465, 2019.

[89] Y. Weng, T. Zhou, Y. Li and X. Qiu, "NAS-Unet: Neural Architecture Search for Medical Image Segmentation," *IEEE Access,* vol. 7, pp. 44247-44257, 2019.

[90] X. He, Z. Kaiyong and C. Xiaowen, "AutoML: A survey of the state-of-the-art," *Knowledge-Based Systems,* vol. 212, 2021.

[91] X. He and S. Xu, Process neural networks: Theory and applications, Springer, 2010.

[92] N. Audebert, B. Saux and S. Lefevre, "Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images.," *Remote Sensing,* vol. 9, p. 368, 2017.

[93] Y. Weng, T. Zhou, Y. Li and X. Qiu, "NAS-Unet: Neural Architecture Search for Medical Image Segmentation," *IEEE Access,* vol. 7, pp. 44247-44257, 2019.

[94] B. Baker, O. Gupta, N. Naik and R. Raskar, "Designing neural network architectures using reinforcement learning," *arXiv preprint arXiv:1611.02167,* 2016.

[95] A. Real, Y. Aggarwal, A. Huang and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the AAAI conference on artificial intelligence*, 2019.

[96] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos and X. E. P, "Neural architecture search with bayesian optimisation and optimal transport," *Advances in neural information processing systems,* vol. 31, 2018.

[97] R. Shin, C. Packer and D. Song, "Differentiable neural network architecture search," 2018.

[98] C. Yao and X. Pan, "Neural architecture search based on evolutionary algorithms with fitness approximation," in *International joint conference on neural networks (IJCNN)*, 2021.

[99] Y. Liu, B. Sun, M. Xue, G. Zhang, G. Yen and K. C. Tan, "A survey on evolutionary neural architecture search," *IEEE transactions on neural networks and learning systems,* 2021.

[100] T. Elsken, J. H. Metzen and F. Hutter, "Neural architecture search: A survey," *The Journal of Machine Learning Research,* vol. 20, p. 1997–2017, 2019.

[101] H. Liu, K. Simonyan and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055,* 2018.

[102] Q. Yu, "C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation," in *IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[103] M. M. Bosma, A. Dushatskiy, M. Grewal, T. Alderliesten and P. Bosman, "Mixed-block neural architecture search for medical image segmentation," *Medical imaging 2022: Image processing,* vol. 12032, p. 193–199, 2022.

[104] T. D. Ottelander, A. Dushatskiy, M. Virgolin and P. Bosman, “Local search is a remarkably strong baseline for neural architecture search,” *International conference on evolutionary multi-criterion optimization,* p. 465–479, 2021.

[105] M. Zhang and e. al., “NAS-HRIS: Automatic design and architecture search of neural network for semantic segmentation in remote sensing images,” *Sensors,* vol. 20, no. 18, p. 5292, 2022.

[106] C. Peng, Y. Li, L. Jiao and R. Shang, “Efficient convolutional neural architecture search for remote sensing image scene classification,” *IEEE Transactions on Geoscience and Remote Sensing,* vol. 59, no. 7, p. 6092–6105, 2020.

[107] W. Jing, Q. Ren, J. Zhou and H. Song, “AutoRSISC: Automatic design of neural architecture for remote sensing image scene classification,” *attern Recognition Letters,* vol. 140, p. 186–192, 2020.

[108] E. Jang, S. Gu and B. Poole, “Categorical reparameterization with gumbel- softmax,” *arXiv preprint arXiv:1611.01144,* 2016.

[109] Z. Zhang, S. Liu, Y. Zhang and W. Chen, “RS-DARTS: A convolutional neural architecture search for remote sensing image scene classification,” *Remote Sensing,* vol. 14, no. 1, p. 141, 2021.

[110] A. Van Etten, D. Lindenbaum and T. Bacastow, “pacenet: A remote sensing dataset and challenge series.,” *arXiv preprint arXiv:1807.01232.,* 2018.

[111] P. Gudžius, O. Kurasova, V. Darulis and E. Filatovas, “VUDataScience,” 2020. [Online]. Available: https://github.com/VUDataScience/Deep-learning-based-object-recognition-in-multispectral-satellite-imagery-for-low-latency-applicatio.

[112] O. Adedeji, P. Owoade, O. Ajayi and O. Arowolo, “Image Augmentation for Satellite Images,” *arXiv preprint arXiv:2207.14580,* 2022.

[113] V. Iglovikov, S. Mushinskiy and V. Osin, “Satellite imagery feature detection using deep convolutional neural network: A kaggle competition,” *arXiv preprint arXiv:1706.06169,* 2017.

[114] S. Ruder, “An overview of gradient descent optimization algorithms,” *Preprint at https://arxiv.org/abs/1609.04747,* 2016.

[115] A. Gulli and S. Pal, Deep Learning with Keras, Packt Publishing, 2017.

[116] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[117] D. Mott and R. Tomsett, “Illuminated Decision Trees with Lucid,” *Preprint at https://arxiv.org/abs/1909.05644 (2019).*

[118] L. Zintgraf, T. Cohen, T. Adel and M. Welling, “Visualising deep neural network decisions,” 2017.

[119] X. Zhang, X. Zhou, M. Lin and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” pp. 6848-6856, 2018.

[120] R. Hunger, “Floating Point Operations in Matrix-Vector Calculus,” 2007.

[121] D. Justus, J. Brennan, S. Bonner and A. S. McGough, “Predicting the computational cost of deep learning models,” 2018.

[122] A. Canziani, A. Paszke and E. Culurciello, “An analysis of deep neural network models for practical applications,” *Preprint at https://arxiv.org/abs/1605.07678,* 2016.

[123] J. Cong and B. Xiao, “Minimizing computation in convolutional neural networks,” pp. 281-290, 2014.

[124] K. He, X. Zhang, S. Ren and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” pp. 1026-1034, 2015.

[125] F. Tanner and e. al., “Overhead imagery research data set—An annotated data library & tools to aid in the development of computer vision algorithms,” *IEEE Applied Imagery Pattern Recognition Workshop,* pp. 1-8, 2009.

[126] S. Razakarivony and F. Jurie, “Vehicle detection in aerial imagery: A small target detection benchmark,” *Journal of Visual Communication and Image Representation,* vol. 34, pp. 187-203, 2016.

[127] L. Wan, L. Zheng, H. Huo and T. Fang, “Affine invariant description and large-margin dimensionality reduction for target detection in optical remote sensing images,” *IEEE Geoscience and Remote Sensing Letters,* vol. 14, no. 7, pp. 1116-1120, 2017.

[128] Y. Wang, E. Gu-Yeon and D. Brooks, “Benchmarking TPU, GPU, and CPU platforms for deep learning,” *arXiv preprint arXiv:1907.10701,* 2019.

[129] P. Gudžius, O. Kurasova, V. Darulis and E. Filatovas, “• Gudžius, P., Kurasova, O., DarutoML-based Neural Architecture Search for Object Recognition in Satellite Iulis, V., & Filatovas, E. (2023). AutoML-based Neural Architecture Search for Object Recognition in Satellite Imagery. Remote Sensing, 25(3), 15-31,” *Remote Sensing,* vol. 25, no. 3, pp. 15-31, 2023.

[130] Y. Weng, T. Zhou, Y. Li and X. Qiu, “Nas-unet: Neural architecture search for medical image segmentation,” *IEEE Access,* vol. 7, p. 44247–44257, 2019.

APPENDIX A

Algorithm 1. NAS-MACU Cell genotype generation

1: Generate a random initial $Weights1$ and $Weights2$ values.
2: for *e* := 1 to *E*
3: *genotype* := []
4: *n* := 2
5: *start* := 0
6: if *cell_type == 'down'*:
7: *dim_change* := 2
8: else:
9: *dim_change* := 1
10: *Mask1[0:Weights1.shape[0]] := False*
11: *Mask2[0:Weights2.shape[0]] := False*
12: for *i* := 1 to *N*
13: *normal_op_end := start + n*
14: *up_or_down_op_end := start + dim_ change*
15: *if cell_type == 'down':*
16: *Mask1[up_or_down_op_end:normal_op_end] := True*
17: *Mask2[start:up_or_down_op_end] := True*
18: *else:*
19: *Mask1[up_or_down_op_end + 1:normal_op_end] := True*
20: *Mask1[start:up_or_down_op_end] := True*
21: *Mask2[up_or_down_op_end] : = True*
22: Assign values to $W1$ and $W2$ from $Weights1$ and $Weights2$ masked by $Mask1$ and $Mask2$
23: *edges1:*= assigns the sorted array of $W1$ row indexes, sorted by row max weight values.
24: *L1 := edges1.length*
25: *for j := 1 to L1*
26: *k_best:*= assigns the index of the biggest value from $W1_j$
27: *gene_items1* array appends *(*$W1_{j,k\_best}$*, down_up_operations[k_best], edge index j)*
28: *edges2:*= assigns the sorted array of $W2$ row indexes, sorted by row max weight values.
29: *L2 := edges2.length*

30: *for j := 1 to L2*
31: *k_best:*= assigns the index of the biggest value from $W2_j$
32: *gene_items2* array appends *($W2_{j,k\_best}$, normal_operations[k_best], edge index j)*
33: genotype array appended with the best item from $gene\_items1$ and $gene\_items2$
34: *start = normal_op_end*
35: *n := n + 1*
36: *if genotype_repeats(genotype) >* $Max\_Patience$*:*
37: *Stop training*

7 LIST OF AUTHOR PUBLICATIONS

The results of the dissertation were published in international research journals with a citation index in the Clarivate Analytics Web of Science (CA WoS) database:

- Gudžius, P., Kurasova, O., Darulis, V., & Filatovas, E. (2023). AutoML-based Neural Architecture Search for Object Recognition in Satellite Imagery. *Remote Sensing,* 25(3), 15-31.
- Gudžius, P., Kurasova, O., Darulis, V., & Filatovas, E. (2021). Deep learning-based object recognition in multispectral satellite imagery for real-time applications. *Machine Vision and Applications*, *32*(4), 1-14.

Conference proceedings and abstracts:

- Gudžius, P., Kurasova, O., Darulis, V., & Filatovas, E. (2019) Optimal U-net architecture for object recognition problems in multispectral satellite imagery. *IEEE/ACS 16th international conference on computer systems and applications (AICCSA),* November, Abu Dhabi, UAE.
- Gudžius, P., Kurasova, O., Darulis, V., & Filatovas, E. (2017). Satellite imagery application to financial markets via machine learning. *9th International workshop on Data Analysis Methods for Software Systems (DAMSS)*, December, Druskininkai, Lithuania.

8 SUMMARY IN LITHUANIAN

Palydoviniai vaizdai keičia mūsų suvokimą apie visame pasaulyje vykdomą ekonominę, geopolitinę ir humanitarinę veiklą ir jos prognozavimą. Dėl patobulėjusios optinių palydovų įrangos ir mažesnių orbitinių raketų paleidimo į orbitą sąnaudų geoerdvinio žvalgymo paslaugų pasiūla ir paklausa išaugo. Komerciniai „Airbus Defence and Space“ ir „Maxar technologies“ palydovai suteikė galimybę beveik realiuoju laiku gauti didelės raiškos vaizdus, apimančius visą Žemę ir atveriančius duris naujiems geoerdvinių duomenų ir analitikos taikymo būdams. Tačiau rankiniu būdu analizuoti palydovinių vaizdų petabaitus anotatoriams yra ypač daug pastangų, laiko ir lėšų reikalaujantis darbas. Naujausiuose šią problemą nagrinėjančiuose kompiuterinės regos tyrimuose vis dar trūksta duomenų apie 1) tikslumą ir 2) prognozavimo greitį, o abu šie rodikliai yra labai svarbūs į prognozavimo delsą jautriai reaguojančioms užduotims. Šioje disertacijoje sprendžiame abu minėtus uždavinius, siūlydami objektų atpažinimo modelio projektavimo, mokymo ir sudėtingumo reguliavimo patobulinimus, taikytinus įvairiems neuroniniams tinklams.

Šioje disertacijoje siūloma pilnai konvoliucinio neuroninio tinklo (FCN) architektūros optimizavimo sistema (UNET) tiksliam ir greitam objektų atpažinimui daugiaspektrėse palydovinėse nuotraukose. Parodome, kad FCN yra našesnis už žmogų, o dėl didelio kiekio jutiklių jo tikslumas yra aukščiausio lygio. FCN pranoksta kitus pateiktus metodus šioje specifinėje objektų atpažinimo daugiaspektrėse palydovinėse nuotraukose srityje. Skaičiavimo požiūriu FCN architektūra nesudėtinga skaičiavimų imlumo atžvilgiu, o tai užtikrina penkis kartus trumpesnį mokymo laiką ir greitą prognozavimą, būtiną norint taikyti FCN realiuoju laiku. Siekdami iliustruoti praktinį modelio veiksmingumą, analizuojame jį finansinių produktų algoritminės prekybos aplinkos kontekste.

Ne tik tobuliname ir pritaikome FCN (UNET), bet ir tiriame rankiniu būdu sukurtų neuroninių tinklų trūkumus. Objektų atpažinimo daugiaspektrėse palydovinėse nuotraukose problema pasižymi unikaliomis sudėtingomis erdvinėmis struktūromis ir duomenų rinkinio savybėmis, tokiomis kaip perspektyvos iškraipymas, skiriamosios gebos kintamumas, duomenų spektriškumas ir kitos savybės, dėl kurių konkrečiam žmogaus sugalvotam neuroniniam tinklui sunku pasiekti gerų rezultatų. Norint priderinti prie neuroninio tinklo architektūros, ją reikia iš naujo kalibruoti rankiniu būdu ir atlikti tolesnius konfigūracijos bandymus. Šioje disertacijoje

vertinama ir siūloma, kaip šiuos apribojimus galima išspręsti taikant automatiniu mašininiu mokymusi („AutoML“) pagrįstus metodus.

Paskui nagrinėjame automatinę (AutoML) ir neuronų architektūros paiešką bei siūlome NAS-MACU tipo architektūrą, kuri pašalina šiuos apribojimus automatiškai projektuodama ir pritaikydama neuronų tinklo architektūrą ląstelės lygmeniu. Sukonstruotas NAS-MACU labai gerai veikia mažai informacijos turinčioje aplinkoje, palyginti su rankiniu būdu suprojektuotais tinklais. Galiausiai, siekdami prisidėti prie tolesnio šios mokslinių tyrimų srities plėtojimo, sukūrėme ir atviro šaltinio principu pasidalinome anotuotų palydovinių vaizdų duomenų rinkiniu su mokslininku bendruomenėje dirbančioje šitoje srityje. Disertacijos išvadas ir technologija taip pat galima nesunkiai pritaikyti sprendžiant kitus objektų atpažinimo uždavinius.

8.1 Tyrimo sritis ir problemos aktualumas

Žemės stebėjimo palydovų komiteto (CEOS) duomenimis, komerciniai palydoviniai vaizdai netrukus aprėps visą Žemę, bus transliuojami beveik realiuoju laiku ir didelės raiškos [1] [2]. „Maxar technologies“ komercinių palydovų konsteliacijos, RADARSAT-2 [3], „Pleiades-1 ir ICESat-2 [4]“, „Airbus Defence and Space“ sukurtas „Vision-1“ [5] ir IRSO sukurtas „Cartosat-3“ [6], savo RGB ir panchromatiniuose vaizduose, kurių skiriamoji geba artima didžiausiam leidžiamam teisės aktuose nustatytam tikslumui, t. y. > 25 cm vienam pikseliui [7], apima visą Žemę.

Dėl didėjančio prieinamumo ir įperkamumo palydovinių vaizdų ir aerofotografijų naudojimas įvairiose srityse labai išaugo. Šiuos duomenis naudoja vyriausybės, karinės, žemės ūkio ir finansų pramonės šakos. Tai taipat leidžia ne pelno siekiančioms organizacijoms ir vyriausybėms pasinaudoti šiomis įžvalgomis humanitariniais tikslais, įskaitant vertinti pasaulinės pandemijos ekonominį poveikį (orlaivių, sunkvežimių tiekimo grandinėse, konteinerinių laivų skaičiavimas), greitai aptikti miškų gaisrus [5], atlikti laiko požiūriu jautrų staigių potvynių hidraulinį modeliavimą [7] [6], vykdyti tikslųjį ūkininkavimą, atlikti poveikio aplinkai prevenciją gavybos pramonėje ir stebėjimą teikiant pagalbą nelaimių atveju [8]. Finansų sektoriuje palydoviniai vaizdai naudojami kaip žvalgybos šaltinis kiekybinių rizikos draudimo fondų finansinės prekybos algoritmams, siekiant gauti investicinę grąžą (alfa) [3]. Alfa – tai investicinės strategijos ar portfelio sukurtos perteklinės grąžos matas, apskaičiuotas atsižvelgiant į riziką ir tikėtiną grąžą. Kiekybinių rizikos draudimo fondų kontekste alfa parodo

pridėtinę vertę, kurią sukuria investicijų valdytojo įgūdžiai išnaudoti rinkos neefektyvumą, pritaikant alternatyvius duomenis, pavyzdžiui, palydovinius vaizdus. Beveik realiuoju laiku daryti palydoviniai vaizdai kartu su kompiuterine rega leidžia investicijų valdytojams pasinaudoti objektyviais duomenimis ir numatyti finansinių vertybinių popierių judėjimą viešosiose akcijų rinkose. Praktinio pritaikymo pavyzdžiai apima įmonių pajamų prognozavimą naudojant automobilių skaičiavimo automobilių stovėjimo aikštelėse duomenis, gamybos produkcijos įvertinimą analizuojant tiekimo grandinės veiklą, žemės ūkio prekių kainų prognozavimą įvertinant derlių ir naftos pasiūlos nustatymą stebint pasaulinius naftos rezervuarus [4]. Atsiranda vis daugiau naujų realių naudojimo atvejų, tad didėja ir poreikis kurti labai tikslius ir realiuoju laiku vykdomus kompiuterinės regos metodus [9].

Žmogaus atliekamas anotavimas reikalauja eksponentiškai daugiau išteklių. Remiantis žinomais standartais [10], profesionalus anotatorius per dieną gali anotuoti maždaug 1–2 km^2 palydovinių vaizdų. Tad 100 km^2 palydovinių vaizdų anotavimas vienam anotatoriui užtruktų maždaug 50–100 dienų [11]. Nors naujausi kompiuterinės regos modeliai yra gerokai greitesni, palyginti su anotatoriais, vis tiek reikia nemažai laiko (daugiau kaip 30 minučių) apdoroti maždaug 100 kvadratinių kilometrų palydovinių vaizdų [12]. Be to, tokių modelių tikslumo lygis [13] yra panašus į profesionalių anotatorių (apie 90 %) arba net mažesnis [14] [15] [16].

Dabartiniuose akademiniuose tyrimuose trūksta išsamių metodų objektų atpažinimo modeliams tobulinti, specialiai pritaikytų tokioms palydovinių vaizdų savybėms, kaip atskiros duomenų kategorijos [17] [18]. Dėl unikalių palydovinių vaizdų savybių, tokių kaip perspektyvos iškraipymas, skiriamosios gebos kintamumas, duomenų spektriškumas ir kt., kyla daug iššūkių, nes įvairūs ir išsklaidyti vaizdiniai elementai neleidžia pasiekti gerų rezultatų įprastiems žmogaus išrastiems neuroniniams tinklams. Pastebėti tikslumo ir prognozavimo greičio apribojimai vis labiau kliudo sklandžiai pritaikyti palydovinius vaizdus realiuoju laiku vykdomoms užduotims, pavyzdžiui, algoritminei prekybai finansinių *vertybinių popierių* srityje [19].

Palydovinius vaizdus dabar galima veiksmingai apdoroti naudojant konvoliucinio neuroninio tinklo (CNN) modelius, populiarius giliojo mokymosi metodus, plačiai naudojamus objektų aptikimo ir segmentavimo užduotims atlikti. CNN plačiai taikomi tokiose kompiuterinės regos užduotyse kaip objektų segmentavimas, objektų sekimas, pokyčių aptikimas, pirmojo plano objektų aptikimas, optinis srautas, pozicijos įvertinimas ir semantinis

segmentavimas. Iš šių užduočių semantinis segmentavimas tapo perspektyviausiu metodu, leidžiančiu spręsti gamtos palydovinių vaizdų duomenų keliamus iššūkius. UNET [14], MACU [20] ir panašios rankiniu būdu sukurtos pilnai konvoliucinio tinklo (FCN) architektūros parodė patenkinamų segmentavimo tikslumo rezultatų, ypač didesnių objektų atveju.

Tačiau pažymėtina, kad yra daug dabartinių metodų limitacijų. Šių architektūrų efektyvumas paprastai būna ribotas dėl siauro architektūrinės erdvės tyrinėjimo. Rankiniu būdu suprojektuoti tinklai dėl ribotų tyrėjo žinių, kūrybiškumo ir išteklių paprastai tiria tik didžiulės architektūrinės erdvės poaibį. Šis apribojimas gali neleisti atrasti naujoviškų architektūrų, kurios galėtų užtikrinti didesnį našumą ar efektyvumą. Be to, rankiniu būdu sukurtų FCN, taikomų nematytiems arba nepaskirstytiems duomenims, našumas paprastai mažesnis. Rankiniu būdu atlikto projektavimo metu priimti architektūros sprendimai gali būti paremti mokymo duomenimis, todėl naujų, nematytų imčių atveju rezultatai gali būti prasti. Našumas taip pat nukenčia, kai mokymo duomenų rinkiniai yra palyginti maži (vadinamoji mažai informacijos turinti aplinka), todėl tenka nuolat rankiniu būdu iš naujo kalibruoti ir testuoti konfigūraciją, kad būtų galima atitinkamai pritaikyti neuroninio tinklo architektūrą.

Rankinis tinklo projektavimas labai priklauso nuo tyrėjo patirties ir srities žinių, todėl reikia gerai išmanyti probleminę sritį, architektūros principus ir atitinkamus metodus. Šios žinios gali būti nelengvai perduodamos, todėl išsamių tinklų projektavimo žinių neturintiems tyrėjams kyla sunkumų kuriant optimalias architektūras. O štai „AutoML“ ir neuroninio tinklo architektūros paieškos (NAS) metodais galima sistemingai tirti platesnį architektūros konfigūracijų spektrą.

Šioje disertacijoje nagrinėjami su objektų atpažinimu palydovinėse nuotraukose susiję viršuje paminėti iššūkiai, atsižvelgiant į unikalias palydovinių nuotraukų savybes, rankiniu būdu sukurtų FCN našumo apribojimus ir poreikį greitai bei tiksliai atpažinti įvairių kategorijų ir duomenų rinkinių tipų objektus. Šiems iššūkiams spręsti naudojame NAS kaip automatizuoto mašininio mokymosi („AutoML“) sistemos dalį. NAS metodas leidžia automatiškai ieškoti konkrečiai problemai pritaikytų CNN architektūrų, o tai maksimaliai padidina šio metodo našumą. Atlikdami savo tyrimą, pasinaudodami „AutoML“ ir NAS galimybėmis, pristatome naują NAS-MACU neuroninį tinklą, pranašesnį ir našesnį už iki šiol rankiniu būdu sukurtus tinklus. Šis naujas metodas, NAS-MACU, yra specialiai pritaikytas ir gali pašalinti rankiniu būdu sukurtų CNN trūkumus.

8.2 Tyrimo objektas

Šios disertacijos sritis – objektų atpažinimas palydovinėse nuotraukose naudojant giliojo mokymosi („Deel Learning“) ir automatinio mašininio mokymosi („AutoML“) metodus.

8.3 Disertacijos tikslas

Šioje disertacijoje siekiama pateikti tikslaus ir greito objektų atpažinimo palydovinėse nuotraukose sprendimus taikant giliojo mokymosi ir „AutoML“ metodus.

8.4 Disertacijos uždaviniai

Šios disertacijai uždaviniai:

1. Atlikti išsamią literatūros apžvalgą apie įvairius giliuoju mokymusi pagrįstus objektų atpažinimo palydovinėse nuotraukose metodus.
2. Pateikti giliuoju mokymusi grindžiamą sistemą tikslesniam ir spartesniam objektų atpažinimui palydovinėse nuotraukose, įskaitant išankstinį vaizdo apdorojimą ir pilnai konvoliucinius neuroninius tinklus (FCN).
3. Atlikti eksperimentinį tyrimą, kad įvertintume konvoliucinio neuroninio tinklo tikslumą ir prognozavimo greitį.
4. Atlikti perspektyviausių objektų atpažinimo neuroninių tinklų lyginamąją eksperimentinę analizę.
5. Sukurti „AutoML“ pagrįstą neuronų architektūros paieškos (NAS) metodą, tinkantį objektų atpažinimo palydovinėse nuotraukose problemoms spręsti, kuris būtų pranašesnis už rankiniu būdu suprojektuotus neuronų tinklus, atsižvelgiant į konkrečios problemos apribojimus (pvz., mokymo aplinką su mažai informacijos ir duomenų rinkinio specifiką).

8.5 Mokslinis tyrimo naujumas

- Siekiant padidinti tikslumą ir pagreitinti objektų atpažinimą lengvųjų automobilių klasėje palydovinėse nuotraukose, pasiūlyta giliuoju mokymusi pagrįsta sistema. Sistema apima išankstinį vaizdo apdorojimą, vaizdo elementų ir kadrų sekos nustatymą, hiperparametrų derinimą, tinklo sudėtingumo vertinimą ir UNET koregavimo metodus;
- Atlikta išsami lyginamoji analizė ir eksperimentinis geriausiai veikiančių FCN (UNET, „FastFCN“, „DeepLab“, MACU) tyrimas, taip pat ištirtos

svarbios neuroninio tinklo konstrukcijos ypatybės ir komponentai, gerinantys segmentavimo užduočių atlikimą.

- Pasiūlytas naujas sprendimas (NAS-MACU), pagrįstas automatine neuroninio tinklo architektūros paieška (NAS) ir MACU tinklo pagrindu, kuris gali automatiškai atrasti gerai veikiančią ląstelių topologiją, optimizuotą tiksliam objektų atpažinimui optiniuose palydoviniuose vaizduose.

8.6 Ginamieji teiginiai

1. Siūloma visiškai konvoliucinio neuroninio tinklo modifikacijos principas „Sat-Modification“, padejo sukurti UNET pagrįstą architektūra, kuri dėl savo skaičiavimo požiūriu efektyvios architektūros užtikrina mažiausią su neuroninio tinklo modeliu susijusią palydovinių vaizdų prognozavimo delsą, palyginti su kitais FCN tinklais, įskaitant MACU, „DeepLab“ ir „FastFCN“ tinklus.
2. Pasiūlytas naujas NAS-MACU sprendimas pranoko rankiniu būdu ekspertų mokslininkų sukurtą ir publikuotą MACU tinklą ir užtikrina tikslesnį objektų atpažinimą lengvųjų automobilių klasėje ypač esant mažai informacijos turinčioje aplinkoje.

8.7 Praktinė reikšmė

1. Šiame darbe sukurtas ir viešai paskelbtas originalus palydovinių vaizdų mokymo rinkinys su paženklintais poligonais, skirtas toliau plėtoti šią mokslinių tyrimų sritį. Naudojant profesionalius duomenų anotavimo metodus ir *QGIS* geoerdvinę programinę įrangą sukurtas aukštos kokybės mokymo rinkinys su 80316 paženklintų objektų. Ženklinimą ir poligonų koordinačių generavimą rankiniu būdu atliko keli profesionalūs anotatoriai, o kokybė peržiūrėta ir patikrinta. Atliekant šį tyrimą nebuvo viešai prieinamų didelės skiriamosios gebos palydovinių vaizdų duomenų rinkinių su pažymėtomis „lengvųjų transporto priemonių“ objektų klasėmis.
2. Šiame darbe sprendžiami du svarbūs praktiniai palydovinių vaizdų pritaikymo algoritminėje finansinių vertybinių popierių prekyboje apribojimai: prognozavimo greitis ir didelis tikslumas mažai informacijos turinčioje aplinkoje. Šias realias kliūtis dabar galima išspręsti taikant šioje disertacijoje pasiūlytus praktinius metodus. Pavyzdžiui, metodus skirtus išmatuoti tinklo skaičiavimo sudėtingumą norint padidinti prognozavimo greitį; ir taikant NAS metodus, skirtus nustatyti tiksliausią objektų

prognozavimą užtikrinančią tinklo architektūrą, ypatingai svarbius kai mokymo duomenų kiekis yra ribotas arba brangus.

3. NAS-MACU metodų atradimas gali būti labai naudingas mokslininkams, nes gerokai sutrumpina laiką, reikalingą rasti optimalius neuroninius tinklus objektų atpažinimo užduotims konkrečiose probleminėse srityse. Tai reiškia, kad galima sutaupyti daug laiko tyrimams, sumažinti priklausomybę nuo srities ekspertizės ir pagreitinti pateikimo rinkai laiką. Be to, NAS-MACU patobulinimai gali būti pritaikyti ir kitoms prognozavimo delsai jautrios pramoninėms ir humanitarinėms užduotims.

8.8 Darbo rezultatų aprobavimas

Disertacijos rezultatai paskelbti tarptautiniuose mokslo žurnaluose, turinčiuose citavimo indeksą „Clarivate Analytics Web of Science“ (CA WoS) duomenų bazėje:

- Gudžius, P., Kurasova, O., Darulis, V., & Filatovas, E. (2021). Deep learning-based object recognition in multispectral satellite imagery for real-time applications. *Machine Vision and Applications*, 32(4), 1–14;
- Gudžius, P., Kurasova, O., Darulis, V., & Filatovas, E. (2023). AutoML-based Neural Architecture Search for Object Recognition in Satellite Imagery. *Remote Sensing,* 25(3), 15–31.

Disertacijos rezultatai pristatyti šiose tarptautinėse konferencijose:

- 2018: „International Conference on Control and Computer Vision (ICCCV)“ (Tarptautinė valdymo ir kompiuterinės regos konferencija (ICCCV), lapkritis, 2018, Singapūras;
- 2019: „16th ACS/IEEE International Conference on Computer Systems and Applications“ (16-oji ACS/IEEE tarptautinė kompiuterių sistemų ir programų konferencija, AICCSA), lapkritis, 2019, Abu Dabis, JAE;
- 2019: „Data Science, E-learning and Information Systems“ (Duomenų mokslas, el. mokymasis ir informacinės sistemos), gruodis, 2019, Dubajus, JAE;
- 2022: „The 8th International Conference on Machine Learning, Optimisation, and Data Science“ (8-oji tarptautinė mašinų mokymosi, optimizavimo ir duomenų mokslo konferencija), birželis, 2022, Siena, Italija.

Disertacijos rezultatai pristatyti šioje nacionalinėje konferencijoje:

- 2017: „9th International Workshop on Data Analysis Methods for Software Systems“ (9-toji tarptautinė konferencija „Duomenų analizės metodai programų sistemoms“), gruodis, 2017 Druskininkai, Lietuva.

8.9 Susiję tyrimai ir konvoliuciniai neuroniniai tinklai

Apibendrinant teigtina, kad atsižvelgiant į nustatytus disertacijos tikslus, UNET [55] ir MACU [20] tinklai pasirinkti kaip perspektyviausios architektūros eksperimentiniams tyrimams atlikti. UNET architektūra, iš pradžių sukurta biomedicininiams vaizdams segmentuoti, pasižymėjo perspektyviais rezultatais ir palydovinių vaizdų srityje. Jos unikalus dizainas, praleidžiamosios jungtys ir didinamieji operatoriai leidžia išskirti didelės skiriamosios gebos požymius ir veiksmingai lokalizuoti. UNET pagrindu sukurtos variacijos, tokios kaip UNET++, „Inception-UNET“ ir „UNet3+“, dar labiau patobulino pirminę architektūrą, nes įvedė papildomus sluoksnius ir pagerino požymių išskyrimą.

Kita vertus, MACU tinklas išsiskiria daugiapakopėmis praleidžiamosiomis jungtimis, asimetriniais konvoliuciniais blokais ir dėmesio mechanizmų integravimu. MACU tinklas pasiekė puikių rezultatų nuotolinio stebėjimo duomenų rinkiniuose, pranoko kitas architektūras, pavyzdžiui, FCAU-NET, „PSPNet“ ir „TransUNET“, ir pasiekė panašių rezultatų kaip „DeepLabv3“ ir „FastFCN“. Į UNET pagrindą įtraukus kanalo dėmesio ir asimetrinius konvoliucinius blokus, patobulinamas požymių išskyrimo procesas ir veiksmingai fiksuojama erdvinė ir kanalo informacija.

Tiek UNET, tiek MACU tinklai – perspektyvūs sprendimai vaizdų segmentavimo uždaviniams spręsti, ypač palydovinių vaizdų ir biomedicininių vaizdų apdorojimo srityse. Dėl savo gebėjimo apdoroti aukšto lygio objekto požymių išskyrimą, erdvinę informaciją ir daugiapakopį kontekstą jie tinka tiksliam ir preciziškam segmentavimui. Atliekant eksperimentinius tyrimus su šiomis architektūromis, galima gauti daugiau informacijos apie jų veikimą ir galimus patobulinimus, o tai galiausiai padės tobulinti vaizdų segmentavimo sritį ir jos taikymą įvairiose srityse.

Be to, neuronų architektūros paieška (NAS) yra perspektyvus automatizuoto mašininio mokymosi („AutoML“) metodas, kuriuo sprendžiami rankiniu būdu sukurtų neuronų tinklų architektūrų trūkumai. NAS automatizuoja neuroninių tinklų projektavimo procesą, todėl jis tampa

prieinamas įvairesnėms sritims ir tyrėjams. Tyrinėdamas pagrindinių statybinių blokų, vadinamų „ląstelėmis“, derinius, NAS efektyviau konstruoja sudėtingus neuroninius tinklus. Jame taikomi įvairūs optimizavimo metodai, tokie kaip sustiprintasis mokymasis, evoliuciniai algoritmai ir gradientais pagrįsti metodai. NAS sėkmingai veikė vaizdų klasifikavimo srityje ir parodė savo potencialą medicininių vaizdų segmentavimo ir nuotolinio stebėjimo srityse. Tolesni NAS moksliniai tyrimai ir taikymas „AutoML“ yra perspektyvūs siekiant tobulinti mašininio mokymosi sistemas įvairiose srityse. Tai ir jo santykinį našumą, palyginti su rankiniu būdu sukurtais tinklais, išsamiau nagrinėjame šioje disertacijoje.

8.10 Spręstino uždavinio apibrėžimas

Šiame darbe sprendžiama objektų atpažinimo problema. Objektų atpažinimo rezultatus gauname naudodami vaizdo semantinio segmentavimo metrikas. Dėl palydovinių vaizdų mažos skiriamosios gebos pobūdžio semantinio segmentavimo metodas tinka objektų atpažinimo palydoviniuose vaizduose problemoms spręsti, nes užtikrina detaliausius, vaizdo elementų lygmens rezultatus.

Empiriniam tyrimui pasirinkta objektų klasė yra „lengvasis automobilis“. Šios klasės objektai yra vos 200 vaizdo elementų dydžio (20 x 10 vaizdo elementų matrica, palyginti su milijonais vaizdo elementų įprastuose vaizduose, gautuose, pavyzdžiui, iš „ImageNet“), kaip parodyta 3.1 paveiksliuke. Todėl kiekvienas vaizdo elementas turėtų suteikti vertingos informacijos. 3.1 paveiksliuke mėlynos spalvos vaizdo elementai žymi segmentavimo metodu atpažintą objektų klasę „lengvoji transporto priemonė“; raudona spalva žymi originalų anotatoriaus pažymėtą objekto poligoną; balta spalva žymi tikslų atitikimą vienam vaizdo elementui. Semantinį segmentavimą galima laikyti kiekvieno vaizdo elemento klasifikavimo problema, nes vaizdo elementą klasifikuojame į dvejetainę išvestį (objektas klasėje arba nėra objekto), ir jis neskiria skirtingų to paties objekto egzempliorių.

Objektų atpažinimo rodiklius išvedame iš semantinio vaizdo segmentavimo rezultatų. Segmentuotus „lengvosios transporto priemonės“ vaizdo elementus uždengiame ant žmogaus anotatoriaus sukurtų kaukių (poligonų) duomenų rinkiniuose (mokymo, patvirtinimo ir testavimo duomenų rinkiniuose). Tada nustatome, kurie objektai teisingai atpažinti, o kurie – ne. Objektas gali atrodyti kitaip skirtinguose kontekstuose, esant skirtingoms apšvietimo sąlygoms, kampams ir pan. Žemesnė riba gali leisti

lanksčiau atpažinti objektą nepaisant šių skirtumų ir rankiniu būdu dirbančio anotatoriaus klaidų. Kad objektas būtų laikomas teisingai atpažintu, bent 25 % objekto vaizdo elementų turi būti vienodai uždengti. Ši riba pasirinkta siekiant atsižvelgti į duomenų rinkinyje esančius žmogaus anotatorių ženklinimo netikslumus (kaip matyti 3.1 paveiksliuke) ir reikiamą minimalią ribą. Atlikę empirinį tyrimą (keli lygiai nuo 15 % iki 40 %) nustatėme, kad norint objektą klasifikuoti kaip teisingai atpažintą, pakanka, kad jis atitiktų 25 % anotuoto poligono, kartu sukuriant minimalius klaidingai teigiamus signalus. Kai objektas teisingai atpažįstamas, jis laikomas tikruoju teigiamu objektu (TP) arba kitaip tinkamai klasifikuojamas kaip klaidingai teigiamas objektas (FP), klaidingai neigiamas objektas (FN) arba tikrasis neigiamas objektas (TN). Remiantis šiais pagrindiniais skaičiais taip pat išvesti kiti našumo rodikliai. Šie rodikliai atspindi ir semantinio segmentavimo, ir objektų atpažinimo našumą. Siekiant nuoseklumo, neuroninio tinklo našumui vertinti naudojamos ir semantinio segmentavimo, ir objektų atpažinimo metrikos.

8.11 „Sat-Modification“ proceso apžvalga

Siekdami įgyvendinti vieną iš pagrindinių šios disertacijos tikslų, t. y. pasiūlyti giliuoju mokymusi ir palydovinių vaizdų modifikavimu grindžiamą sistemą tikslumui padidinti ir objektų atpažinimui pagreitinti, įgyvendinome daugybę patobulinimų visuose pirminio apdorojimo ir tinklo projektavimo etapuose (t. y. „Sat-Modification“ procese). Kartu šie patobulinimai leido gauti moderniausių tinklo veikimo rezultatų ir sėkmingai pasiekti minėtą tikslą. Visas procesas nuo palydovinių vaizdų gavimo (P1) iki galutinio signalo generavimo ir pateikimo (P13) pavaizduotas 3.2 paveiksliuke. P1–P4 ir P10 komponentai atspindi duomenų rinkinį ir su palydoviniais vaizdais susijusius procesus, tokius kaip duomenų gavimas, pirminis apdorojimas, papildymas ir kt. Šie komponentai aprašyti 3.2.1–3.2.4 poskyriuose.

P5–P9 komponentai yra šioje disertacijoje siūlomos mokslinio naujumo ir modelių patobulinimo sritys. Aptariame dvi pagrindines tyrimų sritis: 1) Tinklo gylio konstravimas ir požymių išskyrimas prognozavimo tikslumui užtikrinti ir 2) Skaičiavimo sudėtingumo analizė prognozavimo greičiui užtikrinti.

8.12 Palydoviniai vaizdai

Disertacijos 1 skyriuje daugiausia dėmesio skiriama objektų atpažinimo užduočiai palydovinėse nuotraukose naudojamiems duomenims ir

išankstinio duomenų apdorojimo metodams. Tyrime naudojamas neapdorotų palydovinių vaizdų duomenų rinkinys gautas iš „DigitalGlobe WorldView-3“ palydovo per atvirojo kodo duomenų bazę „SpaceNet“. Duomenų rinkinį sudarė didelės skiriamosios gebos daugiasluoksniai vaizdai iš keturių skirtingų vietovių: Paryžiaus, Šanchajaus, Las Vegaso ir Chartumo. Tačiau duomenų rinkinyje nebuvo „lengvųjų transporto priemonių“ klasės objektų anotacijų, kurios yra tyrimo objektas.

Norint sukurti anotuotą lengvųjų transporto priemonių klasės duomenų rinkinį, nemažai anotavimo darbų atlikta rankiniu būdu. Po mažiausiai 350 valandų anotavimo darbo gautas aukštos kokybės mokomasis rinkinys su daugiau kaip 80 000 pažymėtų objektų. Anotacijos atliktos naudojant QGIS geografinio vaizdo programinę įrangą, o keli profesionalūs anotatoriai užtikrino anotacijų kokybę ir tikslumą. Viešai prieinamų duomenų rinkinių su pažymėtomis „lengvųjų transporto priemonių“ klasėmis nebuvo, todėl anotuotas duomenų rinkinys buvo atviras, kad būtų lengviau atlikti tolesnius šios srities tyrimus.

Siekiant padidinti mokymo duomenų įvairovę ir patikimumą, taikomi duomenų papildymo metodai. Šie metodai apėmė pasukimą, perspektyvos iškraipymą, ryškumo ir kontrasto koregavimą, Gauso triukšmo pridėjimą ir skirtingų oro bei atmosferos sąlygų įvedimą. Papildymo procesas padėjo pritaikyti modelį įvairioms sąlygoms ir pagerinti jo veikimą realaus pasaulio scenarijuose.

Dėl ribotos atminties dideli palydoviniai vaizdai apkarpyti į mažesnius 160 x 160 vaizdo elementų dydžio mokymo kadrus. Šie vaizdo elementų kadrai leido gauti didesnes mokymo partijas ir užtikrinti didesnį konteksto kintamumą kiekviename atgalinio skleidimo cikle. Tačiau sujungus nesusijusias scenas į vieną kadrą, gali atsirasti triukšmo, kuris iškraipo kontekstinę informaciją. Šiai problemai spręsti sukurtas programinis sąlyginis metodas „vaizdo elementų rėmelių atranka“. Jis apėmė atsitiktinę mokymo kadrų atranką, kadrų, kurie sutampa su keliais dideliais palydovo vaizdais, atmetimą ir visiškai besidubliuojančių kadrų atmetimą. Šis metodas sumažino triukšmą ir pagerino mokymo bei prognozavimo tikslumą.

Be to, siekiant pagerinti prognozavimą, sukurtas metodas, vadinamas „prognozavimo rėmų sekos nustatymu“. Tai reiškia, kad objektams klasifikuoti reikia atsižvelgti į bent du skirtingus fonus (prognozavimo rėmus). Nesutampant klasifikacijai objektas laikomas teigiamai atpažintu. Lyginamieji eksperimentai parodė, kad įgyvendinus prognozavimo kadrų

sekos nustatymą objektų atpažinimo tikslumas buvo 3,57 % didesnis, palyginti su standartine prognozavimo funkcija.

Apskritai 1 skyriuje pateikiama išsami neapdorotų palydovinių vaizdų, anotavimo proceso, duomenų papildymo metodų, išankstinio duomenų apdorojimo metodų ir jų poveikio objektų atpažinimo palydoviniuose vaizduose užduočiai apžvalga. Šie etapai padėjo pagrindą tolesniems disertacijos skyriams, kuriuose daugiausia dėmesio skiriama automatinei neuronų architektūros paieškai ir NAS-MACU tinklo našumo vertinimui.

8.13 Skaičiavimo aspektai

Be vaizdo elementų kadrų fragmentų įgyvendinimo kontekstiniam kintamumui pagerinti ir praktinių GPU/TPU atminties apribojimų [55] [113], Ronnebergeris ir kiti [11] taip pat siūlo, kad siekdami sumažinti pridėtines išlaidas ir maksimaliai išnaudoti GPU ir TPU atmintį, turėtume teikti pirmenybę dideliam įvesties vaizdo elementų kadrui, o ne dideliam partijos dydžiui, ir eksperimentuoti su mokymo partijos dydžiais nuo 32 iki 192. Be šios taisyklės, taip pat įgyvendintas impulso optimizavimo algoritmas – Adam [114], [115]. Eksperimentai buvo atliekami specialiai mūsų tyrimo problemai sukurtoje „Google Cloud Platform“ (GCP) architektūroje. Siekiant toliau eksperimentuoti su delsos mažinimu, mūsų GCP sistemoje įdiegtos dvi pažangiausios skaičiavimo mašinos – *GPU NVIDIA Tesla* P100 64 GB (1 branduolio) ir *TPU v3-8 128 GB* (8 branduolių).

8.14 Rankiniu būdu sukurti tinklai (UNET ir MACU, „DeepLab“ ir „FastFCN“)

Šiame skyriuje daugiausia dėmesio skiriama rankiniu būdu sukurtiems neuroniniams tinklams, kurie skirti objektams atpažinti palydovinėse nuotraukose. Apibrėžiama objektų atpažinimo palydovinėse nuotraukose problema ir pasirenkama konkreti objektų klasė „lengvoji transporto priemonė“. Skyriuje aptariamos unikalios duomenų rinkinio savybės ir apribojimai, taip pat išankstinio apdorojimo ir papildymo metodai.

Skyriuje vertinamas kelių rankiniu būdu sukurtų neuroninių tinklų architektūrų, įskaitant UNET, MACU, „DeepLab“ ir „FastFCN“, objektų atpažinimo tikslumas ir prognozavimo greitis. Vertinimas atliekamas naudojant semantines vaizdo segmentavimo metrikas, kurios užtikrina vaizdo elemento lygmens našumą. Objektų atpažinimo rezultatams vertinti naudojamos šios metrikos: tikri teigiami objektai (TP), klaidingai teigiami

objektai (FP), tikri neigiami objektai (TN), klaidingai neigiami objektai (FN), Jaccard indeksas, atšaukimas (*Recall*), tikslumas (*Precision*), per didelės prognozės klaida (FPO) ir F_1 kaip bendro tikslumo metrika.

Siekiant padidinti tikslumą ir pagreitinti objektų atpažinimą, siūloma „Sat-Modification“ sistema. Ši sistema apima išankstinio apdorojimo ir tinklo projektavimo etapų patobulinimus. Pateikiama viso objektų atpažinimo palydovinėse nuotraukose proceso darbo eigos schema. Sistemos komponentai apima tinklo gylio konstravimą, požymių išskyrimą prognozavimo tikslumui užtikrinti ir skaičiavimo sudėtingumo analizę prognozavimo greičiui užtikrinti.

Skyriuje taip pat aprašomi tyrime naudoti neapdoroti palydoviniai vaizdai, gauti iš „SpaceNet“ duomenų bazės. Naudojant QGIS programinę įrangą sukurtas rankiniu būdu anotuotas „lengvųjų transporto priemonių“ klasės objektų duomenų rinkinys. Siekiant sukurti įvairesnį duomenų rinkinį, taikyti duomenų papildymo metodai, o siekiant išspręsti GPU / TPU atminties apribojimus ir pagerinti mokymo tikslumą bei prognozavimo tikslumą – pirminio duomenų apdorojimo metodai.

Aptariami skaičiavimo aspektai, pavyzdžiui, vaizdo elementų rėmelių dėmių naudojimas, partijos dydžiai ir Adamo optimizavimo algoritmas. Skyrius baigiamas dviejų pasirinktų neuroninių tinklų architektūrų – UNET ir MACU – įvertinimu ir jų našumu tikslumo ir prognozavimo greičio požiūriu.

Apskritai šiame skyriuje pateikiama išsami rankiniu būdu sukurtų neuroninių tinklų, skirtų objektams atpažinti palydovinėse nuotraukose, apžvalga, apimanti tokius aspektus kaip tikslumo vertinimas, skaičiavimo sudėtingumas, duomenų rinkinio savybės, pirminis apdorojimas, papildymas ir siūloma „Sat-Modification“ sistema.

8.15 Skaičiavimo sudėtingumas

Šiame skyriuje daugiausia dėmesio skiriama objektų atpažinimo modelių vertinimui ir analizei, atsižvelgiant į jų tikslumą ir skaičiavimo sudėtingumą. Aptariama didelė prognozės signalo delsa, kurią sukelia lėti objektų atpažinimo modeliai, pabrėžiant skaičiavimo sudėtingumą ir skaičiavimo galią kaip veiksnius, turinčius įtakos objektų atpažinimo greičiui. Kaip skaičiavimo sudėtingumo matas įvedamas slankiojo kablelio operacijų skaičius (FLOP).

Skyriuje pateikiamas modelio kompleksiškumui skaičiuoti pritaikytas neuroninio tinklo skaičiavimo sudėtingumo apskaičiavimo metodas, pagrįstas FLOP skaičiumi. Modelio sudėtingumas apibrėžiamas kaip kiekvieno tinklo sluoksnio FLOP skaičių suma. Modelio sudėtingumo svarba aptariama atsižvelgiant į perteklinį pritaikymą, sąnaudas ir efektyvumą.

Keturių siūlomų UNET architektūrų veikimas lyginamas taikant tikslumo, per didelio prognozavimo, Jaccard indekso ir skaičiavimo sudėtingumo kriterijus. UNET_Model_2 pasiekia didžiausią objektų atpažinimo tikslumą (TPO) – 97,67 %, o jo skaičiavimo sudėtingumas palyginti nedidelis (6,9832 G-FLOPS). Siekiant pagerinti UNET_Model_2 veikimą, išbandomos įvairios aktyvacijos funkcijos. Nustatyta, kad ReLU aktyvacijos funkcija užtikrina geriausius tikslumo rezultatus, o hiperbolinio tangento (Tanh) aktyvacijos funkcija sumažina perteklinį prognozavimą ir išlaiko aukštą TPO / FPO santykį. Analizuojamas UNET_Model_2 mokymo procesas ir nustatomas optimalus 35–40 epochų intervalas, siekiant išvengti perteklinio tinklo persimokymo ir sumažinti skaičiavimo sąnaudas. Siūlomas metodas lyginamas su kitais populiariais objektų atpažinimo metodais naudojant išorinius duomenų rinkinius. Siūloma architektūra pasiekia didžiausią tikslumą visuose duomenų rinkiniuose ir metoduose, tuo pat metu naudodama gerokai mažiau epochų ir sumažindama skaičiavimo sąnaudas.

Prognozavimo greičio eksperimentai atliekami naudojant GPU ir TPU skaičiavimo architektūras. Dėl mažesnio skaičiavimo sudėtingumo UNET_Model_2 prognozavimo greičio GPU lenkia TPU. Analizuojamas ryšys tarp objektų atpažinimo tikslumo, skaičiavimo sudėtingumo ir prognozavimo greičio. Nustatyta, kad UNET_Model_2 su 128 x 128 vaizdo elementų dydžio kadru yra optimalus tinklas realiuoju laiku vykdomoms užduotims, nes užtikrina didelį tikslumą, žemą perteklinio prognozavimo lygį ir didelį prognozavimo greitį GPU procesoriuje. Apskritai šiame skyriuje pateikiama išsami objektų atpažinimo modelių analizė, atsižvelgiant į jų tikslumą ir skaičiavimo sudėtingumą. Išvados padeda suprasti, kaip efektyviai kurti modelius ir rasti kompromisą tarp tikslumo ir skaičiavimo išteklių realiuoju laiku vykdomoms užduotims.

8.16 Daugialypės jungties ir asimetrine konvoliucija pagrįstas tinklas (MACU)

Šioje disertacijoje atlikome eksperimentus su keturiais neuroniniais tinklais (MACU, „FastFCN“, UNET ir „DeepLabv3“) trijose skirtingo informacijos intensyvumo aplinkose, kad įvertintume jų prognozavimo

tikslumą ir skaičiavimo sudėtingumą. Tinklai pritaikyti „Google Cloud Platform“ architektūrai ir išbandyti su palydovinių vaizdų duomenų rinkiniu. Tinklus vertinome remdamiesi segmentavimo ir objektų atpažinimo metrikomis, gautomis iš eksperimentinių rezultatų.

Iš 3.5 lentelėje pateiktų rezultatų matyti, kad MACU pasižymėjo geriausiais bendrais rezultatais visose aplinkose, vertinant pagal F_1 balą. UNET geriausiai pasirodė pagal atšaukimo rodiklį, todėl tinka tais atvejais, kai palydovinėse nuotraukose labai svarbu nustatyti kuo daugiau objektų. F_1 balas leidžia visapusiškai įvertinti tinklo našumą, ypač sprendžiant realias problemas, o tikslumas nurodo teisingai nuspėtus objektus.

Be to, „DeepLabv3“ ir „FastFCN“ parodė nedidelio tikslumo rezultatus, kai objektų skaičius mažiausias, tačiau jų prognozės konservatyvios, o perteklinio prognozavimo klaida mažiausia dviejuose iš trijų informacijos intensyvumo scenarijų. 3.14 pav. pateiktas vizualus keturių tinklų gautų rezultatų lyginimas. 3.6 lentelėje palyginome UNET ir MACU prognozavimo greitį ir tikslumą. Nors MACU pranoko UNET pagal tikslumą, jo architektūra buvo sudėtinga skaičiavimo požiūriu, todėl prognozavimo greitis buvo 6,92 kartus lėtesnis už UNET. Tad MACU labiau tinka į tikslumą orientuotoms užduotims, o UNET dėl savo nedidelio skaičiavimo sudėtingumo geriau veikia prognozavimo delsai jautriuose scenarijuose.

Apibendrinant galima teigti, kad MACU pasirinktas kaip perspektyviausia tinklo architektūra ir pagrindas tolesniems „AutoML“ ir NAS metodų tyrimams. Skyriuje pabrėžiama rankiniu būdu projektuojamų tinklų svarba, pripažįstama, kad tokiems tinklams sukurti ir kalibruoti reikia daug laiko. Kitame skyriuje daugiausia dėmesio skiriama naujiems sprendimams naudojant „AutoML“ ir NAS, siekiant pagerinti rankiniu būdu projektuojamų tinklų veikimą.

8.17 Automatizuota neuronų architektūros paieška objektams atpažinti

Šios disertacijos 4.2 skyriuje nagrinėjamas naujo neuronų architektūros paieškos (NAS) metodo, vadinamo NAS-MACU (neuronų architektūros paieška su daugiapakopiu dėmesiu ir kryžminiu panaudojimu), kūrimas ir įvertinimas objektams atpažinti palydovinėse nuotraukose. Skyrius pradedamas aptariant NAS galimybes automatizuoti neuroninių tinklų architektūrų projektavimą. Pabrėžiami iššūkiai, su kuriais susiduria žmogaus ir mašinos mokymosi specialistai, siekdami moderniausių objektų atpažinimo

užduočių rezultatų, ir pristatomas NAS kaip sprendimas šiam procesui automatizuoti ir optimizuoti.

Aprašomas siūlomas NAS-MACU metodas, kurį sudaro trys pagrindiniai komponentai: paieškos erdvė, paieškos strategija ir vertinimo metrika. Paieškos erdvė apima galimas architektūras, kurias tiria NAS algoritmas, o paieškos strategija sujungia sustiprintą mokymąsi ir evoliucinius algoritmus, kuriais vadovaujamasi paieškos procese. Vertinimo metrika matuojamas NAS algoritmo rastų architektūrų našumas, paprastai naudojant tikslumą pagal patvirtinimo duomenų rinkinį.

Toliau paaiškinamas NAS-MACU algoritmas, kuris pradedamas nuo atsitiktinio architektūrų generavimo paieškos erdvėje. Algoritmas iteratyviai gerina šių architektūrų našumą ir grąžina geriausią. NAS-MACU algoritmas pagrįstas ląstelėmis paremta architektūra, o paieška sutelkta į MACU tinklo ląstelės lygmens topologijos optimizavimą.

Aprašomas NAS-MACU projektavimo procesas, pabrėžiami iššūkiai, susiję su giliojo mokymosi architektūros, pritaikytos konkrečioms nuotolinio stebėjimo duomenų užduotims, kūrimu. MACU architektūra įvardijama kaip perspektyvus pagrindas, o NAS-MACU siekiama optimizuoti ląstelių lygmens architektūrą. Skyriuje pristatomas NAS-MACU konstravimo procesas, pabrėžiant savaiminio projektavimo-topologijos metodą, kuris prisitaiko prie įvairių duomenų rinkinių savybių be žmogaus patirties ar rankinio įsikišimo.

Skyriuje taip pat aptariama ląstelės lygio topologijos paieška, kurios metu sukuriamas nukreiptas aciklinis grafas (DAG), vaizduojantis ląstelės architektūrą. DAG naudojamos įvairių tipų operacijos, pavyzdžiui, žemyn, aukštyn ir įprastos, o mišri operacija apibrėžiama remiantis kandidatų operacijų svertiniais deriniais. Pateikiamas ląstelės genotipo generavimo algoritmas, nurodant paieškos ir atrankos proceso etapus.

Toliau pristatomas MACU ir NAS-MACU architektūrų palyginimas, išryškinant NAS-MACU pranašumus automatizuojant ląstelių lygmens architektūros projektavimą. Eksperimentiniais ciklais sukuriami įvairūs NAS-MACU genotipai ir įvertinamas jų santykinis našumas. Išskiriami geriausiai veikiantys genotipai – NAS-MACU-V7 ir NAS-MACU-V8.

Apskritai šiame skyriuje parodytas NAS-MACU efektyvumas automatiškai projektuojant neuroninių tinklų architektūras objektams atpažinti palydovinėse nuotraukose. NAS-MACU metodas naudoja MACU

pagrindą ir optimizuoja ląstelių lygmens topologiją, taikydamas paieškos ir atrankos procesą. Rezultatai rodo NAS-MACU potencialą siekiant geresnių tikslumo rezultatų, palyginti su rankiniu būdu sukurtomis architektūromis. Skyriaus pabaigoje siūlomos būsimos mokslinių tyrimų kryptys ir NAS-MACU reikšmė nuotolinio stebėjimo srityje.

8.18 Eksperimentinis tyrimas ir NAS-MACU

Šios daktaro disertacijos 4 skyriuje pristatomas naujas NAS-MACU metodas, skirtas automatizuotai neuronų architektūros (NAS) paieškai objektų atpažinimo užduotyse naudojant palydovinius vaizdus. NAS-MACU veikimas įvertintas su visu duomenų rinkiniu, naudojant aštuonis skirtingus genotipus, ir palygintas su rankiniu būdu sukurtais tinklais (MACU). Rezultatai parodė, kad lyginant NAS-MACU-V1 su NAS-MACU-V7 ir NAS-MACU-V8 gerokai pagerėjo įvairios metrikos.

NAS-MACU-V8 pasiekė geriausią F_1 rezultatą ir parodė panašų našumą kaip ir NAS-MACU-V7. NAS-MACU metodas pademonstravo gebėjimą greitai mokytis, kai mokymo informacijos intensyvumas nedidelis, todėl jis naudingas tais atvejais, kai gauti mokymo duomenis yra sudėtinga ar brangu, pavyzdžiui, naudojant didelės skiriamosios gebos palydovinius vaizdus. Eksperimentai parodė, kad NAS-MACU pasiekė aukščiausią našumą vos per 15–20 epochų.

Lyginant NAS-MACU ir MACU našumą pagal objektų atpažinimo rodiklius, NAS-MACU šioje konkrečioje užduotyje ir duomenų rinkinyje pranoko MACU. Jis ypač gerai veikė mažo informacijos intensyvumo aplinkoje. Išryškėjo NAS-MACU skaičiavimo efektyvumas, nes labai efektyvi NAS-MACU infrastruktūra sukurta vos per kelias valandas naudojant „AutoML“ procesą „Google Cloud Platform“ (GCP) platformoje. O štai rankiniu būdu projektuoti tinklus ir atlikti objektų atpažinimo užduotis paprastai prireikia mėnesių.

4.4 lentelėje pateikiamas NAS-MACU-V8 ir MACU našumo lyginimas esant skirtingiems mokymo aibės dydžiams. Mažėjant mokymo aibės dydžiui, NAS-MACU-V8 demonstravo geresnius rezultatus, palyginti su MACU, ypač mažai informacijos turinčiose aplinkose. Empirinis tyrimas patvirtino, kad NAS-MACU-V8 pranoko MACU, ypač mažai informacijos turinčiuose scenarijuose. Svarbiausi rodikliai buvo F_1 (bendras našumas) ir tikslumas. NAS-MACU-V8 pranoko MACU pagal bendrą tikslumo (F_1) ir tikslumo

(*Precision*) rodiklius, o mažėjant mokymo aibės dydžiui našumo atotrūkis didėjo.

NAS procesas užtruko nuo 4 iki 58 valandų mokymosi ir paieškos NAS-MACU-V1–NAS-MACU-V8 genotipuose. Svarbu tai, kad NAS-MACU sprendimas sukurtas automatiškai, be žmogaus įsikišimo, todėl yra taikytinas plačiu mastu ir tinka įvairiems realiems poreikiams. 4.11 paveikslėlyje pateiktas vaizdinis MACU ir NAS-MACU-V8 našumo lyginimas dviejuose palydoviniuose vaizduose. NAS-MACU-V8 pasiekė geresnių rezultatų nei MACU, ypač prasto apšvietimo aplinkybėmis.

Apibendrinant, 4 skyriuje pristatomas naujas NAS-MACU metodas, skirtas automatizuotai neuronų architektūros paieškai objektų atpažinimo užduotyse naudojant palydovinius vaizdus. NAS-MACU pranoko kitus rankiniu būdu sukurtus tinklus ir pademonstravo geresnius rezultatus mažai informacijos turinčioje aplinkoje. NAS-MACU tinklo konfigūracijos pasiekė geresnių objektų atpažinimo rezultatų, palyginti su MACU, o NAS-MACU-V8 parodė geriausius rezultatus pagal F_1 metriką. NAS-MACU metodas leidžia savarankiškai atrasti gerai veikiančias ląstelių topologijas ir architektūras, optimizuotas objektams atpažinti daugiaspektriuose palydoviniuose vaizduose.

8.19 Apibendrinimas ir išvados

Šiame skyriuje siekiama kritiškai įvertinti numatytus disertacijos tikslus ir parodyti, kaip šie tikslai sėkmingai įgyvendinti tiek individualiai, tiek kolektyviai.

1 išvada. Neuroninio tinklo modifikavimo procesas „Sat-modification“ pagerino objektų atpažinimo palydoviniuose vaizduose tikslumą ir greitį.

Į „Sat-Modification“ sistemą įtraukus naujus metodus neuroninių tinklų galimybėms didinti, tarp jų požymių išskyrimą, tinklo sudėtingumo matavimą, mokymo proceso derinimą ir prognozavimo greičio optimizavimą, gerokai pagerėjo sistemos tikslumas ir efektyvumas. Ypač svarbu, kad sistemos UNET architektūra pasiekė 97,67 % tikslumą „lengvųjų transporto priemonių“ objektų klasėje. Be to, skaičiavimo požiūriu nesudėtinga UNET_Model_2 architektūra pademonstravo net penkis kartus trumpesnį mokymo laiką, tad ją galima taikyti realiuoju laiku. Šie rezultatai rodo, kad „Sat-Modification“ sistema yra veiksminga didinant objektų atpažinimo galimybes sudėtingoje palydovinių vaizdų srityje.

2 išvada. Dėl savo lengvos skaičiavimo architektūros UNET yra perspektyvus modelis ypač kai prognozavimo laikas yra svarbus aspektas praktiniam uždaviniui spręsti.

Objektų atpažinimo modelių skaičiavimo sudėtingumas turi didelę įtaką prognozavimo laikui ir bendram našumui. Slankiųjų operacijų skaičius (FLOP) – veiksmingas skaičiavimo sudėtingumo matas, kurį galima naudoti išvados trukmei įvertinti. Ne tokie sudėtingi modeliai sumažina ne tik prognozavimo gaištį, bet ir perteklinį pritaikymą, sąnaudas bei didina efektyvumą. UNET_Model_2 rezultatai geriausi prognozavimo spartos požiūriu, tačiau pastebėtas perteklinis prognozavimas ir palyginti nedidelis sudėtingumas – 6,9832 G-FLOP. Veiksmingumui įtakos turi ir aktyvacijos funkcijos pasirinkimas: ReLU užtikrina geriausią tikslumą, o Tanh – mažiausią triukšmo lygį. Be to, nustatytas optimalus 35–40 epochų intervalas, kad būtų kuo labiau sumažintas perteklinis pritaikymas ir mokymo skaičiavimo sąnaudos. Prognozavimo greičio eksperimentai parodė, kad UNET_Model_2 atveju GPU pranoko TPU. Tiesiogiai lyginant su MACU, UNET prognozavo septynis kartus (6,92x) greičiau už MACU (17,53 sek. lyginant su 121,28 sek.), o tikslumo skirtumas siekė tik iki 1 % (F_1 0,939 lyginant su 0,943). Šie rezultatai leidžia daryti išvadą, kad UNET_Model_2 procesas naudojant „Sat-Modification“ sistemą yra tinkamiausia tinklo architektūra ir metodas naudojimo atvejams, jautriai reaguojantiems į išvadų laiką, pavyzdžiui, algoritminei prekybai.

3 išvada. MACU pranoko kitus rankiniu būdu sukurtus tinklus pagal bendrus tikslumo rodiklius ir pasirinktas kaip NAS pagrindas.

Šiame tyrime atlikome eksperimentus, siekdami palyginti keturių neuroninių tinklų – MACU, „FastFCN“, UNET ir „DeepLabv3“ – veikimą skirtingo informacijos intensyvumo aplinkose. Analizuoti ir lyginti gauti segmentavimo ir objektų atpažinimo rodiklių rezultatai. „DeepLabv3“ ir „FastFCN“ pasižymi vidutiniu tikslumu esant mažesniam objektų skaičiui, tačiau išlieka konservatyvūs ir lemia mažiau pervertinimo klaidų. Mūsų išvados rodo, kad MACU tinklas bendras našumas, matuojamas F_1 balu, yra geriausias visose trijose informacijos intensyvumo aplinkose. Remiantis šiomis išvadomis, MACU tinklas pasirinktas kaip perspektyviausia architektūra tolesniems „AutoML“ ir NAS tyrimams.

Išvada 4. Pasiūlytas naujas NAS-MACU modelis užtikrina tikslesnį objektų atpažinimą lengvosios transporto priemonės objektų klasėje,

mažo informacijos intensyvumo aplinkoje, palyginti su rankiniu būdu sukurtu MACU tinklu,.

NAS-MACU tinklo, apimančio automatizuotus neuronų architektūros paieškos (NAS) metodus, sukūrimas yra reikšmingas indėlis į objektų atpažinimo palydovinėse nuotraukose sritį. Naudojant NAS gautos kelios NAS-MACU tinklo konfigūracijos, pranokstančios rankiniu būdu suprojektuoto MACU tinklo našumą ypač mažai informacijos turinčioje aplinkoje. Pažymėtina, kad NAS-MACU-V8 pasiekė geriausią F_1 balą – 0,934, o tai įrodo NAS efektyvumą optimizuojant tinklo architektūras, skirtas lengvųjų transporto priemonių objektų klasėms atpažinti optinėse daugiaspektrėse palydovinėse nuotraukose. Automatizuodamas gerai veikiančių tinklo ląstelių topologijų atradimą, NAS įgyvendinimas MACU tinkle pašalina rankinio įsikišimo poreikį ir supaprastina architektūros optimizavimo procesą. Šios išvados rodo NAS metodų potencialą gerokai padidinti neuroninių tinklų našumą ir efektyvumą palydovinių vaizdų objektų atpažinimo srityje.

Šioje disertacijoje sėkmingai pasiekti užsibrėžti tikslai, išsamiai ištyrus siūlomą „Sat-Modification“ sistemą, atlikus griežtą neuroninių tinklų architektūrų lyginamąją analizę, sukūrus modelį skaičiavimo sudėtingumui ir prognozavimo greičiui įvertinti bei suprojektavus inovatyvų „AutoML“ pagrįstą NAS-MACU tinklą. Šiais darbais pasiekta pažanga „AutoML“ srityje atpažįstant objektus palydovinėse nuotraukose, užtikrinant didesnį tikslumą, prognozavimo greitį ir automatizuotas architektūras, pritaikytas unikalioms ir išsklaidytoms objektų atpažinimo optinėse palydovinėse nuotraukose taikymo sritims.

8.20 Tolesni darbai

Giliojo mokymosi modelių veiksmingumą sprendžiant realaus pasaulio problemas riboja viešai prieinamų palydovinių vaizdų duomenų trūkumas. Šiame tyrime eksperimentas atliktas „Google Cloud“ platformoje, naudojant ribotus skaičiavimo išteklius. Nors atliekant šiuos eksperimentus skaičiavimams prireikė nemažai laiko, ateityje būtų galima pasinaudoti didesniais skaičiavimo ištekliais. Toks išplėtimas padėtų ištirti platesnę ląstelių infrastruktūros paieškos erdvę ir padidinti ląstelių gylį. Be to, tai leistų sušvelninti apribojimus, kuriuos nustato ribojantys hiperparametrai, tokie kaip „max_patience“ ir „Total Epochs“. Be to, siekiant padidinti mažo vėlavimo našumą, galima toliau tirti tris papildomus su neuroninio tinklo architektura

nesusijusius metodus norint gerinti modelio greitį: modelio glaudinimą, procesorių klasteriu spartinimą ir programinės įrangos optimizavimą.

Atsižvelgiant į apribojimus, susijusius su optiniais daugiaspektriais palydoviniais vaizdais, ypač dėl atmosferos ir saulės šviesos sąlygų, būtų naudinga ištirti mokslinių tyrimų galimybes naudojant sintetinės apertūros radaro (SAR) palydovinius vaizdus. SAR vaizdai suteikia galimybę fiksuoti duomenis per debesis, nakties metu ir esant rūkui. Atlikus šiuos tyrimus būtų galima taikyti patobulintas NAS-MACU architektūras.

Be to, galima toliau tirti alternatyvias neuroniniais tinklais pagrįstas architektūras naudojant NAS. Šie tyrimai galėtų neapsiriboti vien objektų atpažinimu palydovinėse nuotraukose ir apimti kitas sritis, pavyzdžiui, medicininių vaizdų analizę (pvz., navikų aptikimą), aerofotografijų apdorojimą (pvz., semantinį segmentavimą bepiločių orlaivių vaizduose), kriminalistiką (pvz., rašysenos aptikimą), autonomines mašinas (pvz., mašinų navigaciją tam tikroje aplinkoje) ir kitas atitinkamas sritis.

Povilas Gudžius

Automated Machine Learning for Accurate and Low-latency Object Recognition in Optical Satellite Imagery

Doctoral Dissertation
Technological Sciences
Informatics Engineering (T 007)
Thesis Editor: Zuzana Šiušaitė

Povilas Gudžius

Automatinis mašininis mokymasis, skirtas tiksliam ir greitam objektų atpažinimui optiniuose palydoviniuose vaizduose

Daktaro disertacija
Technologijos mokslai
Informatikos inžinerija (T 007)
Santraukos redaktorė Jorūnė Rimeisytė-Nekrašienė

Vilniaus universiteto leidykla
Saulėtekio al. 9, III rūmai, LT-10222 Vilnius
El. p. info@leidykla.vu.lt, www.leidykla.vu.lt
bookshop.vu.lt, journals.vu.lt
Tiražas 20 egz.