Machine Vision and Applications (2021) 32:98
https://doi.org/10.1007/s00138-021-01209-2

ORIGINAL PAPER

Deep learning-based object recognition in multispectral satellite imagery for real-time applications

Povilas Gudžius[1] · Olga Kurasova[1] · Vytenis Darulis[1] · Ernestas Filatovas[1]

Received: 25 May 2020 / Revised: 27 January 2021 / Accepted: 3 May 2021 / Published online: 22 June 2021

Abstract

Satellite imagery is changing the way we understand and predict economic activity in the world. Advancements in satellite hardware and low-cost rocket launches have enabled near-real-time, high-resolution images covering the entire Earth. It is too labour-intensive, time-consuming and expensive for human annotators to analyse petabytes of satellite imagery manually. Current computer vision research exploring this problem still lack accuracy and prediction speed, both significantly important metrics for latency-sensitive automatized industrial applications. Here we address both of these challenges by proposing a set of improvements to the object recognition model design, training and complexity regularisation, applicable to a range of neural networks. Furthermore, we propose a fully convolutional neural network (FCN) architecture optimised for accurate and accelerated object recognition in multispectral satellite imagery. We show that our FCN exceeds human-level performance with state-of-the-art 97.67% accuracy over multiple sensors, it is able to generalize across dispersed scenery and outperforms other proposed methods to date. 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 algorithmic trading environment. Additionally, we publish a proprietary annotated satellite imagery dataset for further development in this research field. Our findings can be readily implemented for other real-time applications too.

1 Introduction

Quantamental hedge funds utilize satellite imagery as a source of intelligence for their financial trading algorithms in order to generate access returns (alpha) [40]. Near-real-time satellite imagery combined with computer vision enables investment managers to leverage insights from the "ground-truth" data to predict financial markets. Real-life uses cases include company's revenue prediction based on car count across parking lots; production estimate based on supply chain activity; agricultural commodity prices forecast based on estimated crop yields and oil supply detection based on global oil tank lids [31]. It also enables government entities and non-profits to leverage these insights for humanitarian purposes including assessment of Coronavirus (COVID-19) economic impact (object count of aircraft, lorry, container ship), rapid forest wildfires detection [15], global whale count and extinction prevention [19], time-sensitive flash flood hydraulic modelling [10] and other surveillance for disaster relief [56].

According to The Committee on Earth Observation Satellites (CEOS) [4] commercial satellite imagery will soon reach the coverage of the entire Earth, near-real-time frequency and high-resolution (< 30 cm per pixel). It will consequently increase the demand for the development of high-precision and real-time computer vision techniques [13]. Most recent computer vision models, however, still take a significant time (> 30 min) to process ~ 100 km^2 of satellite imagery [64] with accuracy similar or below [39] the professional human annotator (~ 90%) [48, 32, 42, 14]. Also, current academic research, lacks methods improving object recognition models suited for this purpose [28, 59] increasing the bottleneck for satellite imagery adoption in real-time applications such as algorithmic trading [40, 8]. In Fig. 1 we illustrate the data flow in the algorithmic trading system and identify signal origination latency per data input to illustrate this bottleneck: market data (< 40 ms delay), non-market

✉ Povilas Gudžius
povilas.gudzius@mif.vu.lt

✉ Olga Kurasova
olga.kurasova@mif.vu.lt

Ernestas Filatovas
ernestas.filatovas@mif.vu.lt

[1] Institute of Data Science and Digital Technologies, Vilnius University, Akademijos street 4, 08412 Vilnius, Lithuania

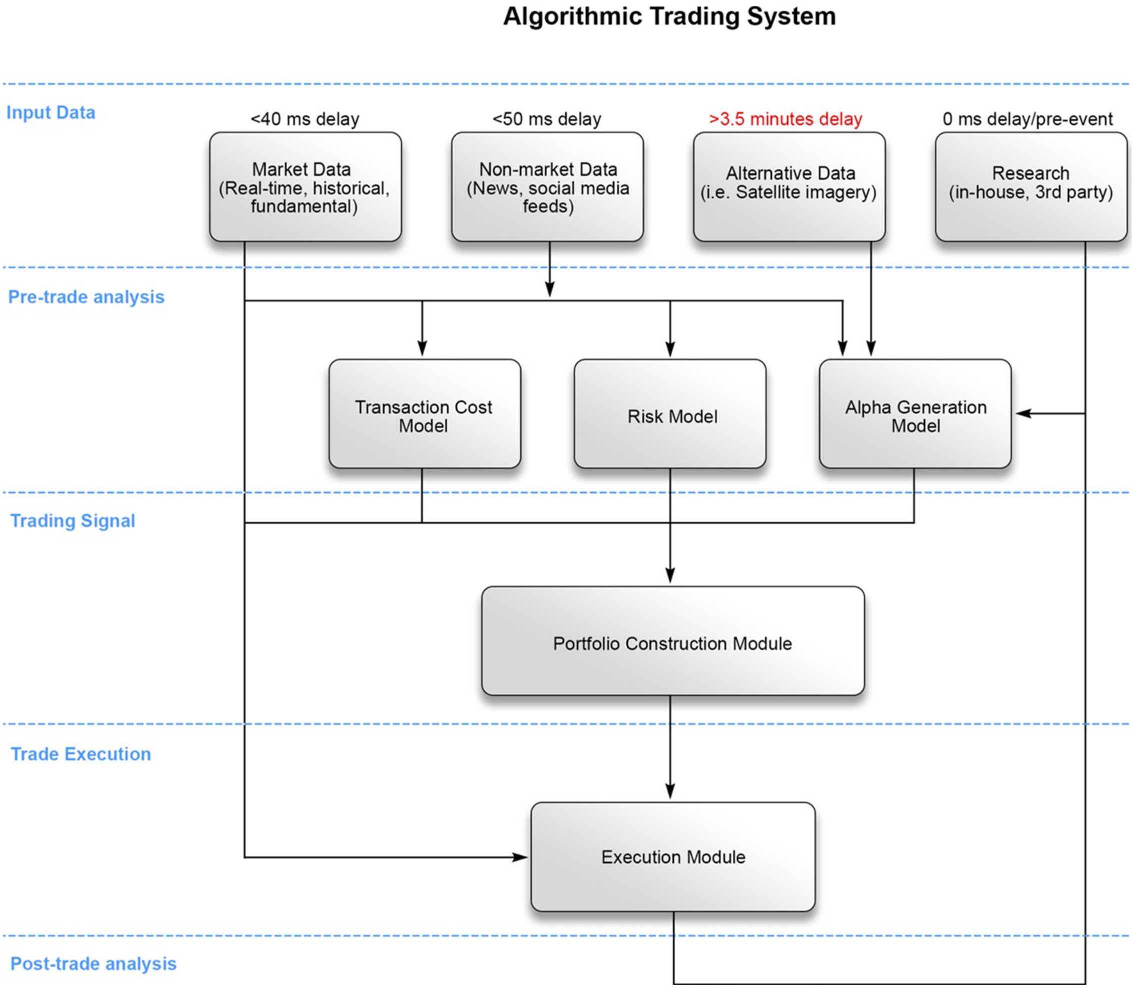

Algorithmic Trading System
Input Data
<40 ms delay
<50 ms delay
>3.5 minutes delay
0 ms delay/pre-event
Market Data (Real-time, historical, fundamental)
Non-market Data (News, social media feeds)
Alternative Data (i.e. Satellite imagery)
Research (in-house, 3rd party)
Pre-trade analysis
Transaction Cost Model
Risk Model
Alpha Generation Model
Trading Signal
Portfolio Construction Module
Trade Execution
Execution Module
Post-trade analysis

Fig. 1 Signal generation bottleneck from satellite imagery data in algorithmic trading system. Modified with an additional layer based on Cliff et al. [8]

data (< 50 ms delay), satellite imagery (> 3.5 min delay), research-based metrics (no delay/pre-event).

Here we propose an improved fully convolutional neural network (FCN) architecture (subtype: U-net) for a semantic segmentation task that allows us to significantly improve accuracy and speed. For empirical investigation, we utilize specific problem: “light-vehicle recognition in satellite imagery”. “Light-vehicle” object recognition requires the highest precision (due to a small object size of ~ 120 pixels only) as well as the model’s ability to generalize across dispersed scenes. This particular problem was selected because the solution could be applicable to larger objects (e.g., aircrafts, trucks, ships, buildings) and generalizable across other satellite imagery datasets. We have adapted a floating-point operation (FLOP) framework [50] to measure the model’s computational complexity (G-FLOPs) and establish its correlation with time-to-predict performance. Latency reduction in real-life practical experiments was tested by adapting two leading-edge computational architectures, modern GPU and TPU.

The research contributions of this paper are summarised in the following list:

- It proposes a U-net architecture optimised for accurate and accelerated object recognition in multispectral satellite imagery suitable for alternative data-driven algorithmic trading;
- It provides and applies a modified G-FLOPS method for measuring network’s architectural complexity;
- It introduces “Pixel frame selection and sequencing” methods for reduction of contextual noise in training and improved accuracy in prediction;
- It presents a well annotated satellite imagery dataset for public use and further development in this research field.

The rest of this paper is organized as follows. In Sect. 2 we review the specifics of multispectral satellite imagery and provide an overview of related work in the field. Section 3 is dedicated to a detailed presentation and analysis of proposed approaches that enhanced the artificial neural network to state-of-the-art (SOTA) level performance. In Sect. 4 we describe and interpret the experimental findings. Section 5 concludes the paper.

2 Related works

Poor pixel resolution, rich multispectral aperture, and a wide aspect ratio are the unique properties of optical satellite imagery [44, 33]. Objects such as light-vehicles are depicted in a relatively small 15 × 8 pixel matrix in contrast to millions of pixels offered by the ImageNet. To capture these rich multispectral properties given the resolution constraints, we deploy semantic segmentation [2] for light-vehicle recognition problem. It outputs the semantically interpretable category of each pixel [21], and it is more precise compared to object detection and scene interpretation [1]. Semantic image segmentation techniques originated from recursive thresholding method [7], spatial constrained k-means approach [35], histogram-based image segmentation, non-parametric clustering, entropic thresholding and edge detection techniques [27]. These methods are manually calibrated [35, [27], consequently lacking generalization and scalability [28].

The vehicle recognition problem has received a lot of research attention [37], and Convolutional Neural Networks (CNNs) were popularised [30] since they do not require prior feature extraction [28, 63]. A CNN processes data in the form of multiple arrays [29], and therefore, multiband satellite imagery dataset is well suited for it by design [36]. Considerable amounts of research papers have been published on the implementation of semantic segmentation using various CNN architectures [37]. Nguyen et al. [43] presented the five-layer CNN and achieved high object recognition accuracy of 91% for large urban area objects [3]. Later, Chen et al. [6] developed a Hybrid Deep Convolutional Networks (HDCN) architecture for light-vehicle objects and claimed best performance at a time [6] and significantly surpassed other Hybrid CNN structures such as Hierarchical Robust CNN (HRCNN) using AlexNet as a backbone [63]. Building on that work Yu et al. proposed the convolutional capsule network that delivered a 93%, state-of-the-art (SOTA) results for vehicle recognition [60].

Ferdous et al. [12] introduced prediction speed criteria in 2019 and argued that Regions-CNN (RCNN) [17], Fast-RCNN [16], and Faster-RCNN [47] are incompatible for real-time applications due to slow multistage regional-proposal based approach. End-to-end detection-based methods like You Only Look Once (YOLO) [46] and Single Shot Detectors (SSD) [34], where suggested to increase prediction speed, yet compromising on accuracy (only 89.21%) [38]. An alternative fully convolutional neural network was proposed by Shelhamer et al. [51] that combined features from complementary resolution levels (contextual and spatial information). The FCN architecture has demonstrated the best precision using semantic segmentation [51] and also improved parameter optimization and gradient flow, as discussed by Estrada et al. [11].

Ronneberger et al. developed FCN called U-net for solving high-level feature extraction in biomedical image segmentation [48] that won a competition at Symposium for Biomedical Imaging [32]. Biomedical images share similar dimensionality, resolution, and perspective properties with satellite imagery, and it was later realised that the overweighting model's higher-level feature extraction (i.e., the contours of the object) improves prediction accuracy in both of dataset types [48]. Subsequently, U-net was adapted to satellite imagery by Iglovikov et al. [25] and won the 3rd place in Kaggle competition achieving the highest Jaccard coefficient confirming U-net suitability for this problem [14]. In addition to feature extraction, network's ability to extract spatial information was researched by Yuan et al. [61] where they discuss benefits of convolution and deconvolutions similar to U-net structure. They also introduce a light network structure MobileNet that suggested ideas of light network infrastructures [61]. In this paper, we significantly enhance U-net architecture and propose techniques enabling U-net deployment in real-time applications.

3 Proposed approaches

We propose advancements to the process of U-net design, hyperparameters tuning, training, and complexity optimisation to enhance the prediction accuracy and speed. The entire process from satellite imagery acquisition (P1) to end-signal generation and delivery to algorithmic trading system (P13) is depicted in Fig. 2. Components from P5 to P10 coloured in blue represent areas of advancements proposed in this article and are described in the following subsections: (1) Network depth construction and feature extraction; (2) Computational complexity analysis; (3) Pixel frame sequencing.

3.1 Network depth and feature extraction

We propose four distinctive architectures to derive an optimal network configuration for solving a prediction speed vs. accuracy problem (Fig. 2, component P5 and P6). To originate these proposed configurations, we have conducted quantitative experiments (see Sect. 4) and visual examination (see Fig. 3 and Fig. 4) [25, 48].

Each proposed U-net model consists of an even number of layers plus a single fully-connected layer with Sigmoid

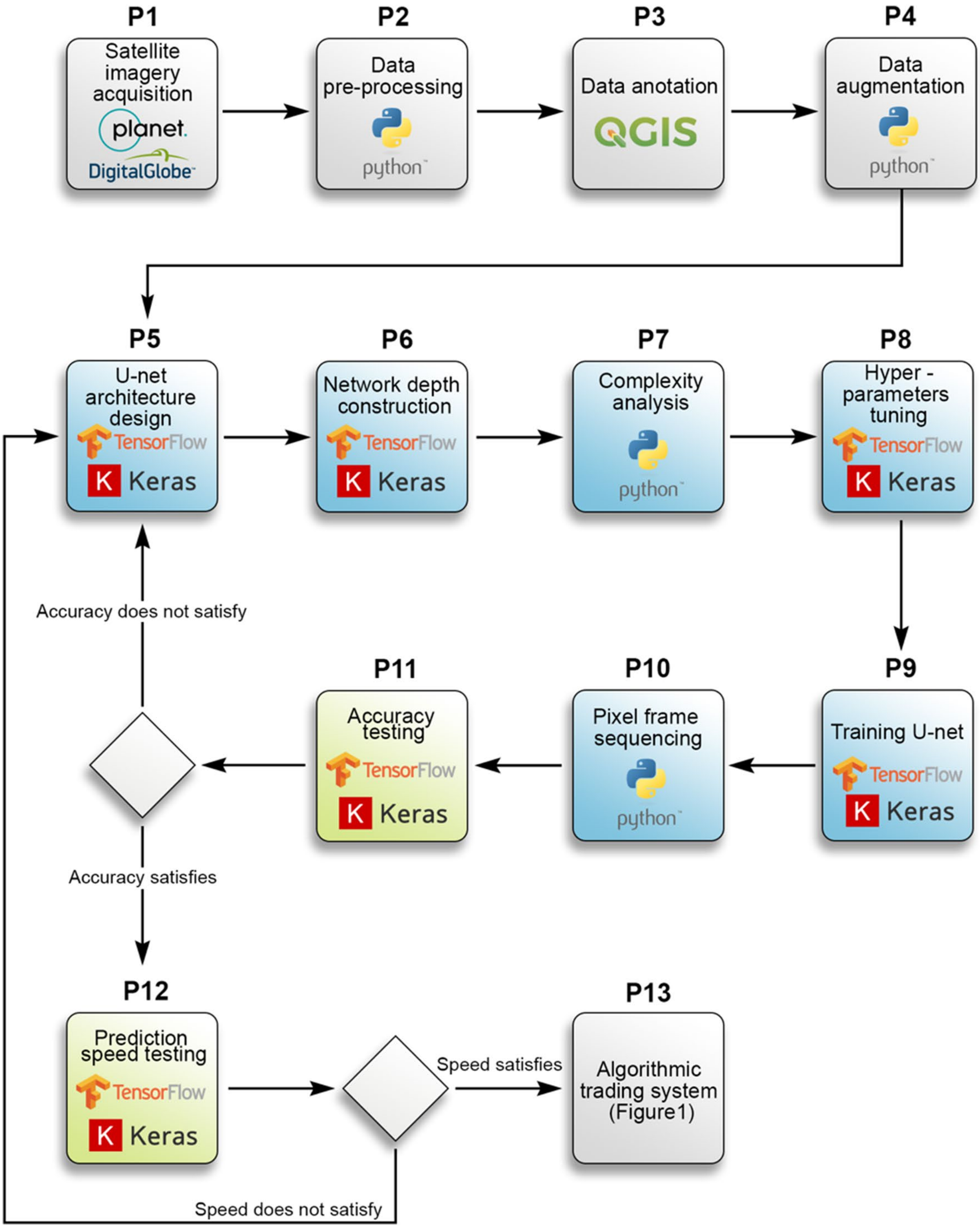


Fig. 2 Schematic workflow diagram for object recognition in satellite imagery

activation function generating per-pixel semantic segmentation as an output (Fig. 3). Models were initiated at fifteen convolutions and sequentially (in four groups) increased by six layers (three in the encoder and three in the decoder part) to a total of thirty-nine layers:

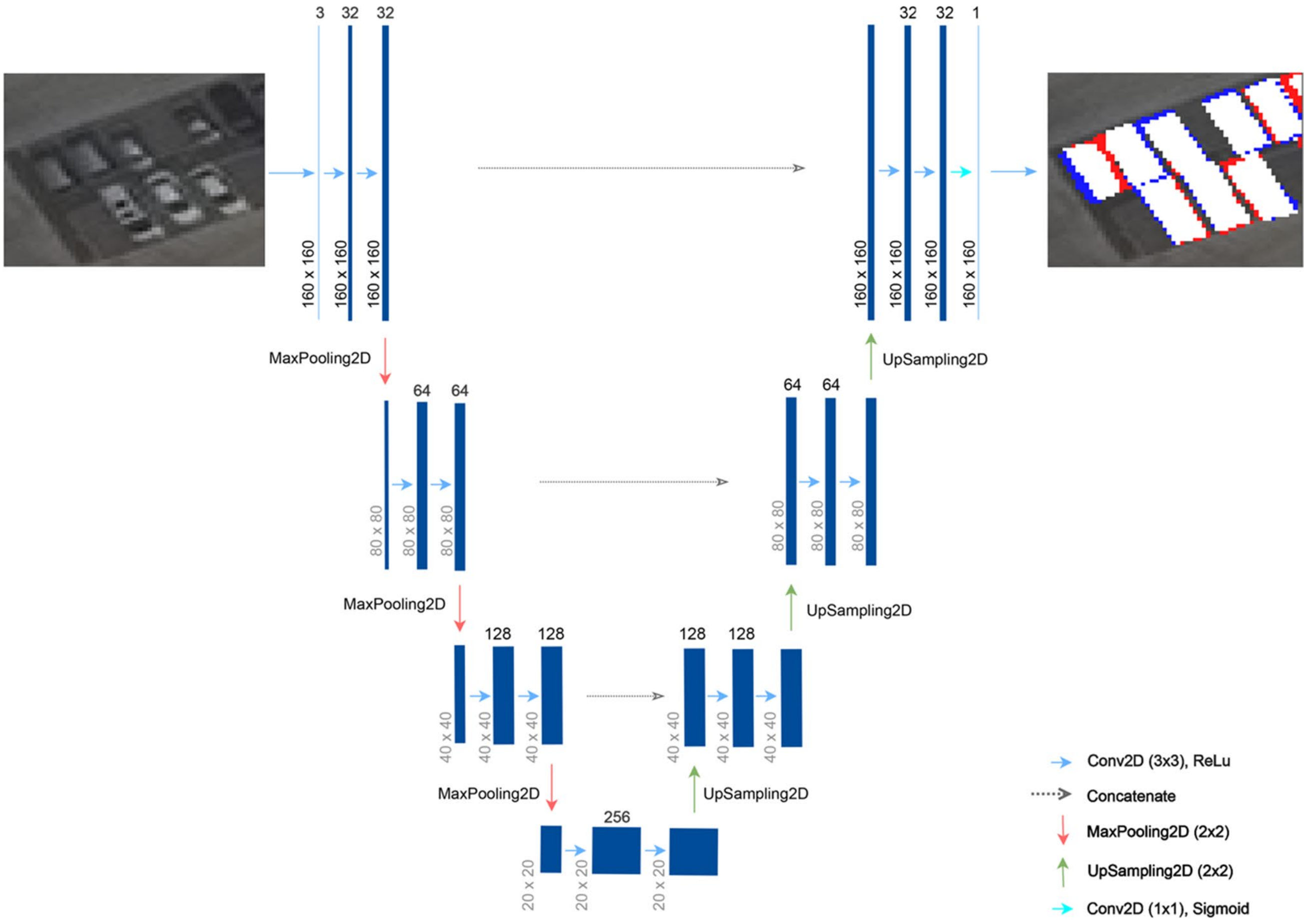


Fig. 3 *U-net_Model_1* design. Input image (left) and an output image (right). Blue colour pixels represent “light-vehicle” object class recognised by the U-net, red colour represents the original annotator marked object contours, white colour represents accurate per-pixel prediction

- *U-net_Model_1*: 21 layers in total (15 conv2d);
- *U-net_Model_2*: 27 layers in total (19 conv2d);
- *U-net_Model_3*: 33 layers in total (23 conv2d);
- *U-net_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 the up-sampling and contains 2 × 2 convolution (up-convolution) that halves the number of feature channels [11]. On the back of these up-sampling operations, we recover the spatial information and enable precise localization 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 scheme was deployed to avoid overfitting [32] and provided a computationally cheap way to regularize the neural network [52] that increased the learning speed.

Given that there are no empirical methods to investigate how effective a network is in performing feature extraction, we deploy a deconvolution-based *Lucid* visualization technique [41]. We compare the feature maps from the last layer of convolution operation of four U-net architectures (Fig. 4) [65]. In general, detectable objects in satellite imagery overall have specific contours (e.g., light-vehicle, truck, ship, or plane), which are consistent due to perspective invariance of the camera. Specific of our particular dataset is detailed in Sect. 4 and in our dataset directory [18]. *Lucid* visualization method allows us to investigate how well the network performed high-level-feature extraction task, i.e., recognizing the contours of the object, which is an important prediction accuracy driver [22].

Based on visual examination we can suggest that contours of the object (i.e., high-level features) are more defined as the network depth increases. We can see a gradual improvement

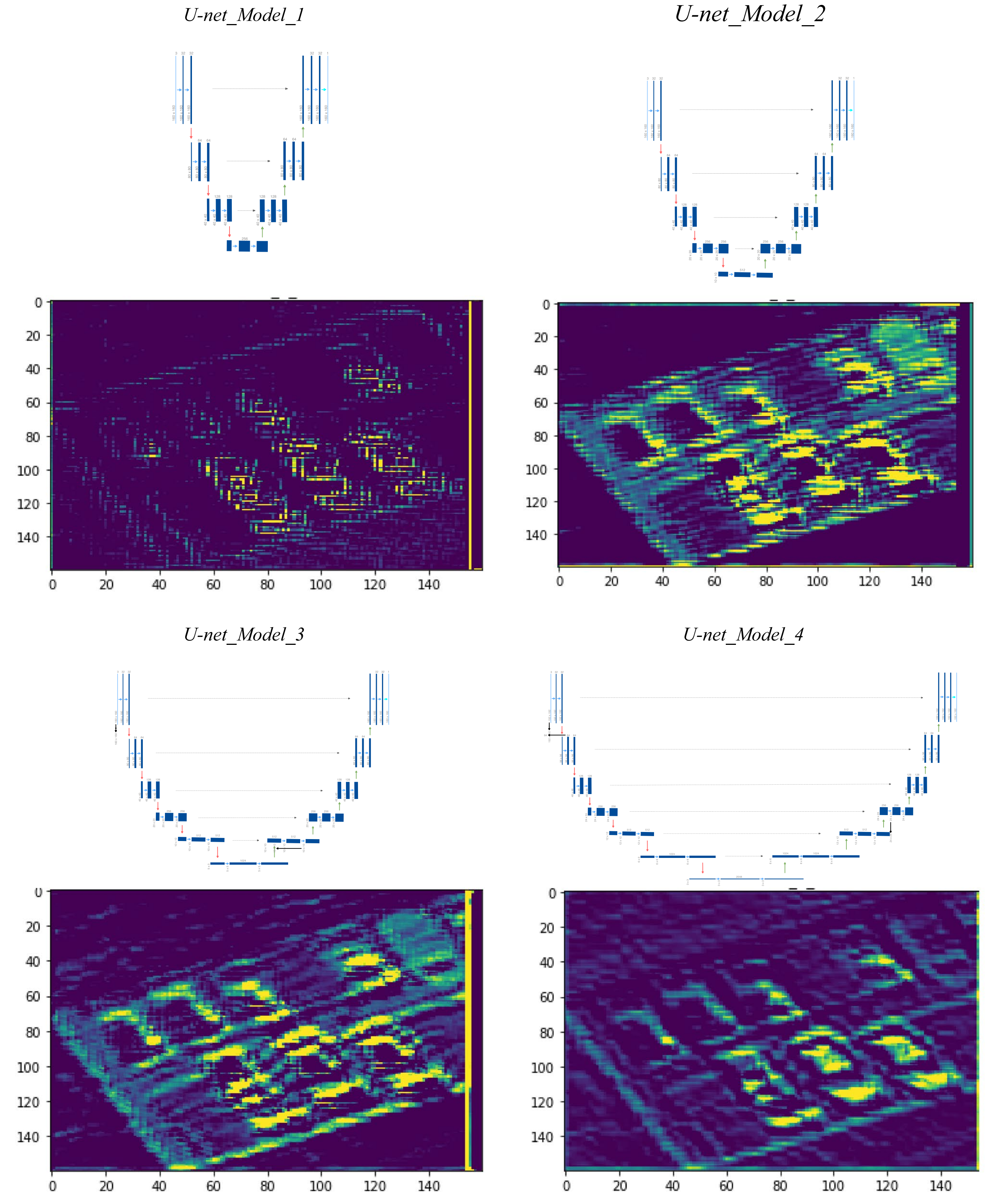


Fig. 4 Feature extraction capabilities with different U-net depth

in feature extraction at each step with the equal depth differential (six convolutional layers) between architectures. *Lucid* visual experimentation method can be utilized in other neural networks performance testing since it allows to compare performance “inside” the network.

3.2 Computational complexity

Significant signal latency from satellite imagery is caused by slow object recognition models (as illustrated in Fig. 1) because complex models take time to “scan”, detect and recognise objects in large land Area of Interest (AOI) (e.g., 10 km^2 at a time). Object recognition speed is a factor of computational complexity and power of computing [62]. 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 (Fig. 2, component P7). 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 [24]. However, to our knowledge, there is no conventional benchmark that sets to define the computational complexity of the neural network [26]. Researches show that the number of operations in a network model can effectively estimate inference time [5]. The number of FLOPs represents how computationally expensive a model is [50]. We customize the FLOPs approach suggested by Sehgal et al. [50] to calculate the computational complexity of a neural network as defined in Eq. (1):

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

Model complexity (G-FLOPs) is a sum of FLOPs for every layer, where E—number of conv2D layers, D_e—number of output dimensions, A_{ed}—size of 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—number of filter dimensions of layer b, P_{bx}—size of x dimension in layer b, Z_b—number of output dimensions of layer b, O_{bz}—size of dimension z in layer b. The *Conv2D* layer count of floating-point operations is dependent on layer parameters count and layer output size [62]. The *MaxPool* layer count of floating-point operations is dependent on filter area size 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*

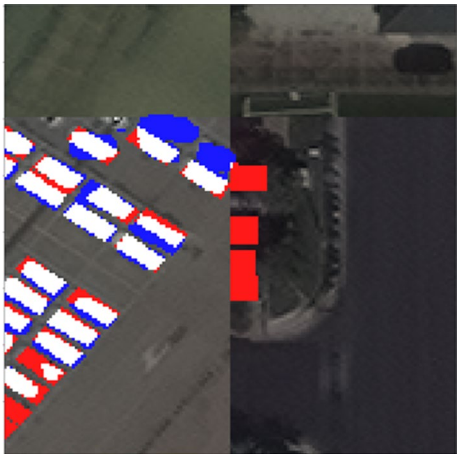


Fig. 5 Four unrelated scenes artificially combined in one frame. When used for training, it captures the partially-cut objects and scene shifts as ground-truth. By doing so it distorts the object-specific and contextual information and generates noise reducing the overall accuracy of the training set. Blue colour pixels represent “light-vehicle” object class recognised by the U-net, red colour represents the original annotator marked object contours, white colour represents accurate per-pixel prediction

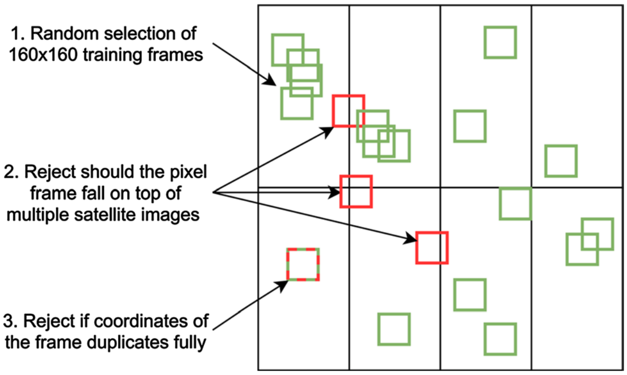

1. Random selection of 160x160 training frames
2. Reject should the pixel frame fall on top of multiple satellite images
3. Reject if coordinates of the frame duplicates fully

Fig. 6 Pixel frame selection approach for network training. Green colour depicts valid pixel frames; red colour represents rejected pixel frames that were excluded from training

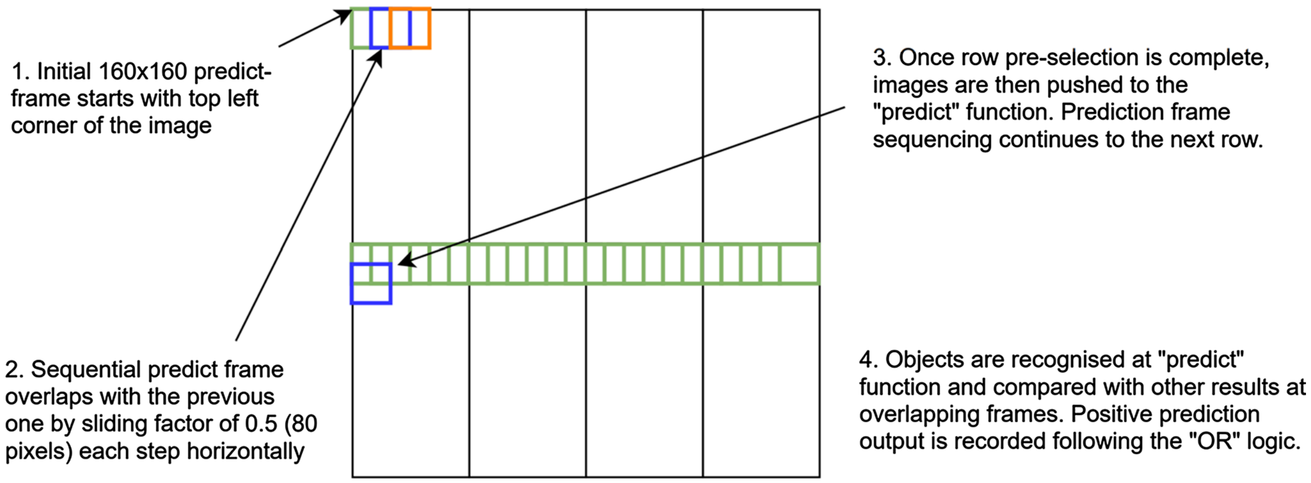

1. Initial 160x160 predict-frame starts with top left corner of the image
2. Sequential predict frame overlaps with the previous one by sliding factor of 0.5 (80 pixels) each step horizontally
3. Once row pre-selection is complete, images are then pushed to the "predict" function. Prediction frame sequencing continues to the next row.
4. Objects are recognised at "predict" function and compared with other results at overlapping frames. Positive prediction output is recorded following the "OR" logic.

Fig. 7 Prediction frame sequencing method

only reads the data from memory and writes to the certain position in the output using indices and other pixels are filled by 0. Indices array always has the same shape as the input. Concatenation is just a memory copy; hence no floating-point operation will be conducted [9]. Depending on the concatenation axis, index calculation might be needed, but our approach ignores such operations. This calculation allows us to examine the relationship between computational cost of the network, prediction accuracy and prediction speed, all further examined in a Sect. 4.

3.3 Pixel frame sequencing

Due to practical GPU/TPU memory limitations, training a neural network using pixel frame size equivalent to full raw satellite image would cap the training batch size to minimum and prevent the network from training effectively [25]. Thus, satellite images with large AOI's are segmented into frames and then consolidated into smaller pixel frame mosaics for training. Smaller pixel frames allow larger training batches as well as context variability in each backpropagation cycle. The drawback of this approach, however, is that on frame edges, it collates mixed landscapes and cropped objects, consequently generating noise that distorts the contextual information in the training set (Fig. 5).

To prevent the above drawback, we developed a programmatic conditional approach called "pixel frame selection" that feeds the U-net (Fig. 2, component P9). It is an improvement from the method proposed by Chen et al. [6]. Via this approach, the network is trained on selected pixel frames (small cropped images) that allows intersections for better augmentation, yet prevents duplications. It follows three rules as described in Fig. 6: (1) selection of 160 × 160 training frames at random; (2) rejection if the 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, allowing improved training accuracy and, consequently, prediction precision.

In addition to solution for training, we introduce a technique called "prediction frame sequencing" for improved prediction (Fig. 2, component P10). It essentially allows the neural network to broaden the contextualization of the object it is classifying. Object classification is done given at least two different backgrounds (prediction frames). In an event of classification mismatch, the object is considered as positively recognized (i.e., "OR" function) as illustrated in four steps at Fig. 7.

To assess the impact of this technique, we trained and tested two identical neural networks on identical datasets. The first network utilized 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 3.57% higher object recognition accuracy rate.

4 Experimental investigation and results

In order to derive the most optimal U-net architecture for real-time applications we have conducted hundreds of experiments with network configuration, complexity and hyperparameters. The training set used in these experiments was created from an open-source raw satellite imagery database *SpaceNet* with high-resolution imagery taken by *DigitalGlobe WorldView-3* satellite. A total of 250 (125 augmented) high resolution (30 cm per pixel) multispectral satellite images, equivalent to 50 km^2 AOI of Paris, Shanghai, Las Vegas, and Khartoum were used for training/validation (80%) and testing (20%). From the 8-band spectrum, Coastal (400–452 nm) to near-infrared (NIR 866–954 nm) a 4-band RGB + P (450–630 nm) bands was applied. In order to expose the training to the desired invariance and ensure model is robust, the following data augmentation was implemented: random brightness (30% of images in the training set with random brightness), rotation (10%), perspective distortion (10%) and random noise addition (30%). Local contrast normalization and pan-sharpening were applied; however, it did not generate significant improvement in accuracy.

A total of 350 h of professional annotation work has been conducted to prepare a high-quality training set with 80,316 labelled objects. Images were annotated using *QGIS* geospatial imagery software. Labelling and polygon coordinate generation has been manually completed by multiple professional annotators and quality cross-checked. There are no publicly available high-resolution satellite imagery datasets with labelled light-vehicle object class. We are publishing our in-house developed, proprietary dataset with labelled polygons online to enable further development in this research field [18].

Stochastic gradient descent (SGD) was implemented via *Keras* [20] as well as momentum optimization algorithm—Adam [49]. As suggested by Ronneberger et al. [48], to minimize the overhead and make maximum use of the GPU and TPU memory, we favour a large input pixel frame over a large batch size and therefore experimented training batch sizes ranging from 32 to 192. 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 64 GB (1 core) and *TPU v3-8 128 GB* (8 cores) were deployed on our *GCP* system.

Table 1 Prediction accuracy results on the test set

	Accuracy (TPO) %	Overprediction (FPO) %	G-Flops	Jaccard coefficient
U-net_Model_1	95.33	12.01	5.3218	0.6402
U-net_Model_2	97.67	17.83	6.9832	0.6162
U-net_Model_3	97.01	26.45	8.6443	0.5573
U-net_Model_4	96.70	16.60	10.3053	0.6226

Table 2 Impact of activation function on prediction results on *U-net_Model_2*

Activation	Accuracy (TPO) %	Overprediction (FPO) %	TPO/FPO	Jaccard coefficient
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

4.1 Results: accuracy

To quantitatively evaluate vehicle recognition results, the following metrics were adopted: True Positive Objects (TPO) and False Positive Objects (FPO), and Jaccard coefficient. TPO reflects proportion (in %) of objects ("light-vehicles") correctly detected as compared to the "ground truth". FPO measures overprediction error i.e., objects

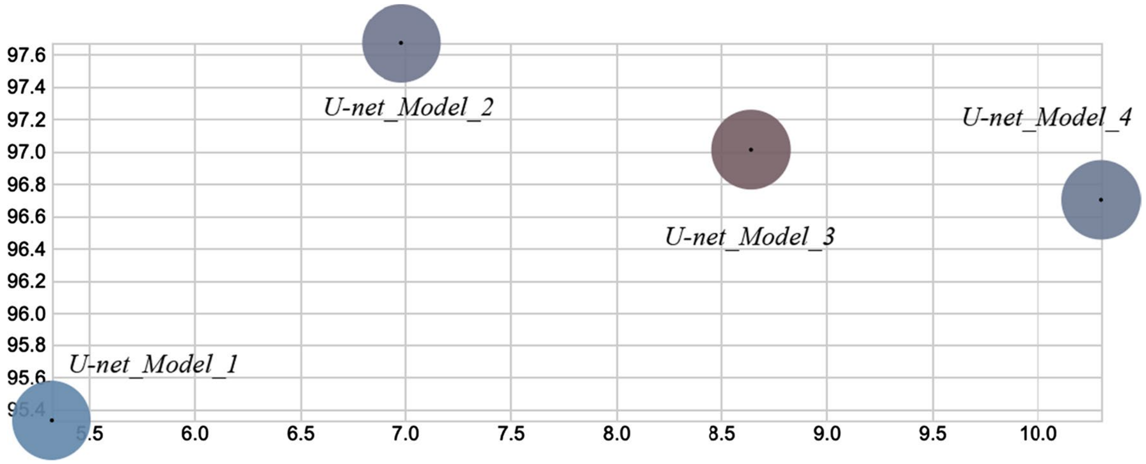
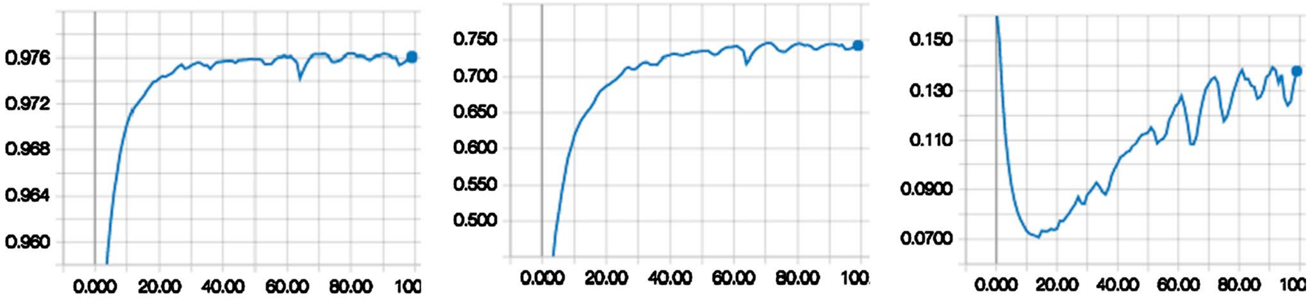


Fig. 8 Comparison of performance results between U-net models (x-axis: computational complexity (G-Flops); y-axis: prediction accuracy (TPO's); red colour: highest and blue: lowest overprediction (FPO's)



(a) *validation accuracy* **(b)** *validation Jaccard coefficient* **(c)** *validation loss*

Fig. 9 Training of *U-net_Model_2* (x-axis: number of epochs)

Table 3 Quantitative evaluation of different leading methods

	Dataset	Proposed method	Y. Yu's method [60]	H. Zhou's method [64]	L. Wan's method [57]
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
E = Epochs K = Iterations	V and O	$E=40$	$E=2000$	$K=3000$	$K=3000$

labelled by the network, not by the annotator. TPO/FPO ratio gives an indication of networks' performance vs. the noise it generates. Jaccard coefficient (see Eq. (2)) is a pixel-level classification accuracy metric of segmentation masks, particularly useful for calibration of network training process [54]:

$$\text{Jaccard coefficient}_c = \frac{TP_c}{TP_c + FP_c + FN_c}. \tag{2}$$

where TP_c is the number of "True positive pixels" in a class c across the entire data set; FP_c is the number of "False Positives pixels" in c; FN_c—"False Negatives" in c. U-net model experimentation results of are provided in Table 1.

We were able to achieve the state-of-the-art object recognition accuracy of 97.67% with *U-net_Model_2*. This network also maintained an FPO level 17.83%, and a 0.6162 Jaccard coefficient. A close second best, *U-net_Model_3* has, however, provided a significant overprediction (FPO = 26.45%) rate. G-Flops metric indicates the computational complexity and *U-net_Model_2* represents relatively light computational complexity with 6.9832 allowing faster prediction. Figure 8 compares the accuracy performance between models as well as their complexity below:

In order to continue enhancing the *U-net_Model_2* performance, we have experimented with the following activation functions (see Table 2) expecting an increase in accuracy and quality.

Table 4 TPU vs GPU prediction speed for U-net_Model_2

Type	Frame size	Batch size	Jaccard coefficient	Time-to-Predict (10 k patches/s)
TPU-v8	128 × 128	128	0.64	20.45
TPU-v8	160 × 160	128	0.64	36.42
TPU-v8	192 × 192	128	0.63	41.49
GPU-p100	128 × 128	128	0.64	6.94
GPU-p100	160 × 160	128	0.65	12.37
GPU-p100	192 × 192	128	0.65	20.45

Rectified Linear Unit activation (ReLU) has provided the best accuracy (TPO) results for U-net [22]. However, activation function that generates the lowest level of noise (FPO = 6.09) is a Hyperbolic Tangent (Tanh) activation function still providing > 90% accuracy and, simultaneously a high TPO/FPO ratio (14.92).

To optimize the network training time, we monitored *U-net_Model_2* with the various number of epochs (20–100). Training completeness was measured using three metrics, as illustrated in Fig. 9. U-net reached the peak validation accuracy at epoch 35–40 and starts to overfit. Validation loss curve (c) confirms the overfit by reaching minimal at 15 and rapidly increasing beyond 35 as well as Jaccard coefficient plateaus beyond epoch 40. The variability of the optimal range has not changed after experimenting with the

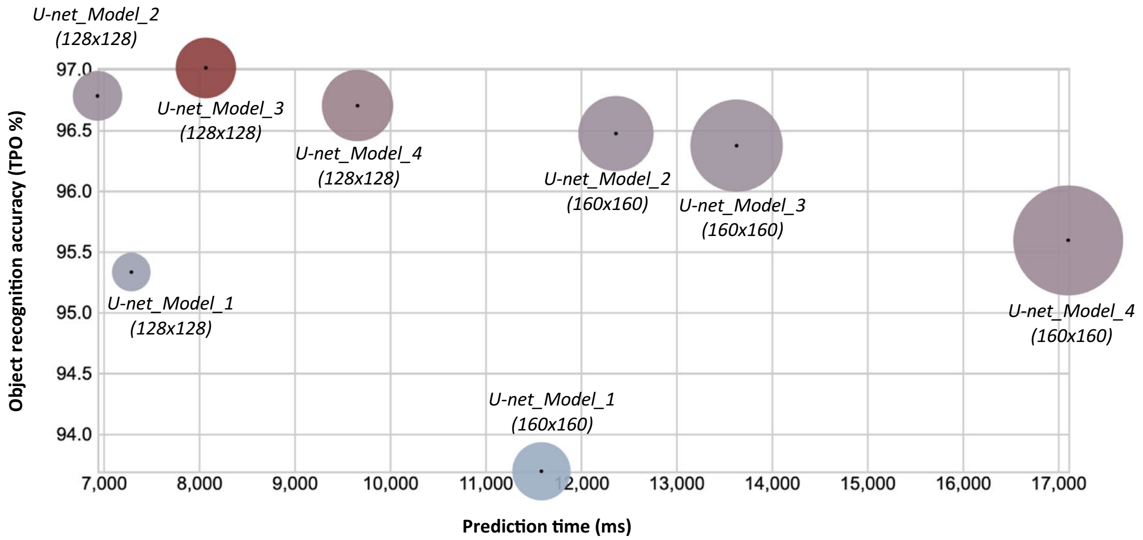


Fig. 10 Object recognition accuracy vs prediction speed vs computational complexity. X-axis: prediction speed (in milliseconds), Y-axis: accuracy (TPO); size of the circle: computational complexity in G-Flops; colour scale: red colour indicates highest overprediction error (FPO) and blue the lowest

other U-net models. Understanding an optimal epoch range (35–40 epochs) minimises computational expense and retraining time and therefore is useful in applications where models need to be recalibrated (retrained) on a frequent basis such as algorithmic trading.

To benchmark our proposed approach performance, we have compared the performance with the latest leading object recognition methods using external datasets. Table 3 provides quantitative evaluations of the performance of our and other competing methods on two high-resolution remote sensing image data sets: OIRDS [53] and VEDAI [45]. Performance metrics of completeness (CPT) and correctness (CRT) were adopted from the competing articles to ensure consistency and are calculated as $PT = \frac{TP}{TP+FN}$, and $CRT = \frac{TP}{TP+FP}$, where TP is True Positives, FN is False Negatives, and FP is False Positives.

Our proposed method 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 *U-net_Model_2* architecture was forty (40) epochs as compared to Yu's [60] of two thousand (2000) epochs resulting in a significantly lower computational cost. Zhou's and Wan's methods have used $K=3000$ algorithm iterations. K iterations is the closest comparable metric 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.

4.2 Results: prediction speed and computational complexity

We have conducted experiments utilizing *U-net_Model_2* for time-to-predict on two computational architectures GPU and TPU to compare its performance in practise. GPU has generated faster prediction speed results (see Table 4).

One of the reasons why TPU might have performed slower at the prediction task is that TPU-v8 is designed for larger complexity computations and longer operations as 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" [60]. *U-net_Model_2* architecture works better on GPU due to its light complexity (6.98 G-Flops). Therefore, we have selected GPU as preferred computational engine for U-net's rapid object recognition in real-time applications.

Total of eight U-net configurations with two different pixel frame parameters (128 × 128 and 160 × 160) were examined on 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 10 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 the identical computational machine. Furthermore, larger input frame size increases computational expense (G-Flops) in the network, slowing down the prediction and not increasing accuracy in return. We can see that the fastest CNN network is *U-net_Model_2* (128 × 128) that 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.

5 Conclusions

In this paper, we propose an accurate and high-speed U-net architecture that is able to conduct a semantic segmentation operation for object recognition in multispectral satellite imagery. We also publish our proprietary training and testing dataset with 350 h of professional annotation work in order to encourage further scientific research in this field. Different from the traditional methods using hand-crafted features, we introduce three generally applicable approaches to enhance neural networks' ability to extract high-level features, measure network complexity, fine-tune training process and increase prediction speed. On the back of the suggested approaches and experimental investigation, our developed U-net architecture exceeded the human-level performance with 97.67% accuracy for a "light-vehicle" object class over multiple sensors. It has also outperformed other known methods to date and was able to generalize across dispersed scenery. Additionally, *U-net_Model_2* computationally light architecture delivered a fivefold improvement in training time and a rapid prediction, essential for real-time applications. The proposed neural network and process enhancement techniques could be readily applied to other latency-sensitive industrial and humanitarian applications. This research topic will become increasingly important with the growth of the near-real-time satellite imagery coverage and a number of cross-disciplinary applications of remote sensing.

6 Declarations

We confirm that this manuscript has not been previously published and is not currently under consideration by any other journal. This research has been self-funded and with no competing interests to be disclosed. All of the authors have provided a significant contribution to this manuscript and approved the contents of this paper. All authors have agreed to the *Machine Vision and Applications* submission policies. Full access to the training dataset is made available via GitHub directory [18].

Acknowledgements This research was supported by a Grant (No. S-MIP-21-53) from the Research Council of Lithuania.

References

1. Angelova, A., Shenghuo, Z.: *Efficient Object Detection and Segmentation for Fine-Grained Recognition* (2013).
2. Audebert, N., Saux, B., Lefevre, S.: Segment-before-detect: vehicle detection and classification through semantic segmentation of aerial images. Rem. Sens. **9**, 368 (2017)
3. Ball, J., Anderson, D., Chan, C.S.: A comprehensive survey of deep learning in remote sensing: theories, tools and challenges for the community. J. Appl. Rem. Sens. **4**(11), 042609 (2017)
4. Belward, A.S., Skøien, J.O.: Who launched what, when and why; trends in global land-cover observation capacity from civilian earth observation satellites. J. Photogr. Rem. Sens. **103**, 115–128 (2015)
5. Canziani, A., Paszke, A., Culurciello, E.: An analysis of deep neural network models for practical applications (2016). Preprint at https://arxiv.org/abs/1605.07678.
6. Chen, X., Xiang, S., Liu, C.-L., Pan, C.-H.: Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geosci. Remote Sens. Lett. **11**(10), 1797–1801 (2014)
7. Cheriet, M., Said, J., Suen, C.: A recursive thresholding technique for image segmentation. IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc. **7**, 918–921 (1998)
8. Cliff, D., Brown, D., Treleaven, P.: 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).
9. Cong, J., Xiao, B.: Minimizing computation in convolutional neural networks. 281–290 (2014).
10. Di Baldassarre, G., Schumann, G., Bates, P.: Near real time satellite imagery to support and verify timely flood modelling. Hydrol. Process. Int. J. **23**(5), 799–803 (2009)
11. Estrada, S., Conjeti, S., Ahmad, M., Navab, N., Reuter, M.: Competition vs. concatenation in skip connections of fully convolutional networks. In: *International Workshop on Machine Learning in Medical Imaging*, 214–222 (2018).
12. Ferdous, S.N., Moktari, M., Nasser, N.: Super resolution-assisted deep aerial vehicle detection. Artif. Intell. Mach. Learn. Multi-Domain Oper. Appl. **11006**, 17 (2019)
13. Franco, D., Kourogiorgas, C., Marchese, M., Panagopoulos, A., Patrone, F.: Small satellite and CubeSats: survey of structures, architectures, and protocols. Int. J. Satell. Commun. Network. **4**(37), 343–359 (2019)
14. Ghosh, S., Das, N., Das, I., Maulik, U.: Understanding deep learning techniques for image segmentation. ACM Computing Surveys (CSUR) **4**(52), 1–35 (2019)
15. Gillespie, T.W., Chu, J., Frankenberg, E., Thomas, D.: Assessment and prediction of natural hazards from satellite imagery. Prog. Phys. Geogr. **31**(5), 459–470 (2007)
16. Girshick, R.: Fast R-CNN. 1440–1448 (2015).
17. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. 580–587 (2014).
18. Gudžius, P., Kurasova, O., Darulis, V., Filatovas, E.: *VUDataScience*(2020). GitHub Depository: https://github.com/VUDataScience/Deep-learning-based-object-recognition-in-multispectral-satellite-imagery-for-low-latency-applicatio
19. Guirado, E., Tabik, S., Rivas, M.L., Alcaraz-Segura, D., Herrera, F.: Whale counting in satellite and aerial images with deep learning. Nat. Sci. Rep. **9**(1), 1–12 (2019)
20. Gulli, A., Pal, S.: *Deep Learning with Keras*. Packt Publishing (2017).
21. Gupta S.: Learning Rich Features from RGB-D Images for Object Detection and Segmentation. *Computer Vision (ECCV), 8695* (2017).

22. Han, J., Zhang, D., Cheng, G., Guo, L., Ren, J.: Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. IEEE Trans. Geosci. Remote Sens. **6**(53), 3325–3337 (2014)
23. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. 1026–1034 (2015).
24. Hunger, R.: *Floating Point Operations in Matrix-Vector Calculus*. Floating Point Operations in Matrix-Vector Calculus: Technical Report (2007).
25. Iglovikov, V., Mushinskiy, S., Osin, V.: Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition. https://arxiv.org/abs/1706.06169 (2017).
26. Justus, D., Brennan, J., Bonner, S., McGough, A.S.: Predicting the computational cost of deep learning models (2018).
27. Krstinic, D., Kuzmanic Skelin, A., Slapnicar, I.: Fast two-step histogram-based image segmentation. Image Process. **5**, 63–72 (2011)
28. Längkvist, M., Kiselev, A., Alirezaie, M., Loutf, A.: Classification and segmentation of satellite orthoimagery using convolutional neural networks. Rem. Sens. **4**(8), 329 (2016)
29. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
30. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **11**(86), 2278–2324 (1998)
31. Letizia, A., Marino, S., Ahmad, U., Alvino, A.: Investigating the global socio-economic benefits of satellite industry and remote sensing applications. IBIMA Publishing **10**(3), 475–480 (2019)
32. Li, H., Chen, J., Lu, H., Chi, Z.: CNN for saliency detection with low-level feature integration. Neurocomputing **226**, 212–220 (2017)
33. Li, X., Yuan, Y., Wang, Q.: Hyperspectral and Multispectral Image Fusion via Nonlocal Low-Rank Tensor Approximation and Sparse Representation. IEEE Trans. Geosci. Remote Sens. **59**(1), 550–562 (2021)
34. Liu, W.: SSD: Single shot multibox detector. 21–37 (2016).
35. Luo, M., Ma, Y.F., Zhang, H.-J.: A spatial constrained K-means approach to image segmentation. **2**, 738–742 (2004)
36. Ma, L., al., e. : Deep learning in remote sensing applications: A meta-analysis and review. ISPRS J. Photogramm. Remote. Sens. **152**, 166–177 (2019)
37. Manana, M., Tu, C., Owolaw, P.A.: A Survey on Vehicle Detection Based on Convolution Neural Networks (2017).
38. Mansour, A., Hassan, A., Hussein, W., Said, E.: Automated vehicle detection in satellite images using deep learning. 610 (2019).
39. Marcum, R.A., Davis, C.H., Scott, G.J., Nivin, T.W.: Rapid broad area search and detection of Chinese surface-to-air missile sites using deep convolutional neural networks. J. Appl. Rem. Sens. **11**(4), 1 (2017)
40. Monk, A., Prins, M., Rook, D.: Rethinking alternative data in institutional investment. J. Financ. Data Sci. **1**(1), 14–31 (2019)
41. Mott, D., & Tomsett, R.: Illuminated decision trees with lucid. https://arxiv.org/abs/1909.05644 (2019).
42. Musa, Z.N., Popescu, I.: A review of applications of satellite SAR, optical, altimetry and DEM data for surface water modelling, mapping and parameter estimation. Hydrol. Earth Syst. Sci. **9**(19), 3755–3769 (2015)
43. Nguyen, T., Han, J., Park, D.-C.: Satellite image classification using convolutional learning (2013).
44. Ose, K., Corpetti, T., Demagistri, L.: Multispectral satellite image processing. Opt. Rem. Sens. Land Surface 57–124 (2016).
45. Razakarivony, S., Jurie, F.: Vehicle detection in aerial imagery: A small target detection benchmark. J. Vis. Commun. Image Represent. **34**, 187–203 (2016)
46. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. 779–788 (2016).
47. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015).
48. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation (2015).
49. Ruder, S.: An overview of gradient descent optimization algorithms (2016). https://arxiv.org/abs/1609.04747.
50. Sehgal, A., Kehtarnavaz, N.: Guidelines and benchmarks for deployment of deep learning models on smartphones as real-time applications. Mach. Learn. Knowledge Extract. **1**(1), 450–465 (2019)
51. Shelhamer, E., Long, J., Darrel, T.: Fully convolutional networks for semantic segmentation. IEEE Ann. Hist. Comput. **39**, 640–651 (2017)
52. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
53. Tanner, F.: Overhead imagery research data set—an annotated data library & tools to aid in the development of computer vision algorithms. IEEE Appl. Imagery Pattern Recognit. Workshop, pp. 1–8.
54. van Beers, F., Lindström, A., Okafor, E., Wiering, M.A.: Deep Neural Networks with Intersection over Union Loss for Binary Image Segmentation (2019).
55. Van Etten, A. (2018). You only look twice: Rapid multi-scale object detection in satellite imagery. Preprint at arXiv: http://arxiv.org/abs/1805.09512.
56. Voigt, S.: Global trends in satellite-based emergency mapping. Science **353**(6296), 247–252 (2016)
57. Wan, L., Zheng, L., Huo, H., Fang, T.: Affine invariant description and large-margin dimensionality reduction for target detection in optical remote sensing images. IEEE Geosci. Remote Sens. Lett. **14**(7), 1116–1120 (2017)
58. Wang, Y. E., Wei, G.-Y., Brooks, D.: Benchmarking TPU, GPU, and CPU platforms for deep learning (2019). https://arxiv.org/abs/1907.10701.
59. Yingyan, L., Sakr, C., Kim, Y., Shanbhag, N.: PredictiveNet: An energy-efficient convolutional neural network via zero prediction (2017).
60. Yu, Y., Gu, T., Guan, H., Li, D., Jin, S.: Vehicle detection from high-resolution remote sensing imagery using convolutional capsule networks. IEEE Geosci. Remote Sens. Lett. **16**(12), 1894–1898 (2019)
61. Yuan, Y., Zhitong, X., Qi, W.: VSSA-NET: vertical spatial sequence attention network for traffic sign detection. IEEE Trans. Image Process. **28**(7), 3423–3434 (2019)
62. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. 6848–6856 (2018).
63. Zhang, Y., Yuan, Y., Yachuang, F., Xiaoqiang, L.: Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection. IEEE Trans. Geosci. Remote Sens. **57**(8), 5535–5548 (2019)
64. Zhou, H., Wei, L., Lim, C.P., Nahavandi, S.: Robust vehicle detection in aerial images using bag-of-words and orientation aware scanning. IEEE Trans. Geosci. Remote Sens. **12**(56), 7074–7085 (2018)
65. Zintgraf, L.M., Cohen, T.S., Adel, T.W.: Visualising deep neural network decisions (2017).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Povilas Gudžius is a Ph.D. candidate in Computer Science (4th year) at the Institute of Data Science and Digital Technologies, Vilnius University. His research interests include object recognition, signal processing, and neural networks with a particular focus on machine learning technology applications in the financial services industry. Povilas is also a Quant Data and Machine Learning technology specialist at the financial technology firm, Bloomberg LP, based out of New York, USA.

Olga Kurasova received a Ph.D. degree in computer science from the Institute of Mathematics and Informatics, Vytautas Magnus University, Lithuania, in 2005. She is currently employed as a Principal Researcher and a Professor at the Institute of Data Science and Digital Technologies, Vilnius University. Her research interests include data mining methods, optimization theory and applications, artificial intelligence, neural networks, visualization of multidimensional data, multiple criteria decision support, parallel computing, and image processing. She is the author of more than 70 scientific publications.

Vytenis Darulis received a Master's degree in information systems engineering from the Kaunas University of Technology. He is currently employed as Chief Technical Officer (CTO) at the DataMe company. He has more than 10 years of experience in the software development and information security industries. His research interests include artificial intelligence, neural networks, and data visualization.

Ernestas Filatovas received the Ph.D. degree in informatics engineering from Vilnius University, Lithuania, in 2012. He was a Postdoctoral Researcher with Vilnius University, where he is currently a Senior Researcher and a Co-founder of the Blockchain Technologies Group, Institute of Data Science and Digital Technologies. His research interests include distributed ledger technologies, machine learning, image processing, multiobjective and global optimization, high-performance computing, and the development and application of various operation research techniques.