INFORMATICA, 2022 Vol. 0, No. 0, 1–0

Neural Architecture Search for object recognition in satellite imagery

Povilas GUŽIUS, Olga KURASOVA, Vytenis DARULIS, Ernestas FILATOVAS

Institute of Data Science and Digital Technologies, Vilnius University
Akademijos str. 4, LT-08412 Vilnius, Lithuania
E-mail: povilas.gudzius@mif.vu.lt; olga.kurasova@mif.vu.lt, vytenisd@gmail.com, ernestas.filatovas@mif.vu.lt

Abstract. Advancements in optical satellite hardware and lowered costs for satellite launches raised a high demand for geospatial intelligence. Object recognition problem in multi-spectral satellite imagery carries training and test dataset properties unique to this problem. Perspective distortion, resolution variability, data spectrality, and other features make it difficult for a specific hand-designed neural network to perform well on a dispersed type of scenery, ranging data quality, and different objects. UNET, MACU, and other architectures manually designed deliver high-performance results for accuracy and prediction speed in large objects. However, once trained on different datasets, the performance dropped and required manual re-calibration or further cellular level testing to adjust the neural network architecture. To solve these issues, Neural Architecture Search (NAS) techniques having an automatic problem-tailored architecture with state-of-the-art accuracy performance can be applied.

In this paper, firstly, we have conducted detailed testing on the top four performing neural networks for object recognition in satellite imagery to compare their performance: FastFCN, DeepLabv3, UNET, and MACU. Then we applied and further developed a neural architecture search technique for the best performing manually designed MACU by optimizing a search space at the cellular level. Our developed AutoML process generated NAS-MACU neural network that produced better performance, especially in a low-information intensity environment. We can state that application of the Neural Architecture Search procedure has the capability to be applied across various datasets and object recognition problems within the remote sensing research field.

Key words: Neural Architecture Search, AutoML, Convolutional Neural Networks, MACU, Satellite Imagery, Object Recognition, Semantic Segmentation.

1. Introduction Commercial satellite constellations from Maxar technologies like RADARSAT-2 [1], Pleiades-1 and ICESat-2 [2], Vision-1 from Airbus Defence and Space [3], and Cartosat-3 by IRSO [4] 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 [5]. With the increase in resolution, a vast amount of new use cases emerge when object recognition and Machine Learning (ML) techniques are applied to solve real-world problems. Economic and ecological

intelligence is generated by processing very high-resolution remote sensing images of the earth's surface and even under the surface. Those cases include deforestation [6], classification of the crop field [7], water body detection on the urban surface [8], resource identification, marine logistics, military and defence, agriculture, manufacturing, urban planning, biodiversity extinction [9], [10].
Dispersity of these use cases requires problem-specific ML techniques that perform well at a given type of dataset, resolution, sensor type, object class, and other typological parameters [11]. Multiple Convolutional Neural Network (CNN) architectures are being developed to address those use cases. Top performing CNNs in object recognition include DeepLabv3, FastFCN, UNET, and MACU [12]. However, these networks are normally 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 [13]. The process of developing CNN architectures, including the ones mentioned above, and parameter experimentation can take months and, in some cases, years to reach the required result [14]. The research and architecture design process are time-consuming and labor-intensive [15].

In addition to limitations of human researcher capabilities and the dispersity of task-specific topologies, yet 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 in the orbit, most cost constraints, and limited public datasets availability [16].

Current state-of-the-art (SOTA) neural network architectures are manually built and include theoretically pre-specified 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. This brings great difficulty when building a high-quality machine learning system in practice and therefore limits ML applications. Automated Machine Learning (AutoML) is a perspective solution part of the meta-learning group that allows building those systems without deep human expert knowledge and months of research [17]. Neural architecture search (NAS) as part of AutoML is a technique to automate the design of neural networks, and it aims to find the best architecture for specific problems. NAS essentially aims to do the work of a human manually working/tuning a neural network faster and more effectively. Therefore, an automatic NAS has become an active research topic in recent years [18]. Specifically, NAS represents a technique for automating the design of artificial neural networks instead of conventional hand-designed ones [19] and recently has obtained gratifying progress [20], [21]. NAS cell-level search space has been looked into for various broader architecture types, including NAS-UNET [22].

Building a high-quality machine learning system, especially in a domain-specific area requires human expertise and therefore limits ML applications. Automated Machine Learning (AutoML) is a perspective solution part of the meta-learning group that allows building those systems human expert knowledge and months of research [23]. Neural architecture search (NAS) as part of AutoML is a technique to automate the design of neural networks and it aims to find the best architecture for specific problems. NAS essentially aims to do the work of a human manually working/tuning a neural network faster and more effectively.

In this paper, we deal with the object recognition problem in satellite imagery. The underlying object recognition technology that we leverage is semantic image segmentation. We solve a semantic segmentation problem to measure the pixel level performance, and then we use the object pixel overlay to detect one object class of "light vehicle" is satellite imagery. For this purpose, we apply the NAS technique as part of the AutoML that could be used across multiple use cases and have auto-calibration features that allow to custom-cater for the problem at hand. Firstly, we have conducted detailed testing on the top four performing neural networks for object recognition in satellite imagery to compare their performance: FastFCN, DeepLabv3, UNET, and MACU. Then, we applied and further developed the NAS for auto-customized best-performing MACU network focused on optimizing a search space at the cellular level. We have produced an optimized and automatically-generated NAS-MACU neural network that was able to generate better accuracy performance.

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

- It provides implementation and in-depth analysis of the top 4 best-performing CNNs: DeepLabv3, FastFCN, UNET, and MACU on a standard satellite imagery dataset.
- It proposes an effective NAS implementation for network meta-learning, which includes a Search space, Search strategy, and a performance measurement strategy.
- It introduces a NAS-MACU network that is capable of self-discover the best-performing MACU cell topology and architecture optimized for accurate and accelerated object recognition in multispectral satellite imagery.
- It presents NAS-MACU performance in four different information-intensity environments and confirms that NAS-MACU is more suited when the training data (e.g., satellite imagery) is limited availability for practical real-world and economic reasons.
- Finally, it includes a well-annotated and updated satellite imagery dataset for public use and further development in this research field.

The rest of this paper is organized as follows. In Section 2, we provide an overview of works in the field related to semantic segmentation in satellite imagery

and neural architecture search. Section 3 is dedicated to a detailed presentation and analysis of the proposed approaches of NAS process design and visual representation to achieve results. There we also describe an important aspect of the NAS-MACU meta-learning algorithm – cloud configuration, computational resources, and other practical NAS implementation-related aspects. In Section 4, we describe and interpret the experimental findings. Section 5 concludes the paper.

2. Related works Image segmentation, like image classification and object detection, is one of the important research areas in the computer vision community. Object detection 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 different types, called semantic segmentation and instance segmentation. Semantic segmentation can be considered as 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 [24].

2.1. CNN networks for image segmentation Today, the best solutions to the segmentation problem are obtained with deep learning-based solutions compared to the classical 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 [25] is based on Fully Connected Networks (FCN). The general classification architecture with CNN consists of convolutional and pooling layers to extract features with lower dimensions. In the last layers of these type of networks, fully connected layers are used to make a final decision. On the other hand, in FCN, fully connected 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 out. There are developed different types of FCN-based architectures in the literature [26], [27]. The proposed FCN architectures use pre-trained classification models such as VGG [28] and ResNet [29] in the feature extraction stage.

Considering that the segmentation of remote sensing images is an important issue, it is seen that segmentation studies are widely carried out in this field as well. FCNs are applied in satellite images, and promising results are obtained in different studies [30]. On the other hand, the main issue of FCN is that the resolution of feature outputs is down-sampled with several convolutional and pooling layers. To eliminate this issue, FCN variants [30] add skip connection from earlier layers to enhance the output for scale changes and perform well in remote sensing images. Various more

advanced FCN-based approaches such as SegNet [31], UNET [32], DeepLab [33] have also been proposed to address this issue.

The architecture named Deeplabv1 [34] applies a Fully Connected Conditional Random Field (FCRF) to enhance the poor localization property of deep networks. Thus, it is more sufficient to localize segment boundaries compared to the previous methods. Deeplabv2 [35] architecture applies atrous convolution (also named dilated convolution) for up-sampling and atrous spatial pyramid pooling (ASPP) to robustly segment objects at multiple scales. ASPP is actually a different variant of Spatial Pyramid Pooling (SPP) proposed in the study [36] and aims to improve the accuracy for different object scales. Deeplabv3 [37] augments the ASPP module with image-level features encoding global context and further boost performance. It improves over previous DeepLab architecture versions, and it still achieves comparable performance with other state-of-art architectures.

Segmentation networks such as UNET and SegNet roughly consist of two stages: encoding and decoding stages. SegNet and UNET 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 [28]. 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 up-sampling. However, UNET transfers the entire feature maps from encoding layers to the decoding layers, and 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, but it also shows good performance for satellite images segmentation [32]. Different UNET-based architectures are proposed in the literature, such as UNET++ [38] and UNET variants like Inception-UNET [39]. Inception variants of UNET apply Inception [40] approach in different ways and enhance the feature extraction stages, while INCSA-UNET uses DropBlock [41] and spatial attention modules [41] to prevent overfitting and enhance important features by focusing on key areas, respectively. MACU [42] is another UNET-based architecture using multiscale skip connections and asymmetric convolution blocks.

The skip connection used in UNET and its variants acts as a bridge between low-level and high-level features. This approach and multi-scale feature extraction make significant performance improvement in the segmentation task. On the other hand, the use of attention modules with an encoding-decoding structure has been widely used for fine-resolution image segmentation. Spatial and channel attention mechanisms perform well in different architectures like MACU, SENet [43] and DANet [43]. A multi-scale UNET study [44] proposes an architecture to merge the low-level and abstract features extracted from the shallow and deep layers. It aims to retain the detailed edge information for building segmentation issue. The MACU

architecture proposed multi-scale skip connections with channel attention blocks and asymmetric convolution blocks in the UNET backbone. The experiments on remote sensing datasets have shown the effectiveness of MACU-NET. In the coordinate attention (CA) mechanism [45], which is a newer approach, the spatial and channel information is effectively captured by embedding positional information into channel attention. FCAU-NET [46] 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. Experimental results on two remote sensing image datasets show that MACU-NET outperforms state-of-the-art architectures like FCAU-NET, PSPNet [47], and TransUNET [48], yet produces similar performance to DeepLabv3 and FastFCN. A summary of the networks and their release year can be reviewed in Table 1.

2.2. Neural architecture search Preeminently performing neural network architectures are currently designed by scholars and practitioners. An effective neural network architecture design often requires substantial knowledge in the particular domain and lengthy manual trialing [49]. The process of network component experimentation can take months and, in some cases, years to reach the required result [23], [50]. Researchers encounter limitations such as the design process being time-consuming and labor-intensive. NAS as part of the AutoML aims to solve this problem and make the process of purpose-built neural network design accessible to wide range of domains and a larger quantity of researchers. NAS aims 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 sing an auxiliary search algorithm to design the characteristics of a neural network. These characteristics are inside of the neural network, such as activation functions, hyperparameters or a cell-level architecture itself.

A NAS search space is used to find the best architecture, while a performance estimation method is used to score the performance of a network. Various search algorithms such as reinforcement learning (RL) [51], evolutionary algorithm (EA) [52], Bayesian optimization method [53], and gradient-based method [54] have been used. At 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 [55]. 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 methods apply

evolutionary computation to solve the NAS issue. A detailed review of EA-based NAS works is given in the paper [13].

Table 1. Breakdown of manually designed neural networks for semantic segmentation

Architectures	Year	Unique approached deployed
UNET	2015	Skip connections from down sampling layers to up-sampling
DeepLabV1	2016	Use fully connected Conditional Random Field (CRF)
SegNet	2017	In skip connection, SegNet transfers only pooling indices to use less memory
PSPNet	2017	Dilated convolutions and pyramid pooling module
DANet	2017	Position and channel attention modules followed by ResNet feature extraction
UNET++	2018	Improved skip connections from down sampling layers to up-sampling
DeepLabV2	2019	Use atrous/dilated convolution and fully connected CRF together
MACU-NET	2019	Has multi-scale skip connections and asymmetric convolution blocks.
NAS-UNET	2020	Search a cell-space for best architecture
DeepLabV3	2021	Improved atrous spatial pyramid pooling (ASPP)
Inception-UNET	2021	Uses Inception modules instead of standard kernels (wider networks)
TransUNET	2021	Transformers encode the image patches in the encoding stage
FastFCN	2021	Fully connected network layers
INCSA-UNET	2021	Uses Dropblock inside Inception modules, and also apply attention between encoding and decoding stages
FCAU-NET	2022	Coordinate attentions, asymmetric convolution blocks to enhance the extracted features and refinement fusion block (RFB) in skip connections

NAS research has been used more in image classification problems so far [56]. Several papers have proposed search space for encoding-decoding-based architectures similar to UNET for medical image segmentation issue. NAS-UNET [57] searches space to select primitive operation sets within cells by using differentiable architecture search (DARTS) [58], while C2FNAS [59] searches for the best topology followed by the best convolution size within cells by using topology-similarity based evolutionary algorithm. In the paper [60] authors firstly 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 is dependent on the selected network types in one respect. Considering that different network types can give better results in different problems, it can also cause a disadvantage depending on the problem. It is called Mixed-Block NAS (MB-NAS), and topology level search is followed by cell level search in this method. It uses a search algorithm, called local search [61].

DARTS uses an efficient strategy over a continuous domain by gradient descent. However, its performance often drops due to overfitting in the search phase. To avoid this, NAS-HRIS [62], GPAS [63] and Auto-RSISC [64], which are based on a gradient descent framework, have been proposed for remote sensing scene classification issue. NAS-HRIS uses the Gumbel-Max trick [65] to improve the efficiency of searching. It is evaluated for remote sensing image segmentation problem, and outperforms the methods in the literature. GPAS applies a greedy and progressive search strategy for a higher correlation between search and evaluation stages. The auto-RSISC algorithm aims to reduce the redundancy in the search space by sampling the architecture in a certain proportion. Thus, Auto-RSISC requires less computational resources but it limits the performance of the model by reducing the architecture diversity. RS-DARTS [66] adds noise to suppress 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 reaches a state-of-the-art performance in remote sensing scene classification while reducing computational overload in the search phase. In our research, we capture the recommendations made for effective semantic segmentation task [57] and develop a NAS-MACU search methodology as an effective NAS for remote sensing.

3. Evaluation of top-performing manually designed CNNs In order to develop NAS for a certain type of network, we have recreated and implemented the top-performing convolutional neural networks to date (as discussed in Subsection 2.1) and conducted thorough experimentation of their performance in semantic segmentation task and object recognition on the satellite imagery dataset [32] [67]. We have conducted the experimentation of these networks under four different information-intensity environments to test their sensitivity to the quantity of the training data. Information intensity is a term used to identify the quantity and completion of the training data that is used for the supervised learning of the network. Training environment that is sufficient for a network to be fully trained is considered a high-information intensity environment, and low-information intensity environment is during conditions where network training data and training-related hyperparameters contain at least one of the following constraints, such as the quantity of images <30k (sized 160x160 pixel), batch size <8 and epochs <30. As per the literature review, the top-performing networks selected for comparable research were MACU, FastFCN, UNET, and DeepLabv3.

3.1. Considered satellite imagery dataset The training set used in these experiments was created from an open-source raw satellite imagery database SpaceNet with high-resolution imagery taken by the DigitalGlobe WorldView-3 satellite [67]. A total of 250 (125 augmented) high-resolution (30 cm per pixel)

multi-spectral satellite images, equivalent to 50 km2 Area Of Interest (AOI) of Paris, Shanghai, Las Vegas, and Khartoum, were used for training and validation (80% of total). 20% of this dataset was used for testing. In order to expose the training to the desired invariance and ensure the model is robust, the following data augmentation was implemented: random brightness (30% of images in the training dataset with random brightness), rotation (10%), perspective distortion (10%) and random noise addition (30%).

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 [68]. Thus, satellite images with large AOIs are segmented into frames and then consolidated into smaller pixel frame mosaics/images (160x160 pixels) for training and validation [69]. Smaller pixel frames/images allow larger training batches as well as a wider context variability in each backpropagation cycle [67].

3.2 Experimental investigation and results of CNN We have adapted the networks to the Google Cloud Platform (GCP) architecture that was used for experimental investigation to be compatible with the satellite imagery dataset. We have used these three information intensity environments (as mentioned in Subsection 3.0 to test and compare top-performing networks. We have recorded individual performance on the following five metrics: Recall, Precision, Overprediction Error (FPO), overall accuracy (F_1) and per-pixel accuracy (Jaccard Index).

Computational environment To derive the most optimal NAS-MACU architecture for applications, we have conducted experiments with network configuration, complexity and hyperparameters. Experiments were executed on the custom-built Google Cloud Platform (GCP) architecture with specifically developed for our research problem, and GPU NVIDIA Tesla P100 64 GB (1 core) was deployed on the system.

Estimtion metrics To quantitatively evaluate vehicle recognition results, the following metrics were adopted: True Positive objects (TP), False Positive objects (FP), Jaccard coefficient, True Positive Rate (TPR), Positive Predictive Value (PPV), and F_1 as the best overall accuracy metric. TP reflects the proportion of objects ("light vehicles") correctly detected as compared to the "ground truth". FP measures overprediction error, i.e., objects labelled by the network, not by the annotator. TP/FP ratio gives an indication of network performance vs. the noise it generates. Jaccard coefficient (see Eq. (1)) is a pixel-level classification accuracy metric of segmentation, particularly useful for the calibration of the network training process:

$$Jaccard\ coefficient_c = \frac{TP_c}{TP_c + FP_c + FN_c}\ ; \tag{1}$$

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.

$$TPR = \frac{TP}{TP+FN}; \quad (2)$$

$$PPV = \frac{TP}{TP+FN}; \quad (3)$$

$$\mathrm{F_1} = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}. \quad (4)$$

Table 2. Performance of four neural networks in three different training environments

Set size	Epochs	Batch sizes	Net	Recall	Precision	FPO	F_1	Jaccard
20000	20	4	MACU	**0.956**	0.942	5.820	**0.949**	**0.659**
			FastFCN	0.955	0.933	6.687	0.944	0.609
			DeepLabV3	0.868	**0.961**	**3.944**	0.912	0.484
			UNET	0.940	0.931	6.928	0.935	0.647
30000	30	4	MACU	0.948	0.945	5.501	**0.946**	**0.661**
			FastFCN	**0.958**	0.926	7.383	0.942	0.615
			DeepLabV3	0.82	**0.968**	**3.156**	0.888	0.441
			UNET	0.955	0.923	7.691	0.939	0.652
30000	30	8	MACU	0.953	0.933	6.675	**0.943**	**0.667**
			FastFCN	0.828	**0.972**	**2.833**	0.894	0.506
			DeepLabV3	0.918	0.950	4.993	0.934	0.538
			UNET	**0.960**	0.919	8.099	0.939	0.658

The performance comaprison of of four investigated neural networks is presented in Table 2. During this experimentation process, 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. UNET, however, provides the best Recall accuracy as is particularly useful in use cases where the objective is to recognize the maximum universe of objects within the given satellite imagery. F_1 score is a better representation of the overall performance of the network, especially when used in assessing the practical application of the network to real-world problems. Precision allows understanding the targeted accuracy of correctly predicted objects.

DeepLabv3 and FastFCN have provided a modest accuracy performance with the lowest quantity of objects, yet it is the most conservative and therefore has the lowest overprediction error (FPO) in two out of free information intensity scenarios. A further visual comparison between the four networks is depicted in Figure 1.

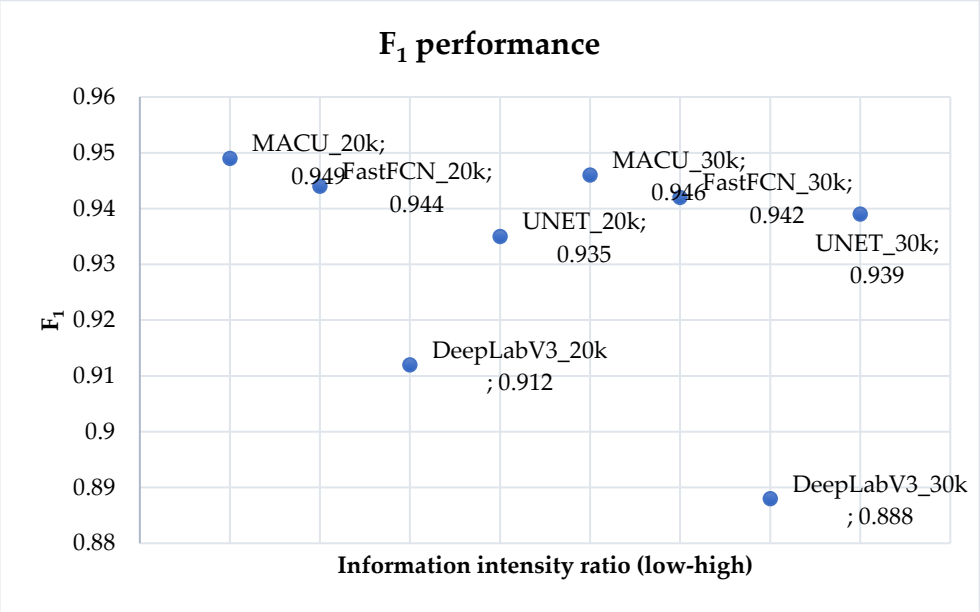

Fig 1. Performance (F_1) comparison in 2 separate information-intensity environments between manually crafted neural nets (*x* axis is the simple sequence and y axis is the overall accuracy (F_1 score)

Finally, we have selected MACU architecture as the state-of-the-art best performing, manually designed architecture for semantic segmentation as the core architecture for our further cell-level NAS research in developing the NAS-MACU.

4. NAS-MACU development process One of the most challenging components in solving real-world problems is to pre-design a rational well-performing deep learning architecture catered for the required task and the type of training data. As previously identified in our empirical research, MACU infrastructure overall is providing promising performance results as compared to others like UNET, FastFCN and DeepLabv3. However, there was no research and/or empirical study done to design and test for NAS-MACU prior to this research paper. In this paper, we design, implement and conduct empirical experimentation on the novel NAS-MACU which has never been accomplished by known research to date.

In order to create, test and deploy NAS-MACU, we have created a reiterative process illustrated in Figure 2.

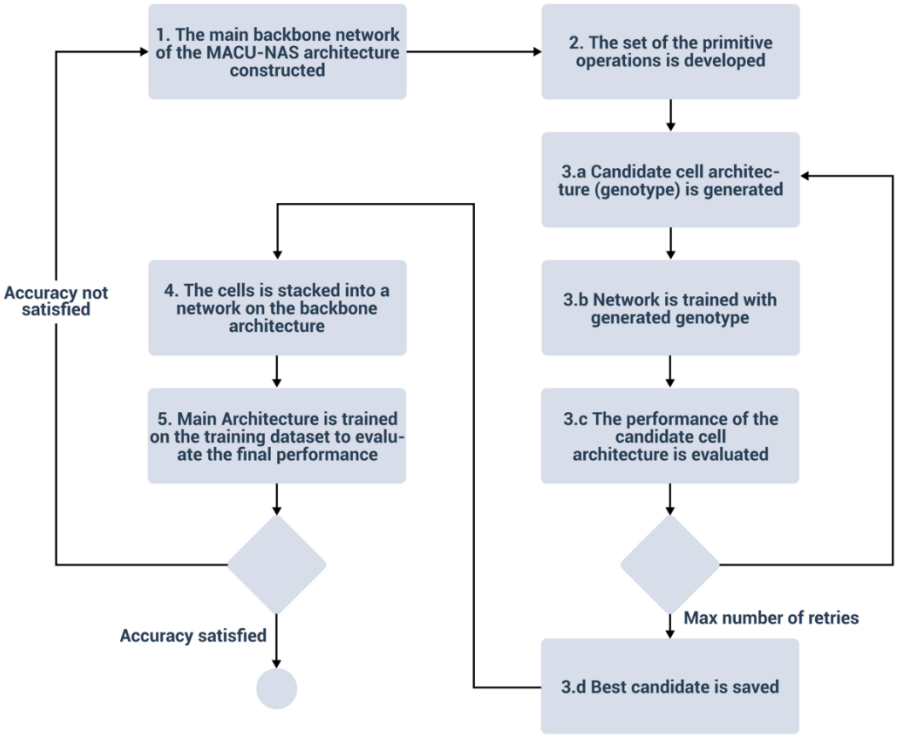

Fig 2. NAS-MACU construction flow

Figure 2 depicts the process used to deliver the state-of-the-art performing, self-topology-designing, NAS-MACU network that adapts to a high dispersity of datasets in a low information environment and without human expertise or manual intervention. Therefore, the NAS-MACU topology follows the iteration cycle until it reaches the pre-defined capacity or is considered that reaches the max performance given the constraints. Those constraints are expressed in the form of hyperparameters and are further discussed in the section below. The research on NAS focuses on three aspects: search space, search strategy, and performance estimation strategy.

NAS calibration The search space defines which architectures can be represented. The search strategy details how to explore the search space. The objective is to find architectures with highly evaluated performance on unseen data. Performance estimation is divided into two parts. First, the performance is evaluated to determine whether to be kept (or expanded) the candidate architecture for the next update. Secondly, we need a deeper network stacked by the cells and evaluate the final performance on a training dataset.

Search Space The primitive operation set was developed on search space to automatically find architecture for DownSC (down-sampling cell) and UpSC (upsampling cells) for semantic image segmentation. Each primitive operation

should have unique properties that cannot be replaced by the others. All convolution operations will limit to 3×3 size, and pooling operation will be 2×2.

Cell genotype generating algorithm In the following sequence of 21 (a – u) events, we describe the high-level logic of the underlying algorithm (see Algortihm 1) defining the cell topology design and iteration process, where E – total epochs and N – total nodes in a cell.

Algorithm 1 Cell genotype generating algorithm

a. Generate a random initial Weights1 and Weights2 values.
b. **for** *e* := 0 to *E*
c. genotype := []
d. **for** *i* := 1 to N
e. Create binary matrix Mask1 and Mask2
f. Assign values to W1 and W2 from Weights1 and Weights2 masked by Mask1 and Mask2
g. edges1 := sorted array of edges from W2
h. L1 := length of edges1 array
i. **for** j := L1
j. k_best := max($W1_{jk}$)
k. gene_items1 array appends ($W1_{j,k\_best}$, operation, edge index j)
l. edges2 := sorted array of edges from W2
m. L2 := length of edges2 array
n. **for** j := L2
o. k_best := max($W2_{jk}$)
p. gene_items2 array appends ($W2_{j,k\_best}$, operation, edge index j)
q. genotype array appended with best item from gene_items
r. model.genotype := genotype
s. if genotype_repeats(genotye) > max_patience:
t. Stop training
u. model.train()

Operations set Following operatinos were considered when searching for the architectures:

- Down operations: Average pooling, max pooling, down cweight, down dilation conv, down depth conv, down conv;
- Up operations: Up cweight, up depth conv, up conv, up dilation conv;
- Normal operations: Identity, cweight, dilation conv, depth conv, conv.

A directed acyclic graph (DAG) in Figure 3 and Figure 4 depicts the process that is used to create the network topology architecture. Every node h_i represents an input image or a feature map, and each edge e_{ij} is related with an operation between nodes h_i and h_j. When the generation method of the DAG is unrestricted, its network architecture space will be very large. Therefore, we use cell-based architecture.

When determining the best cell architecture, then the cells are stacked into a deeper network on the backbone network. The architecture of cell is shared by the entire network.

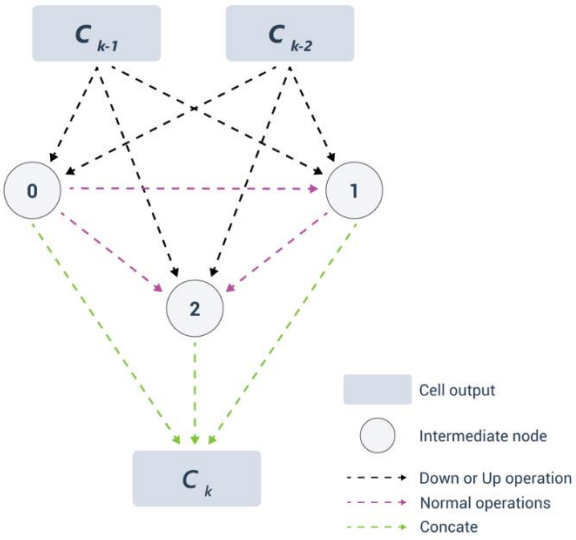

Fig. 3. DAG diagram for cell architecture

An example of the over-parametrized cell architecture is depicted in Figure 3. The blue arrow indicates down or up operation (such as max pooling), the red arrow indicates the normal operation (e.g. identity operation, convolution operation which does no reduce the dimension of feature map), and the green arrow represents a concatenate operation.

Given a cell architecture $C(e_1, \dots, e_E)$ where e_i represents an edge in the DAG. Let $O = o_i$ be one of three types of primitive operation set in the above with N candidate operations. We set each edge to be a mixed operation that has N parallel paths, denoted as MixO. Then cell architecture can be expressed as $C(e_1 = MixO_1, \dots, e_E = MixO_E)$. The output of a mixed operation MixO is defined based on the output of its N paths:

$$\text{MixO}(x) = \sum_{i=1}^{N} w_i o_i(x)(\quad (5)$$

MACU vs NAS-MACU architecture comparison Based on U-Net and asymmetric convolution block, we incorporate multi-scale features generated by different layers of U-Net and interpret a multi-scale skip connected architecture, MACU-NET, for semantic segmentation using high-resolution remote sensing images. This standard MACU design has the following advantages: (1) - The multi-scale skip connections combine and realign semantic features contained both in low-

level and high-level feature maps with different scales; (2) - the asymmetric convolution block strengthens the representational capacity of a standard convolution layer [12].

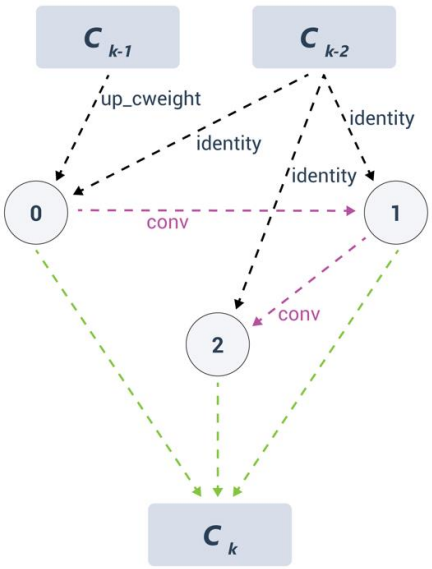

Fig. 4. DAG diagram for the example of the cell architecture searched when the intermediate number is 3

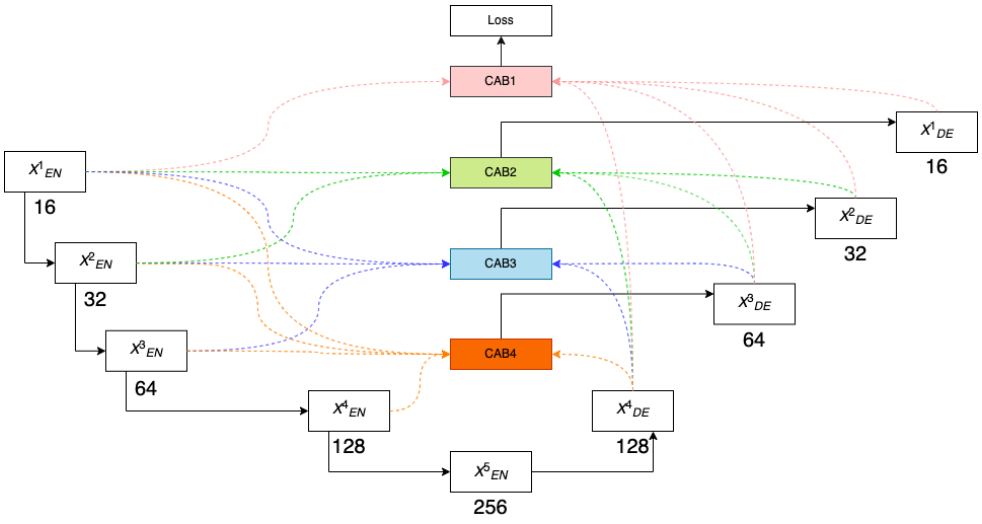

Fig 5. MACU net diagram with multi-scale connectors (adapted from [12])

NAS-MACU leverages the backbone of the MACU network described above, and within the cell structure, we implement the following DownSC (down-sampling cell) and UpSC (up-sampling cell) operations as illustrated in Figure 6.

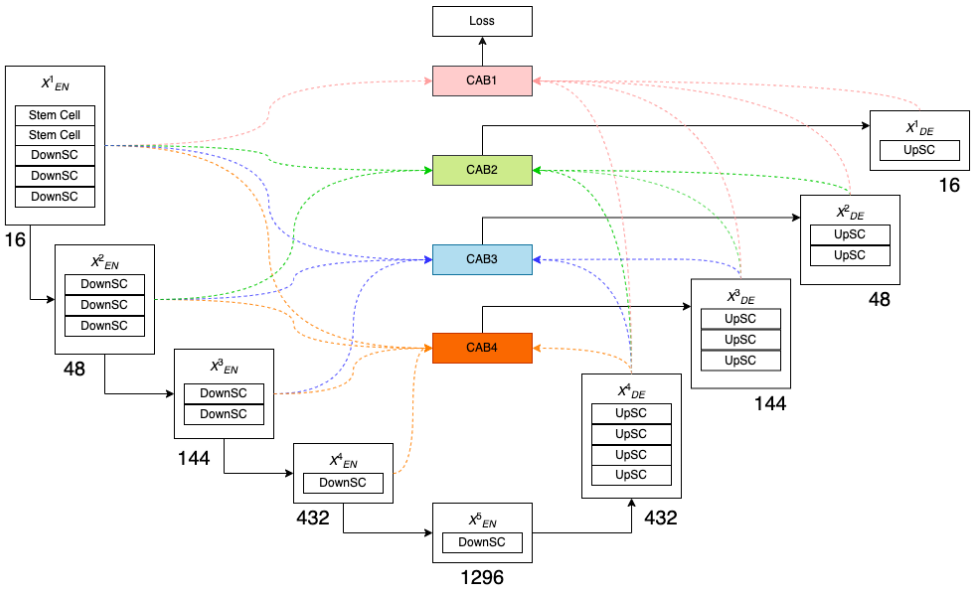

Fig 6. The proposed NAS-MACU architecture and cell-topology at the high level

Cell architectures, search strategy and cell genotypes NAS helps to automatically design two types of cell architectures called DownSC (down-sampling cell) and UpSC (up-sampling cell) based on MACU backbone (Figure 6). Sequence of the search strategy within the construction process is located on Figure 2 diagram as the 3.a block and follows these procedures:

- Initial values assigned to architecture parameters and path weights;
- Path weights transformed to binary values;
- Two paths are sampled, and all the other paths are masked as if they do not exist;
- The path weights and binary gates are reset accordingly;
- The architecture parameters of these two sampled paths updated;
- The path weights are computed by applying softmax to the architecture parameters;
- In each update step, one of the sampled path weight increases and the other sampled path weight decreases while all other paths keep unchanged;
- Once the training of architecture parameters is finished, the compact architecture is derived by pruning redundant paths.

Inside both two cells, the input nodes are defined as the cell outputs in the previous two layers. The architectures of DownSC and UpSC update simultaneously

by a differential architecture strategy during the search stage. An output of each edge is a mixed operation for N candidate primitive operations, which means the output feature maps of all N paths can only be calculated when all operations are loaded into the GPU memory,

In this paper, we use a binary learning and binary path instead of N path. When training network weight parameters, we first freeze the architecture parameters and stochastically sample binary gates for each batch of input data. Then the weights parameters of active paths are updated via standard gradient descent on the training dataset. When training architecture parameters, the weight parameters are frozen, then we reset the binary gates and update the architecture parameters on the validation set. Once the training of architecture parameters is finished, we can then derive the compact architecture by pruning redundant paths.

As an outcome, we have been able to process eight different genotypes of NAS-MACU and test their accuracy rates and performance. We have conducted eight sets of different experiments that took 36 hours of NAS-MACU training on average. We have concluded after best results were reached with NAS-MACU-V7 and NAS-MACU-V8.

Table 4. Comparison of 8 cell architectures (Genotypes)

Genotype Version	(Down Operation, Parent Node Number)	(Up Operations, Parent Node Number)
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)
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)
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)
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)
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)
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)
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)
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)

Graphical representation of a MACU-NAS genotype architectures In order to illustrate the cell level topology generated through the process of cell search described above, we have created Figure 7 and Figure 8 to cover the NAS-MACU cell genotype version NAS-MACU-V7 and version NAS-MACU-V8.

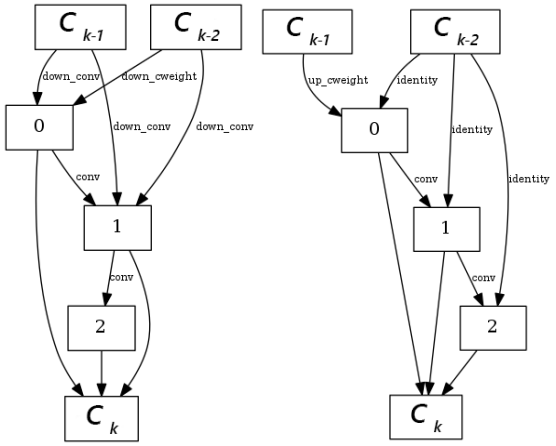

Fig. 7. NAS-MACU-V7 (DownSC (left) and UpSC (right))

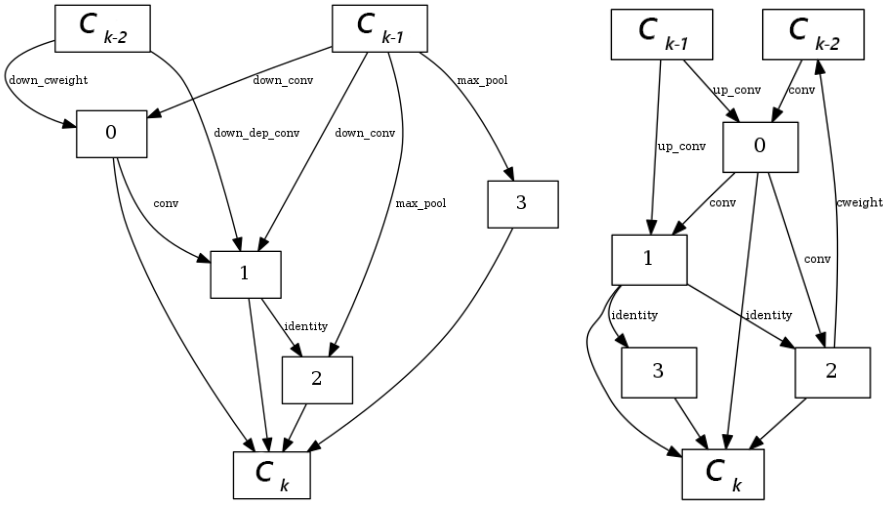

Fig8. NAS-MACU-V8 (DownSC (left) and UpSC(right))

5. Performance evaluation To evaluate the performance of NAS-MACU on the full dataset, eight genotypes were selected and retrieved by different configurations on the architecture searches stage. Results have improved across the spectrum of metrics as we have evolved from NAS-MACU-V1 to NAS-MACU-V8.

Table 5. Accuracy performance between different genotypes

Genotype Version	Precision, PPV, CRT	Recall, TPR, CPT	FPO	F_1	Jaccard
NAS-MACU-V1	0.880	0.949	12.038	0.913	0.598
NAS-MACU-V2	0.893	0.939	10.704	0.915	0.596
NAS-MACU-V3	0.901	0.951	9.865	0.926	0.607
NAS-MACU-V4	0.904	0.945	9.552	0.924	0.608
NAS-MACU-V5	0.824	0.964	17.626	0.889	0.542
NAS-MACU-V6	0.872	0.965	12.835	0.916	0.597
NAS-MACU-V7	**0.924**	**0.957**	**7.616**	0.931	**0.630**
NAS-MACU-V8	0.920	0.953	8.544	**0.934**	0.627

NAS-MACU-V7 and NAS-MACU-V8 showed very similar results. NAS-MACU-V8 achieved the best F_1 score, which combines the precision and recall of a classifier into a single metric by taking their harmonic mean. Also, what's worth mentioning is that the NAS-MACU was able to uptrain itself extremely fast compared to manual networks with low information intensity for training, making it extremely useful in settings where the training set is hard or expensive to acquire (e.g., high resolution satellite imagery). Also, in our experiments, we show that it takes only 15-20 epochs to reach top performance. Figure 9 illustrates that performance.

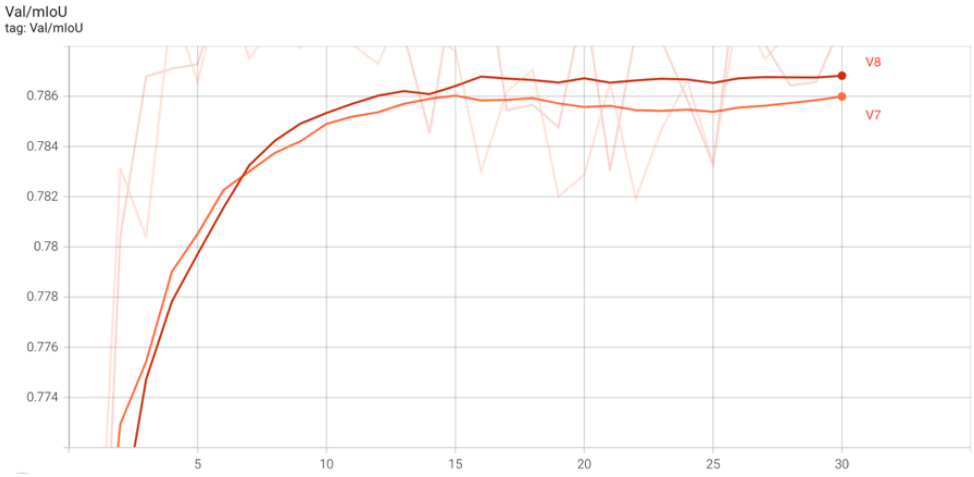

Fig. 9. NAS-MACU performs really well in a low-information environment. Darker color lines of the curves are smoothened curves and lighted color are the actual result curves

NAS-MACU performance vs MACU In Table 6 Performance comparison between NAS-MACU and MACU using training environment parameters of: Set 20000, Epochs 20, Batch 8, image 160x160 is presented.

Table 6. Performance comparison between NAS-MACU and MACU.

Net	GT	TP	FP	Recall, TPR	Precision, PPV	FPO	F_1	Jaccard
MACU	7376	7144	1178	0.969	0.858	14.16	0.910	**0.638**
NAS-MACU-V5	7376	7017	768	0.951	0.901	9.87	0.926	0.607
NAS-MACU-V6	7376	7117	1048	0.965	0.872	12.84	0.916	0.588
NAS-MACU-V7	7376	7058	1011	0.957	0.875	12.53	0.914	0.586
NAS-MACU-V8	7376	7032	657	0.953	**0.915**	**8.54**	**0.934**	0.624

MACU-NAS performance has proven to surpass the manually pitched MACU for this set of object recognition problems for this type of dataset. It performed really well especially in the low-information intensity environment even again the manually-long crafted MACU net. It took a few hours of AutoML work and GCP to compute to produce this high-performing NAS-MACU infrastructure vs months of work otherwise required by the researchers and practitioners in the object recognition/semantic segmentation space. Also, using these brand new AutoML techniques, it's possible to run and calibrate this process for high-performance and a very wide range and dispersity of problems, object types, dataset specifications, and resolution limitations.

In Table 7 performance comparison between NAS-MACU and MACU in low-information environment is compared using such training environment parameters: Set 10000, Epochs 20, Batch 8, image 160x160.

Table 7. NAS-MACU (v8) vs MACU in the low information environment.

Net	GT	TP	FP	Recall	Precision	FPO	F1	Jaccard
MACU	7376	7083	1061	0.960	0.870	13.03	0.913	0.619
NAS-MACU-V8	7376	6922	699	0.938	0.908	9.17	0.923	0.590

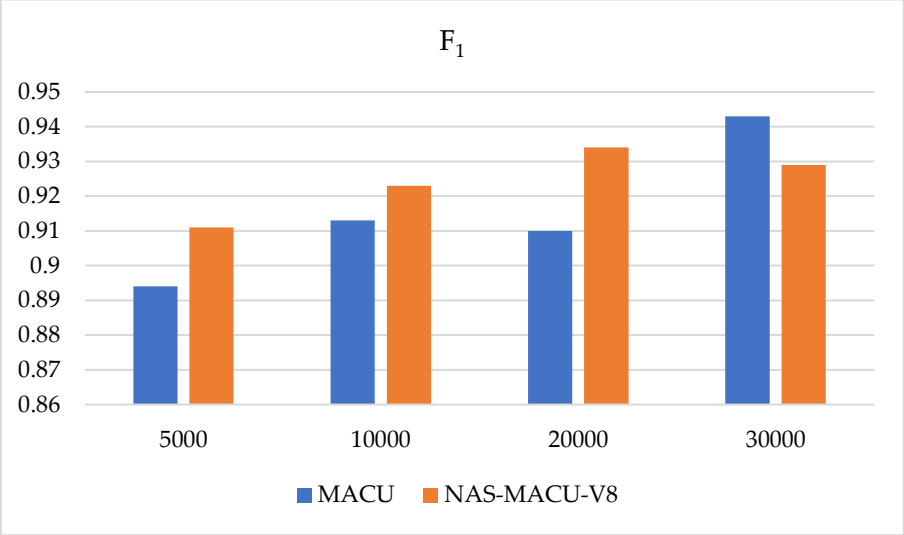

Fig.10. F_1 performance of NAS-MACU-V8 vs MACU in four different information intensity environments: *x* axis presents the number of training size; y axis represents the overall F_1 accuracy performance

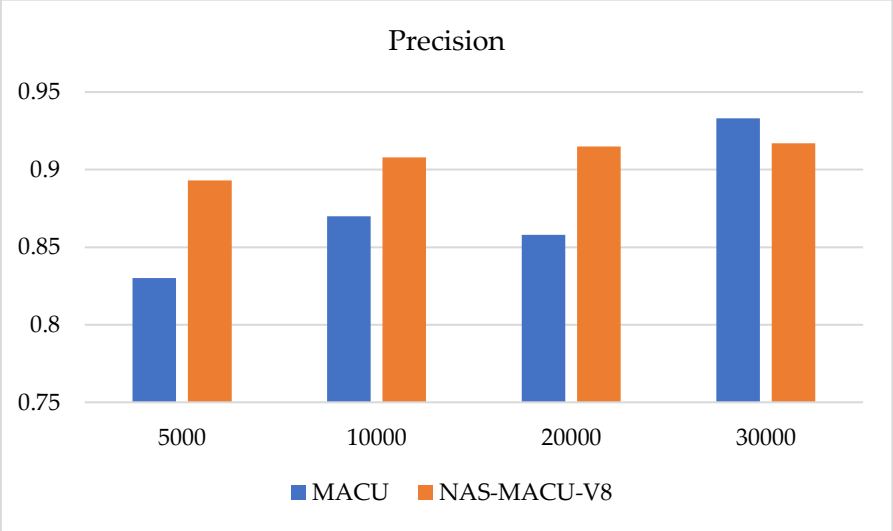

Fig. 11. Precision of NAS-MACU-V8 vs MACU in four information intensity environments: *x* axis presents the number of training size; *y* axis represents the overall performance in precision metric

After conducting an empirical investigation, we can confirm that NAS-MACU-V8 outperforms the MACU network especially once the information intensity gets reduced (30k – 5k). Two most important metrics to measure that are F_1 (overall performances) and Precision. NAS-driven genotype has outperformed a hand-crafted MACU network in both overall accuracy performance (F*1*) and Precision

metrics in any information constraint environment and with increasing delta as training set size gets reduced (Figure 10 and Figure 11). Conducting NAS operation has taken from 4h to 58h of training and search time across NAS-MACU-V1 – NAS-MACU-V7 genotypes. This was done automatically and without human intervention making this solution applicable at scale and a vast range of real-world applications.

6. Conclusions Neural Architecture Search, as part of the AutoML meta-learning field when applied to CNN cell level topology search, discovered the genotype and architectures not previously known for humans previously. It delivered the SOTA performing, self-topology-designing, NAS-MACU network that adapts to high dispersity of datasets and without human expertise or manual intervention. NAS-MACU performed particularly well in in a low information environment as compared to any other network designed to date. It's particularly important discovery for remote sensing field due to limitations of an available training sets of satellite imagery. Currently it's a limitation for the deep learning models to be effectively researched and applied to the real-world problems. NAS-MACU solved this given these real-world constraints at scale. We produced an optimized and automatically-generated MACU-NAS neural network that was able to generate better performance in accuracy and precision. with 0.915 Correctness measure an extremely low False Positive rate of 8.54 and an overall F1 score of 0.934.

This NAS procedure has the capability to be applied across various datasets and object recognition problems within the remote sensing research field. An important finding as a result of the experimentation is that our NAS-MACU v8 has significantly outperformed the manually designed MACU and other networks, especially in low-information and minimal data training environments with 0.923 (F_1) vs 0.913 (F_1) in a constrained training environment (Set =10,000 instead of 30,000) and doubled the difference in performance in a further constrained training environment (Set = 5,000).

Acknowledgments This research has received funding from the Research Council of Lithuania (LMTLT), agreement No. S-MIP-21-53.

References

[1] M. Dabboor, I. Olthof, M. Mahdianpari, F. Mohammadimanesh, M. Shokr, B. Brisco and S. Homayouni, "The RADARSAT Constellation Mission Core Applications: First Results," *Remote Sensing,* vol. 14, no. 2, p. 301, 14(2):301.

[2] A. Le Quilleuc, A. Collin, M. F. Jasinski and R. Devillers, "Very High-Resolution Satellite-Derived Bathymetry and Habitat Mapping Using Pleiades-1 and ICESat-2," *Remote Sensing,* vol. 14, p. 133, 2022.

[3] "European Space Agency," August 1 2022. [Online]. Available: https://earth.esa.int/eogateway/missions/vision-1. [Accessed 1 August 2022].

[4] Department of Space of ISRO, "Indian Space Research Organization," [Online]. Available: https://www.isro.gov.in/Spacecraft/cartosat-3. [Accessed 1 August 2022].

[5] J. G. Singla and T. Sunanda, "Generation of state of the art very high resolution DSM over hilly terrain using Cartosat-2 multi-view data, its comparison and evaluation," *Journal of Geomatics*, vol. 16, no. 1, 2022.

[6] M. Dixit, K. Chaurasia and V. K. Mishra, "Dilated-ResUnet: A novel deep learning architecture for building extraction from medium resolution multi-spectral satellite imagery," *Expert Systems with Applications*, vol. 184, 2021.

[7] H. Liheng, L. Hu and H. Zhou, "Deep learning based multi-temporal crop classification," *Remote sensing of environment*, vol. 221, pp. 430-443, 2019.

[8] X. Yang, "Urban surface water body detection with suppressed built-up noise based on water indices from Sentinel-2 MSI imagery," *Remote sensing of environment*, vol. 219, pp. 259-270, 2019.

[9] J. Cavender-Bares, "Integrating remote sensing with ecology and evolution to advance biodiversity conservation," *Nature Ecology & Evolution*, vol. 6, no. 5, pp. 506-519, 2022.

[10] S. Borra, T. Rohit and D. Nilanjan, Satellite image analysis: clustering and classification, Singapore: Springer, 2019.

[11] L. Baier, F. Jöhren and S. Seebacher, "Challenges in the Deployment and Operation of Machine Learning in Practice," *ECIS*, vol. 1, 2019.

[12] R. Li, D. Chenxi, S. Zheng, C. Zhang and P. Atkinson, "MACU-Net for Semantic Segmentation of Fine-Resolution Remotely Sensed Images," *arXiv preprint arXiv:2007.13083*, 2020.

[13] 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.

[14] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR)*, vol. 9, pp. 381-386, 2020.

[15] M. Lindauer and F. Hutter, "Best practices for scientific research on neural architecture search," *ournal of Machine Learning Research*, vol. 21, no. 243, pp. 1-18, 2020.

[16] A. Cracknell, "The development of remote sensing in the last 40 years," *International Journal of Remote Sensing*, vol. 39, no. 23, pp. 8387-8427, 2018.

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

[18] D. Meng and S. Lina, "Some new trends of deep learning research," *Chinese Journal of Electronics*, vol. 28, no. 6, pp. 1087-1091, 2019.

[19] J. M. a. F. H. T. Elsken, "Neural architecture search: A survey," *Journal of Machine Learning Research*, vol. 20, no. 55, p. 1–21, 2019.

[20] B. Z. a. Q. Le, "Neural architecture search with reinforcement learning," *arXiv preprint, arXiv:1611.01578*, 2016.

[21] V. V. J. S. e. a. B. Zoph, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, 2018.

[22] T. Z. Y. L. a. X. Q. Y. Weng, "NAS-Unet: Neural Architecture Search for Medical Image Segmentation," *IEEE Access*, vol. 7, pp. 44247-44257, 2019.

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

[24] 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.

[25] 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.

[26] 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.

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

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

[29] h. e. a. Kaiming, "Deep residual learning for image recognition," in *IEEE conference on computer vision and pattern recognition*, 2016.

[30] H. Corentin, S. Azimi and N. Merkle, "Road segmen- tation in SAR satellite images with deep fully convolutional neural net- works," *IEEE Geoscience and Remote Sensing*, p. 1867–1871, 2018.

[31] 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.

[32] 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.

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

[34] C. Liang-Chieh, "Semantic image segmentation with deep convo- lutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[35] L.-C. Chen, "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.

[36] 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.

[37] S. C. Yurtkulu, Y. Şahin and G. Unal, "Semantic segmentation with extended DeepLabv3 architecture," in *IEEE: 27th Signal Processing and Communications Applications Conference (SIU)*, 2019.

[38] Z. Zhou, "Unet++: A nested u-net architecture for medical image segmentation," *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pp. 3-11, 2018.

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

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

[41] W. Sanghyun, "Cbam: Convolutional block attention module," *In: Proceedings of the European conference on computer vision (ECCV)*, pp. 3-19, 2018.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] J. Chen, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[49] 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.

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

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

[52] 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.

[53] 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.

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

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

[56] 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.

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

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

[59] 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.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

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

[66] 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.

[67] P. Gudžius, O. Kurasova, V. Darulis and E. Filatovas, "VU DataScience GitHub Depository," 1 August 2021. [Online]. Available: https://github.com/VUDataScience/Deep-learning-based-object-recognition-in-multispectral-satellite-imagery-for-low-latency-applicatio.

[68] V. Iglovikov, S. Mushinskiy and V. Osin, "Satellite Imagery Feature Detection using Deep Convolutional Neural Network: A Kaggle Competition," *https://arxiv.org/abs/1706.06169*, 2017.

[69] P. Gudzius, O. Kurasova, V. Darulis and E. Filatovas, "Deep learning based object recognition in satellite imagery," *Machine Vision and Applications*, 2021.

[70] B. Zoph, V. Vasudevan and J. Shlens, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah, USA, 2018.

[71] B. Le and Q. V. Zoph, "Neural architecture search with reinforcement learning," *arXiv preprint, arXiv:1611.01578*, 2016.

[72] T. Elsken, J. H. Metzen and F. Hutler, "Neural architecture search: A survey," *Journal of Machine Learning Research*, vol. 20, no. 55, p. 1–21, 2019.