Exploring Deep Learning Capabilities for Coastal Image Segmentation on Edge Devices

Jonay Suárez-Ramírez[1][a], Alejandro Betancor-Del-Rosario[2][b], Daniel Santana-Cedrés[2][c] and Nelson Monzón[1,2][d]

[1]*Qualitas Artificial Intelligence and Science, Spain*

[2]*CTIM, Instituto Universitario de Cibernética, Empresas y Sociedad, University of Las Palmas de Gran Canaria, Spain*

www.qaisc.com

Keywords: Computer Vision, Deep Learning, Semantic Segmentation, Seaside Scenes, Edge Devices.

Abstract: Artificial Intelligence (AI) has become a revolutionary tool in multiple fields in the last decade. The appearance of hardware with improved capabilities has paved the way to apply image processing based on Deep Neural Networks to more complex tasks with lower costs. Nevertheless, some environments, such as remote areas, require the use of edge devices. Consequently, the algorithms must be suited to platforms with more constrained resources. This is crucial in the development of AI systems in seaside zones. In our work, we compare a wide range of recent state-of-the-art Deep Learning models for Semantic Segmentation over edge devices. Such segmentation techniques provide a better scene understanding, in particular in complex areas, providing pixel-level detection and classification. In this regard, coastal environments represent a clear example, where more specific tasks can be performed from these approaches, such as littering detection, surveillance, and shoreline changes, among many others.

1 INTRODUCTION

The "Blue Economy" focuses on the role of the marine environment and the coastal zones of our planet as an economic source. Moreover, highlights the importance of managing its resources efficiently by restoring damaged ecosystems, and introducing technology and innovation that allow sustainable use in the future (Addamo et al., 2022). Technologically speaking, the marine ecosystem offers many opportunities for the development and application of Artificial Intelligence tools in interesting topics, such as maritime surveillance (Wiersma and Mastenbroek, 1997; Frost and Tapamo, 2013; Yang et al., 2018), smart tourism (Ulrike Gretzel and Koo, 2015; Tsaih and Hsu, 2018), forecasting algal blooms (Anderson, 2009; Samantaray et al., 2018), forecasts of regional sea-level rise (Yang et al., 2020) and storm surges (Wang et al., 2020), prevention of coastal erosion (Peponi et al., 2019), among a wide spread of applications.

In recent years, Artificial Intelligence strategies, specially approaches based on Neural Networks, have revolutionized many fields. Examples are medical diagnosis (Guo et al., 2017; Amato et al., 2013), Natural Language Processing (NLP) (Collobert et al., 2011; Vaswani et al., 2017) or computer vision (Krizhevsky et al., 2012; Redmon et al., 2015; Guo et al., 2016), opening new uses that did not exist or improving substantially the pre-existing ones. Some of the keys to its recent success are the growing amount of accessible data today and the extraordinary advances in hardware devoted to parallel computing (Shi et al., 2016a; Wang et al., 2019).

Plenty of these advances are guiding us to a "Smarter World". Thus, many proposals (Ullah et al., 2020) have been presented to improve the efficiency of city services in traditional smart city applications. For instance, smart homes, smart healthcare, smart transportation, smart security, etc. Smart seaside cities can also improve their capacities due to Deep Learning approaches and remote sensors. We must also notice that, in order to work in coastal areas where may suffer from poor connectivity, remote sys-

[a] https://orcid.org/0000-0002-6914-8308
[b] https://orcid.org/0000-0003-0591-9553
[c] https://orcid.org/0000-0003-2032-5649
[d] https://orcid.org/0000-0003-0571-9068

Suárez-Ramírez, J., Betancor-Del-Rosario, A., Santana-Cedrés, D. and Monzón, N.
Exploring Deep Learning Capabilities for Coastal Image Segmentation on Edge Devices.
DOI: 10.5220/0011615400003417
In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 409-418
ISBN: 978-989-758-634-7; ISSN: 2184-4321

tems are normally mandatory. In fact, nowadays it is a common assumption that the AI methods must be embedded in edge computing systems (Shi et al., 2016b; Satyanarayanan, 2017).

In this context, we focus on deeply analyzing AI strategies running on edge devices. In particular, we center the study on seaside scenarios captured by cameras. The aim is to provide a fully understanding of the scene, by applying semantic segmentation techniques. Such methods give pixel-level information, assigning a class label to each one of them (person, animal, sea, sand, car, etc.). Hence, it can be obtained low-level information, which is priceless for a wide variety of high-level applications.

Therefore, we analyze the behavior of image segmentation through Deep Learning pre-trained models using two datasets of seasides scenes. With the application of semantic segmentation in these environments, we pursue a better understanding of the images. That allows further applications to guide more specific tasks, such as garbage detection or surveillance, that could have a great impact on user experience in the area. All these artificial intelligence applications in vision should be the key to industrial products for maritime safety, and smart tourism, among others.

Traditionally, urban environments have been deeply explored using semantic segmentation techniques, due to the interest in areas such as autonomous driving. Nevertheless, to the best of our knowledge, there has been no review study of this type on coastal imagery. Moreover, this scenario with particular computing conditions can provide researchers with systematic reference information, which is the motivation of our comparative analysis. Besides, the deployment of Deep Learning models on low compute devices is an increasingly important area of research. In this sense, our comparison is focused on models that work in edge devices.

The paper is organized as follows: in section 2, we include recent state-of-the-art works regarding semantic segmentation, based on Convolutional Neural Networks, as well as Transformers. On the other hand, section 3 describes the experimental setup, encompassing the hardware description, the datasets employed, and the results that we have obtained. Finally, we include the conclusions in section 4.

2 RELATED WORKS

There are two main approaches to tackle image segmentation tasks, namely those based on Convolutional Neural Networks (CNNs) (Lindsay, 2021), and the ones derived from Transformers (Khan et al., 2022). In this work, we have evaluated models belonging to each one of them (about 24, combining 8 different backbones with multiple methods - see Tables 5 and 6 for more details), aiming to provide a clear and objective analysis of such different approaches.

More classical methods are based on Convolutional Neural Networks, although new proposals keep arising. A clear example is the Deeplab method, and its successive iterations (Deeplabv3 and Deeplabv3+) (Chen et al., 2017). Its main characteristic is the use of atrous or dilated convolutions and the Atrous Spatial Pyramid Pooling module to take advantage of information from a larger neighborhood with the same computational cost. A similar inspiration is followed by PSPNet (Pyramid Scene Parsing Network) (Zhao et al., 2017), which is a semantic segmentation method that utilizes a pyramid parsing module that exploits global context information by different-region-based context aggregation. A more reliable prediction is obtained by joining the local and global clues together.

Using a global image representation, APCNet (He et al., 2019) adaptively constructs multi-scale contextual representations with multiple designed Adaptive Context Modules (ACMs). Such modules leverage these global representations, guiding the estimation of local affinity coefficients for each sub-region, and then calculate a context vector with these affinities.

The non-local block is a popular module for strengthening the context modeling ability of a regular CNN. This block attention computation can be split into two terms, a whitened pairwise term accounting for the relationship between two pixels and a unary term representing the saliency or prominence of every pixel. However, the two terms are tightly coupled in the non-local block, which hinders the learning of each. Disentangled Non-Local Neural Networks (DNL) (Yin et al., 2020) decouples these two terms to facilitate learning for both.

As a ResNet variation, ResNeSt (Zhang et al., 2022) proposes the channel-wise attention on different network branches to leverage their success in capturing cross-feature interactions and learning diverse representations, through a single unified Split-Attention block. In this way, feature representation is improved, which is useful in multiple applications.

Another viewpoint is found in SegNeXt (Guo et al., 2022), where a combination of a CNN with an attention module based on MSCA (multibranch spatial-channel attention) is proposed. The authors rely on a better efficiency of the convolutional approach to extract contextual information, by using good characteristics present in previous segmentation models.

Aiming to embed Transformer features into a CNN architecture, in HorNet (Rao et al., 2022), a new operation to perform high-order spatial interactions (Recursive Gated Convolution) is proposed. Thus, a new family of vision backbones (HorNet) is provided, by replacing the spacial mixing layer in various Transformers. This can be done by using the new operation, which is more efficient, extensible, and translation-equivariant.

On the other hand, Transformers were originally applied to Natural Language Processing (NLP), by using their self-attention mechanism to recognize different parts of input data. Their application to Computer Vision (CV) tasks relied on adapting the architecture to the structure of visual data, by modifying network designs and training techniques.

Initially, Vision Transformer (ViT) proposal (Dosovitskiy et al., 2021) was the first step towards the unification and cross-area research sharing between CV and NLP. ViT is the application of a well-known NLP architecture, the Transformer (Vaswani et al., 2017), to CV. For this aim, ViT divides the images into a grid of $S \times S$ patches and considers every patch as a token, working similarly to the original NLP architecture. Swin Transformer (Liu et al., 2021) is an evolution of ViT, but applies a hierarchical structure using windows. In this way, it divides the images into a non-constant size grid of windows and split each window into a constant size grid of patches. This approach allows availing finer details in the image without the need for a large and computationally costly grid with ViT.

Segmenter (Transformer for Semantic Segmentation) (Strudel et al., 2021), is also a ViT approach. It aims to model global context from the very beginning of the architecture and through the whole network without using convolutions. The authors propose a family of models with different levels of resolution, with the intention of a trade-off between time and performance. As a result, it is estimated that this model can provide a unified approach for different sorts of segmentation (semantic, instance, and panoptic).

Following with ViT strategies, in (Chen et al., 2022), an adapter for ViT is proposed. The aim is to avoid the lower performance on dense prediction of ViT, by introducing biases with additional architecture. This architecture consists of a spatial prior module and two feature interaction operators. Such an adapter is connected to a general backbone, in order to introduce prior information of input data, making the network suitable for downstream tasks.

In order to avoid a finetuning of transformer backbone networks, in SeMask model (Jain et al., 2021) is proposed to include a semantic prior to guiding the encoder's feature modeling. In this way, the proposed model can be plugged into any hierarchical ViT, with the objective to acquire semantic context and improve its representation by using semantic attention operation.

Finally, the proposal of SegFormer (Xie et al., 2021), is to unify Transformers with multilayer perceptron decoders. By redesigning the encoder an decoder, the authors consider jointly the efficiency, accuracy, and robustness. The main idea is to avoid the complex designs of previous approaches.

3 EXPERIMENTS

In this section, we describe the experimental setup. Firstly, the hardware system is described, including capture and edge computing devices. Afterward, we include the details regarding the datasets employed in the experiments, to finally explain the experimental results obtained.

3.1 Hardware

Concerning the hardware system, we appraise two main elements. On the one hand, we have the camera, from which the images will be captured. On the other hand, we consider the edge device, devoted to performing the computations on the input data by using different Deep Learning models. Notice that such a description relies on the scenario that we manage, and the configuration of the capture systems for other datasets (such as in the case of ArgusNL) may vary, as well as the edge device features.

Regarding the capture system, a PTZ camera (Hikvision DS-2DF8836I5X) has been used to obtain images of coastal scenes. It has three degrees of freedom, provided by its inherent pan, tilt, and zoom capabilities. It is mounted on a bracket, and at a height enough to provide a wide view of the coast. This camera is able to capture images up to 4K resolution but we use $1920x1080$ resolution to achieve faster results. A more detailed description of camera features can be found in Table 1.

The current technological trends like Internet of Things (IoT) or autonomous vehicles are boosting the use of neural networks in remote devices and so that require appropriate hardware , such as embedded computing boards. NVIDIA Jetson is the family of NVIDIA products specifically designed for Edge Computing, characterized by having a good relation between performance versus energy consumption and

Table 1: Main Hikvision DS-2DF8836I5X camera specifications.

Feature	Description
Image sensor	2/3” CMOS
Shutter time	1/1 s - 1/30000 s
Focal length	7.5 mm - 270 mm
Optical zoom	x36
Pan range	360º
Tilt range	-20º a 90º
Maximum resolution	4K
Dimensions	Θ 266.6 mm × 410 mm

size. In this work, the edge device we used to test memory consumption restriction and inference time measuring is an NVIDIA Xavier NX 8GB with JetPack 4.5 installed. Main Xavier NX model specifications are presented in table 2 (find more details at this link).

Table 2: Nvidia Jetson Xavier NX specifications.

Feature	Description
AI Performance	21 TOPS
GPU	384-core NVIDIA Volta GPU with 48 Tensor Cores
CPU	6-core NVIDIA Carmel 64-bit CPU
Memory	8 GB 128-bit LPDDR4x
Power	3 modes of 10 / 15 / 20 W
Dimension	69.6 mm x 45 mm

Therefore, considering the constraints presented in table 2, it is clear that a detailed analysis of different models and their performance would be useful to identify a trade-off between throughput and results obtained.

3.2 Datasets

In order to test the models described in the above section 2, two different datasets have been used. On the one hand, we experiment with the ArgusNL dataset, which includes a set of images captured on the Dutch coast and manually annotated. On the other hand, a second dataset is proposed, obtained from a location on the South-West coast of the Gran Canaria island. In this section, we include a more detailed description of both datasets.

The ArgusNL dataset (Hoonhout et al., 2015), consists of 192 images (snapshots) taken during the summer of 2013 and manually annotated. To obtain them, 4 different coastal camera stations have been used, placed on the Dutch coast (Egmond, Jan van Speijk, Kijkduin, and Sand Motor). These stations count with multicamera systems, with setups ranging from 5 to 8 cameras. This dataset was originally published in (Hoonhout and Radermacher, 2014), and contains snapshots captured at different moments of the day, providing a variety of light conditions. These images have been captured in RGB color code, JPG format, and with resolutions of 2448 × 2048 and 1392 × 1040 (probably depending on the station setup used to perform the capture). Associated with each one of them, a pickle (.pkl) file with the manual annotations is included.

In addition, we also use a second dataset denominated Smart Coast Segmentation Dataset (SCSD), which is provided by the R&D company Qualitas Artificial Intelligence and Science S.A. (QAISC). In the context of the project ”Smart Coast AI solutions for tourism 4.0” led by QAISC, several cameras have been deployed in harbors, marinas, beaches, and hotels on Gran Canaria island. SCSD includes about 36 images, with different scenes and light/shadow conditions. Such images were captured from two cameras installed in the South-West of Gran Canaria island [1]. The images have been obtained in RGB color code, JPG format, with a resolution of 1920 × 1080. Conversely to ArgusNL dataset, SCSD provides homogenous resolutions. Therefore, is easier to perform resolution dependant experiments with the whole dataset. Along with the images, PNG grayscale files are included with the annotations following ADE20K annotations style and indexes (Barriuso and Torralba, 2012).

To illustrate the classification performed by semantic segmentation techniques, in Table 3, we include the details of the correspondences between the different classes and their associated colors. Please, note that in the case of the SCSD dataset, the color codes are based on the ADE20K dataset, whereas ArgunsNL uses its own colors (with light differences among both datasets). Furthermore, in ArgusNL classification, each pixel that has not been classified as any of the considered classes is assigned to the object class. The classes without an associated color are marked with a dash.

3.3 Experimental Results

In this section, we show the experimental results obtained. To this aim, we apply the models described in section 2 to both datasets. About 24 different models have been used, by combining 8 backbones with multiple methods. Regarding the model implementations and weight files, Upernet-Swin, ConvNext, Segmenter, Segformer, Resnet and ResneSt come from

[1] https://www.smartcoast.info/

Table 3: Correspondences between classes and colors for SCSD and ArgusNL datasets. In the SCSD dataset, the vegetation class is a virtual class we created to join *tree, grass, plant, and palm* classes.

Class	Colors	
	SCSD	ArgusNL
background	⬛	—
wall	◼	—
building	🟫	—
sky	🟦	🟦
floor	🟫	—
tree	🟩	—
grass	🟩	—
earth	🟫	—
plant	🟨	—
sea	🟦	🟦
rock	🟥	—
sand	🟨	🟧
bridge	🟧	—
palm	🟦	—
boat	🟩	—
swim pool	🟦	—
pier	🟪	—
vegetation	🟩	🟩
object	—	◼

MMSegmentation and the rest from their original implementations. In this way, qualitative and quantitative results are included to show the robustness of the different approaches. Our final objective is to obtain a trade-off between precision and computational cost.

Aiming to illustrate the results obtained, in Table 4 the outcomes for some of the best performing Semantic Segmentation models are depicted. Additionally, we consider a variety of illumination conditions: optimum, medium, and bad. In the first two rows, we include the inputs and their associated ground-truths, whereas in the rest of the rows the results for different models are represented. Note that, although the color correspondences seem not to agree with the ground-truth, this is due to the fact that the image segmentation result is superimposed on the input image with transparency, which produces slight differences.

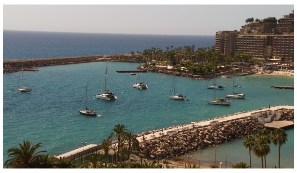
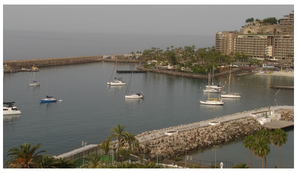
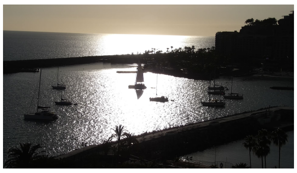
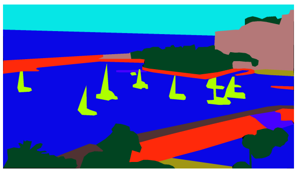
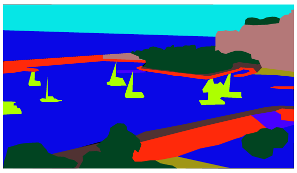
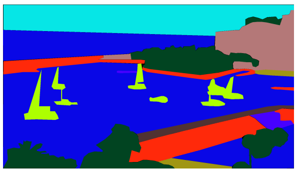
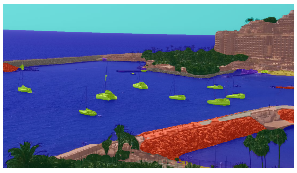
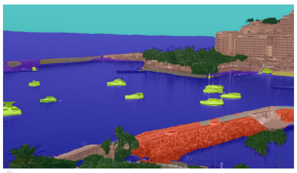
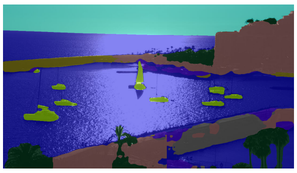
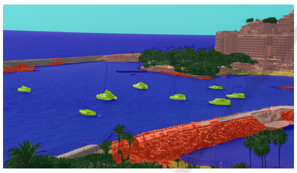
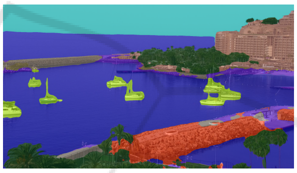
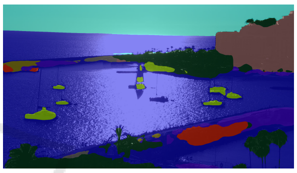
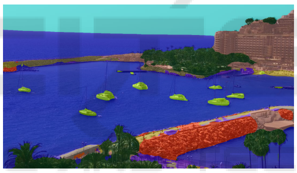
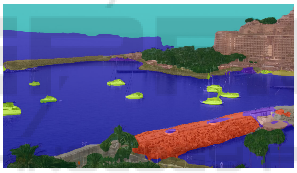
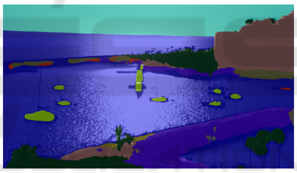
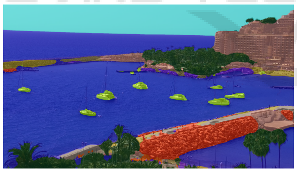
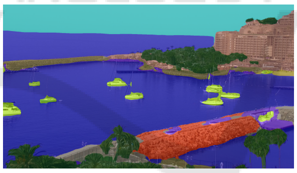
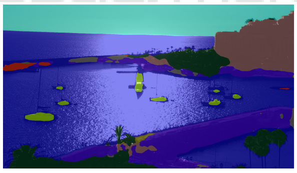
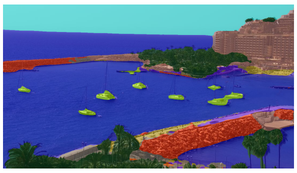
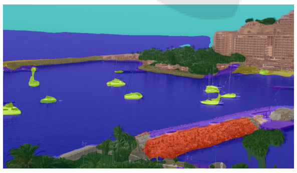
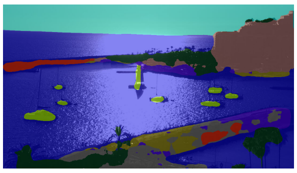
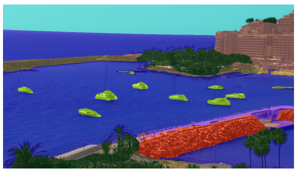
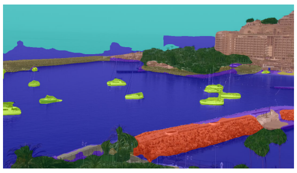
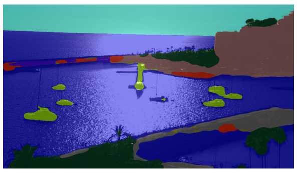
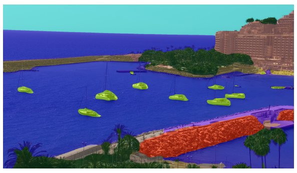
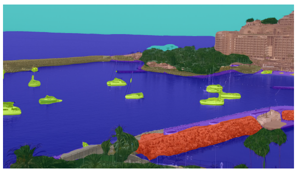
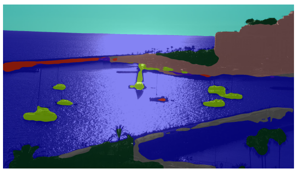
As observed, with optimum illumination all of them correctly segmentate the biggest regions in the image, sea, sky, building, and vegetation. Under worse illumination conditions (third column) some models start to struggle to detect a diffuse horizon line like Swin-B384 and ViT-Adapter with lower resolution. Segmenting an image with a very bad illumination (fourth column) is a challenging task. All of the depicted models struggle to accurately classify both dikes areas. Only Hornet-L-GF is able to correctly keep the sea shape around the dike in the back. With bad illumination, the difference between resolutions is hardly noticeable.

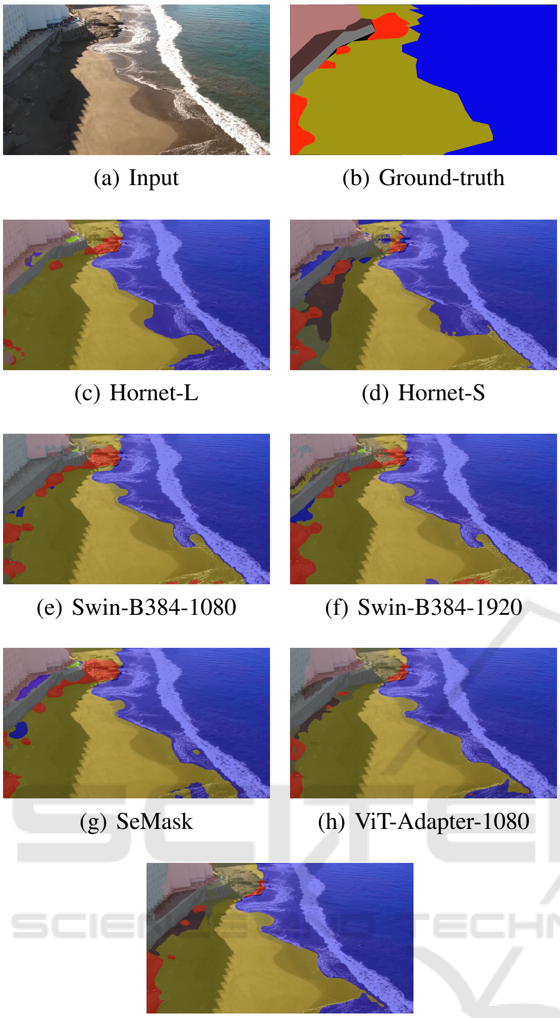
On the other hand, in Figure 1 we show different results when the input image is affected by shadows. Thus, we include the input image with the shadow (Figure 1(a)) and its corresponding ground-truth (Figure 1(b)). In particular, here we can see that there are two regions that are tougher to classify: the building shadow and the wet sand. Most models perform similarly segmenting the sea and most of the sand but also fail to segment the building and shadowed sand area next to the promenade. Like in the illumination experiment, Hornet-L-GF (Figure 1(c)) is the best segmenting in these difficult regions and Swin-B384 (Figure 1(e)) with 1080 pixels width works acceptably on this area but miss-classifies half of the building.

For the purpose of quantitatively characterizing the results obtained with the different models, we include figures in Tables 5 and 6. In each one, the three models with the overall best performance are highlighted in bold. In Table 5 we show the numerical results for the ArgusNL dataset. In this way, numbers related to the mean intersection over union ($\overline{IoU}$), mean accuracy ($\overline{Acc.}$), and absolute accuracy ($|Acc.|$) are presented. On the ArgusNL dataset, three backbones outstand, HorNet, ViT, and Swin. In the same way, ViT-Adapter, HorNet-L-GF, and HorNet-S-7x7 rank top-3 in all metrics, but not in the same order, and achieve more than 70 % in $\overline{IoU}$. Segmenter_B and Swim_B224-22k perform very close to the 70 % $\overline{IoU}$ barrier as well. Averaging the three metrics ViT-Adapter works best in this dataset closely followed by HorNet-L-GF.

Table 6 includes figures regarding the SCSD dataset. As described in section 3.2, this dataset provides a homogeneous set of images with the same resolution, which allows us to perform resolution-dependent experiments. In addition to the metrics presented in the previous table, we also include the inference time for the first inference and the mean for the rest of them. All these experiments have been performed for two image widths: 1080 and 1920, when feasible. For an image width of 1080 pixels, top-3 $\overline{IoU}$ are HorNet-L-GF, ConvNext_B640, and Semask-FPN-Swin-L but their average inference time is bottom-4 and over 7.5 s which is almost prohibitive. Performance and inference time objectives are opposed so we must keep in mind this trade-off. When we use the original image width, 1920 pixels, we are boosting the performance of modern models to the detriment of their resources consumption and execution time. Some models consume more RAM memory than available so must be discarded with this

Table 4: Semantic Segmentation results under optimum, medium, and bad illumination conditions, respectively.

Models	**Illumination**		
	Optimum	**Medium**	**Bad**
Input			
ground-truth			
Hornet-L-GF			
Hornet-S-7x7			
Swin-B384 (1080)			
Swin-B384 (1920)			
SeMask			
ViT-Adapter (1080)			
ViT-Adapter (1920)			



(a) Input (b) Ground-truth

(c) Hornet-L (d) Hornet-S

(e) Swin-B384-1080 (f) Swin-B384-1920

(g) SeMask (h) ViT-Adapter-1080

(i) ViT-Adapter-1920

Figure 1: Results for different models when the input image is affected by shadows.

Table 5: Results for ArgusNL dataset. From left to right: backbone, model, mean IoU, mean accuracy, and absolute accuracy. All HorNet and Swin backbones models were combined with Upernet unless other method is specified.

Backbone	**Model**	**Metrics (%)**		
		$\overline{\textbf{IoU}}$	$\overline{\textbf{Acc.}}$	**|Acc.|**
ConvNext	ConvNext_B640	57.47	74.55	65.17
Hornet	HorNet-L-GF	**71.71**	**84.59**	**82.87**
	HorNet-L-7x7	67.17	80.61	76.21
	HorNet-B-GF	59.54	78.04	73.69
	HorNet-B-7x7	63.59	80.27	74.48
	HorNet-S-GF	62.82	78.64	77.27
	HorNet-S-7x7	**70.60**	**84.60**	**83.22**
MiT	Segformer_B5	66.55	81.48	79.75
MSCAN	SegNext-L	58.50	77.50	71.86
	SegNext-B	62.85	79.15	75.55
Resnet101	DNL	65.88	80.26	80.04
	PSPNet	49.73	69.12	62.94
	APCNet	51.89	71.34	65.53
ResneSt101	PSPNet	55.58	74.79	67.65
	DeepLabv3	53.79	73.07	67.19
	DeepLabv3+	60.15	76.72	71.78
Swin	Upernet_Swin-B384-22k	67.45	81.07	80.51
	Upernet_Swin-B384-1k	61.19	79.33	75.40
	Upernet_Swin-B224-22k	69.85	83.72	82.53
	Upernet_Swin-B224-1k	58.82	78.64	70.65
	Upernet_Swin-S	63.04	80.28	75.56
	SeMask-FPN-Swin-L	68.26	81.59	82.57
ViT	Segmenter_B	69.79	81.29	82.67
	ViT-Adapter_AugReg-B	**71.87**	**84.24**	**84.80**

resolution. Those which can work with this resolution achieve some improvements over lower resolutions, like HorNet-S-7x7, Swin-B384-22k, and ViT-Adapter which perform top-3 on $\overline{IoU}$ with images of 1920x1080 pixels.

With the aim of comparing models in terms of quality and speed, we plotted the mean IoU against average inference time in Figure 2. The more left-up corner the model is placed in, the better overall, in terms of higher $\overline{IoU}$ and faster $\overline{Inf}$. As usually happens in multi-objective optimization problems, there is no model which is best, rather we have a Pareto set composed of 7 models, which means that no other model is better than they in both objectives. Therefore, we can consider different models depending on the scenario and desired results concerning time and accuracy. According to the problem to solve, number of inferences per cycle, and quality requirements, one of those models should be chosen. With a longer working cycle and/or lower number of inferences Swin-B384-22k can be used to obtain higher quality segmentation results. In case the number of inferences required is high, a SegNext variant could be deployed, either SegNext-B with a higher image resolution or SegNext-L with a lower one. In other cases where there has to be a compromise between speed and quality, Segmenter_B with lower or medium resolution input images seems a balanced option between both.

4 CONCLUSIONS

Performance analysis of different Deep Learning models on an edge device to perform semantic segmentation tasks has been presented in this work. To this aim, two different datasets have been used, namely ArgusNL and SCSD. Combining 8 backbones with multiple methods, we have applied a total amount of 24 different models to such datasets, including qualitative and quantitative results.

Table 6: Results for the SCSD dataset. From left to right: backbone, model, mean IoU, mean accuracy, absolute accuracy, the computational time for the first inference, and mean value for the rest of the inferences (for image widths of 1080 and 1920 respectively). All HorNet and Swin backbones models were combined with Upernet unless other method is specified.

Backbone	Model	Image width									
		1080					1920				
		Metrics (%)			Time (sec.)		Metrics (%)			Time (sec.)	
		$\overline{\text{IoU}}$	$\overline{\text{Acc.}}$	|Acc.|	1^{st} Inf.	$\overline{\text{Inf.}}$	$\overline{\text{IoU}}$	$\overline{\text{Acc.}}$	|Acc.|	1^{st} Inf.	$\overline{\text{Inf.}}$
ConvNext	Upernet_ConvNext_B640	**44.82**	57.34	82.04	12.49	7.76	—	—	—	—	—
Hornet	HorNet-L-GF	**44.91**	**58.53**	**82.41**	30.42	10.03	—	—	—	—	—
	HorNet-L-7x7	43.42	56.72	**82.08**	23.97	10.50	—	—	—	—	—
	HorNet-B-GF	42.72	**58.55**	80.36	17.79	5.46	—	—	—	—	—
	HorNet-B-7x7	40.37	54.77	80.05	12.48	5.60	40.66	55.68	80.66	32.62	5.48
	HorNet-S-GF	41.86	53.39	80.90	13.01	4.82	41.89	55.80	81.27	41.55	4.78
	HorNet-S-7x7	42.43	56.94	81.44	10.62	4.54	**44.03**	**58.74**	**82.16**	40.84	4.87
MiT	Segformer_B5	40.07	51.15	79.50	7.41	2.32	—	—	—	—	—
MSCAN	SegNext-L	41.24	51.08	79.88	6.21	**1.16**	42.18	52.68	80.66	13.80	**1.44**
	SegNext-B	37.76	48.33	78.26	4.97	**0.68**	39.52	50.56	79.08	11.15	**1.00**
Resnet101	DNL	36.48	46.45	78.62	7.68	2.35	36.36	46.48	78.01	17.84	3.48
	PSPNet	36.10	46.94	78.35	4.57	2.08	35.55	46.29	77.64	10.88	**2.40**
	APCNet	36.54	47.33	76.27	6.42	2.30	36.90	48.63	76.18	13.94	3.05
ResneSt101	PSPNet	37.86	49.71	77.08	4.84	2.30	37.63	49.26	76.65	9.62	2.71
	DeepLabv3	38.78	50.26	77.29	5.36	2.95	38.57	49.96	77.24	10.10	3.42
	DeepLabv3+	36.66	47.25	76.49	5.02	2.39	37.06	47.70	76.53	11.09	2.90
Swin	Upernet_Swin-B384-22k	43.63	56.94	81.06	8.33	3.51	**44.90**	**58.68**	**81.65**	18.41	4.10
	Upernet_Swin-B384-1k	39.40	50.99	78.22	7.01	3.02	39.82	52.21	78.35	20.21	4.08
	Upernet_Swin-B224-22k	40.47	51.32	80.62	6.97	2.85	41.70	53.24	**81.46**	23.12	3.97
	Upernet_Swin-B224-1k	39.75	49.63	79.48	7.03	2.84	40.65	50.62	79.25	21.37	3.90
	Upernet_Swin-S	40.23	52.41	78.82	7.55	2.55	41.28	52.91	79.01	16.75	3.29
	SeMask-FPN-Swin-L	**44.40**	56.49	**82.05**	10.24	7.91	—	—	—	—	—
ViT	Segmenter_B	42.82	54.76	81.78	7.81	**1.82**	—	—	—	—	—
	ViT-Adapter_AugReg-B	43.54	**57.59**	81.14	11.77	6.01	**44.37**	**59.11**	81.08	39.76	10.75

As observed, the visual results show the strengths and weaknesses of the evaluated models, related to different illumination and shadow conditions. Poor illumination, strongly affects all approaches, whereas with shadows some of them perform better.

Quantitatively, both datasets have been used to study the performance of the models, including figures regarding mean IoU, average, and absolute accuracy. In addition, with the proposed SCSD dataset, inference time has been also compared, as well as different input image resolutions. The selection of the best model relies on a trade-off between precision and computational time. As presented, the scenario determines the best choice, in particular in terms of mean inference time.

To the best of our knowledge, this is the first work that presents a detailed review of the capabilities of Deep Learning models in semantic segmentation running on edge devices environment, in particular, for applications on coastal imagery. Although this is a growing area and further research is needed, we hope to contribute to solutions in remote coastal regions.

ACKNOWLEDGEMENTS

This work is the result of a collaboration between QAISC and CTIM, within the framework of the contract C2021/193 signed between the company and The Canarian Science and Technology Park Foundation of the University of Las Palmas de Gran Canaria.

It has also been supported supported by The Canarian Science and Technology Park Foundation of the University of Las Palmas de Gran Canaria through the research project F2021/05 FEI Innovación y Transferencia empresarial and by Vicepresidencia Primera, Consejería de Vicepresidencia Primera y de Obras Públicas, Infraestructuras, Transporte y Movilidad from Cabildo de Gran Canaria, through the project of reference Resolution No. 45/2021.

We also thank the tourism groups Anfi del Mar and Radisson Blu for their support within the framework of the ”Smart Coast AI Lab for tourism” initiative led by QAISC in collaboration with the CTIM. Finally, we thanks Prof. Luis Álvarez León for his guidance on a previous work precedent to this paper.

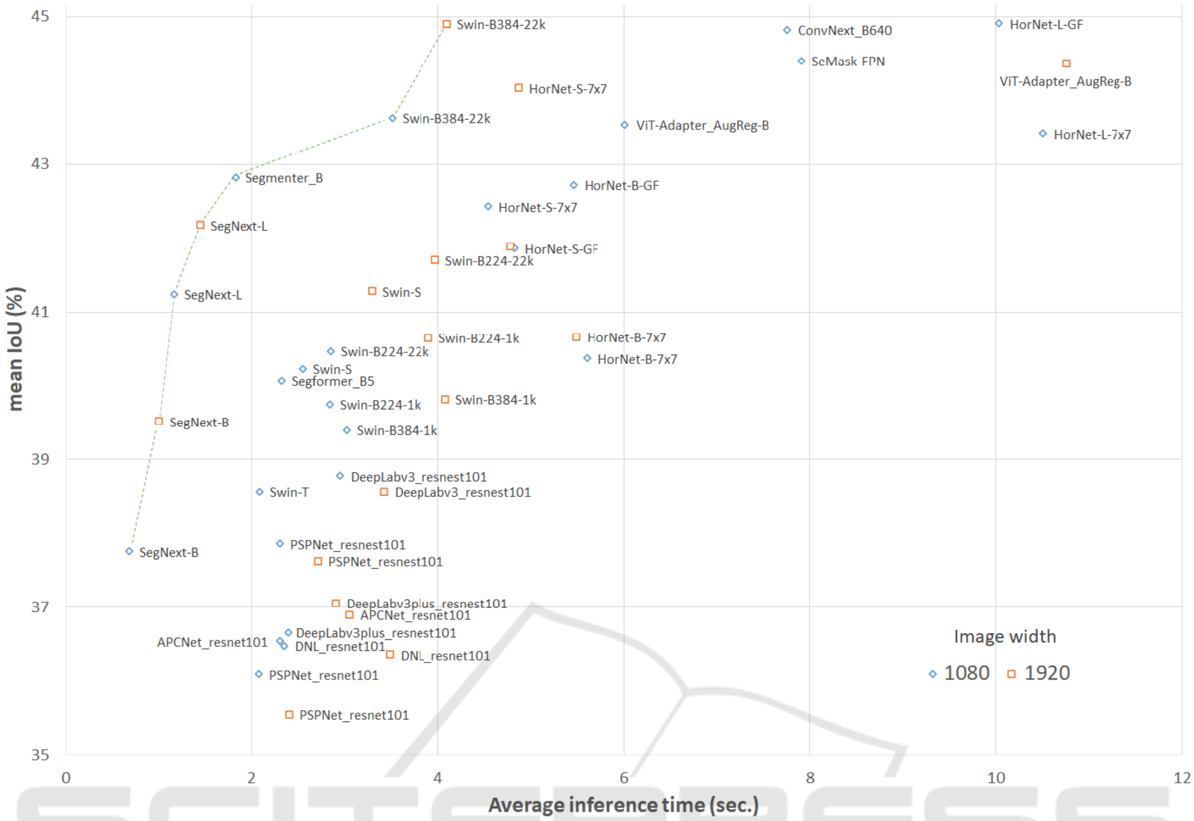


Figure 2: Mean Intersection over Union ($\overline{IoU}$) vs. average inference time ($\overline{Inf.}$) for the models in Table 6. The green dashed line represent the Pareto frontier.

REFERENCES

Addamo, A., Calvo Santos, A., Guillén, J., Neehus, S., Peralta Baptista, A., Quatrini, S., Telsnig, T., and Petrucco, G. (2022). *The EU Blue Economy Report 2022*. European Commission Directorate General for Maritime Affairs and Fisheries, and the Joint research Center. Publications Office of the European Union, Luxemburg.

Amato, F., López, A., Peña-Méndez, E. M., Vaňhara, P., Hampl, A., and Havel, J. (2013). Artificial neural networks in medical diagnosis. *Journal of Applied Biomedicine*, 11(2):47–58.

Anderson, D. M. (2009). Approaches to monitoring, control and management of harmful algal blooms (HABs). *Ocean & coastal management*, 52(7):342–347.

Barriuso, A. and Torralba, A. (2012). Notes on image annotation. Technical report, Massachusetts Institute of Technology (MIT).

Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv preprint arXiv:1706.05587*, abs/1706.05587:1–14.

Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., and Qiao, Y. (2022). Vision Transformer Adapter for Dense Predictions. *ArXiv*, abs/2205.08534:1–25.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. P. (2011). Natural Language Processing (almost) from Scratch. *CoRR*, abs/1103.0398.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Frost, D. P. and Tapamo, J.-R. (2013). Detection and tracking of moving objects in a maritime environment using level set with shape priors. *EURASIP Journal on Image and Video Processing*, 2013:1–16.

Guo, M.-H., Lu, C.-Z., Hou, Q., Liu, Z., Cheng, M.-M., and Hu, S.-M. (2022). SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation. *ArXiv*, pages 1–15.

Guo, Y., Liu, Y., Georgiou, T., and Lew, M. S. (2017). A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, 7:87–93.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., and Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48. Recent Developments on Deep Big Vision.

He, J., Deng, Z., Zhou, L., Wang, Y., and Qiao, Y. (2019). Adaptive Pyramid Context Network for Semantic Segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7511–7520.

Hoonhout, B. and Radermacher, M. (2014). Annotated images of the Dutch coast. https://doi.org/10.4121/uuid:08400507-4731-4cb2-a7ec-9ed2937db119. [Online; accessed 29-September-2022].

Hoonhout, B., Radermacher, M., Baart, F., and van der Maaten, L. (2015). An automated method for semantic classification of regions in coastal images. *Coastal Engineering*, 105:1–12.

Jain, J., Singh, A., Orlov, N., Huang, Z., Li, J., Walton, S., and Shi, H. (2021). SeMask: Semantically Masked Transformers for Semantic Segmentation. *arXiv*, abs/2112.12782:1–14.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. (2022). Transformers in Vision: A Survey. *ACM Comput. Surv.*, 54(10s):1–41.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Lindsay, G. W. (2021). Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of cognitive neuroscience*, 33(10):2017–2031.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002.

Peponi, A., Morgado, P., and Trindade, J. (2019). Combining Artificial Neural Networks and GIS Fundamentals for Coastal Erosion Prediction Modeling. *Sustainability*, 11(4):1–14.

Rao, Y., Zhao, W., Tang, Y., Zhou, J., Lim, S.-N., and Lu, J. (2022). HorNet: Efficient High-Order Spatial Interactions with Recursive Gated Convolutions. *ArXiv*, abs/2207.14284:1–15.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. *CoRR*, abs/1506.02640.

Samantaray, A., Yang, B., Dietz, J. E., and Min, B.-C. (2018). Algae Detection Using Computer Vision and Deep Learning.

Satyanarayanan, M. (2017). The Emergence of Edge Computing. *Computer*, 50(1):30–39.

Shi, S., Wang, Q., Xu, P., and Chu, X. (2016a). Benchmarking State-of-the-Art Deep Learning Software Tools. In *2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, pages 99–104.

Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L. (2016b). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5):637–646.

Strudel, R., Pinel, R. G., Laptev, I., and Schmid, C. (2021). Segmenter: Transformer for Semantic Segmentation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7242–7252.

Tsaih, R.-H. and Hsu, C. C. (2018). Artificial Intelligence in Smart Tourism: A Conceptual Framework. In *Proceedings of The 18th International Conference on Electronic Business*, pages 124–133, Guilin, China. Association for Information Systems.

Ullah, Z., Al-Turjman, F., Mostarda, L., and Gagliardi, R. (2020). Applications of Artificial Intelligence and Machine learning in smart cities. *Computer Communications*, 154:313–323.

Ulrike Gretzel, Marianna Sigala, Z. X. and Koo, C. (2015). Smart tourism: foundations and developments. In *Electron Markets*, pages 179—-188.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is All you Need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Wang, Y., Chen, X., Wang, L., and Min, G. (2020). Effective IoT-Facilitated Storm Surge Flood Modeling Based on Deep Reinforcement Learning. *IEEE Internet of Things Journal*, 7(7):6338–6347.

Wang, Y. E., Wei, G.-Y., and Brooks, D. (2019). Benchmarking TPU, GPU, and CPU Platforms for Deep Learning. *CoRR*.

Wiersma, E. and Mastenbroek, N. (1997). Measurement of Vessel Traffic Service Operator Performance. *IFAC Proceedings Volumes*, 30(24):61–64. 6th IFAC Symposium on Automated Systems Based on Human Skill 1997 (Joint Design of Technology and Organisation), Kranjska gora, Slovenia, 17-19 September.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. (2021). SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090.

Yang, C.-H., Wu, C.-H., and Hsieh, C.-M. (2020). Long short-term memory recurrent neural network for tidal level forecasting. *IEEE Access*, 8:159389–159401.

Yang, X., Sun, H., Fu, K., Yang, J., Sun, X., Yan, M., and Guo, Z. (2018). Automatic Ship Detection in Remote Sensing Images from Google Earth of Complex Scenes Based on Multiscale Rotation Dense Feature Pyramid Networks. *Remote Sensing*, 10(1):1–14.

Yin, M., Yao, Z., Cao, Y., Li, X., Zhang, Z., Lin, S., and Hu, H. (2020). Disentangled Non-Local Neural Networks. *CoRR*, abs/2006.06668:191–207.

Zhang, H., Wu, C., Zhang, Z., Zhu, Y., Lin, H., Zhang, Z., Sun, Y., He, T., Mueller, J., Manmatha, R., Li, M., and Smola, A. (2022). ResNeSt: Split-Attention Networks. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2735–2745.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid Scene Parsing Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239.