Machine Learning (2023) 112:1733–1784
https://doi.org/10.1007/s10994-021-06027-1

Fully convolutional open set segmentation

Hugo Oliveira[1] · **Caio Silva**[1] · **Gabriel L. S. Machado**[1] · **Keiller Nogueira**[1] · **Jefersson A. dos Santos**[1]

Received: 15 June 2020 / Revised: 10 June 2021 / Accepted: 17 June 2021 /
Published online: 9 July 2021

Abstract

In traditional semantic segmentation, knowing about all existing classes is essential to yield effective results with the majority of existing approaches. However, these methods trained in a Closed Set of classes fail when new classes are found in the test phase, not being able to recognize that an unseen class has been fed. This means that they are not suitable for Open Set scenarios, which are very common in real-world computer vision and remote sensing applications. In this paper, we discuss the limitations of Closed Set segmentation and propose two fully convolutional approaches to effectively address Open Set semantic segmentation: OpenFCN and OpenPCS. OpenFCN is based on the well-known OpenMax algorithm, configuring a new application of this approach in segmentation settings. OpenPCS is a fully novel approach based on feature-space from DNN activations that serve as features for computing PCA and multi-variate gaussian likelihood in a lower dimensional space. In addition to OpenPCS and aiming to reduce the RAM memory requirements of the methodology, we also propose a slight variation of the method (OpenIPCS) that uses an iteractive version of PCA able to be trained in small batches. Experiments were conducted on the well-known ISPRS Vaihingen/Potsdam and the 2018 IEEE GRSS Data Fusion Challenge datasets. OpenFCN showed little-to-no improvement when compared to the simpler and much more time efficient SoftMax thresholding, while being some orders of magnitude slower. OpenPCS achieved promising results in almost all experiments by overcoming both OpenFCN and SoftMax thresholding. OpenPCS is also a reasonable compromise between the runtime performances of the extremely fast SoftMax thresholding and the extremely

Editors: Dino Ienco, Thomas Corpetti, Minh-Tan Pham,Sebastien Lefevre, Roberto Interdonato.

✉ Hugo Oliveira
oliveirahugo@dcc.ufmg.br

Caio Silva
caiosilva@dcc.ufmg.br

Gabriel L. S. Machado
gabriel.lucas@dcc.ufmg.br

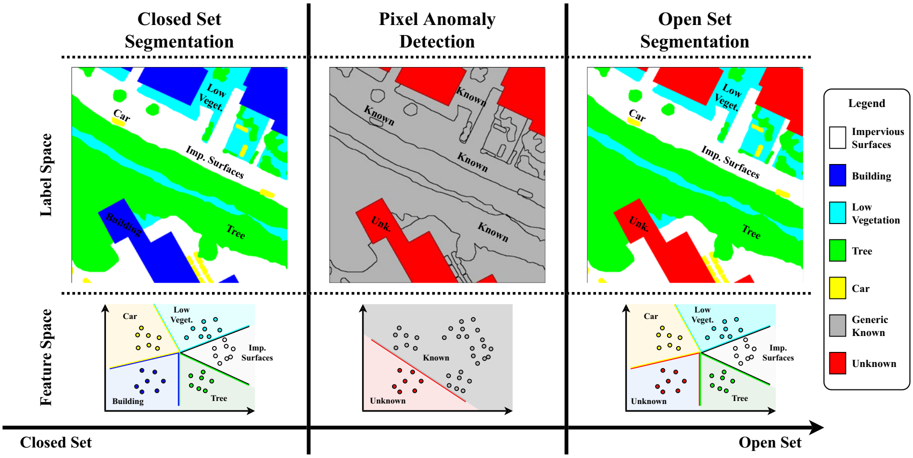
Keiller Nogueira
keiller.nogueira@dcc.ufmg.br

Jefersson A. dos Santos
jefersson@dcc.ufmg.br

[1] Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

slow OpenFCN, being able to run close to real-time. Experiments also indicate that OpenPCS is effective, robust and suitable for Open Set segmentation, being able to improve the recognition of unknown class pixels without reducing the accuracy on the known class pixels. We also tested the scenario of hiding multiple known classes to simulate multimodal unknowns, resulting in an even larger gap between OpenPCS/OpenIPCS and both SoftMax thresholding and OpenFCN, implying that gaussian modeling is more robust to settings with greater openness.

Graphic Abstract



Keywords Open set recognition · Semantic segmentation · Generative modeling · Deep learning · Remote sensing

1 Introduction

The development of new technologies for the acquisition of aerial images onboard satellites or aerial vehicles has made it possible to observe and study various phenomena on the Earth's surface, both on a small and large scale. A highly requested task, in this sense, is automated geographic mapping, which gives an easier and faster approach of monitoring cities, regions, countries, or entire continents. Automatic mapping of remote sensing images is typically modeled as a supervised classification task, commonly known as semantic segmentation, in which a model is first trained using labeled pixels and then used to classify other pixels in a new region. Commonly, this process is based on the Closed Set (or Closed World) assumption: it assumes that all training and testing pixels come from the same label space, e.g., train and test sets have the same set of classes. It is easy to notice that this assumption does not hold in real-world scenarios, mainly for Earth Observation applications, such as geographic mapping, given the huge size of the images and the (possible) elevated number of distinct objects (classes). In these scenarios, the model is likely to observe, during the prediction phase, samples of classes not seen during the training. In these cases, Closed Set semantic segmentation methods are error-prone to unknown classes

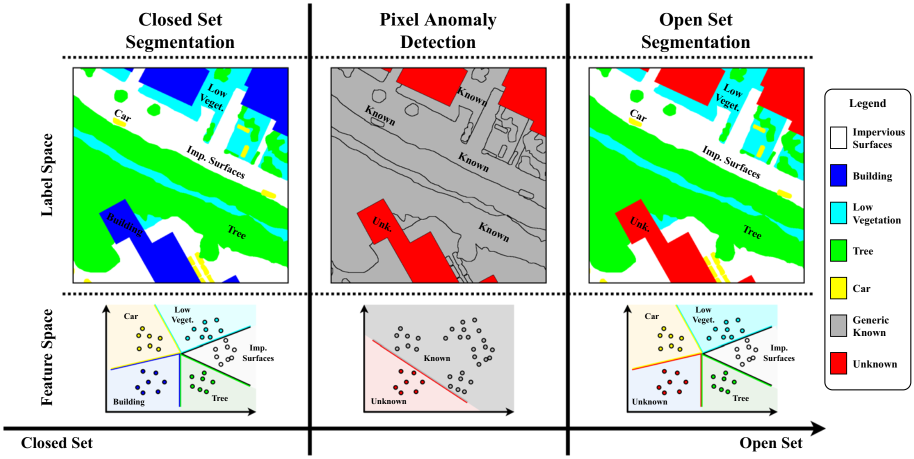


Fig. 1 Graphical depiction of the problem settings of Closed Set, Anomaly Detection and Open Set Recognition in dense labeling scenarios. The *x*-axis ranges from fully Closed Set (that is, assuming full knowledge of the world) on the left side to Open Set on the rightmost example. In the middle, there is the binary task of Pixel Anomaly Detection, wherein pixels are segmented either into KKCs or UUCs without discerning between distinct KKCs. We also depict the Label Space (Semantic Segmentation map) and a sample of some pixels' representation in a 2D manifold of the Feature Space separated by labels and decision boundaries for each class

given that they will wrongly recognize it as one of the known classes during the inference. This limits the use of such approaches to real-world Earth Observation applications, such as automated geographic mapping.

Towards solving this, Open Set Recognition (OSR) can be described as the set of algorithms that address the problem of identifying, during the inference phase, samples of unknown classes, e.g., instances of classes not seen during the training. Using the same definitions of Geng et al. (2020) and Scheirer et al. (2014), during the inference phase, an OSR system, such as an Open Set semantic segmentation approach, should be able to correctly classify the instances/pixels of classes employed during the training (Known Known Classes – KKCs) whereas recognizing the samples/pixels of classes **not** seen during training (Unknown Unknown Classes – UUCs).

By this definition, it is possible to state that the main difference between the closed and Open Set scenarios is related to the knowledge of the world, e.g., the knowledge of all possible classes. Specifically, while in the Closed Set scenario the methods should have full knowledge of the world, Open Set approaches must assume that they do not know all the possible classes during the training. Obviously, different approaches may have a distinct knowledge of the world depending on the problem. A visual example of this difference, in terms of knowledge of the world, is depicted in Fig. 1.

Technically, OSR is usually achieved via transductive learning (Li and Wechsler 2005) by trying to infer samples from UUCs using only the test data distribution. Another approach is to adapt Anomaly Detection techniques for OSR by using auxiliary data to try to learn a generative model via supervision, such as the Outlier Exposure (OE) (Hendrycks et al. 2018). However, neither of these approaches is ideal for Open Set semantic segmentation. This is because transduction requires continuous updates of a model in order to cope with new data, presenting an overhead that might be too expensive in most real-world applications, and the OE (Hendrycks et al. 2018) depends on the existence and availability of auxiliary Out-of-Distribution (OOD) data for training, which may not be possible or even useful in semantic segmentation. Therefore, an Open Set semantic segmentation algorithm must rely solely on inductive learning and use only the available training data and KKCs, while also addressing performance concerns, given the higher output dimensions of segmentation.

In this paper, we proposed two novel approaches for Open Set semantic segmentation of remote sensing image. We evaluate our methods and compare it with baselines by simulating Open Set situations in well known urban scenes such as Vaihingen and Potsdam datasets. It is important to mention that the concept of Open Set semantic segmentation is still very little explored in the literature. The first and unique work that introduces this problem is da Silva et al. (2020), which proposes a method based on OpenMax (Bendale and Boult 2016) for pixelwise classification, which have many limitations concerning both effectiveness and efficiency.

The main contributions of this work are:

1. The first fully convolutional methodology for semantic segmentation in RS imagery or otherwise adapted from OpenMax (Bendale and Boult 2016), explained in Sect. 3.1;
2. Proposal of a completely novel fully convolutional methodology for identifying UUCs in dense labeling scenarios using Principal Components from the internal feature space of the DNNs, further described in Sect. 3.2;
3. Definition of a benchmark evaluation protocol with standard threshold-dependent and threshold-independent metrics for testing OSR Semantic Segmentation tasks;
4. Extensive evaluations of architectures and thresholding for the proposed approaches in both Open and Closed Set baselines.

This paper is organized in sections, as follow: Sect. 2 presents the related work, other papers that try to solve the Open Set semantic segmentation problem; Sect. 3 describes the proposed methods, explaining in details how it works; Sect. 4 shows the setup used for this paper, the datasets evaluated and the metrics; Sect. 5 presents the obtained results from the experimental setup; Finally, Sect. 6 contains the conclusion over the obtained results and the proposed method.

2 Related work and background

Convolutional Neural Networks (CNNs) have become the backbone of visual recognition for the last decade. AlexNet (Krizhevsky et al. 2012) reintroduced image feature learning, allowing for better scalability than the first CNNs (e.g. LeNet LeCun et al. 1998) in order to perform inference over harder tasks (e.g. ImageNet Deng et al. 2009 and CIFAR Krizhevsky et al. 2009). AlexNet took advantage of larger convolutional kernels in the earlier layers and contained a total of eight layers, between convolutional and fully-connected ones. VGG (Simonyan and Zisserman 2014) simplified CNN architectures by using the same 3×3 kernels in all convolutions and max-poolings for downscaling. In contrast to VGG, the GoogleNet architecture (Szegedy et al. 2015)—also known as Inception—studied a diverse set of kernel sizes to enforce disentanglement in activations. Inception modules mix combinations of multiple kernel sizes and poolings. Both VGG and Inception allow for deeper networks with smaller convolutional kernels in each module, which proved to be more efficient than shallower networks with larger convolutions, at least up to around 20 layers.

It was observed that adding layers beyond a total of 20 was detrimental to the training of CNNs, as the gradients did not reach the earlier layers, effectively preventing their training. Residual Networks (ResNets) (He et al. 2016) based on residual identity functions that allow for shortcuts in the backpropagation were then introduced. ResNets with between 18 and 151 convolutional blocks were investigated by He et al. (2016), with little benefit being observed beyond that. With time, some tweaks were proposed to the standard ResNet architecture, with the more noteworthy ones being Wide ResNets (WRNs) (Zagoruyko and Komodakis 2016) and ResNeXt (Xie et al. 2017), which yielded considerable improvements to the traditional ResNets. However, ResNets were observed to be highly inefficient, as the activations of most convolutions throughout the network could be dropped with little-to-no effect on classification performance (Srivastava et al. 2015). Densely Connected Convolutional Networks (DenseNets) (Huang et al. 2017) improved on the parameter efficiency of ResNets by replacing the identity function by concatenation and adding bottleneck and transition layers, which lowered the parameter requirements of the architecture. Huang et al. (2017) tested in DenseNet a variation between 121 and 264 layers, observing them to be more efficient than ResNets in both parameter and flops, when similar error values were compared in the validation set.

The remainder of this section presents the main concepts to the understand of this work and the recent literature about semantic segmentation (Sect. 2.1) and Open Set recognition (Sect. 2.2).

2.1 Deep semantic segmentation

CNNs (Krizhevsky et al. 2012) are considered the state-of-the-art for sparse labeling tasks as object/scene classification due to their feature learning capabilities. The literature quickly learned to adapt CNNs for dense labeling tasks by patchwise training, using the label for the central pixel (Farabet et al. 2012; Pinheiro and Collobert 2014). However, Fully Convolutional Networks (FCNs) (Long et al. 2015) were shown to be considerably more efficient than patchwise training, providing the first end-to-end framework for semantic segmentation. Besides the accuracy and efficiency benefits of fully convolutional training, any traditional CNN architecture could be converted into an FCN by adding a bilinear interpolation to the activations and replacing the dense layers by convolutional ones, as shown in Fig. 2. This simple scheme also allowed for transfer learning from large labeled datasets as ImageNet (Deng et al. 2009) to relatively smaller semantic segmentation datasets as Pascal VOC (Everingham et al. 2015) and MS COCO (Lin et al. 2014). FCNs are also shown to benefit from skip connections that merge the high semantic level activations at the end of the network with the high spatial resolution information from earlier layers (Long et al. 2015).

More recently, several semantic segmentation methods have been proposed specifically to deal with different aspects of remote sensing images such as spatial constraints (Nogueira et al. 2016; Maggiori et al. 2017; Marmanis et al. 2018; Wang et al. 2017; Audebert et al. 2016; Nogueira et al. 2019) or non-RGB data (Kemker et al. 2018; Guiotte et al. 2020). Nogueira et al. (2016) use patchwise semantic segmentation in RS imaging for both urban and agricultural scenarios. Maggiori et al. (2017) proposed a multi-context method based on upsampled and concatenated features extracted from distinct layers of a fully convolutional network. In Marmanis et al. (2018), the authors proposed multi-context methods

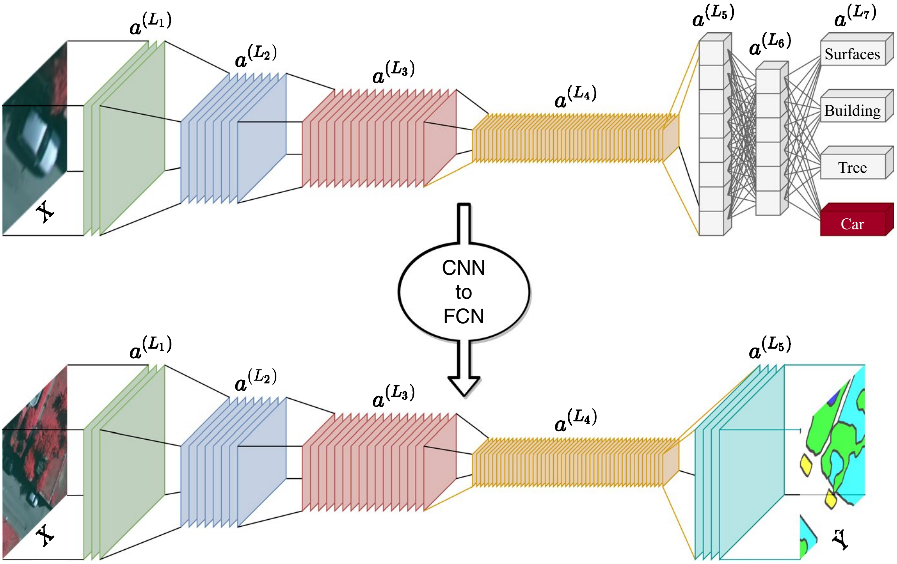


Fig. 2 Architecture example of a CNN for image classification and its equivalent FCN architecture with the same backbone for semantic segmentation. Activations from layer l are depicted as $a^{(l)}$ for all layers in the network (L_1 through L_7 for the CNN and L_1 through L_5 for the FCN). One should notice that in both architectures the input layer $a^{(L_1)}$ has the number of channels n^{ch} depending on the input data's number of channels (in the case of RGB images, $n^{ch} = 3$). In the CNN, the number of neurons in the output layer must match the number of KKCs (n^{KKCs}) in the data. Equivalently, in the FCN, the number of channels in the output layer n^{KKCs}, as suggested by the notation, depends on the number of KKCs of the dataset. In this example, $n^{KKCs} = 4$ as there are 4 known classes in this example

that combine boundary detection with deconvolution networks. In Audebert et al. (2016), the authors fine-tuned a deconvolutional network using 256 × 256 fixed size patches. To incorporate multi-context knowledge into the learning process, they proposed a multi-kernel technique at the last convolutional layer. Wang et al. (2017) proposed to extract features from distinct layers of the network to capture low- and high-level spatial information. In Kemker et al. (2018), the authors adapt state-of-the-art semantic segmentation approaches to work with multi-spectral images. Guiotte et al. (2020) proposed an aproach for semantic segmentation from LiDAR point clouds.

2.2 Open set recognition

The Open Set recognition problem was first introduced by Scheirer et al. (2012). They discussed about the notion of "openness", that occurs when we do not have knowledge of the entire set of possible classes during supervised training, and must account for unknowns during predicting phase. The first studies and applications involving Open Set recognition were adaptations of "shallow" methods that acted in the feature space of visual samples and consisted mainly of threshold-based or support-vector-based methods (Scheirer et al. 2012). More recent work has extended the concept to deep neural networks (Bendale and Boult 2016; Cardoso et al. 2017).

A recent survey by Geng et al. (2020) splits Open Set methods mainly between discriminative and generative approaches. Discriminative approaches usually use 1-vs-All Support

Vector Machines (SVMs) (Scheirer et al. 2012) in order to delineate the space between valid samples from the training classes and outliers—which ideally would identify UUCs. Meta-recognition can also be used to predict failures in visual learning tasks (Scheirer et al. 2012, 2014). Extreme Value Theory (EVT) is one of the most common modelings for meta-recognition using DNNs (Bendale and Boult 2016; Ge et al. 2017; Oza and Patel 2019) for classification.

Earlier deep OSR methods (Bendale and Boult 2016; Ge et al. 2017; Liang et al. 2017) aimed to incorporate the prediction of UUCs directly onto the prediction of the DNN output layer. Bendale and Boult (2016) and Ge et al. (2017) perform this by reweighting the output probabilities of the SoftMax activation to accomodate a UUC into the prediction during test time. This approach is known as OpenMax (Bendale and Boult 2016), and further developments to it have been proposed; for instance, aiding the computation of OpenMax with synthetic images from a Generative Adversarial Network (GAN) in G-OpenMax (Ge et al. 2017). OpenMax (Bendale and Boult 2016) will be further detailed in the methodology (Sect. 3.1), as it is the basis for one of the proposed methods in this paper.

Inspired by adversarial attacks (Goodfellow et al. 2014), Out-of-Distribution Detector for Neural Networks (ODIN) (Liang et al. 2017) insert small perturbations in the input image x in order to increase the separability in the SoftMax predictions between in- and out-of distribution data ($\mathcal{D}^{in}$ and $\mathcal{D}^{out}$, respectively). This separability allows ODIN to work similarly to OpenMax (Bendale and Boult 2016) and operate close to the label space, using a threshold over class probabilities to discern between KKCs and UUCs. The manuscript reports Area Under ROC curve metrics between 0.90 and 0.99 for CIFAR-10 (Krizhevsky et al. 2009) as $\mathcal{D}^{in}$ and between 0.70 and 0.85 for CIFAR-100 (Krizhevsky et al. 2009) as $\mathcal{D}^{in}$, depending on the $\mathcal{D}^{out}$ (e.g. TinyImageNet,[1] LSUN (Yu et al. 2015), random noise, etc). Extensive hyperparameter tuning experiments are reported in the paper as well. As will be further discussed in Sect. 3.2, restricting the information used for OSR to the activations in the last layers has severe limitations. Thus, modern methods have employed different strategies than simply thresholding the output probabilities to split samples into KKCs and UUCs.

A recent trend in both Anomaly Detection and OSR for deep image classification has been to incorporate input reconstruction error in supervised DNN training as a way to identify OOD samples (Yoshihashi et al. 2019; Oza and Patel 2019; Sun et al. 2020). These approaches fall under the branch of generative OSR, according to the taxonomy by Geng et al. (2020). Classification-Reconstruction learning for Open-Set Recognition (CROSR) (Yoshihashi et al. 2019) trains conjointly a supervised DNN for classification model ($x \rightarrow y$) and an AutoEncoder (AE) to encode the input (x) into an bottleneck embedding (z) and then decode it to reconstruct $\tilde{x}$. Conjoint training allows the DNN to optimize a compound loss function that minimizes both the classification and reconstruction errors. During the test phase, the reconstruction error magnitude between $err(x, \tilde{x}) = ||x - \tilde{x}||$ dictates if the input x is indeed from the predicted class $\hat{y}$ or an OOD sample.

Class Conditional AutoEncoder (C2AE) (Oza and Patel 2019), similarly to CROSR, uses the reconstruction error of the input ($||x - \tilde{x}||$) from an AE and EVT modeling to determine a threshold in order to discern between KKC and UUC samples. Following the same trend of thresholding a certain point in the density function of the reconstruction error from the inputs, Conditional Gaussian Distribution Learning (CGDL) (Sun et al.

[1] https://tiny-imagenet.herokuapp.com/.

2020) uses a Variational AutoEncoder (VAE) to model the bottleneck representation of the input images according to a vector of gaussian means μ and standard deviations σ in a lower-dimensional high semantic-level space. This modeling allows CGDL to unsupervisedly discriminate between KKCs and UUCs by thresholding the likelihood of the embedding z_i generated from a novel sample x_i pertaining to the multivariate gaussians $\mathcal{N}(z_i, \mu_k, \sigma_k^2)$, where k represents the predicted class for sample x_i.

2.2.1 Open set semantic segmentation

OpenPixel (da Silva et al. 2020) is based on patchwise training of classification DNNs for image classification (Nogueira et al. 2016) of RS images. The method builds on top of OpenMax (Bendale and Boult 2016) in order to recognize out-of-distribution pixels in urban scenarios. However, OpenPixel is highly inefficient during both training and test times due to the patchwise training using a customly built CNN.

As we have discussed in the introduction, to the best of our knowledge, the unique work in the literature that address the Open Set segmentation problem was proposed by da Silva et al. (2020). The authors introduce the concept and proposes two methods based on OpenMax (Bendale and Boult 2016) for pixelwise classification. Although promising, the approaches proposed in da Silva et al. (2020) have several limitations both in effectiveness and efficiency aspects. In this work we have extended the OpenPixel method proposed in da Silva et al. (2020) to be feasible in practical situations. We better explain the improvements and adaptations in Sect. 3.1.

As far as the authors are aware, there are no fully convolutional architectures for deep Open Set semantic segmentation in neither the remote sensing nor computer vision communities. Section 3 bridges this gap with the proposal of two approaches based on OpenMax (Bendale and Boult 2016) (Sect. 3.1) and Principal Component likelihood scoring (Tipping and Bishop 1999) (Sect. 3.2) in the domain of urban scene segmentation.

3 Proposed methods

This section details the two proposed methods presented in this work: (1) Open Fully Convolutional Network (OpenFCN), a fully convolutional extension of OpenMax (Bendale and Boult 2016; da Silva et al. 2020) for dense labeling tasks (Sect. 3.1); and (2) Open Principal Component Scoring (OpenPCS), a novel approach that uses feature-level information to fit multivariate gaussian distributions to a low-dimensional manifold of the data in order to obtain a score based on the data likelihood for identifying failures in recognition (Sect. 3.2).

3.1 OpenFCN

OpenFCN relies on traditional FCN-based architectures, which are normally composed of traditional CNN backbones with the inference layers replaced by bilinear interpolation and more convolutions. As the dense prediction is treated at training time as a classification task, the distinction between OpenFCN and FCN can be seen more clearly during validation and predicting. A meta-recognition module based on OpenMax (Bendale and Boult 2016) is added to the prediction procedure of traditional FCNs, as can be seen in Fig. 3.

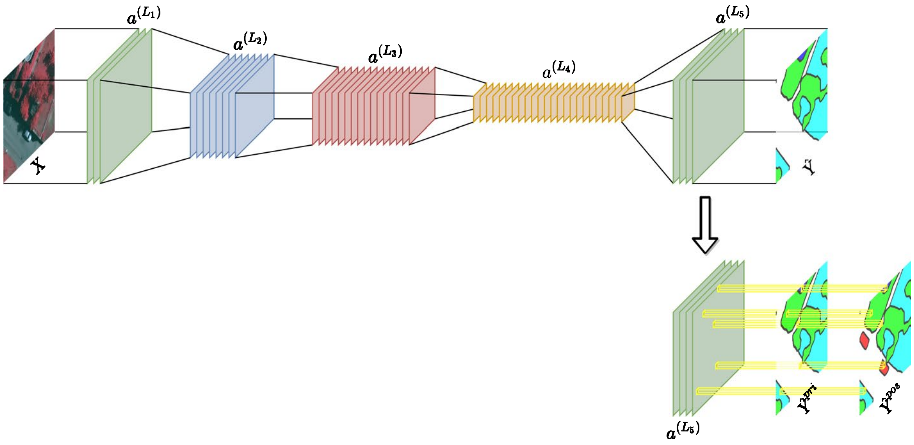


Fig. 3 OpenFCN scheme for Open Set Semantic Segmentation. During training, OpenFCN behaves like the traditional Closed Set FCN, with only Known Known Classes (KKCs) being fed to a supervised loss, such as Cross Entropy. OpenFCN differs from FCN only during validation and testing, when OpenMax is computed and the probabilities are thresholded in order to predict Unknown Unknown Classes (UUCs)

Let $\{X,\ Y\}$ be a paired set of image pixels and semantic labels from a dataset containing C KKCs. A deep model $\mathcal{M}$ can be trained in a stochastic manner by feeding samples X_i to a gradient descent optimizer as Kingma et al. (2014) with a loss function as Cross Entropy, given by:

$$\mathcal{L}_{CE}(Y, \hat{Y}^{pri}) = -Y \log(\hat{Y}^{pri}) - (1 - Y) \log(1 - \hat{Y}^{pri}) \quad (1)$$

This strategy ultimately yields an activation $a_i \in \mathbb{R}^C$ for each pixel i after the last layer. Thus, $\mathcal{M}$ can be seen as a function $\mathbb{R}^3 \rightarrow \mathbb{R}^C$ that converts the input space X_i into the prior SoftMax prediction σ_i for sample i. Obtaining the class prediction $\hat{Y}_i^{pri}$ can be easily done by finding the class with the larger probability in σ_i across all KKCs. Thus, OpenFCNs are trained using the exact same procedure as traditional FCNs for Closed Set semantic segmentation. Posteriori predictions $\hat{Y}^{pos}$ are only computed on validation and testing, as described in the following paragraphs.

OpenMax ($\mathcal{O}$) relaxes the traditional SoftMax requirement that prediction probabilities for KKCs must add to 1, introducing an extra class to the posterior prediction $\hat{Y}^{pos}$ to the prediction set for X. The function $\mathcal{O}(X, Y, \mathcal{M})$ is, therefore, able to reweight the SoftMax predictions, aggregating the probability of misclassifications due to UUCs. Following the protocol of OpenMax (Bendale and Boult 2016), during OpenFCN's validation procedure each KKC $c_k, k \in \{0, 1, \dots, C-1\}$ yields one Weibull distribution $\mathcal{W}_k$. $\mathcal{W}_k$ is fit to the deviations from the mean μ_k of $a^{(L_5)}$ according to some distance (e.g. euclidean, cosine, hybrid distances). Averages μ_k are computed in the validation set according only to the correctly classified pixels of class c_k.

Finally, in order to identify Out-of-Distribution (OOD) samples, quantiles from the Cumulative Distribution Function (CDF) for $\mathcal{W}_k$ are computed, with all pixels in the set $\hat{Y}^{pri}$ predicted to be from c_k and with less confidence than $\mathcal{T}_k$ being attributed to be from a UUC. Thus, the posterior OpenMax prediction $\hat{Y}_i^{pos}$ for a specific pixel X_i is given by:

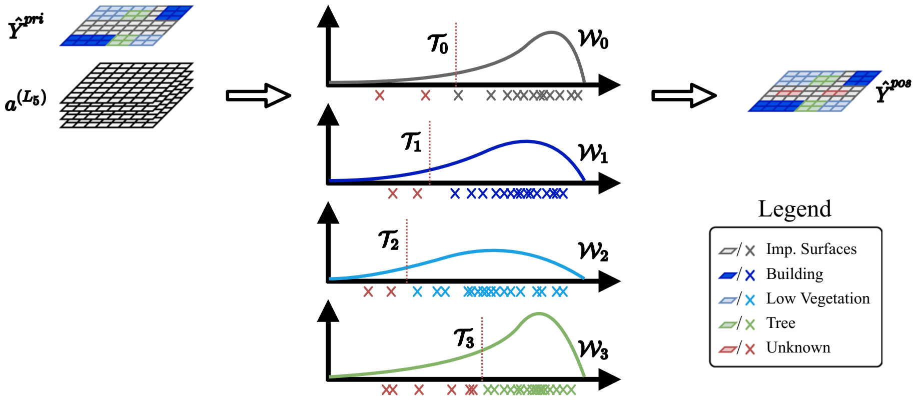


Fig. 4 OpenFCN's generative modeling for detecting UUC samples. A Weibull model $\mathcal{W}_c$ for KKC c is fit using the last layer's activations ($a^{(L_5)}$) from correctly classified samples of this class in the training set. According to the CDF for each class' Weibull distribution, a threshold $\mathcal{T}_c$ is set and samples that do not reach this threshold are classified as pertaining from a UUC

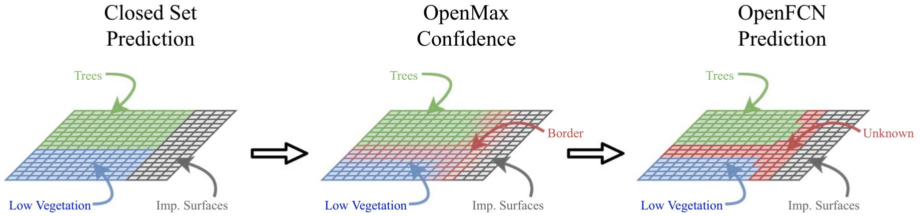


Fig. 5 Depiction of OpenFCN's prediction confidence degradation on object boundaries due to the use of information close to the label space. SoftMax and OpenMax probabilities on dense labeling tasks are naturally lower on boundary regions between objects from distinct classes

$$\hat{Y}_i^{pos} = \begin{cases} c_k, \text{ if } max(a_i^{(l)}) \geq \mathcal{T}_k \\ c_{unk}, \text{ if } max(a_i^{(l)}) < \mathcal{T}_k, \end{cases} \tag{2}$$

where l is the output layer in a DNN and c_{unk} is the identifier for the UUC in the Open Set scenario. This scheme is shown in Fig. 4.

Dense labeling tasks (e.g. instance/semantic segmentation or detection) inherently have higher dimensional outputs than sparse labeling tasks (e.g. classification or single-target regression). Dense predictions also present their particular set of difficulties, including how to handle boundaries between adjacent objects. Early in our experiments, we have seen a large number of border artifacts in OpenFCN predictions. As depicted in Fig. 5, adjacent areas between objects with distinct classes naturally yield class predictions with smaller certainties than the central pixels of these objects, resulting in warped Weibull distributions. This happens because the last layer in the network tries to model directly the label space distribution, retaining little-to-no information

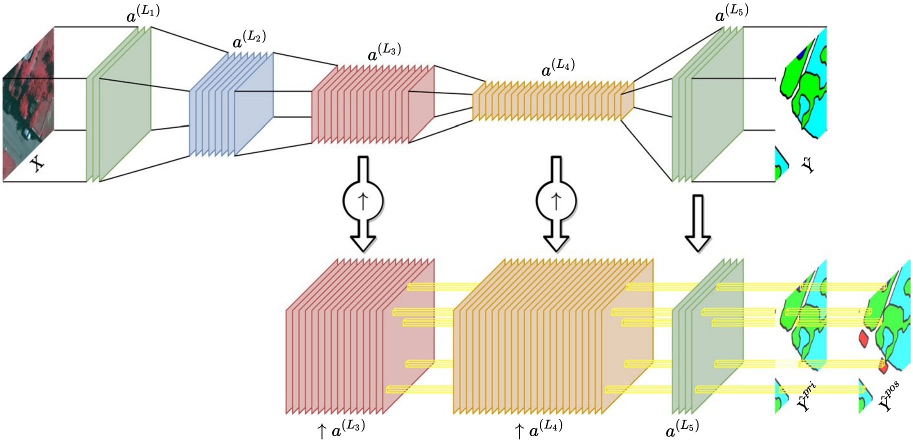


Fig. 6 OpenPCS general schematics. Subsequent activations ($a^{(L_1)}$, $a^{(L_2)}$, ..., $a^{(L_4)}$) from an FCN with a certain CNN backbone (as in Fig. 2) is shown. Activations from the last layers (e.g. $a^{(L_5)}$, $a^{(L_4)}$ and $a^{(L_3)}$, in this case) are concatenated to form column vectors for each predicted output pixel in $\hat{Y}^{pri}$. The prior prediction $\hat{Y}^{pri}$ is then processed according to the scheme shown in Fig. 7 using a generative model $\mathcal{G}$ in order to detect OOD pixels and, thus, classify them as unknowns

about the original input. Hence, activations from later layers in the network are more affected by object boundary uncertainties.

In order to mitigate this limitation of OpenFCNs in dense labeling tasks, we propose a completely novel method that is able to fuse information from low and high semantic level information into one single model. This method will be presented in Sect. 3.2.

3.2 OpenPCS

OpenPCS works similarly to CGDL (Sun et al. 2020), but with three important differences: (1) we fit gaussian priors using not only the input images X, but also the intermediary activations (e.g. $a^{(L_3)}$, $a^{(L_4)}$, $a^{(L_5)}$, etc); (2) we use a PCA instead of a VAE; and (3) the training is purely supervised and the low dimensional gaussians are fitted only during the validation phase, and not conjointly with the training, as CGDL. This was all done for simplicity and aiming to ease computational complexity, as open set semantic segmentation is a very recent field of research.

It is well known that the deeper a certain layer l is placed in a DNN, the closer to the label space the activation features $a^{(l)}$ are (Shwartz-Ziv and Tishby 2017). In fact, Shwartz-Ziv and Tishby (2017) argue that any supervised DNN can be seen as a Markov chain of sequential tensorial representations that gradually morph the information processed by the network from the input space (in the input layer) to the label space (in the output layer). Thus, by using only the last layer's activations to fit Weibull distributions to each KKC, OpenFCNs—and, by extension, OpenMax (Bendale and Boult 2016)—limit themselves to work with information close to the label space.

Unlike OpenFCN, OpenPCS takes into account feature maps from earlier layers, which encode information closer to the input space, and combine them with activations from the last layers, fusing low and high semantic level information. This can be seen in Figs. 6 and 7 in the form of the yellow columns shown in the lower part of the image. Each output pixel in $\hat{Y}^{pri}$ gets a correspondent activation vector (a^{*}) made by the

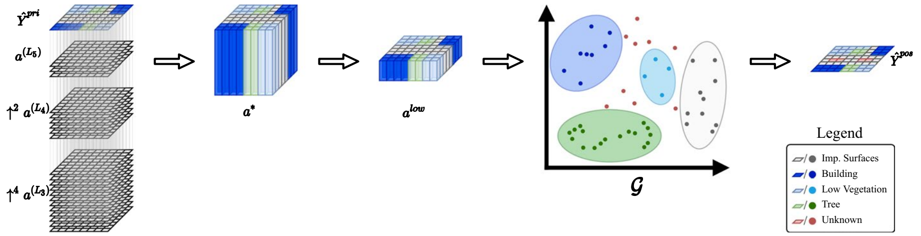


Fig. 7 Graphical scheme for the generative modeling in OpenPCS. In contrast to OpenFCN's Weibull fitting, OpenPCS uses a gaussian modeling in a low-dimensional representation a^{low} of the activations a^*. The Principal Components for each pixel are computed according to the concatenation of activations a^* from multiple layers (e.g. $a^{(L_3)}$, $a^{(L_4)}$, $a^{(L_5)}$, etc) for the corresponding region of each specific pixel. One multivariate gaussian is fit for each KKC and thresholds defined according to likelihoods from these models are used in order to identify OOD pixels

concatenation of earlier layer activations for the corresponding prediction map region in the channel axis. As earlier layers ($a^{(L_4)}$ and $a^{(L_3)}$) have lower spatial resolution due to the network's bottleneck, the activations from these layers are upsampled where the ↑ function is shown in order to match the dimensions of the input image and output prediction. In the example shown in Fig. 6, $a^* = (a^{(L_5)}, \uparrow^2 a^{(L_4)}, \uparrow^4 a^{(L_3)})$, so, for instance, if $a^{(L_5)} \in \mathbb{R}^{4\times MN}$, $\uparrow^2 a^{(L_4)} \in \mathbb{R}^{8\times MN}$ and $\uparrow^4 a^{(L_3)} \in \mathbb{R}^{16\times MN}$, then a^* has dimensionality $28 \times MN$. The concatenated feature vector for each input/output pixel of index i in this example would be, therefore, $a^*_i \in \mathbb{R}^{(28)}$.

Concatenating activations from multiple layers into a^* yields high dimensionality feature vectors for each pixel, as modern CNNs/FCNs easily output hundreds or thousands of activation channels from each layer. The large redundancy found in activation maps from convolutional layers (Srivastava et al. 2015; Huang et al. 2017) should also render a^* to be highly redundant. OpenPCS mitigates both problems by computing a lower dimension manifold a^{low} of each pixel's activation with Principal Component Analysis (PCA) previously to fitting a generative model ($\mathcal{G}$), as shown in Fig. 7. This approach grants two desirable properties to OpenPCS: (1) faster inference time during testing, as PCA implementations can be highly parallelized via vectorial operations and low-dimensional gaussian likelihood scoring can be computed in a fast manner; and (2) PCA feature selection guarantees that only the most important activation channels are used to compute a scoring function to detect OOD samples and, consequently, UUCs.

3.2.1 Open set scoring with principal components

Besides being purely a tensorial operation used for dimensionality reduction, PCA can be seen as a probability density estimator with gaussian priors. As described by Tipping and Bishop (1999), this allows PCA to be used as a generative model for novelty detection. In other words, the low dimensional Principal Components generated by PCA using a multivariate gaussian prior can yield likelihoods that allow for OOD recognition in new data.

PCA reduces dimensionality by finding latent variables composed of combinations of features in the original input space such that the reconstruction error when returning to this original space is minimized. This operation works by computing eigenvalues (λ) and eigenvectors (v) of a covariance matrix A (computed, in this work, using the Singular Value Decomposition on the input data). Those values can be calculated using the equation

$Av = \lambda v$. In the PCA procedure, the eigenvalues represent how impacting their correspondent eigenvector direction is to the data variability. Since reconstructions provided by PCA emphasize variation, in order to perform dimensionality reduction, the standard procedure is to project your data into an orthogonal basis composed by the eigenvectors that have the biggest correspondent eigenvalues, i.o.w., a subspace with the highest variance possible.

Exemplifying the application of PCA in OpenPCS, first we compute a covariance matrix using feature vector a^* – that is, a concatenation of activations from different layers in a DNN for a specific set of pixels. After that, we calculate the eigenvalues and eigenvectors of this matrix and select the set of *low* eigenvectors (v^{low}) that have the correspondent *low* largest eigenvalues (λ^{low}). Finally, we use those eigenvectors as a basis and project our input vectors (a^*) on it, resulting in $a^{low} = a^* \cdot v^{low}$.

3.2.2 OpenIPCS

OpenPCS is highly memory intensive, as fitting n^{KKC} PCA models using millions of pixels with feature vectors in the scale of hundreds or thousands of bins requires all this data to be stored in RAM. In practice, for training the traditional OpenPCS we used a subsample of randomly selected patches to fit the models using a reasonable amount of memory. Empirically, we found that 150 patches with 224 × 224 pixels each resulted in an acceptable trade-off between memory and enough training data for the computation of Principal Components for each class. Even with this approach, OpenPCS required between 20 and 30 GB of memory, depending on the dimensionality of the feature vectors a^*. In addition, PCA models trained using subrepresented classes in the imbalanced datasets (e.g. Cars) did not fit correctly due to a low number of correctly classified samples used for training the generative model. Ultimately, OpenPCS for large and highly imbalanced datasets can result in underperformance on the task of UUC identification due to the subsampling required by the large memory usage of the method.

In an effort to minimize this problem, in addition to the traditional OpenPCS we also evaluate an Incremental PCA (IPCA) for generative modeling and likelihood scoring. The full OSR segmentation methodology will be henceforth referred to as OpenIPCA. In contrast to the traditional offline training of the standard PCA, IPCA allows for mini-batch online training, which is highly useful for correctly computing the Principal Components in large datasets. OpenIPCA also allows for the model to be further updated in an efficient manner whenever new data might arise, not requiring a full retraining of the standard PCAs from scratch, but instead updating the IPCA model by feeding newly acquired patches. We emphasize that, apart from the incremental/online training that allows all the training dataset to be used in the computation of the Principal Components, all other aspects of the implementation of OpenIPCS were identical to the standard OpenPCS.

4 Experimental setup

This section describes the experimental setup for the evaluation of OpenFCN against open and Closed Set baselines. In order to encourage reproducibility, we provide details regarding the datasets and evaluation protocol (Sect. 4.1), fully convolutional architectures and baselines (Sect. 4.2). In addition, we are publicizing the code for OpenFCN, OpenPCS and OpenIPCS in this project's webpage[2] in a conscious effort to encourage reproducibility of

[2] http://patreo.dcc.ufmg.br/2020/03/20/openfcn/.

our results and follow-up research on OSR segmentation. OpenFCN's Weibull fitting and distance computation was based on libMR[3] and OpenPCS' Principal Components and likelihood scoring were computed using the scikit-learn[4] library.

4.1 Datasets and evaluation protocol

In order to validate the effectiveness of OpenFCN and OpenPCS on RS image segmentation, we used two urban scene 2D semantic labeling datasets from the International Society for Photogrammetry and Remote Sensing (ISPRS) with pixel-level labeling: **Vaihingen**[5] and **Potsdam**.[6] Vaihingen presents a spatial resolution of 5 cm/pixel with patches ranging from 2000–2500 pixels in each axis and Potsdam has 9 cm/pixel samples with 6000×6000 patches. Both datasets contain IR-R-G-B spectral channels paired with semantic maps divided into 6 KKCs: impervious surfaces, buildings, low vegetation, high vegetation, cars and miscellaneous; and 1 KUC: segmentation boundaries between objects.

The data also allows for 3D information to be incorporated into the models via Digital Surface Model (DSM) images, which is also made available in its normalized form (nDSM). In order to follow standard procedures in the RS literature (Sherrah 2016; Audebert et al. 2016; da Silva et al. 2020), we ignored the blue channel in our evaluation, limiting the experiments to the IR-R-G channels, in addition to the nDSM data, which was simply added as another channel to the inputs. Aiming to ease the computation complexity of our experiments, we also ignored the miscellaneous class, as it contains a rather small number of samples mainly on Vaihingen. For both datasets we used the standard procedure in the literature of training with most patches, while separating some specific patches for testing: 11, 15, 28, 30 and 34 for Vaihingen; and 2_11, 2_12, 4_10, 5_11, 6_7, 7_8 and 7_10 in the case of Potsdam.

Additionally to Vaihingen and Potsdam, we conducted experiments on the dataset of the 2018 IEEE GRSS Data Fusion Challenge[7], from now on referred to as **Houston** dataset. Like Vaihingen and Potsdam, Houston dataset contains RGB and DSM images with pixel resolutions of 5 cm and 50 cm, respectively. These previously discussed bands are paired with 48 hyperspectral bands in a 1 m resolution in the Houston dataset. For consistency with the experimental setups in Vaihingen and Potsdam datasets, we employed only the RGB and DSM bands in Houston. In order to match the differing RGB and DSM bands in Houston, we simply resize the RGB band to the lower resolution of the DSM using bilinear interpolation. While this is not the best use of the high resolution RGB information, the goal of this work is not to achieve state-of-the-art segmentation through some clever multiscale data fusion scheme, but instead to use the available data for testing OSR in segmentation scenarios.

Compared to Vaihingen and Potsdam, Houston allows for a stress test of the open set segmentation methods in a scenario with a considerably larger amount of known classes: unclassified, healthy grass, stressed grass, artificial turf, evergreen trees, deciduous trees, bare earth, water, residential buildings, non-residential buildings, roads, sidewalks,

[3] https://github.com/Vastlab/libMR.

[4] https://scikit-learn.org/stable/.

[5] http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html.

[6] http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-potsdam.html.

[7] https://ieee-dataport.org/open-access/2018-ieee-grss-data-fusion-challenge-%E2%80%93-fusion-multispectral-lidar-and-hyperspectral-data.

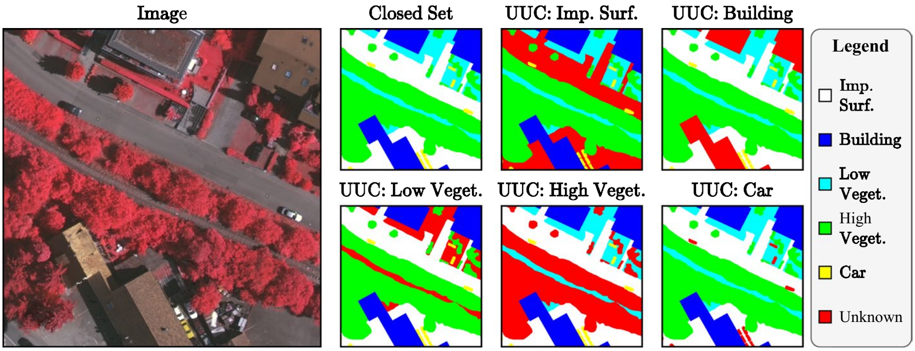


Fig. 8 LOCO procedure for Open Set evaluation. On the left one can see a mini-patch chosen from one of the larger patches that compose sample 11 of Vaihingen. On the right we present the Closed Set labels and different KKCs marked as Unknown using the LOCO protocol

crosswalks, major thoroughfares, highways, railways, paved parking lots, unpaved parking lots, cars, trains, and stadium seats. Among these classes, we explicitly ignore three ones in all experiments during both training and evaluation: unclassified, stadium seats, and water. The unclassified class is closer to a Known Unknown Class (KUC) than to a KKC, even containing unlabeled samples from other KKCs (e.g. cars and buildings) that were not labeled; while the latter two classes were removed due to extreme imbalance in the training set, making their samples unrepresentative for both the closed and open set models without the hyperspectral bands. Similarly to Vaihingen/Potsdam, experiments on Houston are conducted using the standard split between train and test of this dataset.

In order to simulate the Open Set environments on the Closed Set RS datasets, we employed a Leave-One-Class-Out (LOCO) protocol. LOCO works by selecting one class as UUC and ignoring its samples during training as shown in Fig. 8. This protocol allows for verifying both the overall performances of the segmentation architectures and class-by-class metrics, as there is a large class imbalance mainly when comparing classes as impervious surfaces and cars in the datasets. In order to not take into account the samples from the UUCs in each experiment, we only compute and backpropagate the loss according to the labels of the KKCs, with UUC pixels in the ground truths being skipped. This scheme guarantees that no information about the UUCs is fed to the model during the training procedure. Additionally, we test groupings semantically similar classes (e.g. vegetation, buildings, vehicles) as UUCs in experiments on Vaihingen, Potsdam and Houston.

One should notice that all samples in the data used in our experiments pertains to Closed Set Semantic Segmentation datasets, which allows for an objective evaluation of the effect of OSR segmentation on well-known classification metrics. Aiming to capture all nuances of training and testing Open Set scenarios, we propose a new standard set of test metrics that accounts for KKCs and UUCs, while also capturing the overall performance over all known and unknown classes. In order to evaluate the impact of adding the meta-recognition step to the inference procedure of KKCs, we computed the accuracy for known classes (Acc^K). The main metric adopted by da Silva et al. (2020) for evaluating the performance of OSR segmentation on UUCs is the binary accuracy between KKCs and UUCs (Acc^U), as in the case of Anomaly Detection. However, Acc^U is artificially inflated due to the large imbalance between the number of pixels from KKCs and UUC in most cases. We solved this by adding the precision of unknown detection (Pre^U) to the evaluation in order to observe the trade-off in performance in the binary meta-recognition task. The Kappa (κ) metric was used to evaluate the overall performance in both KKCs and UUC of the algorithms, as it is a common metric for the evaluation of semantic segmentation in remote sensing. All of the metrics above are, however, threshold dependent, making it hard to compare methods with distinct threshold ranges. As the thresholds do not represent equal statistical entities in SoftMax Thresholding, OpenFCN and OpenPCS, we define the thresholds for the methods based on preset values of True Positive Rate (TPR). This evaluation protocol also has the benefit of yielding information about the recall of the methods, as TPR and recall are equivalent in this case.

In order to evaluate the methods in a threshold-independent fashion, we complement our analysis with the Receiver Operating Characteristic (ROC) curve—which evaluates the whole range of thresholds for the different methods on a TPR vs. False Positive Rate (FPR) 2D plot—and its corresponding Area Under Curve (AUC). The ROC/AUC is a suitable measure since we want to evaluate the performance of the proposed methods and baselines across the whole threshold spectrum. Note that different choices of threshold values may result in an imbalance in the performance over closed and open classes, possibly skewing the results to one side or another. Thus, we believe that by evaluating the ROC/AUC one can assess if one algorithm is better than another for all possible thresholds.

Due to the large spatial resolution of Vaihingen and Potsdam patches, we sliced them into 224×224 mini-patches for training, in order to fit ImageNet's (Deng et al. 2009) traditional patch size, which is the most commonly expected input size for the CNNs used as backbones in our experiments. During training, we used data augmentation based on random cropping for patch selection, random flips in the horizontal and vertical dimensions, and random rotations by multiples of 90 degrees. In order to solve inconsistencies due to lack of context and to compensate for patch border uncertainties, we employed overlapping patches of 224×224 pixels during testing. We explicit that this procedure considerably increased the computational budget of our experiments mainly in the Potsdam dataset for OpenFCN due to inefficiencies and non-vectorized operations in libMR. A more thorough evaluation of time complexity can be found in Sect. 5.2.4.

4.2 Fully convolutional architectures and baselines

We compared several distinct fully convolutional architectures for OSR semantic segmentation, including five distinct CNN backbones: VGG-19 (Simonyan and Zisserman 2014), ResNet-50 (He et al. 2016), Wide ResNet-50-2 (Zagoruyko and Komodakis 2016) (WRN-50), ResNeXt-50-32x4d (Xie et al. 2017) (ResNeXt-50) and DenseNet-121 (Huang et al. 2017). All FCN implementations were based on the CNN source codes from torchvision[8].

In order to spare memory and computational resources, only a subset of layers in each FCN architecture is fed to the generative model. As the last convolutional layers after the FCN bilinear interpolation already possess high semantic level information about the prediction (classes), we added earlier layers to a^* aiming to insert more information about the raw data (pixels). Table 1 compiles all layers concatenated in a^* for pixel open set recognition, as well as information about which layers were fed via skip connections to the classification layers in order to improve the spatial resolution of FCN predictions.

[8] https://pytorch.org/docs/stable/torchvision/models.html.

Table 1 Layers used in the skip connections of all FCN architectures and passed to the generative gaussian modeling of OpenPCS/OpenIPCS

Backbone	Layers fed to skip connection	Layers fed to generative model
WRN-50 (Zagoruyko and Komodakis 2016)	Block^1	Block^2
	Block^4	Classifier^1
	–	Classifier^2
DenseNet-121 (Huang et al. 2017)	Block^2	Block^2
	Block^4	Classifier^1
	–	Classifier^2
ResNeXt-50 (Xie et al. 2017)	Block^1	Block^2
	Block^4	Classifier^1
	–	Classifier^2
ResNet-50 (He et al. 2016)	Block^1	Block^2
	Block^4	Classifier^1
	–	Classifier^2
VGG-19 (Simonyan and Zisserman 2014)	conv^2	conv^3
	conv^5	Classifier^1
	–	Classifier^2

conv^i layers indicate the index i of convolutional layer on VGG-19, while block^j represent the jth residual or densely connected block on DenseNet and ResNet variations. At last, classifier^k layers indicate that the kth layer after the FCN bilinear interpolation was fed to the generative model

Based on the methodology of OpenMax (Bendale and Boult 2016), we also compare OpenFCN and OpenPCS with traditional SoftMax followed by thresholding. SoftMax Thresholding ($\text{SoftMax}^{\mathcal{T}}$) follows the premise that least certain network predictions may be motivated by outlier classes seen in test time. In addition to these Open Set approaches, we evaluate the proposed methods in comparison with traditional Closed Set fully convolutional architectures for dense labeling using the LOCO protocol. Closed Set FCNs wrongly classify UUCs by design, forcing the pixel to be segmented according to the higher probability prediction among KKCs. One should notice that OpenPCS, OpenFCN, $\text{SoftMax}^{\mathcal{T}}$ and Closed Set FCNs were evaluated according to the same pretrained DNNs, differing only on test time. This protocol allows for direct comparisons in objective metrics disregarding performance variability due to the random nature of gradient descent optimization.

4.3 Cutoff values for OOD detection

All methods investigated in our experimental setup require a cutoff value $\mathcal{T}_k$ to delineate the boundary between KKC pixels from a class k and UUC pixels. Aiming to mimic a true OSR task, these thresholds were defined empirically according to the KKCs available during training. That is, no information about the UUCs should be fed to the process of choosing cutoff values from the network's confidence—in the case of $\text{SoftMax}^{\mathcal{T}}$ and OpenFCN—and to the multivariate gaussian likelihoods, for OpenPCS and OpenIPCS.

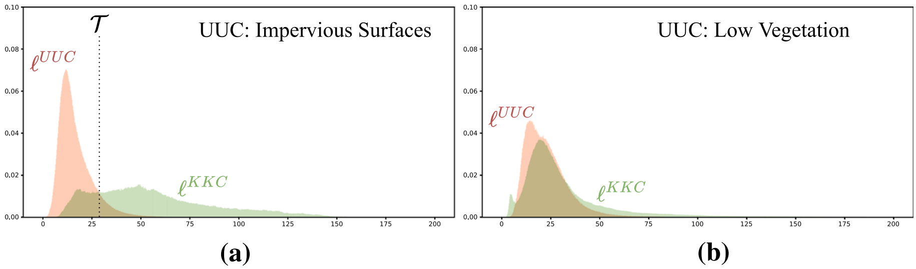


Fig. 9 Log-likelihoods for two scenarios in open set segmentation with distinct UUCs: Impervious Surfaces (**a**) and Low Vegetation (**b**). A much greater separation between the likelihoods of KKCs (ℓ^{KKC}) and the likelihoods of the UUCs (ℓ^{UUC}) can be seen in (**a**) than in (**b**), as the class Impervious Surfaces contains considerably less intra-class variability and similarity with other classes than Low Vegetation in the datasets described in Sect. 4.1. One should notice that worse separations between these distributions of likelihoods results in worse overall OSR performance

OpenFCN and SoftMax$^{\mathcal{T}}$ follow the methodology of OpenMax (Bendale and Boult 2016), with the cutoffs being cross-validated experimentally. OpenPixel (da Silva et al. 2020) also performed this analysis across all UUCs in Vaihingen and Potsdam, finding 0.7 to be a suitable value for thresholding between KKCs and UUCs. In other words, all pixels predicted to be from a certain class with confidence smaller than 0.7 should be considered as OOD, while predictions with confidence of 0.7 or above keep their predicted KKC.

The same cutoff idea is employed in OpenPCS and OpenIPCS, albeit with a crucial distinction: while all SoftMax$^{\mathcal{T}}$ and OpenMax predictions are bounded to the interval [0, 1], gaussian likelihood ranges vary depending on the number of Principal Components of PCA. Thus, in order to consistently compute cutoff values that work on any architecture and with a distinct number of Principal Components, we define threshold values based on TPR quantiles. One last remark about OpenIPCS is that, even if iterative/online training allows it to be fitted on the whole training set, we still used random patch subsampling in order to shorten the training time for the generative part of OpenIPCS.

Additionally to using the log-likelihoods for cutoff in the case of OpenPCS and OpenIPCS, we also observed in our exploratory experiments that the reconstruction error of the encoded a^* tensors produces similar results. Thus, an alternative with considerably lower computational cost to the score-based thresholding can be to perform the thresholding on the reconstruction error, whilst keeping the rest of the pipeline as is. This early finding is, in fact, aligned to other recent works in the OSR literature (Oza and Patel 2019; Yoshihashi et al. 2019). We highlight that the experiments that support this claim had a limited and exploratory nature and that a much more thorough assessment should be conducted to validate this hypothesis. Figure 9 contains the Log-likelihood distributions overall classes for the models in which the UUC were Impervious Surfaces and Low Vegetation, respectively. Those models used a fcndensenet121 as a backbone on the Vaihingen dataset. In (a), we can see a clear separation between the distributions of KKC and UUC, unlike (b), in which most of them overlap each other. It is important to mention that for acceptable discrimination between KKC and UUC, it is desirable that their Log-likelihood distributions are as separate as possible.

Table 2 Average AUC values in the LOCO protocol for each evaluated FCN backbone in the **Vaihingen** dataset

Backbone	SoftMax$^{\mathcal{T}}$	OpenFCN	OpenPCS
WRN-50 (Zagoruyko and Komodakis 2016)	0.68 ± 0.09	0.69 ± 0.10	**0.82 ± 0.12**
DenseNet-121 (Huang et al. 2017)	0.68 ± 0.08	0.68 ± 0.09	0.79 ± 0.12
ResNeXt-50 (Xie et al. 2017)	0.64 ± 0.06	0.65 ± 0.04	0.72 ± 0.12
ResNet-50 (He et al. 2016)	**0.69 ± 0.12**	**0.71 ± 0.12**	0.69 ± 0.16
VGG-19 (Simonyan and Zisserman 2014)	0.67 ± 0.06	0.68 ± 0.07	0.77 ± 0.12

Bold values indicate the best overall results including all methods

Table 3 Average AUC values in the LOCO protocol for **DenseNet-121** and **WRN-50** backbones in the **Potsdam** dataset

Backbone	SoftMax$^{\mathcal{T}}$	OpenFCN	OpenPCS	OpenIPCS
WRN	0.62 ± 0.06	0.62 ± 0.07	**0.73 ± 0.13**	**0.73 ± 0.13**
DenseNet	0.61 ± 0.11	0.62 ± 0.12	0.66 ± 0.25	0.71 ± 0.23

Bold values indicate the best overall results including all methods

5 Results and discussion

In this section we present and discuss the obtained results. We design our experimental evaluation in order to cover all possible aspects desirable to know in an OSR task. First, an analysis is performed in Sect. 5.1 in order to define the most suitable architectures for the proposed approaches. This section contains both overall and per-class threshold-independent analysis (AUC-ROC). Section 5.2 compares the proposed methods with the baseline in terms of quantitative metrics, qualitative segmentation maps (Sect. 5.2.3), and runtime performance (Sect. 5.2.4). At last, Sect. 5.3 presents our analysis of scenarios with a larger proportion of UUCs to KKCs, evaluating the performance of the proposed methods and baselines in settings with greater openness (Scheirer et al. 2012).

Note that, for the sake of simplicity and clarity, only the most relevant results were reported in this Section. For a full report of the results, please, check the supplementary material on this project's webpage linked in Sect. 4.

5.1 Architecture analysis

In this Section, we analyze the networks in order to define the most suitable architectures for the proposed techniques. To perform this evaluation, we employed the AUC, a threshold-independent performance measurement that allows comparisons between methods without resorting to (potentially) arbitrary thresholds. Furthermore, it is important to emphasize that we performed this analysis by using only the Vaihingen dataset. This is due to the fact that the Potsdam dataset is very similar to Vaihingen and, therefore, analysis and decisions made over the latter dataset are also applicable to the former one.

Table 4 AUC values for each UUC of the **Vaihingen** dataset

Backbone	Methods	UUC: Imp. Surf.	UUC: Building	UUC: Low Veg.	UUC: High Veg.	UUC: Car
DenseNet	**SoftMax$^{\mathcal{T}}$**	.78 ± .03	.63 ± .05	**.73 ± .05**	.67 ± .06	.58 ± .06
	OpenFCN	.81 ± .03	.64 ± .05	.72 ± .05	.66 ± .06	.58 ± .05
	OpenPCS	**.84 ± .03** †	**.94 ± .01** †	.70 ± .08	**.81 ± .07** †	**.65 ± .06** †
WRN	**SoftMax$^{\mathcal{T}}$**	.80 ± .03	.64 ± .04	.75 ± .05	**.63 ± .06**	.58 ± .05
	OpenFCN	.83 ± .03	.67 ± .03	.74 ± .05	.62 ± .06	.58 ± .05
	OpenPCS	**.87 ± .03** †	**.94 ± .02** †	**.77 ± .04**	.63 ± .09	**.87 ± .03** †
ResNeXt	**SoftMax$^{\mathcal{T}}$**	.66 ± .06	.55 ± .07	**.71 ± .07**	.66 ± .04	.65 ± .04
	OpenFCN	.67 ± .06	.59 ± .05	.71 ± .07	.64 ± .03	.63 ± .04
	OpenPCS	**.88 ± .03** †	**.81 ± .07** †	.58 ± .08	**.69 ± .09**	**.65 ± .05**
ResNet	**SoftMax$^{\mathcal{T}}$**	.83 ± .03	.62 ± .06	**.81 ± .05**	.60 ± .05	.60 ± .05
	OpenFCN	**.86 ± .02**	.64 ± .05	.80 ± .05	**.61 ± .05**	.62 ± .05
	OpenPCS	.75 ± .03	**.92 ± .02** †	.65 ± .07	.50 ± .07	**.63 ± .05**
VGG	**SoftMax$^{\mathcal{T}}$**	.72 ± .02	.59 ± .04	.74 ± .04	**.66 ± .05**	.66 ± .05
	OpenFCN	.74 ± .03	.59 ± .04	**.74 ± .03**	.65 ± .05	.65 ± .04
	OpenPCS	**.82 ± .04** †	**.90 ± .02** †	.71 ± .06	.60 ± .07	**.81 ± .04** †

Bold values indicate the best results when comparing SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS for each network and UUC. Results followed by † represent a significant increase in performance when using OpenPCS compared to both OpenFCN and SoftMax$^{\mathcal{T}}$. The significance of these results was assessed using a paired one-tailed t-Student hypothesis test with $p = 0.05$

Table 2 presents the average AUC results over all runs of the LOCO protocol. For SoftMax$^{\mathcal{T}}$ and OpenFCN, the ResNet-50 architecture (He et al. 2016) produced the best outcomes, followed closely by WRN-50 (Zagoruyko and Komodakis 2016) and DenseNet-121 (Huang et al. 2017). OpenPCS showed better results with WRN-50 (Zagoruyko and Komodakis 2016), followed closely by DenseNet-121 (Huang et al. 2017) and VGG (Simonyan and Zisserman 2014). Since WRN (Zagoruyko and Komodakis 2016) and DenseNet (Huang et al. 2017) were the most stable networks, producing good results in all approaches, they were selected and used in further experiments shown in the following sections.

Based on the best results for Vaihingen, we also report the threshold independent average AUC analysis over all UUCs on Potsdam. Table 3 shows the results for DenseNet-121 and WRN-50 on the considerably larger Potsdam dataset, wherein extensive architectural comparisons were not feasible.

One can easily see that in both tables OpenPCS and OpenIPCS present better AUC results than OpenFCN and SoftMax$^{\mathcal{T}}$, while the distinction between OpenFCN and SoftMax$^{\mathcal{T}}$ is often rather small or nonexistent. These evaluations are averages and standard deviations computed across all classes in the LOCO protocol, therefore, apart from a raw evaluation of the overall performance of each method, the previously mentioned.

Table 5 AUC values for each UUC of the **Potsdam** dataset

Backbone	Methods	UUC: Imp. Surf.	UUC: Building	UUC: Low Veg.	UUC: High Veg.	UUC: Car
DenseNet	**SoftMax$^{\mathcal{T}}$**	.66 ± .09	.49 ± .07	**.53 ± .13**	**.64 ± .08**	.76 ± .02
	OpenFCN	.66 ± .09	.50 ± .06	.51 ± .13	.62 ± .07	.80 ± .03
	OpenPCS	.83 ± .06†	.87 ± .13†	.28 ± .03	.54 ± .11	.77 ± .04
	OpenIPCS	**.84 ± .07** †	**.88 ± .14** †	.41 ± .07	.54 ± .09	**.91 ± .04** †
WRN	**SoftMax$^{\mathcal{T}}$**	.68 ± .06	.53 ± .06	.60 ± .12	**.62 ± .07**	.66 ± .10
	OpenFCN	**.69 ± .05**	.54 ± .05	.58 ± .12	.60 ± .07	.69 ± .09
	OpenPCS	.66 ± .08	**.86 ± .10** †	.71 ± .08†	.57 ± .10	**.86 ± .04** †
	OpenIPCS	.64 ± .09	**.86 ± .10** †	**.76 ± .07** †	.55 ± .10	.83 ± .04†

Bold values indicate the best results when comparing SoftMax$^{\mathcal{T}}$, OpenFCN, OpenPCS and OpenIPCS for each network and UUC. Results followed by † represent a significant increase in performance when using OpenPCS/OpenIPCS compared to both OpenFCN and SoftMax$^{\mathcal{T}}$. The significance of these results was assessed using a paired one-tailed t-Student hypothesis test with $p = 0.05$

5.2 Baseline comparison

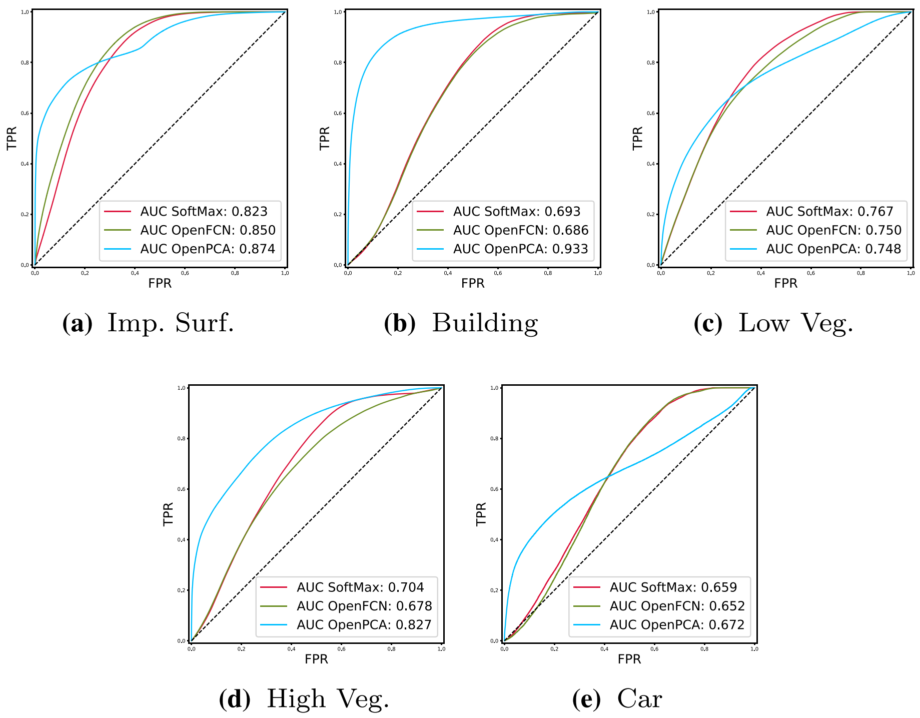
Based on the analysis performed in previous sections, we have conducted several experiments to investigate the effectiveness of the proposed methods using the most promising architectures: WRN-50 and DenseNet-121. We further investigate the performance on both KKCs and UUCs of the proposed methods (OpenFCN and OpenPCS), and baseline (SoftMax$^{\mathcal{T}}$) using the threshold-dependent metrics described in Sect. 4.1. This section also contains qualitative segmentation predictions taken from the experiments.

5.2.1 Per-class analysis

Table 4 presents the results, in terms of AUC per UUC, for the Vaihingen dataset. As can be seen, the OpenPCS obtained most of the best results when evaluating the UUCs and networks independently (bold values). When considering all the networks together and only the UUCs independently (values with †), OpenPCS produced the best outcomes for four UUCs, while OpenFCN yielded the best result for one UUC.

Analysis of Tables 4 and 5 reveal that OpenPCS excels in almost all architectures and UUCs in the LOCO protocol. The overall best results for four out of five UUCs in the LOCO protocol (*Impervious Surfaces*, *Building*, *High Vegetation* and *Car*) pertain to OpenPCS. Most AUCs for OpenPCS in these UUCs achieve values larger than 0.80, peaking at 0.94 in the class *Building*. It is also notable that OpenPCS achieved a significant increase in AUC performance for most of the cases. Not coincidentally, these classes are also the most semantically and visually distinct from the other ones. More detailed insights regarding the nature of the superiority of OpenPCS over the other methods will be discussed further in the next sections.

The outlier in this pattern is the class *Low Vegetation*, as its samples present a much larger intraclass variability than the others, encompassing sidewalks, bushes, grass fields in the interior and exterior of buildings, gardens, backyards, some sections of parking lots and even train tracks partially covered with vegetation. Due to this higher variation in



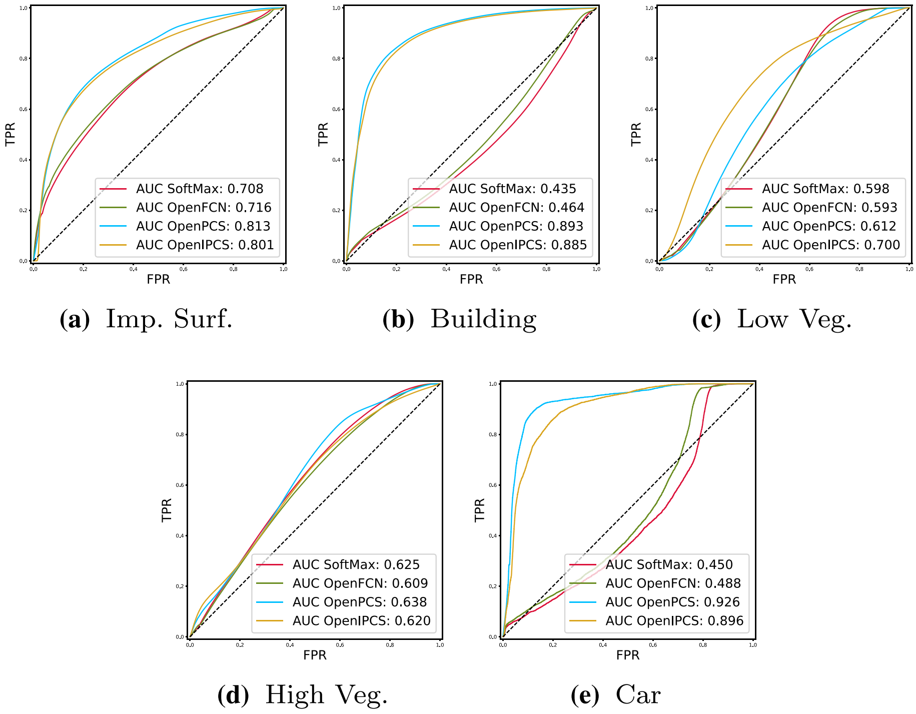
(a) Imp. Surf. **(b)** Building **(c)** Low Veg.

(d) High Veg. **(e)** Car

Fig. 10 ROC curves for $\text{SoftMax}^{\mathcal{T}}$, OpenFCN and OpenPCS on sample 11 from **Vaihingen** using a **DenseNet-121** backbone

samples, the lower-dimensional multivariate gaussian fitting of OpenPCS was not able to properly map the whole variation in the data, as was the simple Weibull fitting of OpenFCN. Thus, $\text{SoftMax}^{\mathcal{T}}$ overcame both proposed methods and achieved a peak AUC of 0.81 on ResNet-50. Even though the best result in the UUC *High Vegetation* was achieved by OpenPCS, the peak AUC for this class was also 0.81, this time using a DenseNet-121 backbone. The main visual distinction between *High Vegetation* and *Low Vegetation* is not in the visible spectrum, though. While textures for most patches of both classes are rather similar when taking into account grass fields and bushes for *Low Vegetation*, the DSM data is more visually distinct than the IRRG data, as the peak altitude for tree tops are above the mostly plain areas of *Low Vegetation*. This implies that using the inputs channels (IRRG and DSM) coupled with the middle and later activations of the DNNs on the computation of PCAs for OpenPCS could have improved the performance of both classes. However, more research on this must be done to either confirm or deny this hypothesis.

To better demonstrate the differences in performance between the techniques, Figs. 10 and 11 present the ROC curves for samples from the Vaihingen and Potsdam datasets, respectively. It is important to emphasize that each figure was created using a binary mask that separates KKCs from the evaluated UUC. Through Fig. 10, it is possible to observe that, in general, OpenPCS produces better results with lower FPRs for the same TPRs. It is also possible to visually assess that the detection capabilities of the methods for UUC *Impervious Surfaces* were considerably superior to other classes. The ROCs for *Buildings*,



(a) Imp. Surf.

(b) Building

(c) Low Veg.

(d) High Veg.

(e) Car

Fig. 11 ROC curves for SoftMax$^{\mathcal{T}}$, OpenFCN, OpenPCS and OpenIPCS on sample 6_7 from **Potsdam** using a **WRN-50** backbone

Table 6 Results for different unknown TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS from an FCN with **DenseNet-121** backbone in **Vaihingen**

TPR	SoftMax$^{\mathcal{T}}$			OpenFCN			OpenPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUC: Impervious Surfaces									
Closed	**.85±.02**	.00±.00	.52±.04	**.85±.02**	.00±.00	.52±.04	**.85±.02**	.00±.00	.52±.04
0.10	.83±.03	.47±.11	.53±.04	**.85±.02**	.67±.12	.55±.04	**.85±.02**	**.97±.02**	.55±.04
0.30	.79±.03	.48±.10	.56±.02	.81±.03	.56±.11	.58±.02	**.85±.03**	.91±.07	.61±.03
0.50	.74±.03	.47±.11	.57±.02	.76±.03	.52±.11	.59±.02	.82±.04	.77±.14	**.65±.03**
0.70	.67±.04	.45±.10	.57±.04	.70±.04	.49±.11	.59±.03	.71±.09	.57±.15	.61±.09
0.90	.54±.05	.41±.09	.51±.06	.58±.04	.43±.10	.54±.05	.38±.06	.35±.08	.35±.06
UUC: Building									
Closed	**.83±.03**	.00±.00	.50±.06	**.83±.03**	.00±.00	.50±.06	**.83±.03**	.00±.00	.50±.06
0.10	.77±.03	.26±.06	.48±.05	.78±.03	.28±.07	.48±.05	.82±.03	**.98±.02**	.53±.05
0.30	.70±.03	.32±.06	.47±.04	.71±.03	.32±.06	.47±.04	.82±.03	.96±.02	.59±.05
0.50	.62±.03	.33±.06	.45±.04	.62±.03	.34±.06	.46±.04	.82±.03	.94±.03	.65±.04
0.70	.50±.06	.33±.06	.40±.06	.51±.06	.33±.06	.41±.06	.80±.03	.87±.06	**.70±.03**
0.90	.30±.13	.30±.06	.26±.13	.29±.15	.30±.06	.25±.14	.71±.04	.64±.07	.68±.03

Bold values indicate the best overall Acc^K, Pre^U and κ results for a certain UUC. "Closed" rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation

High Vegetation and *Car* in Vaihingen also highlight the superiority of OpenPCS when compared to the other approaches.

Figure 11 provides distinct insights, as Potsdam is a much harder dataset than Vaihingen. Figure 11b, e show that OpenPCS and its online/incremental implementation (OpenIPCS) behaves dramatically better than both OpenFCN and SoftMax$^{\mathcal{T}}$. OpenFCN and SoftMax$^{\mathcal{T}}$ stay mostly close to randomly selecting pixels in the images, which is delineated in the figures by the 45 degree dashed line in the plot. For *Building* and *Car*, the ROCs and AUCs show that SoftMax$^{\mathcal{T}}$ and OpenFCN achieved worse than random results, while OpenPCS and OpenIPCS reached close to or above 0.90 in AUC.

Analysis of the per-class ROCs and Tables reveal that OpenPCS (and its variant OpenIPCS) behave considerably better than both SoftMax$^{\mathcal{T}}$ and OpenFCN in the binary task of Pixel Anomaly Detection for most UUCs. One can clearly see the distinction of the distinct methods in Pixel Anomaly Detection performance between UUCs in Figs. 10 and 11. OpenPCS excels by a large margin on some UUCS, especially *Building* and *High Vegetation* on Vaihingen, presenting lower FPR values than OpenFCN and SoftMax$^{\mathcal{T}}$ by almost any TPR threshold in the ROC. The evaluation on the other classes is more nuanced, as the plots cross at some points. OpenPCS tends to perform better on the lower end of FPR detections, providing the first evidence that OSR using Principal Components lessens the effect of UUC segmentation on the KKC classification performance, as smaller FPRs indicate that a smaller number of KKC pixels were misclassified as UUCs. This relation between UUCs and their distinction in performance on both KKCs and UUCs will further explored in the next Section.

5.2.2 Performance for KKCs and UUCs

OSR tasks are inherently multi-objective, as Open Set algorithms must be able to successfully discern UUCs from KKCs, while still being able to correctly classify samples from KKCs. Table 6 presents the results, in terms of accuracy for KKCs (Acc^{K}), Precision for UUC (Pre^{U}), and κ, for the Vaihingen dataset using a FCN with DenseNet-121 backbone. For simplicity, this table only reports the results for UCCs *Impervious Surfaces* and *Building*. However, detailed results, for the Vaihingen and Potsdam datasets, can be found in Appendices A and B.

Through the table, it possible to observe that the comparison of TPRs larger than one—that is, thresholds that allow for OSR—with their Closed Set counterparts (TPR = 0) using κ reveals that, in many cases, assuming openness **improves** object recognition in scenarios where full knowledge of the world is not possible. Specifically, for *Impervious Surfaces*, we obtained gains of 0.13 in terms of κ (0.52 when considering the Closed Set versus 0.65 when using OpenPCS with 0.50 TPR). For the UCC *Building*, the OpenPCS improved κ from 0.50 (in the Closed set) to 0.70 using 0.70 TPR. These gains mean that, for both UUCs, OpenPCS was able to maintain the accuracy of KKCs (Acc^{k}) while considerably increasing the precision of the UUCs (Pre^{U}).

On the other hand, despite also improving (in a smaller magnitude) the results in terms of κ, SoftMax$^{\mathcal{T}}$ and OpenFCN were not as effective at preserving the Acc^{k} while increasing $Prec^{U}$. This means that pixels that were correctly classified by the original DNNs were cast as UUCs by the OSR post-processing resulting in a high FPR.

Aside from these UUCs, similar conclusions can be drawn from other ones, such as *High Vegetation* and *Car*. Furthermore, similar outcomes were obtained for the Potsdam dataset. As aforementioned, a detailed discussion of all obtained results, for the Vaihingen

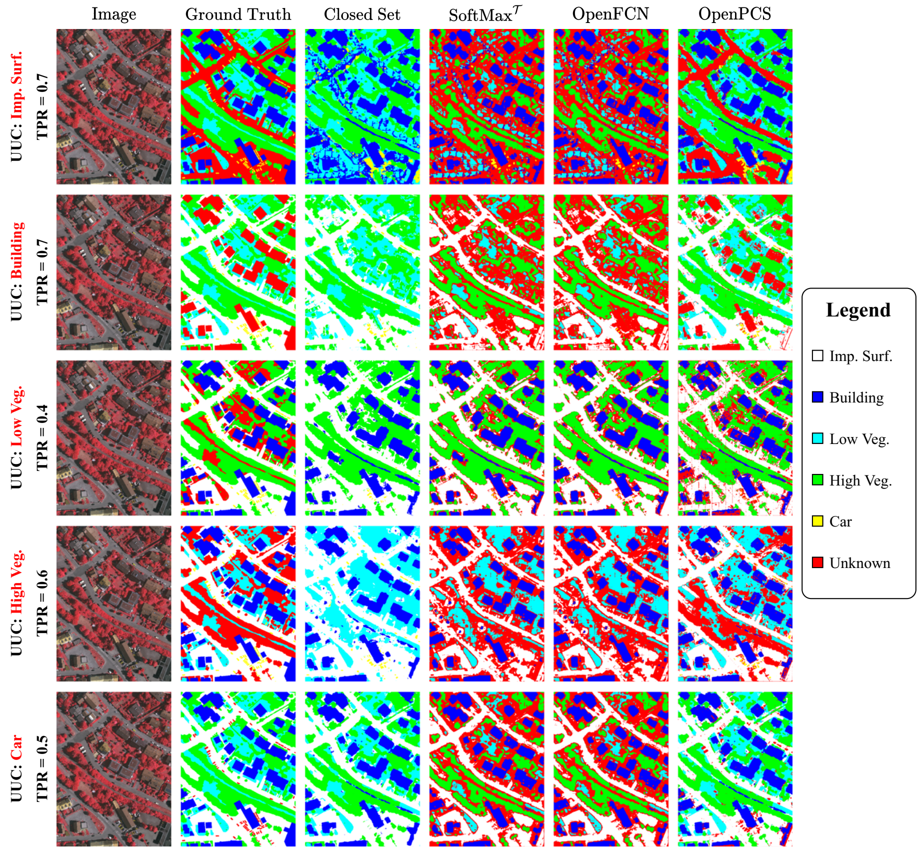


Fig. 12 Some visual result samples obtained for the **Vaihingen** dataset according to distinct UUCs in the LOCO protocol and distinct TPR thresholds for SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS

and Potsdam datasets, can be found in Appendices A and B. Overall, the results allow us to conclude that OpenPCS is more effective to perform open set semantic segmentation when compared to the other approaches. This difference can be better observed in the qualitative results presented in Sect. 5.2.3.

5.2.3 Qualitative analysis

Figures 12 and 13 present some visual examples of the results generated by the FCN with DenseNet-121 backbone in the Vaihingen and Potsdam datasets respectively. Qualitatively, the effectiveness of OpenPCS to distinguish KKCs from UUCs is even clearer. As can be observed, OpenPCS is capable of producing more accurate UUC identification for both KKCs and UUCs, when compared to the ground-truth label. On the other hand, outcomes generated by the SoftMax$^{\mathcal{T}}$ and OpenFCN are very similar to each other, but not very alike to the ground-truth label, mainly for the UUCs. This

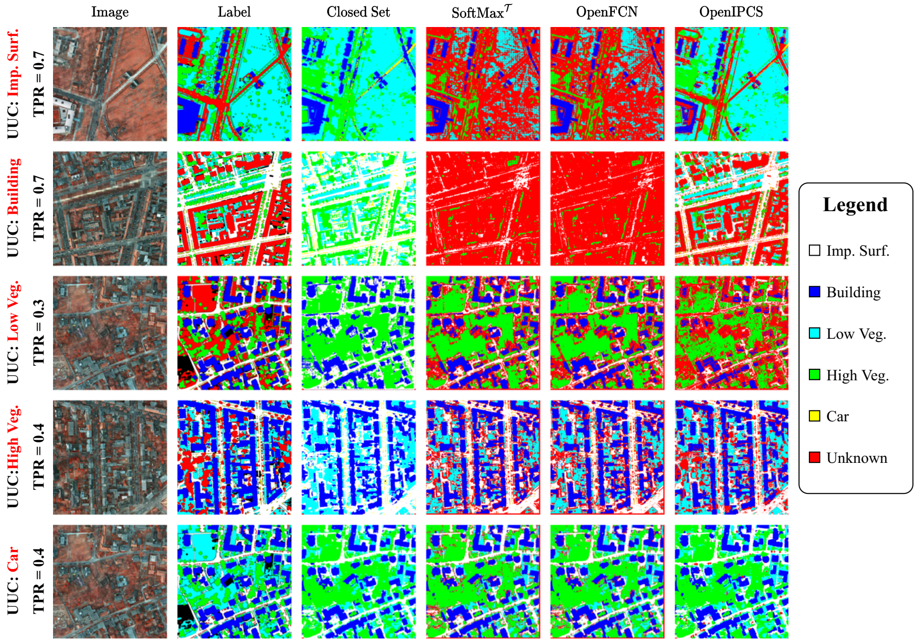


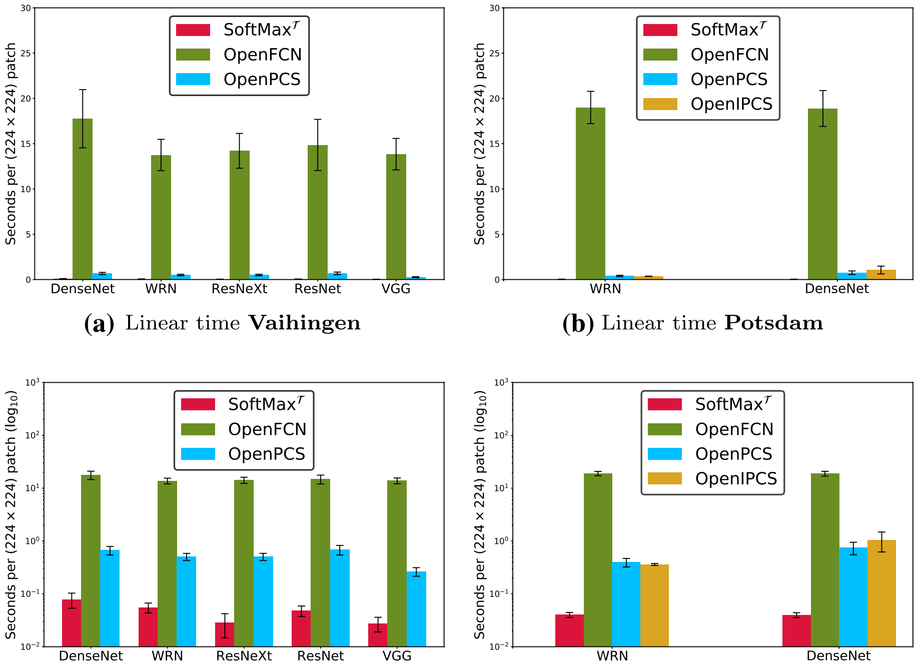
Fig. 13 Some visual result samples obtained for the **Potsdam** dataset according to distinct UUCs in the LOCO protocol and distinct TPR thresholds for SoftMax$^{\mathcal{T}}$, OpenFCN and OpenIPCS

corroborates with previous analysis and conclusions about the effectiveness of the OpenPCS mainly for discriminating UUCs.

Performing a deep analysis of the qualitative results, we can see that the proposed OpenPCS identified the UUC *Building* almost perfectly, while the SoftMax$^{\mathcal{T}}$ and OpenFCN techniques quite confused this UUC with other KCCs using the same threshold TPR. This same outcome can be seen for the UUC *Car*. Although this UCC comprises only a tiny percentage of the total amount of pixels and does not contribute a lot in terms of κ, it was much better identified by the OpenPCS than by other approaches. In fact, the above outcomes are repeated for all UUCs, except for the *Low Vegetation* one, which has enormous intra-class variability (with pixels of grass fields, sidewalk-like areas and other structures) and, consequently, natural erratic behaviour. For a more detailed qualitative analysis, please, check the project's webpage.

5.2.4 Runtime performance analysis

One of the most important motivations for proposing OpenPCS was the observation that OpenMax did not scale well to dense labeling, being prohibitively expensive when operating in a pixelwise fashion, while the simple SoftMax$^{\mathcal{T}}$ performed exponentially faster. In order to properly quantify the time differences between SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS, we computed the per-patch runtimes of each method for 224×224 patch resolution. These results are shown in Fig. 14 for both Vaihingen and Potsdam (Fig. 14a, b, respectively) and these same results are also presented in $\log_{10}$ scale (Fig. 14c, d,



(a) Linear time **Vaihingen**

(b) Linear time **Potsdam**

(c) Log time **Vaihingen**

(d) Log time **Potsdam**

Fig. 14 Per-patch time comparison between the proposed approaches. The time presented in the y-axis is shown in seconds for each (224 × 224) patch across all test patches. The left part of the figure (**a**, **c**) shows times for **Vaihingen**, while the rightmost figures (**b**, **d**) depict times for **Potsdam**. Results are also shown in linear (**a**, **b**) and log_{10} scale (**c**, **d**) in order to show the exponential distinction between execution times across SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS. Confidence intervals for each plot are shown as error bars computed with the t-Student distribution on the average execution runtimes for patches over 5 runs of the LOCO protocol (one for each class set as UUC) for each backbone

respectively), as the linear time comparisons severely hampered the visualization of SoftMax$^{\mathcal{T}}$'s performance when compared to OpenFCN.

Visual analysis of Fig. 14 reveals the discrepancies between OSR methods, with SoftMax$^{\mathcal{T}}$ inference being the fastest, usually taking between 0.03 and 0.1 s per 224 × 224 patch. On the opposite side, OpenFCN was observed to be by far the slowest method, with runtimes for one single patch in the range between 10 and 20 s. This may be justified by the fact that, at each inference, the method needs to sort the predictions (softmax activations) for each sample (e.g., for each pixel) in order to multiply them by the correct alpha value previously calculated by the OpenMax, then recalibrating the prediction scores. This sample-wise sorting at each inference, originally proposed in the OpenMax and consequently incorporated into the OpenFCN, significantly affects the running time of the algorithm, making it highly unsuitable for real-time computer vision applications.

On the faster end of the spectrum between SoftMax$^{\mathcal{T}}$ and OpenFCN, there were OpenPCS and OpenIPCS, with execution times between 0.25 and 0.7 seconds per patch. Both the online and offline PCAs used for the inference of UUCs from the proposed methods are highly vectorized operations, which allows them to be parallelized into several processing cores and be faster even in single-core architectures. While we did not

Table 7 Class divisions in experiments with two and three UUCs among the classes of **Vaihingen** and **Potsdam**

Exp.	KKCs			UUCs		
$E^{(0,1)}$	Low Veg.	High Veg.	Car	Imp. Surf.	Building	–
$E^{(0,4)}$	Building	Low Veg.	High Veg.	Imp. Surf.	Car	–
$E^{(1,4)}$	Imp. Surf.	Low Veg.	High Veg.	Building	Car	–
$E^{(2,3)}$	Imp. Surf.	Building	Car	Low Veg.	High Veg.	–
$E^{(0,1,4)}$	Low Veg.	High Veg.	–	Imp. Surf.	Building	Car
$E^{(0,2,3)}$	Building	Car	–	Imp. Surf.	Low Veg.	High Veg.

These experiments are using the notation $E^{(a,b,\ldots n)}$, with a, b and n representing the indices of the UUC classes in the semantic maps of **Vaihingen** and **Potsdam**

tune the algorithms for this purpose, OpenPCS' and OpenIPCS' runtimes allow for near real-time inference on applications as self-driving cars or autonomous drone control. In contrast, the original implementation of OpenMax from libMR is not naturally vectorized, requiring inferences to be performed linearly on pixels and severely hampering OpenFCN's performance.

5.3 Experiments with multiple UUCs

In addition to the results shown in Sects. 5.1 and 5.2, we also conducted experiments on multiple UUCs aiming to test scenarios with a larger proportion of UUCs to KKCs. The current section will be focused on presenting and discussing these experiments quantitatively and qualitatively for Vaihingen and Potsdam (Sect. 5.3.1) and for GRSS (Sect. 5.3.2).

5.3.1 Multiple UUCs on vaihingein and potsdam

In order to simplify and speed up the experiments, we split the classes in groups of KKCs with considerable semantic similarities, forming the divisions presented in Table 7. This table also presents our nomenclature for the multiple UUC experiments, in order to more easily refer to them in the text.

While some divisions are clear in purpose (e.g. $E^{(0,1,4)}$ and $E^{(2,3)}$ separate man-made constructions from vegetation; and $E^{(0,4)}$ split elements present in streets from the other classes), other combinations of UUCs were added to test the proposed methods in more diverse environments (e.g. $E^{(0,1)}$ or $E^{(0,2,3)}$). All AUC results for the 6 experiments with multiple UUCs ($E^{(0,1)}$, $E^{(0,4)}$, $E^{(1,4)}$, $E^{(2,3)}$, $E^{(0,1,4)}$ and $E^{(0,2,3)}$) in DenseNet-121 and WRN-50 backbones, as well as the additional metrics Acc^K, Pre^U and κ for $E^{(0,1)}$, $E^{(2,3)}$ and $E^{(0,1,4)}$ using the DenseNet-121 backbone are shown in Tables 8, 9, 10 and 11. However, for the sake of simplicity and objectivity, Acc^K, Pre^U and κ for $E^{(0,4)}$, $E^{(1,4)}$ and $E^{(0,2,3)}$ for DenseNet-121, as well as the experiments for all threshold-dependent metrics using the WRN-50 as a backbone are reported only in Appendix C.

Tables 8 and 9 show a much larger margin between OpenPCS/OpenIPCS and OpenFCN/SoftMax$^{\mathcal{T}}$, indicating that likelihood scoring from principal components is considerably more reliable in OSR Segmentation than both SoftMax Thresholding and

Table 8 AUC metrics for SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS on **Vaihingen** for multiple UUC experiments

Backbone	Methods	UUCs: Imp. Surf. Building	UUCs: Imp. Surf. Car	UUCs: Building Car	UUCs: High Veg. Low Veg.	UUCs: Imp. Surf. Building Car	UUCs: Imp. Surf. High Veg. Low Veg.
DenseNet	**SoftMax$^{\mathcal{T}}$**	.55 ± .06	.70 ± .04	.57 ± .04	.65 ± .02	.50 ± .07	.65 ± .02
	OpenFCN	.55 ± .06	.72 ± .04	.57 ± .04	.72 ± .03	.50 ± .06	.72 ± .03
	OpenPCS	**.89 ± .01** †	**.86 ± .02** †	**.94 ± .01** †	**.91 ± .02** †	**.90 ± .01** †	**.91 ± .02** †
WRN	**SoftMax$^{\mathcal{T}}$**	.57 ± .07	.63 ± .02	.55 ± .06	.76 ± .04	.49 ± .06	.70 ± .03
	OpenFCN	.59 ± .06	.65 ± .02	.56 ± .05	.78 ± .04	.48 ± .06	.76 ± .04
	OpenPCS	**.88 ± .02** †	**.76 ± .05** †	**.84 ± .06** †	**.85 ± .02** †	**.86 ± .01** †	**.87 ± .02** †

Results followed by † represent a significant increase in performance when using OpenPCS compared to both OpenFCN and SoftMax$^{\mathcal{T}}$. The significance of these results was assessed using a paired one-tailed t-Student hypothesis test with $p = 0.05$

Table 9 AUC metrics for SoftMax$^{\mathcal{T}}$, OpenFCN and OpenIPCS on **Potsdam** for multiple UUC experiments

Backbone	Methods	UUCs: Imp. Surf. Building	UUCs: Imp. Surf. Car	UUCs: Building Car	UUCs: High Veg. Low Veg.	UUCs: Imp. Surf. Building Car	UUCs: Imp. Surf. High Veg. Low Veg.
DenseNet	**SoftMax$^{\mathcal{T}}$**	.53 ± .10	.64 ± .11	.47 ± .10	.69 ± .09	.46 ± .05	.72 ± .08
	OpenFCN	.56 ± .10	.67 ± .10	.45 ± .10	**.73 ± .09**	.46 ± .05	.77 ± .08
	OpenIPCS	**.91 ± .05** †	**.86 ± .06** †	**.92 ± .07** †	.70 ± .09	**.92 ± .05** †	**.86 ± .07** †
WRN	**SoftMax$^{\mathcal{T}}$**	.47 ± .05	.81 ± .05	.50 ± .09	.62 ± .06	.40 ± .06	.71 ± .10
	OpenFCN	.52 ± .04	**.82 ± .05**	.56 ± .05	.62 ± .06	.40 ± .06	.75 ± .10
	OpenIPCS	**.86 ± .05** †	**.82 ± .06**	**.88 ± .08** †	**.76 ± .07** †	**.88 ± .04** †	**.90 ± .09** †

Results followed by † represent a significant increase in performance when using OpenIPCS compared to both OpenFCN and SoftMax$^{\mathcal{T}}$. The significance of these results was assessed using a paired one-tailed t-Student hypothesis test with $p = 0.05$

Table 10 Acc^K, Pre^U and κ results for different TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS using a **DenseNet-121** backbone in the **Vaihingen** dataset

TPR	SoftMax$^{\mathcal{T}}$			OpenFCN			OpenPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUCs: Imp. Surf./Building ($E^{(0,1)}$)									
Closed	**.82±.04**	.00±.00	.24±.06	**.82±.04**	.00±.00	.24±.06	**.82±.04**	.00±.00	.24±.06
0.10	.76±.04	.51±.07	.24±.06	.78±.04	.57±.08	.25±.06	**.82±.04**	.79±.39	.28±.06
0.30	.65±.05	.53±.08	.24±.06	.67±.05	.54±.08	.25±.06	**.82±.04**	**.99±.01**	.39±.05
0.50	.52±.07	.53±.08	.23±.06	.53±.07	.54±.08	.24±.07	.81±.04	.96±.02	.50±.05
0.70	.37±.07	.53±.08	.21±.06	.37±.07	.53±.08	.21±.07	.77±.04	.90±.03	**.60±.04**
0.90	.17±.09	.52±.08	.14±.11	.15±.08	.52±.08	.11±.10	.54±.04	.70±.07	.53±.04
UUCs: High/Low Veg. ($E^{(2,3)}$)									
Closed	**.91±.02**	.00±.00	.29±.04	**.91±.02**	.00±.00	.29±.04	**.91±.02**	.00±.00	.29±.04
0.10	.86±.02	.54±.09	.30±.04	.86±.02	.52±.10	.29±.04	.90±.01	**.89±.07**	.34±.04
0.30	.77±.03	.58±.08	.33±.03	.77±.03	.58±.08	.33±.03	.89±.02	.88±.04	.44±.04
0.50	.67±.03	.60±.08	.36±.03	.68±.03	.61±.07	.38±.03	.86±.02	.86±.04	.54±.04
0.70	.56±.04	.60±.07	.39±.04	.59±.03	.62±.07	.42±.03	.80±.04	.81±.06	**.61±.04**
0.90	.40±.04	.59±.07	.37±.04	.45±.04	.61±.07	.43±.04	.64±.07	.72±.07	**.61±.07**
UUCs: Imp. Surf./Building/Car ($E^{(0,1,4)}$)									
Closed	**.84±.03**	.00±.00	.23±.05	**.84±.03**	.00±.00	.23±.05	**.84±.03**	.00±.00	.23±.05
0.10	.77±.04	.46±.09	.21±.05	.76±.04	.43±.08	.20±.05	**.84±.03**	**.98±.03**	.28±.05
0.30	.63±.07	.50±.09	.20±.07	.62±.06	.49±.09	.19±.06	**.84±.03**	.97±.03	.39±.05
0.50	.47±.07	.51±.08	.17±.07	.47±.07	.51±.08	.17±.07	.83±.03	.95±.03	.50±.04
0.70	.33±.07	.53±.08	.16±.07	.34±.07	.53±.08	.17±.07	.78±.03	.89±.04	**.60±.03**
0.90	.15±.07	.53±.08	.11±.08	.16±.07	.53±.08	.12±.08	.58±.03	.73±.07	.56±.04

Bold values indicate the best overall Acc^K, Pre^U and κ values for a certain experiment with multiple UUCs. "Closed" rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation. This table focuses on experiments $E^{(0,1)}$, $E^{(2,3)}$ and $E^{(0,1,4)}$, while the complete multiple UUC experiments can be seen in Table 17

OpenMax. Specifically, AUC results show significant improvements in both Vaihingen and Potsdam when using principal components modeling than the baselines for almost all experiments. OpenPCS and OpenIPCS showed far superior performance to both OpenFCN and SoftMax$^{\mathcal{T}}$ in the majority of multiple UUC experiments, even reaching AUCs greater than 0.9 in many cases, allowing for reliable UUC identification in scenarios with larger openness. On Vaihingen five out of six experiments—$E^{(0,1)}$, $E^{(0,4)}$, $E^{(1,4)}$ and $E^{(0,1)}$—showed significantly greater performance for OpenPCS, reaching AUCs of 0.91, 0.86, 0.92, 0.92 and 0.86 using the DenseNet-121 backbone, respectively. With ROC curves this close to the left upper corner, one can set the cutoff value for TPRs between 0.6 and 0.9 with relatively small FPRs in the range of 0.1–0.2. In other words, in scenarios with a high proportion of UUCs to KKCs, the recognition of unknown pixels using OpenPCS and OpenIPCS can be done without compromising as much the performance of KKCs, which might also be caused by a better modeling of the fewer classes by the FCNs. This assessment will be further discussed in the results presented in Tables 10 and 11.

Table 11 Acc^K, Pre^U and κ results for different TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenIPCS using a **DenseNet-121** backbone in the **Potsdam** dataset

TPR	SoftMax$^{\mathcal{T}}$			OpenFCN			OpenPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUCs: Imp. Surf./Building ($E^{(0,1)}$)									
Closed	**.71±.03**	.00±.00	.17±.06	**.71±.03**	.00±.00	.17±.06	**.71±.03**	.00±.00	.17±.06
0.10	.66±.03	.55±.14	.18±.06	.68±.03	.67±.12	.19±.06	**.71±.03**	.28±.44	.18±.06
0.30	.56±.05	.53±.17	.18±.06	.58±.05	.57±.16	.20±.07	.70±.03	**.97±.02**	.30±.06
0.50	.43±.08	.52±.17	.16±.08	.46±.07	.54±.17	.19±.08	.69±.03	.96±.03	.40±.05
0.70	.27±.10	.50±.17	.12±.10	.30±.11	.52±.17	.15±.11	.67±.04	.92±.05	**.50±.04**
0.90	.09±.09	.49±.17	.05±.09	.10±.10	.50±.17	.06±.11	.51±.09	.74±.10	.48±.08
UUCs: High/Low Veg. ($E^{(2,3)}$)									
Closed	**.93±.05**	.00±.00	.29±.15	**.93±.05**	.00±.00	.29±.15	**.93±.05**	.00±.00	.29±.15
0.10	.89±.05	.61±.16	.31±.14	.89±.05	.63±.17	.31±.14	**.93±.05**	**.87±.11**	.34±.15
0.30	.82±.07	.65±.19	.35±.11	.84±.06	.69±.19	.37±.11	.90±.06	.80±.17	.41±.12
0.50	.73±.09	.66±.20	.39±.08	.78±.08	.70±.19	.43±.08	.76±.16	.70±.24	.41±.12
0.70	.61±.13	.64±.20	.41±.09	.68±.11	.68±.20	**.47±.08**	.57±.22	.63±.23	.37±.17
0.90	.33±.17	.57±.20	.29±.17	.40±.18	.60±.21	.36±.18	.32±.11	.57±.18	.28±.10
UUCs: Imp. Surf./Building/Car ($E^{(0,1,4)}$)									
Closed	**.76±.04**	.00±.00	.17±.08	**.76±.04**	.00±.00	.17±.08	**.76±.04**	.00±.00	.17±.08
0.10	.72±.04	.58±.14	.19±.08	.72±.04	.57±.15	.19±.08	**.76±.04**	.28±.44	.18±.07
0.30	.58±.06	.51±.19	.16±.07	.59±.06	.51±.19	.17±.07	**.76±.04**	**.97±.02**	.32±.09
0.50	.37±.08	.47±.20	.08±.06	.38±.08	.48±.20	.08±.06	.75±.05	.96±.02	.43±.09
0.70	.15±.07	.46±.19	-.02±.05	.15±.07	.46±.19	-.02±.05	.72±.06	.93±.03	.54±.07
0.90	.03±.02	.48±.18	-.03±.02	.03±.02	.48±.18	-.03±.02	.59±.13	.79±.07	**.55±.10**

Bold values indicate the best overall Acc^K, Pre^U and κ values for a certain experiment with multiple UUCs. "Closed" rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation. This table focuses on experiments $E^{(0,1)}$, $E^{(2,3)}$ and $E^{(0,1,4)}$, while the complete multiple UUC experiments can be seen in Table 19

The only exception to the superior performance of OpenPCS/OpenIPCS in multiple UUC experiments was $E^{(2,3)}$ on Potsdam—that is, when *High* and *Low Vegetation* were removed from the set of KKCs used during training. This was an expected outcome, as OpenIPCS did show their worse results in *High/Low Vegetation* on the experiments discussed in Sect. 5.1. However, one should notice that no method was able to achieve in $E^{(2,3)}$ the same performance of OpenPCS/OpenIPCS in the other multiple UUC experiments. We attribute this failure case to the high intra-class variability of *Low Vegetation*, while we raise the hypothesis that multimodal modeling for the likelihood distribution and/or a^* multimodal gaussian modeling should improve the results on these classes.

Similarly to Sect. 5.2.1, Tables 10 and 11 show the quantitative measures of Acc^K, Pre^U and κ, aiming to quantitatively assess the performance of the proposed methods and baselines on the KKCs, on the UUCs and the overall performance encompassing known and unknown classes.

These tables show major improvements, in terms of κ, for all experiments, except the ones using *High/Low Vegetation* in the Potsdam dataset. As previously explained, we hypothesize that this is due to the high intra-class variability of those classes.

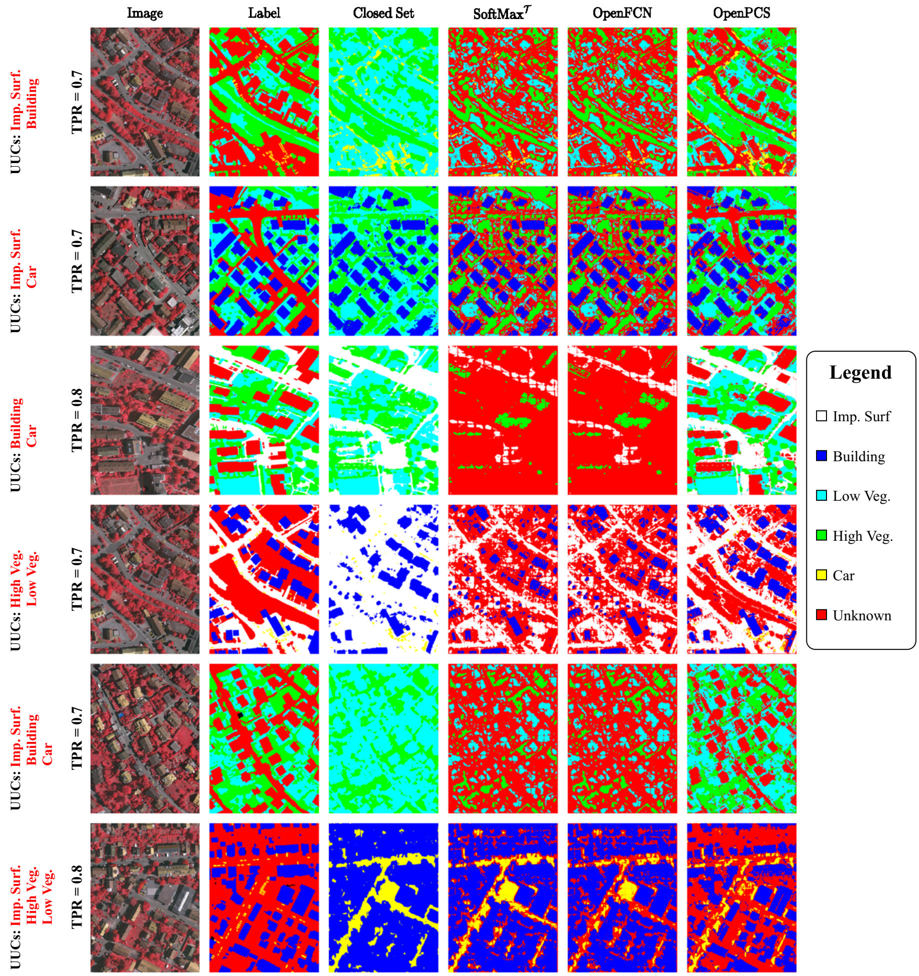


Fig. 15 Images, ground truths and predictions for SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS on the **Vaihingen** dataset for the six experiments with distinct UUC combinations presented in Table 7. These predictions were obtained using a **DenseNet-121**

Disregarding this experiment, for all others, the gains, in terms of κ, were even better than the ones obtained with only one UCC (Sect. 5.2.2), implying that the proposed OpenPCS/OpenIPCS are more robust to problems with higher openness. Deeply analyzing such gains, we may observe that they come from the fact that the proposed methods are capable of improving the recognition of UUCs (Pre^U) without significantly sacrificing the identification of KKCs (Acc^K). Precisely, in many cases, OpenPCS/OpenIPCS achieved more than 0.90 of Pre^U with only 1 2% loss in Acc^K. This shows the capacity of the proposed methods to efficiently perform open set semantic segmentation even in datasets with greater openness. As with the experiments with only one UCC, all obtained results using multiple UUCs as well as a discussion about them can be seen in Appendix C.

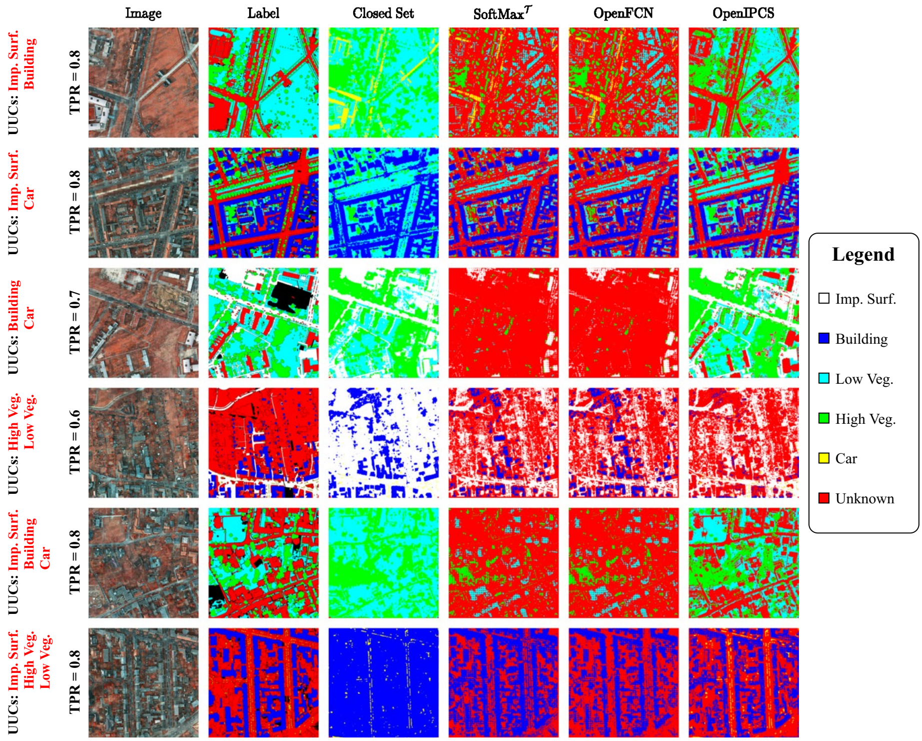
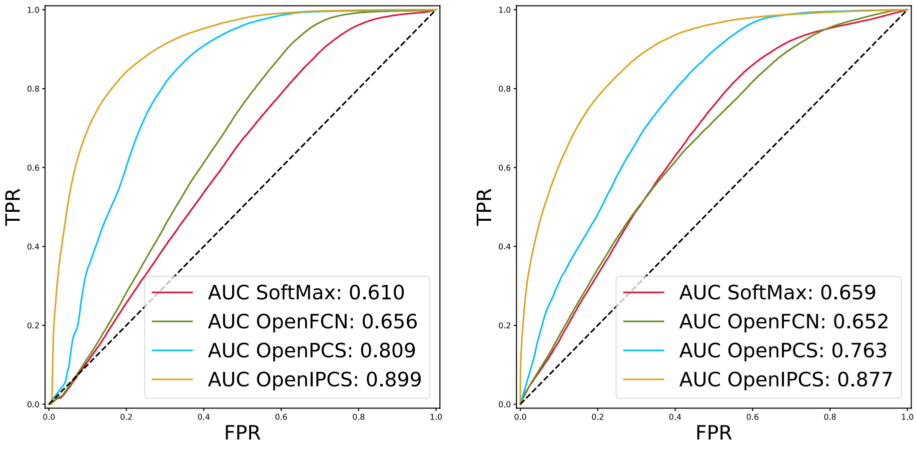


Fig. 16 Images, ground truths and predictions for SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS on the **Potsdam** dataset for the six experiments with distinct UUC combinations presented in Table 7. These predictions were obtained using a **DenseNet-121**



(a) UUCs: Vegetation

(b) UUCs: Building

Fig. 17 ROC curves for SoftMax$^{\mathcal{T}}$, OpenFCN, OpenPCS and OpenIPCS on **Houston** using a **DenseNet-121** backbone

Table 12 Acc^K, Pre^U and κ results for different TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenIPCS using a **DenseNet-121** backbone in the **Houston** dataset

TPR	SoftMax$^{\mathcal{T}}$			OpenFCN			OpenPCS			OpenIPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUCs: Vegetation												
Closed	**0.63**	0.00	0.43	**0.63**	0.00	0.43	**0.63**	0.00	0.43	**0.63**	0.00	0.43
0.10	0.60	0.28	0.42	0.60	0.29	0.42	0.60	0.29	0.42	0.61	0.41	0.43
0.30	0.56	0.32	0.43	0.56	0.34	0.43	0.56	0.34	0.43	0.58	**0.55**	0.46
0.50	0.50	0.33	0.41	0.51	0.36	0.43	0.51	0.36	0.43	0.55	0.53	0.48
0.70	0.40	0.33	0.37	0.43	0.35	0.40	0.43	0.35	0.40	0.51	0.52	**0.49**
0.90	0.26	0.31	0.27	0.32	0.35	0.34	0.32	0.35	0.34	0.44	0.46	0.47
UUCs: Building												
Closed	**0.65**	0.00	**0.53**	**0.65**	0.00	**0.53**	**0.65**	0.00	**0.53**	**0.65**	0.00	**0.53**
0.10	0.63	0.20	**0.53**	0.63	0.21	**0.53**	0.63	0.21	**0.53**	**0.65**	0.00	**0.53**
0.30	0.59	0.20	0.52	0.59	0.21	0.51	0.59	0.21	0.51	0.60	0.32	0.52
0.50	0.54	0.20	0.49	0.53	0.20	0.48	0.53	0.20	0.48	0.53	0.27	0.49
0.70	0.47	0.19	0.44	0.43	0.18	0.41	0.43	0.18	0.41	0.47	0.25	0.45
0.90	0.32	0.17	0.32	0.27	0.16	0.27	0.27	0.16	0.27	0.36	0.21	0.37

Bold values indicate the best overall Acc^K, Pre^U and κ values for a certain experiment with multiple UUCs. "Closed" rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation

It is noticeable in Figs. 15 and 16 that OpenPCS produces better predictions when compared to the other methods for almost all pairs or triplets of UUCs ($E^{(0,1)}$, $E^{(0,4)}$, $E^{(1,4)}$, $E^{(0,1,4)}$ and $E^{(0,2,3)}$), but exhibits worse results in experiment $E^{(2,3)}$. These results reiterate the inability of OpenPCS to deal with the high intra-class variability present in *Low Vegetation*, as previously shown in Sects. 5.1 and 5.2.

As previously reported in Sect. 5.2.3, OpenFCN and SoftMax$^{\mathcal{T}}$ still suffered with the naturally lower prediction confidences in multiple UUC experiments, with most of the UUC pixels predicted by these methods lying on object boundaries, even between two KKCs. Again, this is likely responsible for the considerably poorer quantitative results of these methods shown in Tables 8, 9, 11, and 11, with OpenPCS and OpenIPCS excelling due to their use of middle-level features from the networks, which still contain information about the input space, not being entirely bound by the output class space as OpenFCN and SoftMax$^{\mathcal{T}}$.

One last remark about the qualitative results in Figs. 15 and 16 is regarding $E^{(0,1)}$, $E^{(2,3)}$ and $E^{(0,2,3)}$. In all those cases, the class *Car* is a KKC, being naturally the most unrepresented objects in both Vaihingen and Potsdam due to the large class imbalance. While SoftMax$^{\mathcal{T}}$ and OpenFCN severely struggled with this class imbalance, being unable to properly identify *Cars* as KKCs, OpenPCS and OpenIPCS preserved vastly more correctly predicted *Car* pixels, indicating that they are more robust to high class imbalance during the training of their generative model.

5.3.2 Multiple UUCs on GRSS

Figure 17 shows ROC curves and AUCs for the Houston dataset on two scenarios with multiple UUCs at a time: (i) the first scenario, referenced as *Vegetation*, is composed of the classes healthy grass, stressed grass, artificial turf, evergreen trees, and deciduous trees; and (ii) the second scenario, referenced as *Building*, is composed of residential buildings, and non-residential buildings. As presented in Sects. 5.2 and 5.3.1, one can clearly see the superiority of OpenPCS and OpenIPCS in comparison to SoftMax$^{\mathcal{T}}$ or OpenFCN, with Principal Component Scoring obtaining higher AUCs than the other methods. Additionally, in both experiments, OpenIPCS presented considerably higher performance than OpenPCS, with the former surpassing the latter in AUC by approximately 0.1. Even though further experimentation is required for any definitive assertion, this result serves as initial evidence that OpenIPCS is more adaptable to a scenario with a larger number of KKCs.

Table 12 presents threshold-dependent metrics for experiments in the Houston dataset. Overall, known class accuracy results are not as high as in Vaihingen and Potsdam; an expected outcome given that this is a very fine-grained dataset. Aside from this, we can observe that for *Vegetation* as UUCs the OpenIPCS outperformed all other methods, as well as the Closed Set scenario, in terms of κ and Pre^{U}. This outcome is similar to those previously reported for Vaihingen/Potsdam. For the *Buildings* as UUCs, all approaches produced very similar outcomes, with the best result, in terms of κ, being the one obtained by the Closed Set. We believe that this is due to the class imbalance of the Houston dataset, i.e., *Buildings* classes represent a small fraction of the dataset and therefore do not impact the final result as much as the *Vegetation* classes, which are more prevalent in the dataset.

6 Conclusion

In this manuscript we introduced, formally defined and explored the problem of Open Set Segmentation and proposed two approaches for solving OSR segmentation tasks: (1) OpenFCN, which is based on the well-known OpenMax (Bendale and Boult 2016) approach; and (2) OpenPCS, a completely novel method for Open Set Semantic Segmentation. A comprehensive evaluation protocol based on a set of standard objective metrics to be used on this novel field that takes into account KKCs and UUCs was proposed, executed and discussed for the proposed methods and the main baseline: SoftMax$^{\mathcal{T}}$.

OpenPCS—and its more scalable variant referred to as OpenIPCS—yielded significantly better results than OpenFCN and SoftMax$^{\mathcal{T}}$. OpenPCS and OpenIPCS are able to merge activations from both shallower layers—containing input pixel-level information—and deeper layers that encode class-level information, while OpenFCN can only take into account information about class, as it operates on the outputs of the network. Performance analysis of OpenPCS/OpenIPCS vs. OpenFCN suggests that using middle-level feature information that combines both the input and output spaces of the FCN is highly useful for OSR segmentation. The latter scheme works well for sparse labeling scenarios (Bendale and Boult 2016) (e.g. image classification), however, dense labeling inherently consists of highly correlated samples (pixels) and labels (semantic maps) with border regions naturally presenting lower confidence than the interior pixels of objects. This distinction between sparse and dense labeling severely hampered the capabilities of OpenFCN and SoftMax$^{\mathcal{T}}$ of detecting OOD samples.

As expected, UUCs with larger semantic and visual distinctions in relation to the remaining KKCs also showed considerably better classification performance in the LOCO protocol. Specifically, *Low Vegetation* samples have a high intra-class variability, representing grass fields, sidewalk-like areas and other structures in the Vaihingen and Potsdam datasets, rendering them similar to samples from the classes *Street* and *Tree*. This resulted in lower performances in almost all metrics for all methods assessed in our experimental procedure. Closed Set predictions by the DNN architectures evaluated in Sect. 5.2 generally resulted in lower overall κ results than when OSR was added in the form of OpenPCS mainly, with gains up to 0.20 in this metric for the *Building* UUC. The analysis conducted in Sect. 5.2.4 also revealed the discrepancies between OSR methods in inference runtime, with SoftMax$^{\mathcal{T}}$ being the fastest method, OpenFCN being the exponentially slower and OpenPCS being a reasonable compromise, being able to run close to real-time, depending on GPU memory for larger patches and/or image size.

As OpenFCN is simply a fully convolutional version of OpenPixel (da Silva et al. 2020), also based on OpenMax (Bendale and Boult 2016), we can confidently infer that even OpenFCN performed better than OpenPixel. That is because OpenPixel was based on a traditional CNNs trained in a patchwise fashion for the classification of the central pixel and fully convolutional training was already shown to improve both segmentation performance and dramatically improve runtime efficiency of patchwise approaches in the pattern recognition literature (Long et al. 2015). While OpenFCN differs in this aspect, it even uses the same meta-recognition lib and some of the code of OpenPixel for the computation of OpenMax. One important variation of OpenPixel's approach that still needs to be verified in the context of fully convolutional training is the post processing using morphology, called Morph-OpenPixel by da Silva et al. (2020). This post-processing was shown to considerably improve the performance of OpenPixel. At last, even though time comparisons were not made available by da Silva et al. (2020), it is also possible to infer that OSR segmentation using even OpenFCN is exponentially faster than patchwise training also due to the experiments performed on the original FCN paper (Long et al. 2015).

Experiments in Sect. 5.3 lead to the conclusion that the superiority of OpenPCS and OpenIPCS in comparison with OpenFCN and SoftMax$^{\mathcal{T}}$ is even more pronounced in settings with larger openness—that is, in experiments with a greater ration between the number of UUCs and KKCs. Large AUC values computed from the ROC curves in the binary task of OOD recognition—that is, classifying between UUCs and KKCs—allowed for reliable UUC identification via larger TPR cutoffs with smaller FPRs in OpenPCS/OpenIPCS. These results may aid in the planning of future developments of OSR Segmentation in datasets with a larger variety of classes that can be assessed as UUCs. Hence, we are planning future developments of our approaches in experimental setups, which may include the exploitation of: (i) Computer Vision datasets, such as Pascal VOC (Everingham et al. 2015) and MS COCO (Lin et al. 2014; ii) synthetic datasets, including GTA-V (Richter et al. 2016), SYLVIA (Ros et al. 2016; iii) urban scene understanding datasets, such as CityScapes (Cordts et al. 2016); and (iv) other Remote Sensing datasets, including the DLR-SkyScapes (Azimi et al. 2019) and DOTA/iSAID (Xia et al. 2018; Waqas Zamir et al. 2019). This will allow a better understanding of the effectiveness of the proposed approaches in scenarios with different images (varying from RGB to hyperspectral) and contexts (such as urban and rural), and with a variable number of classes.

Finally, approaches as CGDL (Sun et al. 2020) and C2AE (Oza and Patel 2019) couple the supervised training for KKC classification with the training of the generative model—usually a variation on an AE architecture. These approaches allow for an end-to-end training of DNNs for OSR image classification tasks, and are currently considered the state-of-the-art for this task. Future works for OpenPCS/OpenIPCS include the adaptation of these methods for Open Set Semantic Segmentation, which would incorporate low-, middle- and high-level semantic features into the recognition of OOD samples, possibly allowing for an end-to-end training of Pixel Anomaly Detection and OSR segmentation. We also aim to investigate the capabilities of other lighter and/or more robust models in the OpenPCS pipeline, such as Gaussian Mixture Models (GMMs) (Bishop 2006; Attias 2000) instead of the simpler Principal Components used for computing multimodal likelihoods. Such method should obtain better performance with KKCs that are multimodal in nature. Additionally, we intend to incorporate more lightweight models during inference (e.g. One-Class SVM (OCSVM) (Schölkopf et al. 2001; Scheirer et al. 2012)) in order to perform OSR Segmentation in real-time, which may benefit applications such as autonomous driving or anomaly detection in videos for time-sensitive applications. At last, performing the likelihood scoring on instances of objects may reduce the OpenFCN problem with borders, where the certainty of the network predictions is naturally lower. For that, merging OpenFCN and possibly also OpenPCS with a Faster-RCNN (Ren et al. 2015) and/or Mask-RCNN (He et al. 2017) could improve OSR detection with less noisy UUC regions.

Appendix A performance on vaihingen for KKCs and UUCs

OSR tasks are inherently multi-objective, as Open Set algorithms must be able to successfully discern UUCs from KKCs, while still being able to correctly classify samples from distinct KKCs. Tables 13 and 14 show results from Acc^K, Pre^U and κ in the Vaihingen dataset using FCNs with WRN-50 and DenseNet-121 backbones, respectively.

Tables 13, and 14 show the performance of the OSR segmentation approaches on both KKCs and UUCs. The comparison of TPRs larger than one—that is, thresholds that allow for OSR—with their Closed Set counterparts (TPR = 0) using κ reveals that, in many cases, assuming openness improves object recognition in scenarios where full knowledge of the world is not possible. The most noticeable improvements in κ happen in UUCs that are visually and semantically more distinct from the other classes in the dataset: *Impervious Surfaces*, *Building*, *Tree* and *Car*. In the following paragraphs we will discuss the performance of the methods for each UUC in LOCO, starting from *Impervious Surfaces*.

Impervious Surfaces pixels in both Vaihingen and Potsdam are mainly composed of streets and driveways in residential/commercial buildings. These surfaces are considerably distinct from vegetation areas, as the "gray" signature in IRRG bands of asphalt is considerably distinct from the redness of plants due to their large reflection of near-infrared radiation. On Vaihingen, κ using fully closed DNNs achieve 0.53 with WRN-50 and and 0.52 with DenseNet-121. By adding openness to these DNNs via OpenPCS, we were able to increase these κ metrics to 0.68 and 0.65, respectively, representing gains in κ between 0.13 and 0.15. The OSR methods allowed the methods to correctly delineate most streets and driveways in the samples—as can also be seen in a qualitative manner in Sect. 5.2.3. SoftMax$^{\mathcal{T}}$ and OpenFCN also improved the performance of the closed DNNs for *Impervious Surfaces*, albeit in a smaller magnitude when compared to OpenPCS, achieving gains of between 0.05 and 0.09.

Closed Set WRN-50 achieved an Acc^K of 0.86, followed closely by DenseNet-121 with 0.85 when *Impervious Surface* pixels were set as UUC in LOCO. For the same TPR, Acc^K was only barely affected by the openness of OpenPCS, maintaining their original levels

Table 13 Results for different unknown TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS from an FCN with **WRN-50** backbone in **Vaihingen**

TPR	SoftMax$^{\mathcal{T}}$			OpenFCN			OpenPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUC: Impervious Surfaces									
Closed	**.86±.01**	.00±.00	.53±.04	**.86±.01**	.00±.00	.53±.04	**.86±.01**	.00±.00	.53±.04
0.10	.85±.01	.61±.11	.55±.03	.85±.01	.77±.07	.56±.04	**.86±.01**	**1.00±.00**	.56±.04
0.30	.81±.02	.52±.10	.57±.02	.83±.02	.63±.10	.59±.02	**.86±.01**	.95±.03	.62±.02
0.50	.76±.02	.50±.09	.59±.02	.78±.02	.55±.10	.61±.02	.84±.02	.85±.08	**.68±.02**
0.70	.70±.03	.47±.10	.59±.03	.72±.03	.50±.10	.62±.02	.78±.04	.64±.13	.67±.04
0.90	.56±.05	.42±.10	.52±.06	.59±.04	.44±.09	.56±.05	.53±.07	.41±.10	.49±.08
UUC: Building									
Closed	**.82±.02**	.00±.00	.49±.05	**.82±.02**	.00±.00	.49±.05	**.82±.02**	.00±.00	.49±.05
0.10	.79±.02	.34±.07	.49±.05	.79±.02	.34±.08	.49±.05	**.82±.02**	**1.00±.00**	.53±.05
0.30	.73±.03	.36±.06	.49±.04	.73±.03	.36±.07	.49±.04	**.82±.02**	.99±.01	.59±.04
0.50	.65±.03	.36±.05	.48±.04	.65±.03	.36±.06	.49±.04	**.82±.02**	.93±.04	.65±.03
0.70	.52±.05	.33±.04	.42±.04	.54±.03	.35±.05	.44±.03	.80±.03	.85±.06	**.70±.03**
0.90	.23±.12	.28±.04	.20±.11	.34±.05	.31±.05	.30±.04	.71±.04	.64±.09	.68±.04
UUC: Low Vegetation									
Closed	**.92±.01**	.00±.00	.61±.04	**.92±.01**	.00±.00	.61±.04	**.92±.01**	.00±.00	.61±.04
0.10	.91±.01	**.47±.05**	.62±.04	.90±.01	.46±.05	.62±.04	.90±.02	.46±.12	.62±.04
0.30	.86±.02	.45±.05	**.64±.04**	.86±.02	.44±.06	.63±.04	.84±.04	**.47±.12**	.62±.05
0.50	.79±.03	.43±.05	.63±.04	.77±.04	.41±.06	.61±.05	.77±.05	.44±.08	.61±.05
0.70	.66±.07	.39±.05	.56±.07	.63±.07	.36±.06	.53±.08	.67±.05	.41±.06	.57±.05
0.90	.44±.09	.32±.05	.40±.09	.43±.08	.32±.04	.39±.08	.49±.04	.35±.05	.45±.04
UUC: Tree									
Closed	**.88±.02**	.00±.00	.54±.07	**.88±.02**	.00±.00	.54±.07	**.88±.02**	.00±.00	.54±.07
0.10	.81±.04	.22±.05	.50±.08	.81±.04	.23±.04	.51±.08	**.88±.02**	**.88±.08**	**.57±.06**
0.30	.70±.05	.28±.06	.46±.08	.69±.06	.27±.05	.45±.08	.81±.07	.62±.17	**.57±.10**
0.50	.62±.06	.32±.06	.45±.07	.60±.06	.31±.06	.43±.07	.61±.14	.37±.08	.44±.16
0.70	.53±.06	.35±.07	.43±.06	.51±.06	.34±.07	.41±.06	.38±.07	.29±.05	.28±.07
0.90	.40±.05	.34±.06	.37±.05	.37±.05	.33±.06	.33±.05	.17±.06	.28±.05	.14±.06
UUC: Car									
Closed	**.84±.02**	.00±.00	**.77±.02**	**.84±.02**	.00±.00	**.77±.02**	**.84±.02**	.00±.00	**.77±.02**
0.10	.77±.05	.01±.01	.69±.05	.77±.04	.01±.01	.70±.04	**.84±.02**	**.27±.06**	**.77±.02**
0.30	.64±.06	.01±.01	.56±.06	.64±.06	.01±.01	.56±.06	.83±.02	.25±.05	**.77±.02**
0.50	.54±.05	.01±.01	.46±.05	.55±.06	.01±.01	.47±.05	.81±.02	.13±.04	.75±.02
0.70	.45±.04	.02±.01	.38±.04	.46±.05	.02±.01	.39±.04	.72±.04	.05±.02	.64±.04
0.90	.35±.04	.02±.01	.28±.03	.34±.04	.02±.01	.28±.03	.56±.05	.03±.01	.48±.05

Bold values indicate the best overall Acc^K, Pre^U and κ results for a certain UUC. "Closed" rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation

of 0.86 and 0.85 up until a TPR of 0.30. In contrast, Acc^K was considerably sacrificed in SoftMax$^{\mathcal{T}}$ and OpenFCN for TPRs larger than 0.10, meaning that pixels that were correctly classified by the original DNNs were cast as UUCs by the OSR post-processing. OpenPCS

Table 14 Results for different unknown TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS from an FCN with **DenseNet-121** backbone in **Vaihingen**

TPR	SoftMax$^{\mathcal{T}}$			OpenFCN			OpenPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUC: Impervious Surfaces									
Closed	**.85±.02**	.00±.00	.52±.04	**.85±.02**	.00±.00	.52±.04	**.85±.02**	.00±.00	.52±.04
0.10	.83±.03	.47±.11	.53±.04	**.85±.02**	.67±.12	.55±.04	**.85±.02**	**.97±.02**	.55±.04
0.30	.79±.03	.48±.10	.56±.02	.81±.03	.56±.11	.58±.02	**.85±.03**	.91±.07	.61±.03
0.50	.74±.03	.47±.11	.57±.02	.76±.03	.52±.11	.59±.02	.82±.04	.77±.14	**.65±.03**
0.70	.67±.04	.45±.10	.57±.04	.70±.04	.49±.11	.59±.03	.71±.09	.57±.15	.61±.09
0.90	.54±.05	.41±.09	.51±.06	.58±.04	.43±.10	.54±.05	.38±.06	.35±.08	.35±.06
UUC: Building									
Closed	**.83±.03**	.00±.00	.50±.06	**.83±.03**	.00±.00	.50±.06	**.83±.03**	.00±.00	.50±.06
0.10	.77±.03	.26±.06	.48±.05	.78±.03	.28±.07	.48±.05	.82±.03	**.98±.02**	.53±.05
0.30	.70±.03	.32±.06	.47±.04	.71±.03	.32±.06	.47±.04	.82±.03	.96±.02	.59±.05
0.50	.62±.03	.33±.06	.45±.04	.62±.03	.34±.06	.46±.04	.82±.03	.94±.03	.65±.04
0.70	.50±.06	.33±.06	.40±.06	.51±.06	.33±.06	.41±.06	.80±.03	.87±.06	**.70±.03**
0.90	.30±.13	.30±.06	.26±.13	.29±.15	.30±.06	.25±.14	.71±.04	.64±.07	.68±.03
UUC: Low Vegetation									
Closed	**.92±.01**	.00±.00	.61±.04	**.92±.01**	.00±.00	.61±.04	**.92±.01**	.00±.00	.61±.04
0.10	.90±.01	.45±.08	.62±.04	.90±.01	.46±.09	.62±.04	**.92±.01**	**.70±.06**	.64±.04
0.30	.85±.02	.43±.07	.63±.04	.85±.03	.43±.08	.63±.04	.88±.02	.56±.05	**.66±.04**
0.50	.77±.04	.41±.06	.60±.05	.76±.05	.40±.06	.60±.06	.77±.06	.44±.07	.61±.07
0.70	.64±.06	.37±.04	.54±.06	.61±.07	.35±.05	.51±.07	.55±.14	.33±.07	.45±.14
0.90	.42±.07	.31±.04	.38±.07	.39±.06	.30±.04	.35±.06	.23±.11	.26±.04	.19±.11
UUC: Tree									
Closed	**.87±.02**	.00±.00	.53±.05	**.87±.02**	.00±.00	.53±.05	**.87±.02**	.00±.00	.53±.05
0.10	.84±.02	.33±.09	.53±.05	.84±.02	.34±.09	.53±.05	**.87±.02**	**.97±.04**	.56±.05
0.30	.77±.04	.37±.09	.53±.05	.77±.04	.37±.09	.53±.06	.86±.02	.82±.11	.62±.05
0.50	.68±.05	.37±.09	.51±.06	.68±.06	.37±.08	.50±.06	.80±.04	.64±.10	**.63±.06**
0.70	.57±.06	.37±.08	.46±.06	.54±.07	.35±.08	.44±.07	.70±.07	.50±.04	.60±.08
0.90	.35±.10	.33±.09	.31±.11	.30±.08	.31±.08	.27±.08	.47±.10	.38±.04	.43±.09
UUC: Car									
Closed	**.84±.02**	.00±.00	**.77±.03**	**.84±.02**	.00±.00	**.77±.03**	**.84±.02**	.00±.00	**.77±.03**
0.10	.77±.04	.01±.01	.69±.04	.76±.04	.01±.00	.68±.04	.83±.02	**.17±.08**	.76±.02
0.30	.66±.06	.01±.01	.58±.06	.65±.05	.01±.01	.57±.05	.80±.04	.07±.03	.72±.04
0.50	.55±.07	.01±.01	.47±.06	.56±.05	.01±.01	.48±.05	.68±.07	.03±.01	.60±.08
0.70	.45±.06	.02±.01	.38±.06	.47±.06	.02±.01	.39±.05	.40±.10	.02±.01	.33±.09
0.90	.32±.05	.02±.01	.26±.04	.34±.05	.02±.01	.27±.04	.13±.05	.01±.00	.10±.04

Bold values indicate the best overall Acc^K, Pre^U and κ results for a certain UUC. “Closed” rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation

maintained large Acc^K values up until TPRs of 0.50, with a large drop only being seen in a TPR of 0.70. These results imply that OpenPCS is more accuracy efficient when identifying OOD *Impervious Surface* pixels, barely sacrificing KKC segmentation performances.

The precision of unknowns (Pre^U) is another evidence for the superiority of OpenPCS compared to OpenFCN and SoftMax$^{\mathcal{T}}$, as it reaches a perfect precision of 1.00 in WRN-50 for a TPR (recall) of 0.10, followed closely by DenseNet's Pre^U of 0.97 for the same TPR. Pre^U remains larger than 0.90 for both WRN-50 and DenseNet-121 up to a TPR of 0.30, meaning that UUC predictions for *Impervious Surfaces* from OpenPCS were highly reliable, even allowing for real-world applications of the method. Similarly to Acc^K, the precision of unknowns Pre^U was significantly lower from the start on SoftMax$^{\mathcal{T}}$ and OpenFCN, with the best results for these methods peaking (with 0.10 TPR using WRN-50) on 0.61 and 0.77, respectively.

The UUC *Building* yielded the most drastic gains in κ when OpenPCS was introduced, as showed by the increase from 0.50 to 0.70 with the DenseNet-121 backbone (Table 14). *Building* pixels are certainly among the most visually distinct features in the IRRG images in both texture and shape. In addition to that, these structures are rather distinct in the DSM data, as they can be easily seen on depth maps as sudden variations in altitude in comparison with the surrounding terrain. Finally, context also plays a larger role in *Bulding* detection, as both commercial and residential structures reside close to driveways or parkways (both *Street* samples) and are often surrounded by vegetation in suburban areas as Vaihingen and Potsdam. Predictions for UUC *Building* using a TPR of 0.10 were also highly precise, with Pre^U values of 1.0 (perfect scoring) and 0.98 (near perfect) for WRN-50 and DenseNet-121, respectively.

Overall results from UUC *Tree* followed similar patterns to *Street* and *Building*, but in a smaller scale. κ gains using OpenPCS with WRN-50 and DenseNet-121 backbones were in the scale of 0.03 and 0.10, respectively, showing a noticeable boost in performance mainly in DenseNet-121. In contrast, the best κ values for OpenFCN and SoftMax$^{\mathcal{T}}$ were the Closed Set ones, with κ quickly dropping even since TPR 0.10 on WRN-50. The best Pre^U was also achieved by OpenPCS on both backbones, reaching a maximum of 0.97 for a TPR of 0.10 on DenseNet-121, while no noticeable drop in Acc^K could be seen in this setting. Acc^K remained relatively close to the Closed Set KKC classification performance up to 0.30 of TPR with DenseNet-121 and up to 0.10 on WRN-50.

Car pixels represent a tiny proportion of samples in both Potsdam and Vaihingen. Thus, analysis of Open vs. Closed Set methods by only using κ would yield basically no gain in adding OSR to the pipeline, as can be seen in Tables 13 and 14. This is due to the benefits of detecting UUCs not outweighing the added errors in KKC classification in metrics that measure general multiclass performance as Balanced Accuracy or κ. Instead, a more accurate picture of *Car* segmentation can be seen when taking into account Pre^U and Acc^K, together with the TPR, which is automatically given because it served to compute the thresholds for the methods. One can easily see that OpenPCS' Acc^K performance was barely degraded for TPR values up to 0.50 in WRN-50, while also remaining considerably close to the Closed Set accuracy in DenseNet-121 until a TPR of 0.30. In other words, there was virtually no harm to KKC classification in WRN-50 in exchange for the correct identification of 50% of *Car* pixels, while DenseNet-121 also kept high Acc^K values for 30% of the *Car* pixels being correctly segmented.

Pre^U values were not particularly high in either method, even for small TPRs, reaching a peak of 0.27 in WRN-50 with a TPR of 0.10—that is, only approximately one in four pixels predicted to be pertaining to a vehicle in this setting was really from a vehicle. The low Pre^U of UUC *Car* can be explained by the relatively high intra-class variability in class *Building*, as most of the pixels misclassified as *Car* were, in fact, from parts of *Building* structures that were not common in the rest of the houses and warehouses in the data. We reached this conclusion after the qualitative evaluation shown in Sect. 5.2.3 that can be fully appreciated in the project's webpage. It is natural that houses have distinct shapes, sizes and rooftop textures, resulting. OpenPCS' Pre^U results, though indeed small, are still more than one order of magnitude larger than both OpenFCN and SoftMax$^{\mathcal{T}}$.

Lastly, one important distinction between segmentation and detection must be highlighted in the case for *Car*: while many vehicles were not perfectly segmented, almost all automobiles were at least partially identified by OpenPCS. Qualitative results available at the project's webpage show that in both datasets a large proportion of the automobiles were correctly identified by having a large subset of their pixels correctly identified as pertaining to a UUC. The same cannot be said for OpenFCN and SoftMax$^{\mathcal{T}}$, which correctly detected (even if only partially) a much lower proportion of vehicles. In addition to that, one can see that even for small TPRs of 0.10 OpenFCN and SoftMax$^{\mathcal{T}}$ presented large drops to both Acc^K and κ performances.

Appendix B performance on potsdam for KKCs and UUCs

Similarly to the presentation of the results from Vaihingen, Tables 15 and 16 show the same per-class threshold-dependent metrics (κ, Acc^K and Pre^U) in Potsdam for WRN-50 and DenseNet-121, respectively. As Potsdam is a considerably harder dataset when compared to Vaihingen—mainly due to the large imbalance and much larger spatial resolution—almost all metrics achieve lower values than the ones from Tables 13 and 14. We highlight that the standard OpenPCS with random patch selection for the training of PCA was also computed for Potsdam (as shown in Fig. 11), but this section only reports OpenIPCS values. The incremental training of PCA showed to be much more stable in the larger Potsdam dataset, while we choose to not shot the original OpenPCS results mostly due to spatial constraints and organization.

Discussion regarding Potsdam results closely resembles the ones presented from Vaihingen on Section A, albeit with one major distinction: relatively poor performance in comparison with Vaihingen due to the hardships encountered in processing Potsdam samples. These difficulties include the massive size of the dataset, restricting tuning of the networks' hyperparameters due to computational constraints; the larger spatial resolution (6000×6000 per image) resulting from smaller physical areas covered by each pixel in comparison with Vaihingen, which also accentuate the downsides of introducing a larger correlation between adjacent pixels and providing less context on a 224×224 patch; and an extremely imbalanced set of labels.

One can indeed observe significant κ improvements by introducing OpenIPCS to the segmentation procedure in UUCs *Impervious Surfaces*, *Building*, *Tree* and *Car*, mainly with a DenseNet-121 backbone (Table 16). Closed Set κ values for these classes were 0.42, 0.41, 0.51 and 0.56, while the best OSR results (assuming optimal TPR choice) increased these values to 0.55, 0.57, 0.52 and 0.57. Analogously to Vaihingen, the increases to κ were larger for *Impervious Surfaces* and *Building* (between 0.13 and 0.16), while *Tree* and *Car* achieve much more modest gains of only 0.01, which are not statistically significant. *Building* again achieved a close to perfect Pre^U (0.96) using DenseNet-121 and TPR 0.10, followed by the also high *Impervious Surfaces* Pre^U of 0.87 also with 0.10 of TPR.

Low Vegetation again was an outlier in the LOCO protocol using DenseNet-121, with SoftMax$^{\mathcal{T}}$ and OpenFCN achieving less degradation in Acc^K and a larger Pre^U

Table 15 Results for different unknown TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenIPCS from an FCN with **WRN-50** backbone in **Potsdam**

TPR	SoftMax$^{\mathcal{T}}$			OpenFCN			OpenIPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUC: Impervious Surfaces									
Closed	**.73±.09**	.00±.00	.40±.09	**.73±.09**	.00±.00	.40±.09	**.73±.09**	.00±.00	.40±.09
0.10	.71±.09	.58±.14	.41±.08	.72±.09	**.70±.13**	.42±.08	.71±.08	.54±.18	.41±.08
0.30	.66±.09	.46±.11	.43±.08	.68±.09	.51±.10	**.44±.08**	.64±.08	.47±.09	.40±.07
0.50	.59±.10	.40±.11	.41±.09	.60±.10	.42±.11	.42±.09	.54±.08	.40±.09	.36±.07
0.70	.47±.11	.35±.11	.37±.10	.48±.11	.36±.11	.37±.10	.39±.08	.33±.09	.28±.07
0.90	.29±.11	.31±.11	.25±.10	.28±.12	.31±.11	.25±.11	.16±.04	.28±.08	.13±.03
UUC: Building									
Closed	**.77±.04**	.00±.00	.44±.08	**.77±.04**	.00±.00	.44±.08	**.77±.04**	.00±.00	.44±.08
0.10	.74±.04	.34±.15	.43±.07	.75±.04	.38±.16	.44±.07	**.77±.04**	**.90±.06**	.46±.07
0.30	.64±.07	.29±.15	.40±.06	.65±.07	.30±.15	.40±.06	.76±.04	.82±.05	.51±.06
0.50	.52±.10	.28±.15	.35±.08	.54±.09	.29±.15	.36±.07	.74±.04	.77±.08	.56±.05
0.70	.37±.10	.27±.14	.26±.08	.39±.09	.28±.14	.28±.07	.71±.05	.69±.14	**.59±.06**
0.90	.16±.07	.26±.13	.13±.06	.18±.08	.26±.13	.14±.07	.56±.13	.48±.20	.50±.14
UUC: Low Vegetation									
Closed	**.86±.04**	.00±.00	.46±.14	**.86±.04**	.00±.00	.46±.14	**.86±.04**	.00±.00	.46±.14
0.10	.81±.04	.37±.11	.45±.14	.81±.04	.35±.11	.44±.15	.83±.05	.55±.14	.47±.14
0.30	.71±.09	.39±.12	.43±.16	.69±.11	.38±.12	.41±.18	.78±.07	**.56±.15**	.49±.13
0.50	.60±.14	.40±.12	.40±.18	.58±.15	.39±.12	.38±.19	.72±.08	.55±.15	.51±.12
0.70	.49±.15	.40±.13	.37±.16	.47±.15	.39±.14	.35±.16	.64±.10	.53±.15	**.52±.11**
0.90	.34±.13	.39±.16	.30±.11	.31±.11	.38±.16	.27±.10	.50±.13	.47±.15	.47±.10
UUC: Tree									
Closed	**.89±.05**	.00±.00	**.60±.07**	**.89±.05**	.00±.00	**.60±.07**	**.89±.05**	.00±.00	**.60±.07**
0.10	.85±.05	.26±.04	.58±.07	.85±.05	.25±.04	.58±.07	.80±.06	**.28±.17**	.53±.10
0.30	.77±.05	**.28±.05**	.55±.08	.76±.05	.27±.05	.54±.08	.67±.09	.23±.09	.45±.12
0.50	.65±.08	.26±.05	.48±.11	.63±.10	.25±.06	.46±.12	.55±.11	.23±.07	.38±.13
0.70	.50±.12	.25±.05	.39±.13	.47±.13	.24±.05	.35±.14	.39±.13	.21±.06	.27±.13
0.90	.29±.11	.22±.04	.23±.11	.24±.11	.21±.04	.18±.10	.16±.09	.19±.04	.11±.08
UUC: Car									
Closed	**.75±.07**	.00±.00	**.64±.09**	**.75±.07**	.00±.00	**.64±.09**	**.75±.07**	.00±.00	**.64±.09**
0.10	.74±.08	.22±.13	.63±.10	**.75±.08**	.28±.16	**.64±.09**	**.75±.07**	**.29±.15**	**.64±.09**
0.30	.69±.12	.04±.03	.58±.12	.70±.12	.06±.03	.60±.12	.74±.07	.14±.08	.63±.09
0.50	.59±.13	.02±.02	.49±.12	.62±.13	.03±.02	.52±.12	.71±.06	.07±.05	.60±.08
0.70	.47±.13	.02±.01	.38±.11	.51±.12	.02±.01	.41±.10	.62±.07	.04±.03	.51±.09
0.90	.31±.13	.02±.01	.25±.10	.34±.12	.02±.01	.27±.09	.45±.09	.02±.02	.35±.09

Bold values indicate the best overall Acc^K, Pre^U and κ results for a certain UUC. "Closed" rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation

for the same TPR values. Since *Low Vegetation* fills a much larger proportion of pixels in Potsdam than in Vaihingen, it would be expected that adding OSR would considerably improve κ values in comparison with Closed Set when this class was set to UUC in the LOCO protocol. However, the best κ results were the Closed Set ones for

Table 16 Results for different unknown TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenIPCS from an FCN with **DenseNet-121** backbone in **Potsdam**

TPR	SoftMax$^{\mathcal{T}}$			OpenFCN			OpenIPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUC: Impervious Surfaces									
Closed	**.77±.08**	.00±.00	.42±.10	**.77±.08**	.00±.00	.42±.10	**.77±.08**	.00±.00	.42±.10
0.10	.74±.08	.44±.11	.43±.09	.74±.08	.45±.11	.43±.09	.76±.08	**.87±.11**	.45±.10
0.30	.68±.09	.40±.14	.43±.09	.69±.09	.42±.14	.44±.09	.75±.09	.84±.10	.51±.09
0.50	.59±.11	.37±.14	.41±.10	.60±.10	.37±.14	.42±.10	.73±.10	.76±.12	**.55±.09**
0.70	.47±.14	.35±.14	.36±.13	.47±.14	.35±.14	.36±.13	.66±.10	.63±.13	**.55±.09**
0.90	.29±.18	.31±.13	.25±.17	.28±.18	.31±.13	.25±.16	.46±.12	.42±.11	.40±.09
UUC: Building									
Closed	**.74±.08**	.00±.00	.41±.07	**.74±.08**	.00±.00	.41±.07	**.74±.08**	.00±.00	.41±.07
0.10	.71±.08	.37±.17	.41±.06	.72±.08	.40±.18	.42±.06	**.74±.08**	**.96±.05**	.44±.07
0.30	.61±.10	.30±.17	.37±.06	.62±.09	.32±.17	.38±.06	.74±.08	.91±.11	.49±.06
0.50	.46±.12	.27±.15	.29±.09	.47±.11	.28±.15	.30±.08	.73±.08	.85±.13	.55±.06
0.70	.25±.14	.25±.14	.16±.11	.26±.12	.25±.14	.17±.10	.69±.09	.73±.17	**.57±.09**
0.90	.09±.07	.24±.13	.06±.06	.07±.05	.24±.12	.05±.04	.55±.22	.53±.21	.50±.22
UUC: Low Vegetation									
Closed	**.83±.03**	.00±.00	**.44±.14**	**.83±.03**	.00±.00	**.44±.14**	**.83±.03**	.00±.00	**.44±.14**
0.10	.78±.04	.35±.13	.43±.15	.78±.05	.34±.13	.42±.15	**.74±.06**	.24±.06	.38±.17
0.30	.65±.13	**.36±.14**	.37±.19	.63±.15	.35±.14	.36±.21	.57±.09	.28±.11	.30±.17
0.50	.53±.16	**.36±.14**	.33±.19	.51±.16	.35±.14	.31±.19	.40±.08	.30±.14	.20±.13
0.70	.39±.16	**.36±.15**	.27±.17	.36±.15	.35±.15	.24±.16	.21±.06	.31±.15	.10±.08
0.90	.20±.12	.35±.16	.16±.12	.18±.10	.35±.16	.14±.09	.06±.02	.32±.16	.02±.03
UUC: Tree									
Closed	**.80±.10**	.00±.00	.51±.11	**.80±.10**	.00±.00	.51±.11	**.80±.10**	.00±.00	.51±.11
0.10	.77±.10	.25±.05	.50±.11	.77±.09	.26±.07	.50±.10	.79±.09	**.34±.12**	**.52±.10**
0.30	.69±.10	.25±.05	.47±.11	.69±.09	.25±.06	.48±.10	.69±.09	.26±.08	.47±.13
0.50	.59±.11	.25±.05	.42±.11	.59±.09	.25±.05	.43±.10	.54±.13	.23±.07	.36±.16
0.70	.49±.13	.25±.05	.38±.12	.47±.11	.24±.05	.36±.10	.31±.13	.19±.05	.19±.12
0.90	.39±.12	.25±.04	.34±.10	.32±.10	.23±.04	.27±.09	.10±.06	.18±.04	.06±.04
UUC: Car									
Closed	**.68±.11**	.00±.00	.56±.13	**.68±.11**	.00±.00	.56±.13	**.68±.11**	.00±.00	.56±.13
0.10	**.68±.12**	.15±.10	.56±.14	**.68±.11**	.16±.09	.56±.13	**.68±.11**	**.58±.29**	**.57±.13**
0.30	.65±.12	.05±.03	.53±.13	.65±.11	.06±.03	.54±.13	**.68±.11**	.35±.19	**.57±.14**
0.50	.60±.12	.03±.02	.49±.13	.62±.10	.04±.02	.51±.12	.67±.11	.20±.11	.56±.14
0.70	.52±.10	.03±.01	.42±.10	.56±.09	.03±.02	.45±.10	.65±.11	.10±.06	.54±.13
0.90	.38±.08	.02±.01	.30±.07	.44±.08	.02±.01	.34±.08	.58±.10	.05±.03	.47±.11

Bold values indicate the best overall Acc^K, Pre^U and κ results for a certain UUC. "Closed" rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation

all OSR methods on the DenseNet-121 backbone, implying that neither of the methods converged well enough on this architecture to compensate for the KKC prediction losses. *Low Vegetation* results were opposite on the WRN-50 backbone in Potsdam: κ

increased by 0.06 (for TPR 0.70); the best Pre^{U} results were indeed from OpenIPCS, peaking at 0.56 with TPR 0.30 and Acc^{K} degradation was much slower than OpenFCN and SoftMax$^{\mathcal{T}}$ as TPRs were allowed to grow. These discrepancies between architectures highlights that, even though OpenPCS and OpenIPCS were found to be the current state-of-the-art for OSR segmentation, they can be highly dependent on the DNN architecture, hyperparameters and proved to be rather unstable, being still a work in progress.

For almost all UUCs, Acc^{K} degraded much faster on Potsdam than on Vaihingen, as larger TPRs were evaluated. One important outlier in almost all aspects presented previously was *Car* on Potsdam. While on Vaihingen the κ did not increase at all in comparison to the Closet Set DNNs with the introduction of OpenPCS, the κ value for *Car* had a slight, but noticeable, increase from 0.56 to 0.57. Pre^{U} values for Car segmentation were also considerably larger on Potsdam, peaking at 0.58 for TPR 0.10 using DenseNet-121 and 0.29 using WRN-50. At the same time, Acc^{K} remained close to the Closed Set counterpart with TPRs up to 0.70, only observing a noticeable decrease in KKC classification performance on TPR 0.90. As for OpenFCN and SoftMax$^{\mathcal{T}}$, Pre^{U} results for *Car* were also higher and Acc^{K} also degraded slowly with the increase of TPR (comparing with results from Vaihingen), even if these metrics did not achieve the same gains as the ones from OpenPCS.

Appendix C complementary results for multiple UUCs

Continuing the discussion of Sect. 5.3, in this appendix we present additional results regarding the experiments with multiple UUCs, as foreseen in Sect. 5.3. First we present the threshold-independent AUC metric that compares OpenPCS/OpenIPCS with OpenMax and SoftMax$^{\mathcal{T}}$ on the whole spectrum of cutoffs. Tables 17, 18, 19 and 20 show the threshold-dependent metrics for **Vaihingen** with **DenseNet-121** and **WRN-50** backbones and **Potsdam** with **DenseNet-121** and **WRN-50** backbones.

Table 17 Acc^K, Pre^U and κ results for different TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS using a **DenseNet-121** backbone on **Vaihingen**

TPR	SoftMax$^{\mathcal{T}}$			OpenFCN			OpenPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUCs: Imp. Surf./Building									
Closed	**.82±.04**	.00±.00	.24±.06	**.82±.04**	.00±.00	.24±.06	**.82±.04**	.00±.00	.24±.06
0.10	.76±.04	.51±.07	.24±.06	.78±.04	.57±.08	.25±.06	**.82±.04**	.79±.39	.28±.06
0.30	.65±.05	.53±.08	.24±.06	.67±.05	.54±.08	.25±.06	**.82±.04**	**.99±.01**	.39±.05
0.50	.52±.07	.53±.08	.23±.06	.53±.07	.54±.08	.24±.07	.81±.04	.96±.02	.50±.05
0.70	.37±.07	.53±.08	.21±.06	.37±.07	.53±.08	.21±.07	.77±.04	.90±.03	**.60±.04**
0.90	.17±.09	.52±.08	.14±.11	.15±.08	.52±.08	.11±.10	.54±.04	.70±.07	.53±.04
UUCs: Imp. Surf./Car									
Closed	**.85±.02**	.00±.00	.50±.04	**.85±.02**	.00±.00	.50±.04	**.85±.02**	.00±.00	.50±.04
0.10	.82±.02	.39±.09	.50±.03	.82±.02	.40±.09	.50±.04	**.85±.02**	**.95±.05**	.53±.04
0.30	.77±.03	.44±.10	.52±.02	.77±.03	.44±.10	.52±.02	.84±.01	.92±.06	.60±.03
0.50	.71±.03	.44±.10	.53±.02	.71±.03	.44±.10	.53±.02	.83±.02	.84±.09	**.65±.01**
0.70	.62±.04	.42±.10	.52±.04	.61±.05	.42±.10	.51±.04	.76±.04	.65±.12	**.65±.03**
0.90	.58±.03	.41±.09	.50±.04	.45±.06	.37±.09	.41±.07	.44±.07	.40±.10	.41±.08
UUCs: Building/Car									
Closed	**.83±.02**	.00±.00	.48±.05	**.83±.02**	.00±.00	.48±.05	**.83±.02**	.00±.00	.48±.05
0.10	.77±.03	.24±.04	.46±.05	.78±.03	.26±.05	.47±.05	**.83±.02**	.00±.00	.48±.05
0.30	.69±.03	.30±.05	.45±.04	.69±.03	.30±.05	.45±.04	**.83±.02**	.77±.39	.56±.06
0.50	.58±.03	.31±.05	.41±.03	.59±.03	.32±.04	.41±.04	.82±.02	**.93±.03**	.65±.03
0.70	.44±.06	.31±.06	.33±.05	.43±.06	.31±.05	.33±.06	.81±.02	.86±.05	**.70±.03**
0.90	.15±.09	.27±.06	.11±.10	.11±.10	.26±.06	.08±.10	.72±.03	.64±.07	.69±.03
UUCs: High/Low Veg.									
Closed	**.91±.02**	.00±.00	.29±.04	**.91±.02**	.00±.00	.29±.04	**.91±.02**	.00±.00	.29±.04
0.10	.86±.02	.54±.09	.30±.04	.86±.02	.52±.10	.29±.04	.90±.01	**.89±.07**	.34±.04
0.30	.77±.03	.58±.08	.33±.03	.77±.03	.58±.08	.33±.03	.89±.02	.88±.04	.44±.04
0.50	.67±.03	.60±.08	.36±.03	.68±.03	.61±.07	.38±.03	.86±.02	.86±.04	.54±.04
0.70	.56±.04	.60±.07	.39±.04	.59±.03	.62±.07	.42±.03	.80±.04	.81±.06	**.61±.04**
0.90	.40±.04	.59±.07	.37±.04	.45±.04	.61±.07	.43±.04	.64±.07	.72±.07	**.61±.07**
UUCs: Imp. Surf./Building/Car									
Closed	**.84±.03**	.00±.00	.23±.05	**.84±.03**	.00±.00	.23±.05	**.84±.03**	.00±.00	.23±.05
0.10	.77±.04	.46±.09	.21±.05	.76±.04	.43±.08	.20±.05	**.84±.03**	**.98±.03**	.28±.05
0.30	.63±.07	.50±.09	.20±.07	.62±.06	.49±.09	.19±.06	**.84±.03**	.97±.03	.39±.05
0.50	.47±.07	.51±.08	.17±.07	.47±.07	.51±.08	.17±.07	.83±.03	.95±.03	.50±.04
0.70	.33±.07	.53±.08	.16±.07	.34±.07	.53±.08	.17±.07	.78±.03	.89±.04	**.60±.03**
0.90	.15±.07	.53±.08	.11±.08	.16±.07	.53±.08	.12±.08	.58±.03	.73±.07	.56±.04
UUCs: Imp. Surf./High/Low Veg.									
Closed	**.98±.01**	.00±.00	.08±.02	**.98±.01**	.00±.00	.08±.02	**.98±.01**	.00±.00	.08±.02
0.10	.97±.02	.96±.02	.12±.02	.97±.02	.96±.02	.12±.02	**.98±.01**	.80±.40	.12±.04
0.30	.96±.02	.97±.01	.21±.03	.96±.02	.97±.01	.22±.03	.97±.02	**.99±.00**	.25±.03
0.50	.96±.02	.97±.01	.23±.03	.95±.02	.97±.01	.31±.03	.96±.02	**.99±.01**	.38±.04
0.70	.96±.02	.97±.01	.23±.03	.95±.02	.97±.01	.32±.04	.94±.02	.98±.01	.54±.04
0.90	.96±.02	.97±.01	.23±.03	.95±.02	.97±.01	.32±.04	.77±.04	.92±.03	**.67±.03**

Bold values indicate the best overall Acc^K, Pre^U and κ values for a certain experiment with multiple UUCs. “Closed” rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation

Table 18 Acc^K, Pre^U and κ results for different TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenPCS using a **WRN-50** backbone on **Vaihingen**

TPR	SoftMax$^{\mathcal{T}}$			OpenFCN			OpenPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUCs: Imp. Surf./Building									
Closed	**.81±.04**	.00±.00	.23±.05	**.81±.04**	.00±.00	.23±.05	**.81±.04**	.00±.00	.23±.05
0.10	.76±.05	.52±.09	.23±.06	.78±.04	.61±.08	.24±.06	**.81±.04**	.59±.48	.26±.05
0.30	.66±.06	.55±.09	.24±.06	.67±.06	.56±.09	.25±.05	**.81±.04**	**.97±.03**	.38±.05
0.50	.54±.07	.55±.09	.25±.06	.56±.07	.56±.09	.26±.06	.79±.04	.91±.04	.48±.05
0.70	.41±.08	.55±.09	.24±.08	.42±.08	.56±.08	.25±.07	.72±.05	.83±.05	.55±.05
0.90	.22±.10	.54±.08	.19±.11	.23±.10	.54±.08	.20±.11	.58±.05	.72±.06	**.56±.04**
UUCs: Imp. Surf./Car									
Closed	**.84±.01**	.00±.00	.49±.04	**.84±.01**	.00±.00	.49±.04	**.84±.01**	.00±.00	.49±.04
0.10	.81±.02	.37±.09	.49±.03	.81±.02	.40±.09	.50±.03	.83±.02	.79±.09	.52±.04
0.30	.75±.02	.40±.09	.50±.02	.74±.02	.39±.09	.50±.02	.83±.02	**.86±.09**	.58±.02
0.50	.66±.03	.39±.09	.49±.01	.65±.03	.38±.09	.48±.01	.79±.04	.72±.17	**.61±.04**
0.70	.61±.03	.38±.08	.47±.02	.52±.02	.36±.08	.42±.02	.63±.09	.47±.14	.52±.09
0.90	.61±.03	.38±.08	.47±.02	.41±.03	.33±.07	.34±.03	.20±.07	.29±.07	.17±.08
UUCs: Building/Car									
Closed	**.80±.03**	.00±.00	.46±.06	**.80±.03**	.00±.00	.46±.06	**.80±.03**	.00±.00	.46±.06
0.10	.75±.04	.23±.05	.43±.06	.76±.04	.28±.05	.45±.06	**.80±.03**	.70±.36	.48±.05
0.30	.64±.06	.26±.05	.40±.07	.65±.06	.27±.05	.40±.07	.78±.04	**.81±.14**	.54±.06
0.50	.53±.07	.28±.05	.35±.07	.53±.07	.29±.05	.36±.07	.75±.06	.73±.14	.58±.06
0.70	.39±.07	.29±.05	.28±.06	.38±.07	.29±.05	.28±.06	.69±.08	.61±.15	**.59±.08**
0.90	.24±.07	.30±.05	.21±.06	.23±.08	.29±.05	.20±.07	.42±.11	.39±.10	.39±.11
UUCs: High/Low Veg.									
Closed	**.93±.02**	.00±.00	.30±.05	**.93±.02**	.00±.00	.30±.05	**.93±.02**	.00±.00	.30±.05
0.10	.89±.03	.60±.10	.32±.05	.89±.03	.61±.10	.32±.05	**.93±.02**	**.93±.02**	.35±.05
0.30	.82±.05	.65±.09	.38±.05	.83±.04	.66±.09	.38±.05	.91±.01	.91±.02	.46±.05
0.50	.75±.05	.67±.08	.44±.05	.77±.04	.69±.08	.46±.05	.86±.01	.84±.03	.53±.03
0.70	.67±.05	.67±.07	.50±.05	.72±.04	.71±.07	.54±.04	.77±.02	.78±.05	.59±.02
0.90	.58±.05	.67±.07	.55±.05	.63±.04	.70±.07	**.60±.04**	.58±.05	.69±.07	.56±.05
UUCs: Imp. Surf./Building/Car									
Closed	**.82±.04**	.00±.00	.21±.06	**.82±.04**	.00±.00	.21±.06	**.82±.04**	.00±.00	.21±.06
0.10	.74±.05	.45±.08	.19±.06	.74±.05	.44±.08	.19±.06	**.82±.04**	.76±.38	.25±.06
0.30	.58±.06	.47±.08	.16±.06	.58±.06	.46±.08	.15±.06	.78±.04	**.85±.05**	.34±.05
0.50	.46±.06	.51±.08	.16±.06	.45±.06	.50±.08	.16±.06	.75±.04	.83±.05	.43±.04
0.70	.33±.07	.53±.08	.16±.07	.33±.07	.53±.08	.16±.07	.71±.04	.82±.05	.54±.04
0.90	.13±.08	.52±.08	.08±.09	.12±.08	.52±.08	.07±.09	.58±.05	.75±.07	**.57±.04**
UUCs: Imp. Surf./High/Low Veg.									
Closed	**.98±.01**	.00±.00	.06±.01	**.98±.01**	.00±.00	.06±.01	**.98±.01**	.00±.00	.06±.01
0.10	.97±.01	.96±.02	.10±.01	.97±.01	.96±.02	.10±.01	.97±.01	.95±.02	.11±.02
0.30	.97±.01	**.97±.01**	.20±.02	.97±.01	**.97±.01**	.20±.02	.94±.02	.94±.02	.20±.03
0.50	.96±.01	**.97±.01**	.28±.03	.95±.01	**.97±.01**	.33±.04	.90±.03	.94±.02	.32±.04
0.70	.96±.01	**.97±.01**	.29±.04	.94±.01	**.97±.01**	.39±.06	.83±.03	.92±.02	.45±.04
0.90	.96±.01	**.97±.01**	.29±.04	.94±.01	**.97±.01**	.39±.06	.72±.03	.90±.02	**.62±.03**

Bold values indicate the best overall Acc^K, Pre^U and κ values for a certain experiment with multiple UUCs. “Closed” rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation

Table 19 Acc^K, Pre^U and κ results for different TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenIPCS using a **DenseNet-121** backbone on **Potsdam**

TPR	SoftMax$^{\mathcal{T}}$			OpenFCN			OpenPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUCs: Imp. Surf./Building									
Closed	**.71±.03**	.00±.00	.17±.06	**.71±.03**	.00±.00	.17±.06	**.71±.03**	.00±.00	.17±.06
0.10	.66±.03	.55±.14	.18±.06	.68±.03	.67±.12	.19±.06	**.71±.03**	.28±.44	.18±.06
0.30	.56±.05	.53±.17	.18±.06	.58±.05	.57±.16	.20±.07	.70±.03	**.97±.02**	.30±.06
0.50	.43±.08	.52±.17	.16±.08	.46±.07	.54±.17	.19±.08	.69±.03	.96±.03	.40±.05
0.70	.27±.10	.50±.17	.12±.10	.30±.11	.52±.17	.15±.11	.67±.04	.92±.05	**.50±.04**
0.90	.09±.09	.49±.17	.05±.09	.10±.10	.50±.17	.06±.11	.51±.09	.74±.10	.48±.08
UUCs: Imp. Surf./Car									
Closed	**.72±.11**	.00±.00	.36±.11	**.72±.11**	.00±.00	.36±.11	**.72±.11**	.00±.00	.36±.11
0.10	.69±.11	.43±.13	.37±.11	.70±.11	.52±.10	.38±.11	**.72±.11**	.73±.34	.38±.12
0.30	.62±.14	.38±.16	.36±.13	.64±.13	.42±.15	.38±.12	.71±.11	**.89±.12**	.45±.11
0.50	.54±.16	.36±.16	.36±.14	.55±.15	.38±.15	.37±.14	.71±.11	.87±.10	.52±.11
0.70	.44±.17	.35±.15	.33±.16	.45±.17	.36±.15	.35±.16	.66±.09	.77±.16	**.54±.09**
0.90	.29±.17	.32±.13	.25±.15	.28±.17	.32±.13	.24±.16	.34±.10	.39±.11	.30±.07
UUCs: Building/Car									
Closed	**.69±.11**	.00±.00	.34±.05	**.69±.11**	.00±.00	.34±.05	**.69±.11**	.00±.00	.34±.05
0.10	.63±.11	.24±.12	.32±.05	.62±.12	.21±.13	.31±.06	**.69±.11**	.00±.00	.34±.05
0.30	.52±.13	.25±.15	.28±.06	.50±.13	.24±.14	.27±.06	**.69±.10**	.63±.41	.42±.06
0.50	.40±.14	.26±.15	.23±.09	.38±.15	.25±.15	.21±.09	.68±.10	**.87±.11**	.49±.07
0.70	.25±.15	.26±.15	.15±.11	.23±.15	.25±.14	.14±.11	.65±.10	.70±.17	**.53±.09**
0.90	.07±.07	.25±.13	.03±.06	.06±.07	.25±.13	.03±.06	.52±.19	.54±.22	.47±.20
UUCs: High/Low Veg.									
Closed	**.93±.05**	.00±.00	.29±.15	**.93±.05**	.00±.00	.29±.15	**.93±.05**	.00±.00	.29±.15
0.10	.89±.05	.61±.16	.31±.14	.89±.05	.63±.17	.31±.14	**.93±.05**	**.87±.11**	.34±.15
0.30	.82±.07	.65±.19	.35±.11	.84±.06	.69±.19	.37±.11	.90±.06	.80±.17	.41±.12
0.50	.73±.09	.66±.20	.39±.08	.78±.08	.70±.19	.43±.08	.76±.16	.70±.24	.41±.12
0.70	.61±.13	.64±.20	.41±.09	.68±.11	.68±.20	**.47±.08**	.57±.22	.63±.23	.37±.17
0.90	.33±.17	.57±.20	.29±.17	.40±.18	.60±.21	.36±.18	.32±.11	.57±.18	.28±.10
UUCs: Imp. Surf./Building/Car									
Closed	**.76±.04**	.00±.00	.17±.08	**.76±.04**	.00±.00	.17±.08	**.76±.04**	.00±.00	.17±.08
0.10	.72±.04	.58±.14	.19±.08	.72±.04	.57±.15	.19±.08	**.76±.04**	.28±.44	.18±.07
0.30	.58±.06	.51±.19	.16±.07	.59±.06	.51±.19	.17±.07	**.76±.04**	**.97±.02**	.32±.09
0.50	.37±.08	.47±.20	.08±.06	.38±.08	.48±.20	.08±.06	.75±.05	.96±.02	.43±.09
0.70	.15±.07	.46±.19	-.02±.05	.15±.07	.46±.19	-.02±.05	.72±.06	.93±.03	.54±.07
0.90	.03±.02	.48±.18	-.03±.02	.03±.02	.48±.18	-.03±.02	.59±.13	.79±.07	**.55±.10**
UUCs: Imp. Surf./High/Low Veg.									
Closed	**.98±.01**	.00±.00	.02±.01	**.98±.01**	.00±.00	.02±.01	**.98±.01**	.00±.00	.02±.01
0.10	.93±.02	.83±.08	.04±.02	.93±.02	.84±.08	.04±.03	.97±.02	**.97±.03**	.07±.04
0.30	.88±.04	.89±.06	.12±.06	.89±.05	.89±.06	.12±.06	.96±.04	**.97±.03**	.17±.08
0.50	.84±.09	.90±.07	.22±.10	.84±.09	.90±.07	.23±.10	.93±.06	.96±.03	.30±.12
0.70	.80±.10	.89±.07	.31±.09	.76±.12	.89±.07	.35±.10	.88±.11	.95±.05	.44±.14
0.90	.79±.10	.89±.07	.34±.11	.65±.12	.87±.08	.46±.11	.63±.18	.87±.09	**.51±.14**

Bold values indicate the best overall Acc^K, Pre^U and κ values for a certain experiment with multiple UUCs. "Closed" rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation

Table 20 Acc^K, Pre^U and κ results for different TPR thresholds applied to SoftMax$^{\mathcal{T}}$, OpenFCN and OpenIPCS using a **WRN-50** backbone on **Potsdam**

TPR	SoftMax$^{\mathcal{T}}$			OpenFCNOpenFCN			OpenPCS		
	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ	Acc^K	Pre^U	κ
UUCs: Imp. Surf./Building									
Closed	**.74±.05**	.00±.00	.20±.07	**.74±.05**	.00±.00	.20±.07	**.74±.05**	.00±.00	.20±.07
0.10	.69±.05	.52±.17	.20±.07	.72±.05	.66±.12	.22±.08	**.74±.05**	.28±.44	.21±.07
0.30	.57±.07	.49±.19	.19±.05	.60±.06	.53±.18	.21±.06	.73±.05	**.95±.04**	.33±.07
0.50	.41±.07	.48±.19	.15±.04	.46±.06	.51±.19	.19±.05	.70±.06	.90±.04	.40±.08
0.70	.21±.06	.46±.19	.06±.04	.25±.05	.48±.18	.11±.04	.64±.07	.83±.06	**.47±.07**
0.90	.04±.03	.47±.18	-.01±.02	.05±.03	.47±.18	-.00±.02	.46±.08	.67±.12	.42±.07
UUCs: Imp. Surf./Car									
Closed	**.62±.16**	.00±.00	.29±.12	**.62±.16**	.00±.00	.29±.12	**.62±.16**	.00±.00	.29±.12
0.10	**.62±.16**	.80±.08	.32±.12	**.62±.16**	**.84±.08**	.32±.12	**.62±.16**	.72±.30	.32±.13
0.30	.61±.16	.74±.10	.37±.13	.61±.16	.80±.11	.38±.13	.61±.16	**.84±.10**	.37±.13
0.50	.56±.16	.59±.12	.40±.13	.58±.17	.64±.12	.41±.14	.60±.16	.83±.12	.43±.14
0.70	.50±.16	.50±.13	.40±.14	.49±.16	.50±.13	.39±.14	.55±.14	.72±.15	**.45±.13**
0.90	.37±.15	.40±.13	.34±.13	.36±.16	.40±.13	.33±.14	.26±.05	.37±.11	.23±.06
UUCs: Building/Car									
Closed	**.71±.09**	.00±.00	.36±.05	**.71±.09**	.00±.00	.36±.05	**.71±.09**	.00±.00	.36±.05
0.10	.65±.10	.24±.13	.33±.05	.67±.09	.30±.12	.35±.05	**.71±.09**	.00±.00	.36±.05
0.30	.54±.12	.26±.14	.29±.07	.56±.11	.29±.14	.31±.06	.70±.08	.62±.40	.43±.06
0.50	.42±.12	.27±.14	.25±.08	.44±.11	.28±.14	.27±.07	.68±.08	**.80±.13**	.49±.06
0.70	.27±.10	.26±.13	.17±.07	.29±.10	.27±.13	.18±.07	.62±.10	.64±.19	**.50±.10**
0.90	.08±.06	.25±.12	.05±.06	.09±.06	.25±.12	.06±.05	.43±.18	.43±.23	.38±.20
UUCs: High/Low Veg.									
Closed	**.92±.05**	.00±.00	.29±.16	**.92±.05**	.00±.00	.29±.16	**.92±.05**	.00±.00	.29±.16
0.10	.87±.05	.59±.11	.30±.16	.87±.05	.60±.12	.30±.16	.90±.05	**.75±.12**	.32±.16
0.30	.79±.07	.63±.14	.33±.15	.79±.07	.63±.14	.33±.14	.82±.08	.69±.16	**.36±.14**
0.50	.66±.08	.61±.16	.33±.12	.65±.08	.60±.17	.32±.11	.68±.10	.63±.17	.35±.12
0.70	.45±.08	.57±.18	.27±.08	.45±.07	.57±.18	.27±.08	.49±.10	.59±.17	.31±.11
0.90	.22±.05	.54±.18	.18±.06	.24±.05	.54±.18	.20±.06	.26±.09	.55±.17	.22±.09
UUCs: Imp. Surf./Building/Car									
Closed	**.67±.08**	.00±.00	.11±.04	**.67±.08**	.00±.00	.11±.04	**.67±.08**	.00±.00	.11±.04
0.10	.59±.08	.45±.18	.09±.03	.59±.09	.44±.20	.09±.03	**.67±.08**	.13±.32	.11±.03
0.30	.42±.12	.44±.21	.04±.05	.42±.12	.44±.21	.04±.05	.66±.07	.84±.34	.23±.08
0.50	.27±.11	.45±.20	.00±.06	.27±.11	.45±.20	.00±.06	.65±.07	**.95±.02**	.35±.04
0.70	.13±.07	.46±.19	-.03±.04	.13±.07	.46±.19	-.03±.04	.61±.07	.87±.08	**.43±.04**
0.90	.02±.02	.48±.18	-.04±.01	.02±.02	.48±.18	-.04±.01	.42±.07	.67±.15	.39±.08
UUCs: Imp. Surf./High/Low Veg.									
Closed	**.97±.03**	.00±.00	.05±.03	**.97±.03**	.00±.00	.05±.03	**.97±.03**	.00±.00	.05±.03
0.10	.94±.04	.89±.07	.07±.04	.94±.04	.89±.07	.07±.04	**.97±.03**	.00±.00	.05±.03
0.30	.90±.09	.91±.08	.14±.09	.90±.09	.91±.09	.14±.09	.97±.03	.28±.45	.09±.07
0.50	.83±.14	.90±.08	.22±.13	.83±.14	.90±.08	.22±.13	.94±.09	**.98±.04**	.33±.12
0.70	.76±.11	.88±.08	.28±.14	.71±.15	.87±.08	.31±.14	.90±.12	.96±.04	.47±.13
0.90	.74±.11	.88±.08	.29±.14	.41±.10	.82±.09	.31±.10	.73±.17	.91±.07	**.59±.14**

Bold values indicate the best overall Acc^K, Pre^U and κ values for a certain experiment with multiple UUCs. "Closed" rows mean unknown TPRs of 0.0, which is equivalent to Closed Set segmentation

Acknowledgements The authors would like to thank NVIDIA for the donation of the GPUs that allowed the execution of all experiments in this paper. Additionally, we thank FAPEMIG, CNPq and CAPES for the funding of this research.

Author Contributions The research was coordinated and conducted by *Jefersson A. dos Santos*, and the experimental procedure was designed by *Hugo Oliveira*; *Hugo Oliveira*, and *Caio Silva* conducted the implementation and execution of all experiments. *Hugo Oliveira* wrote the first version of this manuscript, assisted by *Gabriel L. S. Machado*and *Keiller Nogueira*; *Gabriel L. S. Machado*, *Jefersson A. dos Santos* and *Keiller Nogueira* were responsible for the final review of the manuscript.

Funding We would like to thank CAPES, CNPq (424700/2018-2), and FAPEMIG (APQ-00449-17) for the financial support provided for this research.

Availability of data and material Both datasets used for the experiments and all supplementary materials are publicly available, as specified in the paper.Code availability All the source codes used for the experiments of this paper, including OpenFCN, OpenPCS and OpenIPCS will be publicly available in this project's webpage (referenced in the manuscript).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Attias, H. (2000). A variational baysian framework for graphical models. In *Advances in Neural Information Processing Systems* (pp. 209–215).

Audebert, N., Le Saux, B., & Lefèvre, S. (2016). Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. In *Asian Conference on Computer Vision* (pp. 180–196). Springer.

Azimi, S.M., Henry, C., Sommer, L., Schumann, A., & Vig, E. (2019). Skyscapes fine-grained semantic understanding of aerial scenes. In *ICCV* (pp. 7393–7403).

Bendale, A., & Boult, T.E. (2016). Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1563–1572).

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Berlin: Springer.

Cardoso, D. O., Gama, J., & França, F. M. (2017). Weightless neural networks for open set recognition. *Machine Learning, 106*(9–10), 1547–1567.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3213–3223).

da Silva, C.C.V., Nogueira, K., Oliveira, H.N., & dos Santos, J.A. (2020). Towards open-set semantic segmentation of aerial images. arXiv:2001.10063.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). Ieee.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision, 111*(1), 98–136.

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2012). Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1915–1929.

Ge, Z., Demyanov, S., Chen, Z., & Garnavi, R. (2017). Generative openmax for multi-class open set classification. In *British Machine Vision Conference*.

Geng, C., Huang, S.J., & Chen, S. (2020). Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Early Access).

Goodfellow, I.J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv:1412.6572.

Guiotte, F., Pham, M., Dambreville, R., Corpetti, T., & Lefèvre, S. (2020). Semantic segmentation of łd points clouds: Rasterization beyond digital elevation models. IEEE Geoscience and Remote Sensing Letters pp. 1–4.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2961–2969).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).

Hendrycks, D., Mazeika, M., & Dietterich, T. (2019). Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations* arXiv:1812.04606.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K.Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4700–4708).

Kemker, R., Salvaggio, C., & Kanan, C. (2018). Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, *145,* 60–77. Deep Learning RS Data.

Kingma, D.P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:1412.6980.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images. Technical Report. Available at: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097–1105).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

Li, F., & Wechsler, H. (2005). Open set face recognition using transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(11), 1686–1697.

Liang, S., Li, Y., & Srikant, R. (2017). Enhancing the reliability of out-of-distribution image detection in neural networks. arXiv:1706.02690.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C.L. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision* (pp. 740–755). Springer.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3431–3440).

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). High-resolution aerial image labeling with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing, 55*(12), 7092–7103.

Marmanis, D., Schindler, K., Wegner, J. D., Galliani, S., Datcu, M., & Stilla, U. (2018). Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing, 135,* 158–172.

Nogueira, K., Dalla Mura, M., Chanussot, J., Schwartz, W.R., & dos Santos, J.A. (2016). Learning to semantically segment high-resolution remote sensing images. In *2016 23rd International Conference on Pattern Recognition (ICPR)* (pp. 3566–3571). IEEE.

Nogueira, K., Dalla Mura, M., Chanussot, J., Schwartz, W. R., & dos Santos, J. A. (2019). Dynamic multicontext segmentation of remote sensing images based on convolutional networks. *IEEE Transactions on Geoscience and Remote Sensing, 57*(10), 7503–7520.

Oza, P., & Patel, V.M. (2019). C2ae: Class conditioned auto-encoder for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2307–2316).

Pinheiro, P.H., & Collobert, R. (2014). Recurrent convolutional neural networks for scene labeling. In *31st International Conference on Machine Learning (ICML), CONF*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems* (pp. 91–99).

Richter, S.R., Vineet, V., Roth, S., & Koltun, V. (2016). Playing for data: Ground truth from computer games. In *European Conference on Computer Vision* (pp. 102–118). Springer.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A.M. (2016). The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3234–3243).

Scheirer, W. J., Jain, L. P., & Boult, T. E. (2014). Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(11), 2317–2324.

Scheirer, W. J., de Rezende Rocha, A., Sapkota, A., & Boult, T. E. (2012). Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(7), 1757–1772.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation, 13*(7), 1443–1471.

Sherrah, J. (2016). Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. arXiv:1606.02585.

Shwartz-Ziv, R., & Tishby, N. (2017). Opening the black box of deep neural networks via information. arXiv:1703.00810.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Srivastava, R.K., Greff, K., & Schmidhuber, J. (2015). Highway networks. arXiv:1505.00387.

Sun, X., Yang, Z., Zhang, C., Peng, G., & Ling, K.V. (2020). Conditional gaussian distribution learning for open set recognition. arXiv:2003.08823.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1–9).

Tipping, M. E., & Bishop, C. M. (1999). Mixtures of probabilistic principal component analyzers. *Neural Computation, 11*(2), 443–482.

Wang, H., Wang, Y., Zhang, Q., Xiang, S., & Pan, C. (2017). Gated convolutional neural network for semantic segmentation in high-resolution images. *Remote Sensing, 9*(5), 446.

Waqas Zamir, S., Arora, A., Gupta, A., Khan, S., Sun, G., Shahbaz Khan, F., Zhu, F., Shao, L., Xia, G.S., & Bai, X. (2019). isaid: A large-scale dataset for instance segmentation in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 28–37).

Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., & Zhang, L. (2018). Dota: A large-scale dataset for object detection in aerial images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1492–1500).

Yoshihashi, R., Shao, W., Kawakami, R., You, S., Iida, M., & Naemura, T. (2019). Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4016–4025).

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., & Xiao, J. (2015). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv:1506.03365.

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. arXiv:1605.07146.