IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Pre-print to appear in IEEE Access

Digital Object Identifier 10.1109/ACCESS.2022.3192605

Global and Local Features through Gaussian Mixture Models on Image Semantic Segmentation

DARWIN SAIRE,[1] AND ADÍN RAMÍREZ RIVERA,[1,2] (Senior Member, IEEE)
[1]Institute of Computing, University of Campinas, Brazil (e-mail: darwin.pilco@ic.unicamp.br)
[2]Department of Informatics, University of Oslo, Norway (e-mail: adinr@uio.no)

Corresponding author: Adín Ramírez Rivera (e-mail: adinr@uio.no).

This work was financed in part by the São Paulo Research Foundation (FAPESP) under grants No. 2017/16597-7, 2019/07257-3, and 2019/18678-0, and in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under grant No. 307425/2017-7. The Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA) institute provided, in part, infrastructure for this work.
Code available at https://gitlab.com/mipl/phgmm.

ABSTRACT The semantic segmentation task aims at dense classification at the pixel-wise level. Deep models exhibited progress in tackling this task. However, one remaining problem with these approaches is the loss of spatial precision, often produced at the segmented objects' boundaries. Our proposed model addresses this problem by providing an internal structure for the feature representations while extracting a global representation that supports the former. To fit the internal structure, during training, we predict a Gaussian Mixture Model from the data, which, merged with the skip connections and the decoding stage, helps avoid wrong inductive biases. Furthermore, our results show that we can improve semantic segmentation by providing both learning representations (global and local) with a clustering behavior and combining them. Finally, we present results demonstrating our advances in Cityscapes and Synthia datasets.

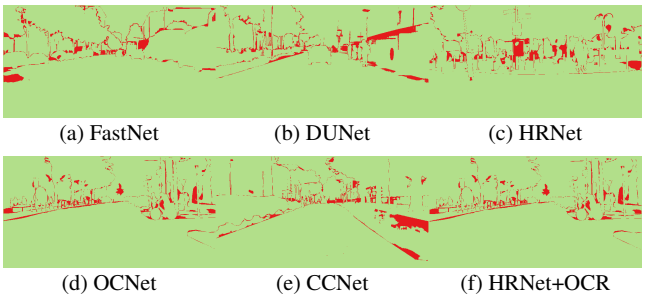
INDEX TERMS Explainable Latent Spaces, Context-aware features, Gaussian Mixture Models, Semantic Segmentation

I. INTRODUCTION

Humans naturally (innate way) receive and understand a large amount of information (multiple samples) [1]. Trying to replicate this process through computers with visual perception, i.e., endowing the machines the ability to see, is called computer vision [2], which is difficult due largely to variance. A vision system is required to infer objects across huge variations in pose, appearance, viewpoint, illumination, and occlusion throughout the image.

Scene understanding is one of the most challenging tasks in computer vision. It plays an essential role in many novels and future applications, e.g., autonomous driving or object detection. While humans can easily capture a scene at a glance (i.e., perceive general and specific details), it is difficult for a machine. Thus, semantic scene segmentation is one crucial step toward scene understanding, and it is one of the key challenges in computer vision. In a broader view, semantic segmentation (SS) [3] is a high-level task that makes complete scene understanding possible. The goal of scene understanding is to make machines see like humans, i.e., to have a complete understanding of visual scenes. SS aims to annotate each pixel of an image with a class label describing what this pixel represents. SS task is also called dense prediction since each pixel in the image is classified. How to extract and interpret the different levels of information details (i.e., global and local context information) is still an open problem and even more for a dense pixel-wise classification. Note that in solving the SS task, this one directly influences different applications, for example, self-drive vehicles [4], [5], segmentation on X-ray [6], detect crown on dental X-ray [7], brain tumor segmentation [8], [9], remote sensing [10]–[12], among others.

Over the last decade, computer vision has benefited from significant developments in deep learning, particularly by convolutional neural networks (CNNs) that perform remarkably on complex vision tasks such as object classification and SS. With the help of ever-growing amounts of labeled data, the deep networks can better generalize and model their context



(a) FastNet (b) DUNet (c) HRNet

(d) OCNet (e) CCNet (f) HRNet+OCR

FIGURE 1. The loss of spatial precision, often produced at the objects' boundary (red color), remains a problem in semantic segmentation.

information than the traditional methods [13]. Long et al. [14] showed that CNNs could be adapted for the segmentation task by adding upsampling layers to perform pixel-wise predictions. However, with the deep learning approach, new problems have been observed (Fig. 1) that derive from using CNNs for the specific task of semantic segmentation. These problems are [15], [16]: (i) low-resolution obtained in the output of the CNNs, and (ii) loss of spatial precision of objects within the image.

These problems are not produced by a specific operation, e.g., downsampling, but rather by a set of operations. For instance, the absence of reconstruction and refinement methods, the excessive down-sampling, the gradient vanishing, or the lack of a better extractor of feature maps [17]. Nowadays, different models [18]–[21] have addressed these problems, especially in the low resolution on the output maps. However, the loss of spatial precision problem persists and is commonly visualized at the edge of segmented objects, cf. Fig. 1. In this work, we found (see Tables 1 and 2) that to preserve spatial precision, we need mechanisms that provide general and specific details to segment, i.e., methods for better extraction, modeling, and treatment of global and local context information.

Thus, some models [15], [22], [23] use heatmap refinement with post-processing steps—e.g., conditional random field [24]—as well as models adjusting the maps from bounding boxes [25]–[27]. Although post-processing helps to have a broader-view field of objects (i.e., global features), the models lack effective global feature extractors embedded in the architecture. The encoder-decoder models [28]–[30] carry out more detailed work to adjust the heat maps by adding operations in the reconstruction, i.e., upsampling and deconvolution. These models perform local and global feature extraction at some level. However, they require a more robust combination of features than skip connections and concatenations. Some models [31]–[34] work with samples at different scales (multi-scale models) to obtain a full context of the images, i.e., global and local context information from upper and lower scales, respectively. However, multiple-size inputs make the merging process more complicated than it needs to be. In contrast, instead of resizing the inputs, models [35]–[38] increase the field of vision of the kernels (i.e., receptive field) through multiple dilated convolutions [15]. With the information from different scopes, they have the ability to tackle objects of varying sizes. However, the sampling ranges distribution (e.g., global features and context priors) cannot ensure that the information can be contained in particular ranges. That is, they have drawbacks with objects of bigger-size than the convolution kernel pyramid. Finally, models [34], [39] are emerging, focusing on smart feature extraction through attention. However, performing attention in limited regions of the image shows similar behavior to the previous models, i.e., larger-size objects escape the focus of attention.

We address the loss of spatial precision problem for SS tasks by including specific structures to extract local and global features (i.e., holistic and specific features extraction blocks). Thus, we propose a Probabilistic Hourglass Gaussian Mixture Model (PHGMM), which combines Gaussian Mixture Model (GMM) [40], [41] by assuming that the data points are generated from a Mixture-of-Gaussians and a Variational AutoEncoder (VAE) [42]. Unlike the previous models, PHGMM learns two distributions by providing structures to the internal representations (i.e., latent spaces) of the SS task given an input. The first distribution extracts the global features (i.e., global context) useful for coarse SS. In contrast, the second distribution focuses on obtaining the specific features (i.e., local context) for a fine SS. Furthermore, in contrast to the previous models, we use a single sample for the multi-scale features extraction; our distributions (i.e., our learning representation) extract information independently of the objects-size to segment, combining the information in one stage robust reconstruction. Thus, we address the loss of spatial precision problem by combining the local and global features of both latent spaces and the spatial features of the encoder through skip connections and upsampling in the decoder. Our main contributions are

- an end-to-end trainable deep model, PHGMM, that combines VAE and GMM for the features' internal representations;
- two latent spaces modeled as a GMM to extract global and local context information that jointly improve the detection of the semantic classes (i.e., holistic and specific features extraction);
- pipeline for image reconstruction by merging the different scales of context information, recovering and improving the geometric information through the decoding stage; and
- a demonstration of the advantages of using coarse and fine modeling for the image information.

II. RELATED WORK

Semantic segmentation aims at dense pixel-wise classification, i.e., assign labels to each pixel in fine-grained fashion. It is repeatedly used as an intermediate step in computer vision applications by facilitating high-level image processing and analysis. Currently, the deep segmentation approaches based on Fully Convolutional Network (FCN) have made remarkable progress. However, this architecture is inherently limited to local receptive fields and short-range contextual information.

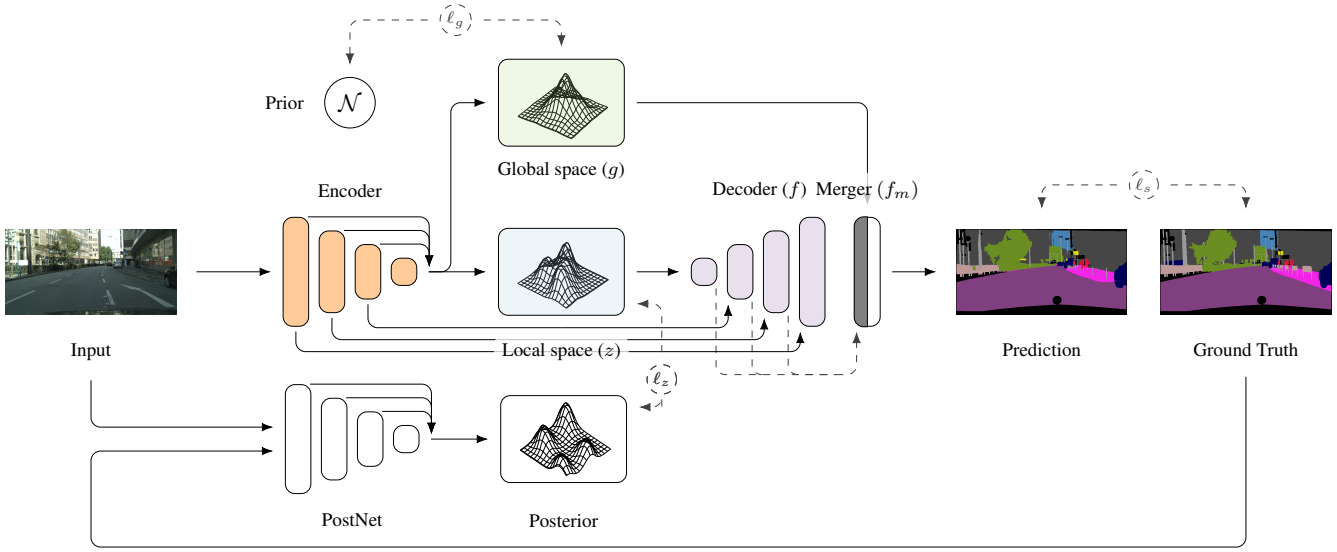


FIGURE 2. Given an input image, we extract local, z, and global, g, features (orange box) to improve the prediction of the semantic classes. We model the local features as GMM that fit a predicted posterior from the data (light blue box). For the global features, we use a single Gaussian to hold the holistic information of the image (green box). Note that we managed to recover the local features at the decoder stage by combining the GMMs of the latent space z with the encoder features through skip connections and deconvolution operations (purple box). Finally, the whole process is regularized by making the predictions for matching the ground truth, ℓ_s, and making the local and global spaces similar to their respective priors, ℓ_z, and ℓ_g, respectively.

Due to this shortage of contextual information, the FCN model suffers from two main drawbacks: low-resolution output maps and a loss in spatial precision.

Existing models [15], [22], [43] present the first effort to address these problems by refining the output maps using Conditional Random Field (CRF) [24]. However, most of the models that perform post-processing steps lack local and global feature extractors that are efficient enough to discriminate the object boundaries. In contrast, instead of performing holistic post-processing, other models [25]–[27] focus on the refinement from bounding boxes, i.e., they obtain a segmentation heat map in each region of interest.

However, the transfer of information (context) remains limited. Recent works [17], [35], [36], [44] verify the importance of context information in SS. How to extract, merge, and employ this context information is the next step in SS. Thus, encoder-decoder models, such as U-Net [28], DeconvNet [29], SegNet [30], or DUNet [45], add a decoder stage (i.e., a set of deconvolution and unpooling operations) with skip connection, managing to merge, at some level, local context from low-level features and global context from high-level ones. For instance, ParseNet [46] adds global information to layers through the global average pooling operation, and FC-DenseNet [47] combines knowledge by concatenating all the features of the previous output layers. Although the encoder-decoder models show to focus on details, they still require additional detailed connections to improve their information reconstruction stage.

Although theoretically, in the layers with lower resolution (e.g., size of $H/32 \times W/32$), there is a greater receptive field; empirical experiments showed that these fields are significantly smaller and are not enough to capture global context information (i.e., global features) [36], [46]. One way to address these drawbacks is through multi-scale models [32], [34], [48] (e.g., ICNet [33] is fed with different input sizes). DeepLabv2 [15], DeepLabv3 [35], DeepLabv3++ [37], DecoupleSegNets [49] and Gated-SCNN [50] preserve the spatial size of the features maps by proposing Atrous Spatial Pyramid Pooling (ASSP). ASPP employs a set of atrous convolutions operations [51] and average pooling to capture several context information (from local to global features). However, dilated convolutions can cause grinding problems [52], [53]. It can induce a loss of local information in the models and capture irrelevant information on a large scale attention models: A different approach is presented by PSPNet [36] and ESPNet [54], operating not on the convolution kernels but on sub-regions of the feature maps generated at different levels. They use the Pyramid Pooling Module (PMM) to get global and local context information by reducing the spatial size of features. Also, HRNetv2 [55] and HRNet+OCR [56] perform multi-scale feature extraction by sharing feature maps across different branches (scales), i.e., broadcasting context information at various resolutions. Nevertheless, how and where to combine multi-scale information still represents a challenge.

Although the previous context fusion models help capture different scales' features, The relationship between objects in a global view (essential to SS) is still limited. Thus, attention mechanisms [57]–[60] can use to extract long-range contextual information. Furthermore, CCNet [44] efficiently makes a feature from any position perceive the other features, i.e., it adds contextual information in horizontal and vertical directions with sparse attention. In contrast, DANet [61] proposes two attention modules (dual attentions), focused

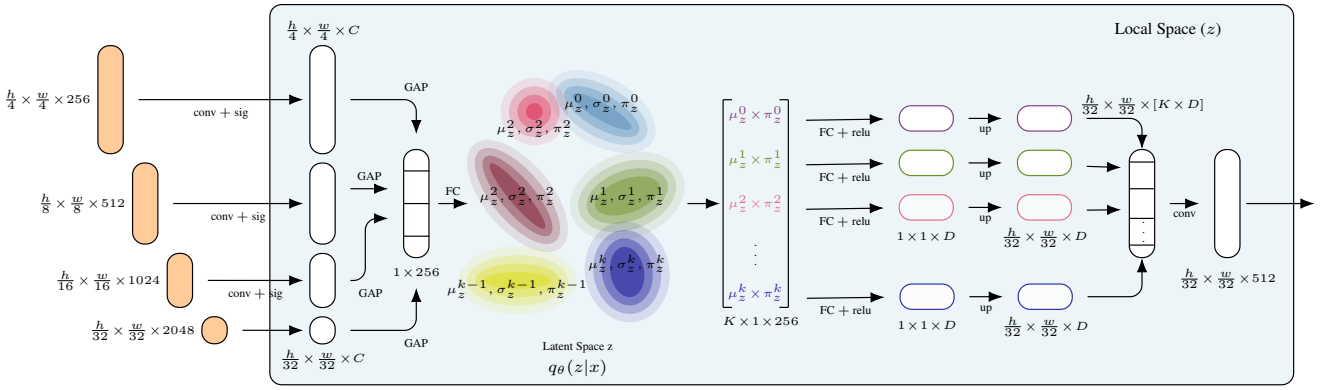


FIGURE 3. Local Space diagram. Note, the blocks and arrows represent the feature vectors and operations, respectively. We extract features at different scales through convolution operations with a sigmoid activation function. Next, we use the Global Average Pooling operation (GAP) to summarize the information and gather it into a concatenated feature vector. To model the latent space z through a GMM, we use the Fully Connected layer (FC) to extract K different Gaussian mixtures. Next, the mean of each Gaussian is used and stored in a list, and apply an FC with ReLU is to do further feature processing. Finally, we return to our initial vector size (i.e., $\frac{h}{32} \times \frac{w}{32}$). We concatenate the K Gaussians processed and perform the last convolution operation to fuse all the information from the different Gaussians. Although we compress the information using GAP to model the Gaussians distribution, the information contained in the latent space z plays an important role in adequately recovering the geometric information (background and contour of the segmented objects) through the decoder stage.

on spatial features and channels. Finally, OCNet [62] extracts global and local context by grouping (by permutation), splitting, and operating each sub-region with a self-attention module (i.e., fully matrix).

Unlike previous models, PHGMM extracts local and global context information from two specific latent spaces, which we provide an adequate structure and behavior to obtain this context information precisely. In addition to providing a decoding stage capable of recovering the geometric information of the objects (background and boundary) by combining both context information (local y global) with the decoder information through skip connections, deconvolutions, and upsampling.

III. GLOBAL AND LOCAL FEATURE MODELING

In segmentation, global features can generalize entire objects or a set in a compact representation, while local features, on the other hand, compute representations for particular (more focused) parts of the image. However, due to the sensitivity to occlusion in global features or noise in local ones, it is not straightforward to combine both features reliably [2]. Consequently, we propose a model capable of merging these local and global features in a suitable manner. Our contribution comes from combining the features in two stages. First, we decode the local features representation, and then add the global ones once the image structure has been recovered in the decoder. We recover structural information, i.e., background and contour of the objects, by merging local and global features (latent spaces) with decoder features through skip connections and upsampling in the decoding stage. Thus, we obtain diverse types of contextual information, reducing overfitting and addressing the loss of spatial precision problem.

Current semantic segmentation models work by extracting features and pooling them to reduce the representation dimension. Then, while decoding the encoded features, these models pass part of the encoded information to aid the decoder in producing better features. In a way, these models use local features and share them across the encoding and decoding tasks to improve the final prediction [29]. Differently, we propose to use local and global features to improve the classes' representation by adding information to the shared knowledge between these two stages. Our proposal is to create a global model that holds the information of macro objects of the scene, while using the traditional local features, as well as the helpful combination of these and encoder features; through the decoder. We propose to use Gaussian Mixture Models to represent both spaces (i.e., global and local), since we need to hold multi-modal representations due to the different classes present in a scene (cf. Fig. 2).

Our objective is to model two latent spaces to perform a better and more robust extraction of local (Fig. 3) and global (Fig. 4) features and an effective way to combine them through the decoder (Fig. 5) to recover the geometric information of the objects. Thus, we propose using an end-to-end model that fits and employs two latent spaces suitable for the segmentation task. In order to provide an internal structure to the learned representation (i.e., latent space), we merge VAE with ResNet-101 [63] (for an encoder-decoder stage for SS). A VAE allows us to represent the latent space through a structure, i.e., a predefined distribution. Furthermore, the ResNet-101 backbone provides robust processing (e.g., residual block) to extract the basic features that will turn into the distributions. As a result, our model obtains a local context for the discrimination of segmented objects (i.e., class-level information) as a mixture of Gaussians. The model mixes the structure of the local-global features by employing concatenation operation and upsampling throughout the deconvolution stage. In addition, the decoder stage is responsible for recovering the geometric information of the objects. Combining the context information (local-global structure features) previously extracted with the features coming from the encoder through skip connections in the

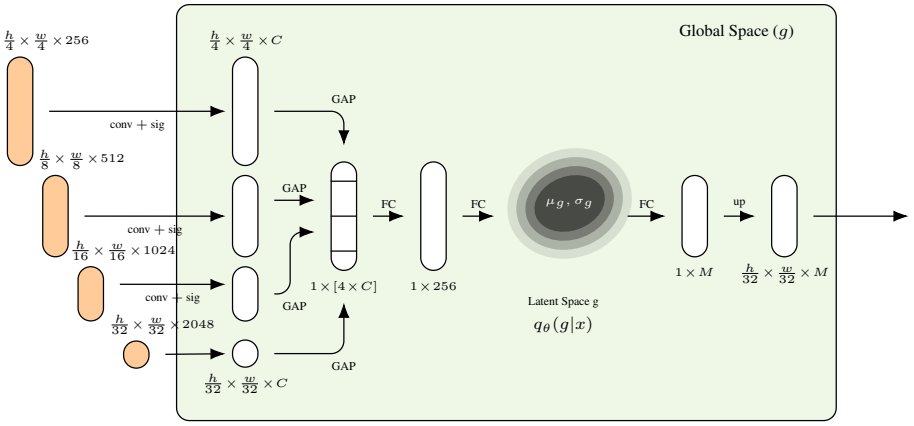


FIGURE 4. Global Space diagram. The blocks and arrows represent the feature vectors and operations, respectively. First, we extract features at different scales through convolution operations with a sigmoid activation function. Next, we use the Global Average Pooling operation (GAP) to summarize the information and gather it into a concatenated feature vector. Subsequently, we carry out two Fully Connected layers (FC) for a better features extraction, obtaining our Normal Gaussian. Then another FC transforms the mean of the Gaussian into a vector feature that is scaled to $\frac{h}{32} \times \frac{w}{32}$.

different scales. Integrating these two latent spaces provides the neural network with a powerful representation (cf. Fig. 6). The latent spaces provide complementary information that improves the semantic segmentation (cf. Tables 1 and 2).

We present an overview of our PHGMM model in Fig. 2, and detail the inference and generation processes in Sections III-A and III-B, respectively. Also, in Section IV-A, we split the description of the operations used for local and global context extraction. The local and global extraction processes are shown in Figs. 3 and 4, respectively. Fig. 5 shows that our decoder is in charge of intelligently merging the previously extracted features so that PHGMM recovers the geometric information of the segmented objects addressing the problem of loss of spatial precision (see Table. 7). Finally, in Section IV-B, we describe how our model is trained.

A. INFERENCE PROCESS

We jointly model the global information of the scene g and the local features z of the image x and assume that they are independent. We set the distribution that models this relation as $q_\theta(z, g \mid x) = q_\theta(z \mid x) q_\theta(g \mid x)$. The local distribution

$$q_\theta(z \mid x) = \sum_{k=1}^{K} \pi^k(x) \mathcal{N}\left(z; \mu_z^k(x), \sigma_z^k(x)^2 \mathbf{I}\right) \qquad (1)$$

is a mixture of Gaussians that models K clusters of related features that are helpful in obtaining different classes for the segmentation. The functions $\mu_z^k(\cdot)$, $\sigma_z^k(\cdot)$, and $\pi^k(\cdot)$ are the parameters of the k-th component of the mixture and depend on the given image x (see Fig. 2). All are approximated through a ResNet-101 backbone with separate projection heads that output the corresponding parameter. On the other hand, the global distribution

$$q_\theta(g \mid x) = \mathcal{N}\left(g; \mu_g(x), \sigma_g(x)^2 \mathbf{I}\right) \qquad (2)$$

is a single Gaussian that models a global context that holds all the image x information. The parameter functions $\mu_g(\cdot)$ and $\sigma_g(\cdot)$ are computed in a similar way as the previous parameters.

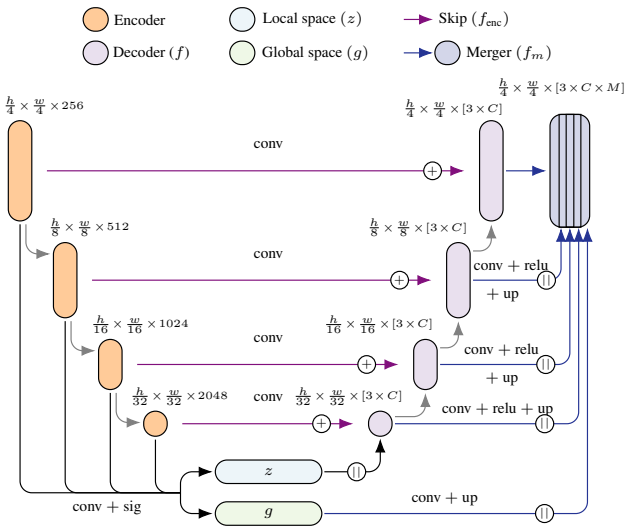


FIGURE 5. The decoding diagram is fed by the previously processed feature vector (i.e., local space z). Next, the decoder takes the skip connections, deconvolution, and upsampling from the encoding. Finally, in its last layer, it concatenates the information of the different levels (i.e., multi-scale information) with the global information (i.e., global space g). Note, we call the latent space g a global representation because this space summarizes features of objects in a holistic way (i.e., a global view of the object). Also, our decoding stage is in charge of merging the different context information extracted at the different levels of our network. This smart fusion recovers the geometric information of the segmented objects.

The local features z hold information about the features, while the global ones g serve as a latent variable to model holistic information from the whole scene. Additionally, due to the way we engineer the decoding stage (see Fig. 2), the global features also serve to spread the gradient back to the encoder.

B. GENERATIVE PROCESS

To obtain the classes y for the segmentation, we model the joint probability $p(x, y, z, g)$ of the data, the images x and the labels y, and the latent variables, the local z and global g. We factorize the joint as $p_{\theta,\phi}(x, y, z, g) = p_\phi(y \mid z, g)p_\theta(z \mid x)p_\theta(g \mid x)p(x)$.

To implement the generative process from $p(y|z, g)$, we use the decoder $f(\bar{\mu}_z)$, where $\bar{\mu}_z = \|_{k=1}^{K} \mu_z^k$ is the concatenation of all the means from the mixture. Later, a merge function $f_m(f(\bar{\mu}_z), \mu_g, f_{\text{enc}})$ combines the partially decoded segmentations with the parameters from the global context and the encoder features, f_{enc}, from the skip connections. Hence, the predicted labels are $y' = f_m(f(\bar{\mu}_z), \mu_g, f_{\text{enc}})$. For a better interpretation, we present the associated diagram in Fig. 5, in later sections.

IV. IMPLEMENTATION

In this section, we explain in detail the PHGMM architecture, i.e., local space z and global space g, as well as the operations involved in modeling latent spaces. In addition, we explain how to merge the different local-global context information and the encoder features to recover the details of the segmented objects (i.e., the geometric information). Finally, we perform the loss functions used to adjust the PHGMM parameters and show the hyperparameters used for our model.

A. ARCHITECTURE

We use a ResNet-101 as a backbone for the PHGMM model due to its good behavior, i.e., the trade-off between good feature extraction and the number of parameters used, that is, most of the residual blocks and depths of the layers. The samples that PHGMM receives as input are passed through convolution with kernel 7×7 and depth 64 to carry out the information compression, thus converting the samples in size $\frac{h}{4}\times\frac{w}{4}\times64$. Next, we use a defined decoder in four groups of blocks of depth 256, 512, 1024, and 2048 respectively. Each group presents a set of 3, 4, 23, and 3 internal residual blocks. Notice that our feature vector shows a reduction in the size of $(\frac{h}{4}\times\frac{w}{4})$, $(\frac{h}{8}\times\frac{w}{8})$, $(\frac{h}{16}\times\frac{w}{16})$, and $(\frac{h}{32}\times\frac{w}{32})$ as it passes through the decoder.

One of the issues we address in this paper is how to take advantage of this multi-scale information to improve the edges of the segmented objects and thus address the problem of the loss of spatial precision. To address this, we create local space z (Fig. 3) to model the latent space through a mixture of Gaussians, focusing on local features and global space g (Fig. 4) to extract global information through Gaussian modeling of a second latent space. Additionally, we use skip connections to pass complemental multi-scale information from the encoder to the decoder.

Thus, in local space z (see Fig. 3), we use 3×3 kernel convolutions together with a non-linear sigmoid function to compress the information of the different levels, reducing the depth from 256, 512, 1024, and 2048 for the number of classes (C). We concatenate the feature vectors to merge the multi-scale information through Global Average Pooling (GAP). To model the latent space z, we use a Fully Connected layer (FC) and produce K mixtures of Gaussians (i.e., μ_z^k, σ_z^k, π_z^k). Note that the concatenation vector and the Gaussian parameters (i.e., μ, σ) have a size of 1×256 (chosen empirically). We combine the local information from the GMMs by stacking the K vectors (i.e., $\mu_z^k\times\pi_z^k$). For each item in the stack, we perform an FC with a non-linear function RELu, transforming the features of size 1×256 to $1\times1\times D$. Then we scale the vectors achieving $\frac{h}{32}\times\frac{w}{32}\times D$. Finally, we perform a non-linear combination on the concatenation vector of size $\frac{h}{32}\times\frac{w}{32}\times[K\times D]$, reaching a depth of 512. These feature vector locales will be used as input for our encoder. Note that even though we use GAPs in the process of converting the samples to Gaussian distributions, we call local latent space to the local space z because, from this feature vector, the geometric reconstruction is performed to recover and refine the details of the edges of segmented objects.

On the other hand, the global space g (see Fig. 4) performs the treatment of multi-scale information to a certain extent similar to the local space z, through the concatenation (of size $1\times[4\times C]$) of feature vectors summarized by GAP operations. Next, we use two FC to model the latent space g and give it the structure of a Normal Gaussian with parameters μ_g and σ_g, both of size 1×256 (chosen empirically). Finally, the information is combined by taking μ_g and passing it through an FC, obtaining a vector of global features of size $1\times M$, followed by an upscale operation achieving a vector size of $\frac{h}{32}\times\frac{w}{32}\times M$. Note that we use the latent space g as a global representation because g is in charge of extracting the holistic information of the objects and then concatenates it in the last reconstruction layer of the decoder.

We merge both latent spaces (z and g), and decoder features through the decoding stage. The decoder (see Fig. 5) obtains as input the latent space z and has four groups of residual units, where each group has 3, 23, 4, and 3 residual units, all with $3\times C$ depth and a feature vector of size $\frac{h}{32}\times\frac{w}{32}$, $\frac{h}{16}\times\frac{w}{16}$, $\frac{h}{8}\times\frac{w}{8}$ and $\frac{h}{4}\times\frac{w}{4}$ respectively. In each residual unit group of the decoder, our PHGMM model performs a geometric information retrieval by combining the local features coming from the GMM (latent space z) with the features extracted in the encoding through an addition operation. We transform the encoder features through nonlinear combination, i.e., convolution operations and non-linearity functions, using a 3×3 kernel with depth C. These features are passed from the different levels of the encoder to the decoder (i.e., multi-level information) with the use of skip connections. In the last layer, we concatenate the decoder's multi-scale reconstruction information through convolution operations with kernel 3×3 and a non-linearity function RELu and upscale applied to each level (i.e., group) of the decoder. Besides, a convolution operation of 3×3 with upscale to latent space g. The concatenation of all this information gives a resulting feature vector of size $\frac{h}{4}\times\frac{w}{4}\times[3\times C\times M]$. Finally, for the output of PHGMM, a convolution operation with kernel 8×8 and a Softmax are applied to combine all the information

and obtain our final heat map.

B. TRAINING

The target function used to train our PHGMM model $\mathcal{L}$, per data sample, is a linear combination of the specific loss functions ℓ_g, ℓ_z, and ℓ_s functions, defined by

$$\mathcal{L} = \lambda_g \ell_g + \lambda_z \ell_z + \lambda_s \ell_s, \tag{3}$$

where ℓ_g and ℓ_z are bound to provide a useful structure (i.e., helpful embedding) for the latent spaces, and ℓ_s on the loss function $\mathcal{L}$ focuses on dense pixel-level classification. Also, $\lambda_{g,z,s}$ are hyperparameters of each term, respectively. Note, to train with a batch of data, we aggregate the losses over it.

For learning the global representation g, we need to find a relevant embedding for the segmentation task (i.e., find useful low-level features). Consequently, we use Kullback-Leibler divergence (KL) [69] to provide a structure to g w.r.t. its prior, i.e.,

$$\ell_g = \mathrm{KL}\big(q_\theta(g \mid x) \,\big\|\, p(g)\big). \tag{4}$$

The KL works as a regularizer over g and measures the divergence between the encoder distribution $q(g \mid x)$ and the prior $p(g)$. We specify $p(g)$ as a standard Normal distribution $p(g) = \mathcal{N}(g; 0, \mathrm{I})$, and define $q(g \mid x)$ as before (2).

Our second term provides the latent space z with a clustering behavior. Instead of using a default prior, we infer a conditional $p(z \mid x, y)$ from data with another net, called PostNet (see Fig. 2). This PostNet uses the ground truth labels and image data to produce the posterior distribution while training to serve as regularizer. The posterior is also a mixture of Gaussians defined by

$$p_\gamma(z \mid x, y) = \sum_{k=1}^{K} \pi_{z^*}^k(x, y) \mathcal{N}\left(z; \mu_{z^*}^k(x, y), \sigma_{z^*}^k(x, y)^2 \mathbf{I}\right). \tag{5}$$

We fit the predicted distribution with the infer one through

$$\ell_z = \mathrm{KL}\big(q_\theta(z \mid x) \,\big\|\, p_\gamma(z \mid x, y)\big). \tag{6}$$

Recall that our PHGMM model learns to densely predict the semantic classes given an input image x. Thus, the reconstruction loss function is expressed as a cross-entropy loss between the prediction y' and its ground-truth y. Penalizing the pixel-wise prediction for each class is not enough. Thus, we penalize the segmented objects' contours as well. We employ the loss function soft intersection-over-union [70]. So our third term, ℓ_s, is

$$\ell_s = -\frac{1}{C}\sum_{c=1}^{C} y_c \log(y_c') + 1 - \frac{1}{C}\sum_{c=1}^{C}\left(\frac{y_c' \cap y_c}{y_c' \cup y_c}\right), \tag{7}$$

where y_c' is our prediction of the ground truth y_c for the class c.

Finally, We provide an algorithm in Alg. 1 to better understand the training framework, which summarizes the PHGMM steps process.

Algorithm 1 Training the PHGMM model

$\theta_{\text{Enc}}, \phi_{\text{Dec}}, \gamma_{\text{PosNet}} \leftarrow$ Initialize networks parameters
repeat
  $X \leftarrow$ mini-batch image from dataset
  $Y \leftarrow$ mini-batch label from dataset
  $X_1', X_2', X_3', X_4' \leftarrow \text{Encoder}(X)$ ▷ Multi-level features
  $Z, q_\theta(z|x) \leftarrow \text{LocalSpace}(X_1', X_2', X_3', X_4')$
  $p_\gamma(z|x, y) \leftarrow \text{PostNet}(X, Y)$
  $G, q_\theta(g|x) \leftarrow \text{GlobalSpace}(X_1', X_2', X_3', X_4')$
  $\ell_z \leftarrow \mathrm{KL}\big(q_\theta(z \mid x) \,\big\|\, p_\gamma(z \mid x, y)\big)$ ▷ Eq. 6
  $\ell_g \leftarrow \mathrm{KL}\big(q_\theta(g \mid x) \,\big\|\, p(g)\big)$ ▷ Eq. 4
  $X_1'', X_2'', X_3'', X_4'' \leftarrow \text{Decoder}(Z)$ ▷ Multi-level features
  $Y' \leftarrow$ convolution of $[X_1'', X_2'', X_3'', X_4'', G]$
  $\ell_s \leftarrow \text{CrossEntropy}(Y', Y) + \text{IoU}_{\text{soft}}(Y', Y)$ ▷ Eq. 7
  $\mathcal{L} = \lambda_g \ell_g + \lambda_z \ell_z + \lambda_s \ell_s$ ▷ Eq. 3
  ▷ Update parameters according to gradients
  $\theta_{\text{Enc}} \xleftarrow{+} -\nabla_{\theta_{\text{Enc}}}(\ell_z + \ell_g + \ell_s)$
  $\phi_{\text{Dec}} \xleftarrow{+} -\nabla_{\phi_{\text{Dec}}}(\ell_g + \ell_s)$
  $\gamma_{\text{PostNet}} \xleftarrow{+} -\nabla_{\gamma_{\text{PostNet}}}(\ell_g + \ell_z)$
until static stopping criterion (epoch = 100) ▷ Number stactic of epochs

C. SETUP

The ResNet inspires PHGMM's architecture because it presents an extensive feature extraction process in the encoding stage. The encoder has four blocks of depths 256, 512, 1024, and 2048. The convolution, atrous convolution operations, and ReLU activation functions are performed within each block. Because of our limited computational resources, we reduce our input image by $\frac{1}{4}$. We use pooling and convolution with a stride size of two to achieve it. To produce the parameters of the latent spaces, we concatenate the output of each block. Thus, we use convolution and a Sigmoid function to adjust the depth (i.e., number of classes). Then, by mean reduction over the height and width, we obtain a feature vector of each block. We concatenate all of them and pass these through a Fully Connected layer (FC) producing our parameters (μ_* or σ_*) in size $1 \times 1 \times 256$ (empirically chosen). Note, we use different parameters (weights) in FC to infer each cluster k (i.e., μ_z^k, σ_z^k). The same configuration is also applied for PostNet.

In the decoder, we concatenate the μ_z^k of every K clusters. Then, we use FC to bring our GMM vector (from z) from 1D to 2D and feed our decoder with it. The decoding stage presents operations similar to the encoder adding a bilinear interpolation operation. Moreover, it also has four blocks, all with depth equal to the number of classes. Note that in each block, the PHGMM model retrieves the geometric information (background and borders of the objects) by merging the local-context features z with the convolution operations, upscale, and the features coming from the encoder through skip connections. Finally, we add a vector of specific features from μ_g (passed through FC operations and bilinear interpolation) to the decoder output. In the end, we perform two last convolution operations followed by the softmax function. Furthermore, we set our hyperparameters $\lambda_{g,z,s}$ to 1.1, 0.4, and 0.4, respectively. Finally, we use a static stopping

TABLE 1. IoU results on Cityscapes validation set for semantic segmentation, using 11 classes and with crop size of 384×768. The classes are: Sky, Building, Road, Sidewalk, Fence, Vegetation, Pole, Car, Sign, Person, Cyclist.

Model												mIoU (%)
SegNet [30]	73.74	79.29	92.70	59.88	13.63	81.89	26.18	78.83	31.44	45.03	43.46	56.92
DeconvNet [29]	89.38	83.08	95.26	68.07	27.58	85.80	34.20	85.01	27.62	45.11	41.11	62.02
DeepLab v2 [15]	74.28	81.66	90.86	63.30	26.29	84.33	27.96	86.24	44.79	58.89	60.92	63.59
FCN8 [14]	76.51	83.97	93.82	67.67	24.91	86.38	31.71	84.80	50.92	59.89	59.11	65.43
FastNet [64]	77.69	86.25	94.97	72.99	31.02	88.06	38.34	88.42	52.34	61.76	61.83	68.52
ParseNet [46]	90.76	85.20	92.01	63.49	39.00	88.21	46.82	89.20	61.90	63.91	62.73	71.20
ESPNet [54]	91.79	86.36	95.73	71.84	48.52	88.44	49.06	87.29	54.60	61.83	57.51	72.09
FC-DenseNet67 [47]	92.19	86.77	96.60	75.40	41.55	88.07	52.92	87.09	63.89	60.48	52.92	72.54
BiSeNet [65]	91.64	87.42	96.48	75.41	44.05	89.07	39.50	89.34	58.63	66.81	63.08	72.86
ENet [31]	91.63	87.48	96.44	75.34	48.44	89.23	43.14	89.24	56.04	65.13	63.70	73.26
DeepLab v3 [35]	92.82	89.02	96.74	78.13	41.00	90.81	49.74	91.02	64.48	66.52	66.98	75.21
PSPNet [36]	91.94	89.93	96.94	78.37	53.64	90.19	43.47	92.12	64.40	70.71	70.94	76.61
DANet [61]	92.25	90.26	97.25	79.95	51.33	90.60	45.20	92.50	66.38	71.47	71.25	77.13
AdapNet++ [66]	93.07	89.46	97.06	80.03	49.46	90.58	52.10	92.22	66.26	72.88	70.62	77.61
CCNet [44]	90.97	89.01	96.59	77.36	42.49	91.36	58.24	91.22	71.18	74.80	70.57	77.62
OCNet [62]	92.72	90.73	97.39	80.80	54.58	90.86	45.60	92.64	67.35	71.81	71.98	77.86
CGBNet [67]	92.98	89.40	96.66	77.60	42.80	91.88	57.52	91.14	73.29	75.26	71.18	78.16
DUNet [45]	93.33	91.05	97.28	80.18	55.15	91.35	53.70	92.89	68.33	73.33	72.29	78.99
HRNet [55]	94.56	90.98	97.48	82.46	50.27	92.35	61.57	93.96	73.14	78.55	75.44	80.98
HRNet + OCR [68]	94.44	91.57	97.59	83.00	**55.46**	92.49	62.68	93.98	**75.67**	78.70	75.35	81.90
PHGMM	**94.63**	**91.60**	**97.70**	**83.65**	55.27	**92.52**	**62.71**	**94.20**	75.35	**79.21**	**76.45**	**82.12**

TABLE 2. IoU results on Synthia validation set for semantic segmentation, using 11 classes and with crop size of 384×768. The classes are: Sky, Building, Road, Sidewalk, Fence, Vegetation, Pole, Car, Sign, Person, Cyclist.

Model												mIoU (%)
SegNet [30]	91.90	87.19	83.72	80.94	50.02	71.63	26.12	71.31	1.01	52.34	32.64	58.98
FCN8 [14]	92.36	91.92	88.94	86.46	48.22	77.41	36.02	82.63	30.37	57.10	46.84	67.11
DeconvNet [29]	95.88	93.83	92.85	90.79	66.40	81.04	48.23	84.65	0.00	69.46	52.79	70.54
ParseNet [46]	93.80	93.09	91.05	88.98	53.22	79.48	46.15	85.37	36.00	63.30	50.82	71.02
DeepLab v2 [15]	94.07	93.34	88.07	88.93	55.57	80.22	45.97	85.87	38.73	64.40	52.54	71.61
FC-DenseNet67 [47]	92.74	89.94	83.35	85.72	74.47	70.47	51.13	84.21	35.11	68.78	54.54	71.86
FastNet [64]	92.21	92.41	91.85	89.89	56.64	78.59	51.17	84.75	32.03	69.87	55.65	72.28
ESPNet [54]	95.44	93.16	91.52	89.62	65.98	80.11	48.09	84.93	39.62	66.80	53.91	73.56
BiSeNet [65]	95.04	92.97	91.25	89.08	63.72	79.67	48.44	84.77	45.68	65.89	54.01	73.68
ENet [31]	94.81	93.01	91.51	89.67	65.56	79.04	50.78	85.22	42.09	67.67	54.14	73.95
ICNet [33]	95.78	94.32	92.62	90.76	64.91	83.17	54.37	87.53	51.15	68.03	56.47	76.28
DeepLab v3 [35]	95.30	92.75	93.58	91.56	73.37	80.71	55.83	88.09	44.17	75.65	60.15	77.38
DANet [61]	96.93	96.00	95.22	93.82	70.73	87.36	64.01	92.67	61.40	76.19	65.70	81.82
PSPNet [36]	96.92	96.00	95.27	93.85	71.17	87.30	65.20	92.98	62.73	76.62	66.24	82.21
CGBNet [67]	97.07	96.21	95.45	94.06	71.59	87.91	66.28	93.27	64.46	77.09	66.97	82.76
AdapNet++ [66]	97.09	96.24	95.47	94.09	71.79	87.97	66.23	93.33	64.92	77.16	67.21	82.86
CCNet [44]	97.22	96.17	95.45	93.88	72.14	88.36	66.74	93.19	67.74	76.88	67.70	83.22
OCNet [62]	97.39	**96.50**	95.69	94.37	72.49	**88.84**	67.50	93.64	67.92	77.59	67.69	83.60
HRNet [55]	96.73	94.76	94.83	93.95	87.00	83.62	72.81	91.90	61.74	85.31	71.08	84.88
HRNet + OCR [68]	96.74	94.83	94.30	93.47	**87.80**	84.33	**75.36**	92.54	62.78	**87.14**	**72.88**	85.65
DUNet [45]	**97.57**	96.48	96.11	95.05	80.57	88.09	73.95	93.71	67.48	82.38	71.10	85.68
PHGMM	97.01	95.89	**96.23**	**95.17**	85.03	84.91	71.78	**94.26**	**68.26**	85.50	72.42	**86.04**

criterion when training our model. We define a static number of epochs (one hundred epochs) to stop the training.

V. EXPERIMENTS

In this section, we introduced datasets for SS used in the training and testing of our PHGMM model and the evaluation metrics used to compare the different existing models. We use the dataset with reduced classes proposed by Valada et al. [66]. It is due to our limited computational resources. To make the comparison fair (see Table 1 and 2), we train all models from scratch and each model with the same number of classes and same resolution, and use their respective hyperparameters from each model.

A. DATASETS

We evaluated several types of urban/forest scenarios datasets. They are Cityscapes [71] and Synthia [72].

Cityscapes: The dataset has 5000 samples. However, we used crop for augmenting the training set (i.e., transformations of contrast, brightness, and horizontal flip) to generate 17 500 samples. For comparison with other models, we employ the original validation set with a resolution of 768×384 (resize). To facilitate the models comparison, AdapNet++ [73] proposed the class reduction produced by combining some types of classes. Thus, we use 11 classes: *sky, building, road, sidewalk, fence, vegetation, pole, car/truck/bus, traffic sign, person, rider/bicycle/motorbike, and background.*

Synthia: This dataset contains realistic photo images from a virtual city. We use the original 9000 samples (7000 for training and 2000 testing) resized to 768×384 resolution. The classes of object labels are the same as the Cityscapes mentioned above label set.

B. EVALUATION METRICS

To evaluate our results on segmentation, we chose accuracy and intersection-over-union metrics as validation measures [70]. The intersection-over-union (IoU) is defined by

$$\text{IoU} = \sum_{i}^{N} \frac{P_i \cap Y_i}{P_i \cup Y_i} = \sum_{i}^{N} \frac{TP_i}{TP_i + FP_i + FN_i}, \quad (8)$$

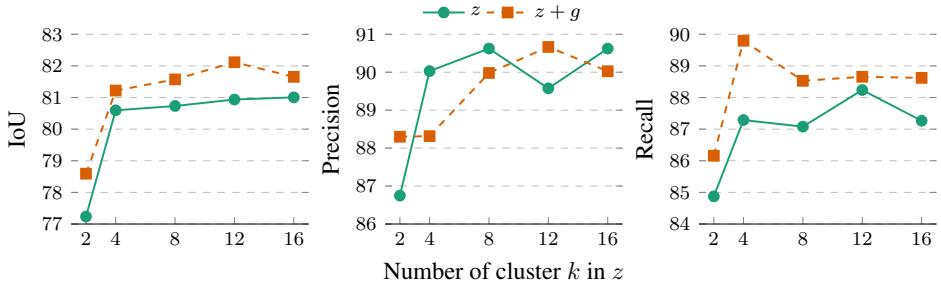


FIGURE 6. Ablation study on the latent spaces z (number of clusters k in the GMM) and g (include or exclude from the model) using the Cityscape validation dataset.

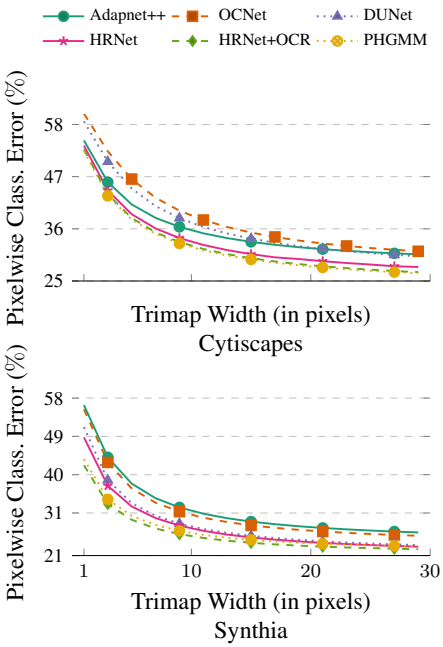


FIGURE 7. Pixelwise classification error (lower is better) vs. trimap width (to measure the pixel boundary labeling accuracy) for AdapNet++ [66], OCNet [62], DUNet [45], HRNet [55], HRNet+OCR [68], and our PHGMM on the validation datasets.

the precision (Prec) is

$$\text{Prec} = \sum_{i}^{N} \frac{TP_i}{TP_i + FP_i}, \tag{9}$$

and the recall (Rec) is

$$\text{Rec} = \sum_{i}^{N} \frac{TP_i}{TP_i + FN_i}. \tag{10}$$

We assume that P_i is the set of pixels predicted as the ith class, Y_i is pixels set belonging to the ith class, and N is the number of classes. Besides, TP_i, FP_i, TN_i, and FN_i represent True/False Positives and True/False Negatives, respectively, for a given class i. Note, these metrics are widely used in SS

We use clustering metrics to measure the latent space behavior. Consequently, we utilize the Silhouette Coefficient, SSI [74] defined by

$$\text{SSI} = \sum_{i}^{N} \frac{b_i - a_i}{\max\{a_i, b_i\}}, \tag{11}$$

where a_i, b_i are the mean intra-cluster and nearest-cluster distances from i, respectively. The Calinski-Harabasz Index, CHI [75] is given by

$$\text{CHI} = \frac{\text{SS}_M}{\text{SS}_W} \frac{N-k}{k-1}, \tag{12}$$

where k is the number of clusters, and N is the samples, SS_W and SS_M are the overall within-cluster and between-cluster variances, respectively. Finally, the Davies-Bouldin Index, DBI [76] denoted by

$$\text{DBI} = \frac{1}{k} \sum_{i}^{k} \max_{j \neq i} \left(\frac{s_i + s_j}{d_{ij}} \right), \tag{13}$$

where s_i is the average distance between each point of cluster i and its centroid, and d_{ij} is the distance between cluster centroids i and j. Note, high values are better for SSI and CHI, while low ones are better for DBI.

C. COMPARISONS

We present a set of experiments (quantitative and qualitative) to compare our PHGMM model with existing models in the literature. For these, we employed Cityscape and Synthia datasets presented in the previous section. We present the quantitative results in Tables 1 and 2. We use the IoU metric (higher is better) for each class on both datasets. Note that to perform the comparison, we train all the models from scratch, with the same sample size and the same number of classes. Due to our limited computational resources, we use the sample size (i.e., 768×384) and the number of classes (i.e., 11 target classes) defined in previous works by Valada et al. [77].

We present the qualitative results in Fig. 8. The results are displayed on a set of random samples from the validation set. These figures show the inferences of the models with the best segmentation performance (i.e., IoU). We present the predictions of the different models named in the columns. These predictions show each dataset's respective color (label of classes) for Cityscape and Synthia and are displayed in the odd rows named output. The even rows named comparison

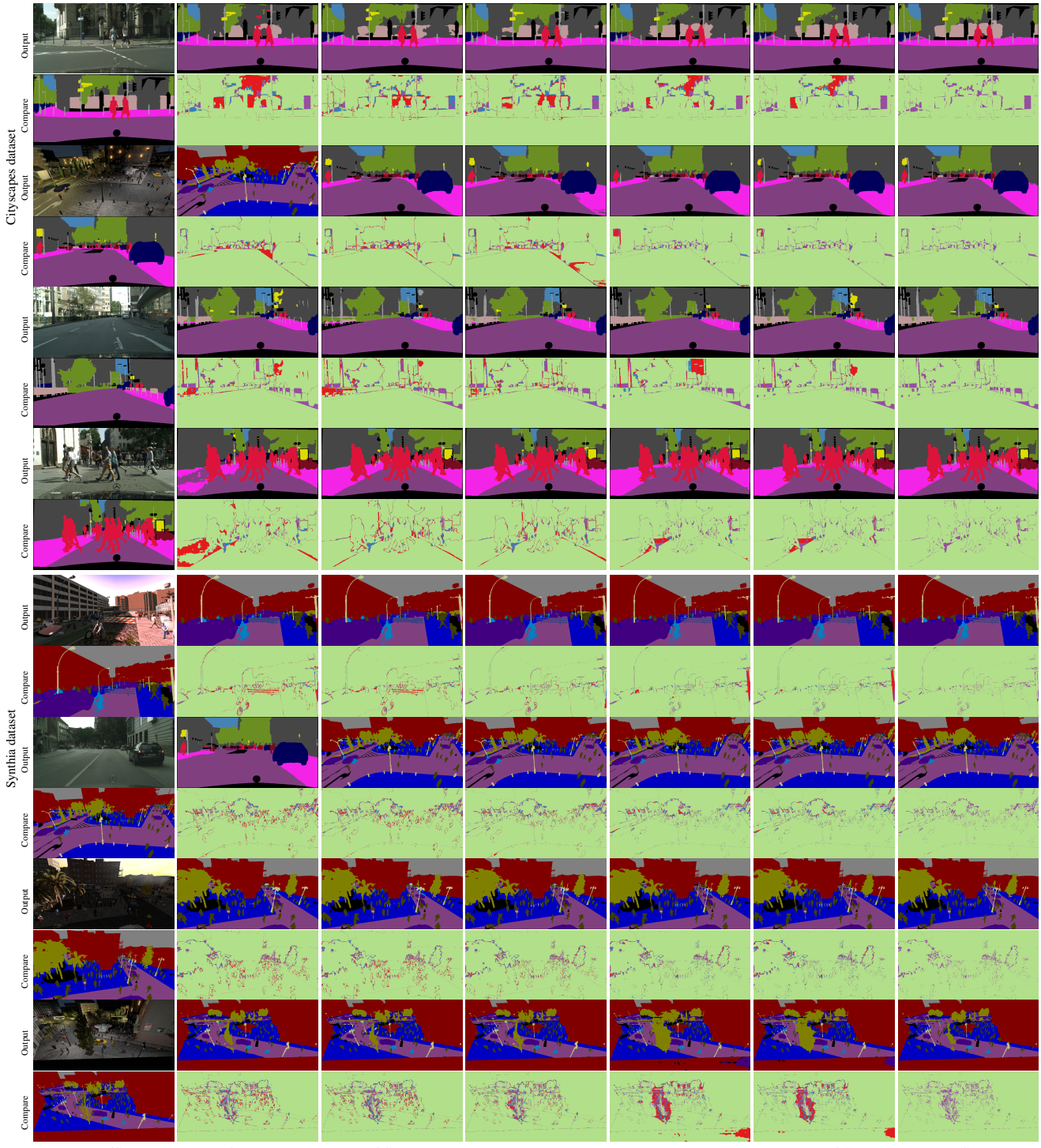


FIGURE 8. Qualitative comparison of the predictions of the AdapNet++ [66], OCNet [62], DUNet [45], HRNet [55], HRNet+OCR [68], and PHGMM models against the ground-truth. The samples are chosen randomly from the validation set. The prediction rows (output) show the outputs of the different models with their respective labels (colors) of the datasets. The comparison row shows an overlap of the predictions (PHGMM vs. the model indicated in the column) against the ground truth. The first column is PHGMM vs. AdapNet++; the second one is PHGMM vs. OCNet++, and so on. In the comparison images, the green regions are correctly segmented, the red regions are erroneously segmented by original models (AdapNet++, OCNet, DUNet, HRNet, and HRNet+OCR) while the blue ones are errors made by PHGMM. Regions incorrectly segmented by both predictions are purple.

exhibit the comparison (overlap) of our PHGMM model against the different models (AdapNet++, OCNet, DUNet, HRNet, and HRNet+OCR). In the comparison image (e.g., the first column is PHGMM vs. AdapNet++ second one is PHGMM vs. OCNet++), blue represents regions that the PHGMM model erroneously segments, and red represents

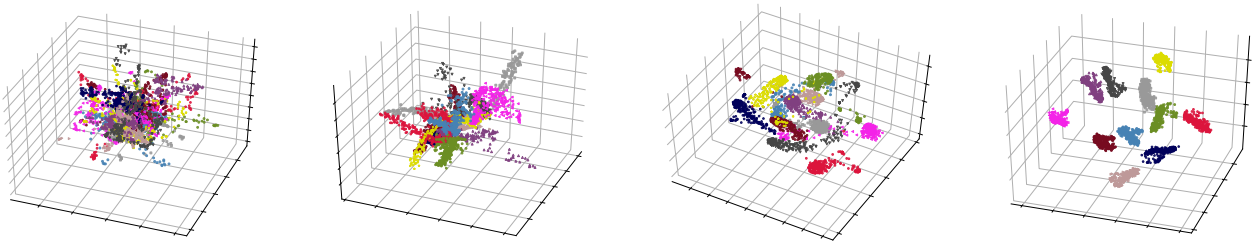


FIGURE 9. We show latent space clustering behavior for z, using the Cityscape validation dataset, using 11 classes and with crop size of 384×768. The classes are: Sky, Building, Road, Sidewalk, Fence, Vegetation, Pole, Car, Sign, Person, Cyclist. It through the different iterations measured by Silhouette Coefficient, SSI (higher is better), Calinski-Harabasz Index, CHI (higher is better), and Davies-Bouldin Index, DBI (lower is better).

those erroneously segmented by the several models (name of the columns). The regions correctly and incorrectly segmented by both predictions are green and purple, respectively. Finally, in the last comparison column, we present our model against itself, i.e., PHGMM vs. PHGMM. Here, we interpret the purple region as showing the incorrectly segmented regions of our PHGMM model. Note that our results decrease spatial precision loss at the boundaries compared to the other models (quantitative improvement in Fig. 7).

Remember, we conducted this research to address the loss of spatial precision in segmentation. We use Trimap [24], [70], which focuses on measuring the pixel-wise error on boundary regions of segmentation, with a distance to the boundaries given by trimap width (pixels). The plot in Fig. 7 (error curve comparison) demonstrates that our model has less loss of spatial precision compared to the literature (i.e., HRNet + OCR).

In addition, we present an ablation study (Fig. 6) on the latent space z (number of clusters K used in the GMM) and latent space g (including or excluding it from the architecture). While it may be interesting to also evaluate the global features alone, we did not perform this experiment due to the low performance that these features may produce given their location on the decoder (cf. Fig. 2). However, from these plots, we can observe that our best results are produced when using the learning representation z (local features) with the number of clusters k equal to the number of classes c in the dataset (i.e., $c = k = 12$). Thus, we intuit that, in an unsupervised way, each cluster of z (i.e., GMM) is modeled to obtain the best representation of local features focused on each class. Furthermore, our ablation study also shows that the latent space g is crucial to improving the final segmentation. The combination of features $(z + g)$ produces the best predictions.

Finally, we visualize the behavior of latent space z (Fig. 9). A Mixture of Gaussian fits z through the different iterations using the multidimensional projection method t-SNE [78]. Additionally, we show the influence of z in the prediction of the segmentation. The visualization, obtained with t-SNE [78], shows that the segmentation results are improved by providing clustering behavior to the latent space z.

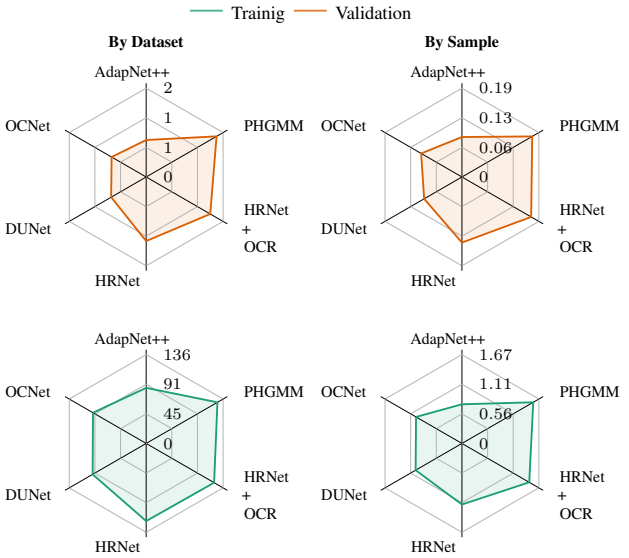


FIGURE 10. Comparison of the execution time of semantic segmentation models for the entire dataset (left) and per sample (right), both in minutes. We use the Cityscapes dataset with 17 500 and 500 samples for training and validation, respectively. Note that the PostNet network is only used in the training phase. Our PHGMM model in the testing phase presents an execution time comparable to the state-of-the-art.

Also, we present an efficiency comparison of the SS models (see Fig. 10) for the entire dataset and per sample. We used the Cityscapes dataset with 17 500 training samples and 500 validation samples (in our case, they are the validation samples) for these experiments. The results show the execution time of one epoch (a forward-pass over the entire dataset) in a single GPU. We executed the process five times and reported the averages of the entire training and validation dataset (left) and per sample (right), both in minutes. Our plots on training (Fig. 10) show an increase in the time required to train the PHGMM model. This increment is directly related to the use of the PostNet network and the optimization algorithm used to fit the weights in the training phase. Note that the PostNet proved robust enough to fit a Gaussian mixture model structure in the latent space. Remember, we do not use PostNet for the validation dataset, significantly reducing execution

time. Finally, the previous experiments have been conducted on four Nvidia GTX Titan X GPU 12 GB graphics cards, 1 TB of RAM, and 56 Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40 GHz (multi-GPU). Also, we use Python3, Numpy, Pillow, Tensorflow 1.8, Docker, and Ubuntu operating system.

Our previous experiments show that the SS task can be improved by endowing/providing a structure to the latent space z (i.e., learning representation) in the form of a Gaussian mixture. In order to take advantage of this rich feature extraction with structure and behavior clustering, we intelligently merge these features with encoder skip connections at the decoding stage. We are achieving in this way to recover the geometric information of the segmented objects (i.e., background and boundary of the objects). In addition, it is necessary to add complementary information (latent space t) focused on the contour of the objects, to improve the boundary of the segmentation (see Fig. 6). Finally, we address the loss of spatial precision due to the combination of three factors: the PHGMM architecture (merge features in the decoder stage), the extraction of specific features given by the latent space t, and the internal structure (Gaussian mixture model) imposed on the latent space z.

VI. FUTURE WORKS

Current models extract the context (generally global) of the objects in the images and delimit the object regions. However, small regions on the edges of these objects demonstrate spatial difficulties and are lost (see Fig. 1). This loss in the edges of objects can be caused by noise, blurring, smoothing, and image resolution low, among others. This work addresses the loss of spatial precision problem in semantic segmentation. As described in previous sections, the loss of information is commonly reflected in the edge of segmented objects.

Our PHGMM model works with local and global contexts presenting improved results at the edges of segmented objects. This improvement in the results has a direct correlation with the provided (pushed) clustering behavior in the latent space, see Figs. 7 and 9. In addition, this correlation is influenced by the number of clusters used in the clustering algorithm in the training stage, shown in Fig. 6. The following research step would be to increase the number of clusters through hierarchical clustering. In this way, we would group objects semantically at a higher level, and break them down into other sub-classes down in the hierarchy. Also, to accomplish this model idea, we would need a larger number of classes.

VII. CONCLUSION

In this work, we show that by endowing the latent spaces (z and g) with clustering behavior and providing them with a structural representation (i.e., GMM for z and Normal distribution for g), we improved the results of the SS task. This improvement is mainly due to the combination (produced in the decoding stage) of low-level features (from global-context information in g) and high-level ones (from local-context information in z). The combination of features is performed in the encoder stage with the latent spaces z, g, and the multi-scale features from the encoder in order to recover the geometric information. Furthermore, our comparative results show that more details can be extracted from PHGMM by setting the number of clusters equal to the number of classes. The improvements are produced at the segmented objects' boundaries, thus addressing the loss of spatial precision problem.

REFERENCES

[1] J. Tenenbaum, A. Witkin, and B. Wandell, "Vision: A computational investigation into the human representation and processing of visual information," *Psyccritiques*, vol. 28, no. 8, pp. 583–584, 1983.
[2] R. Gonzalez and R. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, 2006.
[3] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr, "Higher order conditional random fields in deep neural networks," in *European Conf. Comput. Vis. (ECCV)*. Springer, 2016, pp. 524–540.
[4] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3D object detection for autonomous driving," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 2147–2156.
[5] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot, "Automated evaluation of semantic segmentation robustness for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, 2019.
[6] J. Bullock, C. Cuesta-Lázaro, and A. Quera-Bofarull, "XNet: a convolutional neural network (CNN) implementation for medical x-ray image segmentation suitable for small datasets," in *Med. Imag. Biomed. Appl. Mol. Struc. Funct. Imag.*, vol. 10953. International Society for Optics and Photonics, 2019, p. 109531Z.
[7] C.-W. Wang, C.-T. Huang, J.-H. Lee, C.-H. Li, S.-W. Chang, M.-J. Siao, T.-M. Lai, B. Ibragimov, T. Vrtovec, O. Ronneberger *et al.*, "A benchmark for comparison of dental radiography analysis algorithms," *Med. Image Anal.*, vol. 31, pp. 63–76, 2016.
[8] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *Med. Image Anal.*, vol. 35, pp. 18–31, 2017.
[9] S. Pereira, A. Pinto, J. Amorim, A. Ribeiro, V. Alves, and C. A. Silva, "Adaptive feature recombination and recalibration for semantic segmentation with fully convolutional networks," *IEEE Trans. Med. Imag.*, 2019.
[10] J. Sherrah, "Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery," *arXiv*, vol. 0, no. arXiv:1606.02585v1, 2016.
[11] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 881–893, 2017.
[12] A. Bokhovkin and E. Burnaev, "Boundary loss for remote sensing imagery semantic segmentation," in *Inter. Symp. Neural Netw. (ISNN)*. Springer, 2019, pp. 388–401.
[13] M. Thoma, "A survey of semantic segmentation," *arXiv*, vol. 0, no. arXiv:1602.06541v2, 2016.
[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, pp. 1–1, 2016.
[15] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.
[16] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, "Exploring context with deep structured models for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1352–1366, 2017.
[17] D. Saire and A. Ramírez Rivera, "Empirical study of multi-task hourglass model for semantic segmentation task," *IEEE Access*, vol. 9, pp. 80 654–80 670, 2021.
[18] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Appl. Soft Comput.*, vol. 70, pp. 41–65, 2018.
[19] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, 2019.
[20] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, 2020.

[21] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021.

[22] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1529–1537.

[23] S. Jayasumana, K. Ranasinghe, M. Jayawardhana, S. Liyanaarachchi, and H. Ranasinghe, "Bipartite conditional random fields for panoptic segmentation," *British Mach. Vis. Conf. (BMVC)*, 2020.

[24] P. Krahenbuhl and V. Koltun, "Efficient inference in fully connected CRFs with gaussian edge potentials," in *Adv. Neural Inf. Process. Sys. (NeurIPS)*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 109–117.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2961–2969.

[26] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 8759–8768.

[27] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam, "MaskLab:instance segmentation by refining object detection with semantic and direction features," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 4013–4022.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *IEEE Inter. Conf. Med. Image Comput. Comput. Assist. Interv. (MICCAI)*. Springer, 2015, pp. 234–241.

[29] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1520–1528.

[30] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[31] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," *arXiv*, vol. 0, no. arXiv:1606.02147v1, 2016.

[32] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 1925–1934.

[33] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *European Conf. Comput. Vis. (ECCV)*, 2018, pp. 405–420.

[34] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," *arXiv*, vol. 0, no. arXiv:2005.10821, 2020.

[35] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv*, vol. 0, no. arXiv:1706.05587v1, 2017.

[36] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 2881–2890.

[37] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *European Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[38] X. Lian, Y. Pang, J. Han, and J. Pan, "Cascaded hierarchical atrous spatial pyramid pooling module for semantic segmentation," *Pattern Recogn.*, vol. 110, p. 107622, 2021.

[39] Y. Jin, D. Han, and H. Ko, "Trseg: transformer for semantic segmentation," *Pattern Recognition Letters*, 2021.

[40] D. A. Reynolds, "Gaussian mixture models," *Encyclopedia of biometrics*, vol. 741, 2009.

[41] C. M. Bishop, "Pattern recognition," *Machine learning*, vol. 128, no. 9, 2006.

[42] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *stat*, vol. 1050, p. 1, 2014.

[43] R. Vemulapalli, O. Tuzel, M.-Y. Liu, and R. Chellapa, "Gaussian conditional random field network for semantic segmentation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 3224–3233.

[44] Z. Huang, X. Wang, Y. Wei, L. Huang, H. Shi, W. Liu, and T. S. Huang, "CCNet: Criss-cross attention for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[45] Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, and R. Su, "DUNet: A deformable network for retinal vessel segmentation," *Knowledge-Based Systems*, vol. 178, pp. 149–162, 2019.

[46] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," *arXiv*, vol. 0, no. arXiv:1506.04579v2, 2015.

[47] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *IEEE Inter. Conf. Comput. Vis., Pattern Recog. Wksps. (CVPRW)*, 2017, pp. 11–19.

[48] H. Li, P. Xiong, H. Fan, and J. Sun, "DFANet: Deep feature aggregation for real-time semantic segmentation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 9522–9531.

[49] X. Li, X. Li, L. Zhang, G. Cheng, J. Shi, Z. Lin, S. Tan, and Y. Tong, "Improving semantic segmentation via decoupled body and edge supervision," in *European Conf. Comput. Vis. (ECCV)*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 435–452.

[50] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-SCNN: Gated shape CNNs for semantic segmentation," in *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2019, pp. 5229–5238.

[51] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *Inter. Conf. Learn. Represent. (ICLR)*, 2016.

[52] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *IEEE Wint. Conf. Appl. Comput. Vis. (WACV)*. IEEE, 2018, pp. 1451–1460.

[53] Z. Wang and S. Ji, "Smoothed dilated convolutions for improved dense prediction," *Data Min. Knowl. Discov.*, vol. 35, no. 4, pp. 1470–1496, 2021.

[54] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "ESPNetv2: A light-weight, power efficient, and general purpose convolutional neural network," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 9190–9200.

[55] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[56] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *European Conf. Comput. Vis. (ECCV)*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 173–190.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2017, pp. 5998–6008.

[58] X. Wang, R. Girshick, A. Gupta, and K. He, in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 7794–7803.

[59] J. Fu, J. Liu, Y. Wang, Y. Li, Y. Bao, J. Tang, and H. Lu, "Adaptive context network for scene parsing," in *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2019, pp. 6748–6757.

[60] X. Li, L. Zhang, A. You, M. Yang, K. Yang, and Y. Tong, "Global aggregation then local distribution in fully convolutional networks," in *British Mach. Vis. Conf. (BMVC)*, 2019.

[61] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 3146–3154.

[62] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "OCNet: Object context network for scene parsing," *arXiv*, vol. 1, no. arXiv:1809.00916, 2018.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 770–778.

[64] G. L. Oliveira, A. Valada, C. Bollen, W. Burgard, and T. Brox, "Deep learning for human part discovery in images," in *IEEE Inter. Conf. Robot. Autom. (ICRA)*. IEEE, 2016, pp. 1634–1641.

[65] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "BiSeNet: Bilateral segmentation network for real-time semantic segmentation," in *European Conf. Comput. Vis. (ECCV)*, 2018, pp. 325–341.

[66] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *Inter. J. Comput. Vis.*, pp. 1–47, 2019.

[67] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Semantic segmentation with context encoding and multi-path decoding," *IEEE Trans. Image Process.*, vol. 29, pp. 3520–3533, 2020.

[68] Y. Yuan, X. Chen, X. Chen, and J. Wang, "Segmentation transformer: Object-contextual representations for semantic segmentation," in *European Conf. Comput. Vis. (ECCV)*, vol. 1, 2021.

[69] S. Sherman, "Solomon kullback, information theory and statistics," *Bulletin of the American Mathematical Society*, vol. 66, no. 6, pp. 472–472, 1960.

[70] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan, "What is a good evaluation measure for semantic segmentation?." in *British Mach. Vis. Conf. (BMVC)*, vol. 27, 2013, p. 2013.

[71] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016.
[72] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 3234–3243.
[73] A. Valada, G. L. Oliveira, T. Brox, and W. Burgard, "Deep multispectral semantic scene understanding of forested environments using multimodal fusion," in *Inter. Symp. Exp. Robot. (ISER)*. Springer, 2016, pp. 465–477.
[74] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987.
[75] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat.-Theory Methods*, vol. 3, no. 1, pp. 1–27, 1974.
[76] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, 1979.
[77] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "AdapNet: Adaptive semantic segmentation in adverse environmental conditions," in *IEEE Inter. Conf. Robot. Autom. (ICRA)*. IEEE, 2017, pp. 4644–4651.
[78] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

DARWIN SAIRE received his B.Eng. degree in Computer Engineering from Universidad de San Agustin (UNSA), Arequipa in 2013. He completed his M.Sc. degree in Computer Engineering from University of Campinas, Brazil in 2017. He is currently a Ph.D. candidate at the Institute of Computing, University of Campinas, Brazil. His research interests are computer vision, pattern recognition, image processing, machine learning and deep learning.

ADÍN RAMÍREZ RIVERA (S'12, M'14, SM'21) received his B.Eng. degree in Computer Engineering from Universidad de San Carlos de Guatemala (USAC), Guatemala in 2009. He completed his M.Sc. and Ph.D. degrees in Computer Engineering from Kyung Hee University, South Korea in 2013. He is currently an Associate Professor at the Department of Informatics, University of Oslo, Norway. His research interests are video understanding (including video classification, semantic segmentation, spatiotemporal feature modeling, and generation), and understanding and creating complex feature spaces.