IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Pre-print to appear in IEEE Access

Digital Object Identifier 10.1109/ACCESS.2021.3085218

Empirical Study of Multi-Task Hourglass Model for Semantic Segmentation Task

DARWIN SAIRE,[1] AND ADÍN RAMÍREZ RIVERA,[2] (Member, IEEE)
[1]Institute of Computing, University of Campinas, Brazil (e-mail: darwin.pilco@ic.unicamp.br)
[2]Institute of Computing, University of Campinas, Brazil (e-mail: adin@ic.unicamp.br)

Corresponding author: Adín Ramírez Rivera (e-mail: adin@ic.unicamp.br).

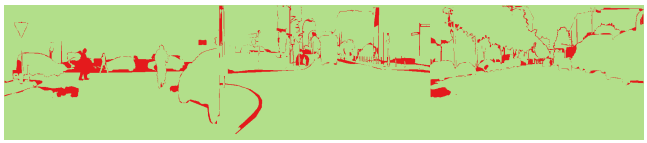
This work was financed in part by the São Paulo Research Foundation (FAPESP) under grants No. 2017/16597-7, 2019/07257-3, and 2019/18678-0, and in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under grant No. 307425/2017-7. The Laboratoire Lorrain de Recherche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

ABSTRACT The semantic segmentation (SS) task aims to create a dense classification by labeling at the pixel level each object present on images. Convolutional neural network (CNN) approaches have been widely used, and exhibited the best results in this task. However, the loss of spatial precision on the results is a main drawback that has not been solved. In this work, we propose to use a multi-task approach by complementing the semantic segmentation task with edge detection, semantic contour, and distance transform tasks. We propose that by sharing a common latent space, the complementary tasks can produce more robust representations that can enhance the semantic labels. We explore the influence of contour-based tasks on latent space, as well as their impact on the final results of SS. We demonstrate the effectiveness of learning in a multi-task setting for hourglass models in the Cityscapes, CamVid, and Freiburg Forest datasets by improving the state-of-the-art without any refinement post-processing.

INDEX TERMS Explainable Latent Spaces, Multi-Task Learning, Semantic Segmentation

I. INTRODUCTION

HUMANS possess a remarkable ability to parse images simply by looking at them. In a blink of an eye, a human can fully analyze an image and separate all its components. People can perform several tasks simultaneously by analyzing an image, e.g., object detection and contour detection. Furthermore, humans can easily generalize from observing a set of objects to recognizing objects that have never been seen before. Although humans enjoy an inherent capacity for generalization, they lack the processing power given by computers. That is, to process a large amount of information (e.g., images) in a reduced interval of time. The separation of an image into its components (i.e., join pixels into regions) according to some features is called image segmentation [1]. Reproducing this process at or above the human level on a computer is not an easy task, and several approaches have been proposed to address it [2]. Nevertheless, the segmentation task continues to be challenging mainly due to variability, i.e., when the visual tasks are performed on a computer there is a considerable variation in pose, appearance, viewpoint, illumination, and occlusion throughout different instances of the same image. Thus, a type of segmentation commonly used is semantic segmentation (SS). SS is an essential part of the pipeline in computer vision projects. It extracts and analyzes useful and meaningful information, in addition to classifying the regions obtained within an image. In other words, by improving the segmentation stage, the computer vision's final output is also enhanced.



(a) SegNet (b) AdapNet++ (c) FastNet

FIGURE 1. The main problem of current segmentation methods lie in the loss of spatial precision at the boundaries or small objects. Green and red regions denote correctly and incorrectly segmented regions, respectively.

In recent years, Convolutional Neural Networks (CNNs) have led to several improvements in computer vision. Fully convolutional networks (FCN) [3] achieved a significant improvement in the SS task in contrast to the traditional SS

techniques [4]. However, (i) the low-resolution at the CNNs output and (ii) the loss of spatial precision of objects within the image are still the main problems that affect the segmentation results [5], [6], see Fig. 1. We believe that these problems are not caused by a specific operation (e.g., down-sampling) but by a set of factors. For instance, the absence of reconstruction and refinement methods, the excessive down-sampling, the gradient vanishing problem, or lack of a better extractor of feature maps.

Nowadays, different models [7], [8] have tackled these problems and advancing solutions to the problem of the low resolution on the output maps. For better refinement, previous models [9] post-process the results to enhance them, e.g., Conditional Random Fields (CRF) [10]. For global feature extraction with more information, new architectures [11]–[13] were created as well as sparse convolution operations [5], [14]. Commonly, methods in SS use hourglass models [15]–[17] that comprise a coding and decoding stages to recover the pixel-wise position of the segmented objects.

Other models use Multi-Task Learning (MTL) based on the idea that simultaneously learning related tasks can improve the performance on all of them [18]–[20]. Hence, related tasks facilitate the transfer of shared knowledge among them. E.g., edge detection improves segmentation by adjusting the edge at each level of a CNN [21]–[23] or by learning edge-aware features [24]. Although several MTL approaches [13], [25]–[27] were applied to the SS task, it is still difficult to say which auxiliary tasks most beneficial for the final SS results. Even more, which of these auxiliary tasks provide the necessary complementary information so that the models can address the loss of spatial precision. We deduce that by reinforcing the object contours' information (through auxiliary tasks), we will be able to force greater attention (in the training phase) on the segmented objects' contours through the MTL approach.

In this work, we use a multi-task approach with complementary contour-based tasks for rich and robust feature extraction and address the problem of spatial precision loss. We work specifically with hourglass (encoder-decoder) models because we (empirically) discovered that a multi-task setup helps to adjust the latent space in these models (i.e., it exhibits a clustering behavior). We also show that the improved results of different hourglass (encoder-decoder) architectures are directly related to this clustering behavior. We present four different studies on the latent space for SS: (i) visualization of the latent space behavior, (ii) activation maps used by the models to predict the segmentation, (iii) reduction of over-fitting of the models, and (iv) the ablation studies on loss functions and complementary tasks. The latent space's visualizations show different induced clusters when influenced by adding or removing the different complementary contour-based tasks. Fig. 2 depicts the contour-based tasks used in our study. Namely, they are edge detection [1], semantic contours [28], semantic segmentation [29], and distance transform [30].

In summary, our main contributions are:

- Address the problem of spatial precision loss by obtaining hourglass models with better generalization by focusing on segmented objects' contours.
- Improve the SS results in hourglass models by using complementary information from contour-based tasks, and thus induce a clustering behavior in the latent space.
- Extensively evaluate common hourglass models, namely SegNet [17] and UNet [15], on Cityscapes [31], and CamVid [32] datasets. Moreover, we show that the use of complementary tasks improves the state-of-the-art in hourglass (encoder-decoder) models.

II. RELATED WORK

Over the past years, the SS task has been done with deep learning as the preferred option due to deep neural networks' (DNNs) extraordinary ability for feature extraction. The initial layers learn the low-level features (e.g., edges and texture), while the last layers learn the higher-level ones (e.g., identify different objects). The feature extraction of DNNs can be improved by adding complementary information (though a multi-task approach) in the training phase. This section presents a review of relevant literature approaches that focus on SS tasks, using several approaches, such as deep learning and MTL.

A. SEMANTIC SEGMENTATION

Semantic segmentation refers to the process of linking each pixel in an image to a class label. In the deep models, FCN showed to be useful in this task. However, the first SS models produced low-resolution maps with a loss in spatial precision. Here we discuss different models created to deal with these problems.

Some researchers [9] used FCN with CRFs as a post-processing step, but it is computationally expensive. Consequently, embedding the post-processing steps within a network [5], [33] was a viable solution. In contrast, we improve the hourglass (encoder-decoder) models without the need to use post-processing steps by introducing additional tasks that refine the latent space.

Other models [34]–[36] adjusted the bounding boxes. The intuition was to do object detection first and then refine the instances' contours. Mask R-CNN [34] used a feature pyramid network [37] to extract a feature hierarchy in-network and an FCN to get a segmentation mask in each region of interest. This region-based approach had proper segmentation but depended on the accurate detection of objects (bounding box). Hourglass models, on the other hand, do not have this constraint. Other models [38] require more delimited boundaries for the segmentation as masks instead of just a box. They also use sliding operations to obtain better adjustment [39] to the final targets.

Instead of an abrupt prediction of the last layer, the hourglass approach [11], [12], [40], [41] (i.e., models with encoder-decoder stages, such as U-Net [15], DeconvNet [16], SegNet [17]) created a decoder stage to gradually recover the spatial information by combining multi-level feature

maps from the encoder. Thus, the flow of information from a lower scale to a higher one is done by an upsampling operation, i.e., bilinear interpolation [1], unpooling [16], or DUpsampling [42]. We consider hourglass models to have a robust decoding stage for the reconstruction of the pixel-wise predicted image. Although hourglass models proved efficient in SS, they still need a more significant transfer of information between its stages, e.g., FC-DenseNet [43] or UPSNet [44]. For this reason, we add complementary information to the models by adding the MTL approach (i.e., auxiliary task).

Though the previous models improved the objects' boundary, we need models that observe larger regions. Thus, multi-scales models emerged [33], [45], [46]. They obtain a full semantic map in low-resolution (coarse prediction map), then refine it with different fusion operations, e.g., fusion cascade [47], attention blocks [48], [49], layer aggregation [50], residual units [51], and gated fusion [52]. These models are unnecessarily complex to extract robust features. Instead, we use auxiliary tasks to reinforce the gradient and achieve better information extraction.

Current models (e.g., HRNetv2 [53], HRNet+OCR [54]) perform multi-scale feature extraction by sharing feature maps across their different levels (scales), i.e., broadcasting context information at various resolutions. Contrary to multi-scale models and to capture high-resolution feature maps, PSP-Net [55] performs pooling operations at multiple grid scales. Simultaneously, DeepLabv3+ [56] and CasiNet [57] use Atrous Spatial Pyramid Pooling (ASPP) [5], [58] (i.e., several sparse filters) to modify the filters' size instead of the images' size [59]. Later experiments showed that there are still limitations to get global features [60]. Moreover, the introduced dilated convolutions bring heavy computation complexity and a memory footprint, thus limiting many applications' usage.

The first attempt to address the high resource consumption of ASPP was FastFCN [61], which performs a new method of ascending pyramidal sampling. Besides, AdapNet++ [12], [62] proposed cascaded and parallel Atrous convolutions to capture long-range context using fewer parameters. However, the problem of spatial precision loss persisted. These results lead us to believe that we need models that make use of inductive biases, i.e., more specific features from prior information. We address this problem by using well-behaved hourglass models paired with multi-task learning to improve the learned features.

B. MULTI-TASK LEARNING

In machine learning, we generally train a single model to perform a specific task. By focusing on a single task, we risk ignoring additional information that could help us learn a better representation of the desired task. Instead, MTL [63] aims to solve multiple related tasks simultaneously. Thus, it facilitates the transfer of shared knowledge across relevant tasks [20], [64].

In this literature review, we focus on supervised learning tasks due to similarity with our work. Currently, machine learning models share knowledge in two ways [20], [65]: (i) feature-based MTL that distributes knowledge across training representative features, and (ii) parameter-based MTL that uses the model parameters trained in a specific task to fit the related tasks. In this work, we are interested in studying the latent space shared among all tasks while focusing on SS as the main task restricted to hourglass models.

The previous models [66]–[68] used handcrafted features and assume that the data-to-target has a direct relationship. Many times the data exhibit a complex data-to-target relationship [20]. This assumption can restrict the models' performance. For this reason, deep learning with MTL is used due to its capacity to learn nonlinear complex latent representations. Deep MTL is grouped into two types [20]: hard (i.e., sharing parameters between all tasks) and soft (i.e., each task has its model, hidden layers, and parameters).

Previous models [69], [70] used two separate architectures with soft parameter sharing. They used cross-stitch units or task transfer connections to leverage the knowledge of the task-specific networks. In contrast, MRN [64] learned a Bayesian transfer relationship (both the last layers). The previous models, having their own parameters for each task, made it easy to increase the number of required resources.

Unlike soft models, the hard ones do not need any assumption for the tasks' relation; they do this internally. Thus, some MTL models [25], [38] use a cascade-based approach to learn a task from the previous one. However, this approach restricts the feature space. Accordingly, models [71], [72] (with independent tasks) focus on merging multiple loss functions (depending on the task) to ensure the convergence of the models and robustness to noise [73]. Some models [74]–[76] combined semantic segmentation with geometric information and others [26], [77] with depth. Other models [26], [78], [79] measured and adjusted the degree of uncertainty of the samples along with the segmentation. We noted that the uncertainty (i.e., either due to noise at the capture or due to the prediction's degree of confidence) is related to the segmented objects' edges. Similarly, previous works [80], [81] also used this line of reasoning by merging semantic contours with edge detection using the multi-scale feature in Acuna et al.'s [82] work or adjusting the contours across the entire network as done by Cheng et al. [80] (i.e., multiple losses for detection). Finally, the combination of SS with edge detection (in models [24]) showed significant improvements for the segmentation task. These models (CCL [22], CGB-Net [23]) generally present an hourglass architecture with skip connections for the edge detection map.

Despite the useful latent space for related tasks obtained by the deep MTL approaches [23], it is not yet explored or understood how this latent space behaves to improve the target task. In other words, understanding what parts of the latent space improve SS is an open problem. Moreover, information is even scarcer when the target task is SS on images (i.e., multi-label pixel-wise classification).

III. OVERVIEW

The idea of using CNNs as feature extractors and generators is not new. It has been widely used and achieved better results against traditional methods [83]. Previous works (see Section II-A) use a CNN for SS tasks and bring up challenging problems like the loss of spatial precision as the main problem. Besides, we discussed in Section II-B that deep MTL models obtain additional information from related tasks and learn at some level a new feature space shared across all tasks, specifically in hourglass models. However, there is still no analysis of deep MTL models specifically for the SS task. In particular, there is no indication of what happens with the shared features, how they behave, nor what are the most relevant related tasks for SS to enhance them. In this work, we are interested in a particular type of behavior in the latent space, i.e., clustering, which improves the SS results in hourglass (encoder-decoder) architectures. We empirically analyze how clustering in the latent space is influenced by different contour-based auxiliary tasks. We highlight that the clustering behavior is only observed in hourglass architectures. These models (e.g., UNet, SegNet, ParseNet) depend largely on the latent space to perform the reconstruction (a stream or up-down-up route). In contrast, other models (e.g., HRNet) perform feature extraction across multi-resolution (i.e., multi-stream). They distribute more contextual information but deprive the intermediate representation spaces of internal interpretability (i.e., decreasing the interpretability in latent representation). In this section, we introduce our learning framework. Also, we introduce the datasets and define the metrics we used in this work. Recall that our study is entirely empirical. Our objectives are (i) to propose and evaluate the use of contour-based auxiliary tasks to address the problem of loss of spatial precision, and (ii) to show how the addition or deletion of these contour-based auxiliary tasks helps improve the SS results for hourglass models.

A. LEARNING A MULTI-TASK APPROACH

Deep MTL approaches learn features that might not be easy or possible to learn within original task. We want to know if we can leverage the information in the training signals of other related SS tasks during the learning phase. An effective way to achieve this is by giving cues to the model from other related tasks, i.e., predicting the features with an auxiliary task.

The goal of an auxiliary task in MTL is to learn useful shared representations for the main task (i.e., add a regularizing factor [84]). They are closely related to the main task, so adding them allows the model to learn beneficial representations. However, finding an auxiliary task that helps improve the SS task is not trivial [85]. At first glance, tasks that seem different can use similar representations, and tasks that seem related can adjust different internal functions [63]. We still do not know which auxiliary tasks will help in practice for SS. Finding an auxiliary task is largely based on the assumption that they should be related to the main task in some way. We perceive, from Fig. 1, that spatial precision loss is generally produced on the edge of segmented objects. So

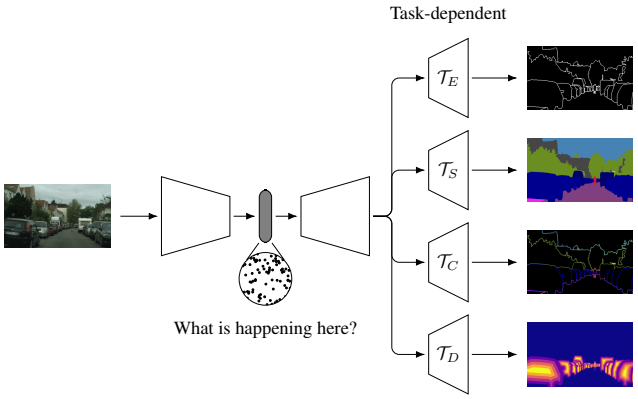


FIGURE 2. Illustration of a multi-task hourglass model, for tasks of edge detection (E), semantic segmentation (S), semantic contours (C), and distance transform (D), from top to bottom. Note that the model share weights in the first layers (encoder and decoder), and the specific features for each task are obtained in the last layers (specific task decoders $\mathcal{T}_\cdot$).

we use tasks related to the gradient or edge regions. That is, we give more attention to the contours of the objects. With this in mind, we propose to employ three types of contour-based auxiliary tasks to improve the boundary of segmented object and, therefore, the SS task. We choose auxiliary tasks to reinforce and complement the information obtained from the edges of objects. Thereby, we address the problem of spatial precision loss, generally reflected in the segmented objects' contours, cf. Fig. 1.

We propose to use the additional tasks of edge detection (E), semantic segmentation (S), semantic contours (C), and distance transform (D), cf. Fig. 2. Edge detection [1] aims to extract object boundaries. Distance transform [30], in our case, is a distance function to the objects' edges. Semantic contour [28] produces a pixel-wise level dense classification on objects' contour. Initially, we tried to use a continuous distance transform (i.e., without quantification). However, we discard it because of the longer training time to convergence, and the results were comparable with the quantized distance transform. We intuit that this behavior is due to the higher degrees of freedom when fitting a regressor. In Appendix A, we detail how to quantize the distance transform.

Although these auxiliary tasks were previously used in MTL models [48], [76], [80], [81], they were not used together in the same model. Besides, the impact produced by adding each of the auxiliary tasks has not yet been studied. Consequently, we show how each of them improves the latent space separation (Section IV-B). We evaluated each of these auxiliary tasks' contribution (quantitative results) to the SS task (see our ablation study in Section IV-A).

Unlike the previous hourglass models (encoder-decoder), the hourglass with MTL adds specific heads for each task (i.e., deconvolution layers in the last decoder stage). Our hourglass model with MTL (Fig. 2) has two types of hidden layers, the shared layers, and task-specific layers. The shared layers learn a low-level representation of the data, influenced by all tasks, while the task-specific layer learns the parameters for

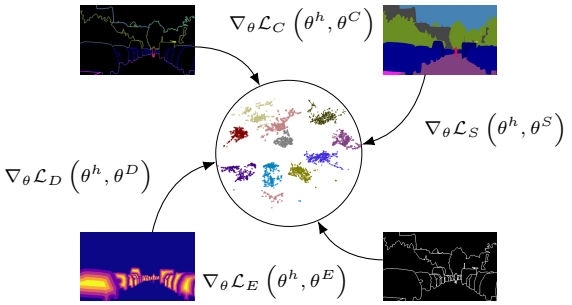


FIGURE 3. The contour-based auxiliary tasks influence the latent space through their gradients (by backpropagation). Due to the contour origin of the task their corresponding losses penalize the contours of the objects to improve.

the pixel-wise classification network. These specific layers map the learned latent representations from the previously shared layers to the task-specific output layers (i.e., target for each task).

Consider that the hourglass models with MTL work over a set of images X, and has a corresponding ground truth per task $\{Y^t\}_{t\in T}$, i.e., set of pixel-wise labeled images by task. Each ith sample has a corresponding ground truth image y_i^t for the corresponding task t. Thus, the hourglass model is represented by $f(x;\theta^h,\theta^t)=y^t$ such that some parameters, θ^h, are shared between the contour-based tasks, and some, $\theta^t \in \{\theta^E,\theta^S,\theta^C,\theta^D\}$, are particular to each specific task.

The hourglass parameters are learned by solving an optimization problem that minimizes a weighted sum of the losses for each task. It is defined by

$$\mathcal{L}_{final} = \min_{\theta^h,\theta^E,\theta^S,\theta^C,\theta^D} \frac{1}{|T|N} \sum_{t\in T}\sum_{i=1}^{N} \lambda_t \mathcal{L}_t\left(\theta^h,\theta^t\right), \quad (1)$$

where we used four tasks $T=\{E,S,C,D\}$, N is the number of samples, and the loss of the auxiliary task t is defined as

$$\mathcal{L}_t(\theta^h,\theta^t) = \mathcal{L}_t(f^t(x_i;\theta^h,\theta^t), y_i^t), \quad (2)$$

and where $\mathcal{L}_t = \{\mathcal{L}_E,\mathcal{L}_S,\mathcal{L}_C,\mathcal{L}_D\}$ represents the loss functions of each task. To moderate each task's importance on the model loss (1), we use a scalar λ_i to weigh each task loss (2). Each loss function helps to adjust the latent space into a useful representation for each task. In this work, we use for each task's loss the cross-entropy and soft IoU loss functions (see details in Appendix C).

In Fig. 2, the $\mathcal{T}_E$, $\mathcal{T}_S$, $\mathcal{T}_C$, and $\mathcal{T}_D$ blocks represent the layers that extract specific information to discriminate each task. In other words, each distinct decoding stage for each task have independent parameters. Since we work with the latent space, we analyze how the auxiliary tasks influence the latent space (feature representation). We do not need to use large networks for the specific tasks. Thus, each specific task-block contains two layers of convolutions, ensuring that the enhancement is performed on the shared parameters (i.e., a more robust feature extraction). The gradients of the separate tasks carry out the influence of the additional tasks on latent space (see Fig. 3). Due to the chosen tasks, gradients are prone to bring more significant changes to the edges of objects. That is, to provide further attention to the edges of the objects when performing the segmentation.

We are aware of the existence of non-deep-learning-based methods to do edge detection (e.g., Canny [86], Sobel [1], hierarchical method [87]) or distance transform (e.g., mathematical morphology [1]). We plan to use these auxiliary tasks only in the training phase, and not as final tasks that would replace the existing methods. The multi-task setup is to provide complementary information to the latent space and adjust the SS task. Additionally, we evaluate the impact of each auxiliary task.

B. EXPERIMENTS DESCRIPTION

We describe a set of empirical studies in order to show how the addition or removal of contour-based auxiliary tasks helps improve the semantic segmentation task. We also demonstrate that the use of auxiliary tasks diminishes the loss of spatial precision in the segmented objects.

Ablative Studies (Section IV-A): We performed two ablation studies. The first study is on the loss functions. We want to know which of the loss functions (cross-entropy and soft IoU) trains a better model for the SS task. We determined the best results according to both loss functions and a data augmentation technique explained in Section III-C. We did a second study to know which contour-based task helps improve the prediction. In other words, we evaluated quantitatively how the addition or removal of related tasks impacts the final segmentation result. We obtained the best results by training the models using all tasks together.

Visualization of Latent Space Behavior (Section IV-B): Here, we show the latent space behavior in the well-known SegNet model using complementary information from contour-based auxiliary tasks. To plot the samples in this study, we used the multidimensional projection method t-SNE [88]. Experiments show that the latent space exhibits clustering behavior, improving dissimilarity and segmentation results by adding auxiliary tasks.

Showing Activation Maps (Section IV-C): With the previous experiments, we obtained the best results using all the auxiliary tasks. In this study, we plot the activation maps used by the hourglass models to predict the segmentation. In this work, the activation maps are the regions that the network use for dense classification at the pixel-wise level. Therefore, we observe that by training hourglass models with contour-based auxiliary tasks, the model employs activation maps not previously used to improve the contours of segmentation.

Reducing the Over-Fitting (Section IV-D): In this study, we investigated whether there is a segmentation improvement on the segmented object's edge. To do so, we evaluated the classification errors of the segmented object's edge. We performed these experiments for various hourglass models for binary and multi-label segmentation. The empirical results show that there is an improvement at the segmented object's edges. This improvement appears due to the having a more robust latent space that better defines the objects. By using

complementary information, we learn models that generalize better than the traditional ones. Thus, we address the problem of spatial precision loss.

Comparing Results (Section IV-E): Previously, we carried out extensive studies on CamVid dataset due to the shorter required training time. We present final results on a set of hourglass models with and without MTL for the SS task and comparison tables for the CamVid, Cityscape, and Freiburg Forest datasets. We improve the final segmentation results when using MTL with contour-based tasks. The improvement may seem modest. However, the amount of pixels at the objects' edges is small in comparison to the image total amount of pixels.

C. DATASETS

We evaluated our proposed methodology on Cityscapes [31], CamVid [32], and Freiburg Forest [89] datasets. They contain several types of urban/forest scenarios.

Cityscapes: The dataset has 5000 samples with 2048×1024 size images and pixel-level labels of 19 semantic classes. There are 2979, 500, and 1525 images in the training, validation, and test set, respectively. We do not use coarse data in our experiments. For this work, we required a wide variety of samples; for this reason, we employ data augmentation. We applied a random crop of 300×500 and some random transformations of contrast, brightness, and horizontal flip; thus, we generated 17 500 training samples. We use the original validation set to compare the MTL models with a resolution of 768×384 pixels (resize). For this, we employ bilinear interpolation (for RGB images) and the nearest-neighbor interpolation (for the labels). To facilitate comparison with previous approaches, we report results on the reduced 11 class label set consisting of: *sky, building, road, sidewalk, fence, vegetation, pole, car/truck/bus, traffic sign, person, rider/bicycle/motorbike, and background*.

CamVid: It is road scene understanding dataset for SS with 11 classes: *building, tree, sky, car, sign, road, pedestrian, fence, pole, sidewalk, and cyclist*. The dataset has 367, 101, and 233 samples for training, validation, and test set, respectively, with images size of 360×480. We apply the same transformations of Cityscapes for the data augmentation, with a random crop size of 260×346, generating 5616 samples for the training set. We report results on the original test set.

Freiburg Forest: It is a dataset on forests with six classes: *sky, trail, grass, vegetation, obstacle, and void*. Note, forested environments are unstructured (e.g., trails), unlike urban scenes that are highly structured (rigid and geometric objects, e.g., buildings). We do data augmentation using transformations of Cityscape, generating 1840 samples for the training set. We conserve the original testing set (136 images). Note, all the images are resized at 768×384 pixels.

D. EVALUATION METRICS

The success achieved by the SS methods must be measured by the achievements of the final applications. They are generally too difficult to evaluate because graphical applications often require an expert user. For this reason, it is necessary to use application-independent measures of accuracy. Thus, to evaluate our results on segmentation, we chose accuracy, intersection-over-union, precision, and recall metrics as validation measures (from Csurka et al. [90]). The intersection-over-union (IoU) is defined by

$$\text{IoU} = \sum_{i}^{N} \frac{P_i \cap Y_i}{P_i \cup Y_i} = \sum_{i}^{N} \frac{TP_i}{TP_i + FP_i + FN_i}, \tag{3}$$

the accuracy (Acc), i.e., pixel-wise accuracy is

$$\text{Acc} = \sum_{i}^{N} \frac{P_i \cap Y_i}{Y_i} = \sum_{i}^{N} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}, \tag{4}$$

the precision (Prec) is

$$\text{Prec} = \sum_{i}^{N} \frac{TP_i}{TP_i + FP_i}, \tag{5}$$

and the recall (Rec) is

$$\text{Rec} = \sum_{i}^{N} \frac{TP_i}{TP_i + FN_i}. \tag{6}$$

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.

Furthermore, to measure the behavior of the latent space, we use metrics for clustering. Thus, we utilize the Silhouette Coefficient (SSI) [91] defined by

$$\text{SSI} = \sum_{i}^{N} \frac{b_i - a_i}{\max\{a_i, b_i\}}, \tag{7}$$

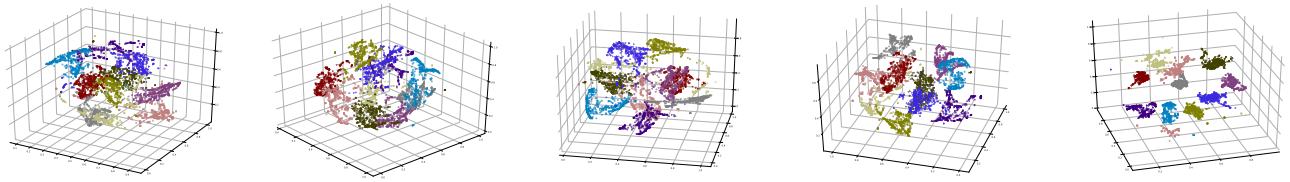
where a_i is the mean intra-cluster distance from sample i, and b_i is the mean nearest-cluster distance from i to each sample. Note that a higher value is related to better-defined clusters. The Calinski-Harabasz Index (CHI) [92] is given by

$$\text{CHI} = \frac{\text{SS}_M}{\text{SS}_W} \frac{N-k}{k-1}, \tag{8}$$

where k is the number of clusters, and N is the total number of observations (i.e., data points), SS_W is the overall within-cluster variance and, SS_M is the overall between-cluster variance. Note that a higher value is associated with dense and well-distributed clusters. Finally, we employ the Davies-Bouldin Index (DBI) [93] denoted by

$$\text{DBI} = \frac{1}{k} \sum_{i}^{k} \max_{j \neq i} \left(\frac{s_i + s_j}{d_{ij}} \right), \tag{9}$$

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 that a lower value is related to better separation between the clusters.



(a) S SSI = 0.384, DBI = 1.360 (b) S+E SSI = 0.391, DBI = 1.141 (c) S+D SSI = 0.394, DBI = 1.275 (d) S+E+C SSI = 0.437, DBI = 1.150 (e) S+E+C+D SSI = 0.636, DBI = 0.774

FIGURE 4. We show the shared latent space on the Camvid testing dataset. We combine the different tasks of edge detection (E), semantic segmentation (S), semantic contour (C), and distance transform (D). When adding tasks related to semantic segmentation, i.e., by providing complementary information, maps of similar features (within a unimodal multi-task hourglass model) are clustered together in a similar latent space, and they are not arbitrarily placed. We confirm this behavior by using a set of metrics for clustering shown in Table 2.

TABLE 1. Ablative study on loss functions of the SegNet [17] model with a multi-task approach on the CamVid test set.

W/O			Metrics Segmentation			
Cross	IoU	Aug	Acc↑	IoU↑	Prec↑	Rec↑
✓	–	–	0.6241	0.5206	0.6794	0.6241
–	✓	–	0.6475	0.5402	0.7049	0.6475
✓	✓	–	0.6582	0.5491	0.7165	0.6582
✓	–	✓	0.6665	0.5569	0.7256	0.6665
–	✓	✓	0.6922	0.5774	0.7535	0.6922
✓	✓	✓	**0.7067**	**0.5895**	**0.7693**	**0.7067**

TABLE 2. Ablative study on tasks of edge detection (E), semantic segmentation (S), semantic contours (C), and distance transform (D) of the SegNet [17] on the CamVid test set.

Task				Metrics Segmentation				Metrics Clustering		
S	E	C	D	Acc↑	IoU↑	Prec↑	Rec↑	SSI↑	CHI↑	DBI↓
✓	–	–	–	0.7067	0.5895	0.7693	0.7067	0.3843	1847.40	1.3602
✓	✓	–	–	0.7087	0.5912	0.7715	0.7087	0.3906	2262.23	1.1411
✓	–	–	✓	0.7117	0.5938	0.7748	0.7117	0.3940	2009.04	1.2752
✓	✓	✓	–	0.7329	0.6114	0.7979	0.7329	0.4369	2507.33	1.1495
✓	✓	–	✓	0.7306	0.6095	0.7954	0.7306	0.4437	3023.76	0.9200
✓	✓	✓	✓	**0.7503**	**0.6259**	**0.8168**	**0.7503**	**0.6360**	**4060.19**	**0.7743**

IV. EXPERIMENTS

This section presents a set of empirical studies of the latent space on hourglass models based on the MTL approach. Then we perform a series of ablation experiments on the CamVid dataset using the well-known SegNet model (see Fig. 2). Also, we present comparisons between the different hourglass models with and without multi-task (contour-based tasks) for the Camvid, Cityscape, and Freiburg Forest datasets.

A. ABLATIVE STUDIES

We present two types of ablative analysis on the CamVid dataset, using the well-known hourglass model, SegNet, for the semantic segmentation task. The first study, presented in Table 1, shows the distinct behavior of the SegNet model by using the different loss functions (cross-entropy and loss-IoU) and data augmentation. The reported results focus on a single semantic segmentation task on the CamVid test set. The best performance of the model presented in Table 1 was achieved using both loss functions and data augmentation. So we opted for this configuration for the following experiments.

The second study, presented in Table 2, focuses on the internal behavior of the SegNet model's latent space when adding contour-based auxiliary tasks. These tasks are edge detection (E) [1], semantic contours (C) [28], quantized distance transform (D) [74], and semantic segmentation (S) [29] as the main task. Here, we use the segmentation metrics to evaluate the predicted regions. To evaluate the behavior (distribution), we use several clustering metrics. We notice a direct correlation between clustering behavior and segmentation results. These quantitative results on CamVid are complementary results to those presented in Section IV-B. In Table 2, our best results are produced by using both loss (cross-entropy and loss-IoU), data augmentation, and all tasks. We replicate this setting in Cityscapes and Freiburg Forest datasets.

B. VISUALIZATION OF LATENT SPACE BEHAVIOR

One way to understand the latent space in the hourglass model is to look at it, how it behaves, and its influence on the segmentation predictions. Thus, we plot the latent space in which all the tasks are involved. We illustrate this space for the segmentation task using t-SNE in Fig 4. Note that by using more related tasks, the space is better delimited.

We see that the SS task by itself presents a poorly distributed latent space; see Fig. 4(a). By adding the edge detection task, the latent space improves its clusters per class, although it could be improved, see Fig. D.1(b). A particular task that supports segmentation is the distance transform quantified by adding geometric information to the feature maps, see Fig. D.1(c). On the other hand, with the semantic contours task, the semantic information is reinforced, achieving a better distribution in the latent space, see Fig. 4(d). Thus, by adding geometric information and a higher quality of semantic information, the latent space presents a better delimitation and, therefore, better quantitative results, see Fig. 4(e).

We deduced that by adding complementary information (i.e., auxiliary tasks) in the MTL stage, the features that stimulate the activation of the same neurons on the network are reinforced across tasks. Moreover, these features are clustered together. Based on the set of clustering metrics (SSI, CHI, and DBI), shown in Table 2, we can say that maps of similar features within an MTL hourglass model are correctly grouped, and they are not placed arbitrarily.

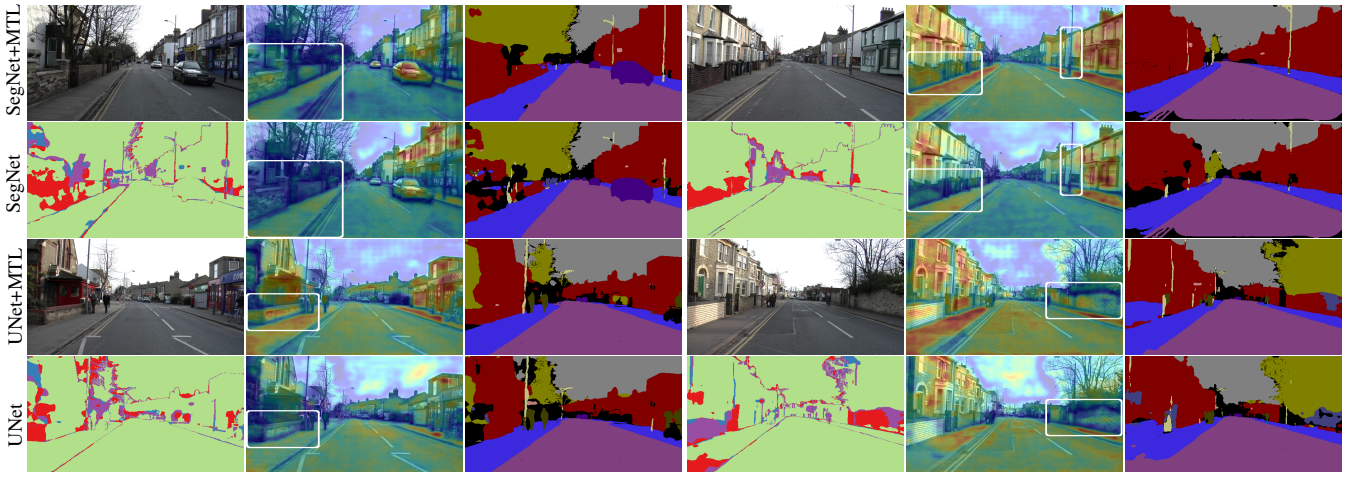


FIGURE 5. A comparison of activation maps (i.e., regions that are responsible for CNN's prediction) produced by the SegNet [17] and UNet [15] models with and without a multi-task approach. By using related tasks (i.e., by adding complementary information), the activation maps are better delimited (white bounding box). In the comparison images, correctly segmented regions are green while incorrectly segmented ones by the original models (SegNet and UNet) are red, by the multi-task models (SegNet+MTL and UNet+MTL) are blue; and by both are purple.

C. VISUALIZATION OF ACTIVATION MAPS

Another way to understand CNNs is to look at the important image regions that influence their SS predictions. In this study, we analyze the regions used by the hourglass models to make the best prediction of the segmentation (activation map) when applying the latent space adjusted by the multi-tasks. The proposed visualization of activation maps is typically performed during inference (testing) to provide visual explanations for the network's prediction.

We present, in Fig. 5, a comparison between the image regions that were responsible for CNNs prediction (i.e., activation maps) of SegNet [17] and UNet [15] models with and without MTL. We notice that the activation maps are better adjusted to the objects' contours (white bounding box). This behavior happens due to the better distributed latent space (i.e., decoder stage) on the models trained with MTL, see Fig. 4(e). The latent space's clustering behavior provides the networks' ability to use regions they did not use before to make the segmentation prediction.

The prediction columns in Fig. 5 show the predictions made by both models; in addition, the rows show the models with and without MTL. In the comparison columns, we show the correct and incorrect segmented regions color coded. The green regions are correctly segmented. Red and blue color regions indicate the regions incorrectly segmented for models without and with MTL, respectively.. Lastly, the purple regions are the incorrectly segmented ones produced by both models.

We show different activation for SegNet and UNet with and without MTL in Fig. 5. First, we reaffirm the spatial precision loss as the main problem of SS due to the incorrect segmentation in the objects' boundary. Second, adding tasks focused on object contours helps hourglass models highlight regions that were not adequately delimited (white bounding box). In conclusion, we can say that the improvement of SegNet+MTL and UNet+MTL (quantitative results in Section IV-A) happens mainly due to a higher flow of information provided by the contour-based auxiliary tasks. It better delimits the activation maps used for dense pixel prediction.

D. REDUCING THE OVER-FITTING

In this study, analyzed the contours of the segmented objects. We evaluate whether there is improvement in the objects' edges, and if the proposal addresses the problem of spatial precision loss.

Our study (in Fig. 6) shows that there is improvement in the segmented objects' boundary. We compare the popular SegNet [17], DeconvNet [16], and UNet [15] hourglass models for SS on the CamVid dataset. Note, these models share similar operations to avoid using models with additional operations (e.g., atrous convolutions) and ensure that the improvement is not due to the use of these operations.

For comparison, we report experiments employing Trimap [10], [90], which focuses on boundary regions of segmentation; see evaluation region in Fig 7(c). The Trimap is a rough image segmentation in the foreground, background, and unknown regions, shown in Fig. 7(b) with white, black, and gray regions, respectively. The idea is to define a narrow band (gray region defined using a width pixel) around each contour and compute pixel-wise accuracy in the given band. The error curve comparison (plots in Fig. 6) shows that learning the contour-based auxiliary tasks did not allow the network to overfit, enabling the networks to generalize better.

From the previous analyzes, we conclude that the IoU improvement is especially due to better performance near the objects' boundary. Qualitatively (overlapping of segmented regions in Fig. 5) and quantitatively (error curve comparison in Fig. 6), on SS task, we find an improved performance near boundaries by adding a multi-task approach to the hourglass models. Besides, auxiliary tasks in hourglass models

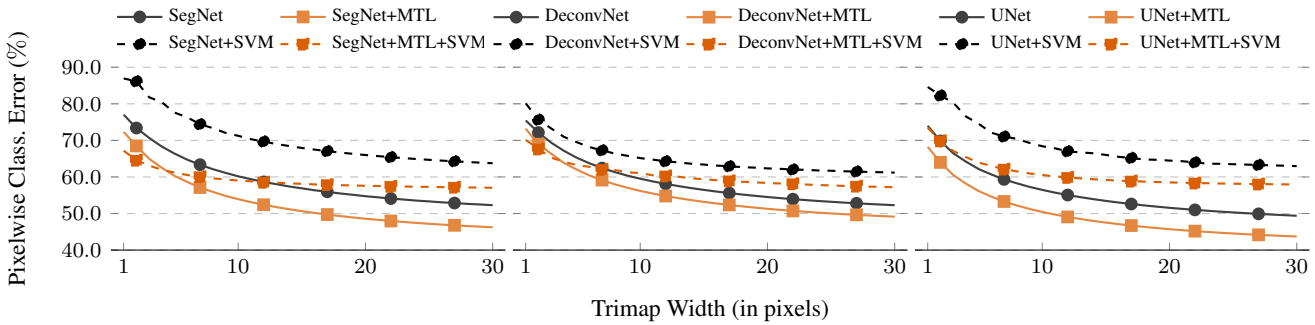
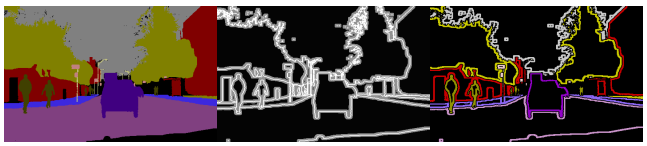


FIGURE 6. Pixelwise classification error vs. trimap width for the hourglass models focused on semantic segmentation (SegNet [17], DeconvNet [16], and UNet [15]) on the CamVid dataset. Circle marks represent the base network, and square ones represent the addition of MTL. The dashed-line lighter-tone denotes multi-label segmentation, while the solid-line darker-tone represents binary segmentation (also denoted with a +SVM).



(a) Ground truth (b) Trimap (8 px) (c) Evaluation region

FIGURE 7. Illustration of boundary accuracy evaluation using Trimap [10], [90]. (a) The image ground-truth from the Camvid dataset. (b) The Trimap used for measuring the pixel boundary labeling accuracy (gray region) with a width of 8 pixels. And, (c) an example of the evaluation region.

TABLE 3. IoU results on the CamVid test set for semantic segmentation.

Model	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Cyclist	mIoU w/ MTL	mIoU w/o MTL
ENet [51]	72.95	63.58	83.16	75.57	31.03	92.93	41.84	15.35	24.19	76.28	42.68	56.32	51.36
DeconvNet [16]	76.88	67.99	86.91	78.67	27.74	93.55	41.80	26.52	25.83	78.22	46.56	59.15	48.93
SegNet [17]	78.17	71.05	88.42	80.65	39.38	93.75	46.88	34.46	28.14	78.94	48.68	62.59	55.69
UNet [15]	**79.61**	**73.21**	**89.17**	81.37	**42.41**	93.81	58.00	32.65	31.34	79.94	47.98	64.50	56.12
FCN8 [3]	78.84	71.82	85.14	**84.60**	40.69	94.11	54.19	**40.48**	29.35	80.61	**52.19**	64.73	57.10
CGBNet [23]	79.35	72.02	85.97	82.43	40.86	94.30	56.48	38.48	31.11	80.72	50.85	64.78	58.86
FC-DenseNet67 [43]	79.07	71.38	86.48	84.59	40.44	**94.41**	**58.10**	39.85	**36.75**	**82.62**	50.45	**65.83**	**65.82**

encourage clustering behavior in similar feature maps (i.e., latent space). This behavior is reflected in Fig. 4, where we visualize that the latent space influenced by the multi-task approach (contour-based tasks) is not spaced arbitrarily.

E. COMPARING RESULTS

Finally, we report comparative results (models with and without MTL approach) of several hourglass models existing in the literature. We present our quantitative results for the CamVid, Cityscape, and Freiburg Forest datasets in Tables 3, 4, and 5, respectively. We use the IoU metric (higher is better) for each class on all datasets. Note that the second to last and last columns (in all tables) show the mean IoU of the classes for each model, with (w/) and without (w/o) MTL.

In Fig. 8, we show qualitative results using the AdapNet++, UNet, and FastNet models. The columns from left to right represent the ground-truth image, the prediction without and with MTL, and a comparison (overlap) of both prediction maps. In the comparison image, the green regions are correctly segmented. The red color represents regions erroneously segmented by original models (AdapNet++, UNet, or FastNet),

TABLE 4. IoU results on Cityscapes validation set for semantic segmentation, using 11 classes and with crop size of 384×768.

Model	Sky	Building	Road	Sidewalk	Fence	Vegetation	Pole	Car	Sign	Person	Cyclist	mIoU w/ MTL	mIoU w/o MTL
ParseNet [40]	92.68	89.16	96.65	78.68	38.89	90.31	51.26	92.23	69.63	72.51	71.13	76.65	71.02
DeconvNet [16]	93.38	89.30	96.88	77.75	47.10	90.94	53.39	92.33	62.90	70.13	69.29	76.67	62.03
FCN8 [3]	92.49	89.24	96.94	77.98	49.74	90.24	49.10	91.91	65.96	70.76	70.78	76.83	59.98
FastNet [41]	93.04	89.37	96.95	78.79	48.77	90.31	53.64	92.09	68.72	71.25	69.62	77.50	68.52
AdapNet++ [12]	93.07	89.46	97.06	80.03	49.46	90.58	52.10	92.22	66.26	72.88	70.62	77.61	**72.79**
CGBNet [23]	92.98	89.40	96.66	77.60	42.80	91.88	57.52	91.14	73.29	75.26	71.18	78.16	**73.25**
FC-DenseNet67 [43]	**93.88**	89.73	96.81	77.81	49.14	89.94	**58.74**	92.34	66.77	**75.19**	69.63	78.18	72.50
SegNet [17]	93.75	**90.09**	**97.37**	**81.58**	**51.84**	**91.75**	56.85	**92.82**	**67.52**	72.61	**72.12**	**78.94**	52.17

TABLE 5. IoU results on Freiburg Forest test set for semantic segmentation, using 5 classes and with crop size of 384×768.

Model	Trail	Grass	Vegetation	Sky	Obstacle	mIoU w/ MTL	mIoU w/o MTL
FC-DenseNet67 [43]	79.89	80.38	85.41	92.04	34.61	74.47	73.76
FCN8 [3]	85.12	87.43	89.82	91.89	45.98	80.05	77.49
ParseNet [40]	86.30	87.73	90.20	91.97	**47.41**	80.72	78.98
FastNet [41]	86.90	**88.08**	**90.77**	**92.82**	45.88	80.89	**79.67**
DeconvNet [16]	87.16	87.42	90.48	92.79	47.16	81.00	78.04
CGBNet [23]	87.59	87.62	90.63	92.78	46.58	81.04	77.89
SegNet [17]	**88.04**	88.04	90.61	92.68	46.22	**81.12**	74.58

and the MTL models erroneously segment the blue ones. The regions incorrectly segmented by both predictions are purple.

Our experimental results show that adding a multi-task approach to the already defined hourglass models improves the SS task's performance. It is important to note that we add contour-based auxiliary tasks because the original models still exhibit the problem of spatial precision loss. As we saw, this problem is reflected in the boundary of the segmented objects (see Fig. 1).

Also, we present an efficiency comparison of the hourglass models (see Fig. 9) in the training and testing phase. We used the Cityscapes dataset with 17 500 training samples and 500 testing 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. Note that we only do this for an increasing sequence of combinations: S, SE, SEC, and SECD, due to our computational limitations. We executed the process five

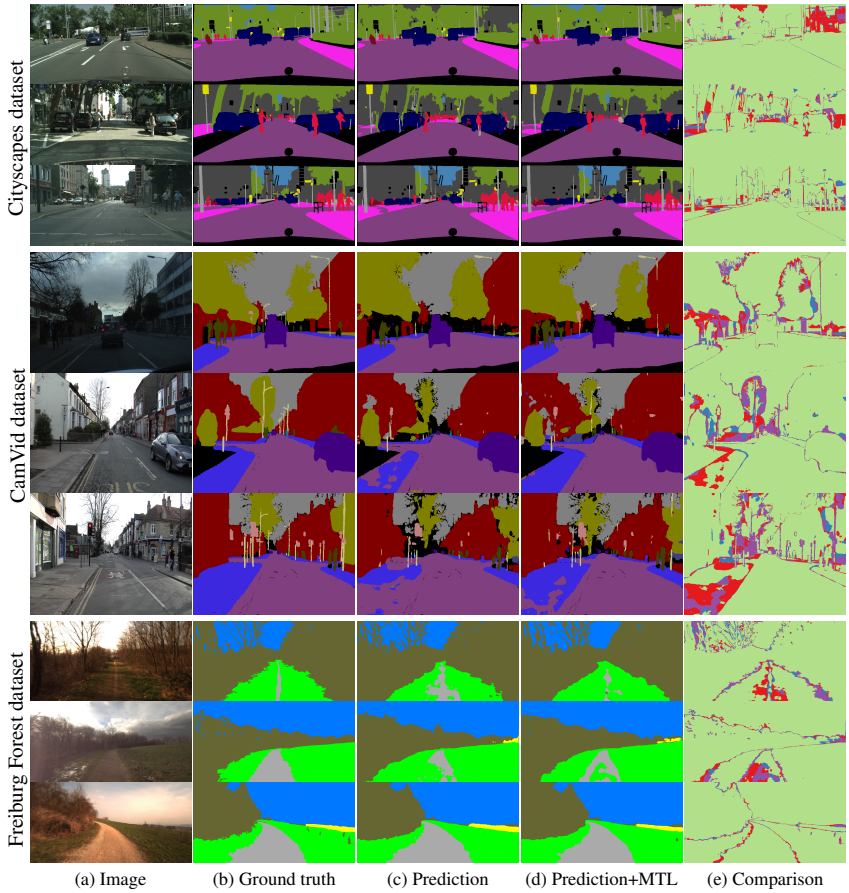


FIGURE 8. Qualitative comparison of the predictions of the AdapNet++ [12], UNet [15] and FastNet [41] models (without and with multi-task) against the ground-truth. The comparison column shows an overlap of the predictions against the ground-truth. Note, the green regions are correctly segmented. The red color represents regions erroneously segmented by original models (AdapNet++, UNet, and FastNet), and the blue ones are erroneously segmented by models with multi-task. The regions incorrectly segmented by both predictions are purple.

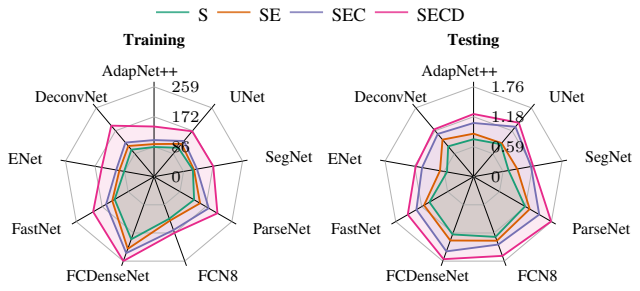


FIGURE 9. Comparison of the execution time of hourglass models (for a growing combination of tasks, i.e., S, SE, SEC, SECD) in the training phase (left) and in the testing phase (right), both in minutes. We use the Cityscapes dataset with 17 500 and 500 samples for training and testing, respectively. Note that the auxiliary tasks are only used in the training phase. In the testing phase, we only use the time shown in blue (right).

times and report the averages of the training (left) and testing phase (right), both in minutes. Our graphics on training using multitasking show an increase in the time required to train hourglass models. This increment is directly related to the complexity of the contour-based auxiliary tasks, which proved to be challenging enough to fit a latent space. Remember, for testing; we only use a single task: semantic segmentation. In other words, for 500 samples, the models need the time presented by the segmentation plot (S) in the right of Fig. 9. Finally, the previous experiments have been conducted on NVIDIA GTX Titan X with 12 GB of memory and four GPUs (multi-GPU).

F. DISCUSSION

In the previous experiments, we showed that by learning multiple contour-based tasks on the hourglass models the redundant information needed to solve the tasks improves the models' learning. We noticed that the tasks increase the models' ability to accommodate noise during the training phase. Consequently, the tasks reduce the model's overfitting risk by providing a gradient that tends to keep the latent space from overfitting. This advantage in the feature space is largely due to the clustering behavior of latent space that we achieved by a more robust feature extraction.

The addition of auxiliary tasks changes the weight updating dynamics (i.e., gradient updating) such that the model learns robust features that work in all the used tasks. The robustness of the space comes from adding related tasks to the blob prediction one (i.e., the SS task). Hence, by unsupervisedly

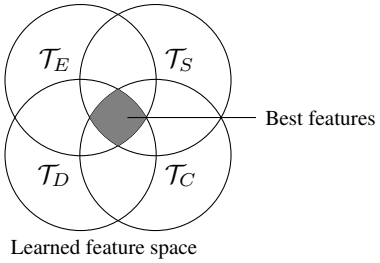


FIGURE 10. Example of the latent space (i.e., feature representations) learned by backpropagation when using several auxiliary tasks. The representations at the intersection of all the tasks are better since they satisfy several tasks simultaneously.

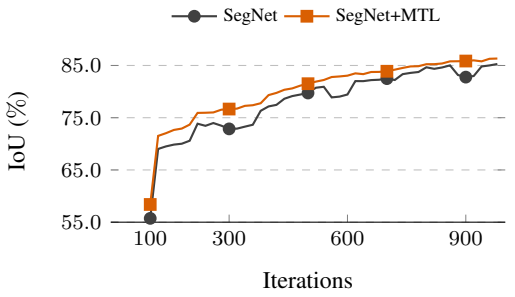


FIGURE 11. IoU over the iterations in the training phase of the hourglass model SegNet [17] focused on the semantic segmentation task in the CamVid dataset. By adding contour-based auxiliary tasks in the training phase, we achieve a robust feature extraction that increases the ability to treat noise and reduces the risk of overfitting. In consequence, the model that uses the MTL achieves a more stable learning than its counterpart.

restricting the latent space through more tasks we robustify the latent space, as show by our experiments.

For example, one of the main problems with SS methods is to correctly predict the boundaries of the objects since most of the accuracy comes from correctly detecting the main blob. By adding a contour-based auxiliary task, we increase the learning rate's effectiveness for this case. This increment happens since the same model is forced to understand the boundaries of the objects to predict the contours while been asked to predict the blobs as well (through the other task). We found that by simultaneously learning to solve related tasks the learned features improve with the tasks added (cf. Fig. 4). This result is intuitive if we assume that the different tasks have a common latent space, as show in Fig. 10. Then, by simultaneously optimizing in this shared space, our multi-task learning problem is finding solutions in the intersection of the tasks which in turn improves the others.

When training multiple tasks simultaneously, from the model point of view, the hidden units in the hourglass models improve two-fold: (i) There is a more significant number of parameters involved in updating the weights (i.e., backpropagation) using MTL, and (ii) the most relevant parameters (i.e., the most frequently influenced by all tasks) achieve a better (robust) feature extraction. This adjustment on the parameters of the MTL hourglass models has a regularization effect on the parameters in addition to a more stable training (i.e., the variance in the loss function plot shown in Fig. 11).

Unlike other architectures (e.g., DeepLabV3), hourglass models have a decoding stage that is complex enough (i.e., a set of reconstruction operations) to highlight the changes produced in the latent space. For this reason our research is focused on hourglass models. Note that each addition to the auxiliary tasks has a different effect on what is learned in the latent space. Changes in architecture can alter, in different ways, the way backpropagation benefits from contour-based auxiliary tasks.

V. FUTURE WORK

In our work, we observed the clustering behavior of the latent space. Future work may focus on using a clustering framework to impose particular biases to learn the latent representations. By forcing the latent space into clusters, we intuit that we will need fewer tasks in the training phase. Another venue to explore is the influence of the auxiliary contour-based tasks on architectures other than hourglass-based.

Regarding extending our work to videos, on one hand, we need to extend the architectures to work with 3D data. Thus, we will need to use bigger models that rely on 3D convolutions to perform the segmentation on volumetric data. On the other hand, we will need to maintain not only spatial consistency but also temporal one. This new constraint will be akin the problems we face today trying to reduce the instabilities in the segmentation boundaries. Hence, future work will need to find relevant tasks that help to stabilize the temporal consistencies as well. Perhaps, we could explore optical flow as a first approach to tackle since it is similar to the spatial boundaries.

VI. CONCLUSIONS

In this paper, we incorporated auxiliary contour-based tasks to address the loss of spatial precision. This problem is commonly in bounding segmented objects. Thus, we propose to use edge detection and semantic contour tasks to reinforce the semantic information on the boundary objects. We also proposed using quantized distance transform to add geometric information into the internal representation of deep neural networks (i.e., hourglass models). We observed (by empirical experiments) that the latent space behavior in hourglass models is clustering when adding complementary information (due to auxiliary tasks). Note that the latent space does not present a random distribution. Instead, better-distributed clusters produce, in turn, better segmentation results. We also showed that when using all the tasks, the activation maps (regions used by the networks to perform the segmentation prediction) adjust better to the edges of objects. Although the activation maps vary depending on the input image, the latent space's behavior produces an improvement in the quality of segmentation. Additionally, we verify (empirically) that the improvement produced by using multiple tasks addresses the problem of loss of spatial precision in segmentation. In other words, we verified (by using trimap) that using the clustered latent space improves the edges of the segmented objects and, consequently, the final segmentation. We also interpret that by adding contour-based auxiliary tasks, the models obtain a more powerful generalization. In order not to limit our

study (by using three models), we compared the results of the different hourglass models (with and without MTL) existing in the literature on other datasets (Cityscapes and Freiburg Forest). Finally, the empirical exploration showed that it is possible to better fit the models (obtain a latent space with cluster behavior) for the semantic segmentation task when we use complementary information (by adding contour-based auxiliary tasks) in the training phase.

APPENDIX A. FINAL REPRESENTATIONS

The methodology proposes to combine information on similar tasks using supervised learning. Then, we need to know the operations used to obtain the comparison masks in the different tasks. Keep in mind; all datasets perform the same preprocessing to obtain the edges and the quantized distance transform. Thus, to obtain the objects' edges, we take the instances' masks, and we look for a difference of instance labeling. For this, we used D-4 connectivity (up, down, right, left) [1], and to better highlight the boundaries, we use the morphology operation of dilation [1], with a structural element of 2-size and disk-shaped.

On the other hand, for adding geometric information, we extract the distance of pixels to objects' boundaries (i.e., distance transform [1]). Using this distance transform as a task learning gives us the following advantages: i) we can easily extract it from the instance masks, and ii) the quantized distance transform can be easily trained with the existing loss functions. Note that this representation, based on the distance transform, allows us to infer the complete shape of an object instance even with incomplete information (i.e., when a part of the object is shown).

The distance transform produces a wide range of values when objects have different shapes and sizes, see Fig. A.1(b). For this reason, we truncate the transformation given a threshold R, thus guaranteeing a limited range of values, see Fig. A.1(c).

Therefore, similar to models [74], [76], we define Q as the set of pixels on the object boundary the object and IS_i the set of pixels belonging to instance mask i. For every pixel p, we compute a truncated distance $D_t(p)$ to Q as,

$$D_t(p) = \gamma_p \min\left(\min\lceil d(p,q)\rceil, R\right), \quad \forall\, q \in Q \tag{A.1}$$

where $d(p,q)$ represent the Euclidean distance between pixel p and q, $\lceil z \rceil$ give us the nearest integer larger than z, and R is the truncation threshold. Finally, the function γ_p denotes if the pixel p is inside or outside of an IS_i instance mask

$$\gamma_p = \begin{cases} 1, & \text{if } p \in \text{IS}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{A.2}$$

To facilitate the energy labeling (i.e., continuous distance values), we quantify these values in K uniform bins by one-hot encode the distance map into a binary vector representation $b(p)$ as [74]

$$D_q(p) \sum_{k=1}^{K} r_n b_k(p), \qquad \sum_{k=1}^{K} b_k(p) = 1, \tag{A.3}$$

where r_n is distance value corresponding to bin k. The K binary maps are the classification maps for each of the k-th edge distance. We can see an example in Fig A.1(d).

This quantized distance transform operation is not new, having been explored in previous models [74], [76]. However, contrary to others that use this transformation in a bounding box [74] or for a single class (i.e., buildings) [76], our technique applies the transformation for instances that belong to different classes.

APPENDIX B. CLASS IMBALANCE

In this work, we have class imbalance during training. The dataset imbalance causes (i) inefficient training, because it has few samples (of some kinds) in the training stage, the network may not observe all the samples; and, by having a small number of samples, (ii) the network can fall into overfitting and degenerate the model. To address this problem, we use median frequency (counting) balancing [94], defined by

$$\tau_c = \frac{\bar{f}}{f(c)}, \tag{B.1}$$

where $f(c)$ is the number of pixels of class c divided by the total number of pixels in images where c is present, and $\bar{f}$ is the median of these frequencies (counting). Finally, we use this class weighted in Sections C-B with α_i and C-C with μ_i.

APPENDIX C. LEARNING MULTI-TASK FRAMEWORK

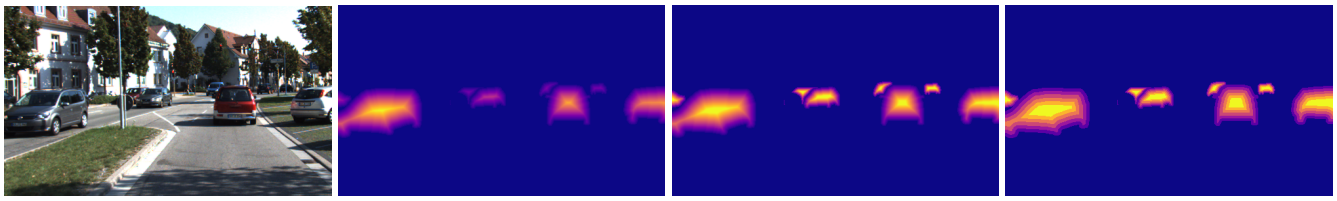
We used several hourglass networks based on the MTL approach, where tasks help each other adjust their parameters. With this approach, we get a good delimitation of the objects' edge by sharing the information extracted from all the tasks (i.e., share the parameters). In the last two layers of the decoding stage, we extract specific information to discriminate each task. Next, we explain details about the output learning using MTL for edge detection, semantic segmentation, semantic contours, and truncated distance transform (energy level).

A. EDGE DETECTION TRAINING

In the first specific decoding stage, we learn to detect the edges of each instance object. In order to handle the imbalance between the two binary classes (edge, no edge), we used the HED-loss function [95] a class-balanced cross-entropy function. Then we consider the edge-class objective function as

$$\begin{aligned} \mathcal{L}_c = &- \beta \sum_{i \in Y_+} \log P\left(y_i = 1 \mid X; \theta\right) \\ &- (1-\beta) \sum_{j \in Y_-} \log P\left(y_j = 0 \mid X; \theta\right), \end{aligned} \tag{C.1}$$

where y_i and y_j are the indexed predicted edge (output) for the i-th and j-th pixel, respectively. Here, θ represents the network parameters to be optimized in the edge stage. The proportion of positive (edge) and negative (no edge) classes on the ground-truth edges Y are $\beta = |Y_+|/|Y|$ and $1-\beta = |Y_-|/|Y|$, where $Y = Y_+ \cup Y_-$. Moreover, P is the probability



(a) Image (b) Distance Transform (c) Truncate Distance (d) Quantized Distance

FIGURE A.1. Intending to merge semantic and geometric information, we use a (d) quantized distance obtained from an (b) Euclidean distance transform. To enhance the distance transform energy, (c) we use truncation and normalization. Note that this quantized distance is easy to combine with the multi-label cross-entropy loss function.

that a pixel contained edges (output of edge stage), this is defined by a sigmoid function, such that

$$P = P\left(y_i \mid X; \theta\right) = \sigma(y_i) \in [0, 1]. \tag{C.2}$$

Although the HED-loss function proved to be useful in training for edge detection. Training time can be reduced and edges further penalized by maximizing intersection-over-union [90]. Then, we consider the objective function,

$$\mathcal{L}_{iou} = 1 - \frac{P \cap Y}{P \cup Y} = 1 - \frac{\sum_{v \in Y} P_v Y_v}{\sum_{v \in Y} P_v + Y_v - P_v Y_v}. \tag{C.3}$$

Finally, for edge detection, we combine both loss functions to obtain our final objective function, defined by

$$\mathcal{L}_E = \psi_1 \mathcal{L}_c + \psi_2 \mathcal{L}_{iou}, \tag{C.4}$$

where ψ_1 and ψ_2 are hyper-parameters that define the contribution of each loss to the learning process.

B. SEMANTIC SEGMENTATION TRAINING

In the second specific decoding stage, we learn to classify each object in pixel-wise level (i.e., semantic segmentation). We use a multi-label balanced cross-entropy loss function to address the problem of imbalance. Thus we define this function as,

$$\mathcal{L}_{cross\ ss} = -\frac{1}{N} \sum_{i=1}^{N} \alpha_i \log P(s = s_i \mid X; \phi), \tag{C.5}$$

where s_i is the indexed predicted classification (output) for the i-th class from the set of ground-truth S on semantic segmentation, additionally, N is the number of classes, and ϕ denotes the network parameters to optimize in the semantic segmentation stage. Also, $P(\cdot)$ is the probability that a pixel belongs to the ith class. Similar to (C.2), this function is defined by a sigmoid activation function.

Besides, similar to the previous section, we use a target function intersection-over-union to penalize the boundary of segmentation. Contrary to $\mathcal{L}_{iou}$ (C.3), at this stage, we use a multi-label function, defined by

$$\mathcal{L}_{iou\ ss} = 1 - \sum_{i=1}^{N} \frac{P_i \cap S_i}{P_i \cup S_i}. \tag{C.6}$$

Subsequently, we combine both loss functions in our final objective function for semantic segmentation. Thus, we define the function as,

$$\mathcal{L}_S = \psi_3 \mathcal{L}_{cross\ ss} + \psi_4 \mathcal{L}_{iou\ ss}, \tag{C.7}$$

where ψ_3 and ψ_4 are hyper-parameters used to control the influence of each part of the function (i.e., weighted sum). Keep in mind that the semantic contour task uses the same loss functions $\mathcal{L}_S$ with the name $\mathcal{L}_C$ but with hyper-parameters ω.

C. ENERGY LEVEL TRAINING

In the last specific decodification stage, we learn to classifier the bins of each level of the truncated distance transform. In other words, train the network (with φ parameters) to learn how to classify each level of the discretized distance transform (i.e., K bins classifier). Thus, similar to the previous section, we use multi-label balanced cross-entropy loss function,

$$\mathcal{L}_{cross\ e} = -\frac{1}{K} \sum_{i=1}^{K} \mu_i \log P(k = k_i \mid X; \varphi), \tag{C.8}$$

and multi-label intersection-over-union loss function,

$$\mathcal{L}_{iou\ e} = 1 - \sum_{i=1}^{K} \frac{P_i \cap K_i}{P_i \cup K_i}, \tag{C.9}$$

on a set of K bins.

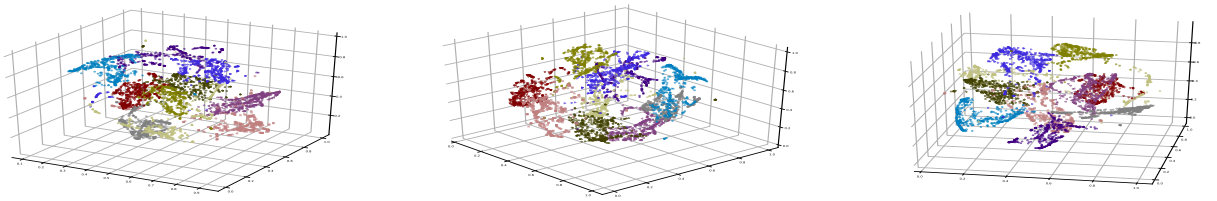
Finally, we merge our loss functions for energy level by,

$$\mathcal{L}_D = \psi_5 \mathcal{L}_{cross\ e} + \psi_6 \mathcal{L}_{iou\ e}, \tag{C.10}$$

where ψ_5 and ψ_6 are hyper-parameters that define the contribution of each loss to the learning process.

APPENDIX D. MORE VISUALIZATIONS OF LATENT SPACE BEHAVIOR

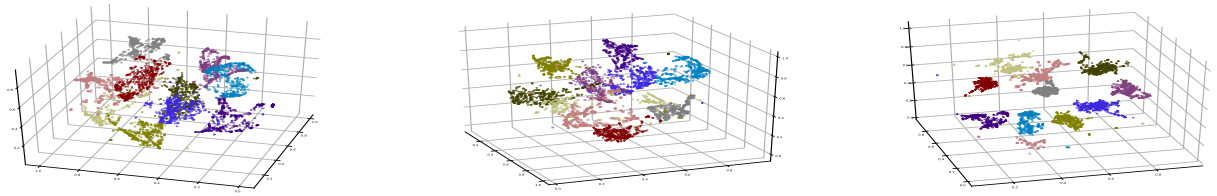
In this section, we show complementary visualizations in Fig. D.1 of the main document Fig. 4. We present additional plotting of the behavior visualizations of the latent space on a subset of the CamVid dataset (in Fig. D.2). We build the subset with 10 random image samples. From these images, we selected 446 random label-pixels for each class (all on the testing set).



(a) S
SSI = 0.384, DBI = 1.360, DBI = 1.360

(b) S+E
SSI = 0.391, DBI = 1.141, DBI = 1.141

(c) S+D
SSI = 0.394, DBI = 1.275, DBI = 1.275

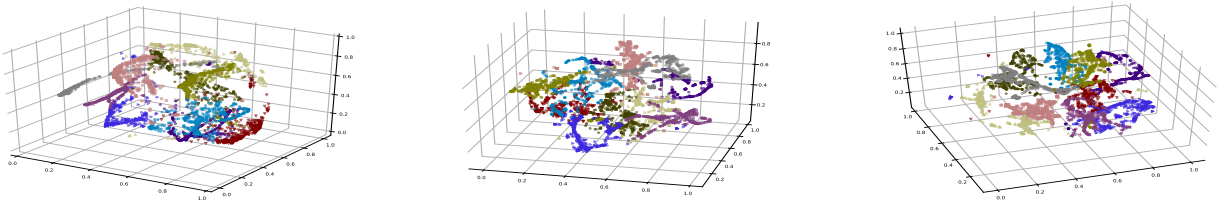


(d) S+E+C
SSI = 0.437, DBI = 1.150, DBI = 1.149

(e) S+E+D
SSI = 0.444, DBI = 0.920, DBI = 0.920

(f) S+E+C+D
SSI = 0.636, DBI = 0.774, DBI = 0.774

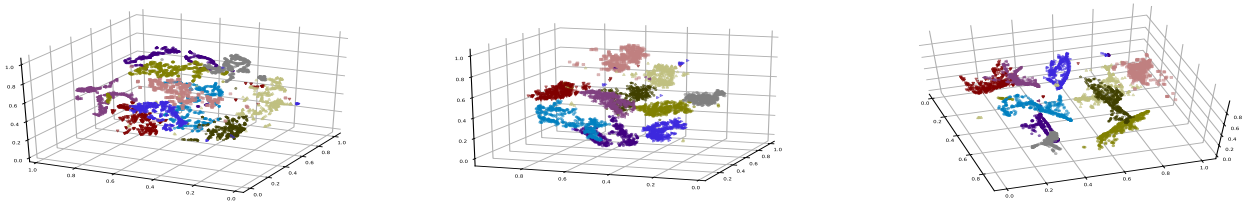
FIGURE D.1. We are plotting of the shared latent space on Camvid testing dataset. Here we combine the different tasks of edge detection (E), semantic segmentation (S), semantic contour (C), and distance transform (D). Note that when adding tasks related to semantic segmentation, i.e., by providing complementary information, maps of similar features (within a multi-task hourglass model) are clustered together in a similar latent space, and they are not spaced arbitrarily. We confirm this behavior by using a set of metrics for clustering shown in Table 2.



(a) S
SSI = 0.325, CHI = 1468.3, DBI = 1.572

(b) S+E
SSI = 0.342, CHI = 1678.9, DBI = 1.114

(c) S+D
SSI = 0.365, CHI = 1660.9, DBI = 1.275



(d) S+E+C
SSI = 0.413, CHI = 1806.5, DBI = 0.966

(e) S+E+D
SSI = 0.508, CHI = 2578.5, DBI = 0.922

(f) S+E+C+D
SSI = 0.510, CHI = 2795.3, DBI = 0.776

FIGURE D.2. Additional results of the shared latent space for the dataset subset (random labeled-pixels sample on CamVid testing dataset). Merging tasks of edge detection (E), semantic segmentation (S), semantic contour (C), and distance transform (D).

APPENDIX E. ARCHITECTURES

The architectures for the semantic segmentation models used in this paper are the same as those used in the original papers. We keep the same number of convolution and deconvolution layers for the encoding and decoding stages. We maintain the same amount of hidden units for each, that is, channels per layer, and we maintain the same non-linear activation functions and hyperparameters. The setup of the hourglass models is defined in their respective papers for the architectures we used, namely, FCN8 [3], ParseNet [40], SegNet [17], FastNet [41], UNet [15], DeconvNet [16], AdapNet++ [12], CGBNet [23], FC-DenseNet67 [43], and ENet [51].

Finally, for each specific task-block, we use two capable of convolution with kernels of 1×1 and 8×8, respectively, both with depth (channels) of the same number of classes of the dataset for tasks S and C, depth of 1 for task B and depth of 6 for task D.

REFERENCES

[1] R. Gonzalez and R. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, 2006.
[2] S. S. Chouhan, A. Kaul, and U. P. Singh, "Image segmentation using computational intelligence techniques: Review," *Arch. Comput. Methods Eng*, vol. 26, no. 3, pp. 533–596, Jul. 2019.
[3] 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.
[4] N. M. Zaitoun and M. J. Aqel, "Survey on image segmentation techniques," *Procedia Comput. Sci.*, vol. 65, pp. 797–806, 2015.
[5] 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.
[6] 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.
[7] 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.
[8] F. Lateef and Y. Ruichek, "Survey on semantic segmentation using deep learning techniques," *Neurocomputing*, vol. 338, pp. 321–348, 2019.
[9] 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.
[10] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2011, pp. 109–117.
[11] M. Amirul Islam, M. Rochan, N. D. Bruce, and Y. Wang, "Gated feedback refinement network for dense image labeling," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 3751–3759.
[12] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *Inter. J. Comput. Vis.*, pp. 1–47, 2019.
[13] N. Alshammari, S. Akçay, and T. P. Breckon, "Multi-task learning for automotive foggy scene understanding via domain adaptation to an illumination-invariant representation," *arXiv*, no. arXiv:1909.07697v1, 2019.
[14] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *Inter. Conf. Learn. Represent. (ICLR)*, 2016.
[15] 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.
[16] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1520–1528.
[17] 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.
[18] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv*, no. arXiv:1707.08114v2, 2017.
[19] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv*, no. arXiv:1706.05098v1, 2017.
[20] K.-H. Thung and C.-Y. Wee, "A brief review on multi-task learning," *Multimedia Tools Appls.*, vol. 77, no. 22, pp. 29 705–29 725, Nov. 2018.
[21] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: Improving semantic image segmentation with boundary detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 135, pp. 158–172, 2018.
[22] H. Ding, X. Jiang, B. Shuai, A. Q. Liu, and G. Wang, "Context contrasted feature and gated multi-scale aggregation for scene segmentation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 2393–2402.
[23] ——, "Semantic segmentation with context encoding and multi-path decoding," *IEEE Trans. Image Process.*, vol. 29, pp. 3520–3533, 2020.
[24] H. Ding, X. Jiang, A. Q. Liu, N. M. Thalmann, and G. Wang, "Boundary-aware feature propagation for scene segmentation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 6819–6829.
[25] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 3150–3158.
[26] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2018.
[27] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 433–442.
[28] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *IEEE Inter. Conf. Comput. Vis. (ICCV)*. IEEE, 2011, pp. 991–998.
[29] 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.
[30] G. Borgefors, "Distance transformations in digital images," *Computer vision, graphics, and image processing*, vol. 34, no. 3, pp. 344–371, 1986.
[31] 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.
[32] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recogn. Lett.*, vol. 30, no. 2, pp. 88–97, 2009.
[33] 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.
[34] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2961–2969.
[35] 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.
[36] 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.
[37] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jul. 2017.
[38] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European Conf. Comput. Vis. (ECCV)*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 75–91.
[39] S. Liu, X. Qi, J. Shi, H. Zhang, and J. Jia, "Multi-scale patch aggregation (mpa) for simultaneous detection and segmentation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 3141–3149.
[40] W. Liu, A. Rabinovich, and A. C. Berg, "ParseNet: Looking wider to see better," *arXiv*, no. arXiv:1506.04579v2, 2015.
[41] 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.
[42] Z. Tian, T. He, C. Shen, and Y. Yan, "Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 3126–3135.
[43] 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.
[44] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun, "UPSNet: A unified panoptic segmentation network," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2019.
[45] 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.
[46] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," *arXiv*, no. arXiv:2005.10821, 2020.
[47] 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.
[48] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 1857–1866.
[49] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: criss-cross attention for semantic segmentation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 603–612.
[50] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 2403–2412.
[51] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," *arXiv*, no. arXiv:1606.02147v1, 2016.
[52] X. Li, H. Zhao, L. Han, Y. Tong, S. Tan, and K. Yang, "Gated fully fusion for semantic segmentation," in *AAAI Conf. Artif. Intell. (AAAI)*, vol. 34, no. 07, 2020, pp. 11 418–11 425.
[53] 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.
[54] 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.
[55] 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.
[56] 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.
[57] X. Jin, C. Lan, W. Zeng, Z. Zhang, and Z. Chen, "CASINet: content-adaptive scale interaction networks for scene parsing," *Neurocomputing*, vol. 419, pp. 9–22, 2021.
[58] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv*, no. arXiv:1706.05587v1, 2017.
[59] T. Ziegler, M. Fritsche, L. Kuhn, and K. Donhauser, "Efficient smoothing of dilated convolutions for image segmentation," *arXiv*, no. arXiv:1903.07992, 2019.
[60] 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.
[61] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yizhou, "FastFCN: Rethinking dilated convolution in the backbone for semantic segmentation," in *arXiv*, no. arXiv:1903.11816v1, 2019.
[62] 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.
[63] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.
[64] M. Long, Z. Cao, J. Wang, and S. Y. Philip, "Learning multiple tasks with multilinear relationship networks," in *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2017, pp. 1594–1603.
[65] Y. Zhang and Q. Yang, "An overview of multi-task learning," *Natl. Sci. Rev.*, vol. 5, no. 1, pp. 30–43, 2018.
[66] G. Obozinski, B. Taskar, and M. Jordan, "Multi-task feature selection," Statistics Department, UC Berkeley, Tech. Rep. 2.2, 2006.
[67] T. Jebara, "Multitask sparsity via maximum entropy discrimination," *J. Mach. Learn. Res.*, vol. 12, no. Jan, pp. 75–110, 2011.
[68] Y. Zhang, D.-Y. Yeung, and Q. Xu, "Probabilistic multi-task feature selection," in *Adv. Neural Inf. Process. Sys. (NeurIPS)*, 2010, pp. 2559–2567.
[69] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 3994–4003.
[70] Y. Fang, Z. Ma, Z. Zhang, X.-Y. Zhang, X. Bai *et al.*, "Dynamic multi-task learning with convolutional neural network." in *Inter. Joint Conf. Artif. Intell.*, 2017, pp. 1668–1674.
[71] Y. Liao, S. Kodagoda, Y. Wang, L. Shi, and Y. Liu, "Understand scene categories by objects: A semantic regularized scene classifier using convolutional neural networks," in *IEEE Inter. Conf. Robot. Autom. (ICRA)*. IEEE, 2016, pp. 2318–2325.
[72] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, "Multinet: Real-time joint semantic reasoning for autonomous driving," in *IEEE Int. Veh. Symp. (IV)*. IEEE, 2018, pp. 1013–1020.
[73] M. Klingner, A. Bar, and T. Fingscheidt, "Improved noise and attack robustness for semantic segmentation by using multi-task training with self-supervised depth estimation," in *IEEE Inter. Conf. Comput. Vis., Pattern Recog. Wksps. (CVPRW)*, 2020, pp. 320–321.
[74] Z. Hayder, X. He, and M. Salzmann, "Boundary-aware instance segmentation," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jul. 2017.
[75] C. Tan, L. Zhao, Z. Yan, K. Li, D. Metaxas, and Y. Zhan, "Deep multi-task and task-specific feature learning network for robust shape preserved organ segmentation," in *IEEE Inter. Sym. Biomed. Imag. (ISBI)*, 2018, pp. 1221–1224.
[76] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," in *IEEE Inter. Conf. Image Process. (ICIP)*. IEEE, 2019, pp. 1480–1484.
[77] S. Kong and C. C. Fowlkes, "Recurrent scene parsing with perspective understanding in the loop," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 956–965.
[78] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," in *British Mach. Vis. Conf. (BMVC)*, 2017.
[79] 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.
[80] D. Cheng, G. Meng, S. Xiang, and C. Pan, "FusionNet: Edge aware deep convolutional networks for semantic segmentation of remote sensing harbor images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 12, pp. 5769–5783, 2017.
[81] Y. Liu, M.-M. Cheng, D.-P. Fan, L. Zhang, J. Bian, and D. Tao, "Semantic edge detection with diverse deep supervision," *arXiv*, no. arXiv:1804.02864v3, 2020.
[82] D. Acuna, A. Kar, and S. Fidler, "Devil is in the edges: Learning semantic boundaries from noisy annotations," in *IEEE/CVF Inter. Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019, pp. 11 075–11 083.
[83] M. Thoma, "A survey of semantic segmentation," *arXiv*, no. arXiv:1602.06541v2, 2016.
[84] B. Cheol Song, D. Ha Kim *et al.*, "Metric-based regularization and temporal ensemble for multi-task learning using heterogeneous unsupervised tasks," in *IEEE Inter. Conf. Comput. Vis. Wksps. (ICCVW)*, 2019, pp. 0–0.
[85] P. Guo, C.-Y. Lee, and D. Ulbricht, "Learning to branch for multi-task learning," in *Inter. Conf. Mach. Learn. (ICML)*, 2020, pp. 8912–8921.
[86] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[87] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2010.
[88] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
[89] 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.
[90] 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.
[91] 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.
[92] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Stat.-Theory Methods*, vol. 3, no. 1, pp. 1–27, 1974.
[93] 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.
[94] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *IEEE Inter. Conf. Comput. Vis. (ICCV)*. Institute of Electrical and Electronics Engineers Inc., 2016, pp. 2650–2658.
[95] S. Xie and Z. Tu, "Holistically-nested edge detection," in *IEEE Inter. Conf. Comput. Vis. (ICCV)*, 2015, pp. 1395–1403.

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) 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 Assistant Professor at the Institute of Computing, University of Campinas, Brazil. His research interests are video understanding (including video classification, semantic segmentation, spatiotemporal feature modeling, and generation), and understanding and creating complex feature spaces.