CAM-SegNet: A Context-Aware Dense Material Segmentation Network for Sparsely Labelled Datasets

Yuwen Heng[a], Yihong Wu[b], Srinandan Dasmahapatra and Hansung Kim[c]
Vision, Learning and Control Research Group (VLC), School of Electronics and Computer Science (ECS), University of Southampton, U.K.

Keywords: Dense Material Segmentation, Material Recognition, Deep Learning, Scene Understanding, Image Segmentation.

Abstract: Contextual information reduces the uncertainty in the dense material segmentation task to improve segmentation quality. Typical contextual information includes object, place labels or extracted feature maps by a neural network. Existing methods typically adopt a pre-trained network to generate contextual feature maps without fine-tuning since dedicated material datasets do not contain contextual labels. As a consequence, these contextual features may not improve the material segmentation performance. In consideration of this problem, this paper proposes a hybrid network architecture, the CAM-SegNet, to learn from contextual and material features during training jointly without extra contextual labels. The utility of our CAM-SegNet is demonstrated by guiding the network to learn boundary-related contextual features with the help of a self-training approach. Experiments show that CAM-SegNet can recognise materials that have similar appearances, achieving an improvement of 3-20% on accuracy and 6-28% on Mean IoU.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

The dense material segmentation task aims to recognise the physical material categories (*e.g.* metal, plastic, stone, etc.) for each pixel in the input image. The material cues can provide critical information to many applications. One example is to teach a robot to perform actions such as "cut" with a tool. This action indicates that the robot should grasp a knife at the wooden grip and cut with the metal blade (Shrivatsav et al., 2019). We can also estimate the acoustic properties (how sound interacts with surroundings (Delany and Bazley, 1970)) from physical material categories to synthesise immersive sound with spatial audio reflections and reverberation (McDonagh et al., 2018; Kim et al., 2019; Tang et al., 2020).

One of the main challenges in the dense material segmentation task is that materials could have a variety of appearances in different contexts, such as objects and places (Schwartz and Nishino, 2020). For example, a metal knife is glossy under bright lighting condition, but a rusted metal mirror can be dull. In order to achieve high accuracy, an ideal network should know all possible combinations; thus a large dataset is necessary. However, the similarity between the appearances of different materials can make annotation work challenging. Consequently, material datasets are often sparsely labelled in terms of the number of images and the integrity of labelled material regions. For example, some training set segments in the Local Material Database (LMD) (Schwartz and Nishino, 2016; Schwartz and Nishino, 2020) cover only a small region of the material, as shown in the ground truth images (in the second column) in Figure 3 and Figure 4.

As suggested in (Schwartz and Nishino, 2020; Schwartz and Nishino, 2016), a possible solution is to train a network with small image patches cropped from material regions, without contextual cues, to force the network to learn from the visual properties of materials. It was also found that extra contextual information (*e.g.*object, place labels or feature maps) can reduce the uncertainty in identifying materials and increase the segmentation accuracy (Schwartz and Nishino, 2016; Hu et al., 2011; Sharan et al., 2013). To efficiently combine contextual and material features, we propose our Context-Aware Dense Material Segmentation Network (CAM-SegNet), which

[a] https://orcid.org/0000-0003-3793-4811
[b] https://orcid.org/0000-0003-3340-2535
[c] https://orcid.org/0000-0003-4907-0491

Heng, Y., Wu, Y., Dasmahapatra, S. and Kim, H.
CAM-SegNet: A Context-Aware Dense Material Segmentation Network for Sparsely Labelled Datasets.
DOI: 10.5220/0010853200003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 5: VISAPP*, pages 190-201
ISBN: 978-989-758-555-5; ISSN: 2184-4321
Copyright © 2022 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

consists of global, local and composite branches. The global branch is responsible for extracting contextual features from the full image, while the local branch is designed to learn the material features from image patches. Finally, the composite branch produces the final predictions from merged features. This paper demonstrates the efficiency of CAM-SegNet by adjusting the global branch to extract boundary-related contextual features with the loss function that measures the quality of boundaries of generated segments. Since existing datasets are sparsely labelled, we adopt a self-training approach to augment the training set with predicted pseudo labels. The networks are evaluated on the sparse LMD test set as well as our dense LMD (DLMD) test set, which contains eight densely labelled indoor scene images. Our main contributions are the following:

- A CAM-SegNet to combine extracted boundary features from the full image with material features learnt from the image patches.
- A self-training approach to augment sparsely labelled datasets to provide boundary features for the CAM-SegNet.

The proposed CAM-SegNet achieves an improvement of 3-20% on accuracy and 6-28% on Mean IoU against the state-of-the-art network architectures and single-branch approaches in the control group.

2 BACKGROUND

Dense Material Segmentation. There have been a few attempts (Farhadi et al., 2009; Zheng et al., 2014) to annotate the materials in existing datasets (Everingham et al., 2015; Silberman et al., 2012), but the number of categories covered is not enough to segment common scenes. Bell *et al.* (Bell et al., 2015) created the first dedicated material dataset which contains 3 million samples from 23 material categories. Since its training set contains only labelled isolated pixels, this dataset is hard to be used for robust dense segmentation networks (Xie et al., 2017; Chen et al., 2017; Long et al., 2015; Xie et al., 2020), which requires the training set to provide labelled segments. Recently, Schwartz and Nishino (Schwartz and Nishino, 2020) released the LMD, which contains 16 material categories and 5,845 images with segments that each covers a single material category. This dataset can be used to train segmentation networks in an end-to-end manner, but still has several problems. First, the number of samples is insufficient since the LMD is very diverse in terms of material categories and scenes. Second, the ground truth segments do not cover all pixels belonging to the same category, as shown in Figure 3, 4.

Global and Local Training. Global and local training is an approach to combine features extracted from original images by the global branch and image patches by the local branch. Chen *et al.* (Chen et al., 2019) adopted this approach to preserve local details when processing down-sampled images. Due to the memory bottleneck when processing high-resolution patches, they split these patches into multiple batches and gather the full feature maps with several forward steps. This method makes the feature combining process complicated and costs more training time. To reduce the training time, Zhang *et al.* (Zhang et al., 2020) reduced the number of trainable parameters by sharing the weights between local and global branches. Wu *et al.* (Wu et al., 2020) alleviated the training burden by proposing only critical patches to refine the global segmentation. Likewise, Iodice and Mikolajczyk (Iodice and Mikolajczyk, 2020) proposed to crop the extracted global feature maps into equal blocks as the local features. For the dense material segmentation task, our CAM-SegNet compensates for the lost features when training with a single branch alone. According to Schwartz (Schwartz, 2018), the network trained with original images tends to ignore material properties, while the network trained with patches drops contextual cues. Moreover, the LMD contains no high-resolution images so that our CAM-SegNet can jointly train the global and local branches in an end-to-end manner without a severe training burden.

Boundary Refinement. For dense material segmentation task, the neural network based methods may not predict the pixels near the boundary accurately, due to the lack of labelled pixels near the boundary to train the network (Schwartz, 2018, p. 75). Therefore, the dense Conditional Random Field (dense-CRF) (Krähenbühl and Koltun, 2013) is often used to refine the boundary quality (Bell et al., 2015; Schwartz and Nishino, 2016), which assumes that similar pixels should be classified as the same category. However, the downside is that the output of the dense-CRF is sensitive to the parameters tuned by grid search. To optimise the CRF parameters together with the network, Zhao *et al.* (Zhao et al., 2017a; Zhao et al., 2020) chose to refine the material segmentation with the CRFasRNN proposed by Zheng *et al.* (Zheng et al., 2015). In this paper, we compare two GPU trainable CRF variants, Conv-CRF (Teichmann and Cipolla, 2019) and PAC-CRF (Su et al., 2019), to speed up the training process.

Another method to refine the boundary quality is to use a loss function that measures the quality of the segmentation boundary. A possible choice is the boundary metric in (Csurka et al., 2013), which measures the overlap between the ground truth boundaries and the predicted segmentation boundaries. Bokhovkin and Burnaev (Bokhovkin and Burnaev, 2019) utilised the max pooling operation to detect the boundaries and make the boundary metric a differentiable loss function. Although experiments in (Kang et al., 2021; Bokhovkin and Burnaev, 2019) have shown that this loss function can help the network to optimise the predictions near the boundaries, the loss value may not decrease when used in isolation since it does not contribute to the segmentation accuracy directly. Therefore, the local branch features, which are designed to achieve high accuracy, are passed to the global branch to make sure that our CAM-SegNet can extract boundary features steadily.

Self-training. Semi-supervised learning is one possible way to improve the segmentation results with sparsely labelled datasets. It utilises both labelled and unlabeled pixels during training. Among all semi-supervised learning approaches (Zhu, 2005), self-training is the most simple yet efficient one to fill in unlabelled pixels with generated pseudo labels. Recent experiments show that this approach can achieve state-of-the-art segmentation performance with limited labelled samples (Le et al., 2015; Cheng et al., 2020; Zoph et al., 2020). Although the self-training method may introduce more misclassified labels as noise to the dataset compared with more robust methods based on a discriminator to control pseudo label quality (Souly et al., 2017), the noise can also prevent the network from overfitting (Goodfellow et al., 2016, p. 241) since the LMD is a small dataset. Therefore, we choose this self-training method to generate pseudo labels and provide the boundary information for our CAM-SegNet. In our experiments, we show that for the material segmentation task, self-training approach is not the factor that improves the performance. Instead, the combined boundary and material features are the reason why our CAM-SegNet can perform well.

3 METHODOLOGY

In this section, we present our CAM-SegNet for the dense material segmentation task. Figure 1 illustrates the overall network structure. The global branch takes the original image as input while the cropped patches are fed into the local branch. The encoders extract features from both branches independently and downsample the feature maps. The decoders recover the feature map size jointly (with the feature sharing connection) and generate the outputs for each branch. The composite branch crops and concatenates the global branch output O_G to the local branch output O_L. Then the network merges the upsampled feature maps, and generates the composite output O_C. The last convolutional layer is applied to patch feature maps $O_{C'}$, to ensure that the overall network still focuses on material information. Finally, the optional CRF layer can be used to refine the composite output O_C. While training, the contextual features extracted from the global branch is controlled by the loss function applied to the global branch output O_G. During evaluation time, only the composite output O_C is kept to generate the final segmentation. The algorithm used to crop the input images is described in Appendix section.

3.1 Feature Sharing Connection

The decoders in Figure 1 gradually upsample the feature maps with three convolutional blocks. To train the two branches collaboratively, at the input of each block, the feature maps are shared between the global and local branches through the feature sharing connection showed in detail in Figure 2. We define the feature maps as $X_G \in \mathbb{R}^{c\times h_G\times w_G}$ for the global branch, and $X_L \in \mathbb{R}^{b\times c\times h_L\times w_L}$ for the local branch. Here c represents the channel number, h,w are the height and width, and b is the number of patches. First, the global branch feature maps X_G are cropped into patches, $X'_G \in \mathbb{R}^{b\times c\times h_L\times w_L}$, and these patches are concatenated with the local branch feature maps. Then the network merges the patch feature maps X_L from the local branch to produce $X'_L \in \mathbb{R}^{c\times h_G\times w_G}$. Finally, the merged feature maps are concatenated with the global branch feature maps. The number of channels in the concatenated feature maps, X_{CG} and X_{CL}, are doubled to $2c$. To ensure that the global and local feature maps can match each other spatially, the same patch cropping method is used as the one used to crop the input images.

3.2 Context-Aware Dense Material Segmentation

The three outputs (O_G, O_L, O_C) generated from our CAM-SegNet make it convenient to control the features extracted from each branch, by optimising the branches to achieve different tasks with different loss functions. To optimise the CAM-SegNet, the total loss function L_{total} can be represented as

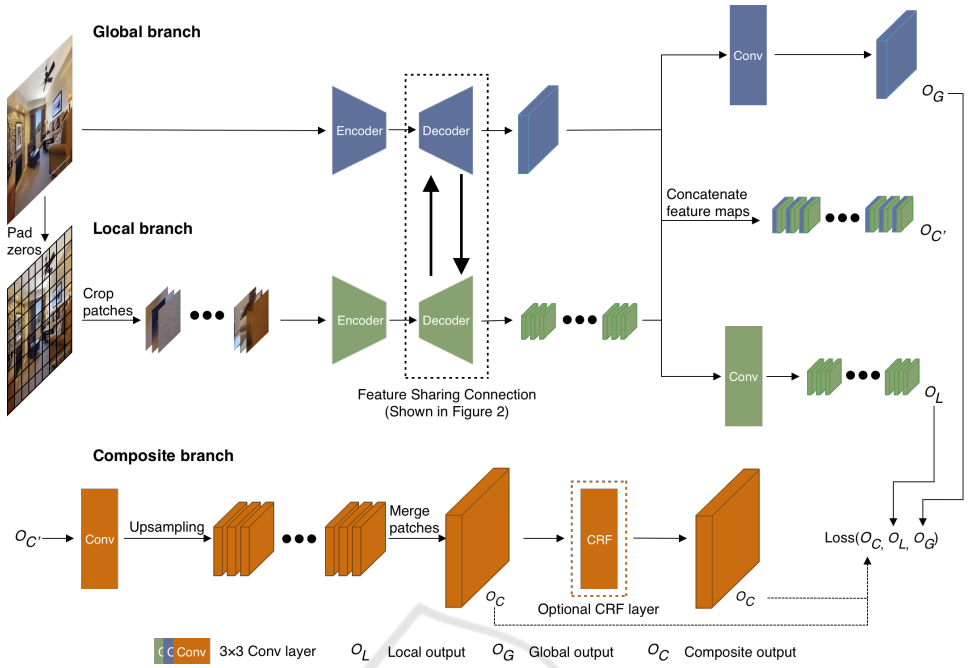

Figure 1: **CAM-SegNet** architecture. The feature maps in the decoders are shared between the global and local branches. After the encoder-decoder component, the feature maps at the same spatial location are concatenated together and passed into the composite branch, which upsamples the feature maps to the same size as the original input image. The composite output can be refined by an optional CRF layer.

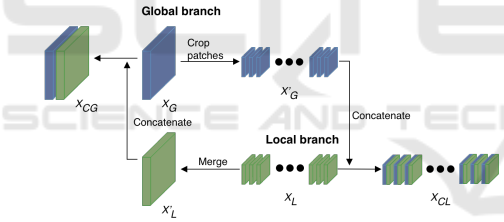

Figure 2: The feature sharing connection between the decoders. X_{CG} is the concatenated global branch feature maps, while X_{CL} is the concatenated local branch feature maps.

$$L_{total} = L_{global}(O_G, Y_{1/4}) + L_{local}(O_L, Y_{1/4}) + L_{composite}(O_C, Y) \quad (1)$$

where Y is the ground truth segment, and $Y_{1/4}$ is the downsampled ground truth segment. The downsampled ground truth is used to reduce the memory capacity needed during training. This paper aims to combine contextual and material features to generate dense material segmentation. According to Schwartz and Nishino (Schwartz and Nishino, 2016; Schwartz and Nishino, 2020; Schwartz, 2018), material patches without contextual cues can force the network to extract material features. Since the local branch is responsible to learn from image patches, it is optimised to provide material features with the focal loss (Lin et al., 2017b), *i.e.*, $L_{focal} = \frac{1}{N}\sum_i -(1-p_i)^3 log(p_i)$. Here N is the number of pixels in O_L, and p_i is the estimated probability of pixel i for the true category. Similarly, the global branch is optimised to provide contextual features since the original images contain contextual information. However, context labels (*e.g.*objects or places) are needed to extract corresponding contextual features. Although these features can reduce the material segmentation uncertainty (Schwartz and Nishino, 2016), the cost of extra labels is not desired.

Instead of exploring contextual features that need extra labels, our CAM-SegNet investigates the contextual information that is missing in the image patches — the boundary between different materials. For pixels along the boundaries of material c, let R^c, P^c be the recall and precision score. To provide boundary related features, the boundary loss (Bokhovkin and Burnaev, 2019), $L_{boundary} = \sum_c 1 - \frac{2R^cP^c}{R^c+P^c}$, is applied to the global branch output O_G, which aligns the predicted material boundaries with the ground-truth segments. Ideally, the composite branch should be able to generate predictions accurately with good boundary quality, if the composite branch can learn from the outputs from both branches properly. Therefore, we set the composite branch loss function $L_{composite}$ to $L_{boundary}(O_C, Y) + L_{focal}(O_C, Y)$, to ensure that it is optimised to achieve these two goals at the same time.

3.3 Self-training Approach

Since not all training segments in LMD cover the whole material region, the detected ground truth boundaries may provide misleading information to the boundary loss (Bokhovkin and Burnaev, 2019). Therefore, we choose to complete the labels first. A network is trained with the focal loss (Lin et al., 2017b) and sparsely labelled LMD as the initial teacher model to generate pseudo labels. We assume that the LMD augmented with pseudo labels can provide necessary boundary information for our CAM-SegNet. The teacher-student-teacher self-training approach (Zoph et al., 2020) contains four stages:

1. The initial teacher model is trained by setting all the loss terms in Equation 1 to L_{focal}.
2. The teacher model generates the feature maps for the training set, and replaces the known pixels with ground truth labels.
3. The feature maps are refined by the CRF layer, to produce the final pseudo labels with better material boundary.
4. A CAM-SegNet is trained as a student model with the augmented LMD.

To improve the pseudo label quality and achieve the best performance, typical self-training cases such as (Zoph et al., 2020; Le et al., 2015; Cheng et al., 2020) repeat this training approach many times, to produce a series of student models. In detail, the student model at round t is considered as the new teacher model, to produce a new augmented dataset with the second and third stages. Then this dataset is used to produce a new student model, S_{t+1} with the fourth stage. It is worth noting that, self-training may not work well if the initial teacher model cannot predict most of the labels correctly. According to Bank *et al.* (Bank et al., 2018), an initial accuracy of 70% is not enough. Since the reported material segmentation accuracy is about 70% in (Bell et al., 2015; Schwartz and Nishino, 2016; Schwartz and Nishino, 2020), we don't expect to achieve a much higher accuracy with the self-training approach. Instead, our objective is to show that the additional boundary information can help the network to generate segments with good boundary quality, and the self-training approach is one way to provide such information. The results in Table 3 illustrate how our network performs with and without the boundary information under the same self-training approach.

4 EXPERIMENTS

Dataset. We evaluate our proposed method on the local material database (LMD) (Schwartz and Nishino, 2016; Schwartz and Nishino, 2020), and follow their suggestion to crop the images into 48×48 patches. We randomly split all the samples into training (70%), validation (15%) and test (15%) sets. Since our contribution mainly focuses on indoor material segmentation, we qualitatively evaluate the segmentation results only with images taken in indoor scenes such as kitchens and living rooms.

Evaluation Metrics. We report the per-pixel average accuracy (Pixel Acc) and the mean intersection over union value (Mean IoU). It is worth pointing out that the sparsely labelled segments in LMD cannot reflect segmentation quality, especially for pixels near the material boundaries. Therefore, in addition to the LMD test set, we exhaustively labelled eight indoor images in the LMD test set to evaluate the performance of our model. We refer to these eight images as DLMD in our experiments.

Baseline Models. Our main contribution is to combine both global contextual features and local material features to achieve dense material segmentation. To show the advantage of our model among state-of-the-art networks for image segmentation task, we use DeepLabV3+ (Chen et al., 2018), BiSeNetV2 (Yu et al., 2020), and PSPNet (Zhao et al., 2017b) as our baselines. In our experiments, we fine-tune the pre-trained models implemented by (Yakubovskiy, 2020). Since these networks have not been evaluated on the LMD previously, we adopt the training procedures from their original papers and use the same backbone described below as our CAM-SegNet. The results are refined by the same CRF layer for a fair comparison.

Implementation Details. The ResNet50 (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009) is used as the encoder and the Feature Pyramid Network (FPN) (Lin et al., 2017a) is used as the decoder. The skip connections are added between the encoder and decoder as in (Chen et al., 2019). The patch size 48 is not divisible by the default encoder downsampling factor 32, which may cause a spatial mismatch between the local and global feature maps. Therefore, the downsampling factor is changed to 16, by setting the stride of the final block convolutional layer to 1. Since Schwartz and Nishino (Schwartz and Nishino, 2016; Schwartz and Nishino, 2020) did not release the segmentation task training configuration, we follow the work in (Bell et al., 2015) to normalise the im-

ages by subtracting the mean (124, 117, 104) for the RGB channels respectively. To refine the segmentation outputs, the trainable Conv-CRF (Teichmann and Cipolla, 2019) is adopted. First, the Adam optimiser with learning rate 0.00002 is used to train the network without a CRF layer. Then the network parameters are frozen to train the CRF layer with learning rate 0.001. Finally the network is refined together with the CRF layer with learning rate 0.0000001. Each stage is trained for 40 epochs. Since the images have different sizes, the gradients are accumulated to achieve an equivalent batch size of 32. According to Chen *et al.* (Chen et al., 2019), a mean squared error regularisation term between the global and local outputs can help the network to learn from both branches. This regularisation term is removed when the CRF layer is appended to the network, to encourage the branches to learn more diverse features. The self-training approach is repeated three times.

Quantitative Evaluation. In Table 1, we compare the performance of our CAM-SegNet against the baseline models. In order to illustrate the model performance for individual material, seven common materials that exist in indoor scenes from DLMD are chosen to report the Pixel Acc values. Our CAM-SegNet achieves 3.25% improvement in terms of Pixel Acc and 27.90% improvement on Mean IoU, compared with the second highest score achieved by DeepLabV3+. DeepLabV3+, BiSeNetV2 and PSPNet got low scores on materials of small objects, such as foliage (plants for decoration) and paper. Another observation from Table 1 is that these three networks can still achieve comparable performance when recognising materials that usually cover a large area of the image, such as plaster (material of the wall and ceiling) and wood (usually wooden furniture).

One reason for the low scores may be that the networks failed to learn from local material features such as texture. PSPNet relies on the pooling layers to learn from multi-scale features, DeepLabV3+ uses dilated convolutional layers. Although BiSeNetV2 adopts two branches to learn from local and global features, they all take the full-size images as input, and the intermediate layers do not communicate during training. The local features can fade out especially when the image resolution is low. As a consequence, these networks tend to depend on global features and may not recognise small material regions well.

In contrast, our CAM-SegNet adopts both full-size images and cropped patches, to learn from the global and local features, which are combined and co-trained. This enables our CAM-SegNet to recognise materials that are hard to identify (foliage and paper) for the baseline models.

Qualitative Evaluation. In Figure 3, we compared the segmentation quality of our CAM-SegNet with DeepLabV3+. As indicated by the Mean IoU score, CAM-SegNet is better at recognising pixels around material boundaries. In the kitchen image, we can see the clear boundary between the ceramic floor and the wooden cupboard. In the toilet image, the ceramic close-stool is successfully separated from the wall covered with plaster.

Ablation Study. In Table 2 and Table 3, we evaluate the effectiveness of each component of our method. The components include the network architecture, the loss function, and the CRF layer. For fairness, all models are trained with the same training procedure as our CAM-SegNet. In detail, to show the advantages of our two-branch architecture, we train two single branch models with full-size images and image patches separately, and refer to them as the Global and Local models respectively.

Since the LMD is sparsely labelled, it is not straightforward to train our CAM-SegNet without the self-training approach. In order to control for the influence of the self-training approach, we retrain our CAM-SegNet with the focal loss (Lin et al., 2017b) applied to all three outputs in Equation 1, and name this model as the Self-Adaptive CAM-SegNet (SACAM-SegNet). To avoid confusion, we refer to our CAM-SegNet trained with the boundary loss as the Boundary CAM-SegNet or BCAM-SegNet. We see from Table 2 that our SACAM-SegNet achieves an improvement of 12-20% on Pixel Acc and 6-19% on Mean IoU, compared with single branch models without the self-training approach. Although PAC-CRF refined models tend to get higher Pixel Acc, Conv-CRF refined models can achieve higher Mean IoU.

Figure 4 shows that our SACAM-SegNet can produce correct labels for pixels that are hard to recognise for the Global or Local models. For example, our SACAM-SegNet can label the window in the kitchen as glass correctly. Moreover, our model can ignore object boundaries and cover all adjacent pixels belonging to the same material category. A good example is the ceiling and the wall in the living room picture. Surprisingly, our SACAM-SegNet can even tell the difference between the scene outside the window and the scene drawing in the painting in the living room, and successfully classify them as glass and paper respectively. However, we also notice that the PAC-CRF refined SACAM-SegNet tends to predict wrong labels if the material region has rich textural

Table 1: Quantitative evaluation results for our CAM-SegNet and baseline models. The values are reported as percentage. The highest value for each evaluation metrics is in bold font. Seven common indoor materials are selected to report the performance Pixel Acc. The Pixel Acc is evaluated on both LMD (the first column) and DLMD (the second column). Since LMD test set provides sparsely labelled images, it is not meaningful to report Mean IoU on LMD. Therefore, Mean IoU is reported on DLMD only.

Models	ceramic	fabric	foliage	glass	paper	plaster	wood	Pixel Acc		Mean IoU
DeepLabV3+	**97.68**	27.56	0.00	48.91	0.00	**88.94**	73.69	71.37	67.09	32.04
BiSeNetV2	18.86	3.07	0.00	23.00	0.34	58.68	70.77	45.66	37.66	15.08
PSPNet	55.59	0.12	0.00	**66.73**	1.47	79.25	73.76	50.12	52.11	23.39
CAM-SegNet (ours)	92.65	**32.72**	**88.81**	21.99	**30.67**	87.77	**93.82**	**71.65**	**69.27**	**40.98**

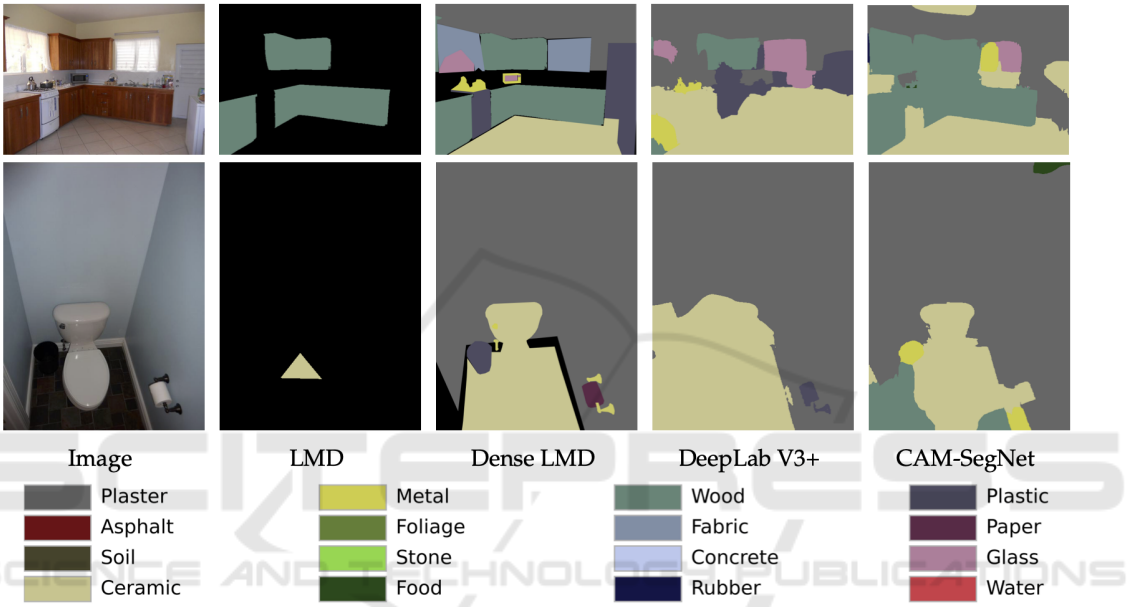

Figure 3: Qualitative comparison. The sparsely labelled images are taken from LMD test set, and densely labelled with all known material categories manually.

Table 2: Quantitative results for our SACAM-SegNet and single branch models. Our network outperforms single branch models.

Metric	CRF Layer	Local	Global	SACAM-SegNet
Pixel Acc	PAC-CRF	61.95	60.58	**69.25**
	Conv-CRF	58.07	55.67	**66.83**
Mean IoU	PAC-CRF	27.07	30.52	**32.25**
	Conv-CRF	31.77	32.25	**34.16**

clues. For example, the striped curtain covers the window in the kitchen. The PAC-CRF forces the network to label pixels between the stripes to different categories. This behaviour is not desired since it can give wrong boundary information. That is the reason why we choose to use a Conv-CRF refined model to generate the pseudo labels. More results can be found in the Appendix section.

Table 3 compares the performance between SACAM-SegNet and BCAM-SegNet with the self-training approach. Without boundary loss, the SACAM-SegNet performs worse compared with the BCAM-SegNet. This shows that self-training alone does not result in the good performance of our BCAM-SegNet. The boundary information can stabilise our CAM-SegNet to learn from noisy pseudo labels and gradually correct the pseudo labels to achieve higher accuracy. The qualitative comparison is shown in the Appendix section.

Table 3: Quantitative performance of our CAM-SegNet trained on augmented LMD with the self-training approach.

Models	SACAM-SegNet		BCAM-SegNet	
	Pixel Acc	Mean IoU	Pixel Acc	Mean IoU
Student 1	66.42%	37.93	67.38%	39.26
Student 2	67.26%	38.97	68.18%	39.81
Student 3	64.85%	32.19	**69.27%**	**40.98**

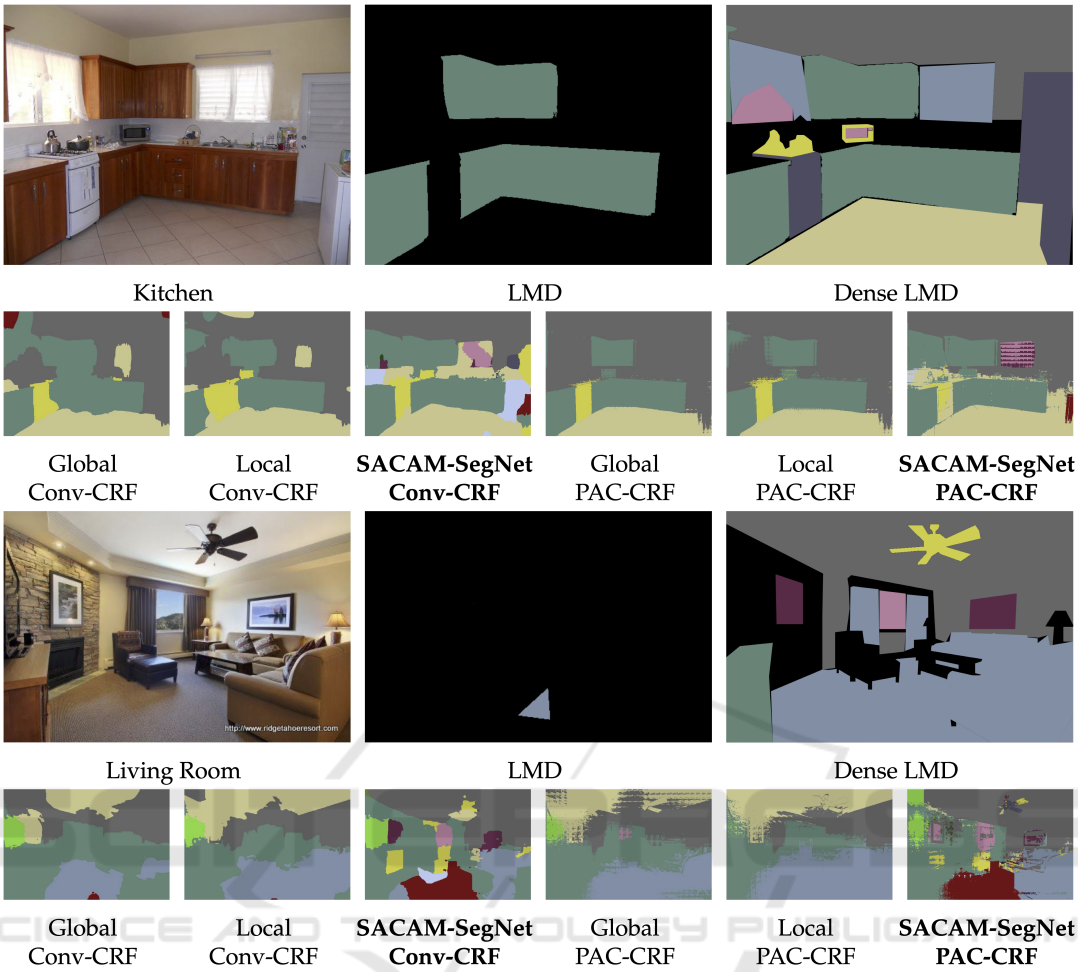

Figure 4: Dense material segmentation results for Kitchen and Living Room images.

5 CONCLUSIONS

This paper proposed a hybrid network architecture and training procedure to combine contextual features and material features. The effectiveness of the CAM-SegNet is validated with boundary contextual features. We show that the combined features can help the network to recognise materials at different scales and assign the pixels around the boundaries to the correct categories. In addition to boundary features, it is possible to enhance the network performance one step further with generated object and scene pseudo labels. We will investigate the possibility of combining multiple kinds of pseudo labels with semi-supervised training approach in future studies.

ACKNOWLEDGMENT

This work was supported by the EPSRC Programme Grant Immersive Audio-Visual 3D Scene Reproduction Using a Single 360 Camera (EP/V03538X/1).

REFERENCES

Bank, D., Greenfeld, D., and Hyams, G. (2018). Improved training for self training by confidence assessments. In *Science and Information Conference*, pages 163–173. Springer.

Bell, S., Upchurch, P., Snavely, N., and Bala, K. (2015). Material recognition in the wild with the materials in context database. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3479–3487.

Bokhovkin, A. and Burnaev, E. (2019). Boundary loss for remote sensing imagery semantic segmentation. In

International Symposium on Neural Networks, pages 388–401. Springer.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818.

Chen, W., Jiang, Z., Wang, Z., Cui, K., and Qian, X. (2019). Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8924–8933.

Cheng, H., Gu, C., and Wu, K. (2020). Weakly-supervised semantic segmentation via self-training. In *Journal of Physics: Conference Series*, volume 1487, page 012001. IOP Publishing.

Csurka, G., Larlus, D., Perronnin, F., and Meylan, F. (2013). What is a good evaluation measure for semantic segmentation?. In *BMVC*, volume 27, pages 10–5244.

Delany, M. and Bazley, E. (1970). Acoustical properties of fibrous absorbent materials. *Applied Acoustics*, 3(2):105–116.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

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

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. (2009). Describing objects by their attributes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hu, D., Bo, L., and Ren, X. (2011). Toward robust material recognition for everyday objects. In *BMVC*, volume 2, page 6. Citeseer.

Iodice, S. and Mikolajczyk, K. (2020). Text attribute aggregation and visual feature decomposition for person search. In *BMVC*.

Kang, J., Fernandez-Beltran, R., Sun, X., Ni, J., and Plaza, A. (2021). Deep learning-based building footprint extraction with missing annotations. *IEEE Geoscience and Remote Sensing Letters*.

Kim, H., Remaggi, L., Jackson, P. J., and Hilton, A. (2019). Immersive spatial audio reproduction for vr/ar using room acoustic modelling from 360 images. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 120–126. IEEE.

Krähenbühl, P. and Koltun, V. (2013). Parameter learning and convergent inference for dense random fields. In *International Conference on Machine Learning*, pages 513–521. PMLR.

Le, T. H. N., Luu, K., and Savvides, M. (2015). Fast and robust self-training beard/moustache detection and segmentation. In *2015 international conference on biometrics (ICB)*, pages 507–512. IEEE.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017a). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017b). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

McDonagh, A., Lemley, J., Cassidy, R., and Corcoran, P. (2018). Synthesizing game audio using deep neural networks. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*, pages 1–9. IEEE.

Schwartz, G. (2018). *Visual Material Recognition*. Drexel University.

Schwartz, G. and Nishino, K. (2016). Material recognition from local appearance in global context. In *Biol. and Artificial Vision (Workshop held in conjunction with ECCV 2016)*.

Schwartz, G. and Nishino, K. (2020). Recognizing material properties from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8):1981–1995.

Sharan, L., Liu, C., Rosenholtz, R., and Adelson, E. H. (2013). Recognizing materials using perceptually inspired features. *International journal of computer vision*, 103(3):348–371.

Shrivatsav, N., Nair, L., and Chernova, S. (2019). Tool substitution with shape and material reasoning using dual neural networks. *arXiv preprint arXiv:1911.04521*.

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer.

Souly, N., Spampinato, C., and Shah, M. (2017). Semi supervised semantic segmentation using generative adversarial network. In *Proceedings of the IEEE international conference on computer vision*, pages 5688–5696.

Su, H., Jampani, V., Sun, D., Gallo, O., Learned-Miller, E., and Kautz, J. (2019). Pixel-adaptive convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11166–11175.

Tang, Z., Bryan, N. J., Li, D., Langlois, T. R., and Manocha, D. (2020). Scene-aware audio rendering via deep

acoustic analysis. *IEEE transactions on visualization and computer graphics*, 26(5):1991–2001.

Teichmann, M. and Cipolla, R. (2019). Convolutional crfs for semantic segmentation. In *BMVC*.

Wu, T., Lei, Z., Lin, B., Li, C., Qu, Y., and Xie, Y. (2020). Patch proposal network for fast semantic segmentation of high-resolution images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12402–12409.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. (2020). Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

Yakubovskiy, P. (2020). Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch.

Yu, C., Gao, C., Wang, J., Yu, G., Shen, C., and Sang, N. (2020). Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *arXiv preprint arXiv:2004.02147*.

Zhang, H., Liao, Y., Yang, H., Yang, G., and Zhang, L. (2020). A local-global dual-stream network for building extraction from very-high-resolution remote sensing images. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhao, C., Sun, L., and Stolkin, R. (2017a). A fully end-to-end deep learning approach for real-time simultaneous 3d reconstruction and material recognition. In *2017 18th International Conference on Advanced Robotics (ICAR)*, pages 75–82. IEEE.

Zhao, C., Sun, L., and Stolkin, R. (2020). Simultaneous material segmentation and 3d reconstruction in industrial scenarios. *Frontiers in Robotics and AI*, 7.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017b). Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890.

Zheng, S., Cheng, M.-M., Warrell, J., Sturgess, P., Vineet, V., Rother, C., and Torr, P. H. (2014). Dense semantic image segmentation with objects and attributes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3214–3221.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537.

Zhu, X. J. (2005). Semi-supervised learning literature survey.

Zoph, B., Ghiasi, G., Lin, T.-Y., Cui, Y., Liu, H., Cubuk, E. D., and Le, Q. V. (2020). Rethinking pre-training and self-training. *arXiv preprint arXiv:2006.06882*.

APPENDIX

Crop the Images or Feature Maps. Algorithm 1 is designed to calculate the parameters when cropping the input images or feature maps. The same parameters are used to merge the patches to ensure the feature value at the corresponding position describes the same image region in the global and the local branch.

Algorithm 1: Calculate Patch Cropping Parameters.

procedure GETPATCHINFO(*PatchSize*, *S*) ▷ *S* is the height or width of the original image
  Initialize
    $num\_patch \leftarrow 0$ ▷ Number of patches cropped along one dimension
    $stride \leftarrow 0$ ▷ Number of pixels to next patch
    $pad \leftarrow 0$ ▷ Number of zeros to pad at a particular dimension
  if *S* **mod** *patch_size* **equal** 0 **then** ▷ When the patches accurately cover the image
    $num\_patch \leftarrow S$ **divide** $patch\_size$
    $stride \leftarrow patch\_size$
  else ▷ Allow padding and overlapping for one more patch
    $num\_patch \leftarrow (S$ **divide** $patch\_size)$ **plus** 1
    $stride \leftarrow (S$ **divide** $num\_patch)$ **plus** 1
    $pad \leftarrow (stride$ **multiply** $(num\_patch$ **minus** $1))$ **plus** $patch\_size$ **minus** *S*
  return num_patch, $stride$, pad

SACAM-SegNet Segmentation Images. More segmentation images generated by the SACAM-SegNet refigned by the Conv-CRF (Teichmann and Cipolla, 2019) layer are shown in Figure 5. The second column images are the ground truth segments in the LMD (Schwartz and Nishino, 2016; Schwartz and Nishino, 2020), and the third column images are manually labelled dense segments.

BCAM-SegNet Segmentation Images. More segmentation images generated by the three BCAM-SegNet student models trained with the self-training approach are shown in Figure 6. Our BCAM-SegNet managed to refine the material boundaries for some images, such as the window in the first image, and the ceramic close-stool in the sixth image.

Figure 5: Dense material segmentation results for the SACAM-SegNet, refined by the Conv-CRF layer.

Figure 6: Dense material segmentation results generated by BCAM-SegNet, refined with the Conv-CRF layer. The self-training approach is repeated three times to train Student 1, 2, and 3.