2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

HS3-Bench: A Benchmark and Strong Baseline for Hyperspectral Semantic Segmentation in Driving Scenarios

Nick Theisen, Robin Bartsch, Dietrich Paulus and Peer Neubert

***Abstract*— Semantic segmentation is an essential step for many vision applications in order to understand a scene and the objects within. Recent progress in hyperspectral imaging technology enables the application in driving scenarios and the hope is that the devices perceptive abilities provide an advantage over RGB-cameras. Even though some datasets exist, there is no standard benchmark available to systematically measure progress on this task and evaluate the benefit of hyperspectral data. In this paper, we work towards closing this gap by providing the HyperSpectral Semantic Segmentation benchmark (HS3-Bench). It combines annotated hyperspectral images from three driving scenario datasets and provides standardized metrics, implementations, and evaluation protocols. We use the benchmark to derive two strong baseline models that surpass the previous state-of-the-art performances with and without pre-training on the individual datasets. Further, our results indicate that the existing learning-based methods benefit more from leveraging additional RGB training data than from leveraging the additional hyperspectral channels. This poses important questions for future research on hyperspectral imaging for semantic segmentation in driving scenarios. Code to run the benchmark and the strong baseline approaches are available under `https://github.com/nickstheisen/hyperseg`.**

I. INTRODUCTION

Semantic segmentation models assign class labels to pixels and partition the image into regions with a problem-dependent semantic meaning. This is an important step in many vision applications such as scene understanding or the identification of image areas with certain task relevant properties. Hyperspectral imaging (HSI) systems capture light in up to hundreds of very narrow spectral bands, often including ranges of the electromagnetic spectrum that are invisible to classical RGB-cameras and the human eye. These advantages were exploited in the past to solve problems in many different domains, e. g. remote sensing [1], medicine [2] and agriculture [3]. The application in dynamic scenes is difficult, because many HSI-sensors rely on scanning along the spectral dimension or along the spatial dimensions to capture a full hyperspectral cube, which takes time. However, through steady improvement of imaging systems, sensors became smaller, cheaper and simpler in usage. The latter is especially true for snapshot hyperspectral cameras which do not rely on scanning techniques but instead capture a full hyperspectral cube in an instant. This progress enables the adoption of HSI-sensors in novel applications and domains, e. g. dynamic driving scenarios, thus increasing the need for well-generalizing models for tasks such as semantic segmentation. However, the question whether HSI-sensors provide a significant advantage in dynamic driving scenarios remains open.

All authors are with the Institute of Computational Visualistics, University of Koblenz, Germany. Correspondence Email: `nicktheisen@uni-koblenz.de`

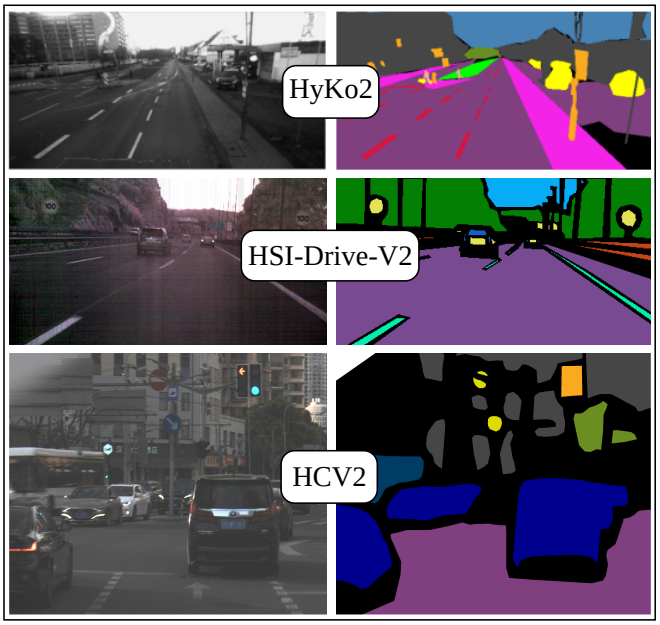

Fig. 1. Examples scenes from the hyperspectral image datasets used in HS3-Bench (HyKo2[4], HSI-Drive[5], and HCV2 [6]) together with ground-truth semantic segmentation labels.

The lack of a standardized benchmark makes the comparison of different approaches very challenging. In the literature only evaluation results on individual datasets without a common evaluation protocol are published. To address this problem, we present HS3-Bench, a hyperspectral semantic segmentation benchmark with the focus on driving scenarios. Example images can be seen in Fig. 1. The benchmark includes three datasets and allows systematic comparison of different approaches.

Our contributions can be summarized as follows:

- We introduce HS3-Bench a hyperspectral semantic segmentation benchmark focused on driving scenarios systematic evaluation.
- We propose two strong baselines for our benchmark, one based on the U-Net architecture, that uses only the HSI data from the benchmark datasets and one based on DeeplabV3+ (DL3+) that leverages additional data through pre-trained model weights. Our baselines outperform current state-of-the-art models.
- We provide evidence that the existing learning based-methods benefit more from leveraging additional RGB

training data than from leveraging the additional HSI channels. This poses important questions for future HSI research.

This paper is organized as follows. Section II discusses the availability of hyperspectral benchmarks in the literature. In III we introduce HS3-Bench and present the used datasets, evaluation metrics as well as benchmark guidelines. Section IV describes the strong baselines. Description and results of our experiments can be found in section V, followed by a summary of our findings in section VI.

II. Related Work

Deep learning and particularly Convolutional Neural Network (CNN)-based architectures establish the state of the art in semantic segmentation. There exist multiple datasets with well-defined benchmarks for *RGB-images* in driving scenarios, e. g. Cityscapes Pixel-Level Semantic Labeling Task [7] or KITTI Semantic Segmentation benchmark [8]. However, the combination of deep learning and *HSI* suffers from the limited amount of available data. Typically, hyperspectral datasets are small and therefore only very limited data is available for training. The most common datasets are remote sensing datasets consisting only of a single image, which can be considered solved, e. g. [1]. The train-test split is created by splitting the image into pixels (or patches) which restricts the applicable models to pixel classification models. In recent years, some larger, multi-image datasets showing urban and driving scenarios have been published (HyKo2 [4], HSI-Drive v1 [5] and v2 [9], HCV2 [6], HSI-Road [10]). They allow the application of encoder-decoder models that predict pixel-precise classification maps for whole images during inference, such as U-Net [11] or DL3+ [12]. Unfortunately, for HyKo2 a well-defined benchmark does not exist and HSI-Road consists of only two classes – road and not road – from which only one is considered during evaluation, making it closer to segmentation than semantic segmentation. Results for HSI-Drive were published in [13] and [9] using a fully-convolutional network but in their experiments the authors use at most six different classes, as they combine certain combinations of minority classes into the class 'other'. On HCV2 the authors of [14] achieved good results with a dual stream model using HSI as well as synthesized RGB. This allowed them to use a ResNet50 [15] backbone pre-trained on ImageNet, which led to a significant improvement of the model performance.

III. Benchmark: HS3-Bench

This section describes HS3-Bench, the HyperSpectral Semantic Segmentation Benchmark for driving scenarios. Detailed results and implementations are available in the benchmark repository.

A. Datasets

For our benchmark we built upon the three existing datasets HyKo2 [4], HSI-Drive [5], and HCV2 [6]. Example images can be seen in Fig. 1. An overview over the datasets is given in Table I. We do not include the HSI-Road [10] dataset since it only uses two classes (cf. section II). All datasets show a high class imbalance, as apparent in many semantic segmentation datasets.

TABLE I
Overview of the HSI datasets used in HS3-Bench

Name	HyKo2-VIS	HCV2	HSI-Drive
Image size	254×510	1400×1800	409×216
Bands	15	128	25
Range (nm)	470-630	450-950	600-975
Images	371	1330	752
Classes	10	19	9
Train/Test/Val-split (%)	50/20/30	72/8/20	60/20/20

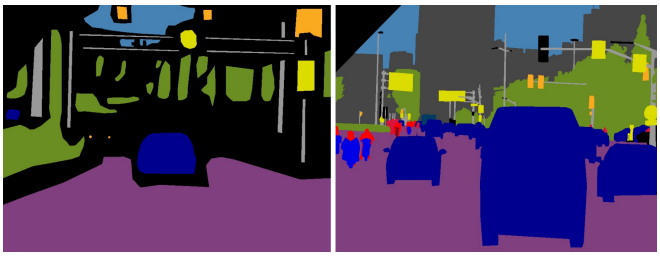

Fig. 2. Coarse labels from HCV2 train set (left) and fine labels from test set (right).

HyKo2-VIS consists of images showing urban and rural driving scenarios segmented into 10 classes with classes "road" (35.8%), "sky" (15.2%), "grass" (14.7%) and "vegetation" (14.1%) being most apparent, and "lane markers" (1.1%), "panels" (1.5%) and "person" (.03%) being least apparent. As the name suggests the images are captured in the visual light spectrum. We omitted the near-infrared data set of HyKo2 because of the unfavorable sensor setup, which resulted in most images only showing a small patch of street in front of the vehicle.

HCV2 was published in the context of the 2021 physics based learning ICCV workshop and shows urban scenarios. It has the highest spectral and spatial resolution and spectral range of the available datasets, covering the visual and near-infrared spectrum. With 19 classes it also has the highest number of distinct class labels. However, eight of the available classes make up less than 1% of all labeled data and there exist no samples for the class "pole". The train set includes only coarse labels, while the test set includes fine labels, i. e. objects are very precisely labeled as shown in Fig. 2.

HSI-Drive contains 10 classes with "road" (60.7%) and "vegetation" (21.3%) being most apparent. All other classes are below 6%, the authors even propose to not use the class "water" (.03%) because of the low amount of samples available. We follow this recommendation in our benchmark and ignore pixels with this class. When we mention HSI-Drive in this paper we always refer to v2 if not stated otherwise, as it includes v1.

HCV2 comes with a defined train-test split. For the others we calculate dataset splits using pytorch's *random_split*-method to sample train-val-test-splits with proportions as shown in Table I. The sample lists for each split and each dataset will be published with the code.

B. Metrics

The benchmark implements the commonly used metrics Accuracy (Acc), F1-Score (F_1) as defined in [16], and Jaccard Index (J) as defined in [17].

Micro- and macro-averaging [16]: Accuracy is calculated in micro- and macro-averaged form, denoted with μ and M in subscript, respectively. All other metrics are only calculated in macro averaged form. With micro averaging, the metric is calculated by averaging over all sample predictions (per pixel), while in macro averaging the metric is calculated per class and the average is then calculated over all class-wise scores. As we are dealing with highly class imbalanced datasets, our evaluation is mostly focused on macro averaged metrics, as micro averaged metrics can be very much influenced by a data set's majority class. For example in an extreme case of a binary classification problem where 99% of pixels belong to class A a classifier that always predicts A will have a micro-averaged accuracy of 99% and a macro-averaged accuracy of $(100\% + 0\%)/2 = 50\%$. The former would in many cases be seen as overoptimistic, while the latter may be seen as overly pessimistic. Which value carries more weight depends on the problem at hand. Therefore, we provide both values for Accuracy.

Summary statistics: In order to capture the performance across multiple datasets in a single number, the benchmark reports the average performance over all datasets $d \in \mathcal{D}$ and metric scores $S = \{\mathrm{Acc}, \mathrm{F}_1, \mathrm{J}\}$. To identify potential generalization problems, the benchmark also reports the worst case performance across all dataset (i.e. the minimum value for a metric across all datasets).

$$S_{avg} = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} S_d, \qquad S_{wc} = \min_{d \in \mathcal{D}} S_d \tag{1}$$

These summary statistics will become even more important when more datasets become available for inclusion in the benchmark.

C. Benchmark guidelines

The provided benchmark implementation is intended to facilitate the creation of comparable results across different research groups. There are a couple of benchmark guidelines that have to be followed by each user:

1) *All* datasets and *all* metrics should be evaluated to avoid suspicion of cherry-picking.
2) The intention is to evaluate a single approach on multiple datasets. Therefore, the *same* algorithm or the *same* model architecture should be used for *all* datasets.
3) The model can be trained individually for each dataset.
4) If different hyperparameters are used for different datasets, this should be stated explicitly.
5) Only train and validation data splits can be used for training and hyperparameter tuning. The test data should only be used once for the final evaluation.
6) If additional datasets are used for training or parameter tuning or pre-trained models are used, this should be stated explicitly.
7) We strongly encourage benchmark users to also provide information about the computational effort (runtime, memory, used hardware) and to share implementations and trained models.

IV. Strong baselines

We present two strong baselines. One based on a small and well regularized model that uses only the training data provided with the benchmark HSI dataset. And a second, larger model that is well suited to leverage additional data through initialization with pre-trained weights.

A. Models

Our baseline approaches build upon two different model architectures, U-Net [11] and DL3+ [12]. These where chosen as they were suitable for our experiments and because they represent well-established architectures for the given task. The U-Net model implements an Encoder-Decoder-Architecture. In the encoder, the model first compresses data to a discriminative low-resolution feature map and then recovers the original resolution in the decoder, guided by skip-connections in intermediate layers. In contrast to the original U-Net model we (1) use bilinear upsampling instead of transposed convolutions in the decoder to avoid checkerboard artifacts [18], (2) adapt the channel size of the input layer to allow the training of images with an arbitrary feature dimension, and (3) apply a combination of regularization techniques to account for small training set sizes. The following subsection IV-B will demonstrate how the HS3-Bench can be used for hyperparameter tuning of the resulting regularized U-Net (RU-Net).

The DL3+ model extends the DeeplabV3 [19] model. Both models use an atrous spatial pyramid pooling (ASPP) module to extract rich semantic features at multiple image resolutions without poooling [12]. To combine this property with the advantage of Encoder-Decoder-based architectures, DL3+ extends DeeplabV3 with a decoder module that guides the model in recovering the original image resolution with intermediate feature maps, allowing it to better recover small objects and object boundaries compared to its predecessor. For initial input processing we use a MobileNetV2 [20] backbone network. To train the model with images of arbitrary feature dimension, we introduce a 1×1 convolutional layer as input layer that reduces the dimensionality to 3 and makes it compatible with the expected input dimension of the original model for RGB data.

B. Using HS3-Bench for Hyperparameter Tuning

The proposed HS3-Bench can be used to derive model hyperparameters. As a demonstration, we used this for deriving a regularization configuration for RU-Net in order to reduce the risk of overfitting. We considered data augmentation (DA), weight decay (WD), class weighting (CW), batch normalization (BN), and dropout (DO) as potential regularization techniques. We systematically tested different combinations and parameterizations by fitting the model parameters on the HS3-Bench training datasets and

evaluated the trained models on the validation set to receive an estimation of the model performance on unseen data. By monitoring the average performance values we were able to identify hyperparameter settings that generalize well across all datasets.

Following this system, we identified BN as well as DA and DO with respective probabilities of 0.1 and 0.25 as best configuration. Table III shows the resulting improvement on the HS3-Bench test data. A detailed discussion of the results from this table will follow in section V. As an example, compared to regular U-Net the average Jaccard score improves by $+4.32\%$

C. Synthesizing (Pseudo-)RGB Images

It is common ground that in many deep learning applications, access to more data can provide substantial improvements. In driving scenarios, there are large amounts of RGB training images and pre-trained networks available. For example, DL3+ uses a MobileNetV2 backbone network for which model weights pre-trained on ImageNet[1] are publicly available. To benefit from this additional information we can synthesize RGB images from hyperspectral images.

We follow the simple approach of Ding et al. [14], by manually selecting three bands from all spectral bands in the HSIs that resemble the red, green, and blue channels. We used channels $(63, 19, 1)$, $(14, 7, 0)$ and $(2, 1, 0)$ for HCV2, HyKo2 and HSI-Drive respectively. We scale all pixels $\mathbf{p} \in \mathbb{R}^3$ in the synthesized image according to the channel-specific min- and max-values $\mathbf{p}_{min}$, $\mathbf{p}_{max} \in \mathbb{R}^3$ derived from the whole dataset. Hence, for each image pixel $\mathbf{p}$ and all images in the dataset we calculate the new pixel values $\mathbf{p}'$

$$\mathbf{p}' = (\mathbf{p} - \mathbf{p}_{min})/(\mathbf{p}_{max} - \mathbf{p}_{min}) \tag{2}$$

As HSI-Drive only covers the red and near-infrared spectrum, which does not allow the estimation of RGB images, we will refer to the synthesized images as pseudo-RGB (pRGB) images from now on. Example images are shown in Fig. 3. Note that this drastically reduces the data volume and feature dimensionality.

As a special property, HCV2 also provides images from an RGB camera together with the HSI data. The RGB images are synchronized and cropped to the same resolution as their HSI counterparts. The images were exactly registered such that pixels at the same image coordinates refer to the same object in world coordinates. However, in general such data is not available for HSI datasets. When we use this data for comparison, we will apply the same normalization strategy from eq. 2.

V. Experiments & Results

In this section, we will use HS3-Bench to evaluate the performance of different algorithms and input data configurations for semantic segmentation on HSI data. In section V-A, we first train our baseline models on the HSI data and also evaluate the influence of dimensionality reduction on model performance. In section V-B we compare the performance of models using (pseudo-)RGB images to those using full spectrum HSI. Then, we quantify the performance improvement from pre-training on additional data in combination with (pseudo-)RGB data in section V-C, followed by qualitative assessment of prediction results and a discussion of the improvement of the HS3-baseline models over the state of the art in sections V-D and V-E.

[1]The pre-trained model weights provided by Pytorch were used: `https://pytorch.org/vision/stable/models.html`

Fig. 3. Examples of pRGB images synthesized from HSI images for HyKo2 (top left), HCV2 (top right), HSI-Drive (bottom left) and RGB images supplied together with HCV2 (bottom right).

TABLE II

Fixed training parameters of all training runs for each dataset, unless otherwise specified for individual runs.

Training Parameter	HyKo2-VIS	HCV2	HSI-Drive
optimizer	AdamW	AdamW	AdamW
learning rate	10^{-3}	10^{-3}	10^{-3}
optimizer epsilon	10^{-8}	10^{-4}	10^{-8}
batch size	16	4	32
max epochs	500	100	300
early stopping	✓	✓	✓
loss	cross-entropy	cross-entropy	cross-entropy

Unless stated otherwise, we kept all hyperparameters constant. The parameters are shown in Table II. We used cross-entropy loss for all of our experiments. All experiments were performed on a single Nvidia-A100 GPU with 40GB VRAM. Table III provides the main results from this paper. The upper part provides results for the individual datasets, the lower part provides summary statistics across all datasets.

A. Applying HS3-Bench for Comparison of Full-Spectrum HSI Data and Reduction to a Single Channel

We applied the benchmark baseline models U-Net, RU-Net and DL3+ from Sec. IV on different input data. For example, the entry HSI in the column Data indicates that all spectral bands are used. As mentioned before, the results in Table III suggest that RU-Net generally performs better than U-Net. Notably, this smaller model also performs better than the larger DL3+ model in this comparison where no additional data is available.

Training a model with all spectral bands available in the benchmark datasets defines one extreme, another extreme is reducing the spectral information to a single feature channel with principal component analysis (PCA), denoted as PCA1 in Table III. Surprisingly, as also depicted in Fig. 4, the

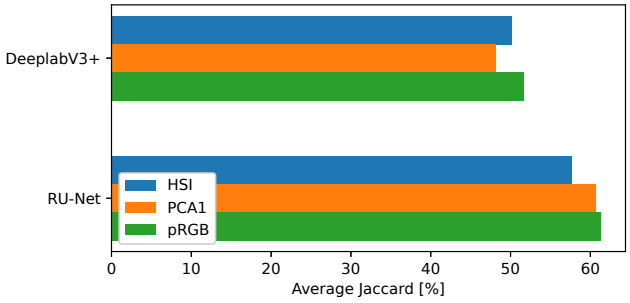

Fig. 4. Average Jaccard score per model and data type.

average performance of DL3+ with PCA1 data compared to HSI data is only slightly worse. Further, the average performance of RU-Net with PCA1 is even better than with HSI. Looking at the individual datasets, we see that on HCV2 both architectures benefit from PCA1. On HSI-Drive and HyKo2, RU-Net profits significantly on the former and has no significant effect on the latter, while the performance of DL3+ is decreased in both cases. We suspect that because HyKo2 has only 16 spectral channels, much fewer than the 128 spectral channels of HCV2, models trained on the former are less affected by the curse of dimensionality, while models trained on the latter are not presented with enough data to produce a robust classifier.

B. Comparison of HSI and (Pseudo-)RGB Data

In this section we quantify the discrepancy in performance of models trained on HSI data and on pRGB data. We synthesized pRGB images from HSI data, as described in section IV, and used the same dataset splits to keep the class distribution constant in our experiments.

We repeat the experiments from section V-A with the synthesized pRGB images. Note that all information in the pRGB images was derived from HSI. We trained our baseline models from scratch and did not use pre-training. The results are also presented in Table III. Fig. 4 illustrates that the average jaccard score of RU-Net with pRGB data improves by $+3.66\%$ and for DL3+ by $+1.54\%$ over the performance of the same model trained on HSI. The improvement can mainly be traced back to HyKo2 and HCV2, HSI-Drive results improve only slightly. HSI-Drive covers only red and near-infrared channels so we could not select bands from the spectral intervals corresponding to RGB-wavelengths. Therefore, the results might be explained by suboptimal band selection during RGB image synthesis. In summary using pRGB data leads to an overall improvement in model performance.

C. Impact of Pretraining on Model Performance

To investigate the potential benefit of pre-training on RGB data we compare the performance of the DL3+ model on pRGB data without pre-training and with pre-training. In our first test, we intialize the backbone networks of DL3+ with pre-trained weights (cf. IV) and fine-tune the full model on our data. Then, to see how well information from similar domains can be transferred we initialize *all* of our model parameters with model weights pre-trained on Cityscapes[3] [7]. We only replaced the output layer, such that the number of predicted classes matched the number of classes of each dataset in HS3-Bench. In our second test *all* model parameters were frozen, except the ones in the output layer to avoid adapting the models feature extraction to the new dataset.

The result of using pre-trained weights and fine-tuning the full model are summarized in Table III denoted with BB. The average performance increased significantly, by around $+7\%$ compared to the best model that does not use pre-trained weights. Fig. 5 shows that the performance improvement compared to pRGB images without pre-training is most

TABLE III

BENCHMARK SCORES (%) ON THE HS3-BENCH TEST DATA.

			Testing			
Dataset	Approach	Data	Acc_μ	Acc_M	F1_M	J_M
HCV2	U-Net	HSI	85.25	48.62	48.18	37.73
	RU-Net	HSI	87.63	54.14	53.26	42.23
	RU-Net	PCA1	88.25	58.07	55.43	44.26
	RU-Net	pRGB	87.95	56.65	55.46	44.03
	DL3+	HSI	86.60	53.15	51.83	40.79
	DL3+	PCA1	86.64	54.46	52.90	41.58
	DL3+	pRGB	87.00	55.33	54.08	42.58
	DL3+(BB)	pRGB	**90.26**	**64.10**	**61.93**	**50.04**
	(DL3+(BB))[2]	(RGB)	(91.22)	(65.87)	(63.33)	(52.11)
	DL3+(PT)	pRGB	89.62	61.91	60.17	48.47
HyKo2	U-Net	HSI	85.36	68.15	68.55	57.39
	RU-Net	HSI	86.72	68.79	69.19	58.64
	RU-Net	PCA1	85.61	68.09	70.01	58.67
	RU-Net	pRGB	89.18	73.92	75.04	64.67
	DL3+	HSI	84.10	63.01	64.90	53.22
	DL3+	PCA1	79.99	61.59	63.00	50.40
	DL3+	pRGB	84.64	65.30	66.56	54.82
	DL3+(BB)	pRGB	**90.49**	**74.87**	**77.11**	**66.77**
	DL3+(PT)	pRGB	88.62	73.97	76.79	65.41
HSI-Drive	U-Net	HSI	94.95	74.74	76.08	64.95
	RU-Net	HSI	96.08	79.82	82.34	72.18
	RU-Net	PCA1	97.02	**86.80**	**87.76**	**79.23**
	RU-Net	pRGB	96.32	82.70	84.91	75.31
	DL3+	HSI	92.51	65.58	67.86	56.63
	DL3+	PCA1	90.88	62.93	64.31	52.62
	DL3+	pRGB	92.74	66.59	69.46	57.84
	DL3+(BB)	pRGB	**97.09**	83.93	86.41	77.44
	DL3+(PT)	pRGB	95.69	81.95	84.09	73.84
Average Perf.	U-Net	HSI	88.52	63.84	64.27	53.36
	RU-Net	HSI	90.14	67.58	68.26	57.68
	RU-Net	PCA1	90.29	70.99	71.07	60.72
	RU-Net	pRGB	91.15	71.09	71.80	61.34
	DL3+	HSI	87.74	60.58	61.53	50.21
	DL3+	PCA1	85.84	59.66	60.07	48.20
	DL3+	pRGB	88.13	62.41	63.37	51.75
	DL3+(BB)	pRGB	**92.61**	**74.30**	**75.15**	**64.75**
	DL3+(PT)	pRGB	91.31	72.61	73.68	62.57
Worst-Case Perf.	U-Net	HSI	82.25	48.63	48.18	37.73
	RU-Net	HSI	86.72	54.14	53.26	42.23
	RU-Net	PCA1	85.61	58.07	55.43	44.26
	RU-Net	pRGB	87.95	56.65	55.46	44.03
	DL3+	HSI	84.10	53.15	51.83	40.79
	DL3+	PCA1	79.99	54.46	52.90	41.58
	DL3+	pRGB	84.64	55.33	54.08	42.58
	DL3+BB	pRGB	**90.26**	**64.10**	**61.93**	**50.04**
	DL3+PT	pRGB	88.62	61.91	60.17	48.47

[2]The result on RGB data is surrounded in parenthesis as the data was collected with an additional sensor (cf. Sec. V-B). RGB data was only provided for HCV2 and therefore we could not calculate summary statistics.

[3]The model weights were downloaded from this repository: `https://github.com/VainF/DeepLabV3Plus-Pytorch`

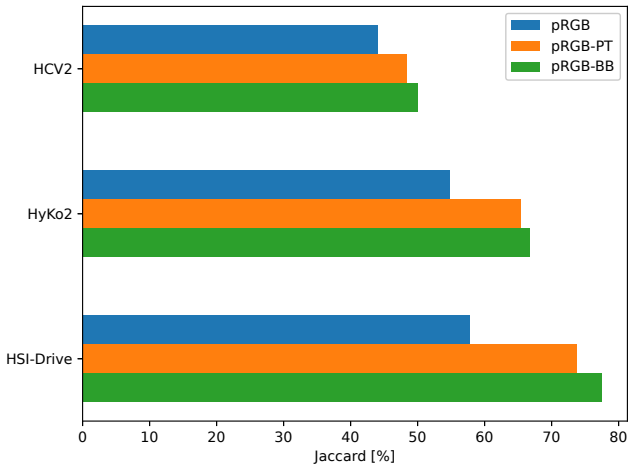

Fig. 5. Jaccard scores for DL3+ on pRGB data per dataset without pre-training (pRGB), with pre-trained backbone and fine-tuning (pRGB-BB) and with pre-trained weights transferred from a similar domain, namely CityScapes (pRGB-PT). In the latter, all weights except for output-layer were frozen.

apparent in HSI-Drive. This is especially interesting, as HSI-Drive contains only spectral information from the red and the near-infrared spectrum. Hence, the synthesized images do not have the same distribution typically apparent in RGB images. This observeration indicates that the features extracted by the backbone model are general enough to be successfully applied to neighboring spectral domains.

In our second experiment on pre-training, fine-tuning only the output layer showed only slightly lower average performance than fine-tuning the full model. Nonetheless, the performance is still significantly better than all approaches that do not use pre-training, which shows that the feature extraction modules can be directly applied to similar domains. When only considering pRGB data all results on each individual dataset improve with pre-training by a large margin, as depicted in Fig. 5. The strong improvements in model performance indicate that exploiting knowledge through pre-training is very effective. It has an even stronger relative effect than using pRGB data instead of full-spectrum HSI. Further, the observation that models using pRGB data outperformed all models using full spectra, support that for driving scenarios introducing knowledge from related domains is more beneficial than adding additional spectral features for the available datasets.

Note that we used a very simple way of synthesizing pRGB images (see Sec. IV-C). The spectral bands are very narrow - especially in HCV2 - which leads to lower signal intensities and in turn to noisy bands. The synthesized images show an unnatural color distribution (see Fig. 3), distinct from typical RGB-images. To estimate the upper limit of model performance that can be expected with better RGB-image synthesis, we trained an additional model on the RGB images provided with HCV2 (cf. section V-B). We fine-tuned DL3+ using hyperparameter settings from Table II and a pre-trained MobileNetV2 backbone and achieved a Jaccard score of 52.11% ($+0.43\%$ as compared to the current state-of-the-art results that were published in [14]). Hence, we expect that with more sophisticated RGB image synthesis methods the results on all data sets are likely to improve.

D. Qualitative Evaluation

To give a visual impression of the models segmentation performance, Fig. 6 shows example inferences for each data set in HS3-Bench side by side. The top row shows ground-truth label maps, followed by inferences on HSI data, then pRGB and finally pRGB with pre-trained backbone. For inference we applied the model that showed the best average performance for the given data type or pre-training configuration consistently to all datasets. The best models are RU-Net for HSI and pRGB without pre-training (row 2 and 3) and fine-tuned DL3+ with pre-trained backbone network (row 4). The example predictions support the impression of the statistical results. The predictions on pRGB data are less noisy than HSI and contours are more precise. The difference between row 3 and row 4 are subtle. It seems that object contours are a bit more precise for the pre-trained model and object surfaces are more homogeneous.

E. Comparison to the State of the Art

HCV2 was introduced in the context of a challenge for a workshop at ICCV 2021. The best reported results in the competition achieved a Jaccard score of 51.4%. In [14] this result was raised to 51.76% by HRNet [21] that was fine-tuned on the RGB-images provided with HCV2. Our best pre-trained model on these RGB-images achieved a Jaccard score of 52.11% ($+0.43\%$ increase compared to the current state-of-the-art).

Under the conditions, that no pre-training and only HSI data or data derived from HSI is used, the best listed model (FCN101 [22]) in [14] achieved a Jaccard score of 41.13%. With RU-Net and HSI data we improved this result to 42.23% ($+1.1\%$) and with data derived from HSI, i. e. PCA1, the same model further improves to 44.26% ($+3.13\%$).

VI. Summary

In this paper we presented HS3-Bench, a hyperspectral semantic segmentation benchmark for driving scenarios which is designed for systematic comparison of different models and algorithms. Based on this benchmark, we performed systematic evaluation of hyperspectral image representations, i. e. full spectrum, PCA-reduced spectrum and synthesized pseudo-RGB images as well as the impact of knowledge transfer through pre-trained weights. We demonstrated the application of HS3-Bench by deriving a suitable configuration of regularization approaches to a U-Net model (RU-Net). In our experiments we used RU-Net as well as DeeplabV3+ (DL3+) with a MobileNetV2 backbone.

We consider both models - RU-Net and DL3+ - as strong baselines for HS3-Bench. Under the condition that only limited hyperspectral data is available the regularized U-Net with dimensionality reduction outperforms DL3+ as well as the current state of the art model. However, if additional RGB data is available in the problem domain, DL3+ with pRGB

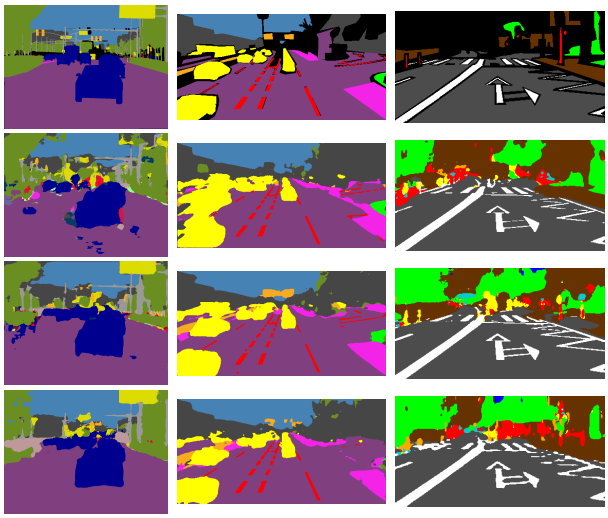

Fig. 6. Example inferences for HCV2, HyKo2 and HSI-Drive. The top row shows the ground-truth label map. Row 2 and 3 show inferences on HSI and pRGB data without pre-training, respectively. The bottom row shows inferences with pRGB data with pre-training. We consistently used the models that showed best performance for the given data type, i. e. RU-Net(row 2 and 3) and DL3+ with pretrained backbone (row 4).

images synthesized from HSI data can effectively leverage the domain knowledge through pre-training and should be preferred. DL3+ with a pre-trained backbone network fine-tuned on RGB data outperforms the previous state-of-the-art models using pre-trained weights as well.

Our results pose interesting questions for future research. In [14] the authors state that their dual fusion network effectively utilizes knowledge from pre-trained RGB models and hyperspectral data. However, our results suggest that major improvements can be traced back to leveraging domain knowledge through pre-trained model parameters. Further, our experiments support that available learning-based models benefit more from leveraging additional RGB training data than from leveraging additional HSI channels. We believe the proposed HS3-Bench can be a valuable tool to support research directions such as finding general backbone models for HSI data and models that better exploit all channel information in HSI data. Also, further investigation is required to identify the causes of the performance discrepancy between HSI and RGB.

REFERENCES

[1] Y. Chen, P. Liu, J. Zhao, K. Huang, and Q. Yan, "Shallow-Guided Transformer for Semantic Segmentation of Hyperspectral Remote Sensing Imagery," vol. 15, no. 13, p. 3366, 2023. [Online]. Available: https://www.mdpi.com/2072-4292/15/13/3366

[2] M. A. Calin, S. V. Parasca, D. Savastru, and D. Manea, "Hyperspectral imaging in the medical field: Present and future," *Applied Spectroscopy Reviews*, vol. 49, no. 6, pp. 435–447, 2014. [Online]. Available: https://doi.org/10.1080/05704928.2013.838678

[3] B. Lu, P. D. Dao, J. Liu, Y. He, and J. Shang, "Recent advances of hyperspectral imaging technology and applications in agriculture," *Remote Sensing*, vol. 12, no. 16, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/16/2659

[4] C. Winkens, F. Sattler, V. Adams, and D. Paulus, "Hyko: A spectral dataset for scene understanding," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. Institute of Electrical and Electronics Engineers,, https://www.ieee.org/, 2017, pp. 254–261.

[5] K. Basterretxea, V. Martínez, J. Echanobe, J. Gutiérrez–Zaballa, and I. Del Campo, "Hsi-drive: A dataset for the research of hyperspectral image processing applied to autonomous driving systems," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021, pp. 866–873.

[6] Y. Li, Y. Fu, S. Liang, Y. Zheng, L. Chen, Q. Shen, E. Huang, Y. Huang, Y. Zhuang, Y. Li, D. Zhang, Y. Li, S. You, Y. Zheng, F. Lu, B. Shi, and R. T. Tan, "HyperspectralCityV2.0," 2021, last Accessed: 13.03.2024. [Online]. Available: https://pbdl-ws.github.io/pbdl2021/challenge/download.html

[7] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] H. Alhaija, S. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International Journal of Computer Vision (IJCV)*, 2018.

[9] J. Gutiérrez-Zaballa, K. Basterretxea, J. Echanobe, M. Victoria Martínez, and U. Martinez-Corral, "Hsi-drive v2.0: More data for new challenges in scene understanding for autonomous driving," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2023, pp. 207–214.

[10] J. Lu, H. Liu, Y. Yao, S. Tao, Z. Tang, and J. Lu, "Hsi road: A hyper spectral image dataset for road segmentation," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[12] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 833–851.

[13] J. Gutiérrez-Zaballa, K. Basterretxea, J. Echanobe, M. V. Martínez, U. Martinez-Corral, Ó. Mata-Carballeira, and I. del Campo, "On-chip hyperspectral image segmentation with fully convolutional networks for scene understanding in autonomous driving," *J. Syst. Archit.*, vol. 139, p. 102878, 2023.

[14] X. Ding, S. Gu, and J. Yang, "Dual fusion network for hyperspectral semantic segmentation," in *Image and Graphics*, H. Lu, W. Ouyang, H. Huang, J. Lu, R. Liu, J. Dong, and M. Xu, Eds. Cham: Springer Nature Switzerland, 2023, pp. 149–161.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[16] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[17] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015. [Online]. Available: https://doi.org/10.1007/s11263-014-0733-5

[18] Z. Wojna, V. Ferrari, S. Guadarrama, N. Silberman, L.-C. Chen, A. Fathi, and J. Uijlings, "The devil is in the decoder: Classification, regression and gans," *International Journal of Computer Vision*, vol. 127, pp. 1694–1706, 2019.

[19] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *ArXiv*, vol. abs/1706.05587, 2017.

[20] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:4555207

[21] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3349–3364, oct 2021.

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2015, pp. 3431–3440.