The UAVid Dataset for Video Semantic Segmentation

Ye Lyu[1], George Vosselman[1], Guisong Xia[2], Alper Yilmaz[3], Michael Ying Yang[1*]

Abstract— **Video semantic segmentation has been one of the research focus in computer vision recently. It serves as a perception foundation for many fields such as robotics and autonomous driving. The fast development of semantic segmentation attributes enormously to the large scale datasets, especially for the deep learning related methods. Currently, there already exist several semantic segmentation datasets for complex urban scenes, such as the Cityscapes and CamVid datasets. They have been the standard datasets for comparison among semantic segmentation methods. In this paper, we introduce a new high resolution UAV video semantic segmentation dataset as complement, *UAVid*. Our UAV dataset consists of 30 video sequences capturing high resolution images. In total, 300 images have been densely labelled with 8 classes for urban scene understanding task. Our dataset brings out new challenges. We provide several deep learning baseline methods, among which the proposed novel Multi-Scale-Dilation net performs the best via multi-scale feature extraction. We have also explored the usability of sequence data by leveraging on CRF model in both spatial and temporal domain.**

I. INTRODUCTION

Visual scene understanding has been advancing in recent years, which serves as a perception foundation for many fields such as robotics and autonomous driving. The most effective and successful methods for scene understanding tasks adopt deep learning as their cornerstone, as it can distil high level semantic knowledge from the training data. However, the drawback is that deep learning requires tremendous number of samples for training to make it learn useful knowledge instead of noise, especially for real world applications. Semantic segmentation, as part of scene understanding, is to assign labels for each pixel in the image. To make the best of deep learning method, a large number of densely labelled images are required. At present, there are only several public semantic segmentation datasets available, which focus only on certain applications. MS COCO [1] provides semantic segmentation dataset containing common objects recognition in common scenes, and its semantic labelling task focuses on person, car, animal and different stuffs. Pascal VOC dataset [2] also provides objects like bus, car, cow, dog for semantic segmentation task. Other semantic segmentation datasets are designed for street scene objects recognition. Their target objects include pedestrians, cars, roads, lanes, traffic lights, trees and other street scene related objects. Specially, CamVid [3] provides continuously labelled driving frames, which can be used for temporal consistency evaluation. Highway Driving dataset [4] provides 30Hz labels that are even denser in temporal domain, and it is designed for semantic video segmentation for driving scenes. Daimler Urban Segmentation dataset [5] is also a video dataset for street scene understanding, but its labels are sparser in temporal domain. Cityscapes dataset [6] focuses more on data variation as it is much larger in the number of labelled frames, which are collected from 50 cities, making it closer to real world complexity. Each frame is much larger in size compared with CamVid. The newly published Berkeley Deep Drive dataset [7] has even more image labels with medium image size across multiple street scenes. The KITTI Vision Benchmark Suite [8] also provides images of medium size for the task. To help learning models to generalize well across different scenes, ADE20K dataset [9] contributes as it spans more diverse scenes, and objects from much more different categories are labelled. ADE20K dataset brings more variability and complexity for general object representations in images. For remote sensing community, aerial image dataset is provided for ISPRS 2D semantic labelling contest [10]. All datasets above have had great impacts on the development of current state-of-the-art semantic segmentation methods.

Dynamic scene understanding is another interesting topic. There are several video datasets for moving foreground objects segmentation, such as Video Segmentation Benchmark(VSB100) [11], [12], Freiburg-Berkeley Motion Segmentation dataset(MoSeg) [13], [14] and Densely Annotated VIdeo Segmentation dataset(DAVIS) [15]. In these datasets, foreground objects are labelled densely in both spatial and temporal domain. The challenge for continuous foreground segmentation is that the prediction across highly correlated frames should be consistent. Segmenting foreground objects of interest with consistency is difficult, but useful for surveillance and monitoring.

As present, most of the modern visual semantic segmentation tasks use information acquired on the ground. However, another data acquisition platform is more and more utilized, which is the unmanned aerial vehicle(UAV). Compact and light weighted UAVs are a trend for future data acquisition. The UAVs make image retrieval in large area cheaper and more convenient, which allows quick access to useful information around certain area. Distinguished from collecting images by satellites, UAVs capture images from the sky with flexible flying schedule and higher resolution, bringing the possibility to monitor and analyze landscape at specific location and time swiftly. These abilities make UAVs an effective data collection means for various applications.

The inherently fundamental applications for UAVs are

[1] University of Twente, The Netherlands. {y.lyu, george.vosselman, michael.yang}@utwente.nl
[2] Wuhan University, China. guisong.xia@whu.edu.cn
[3] Ohio State University, USA. yilmaz.15@osu.edu
[*] Corresponding Author.

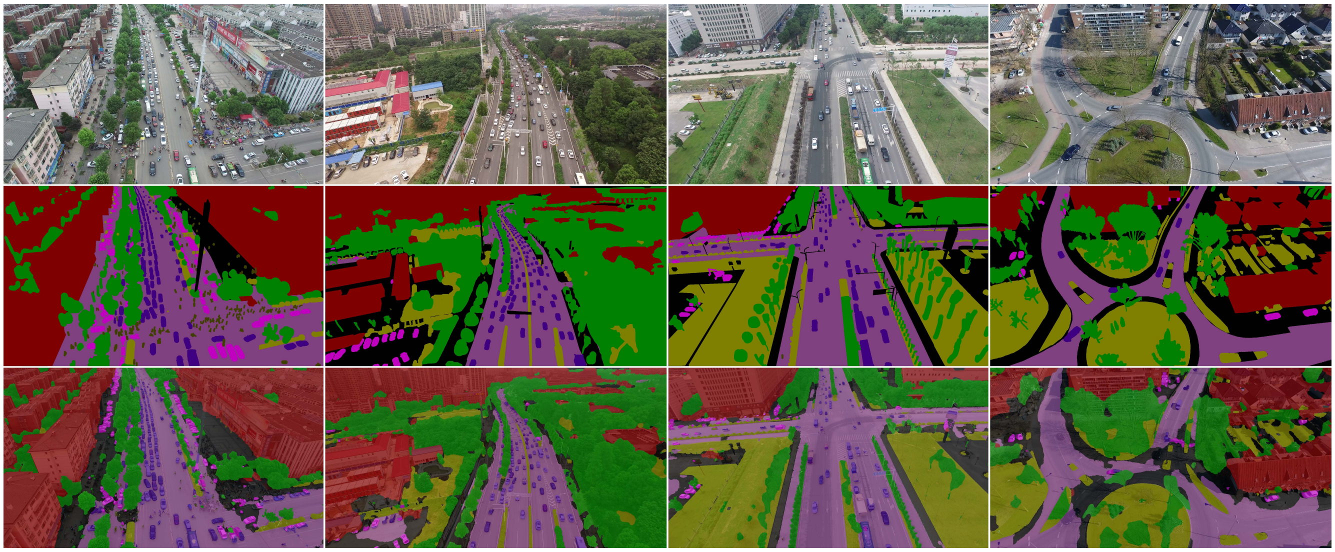


Fig. 1. **Example images and labels from UAVid dataset.** First row shows the images captured by UAV. Second row shows the corresponding ground truth labels. Third row shows the prediction results of MS-Dilation net+PRT+FSO model as in Tab. I.

surveillance [16], [17] and monitoring [18] in the target area. They have already been used for smart farming [19], precision agriculture [20] and weed monitoring [21]. To make the system more intelligent, it could rely on techniques like semantic segmentation and video object segmentation. In this aspect, UAV is a great platform to combine both of the two tasks. These two visual understanding tasks could also be the main foundations for higher level smart applications. As the data from UAVs has its own specialties, semantic segmentation and video object segmentation tasks using UAV data deserve more attentions. There are existing UAV datasets for detection and behaviour analysis [22], but to the best of our knowledge, public datasets for UAV video semantic segmentation do not exist.

In this paper, a new high resolution UAV video semantic segmentation dataset, UAVid, is brought out, which covers semantic segmentation and video object segmentation as a video semantic segmentation task. In total, 300 images from 30 video sequences are densely labelled with 8 object classes. All the labels are acquired with our in-house video labeller tool. To test the usability of our dataset, several typical deep neural networks(DNNs) designed for image semantic segmentation together with CRF based video semantic segmentation methods are evaluated as baselines. In addition, we also show that our novel multi-scale-dilation net model is useful to deal with multi-scale problems for UAV images.

II. DATASET

Designing an UAV video dataset requires careful thought about the data acquisition strategy, UAV flying protocol and object classes selection for annotation. The whole process is designed considering the usefulness and effectiveness for UAV video semantic segmentation research.

A. Data Specification

Our data acquisition and annotation methodology is designed for UAV video semantic segmentation in complex scenes, featuring on both static and moving object recognition. To capture data that contributes the most towards researches on UAV scene understanding, the following features for the dataset are taken into consideration.

- High resolution. We adopt 4K resolution video recording mode with safe flying height of 30 to 50 meters. In this setting, it is visually clear enough to differentiate most of the objects, and objects that are horizontally far away could also be detected. In addition, it is even possible to detect humans that are not too far away.
- Consecutive labelling. Our dataset is designed for video semantic segmentation, it is preferred to label images in sequence, where prediction stability could be evaluated. As it is too expensive to label densely in temporal space, we label 10 images with 5 seconds interval for each sequence.
- Complex and dynamic scenes with diverse objects. Our dataset aims at achieving real world complexity, where there are both static and moving objects. Scenes near streets are chosen for the UAVid dataset as they are complex enough with more dynamic human activities. A variety of objects appear in the scene such as cars, pedestrians, buildings, roads, vegetation, billboards, light poles, traffic lights and so on. We fly UAVs with *slant view* along the streets or across different street blocks to acquire such scenes.
- Data variation. In total, 30 small UAV video sequences are captured in 30 different places to bring variance to the dataset, preventing learning algorithms from overfitting. Data acquisition is done in good weather condition with sufficient illumination. We believe that data acquired in dark environment or other weather

conditions like snowing or raining require special processing techniques, which are not the focus of our dataset.

B. Class Definition and Statistical Analysis

To fully label all types of objects in the street scene in a 4K UAV image is very expensive. As a result, only the most common and representative types of objects are labelled for our current version dataset. In total, 8 classes are deliberately selected for video semantic segmentation, they are building, road, tree, low vegetation, static car, moving car, human and clutter. Example instances from different classes are shown in Fig. 2. We deliberately divide the car class into moving car and static car classes. Moving car is such special class designed for moving object segmentation. Other classes can be inferred from their appearance and context, while moving car class may need additional temporal information in order to be separated properly from static car class. Achieving high accuracy for both static and moving car classes is one possible research goal for our dataset.

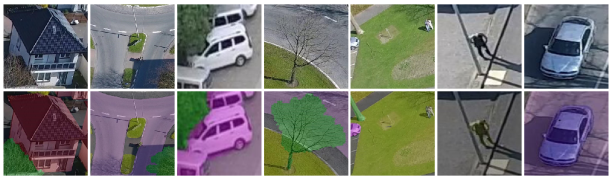


Fig. 2. **Example instances from different classes.** The first row shows the cropped instances. The second row shows the corresponding labels. From left to right, the instances are building, road, static car, tree, low vegetation, human and moving car respectively.

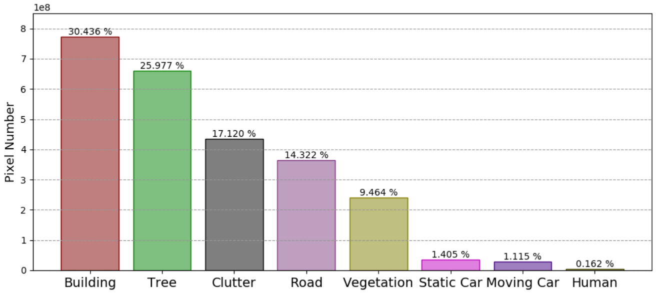


Fig. 3. **Pixel number histogram.**

Number of pixels for each class is reported in Fig. 3. It clearly shows the unbalanced pixel number distribution of different classes. Most of the pixels are from classes like building, tree, clutter, road and low vegetation, and fewer pixels are from moving car and static car classes, which are both fewer than 2% of the total pixels. For human class, it is almost zero, fewer than 0.2% of the total pixels. Smaller pixel number is not necessarily resulted by fewer instances, but the size of each instance. A single building can take more than 10k pixels while a human instance in the image may only take fewer than 100 pixels. Normally, classes with too small pixel numbers are ignored in both training and evaluation for semantic segmentation task [6]. But we believe humans and cars are important classes that should be kept in street scenes rather than being ignored.

C. Annotation Method

We provide densely labelled fine annotations for high resolution UAV images. All the labels are acquired with our own video labeller tool. Pixel level, super-pixel level and polygon level annotation methods are provided for users. For super-pixel annotation, we adopt SLIC method [23] to achieve super-pixel segmentation with 4 different scales, which can be useful for objects with fuzzy boundaries like trees. Polygon annotation is used for regular shape annotation like buildings, while pixel level annotation serves as a basic annotation method. Our tool also provides video play functionality around certain frames to help inspecting whether certain objects are moving or not. As there might be overlapping objects, we label the overlapping pixels to be the class that is closer to the camera.

D. Dataset Splits

The whole 30 densely labelled video sequences are divided into training, validation and test splits. We do not split the data completely randomly, but in a way that makes each split to be representative enough for the variability of different scenes. All three splits should contain all classes. Our data is split at sequence level, and each sequence comes from a different scene place. Following this scheme, we get 15 training sequences(150 labelled images) and 5 validation sequences(50 labelled images) for training and validation splits respectively, whose annotations will be made publicly available. The test split consists of the left 10 sequences(100 labelled images), whose labels are withheld for benchmarking purposes. The ratios among training, validation and test splits are 3:1:2.

III. VIDEO SEMANTIC LABELLING

The task for UAVid dataset is to predict per-pixel semantic labelling for the UAV video sequences. The original video file for each sequence is provided together with the labelled images.

A. Tasks and Metrics

The semantic labelling performance is assessed based on the standard IoU metric [2]. The goal for this task is to achieve as high IoU score as possible. For UAVid dataset, clutter class has a relatively large pixel number ratio and consists of meaningful objects, which is taken as one class for both training and evaluation rather than being ignored.

B. Networks for Baselines

To test the usability of our UAVid dataset for semantic labelling task, we have evaluated the performance of several deep learning models for single image prediction. Although static car and moving car cannot be differentiated by their appearance from only one image, it is still possible to predict based on their context. We start with 3 typical deep fully convolutional neural networks, they are FCN-8s [24], Dilation net [25] and U-Net [26].

FCN-8s [24] has been a good baseline candidate for semantic segmentation. It is a giant model with strong and effective feature extraction ability, but yet simple in structure. It takes a series of simple 3x3 convolutional layers to form the main parts for high level semantic information extraction. This simplicity in structure also makes FCN-8s popular and widely used for semantic segmentation.

Dilation net [25] has similar front end structure with FCN-8s, but it removes last two pooling layers in VGG16. Instead, convolutions in all following layers from conv5 block are dilated by a factor of 2 due to the ablated pooling layers. Dilation net also applies a multi-scale context aggregation module in the end, which expands the receptive field to boost the performance for prediction. The module is achieved by using a series of dilated convolutional layers, whose dilation rate gradually expands as the layer goes deeper.

U-Net [26] is a typical symmetric encoder-decoder network originally designed for segmentation on medical images. The encoder extracts features, which are gradually decoded through the decoder. The features from each convolutional block in the encoder are concatenated to the corresponding convolutional block in the decoder to gradually acquire features of higher and higher resolution for prediction. U-Net is also simple in structure but good at preserving object boundaries.

C. *Multi-Scale-Dilation Net*

For a high resolution image captured by UAV in slant view, size of objects in different horizontal distances can dramatically vary. Such large scale variation in an UAV image can affect the accuracy for prediction. In a network, each output pixel in the final prediction layer has a fixed receptive field, which is formed by pixels in the original image that can affect the final prediction of that output pixel. When the objects are too small, the neural network may learn the noise from the background. When the objects are too big, the model may not acquire enough information to infer the label correctly. This is a long standing notorious problem in computer vision. To reduce such large scale variation effect, a novel multi-scale-dilation net (MS-Dilation net) is proposed in this paper.

One way to expand the receptive field of a network is to use dilated convolution. Dilated convolution can be implemented through different ways, one of which is to leverage on space to batch operation(S2B) and batch to space operation(B2S), which is provided in Tensorflow API. Space to batch operation outputs a copy of the input tensor where values from the height and width dimensions are moved to the batch dimension. Batch to space operation does the inverse. Applying a standard 2D convolution on the image after S2B is the same as a dilated convolution on the original image. A single dilated convolution can be performed as $S2B->convolution->B2S$. This implementation for dilated convolution is efficient when there is a cascade of dilated convolutions, where intermediate S2B and B2S cancel out. For instance, 2 consecutive dilated convolution with the same dilation rate can be performed as $S2B->convolution->convolution->B2S$.

By utilizing space to batch operation and batch to space operation, semantic segmentation can be done in different scales. In total, three streams are created for three scales as shown in Fig. 4. For each stream, a modified FCN-8s is used as the main structure, where the depth for each convolutional block is reduced due to the memory limitation. Here, filter depth is sacrificed for more scales. To reduce detail loss in feature extraction, the pooling layer in the fifth convolutional block is removed to keep a smaller receptive field. Instead, features with larger receptive field from other streams are concatenated to higher resolution features through skip connection in conv7 layers. Note that these skip connections need batch to space operation to retain spatial and batch number alignment. In this way, each stream handles feature extraction in its own scale and features from larger scales are aggregated to boost prediction for higher resolution streams.

Multiple scales may also be achieved by down sampling images directly [27]. However, there are 3 advantages for our multi-scale processing. First, every pixel is assigned to one batch in space to batch operation and all the labelled pixels shall be used for each scale with no waste. Second, there is strict alignment between image and label pairs in each scale as there is no mixture of image pixels or mixture of label pixels. Finally, the concatenated features in the conv7 layer are also strictly aligned.

For each scale, corresponding ground truth labels can also be generated through space to batch operation in the same way as the generation for input images in different streams. With ground truth labels for each scale, deeply supervised training can be done. The losses in three scales are all cross entropy loss. The loss in stream1 is the target loss while the losses in stream2 and stream3 are auxiliary losses, which we call the deep supervision losses. The final loss to be optimized is the weighted mean of the three losses, shown in the equation below. $m1, m2, m3$ are numbers of pixels of an image in each stream. n is batch index and t is pixel index. p is target probability distribution of a pixel, while q is the predicted probability distribution.

$$CE_1 = \frac{1}{m_1} \sum_{t=1}^{m_1} -p_t \log(q_t) \tag{1}$$

$$CE_2 = \frac{1}{4m_2} \sum_{n=1}^{4} \sum_{t=1}^{m_2} -p_t^n \log(q_t^n) \tag{2}$$

$$CE_3 = \frac{1}{16m_3} \sum_{n=1}^{16} \sum_{t=1}^{m_3} -p_t^n \log(q_t^n) \tag{3}$$

$$Loss = \frac{w_1 \times CE_1 + w_2 \times CE_2 + w_3 \times CE_3}{w_1 + w_2 + w_3} \tag{4}$$

It is also interesting to note that every layer becomes a dilated version for stream2 and stream3, especially for pooling layer and transposed convolutional layer, which turn into dilated pooling layer and dilated transposed convolutional layer respectively. Compared to layers in stream1, layers in

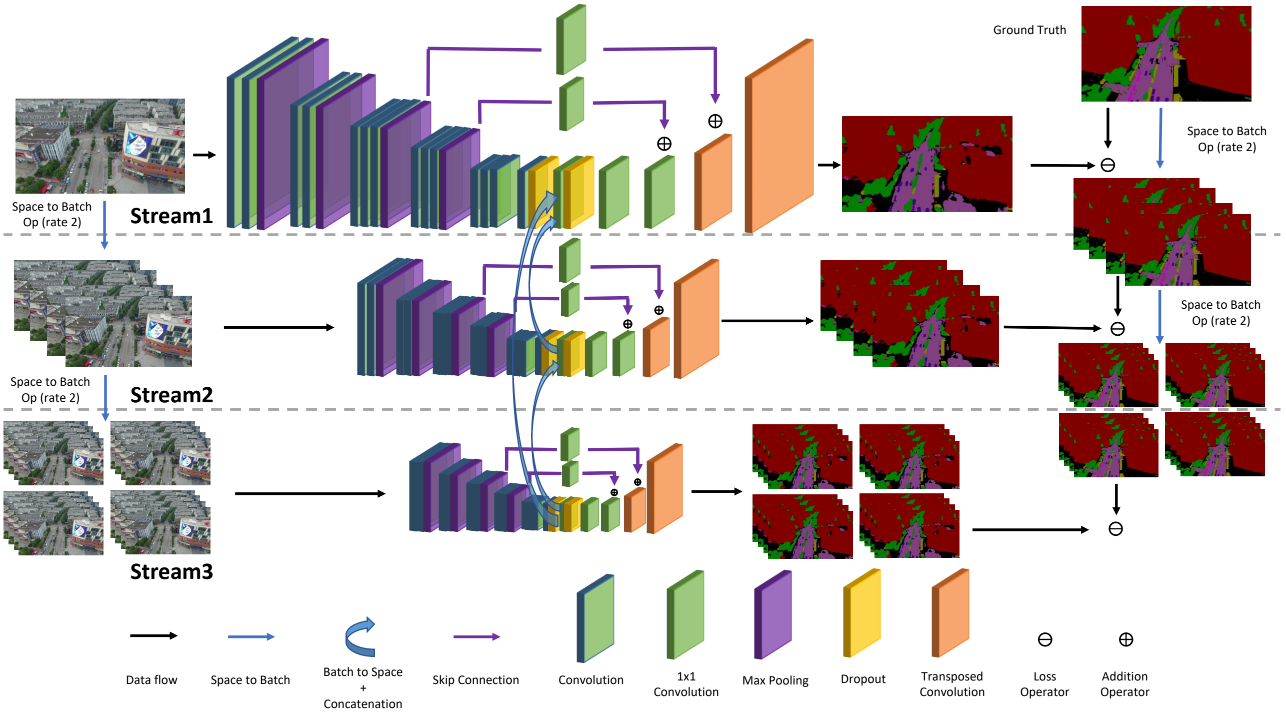


Fig. 4. **Structure of the proposed Multi-Scale-Dilation network.** Three scales of images are achieved by Space to Batch operation with rate 2. Standard convolutions in stream2 and stream3 are equivalent to dilated convolutions in stream1. The main structure for each stream is FCN-8s [24], which could be replaced by any other networks. Features are aggregated at conv7 layer for better prediction on finer scales.

stream2 are dilated by rate of 2 and layers in stream3 are dilated by rate of 4. Theses 3 streams together form the MS-Dilation net.

D. *Fine-tune Pre-trained Networks*

Due to the limited size of our UAVid dataset, training from scratch may not be enough for the networks to learn diverse features for better label prediction. Pre-training a network has been proved to be very useful for various benchmarks [28], [29], [30], [31], which boosts the performance by utilizing more data from other dataset. To reduce the effect of limited training samples, we also explore how much pre-training a network can boost the score for UAVid semantic labelling task. We pre-train all the networks with cityscapes dataset [6], which comprises many more images for training.

E. *Video Semantic Segmentation*

For video semantic labelling task, it is ideal to output prediction consistently for the same objects observed in multiple different images. Taking advantage of temporal information effectively is valuable for video sequence label prediction. Normally, deep neural networks trained on individual images cannot provide completely consistent predictions spanning several frames. However, different frames provide observations from different viewing positions, through which multiple clues can be collected for object prediction. To utilize temporal information in UAVid dataset, we adopt feature space optimization(FSO) [32] method for sequence data prediction. It smooths the final label prediction for the whole sequence by applying 3D CRF covering both spatial and temporal domain. It is the optical flows and tracks in the method that link the images in temporal domain.

IV. EXPERIMENTS

Our experiments are divided into 3 parts. Firstly, we compare semantic segmentation results by training deep neural networks from scratch. These results serve as the basic baselines. Secondly, we analyse how pre-trained models can be useful for UAVid semantic labelling task, and we fine-tune deep neural networks that are pre-trained on cityscapes dataset [6]. Finally, we explore the influence of spatial temporal regulation by using video sequence data for semantic video segmentation.

It should be noted that the resolution of our UAV images is quite high. The size of each image is 4096×2160 or 3840×2160, which requires too much GPU memory for intermediate feature storage in deep neural networks. As a result, we clip each UAV image into 9 evenly distributed smaller overlapped images that cover the whole image for training. Each clipped image is of size 2048×1024. We keep such a moderate image size in order to reduce the ratio between zero padding area and valid image area. Bigger image size also resembles larger batch size if each pixel is taken as a training sample.

A. *Train from Scratch*

To have a fair comparison among different deep neural networks, we re-implement all the networks with Tensorflow [33], and all networks are trained with a Nvidia Titan X GPU. To accommodate the networks into 12G GPU memory,

TABLE I

IoU scores for different models. PRT stands for pre-train and **FSO** stands for feature space optimization [32]. IoU scores are reported in percentage and best results are shown in bold.

Model	Building	Tree	Clutter	Road	Low Vegetation	Static Car	Moving Car	Human	Mean IoU
FCN-8s	64.3	63.8	33.5	57.6	28.1	8.4	29.1	0.0	35.6
Dilation Net	72.8	66.9	38.5	62.4	34.4	1.2	36.8	0.0	39.1
U-Net	70.7	67.2	36.1	61.9	32.8	11.2	**47.5**	0.0	40.9
MS-Dilation Net(ours)	**74.3**	**68.1**	**40.3**	**63.5**	**35.5**	**11.9**	42.6	0.0	**42.0**
FCN-8s+PRT	77.4	72.7	44.0	63.8	45.0	19.1	49.5	0.6	46.5
Dilation Net+PRT	**79.8**	73.6	44.5	64.4	44.6	24.1	53.6	0.0	48.1
U-Net+PRT	77.5	73.3	44.8	64.2	42.3	**25.8**	**57.8**	0.0	48.2
MS-Dilation Net(ours)+PRT	79.7	**74.6**	**44.9**	**65.9**	**46.1**	21.8	57.2	**8.0**	**49.8**
FCN-8s+PRT+FSO	78.6	73.3	45.3	64.7	46.0	19.7	49.8	0.1	47.2
Dilation Net+PRT+FSO	80.7	74.0	45.4	65.1	45.5	24.5	53.6	0.0	48.6
U-Net+PRT+FSO	79.0	73.8	**46.4**	65.3	43.5	**26.8**	56.6	0.0	48.9
MS-Dilation Net(ours)+PRT+FSO	**80.9**	**75.5**	46.3	**66.7**	**47.9**	22.3	**56.9**	**4.2**	**50.1**

TABLE II

IoU scores for different deep supervision plans. W stands for with and **W/O** stands for without.

Method	Building	Tree	Clutter	Road	Low Vegetation	Static Car	Moving Car	Human	Mean IoU
fine-tune w/o deep supervision	78.5	72.2	44.0	65.3	43.5	17.4	51.5	1.2	46.7
fine-tune w deep supervision	79.2	72.5	**44.8**	64.6	44.3	17.0	52.8	3.4	47.3
fine-tune w+w/o deep supervision	**79.4**	**73.1**	43.7	**65.5**	**45.3**	**21.3**	**55.8**	**6.3**	**48.8**

depth of some layers in Dilation net, U-Net and MS-Dilation net are reduced to maximally fit into the memory. The model configuration detail of different networks is shown in Fig. 5 in appendix.

The neural networks share similar hyper-parameters for training from scratch. All models are trained with Adam optimizer for 27K iterations(20 epochs). The base learning rate is set to 10^{-4} exponentially decaying to 10^{-7}. Weight decay for all weights in convolutional kernels is set to 10^{-5}. Training is done with one image per batch. For data augmentation in training, we apply random left and right flip. We also apply a series of color augmentation, including random hue operation, random contrast operation, random brightness operation, random saturation operation.

Deep supervision losses are used for our MS-Dilation net. The loss weights for three streams are 1.8, 0.8 and 0.4 respectively. The loss weights for stream2 and stream3 are set smaller than stream1 as the main goal is to minimize the loss in stream1. For Dilation net, basic context aggregation module is used and initialized as it is in [25]. All networks are trained end-to-end and their mean IoU scores are reported in percentage as shown in Tab. I.

For the four networks, they are all better at discriminating building, road and tree classes, achieving IoU scores higher than 50%. The scores for car, vegetation and clutter classes are relatively lower. All four networks completely fail to discriminate human class. Normally, classes with larger pixel number have relatively higher IoU scores. However, IoU score for moving car class is much higher than static car class even though the two classes have similar pixel number. The reason may be that static cars may appear in various context like parking lot, garage, side walk or partially blocked under the trees, while moving cars are normally running in the middle of road with very clear view.

Our model achieves the best mean IoU score and the best IoU score for most of the classes among the four networks. It shows the effectiveness of multi-scale feature extraction.

B. Fine-tune Pre-trained Models

For fine-tuning pre-trained networks, all the networks are pre-trained with cityscapes dataset [6]. Finely annotated data from both training and validation splits are used, that is 3,450 densely labelled images in total. Hyper-parameters and data augmentation are set the same as they are in section IV-A, except that the iteration is set to 52K. Next, all the networks are fine-tuned with data from UAVid dataset. As there is still large heterogeneity between these two datasets, all layers are trained for all networks. We only initialize feature extraction parts of the networks with pre-trained models, while the prediction parts are initialized the same as training from scratch. The learning rate is set to 10^{-5} exponentially decaying to 10^{-7} for FCN-8s, and 10^{-4} exponentially decaying to 10^{-7} for other 3 networks as they are easily stuck at local minimum with initial learning rate to be 10^{-5} during training. The rest of the hyper-parameters are set the same as training from scratch. The performance is also shown in Tab. I.

To find out whether deep supervision losses are important, we have fine-tuned MS-Dilation net with 3 different deep supervision plans. For the first plan, we fine-tune MS-Dilation net without deep supervision losses for 30 epochs by setting loss weights to 0 in stream2 and stream3. For the second plan, we fine-tune MS-Dilation net with deep supervision losses for 30 epochs. For the final plan, we fine-tune MS-Dilation net with deep supervision losses for 20 epochs and without deep supervision losses for another 10 epochs. The IoU scores for three plans are shown in Tab. II. As it is shown, the best mean IoU score is achieved by the third plan. The better result for MS-Dilation net+PRT in Tab. I is achieved by fine-tuning 20 epochs without deep supervision losses after fine-tuning 20 epochs with deep supervision losses.

Clearly, deep supervision losses are very important for MS-Dilation net. However, neither purely fine-tuning the MS-Dilation net with deep supervision losses nor without achieves the best score. It is the combination of these two fine-tuning processes that brings the best score. Deep supervision losses are important as they can guide the multi-scale feature learning process, but the network needs to be further fine-tuned without deep supervision losses to get the best multi-scale filters for prediction.

By fine-tuning the pre-trained models, the performance boost is huge for all networks across all classes except human class. The networks still struggle to differentiate human class. Nevertheless, the improvement is evident for MS-Dilation net with 8% improvement. Decoupling the filters with different scales can be very beneficial when objects appear in large scale difference.

C. Video Semantic Segmentation

For video semantic segmentation, we apply methods used in feature space optimization (FSO) [32]. As FSO process a block of images simultaneously, 5 consecutive frames with 15 frames interval (0.5s-0.7s gap) are extracted from provided video files, which form a block spanning 2s to 3s, and the test image is located at the center in each block. The gap between consecutive frames is not set too big so as to get good flow extraction. It is better to have longer sequence to gain longer temporal regularization, but due to memory limitation, it is not possible to support more than 5 images in a 30G memory without sacrificing the image size.

FSO process in each block requires several ingredients. Contour strength for each image is calculated according to [34]. The unary for each image is set as the softmax layer output from each fine-tuned network. Forward flows and backward flows are calculated according to [35], [36]. As the computation speed for optical flow at original image scale is extremely low, the images to be processed are downsized by 8 times for both width and height, and the final flows at original scale are got through bicubic interpolation and magnification. Then, points trajectories can be calculated according to [37] with the forward and backward flows. Finally, a dense 3D CRF is applied after feature space optimization as described in [32].

The IoU scores for FSO results with unaries from different fine-tuned networks are reported in Tab. I. For each model, there is improvement in mean IoU score and IoU score for each individual class except for human and moving car classes. FSO favors more for class whose instance covers more image pixels, and IoU score improves less for class with smaller instance like static car and it drops for moving car and human classes. The human class IoU score for MS-Dilation net drops by a large margin, nearly 4%.

V. Conclusion and Outlook

In this paper, we present a new UAVid dataset to advance the development of video semantic segmentation. It captures complex street scenes in slant view style with very high resolution videos. Classes for the video semantic labelling task have been defined and labelled. The usability of our UAVid dataset has also been proved with several deep convolutional neural networks, among which the proposed Multi-Scale-Dilation net performs the best via multi-scale feature extraction. It has also been shown that pre-training the network is beneficial for all classes in UAVid semantic labelling task. In the future, we will continually collect new UAV video data, which will be labelled densely in temporal space. We will extend labelling from current classes to more classes including window, door, balcony, etc. The benchmark together with our labelling tool will be published online.

References

[1] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014, pp. 740–755.

[2] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *IJCV*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[3] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV*, 2008, pp. 44–57.

[4] B. Kim, J. Yim, and J. Kim, "Highway driving dataset for semantic video segmentation," in *BMVC*.

[5] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth, "Efficient multi-cue scene segmentation," in *GCPR*, 2013, pp. 435–445.

[6] 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 *CVPR*, 2016.

[7] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[9] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *CVPR*, vol. 1, no. 2, 2017, p. 4.

[10] F. Rottensteiner, G. Sohn, M. Gerke, J. Wegner, U. Breitkopf, and J. Jung, "Results of the isprs benchmark on urban object detection and 3d building reconstruction," *ISPRS journal of photogrammetry and remote sensing*, vol. 93, pp. 256–271, 2014.

[11] F. Galasso, N. Shankar Nagaraja, T. Jimenez Cardenas, T. Brox, and B. Schiele, "A unified video segmentation benchmark: Annotation, metrics and analysis," in *ICCV*, 2013, pp. 3527–3534.

[12] P. Sundberg, T. Brox, M. Maire, P. Arbeláez, and J. Malik, "Occlusion boundary detection and figure/ground assignment from optical flow," in *CVPR*, 2011, pp. 2233–2240.

[13] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *ECCV*, 2010, pp. 282–295.

[14] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *PAMI*, vol. 36, no. 6, pp. 1187–1200, 2014.

[15] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *CVPR*, 2016.

[16] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous uav surveillance in complex urban environments," in *WI-IAT*, 2009, pp. 82–85.

[17] D. Perez, I. Maza, F. Caballero, D. Scarlatti, E. Casado, and A. Ollero, "A ground control station for a multi-uav surveillance system," *Journal of Intelligent&Robotic Systems*, vol. 69, no. 1-4, pp. 119–130, 2013.

[18] H. Xiang and L. Tian, "Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (uav)," *Biosystems Engineering*, vol. 108, no. 2, pp. 174–190, 2011.

[19] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss, "Uav-based crop and weed classification for smart farming," in *ICRA*, 2017, pp. 3024–3031.

[20] N. Chebrolu, T. Läbe, and C. Stachniss, “Robust long-term registration of uav images of crop fields for precision agriculture,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, 2018.

[21] A. Milioto, P. Lottes, and C. Stachniss, “Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks,” *ISPRS Annals*, vol. 4, p. 41, 2017.

[22] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, “Learning social etiquette: Human trajectory understanding in crowded scenes,” in *ECCV*, 2016, pp. 549–565.

[23] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, *et al.*, “Slic superpixels compared to state-of-the-art superpixel methods,” *PAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.

[24] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *CVPR*, June 2015.

[25] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in *ICLR*, 2016.

[26] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *MICCAI*, 2015, pp. 234–241.

[27] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, “Pyramid methods in image processing,” *RCA Engineer*, vol. 29, no. 6, pp. 33–41, 1984.

[28] Y. Liu, B. Fan, L. Wang, J. Bai, S. Xiang, and C. Pan, “Semantic labeling in very high resolution images via a self-cascaded convolutional neural network,” *ISPRS Journal of Photogrammetry and Remote Sensing*, 2017.

[29] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, “One-shot video object segmentation,” in *CVPR*, 2017.

[30] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” *arXiv preprint arXiv:1802.02611*, 2018.

[31] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *CVPR*, 2017, pp. 2881–2890.

[32] A. Kundu, V. Vineet, and V. Koltun, “Feature space optimization for semantic video segmentation,” in *CVPR*, 2016, pp. 3168–3175.

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, “Tensorflow: a system for large-scale machine learning.” in *OSDI*, vol. 16, 2016, pp. 265–283.

[34] P. Dollár and C. L. Zitnick, “Fast edge detection using structured forests,” *PAMI*, vol. 37, no. 8, pp. 1558–1570, 2015.

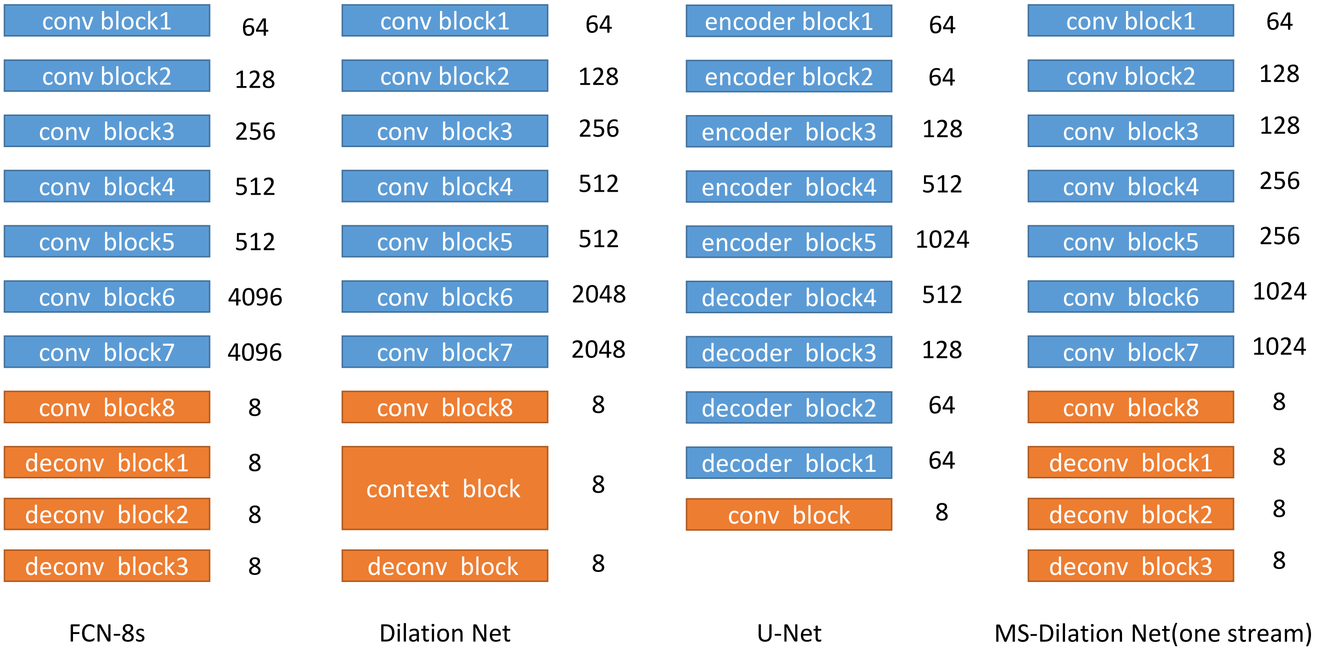
[35] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, “High accuracy optical flow estimation based on a theory for warping,” in *ECCV*, 2004, pp. 25–36.

[36] T. Brox and J. Malik, “Large displacement optical flow: descriptor matching in variational motion estimation,” *PAMI*, vol. 33, no. 3, pp. 500–513, 2011.

[37] N. Sundaram, T. Brox, and K. Keutzer, “Dense point trajectories by gpu-accelerated large displacement optical flow,” in *ECCV*, 2010, pp. 438–451.

APPENDIX

A. Network Details for Experiment



FCN-8s		Dilation Net		U-Net		MS-Dilation Net(one stream)	
conv block1	64	conv block1	64	encoder block1	64	conv block1	64
conv block2	128	conv block2	128	encoder block2	64	conv block2	128
conv block3	256	conv block3	256	encoder block3	128	conv block3	128
conv block4	512	conv block4	512	encoder block4	512	conv block4	256
conv block5	512	conv block5	512	encoder block5	1024	conv block5	256
conv block6	4096	conv block6	2048	decoder block4	512	conv block6	1024
conv block7	4096	conv block7	2048	decoder block3	128	conv block7	1024
conv block8	8	conv block8	8	decoder block2	64	conv block8	8
deconv block1	8	context block	8	decoder block1	64	deconv block1	8
deconv block2	8			conv block	8	deconv block2	8
deconv block3	8	deconv block	8			deconv block3	8

Fig. 5. **Network specifications.** Structures of 4 different networks are presented in 4 columns. Dropout layers, pooling layers and softmax layers are removed for simplicity. Digits beside the blocks present the number of filters for each convolution kernel in the block. As Dilation net, U-Net and MS-Dilation net cannot fit into a 12G GPU memory during training when image is of size 2048×1024, number of filters are reduced for some layers. During fine-tuning, blocks in blue colour are initialized with pre-trained model, while blocks in orange colour are initialized the same as training from scratch.

B. Big Image Clipping

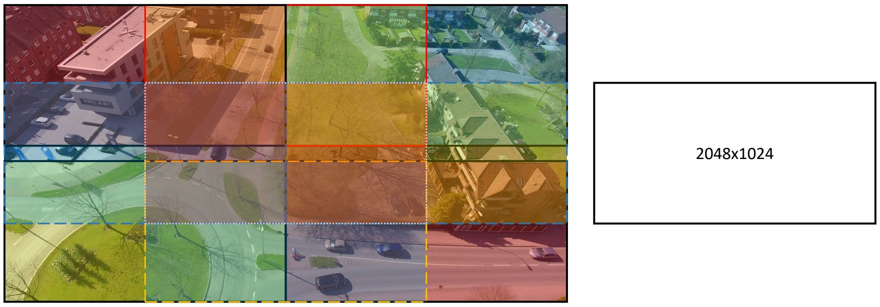


Fig. 6. **Image clipping illustrator.** The original images are of size 4096×2160 or 3840×2160, which are too big for training deep neural networks. Instead, the whole image is clipped into 9 evenly distributed smaller overlapped images for training, each of size 2048×1024.