2019 International Conference on Robotics and Automation (ICRA)
Palais des congres de Montreal, Montreal, Canada, May 20-24, 2019

Real-Time Joint Semantic Segmentation and Depth Estimation Using Asymmetric Annotations

Vladimir Nekrasov[1], Thanuja Dharmasiri[2], Andrew Spek[2], Tom Drummond[2], Chunhua Shen[1] and Ian Reid[1]

***Abstract*— Deployment of deep learning models in robotics as sensory information extractors can be a daunting task to handle, even using generic GPU cards. Here, we address three of its most prominent hurdles, namely, i) the adaptation of a single model to perform multiple tasks at once (in this work, we consider depth estimation and semantic segmentation crucial for acquiring geometric and semantic understanding of the scene), while ii) doing it in real-time, and iii) using asymmetric datasets with uneven numbers of annotations per each modality. To overcome the first two issues, we adapt a recently proposed real-time semantic segmentation network, making changes to further reduce the number of floating point operations. To approach the third issue, we embrace a simple solution based on hard knowledge distillation under the assumption of having access to a powerful 'teacher' network. We showcase how our system can be easily extended to handle more tasks, and more datasets, all at once, performing depth estimation and segmentation both indoors and outdoors with a single model. Quantitatively, we achieve results equivalent to (or better than) current state-of-the-art approaches with one forward pass costing just 13ms and 6.5 GFLOPs on 640×480 inputs. This efficiency allows us to directly incorporate the raw predictions of our network into the SemanticFusion framework [1] for dense 3D semantic reconstruction of the scene.[3]**

I. INTRODUCTION

As the number of tasks on which deep learning shows impressive results continues to grow in range and diversity, the number of models that achieve such results keeps analogously increasing, making it harder for practitioners to deploy a complex system that needs to perform multiple tasks at once. For some closely related tasks, such a deployment does not present a significant obstacle, as besides structural similarity, those tasks tend to share the same datasets, as, for example, the case of image classification, object detection, and semantic segmentation. On the other hand, tasks like segmentation and depth estimation rarely (fully) share the same dataset; for example, the NYUD dataset [2], [3] comprises a large set of annotations for depth estimation, but only a small labelled set of segmentations. One can readily approach this problem by simply updating the parameters of each task only if there exist ground truth annotations for that task. Unfortunately, this often leads to suboptimal results due to imbalanced and biased gradient updates. We note that while it is not clear how to handle such a scenario in the most general case, in this paper we assume that we have access to a large and powerful model, that can make an informative prediction to acquire missing labels. For each single task considered separately, this assumption is often-times valid, and we make use of it to predict missing segmentation masks.

Another issue that arises is the imperative in the context of robotics and autonomous systems for extraction of sensory information in real time. While there has been a multitude of successful approaches to speed up individual tasks [4]–[6], there is barely any prior work on performing multiple tasks concurrently in real-time. Here we show how to perform two tasks, depth estimation and semantic segmentation, in real-time with very few architectural changes and without any complicated pipelines.

Our choice of tasks is motivated by an observation that, for all sorts of robotic applications it is important for a robot (an agent) to know the semantics of its surroundings and to perceive the distances to the surfaces in the scene. The proposed methodology is simple and achieves competitive results in comparison to large models. Furthermore, we believe that there is nothing that prohibits practitioners and researchers to adapt our method for more tasks, which, in turn, would lead to better exploitation of deep learning models in real-world applications. To confirm this claim, we conduct additional experiments, predicting besides depth and segmentation, surface normals. Moreover, we successfully train a single model able to perform depth estimation and semantic segmentation, together in both indoor and outdoor settings. In yet another case study, we demonstrate that raw outputs of our joint network (segmentation and depth) can be directly used inside the SemanticFusion framework [1] to estimate dense semantic 3D reconstruction of the scene in real-time.

To conclude our introduction, we re-emphasise that our results demonstrate that there is no need to uncritically deploy multiple expensive models, when the same performance can be achieved with one small network - a case of one being better than two!

II. RELATED WORK

Our work is closely related to several topics. Among them are multi-task learning, semantic segmentation, depth estimation, and knowledge distillation.

According to the classical *multi-task learning* paradigm, forcing a single model to perform several related tasks simultaneously can improve generalisation via imposing an inductive bias on the learned representations [7], [8]. Such an approach assumes that all the tasks use a shared representation before learning task-specific parameters. Multiple works

[1]School of Computer Science, the University of Adelaide, Australia {firstname.lastname}@adelaide.edu.au
[2]Monash University, Australia {firstname.lastname}@monash.edu
[3]The models are available here: https://github.com/drsleep/multi-task-refinenet

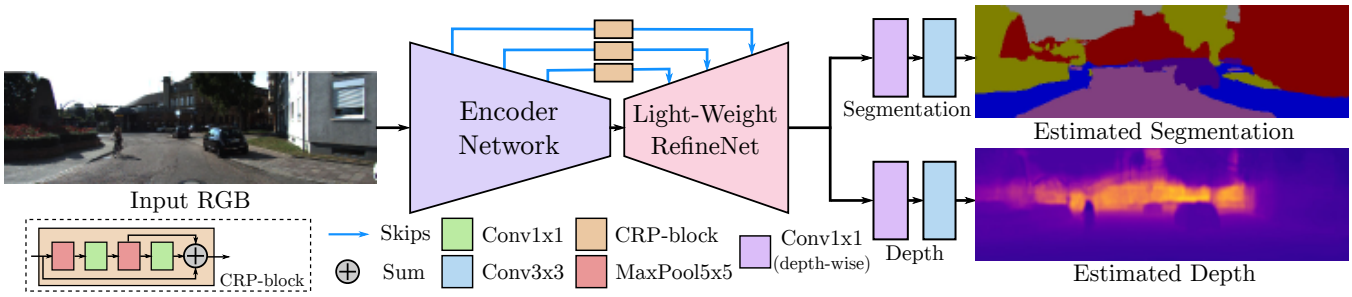

Fig. 1 – General network structure for joint semantic segmentation and depth estimation. Each task has only 2 specific parametric layers, while everything else is shared

in computer vision have been following this strategy; in particular, Eigen & Fergus [9] trained a single architecture (but with different copies) to predict depth, surface normals and semantic segmentation, Kokkinos [10] proposed a universal network to tackle 7 different vision tasks, Dvornik *et al.* [11] found it beneficial to do joint semantic segmentation and object detection, while Kendall *et al.* [12] learned optimal weights to perform instance segmentation, semantic segmentation and depth estimation all at once. Chen *et al.* [13] built a single network with the ResNet-50 [14] backbone performing joint semantic segmentation, depth estimation and object detection. To alleviate the problem of imbalanced annotations, Kokkinos [10] chose to accumulate the gradients for each task until a certain number of examples per task is seen, while Dvornik *et al.* [11] simply resorted to keeping the branch with no ground truth available intact until at least one example of that modality is seen.

We note that none of these methods makes any use of already existing models for each separate task, and none of them, with the exception of BlitzNet [11], achieves real-time performance. In contrast, we show how to exploit large pre-trained models to acquire better results, and how to do inference in real-time.

Semantic segmentation is a task of per-pixel label classification, and most approaches in recent years have been centered around the idea of adapting image classification networks into fully convolutional ones able to operate on inputs of different sizes [15]–[17]. Real-time usage of such networks with decent performance is a non-trivial problem, and few approaches are currently available [6], [18]–[20]. We have chosen recently proposed Light-Weight RefineNet [20] on top of MobileNet-v2 [21] as our baseline architecture as it exhibits solid performance on the standard benchmark dataset, PASCAL VOC [22] in real-time, while having fewer than 4M parameters.

Depth estimation is another per-pixel task, the goal of which is to determine how far each pixel is from the observer. Traditionally, image based depth reconstruction was performed using SLAM based approaches [23]–[25]. However, recent machine learning approaches have achieved impressive results, where a CNN has been successfully employed to predict a depth map from a single RGB image using supervised learning [9], [26]–[28], unsupervised learning [29], [30] and semi-supervised learning [31]. Predicting multiple quantities including depths from a single image was first tackled by Eigen & Fergus [9]. Dharmasiri *et al.* [32] demonstrated that predicting related structural information in the form of depths, surface normals and surface curvature results in improved performances of all three tasks compared to utilising three separate networks. Most recently, Qi *et al.* [33] found it beneficial to directly encode a geometrical structure as part of the network architecture in order to perform depth estimation and surface normals estimation simultaneously. Our approach is fundamentally different to these previous works in two ways. Firstly, our network exhibits real-time performance on each individual task. Secondly, we demonstrate how to effectively incorporate asymmetric and uneven ground truth annotations into the training regime. Furthermore, it should be noted that despite using a smaller model running in real-time, we still quantitatively outperform these approaches.

Finally, we briefly touch upon the *knowledge distillation* approach [34]–[37] that is based on the idea of having a large pre-trained teacher (expert) network (or an ensemble of networks), and using its logits, or predictions directly, as a guiding signal for a small network along with original labels. Several previous works relied on knowledge distillation to either acquire missing data [38], or as a regulariser term [39], [40]. While those are relevant to our work, we differ along several axes: most notably, Zamir *et al.* [38] require separate network copies for different tasks, while Hoffman *et al.* [39] and Li & Hoiem [40] only consider a single task learning (object detection and image classification, respectively).

III. Methodology

While we primarily discuss the case with only two tasks present, the same machinery applies for more tasks, as demonstrated in Sect. V-A.

A. Backbone Network

As mentioned in the previous section, we employ the recently proposed Light-Weight RefineNet architecture [20] built on top of the MobileNet-v2 classification network [21]. This architecture extends the classifier by appending several simple contextual blocks, called Chained Residual Pooling (CRP) [41], consisting of a series of 5×5 max-pooling and 1×1 convolutions (Fig. 1).

Even though the original structure already achieves real-time performance and has a small number of parameters, for the joint task of depth estimation and semantic segmentation (of 40 classes) it requires more than 14 GFLOPs on inputs

of size 640×480, which may hinder it from the direct deployment on mobile platforms with few resources available. We found that the last CRP block is responsible for more than half of the FLOPs as it deals with the high-resolution feature maps ($1/4$ from the original resolution). Thus, to decrease its influence, we replace 1×1 convolution in the last CRP block with its depthwise equivalent (i.e., into a grouped convolution with the number of groups being equal to the number of input channels) [42]. Doing so reduces the number of operations by more than half, down to just 6.5 GFLOPs.

B. Joint Semantic Segmentation and Depth Estimation

In the general case, it is non-trivial to decide where to branch out the backbone network into separate task-specific paths in order to achieve the optimal performance on all of them simultaneously. For simplicity, we branch out right after the last CRP block, and append two additional convolutional layers (one depthwise 1×1 and one plain 3×3) for each task (Fig. 1).

If we denote the output of the network before the branching as $\tilde{y} = f_{\theta_b}(I)$, where f_{θ_b} is the backbone network with a set of parameters θ_b, and I is the input RGB-image, then the depth and segmentation predictions can be denoted as $\tilde{y}_s = g_{\theta_s}(\tilde{y})$ and $\tilde{y}_d = g_{\theta_d}(\tilde{y})$, where g_{θ_s} and g_{θ_d} are segmentation and depth estimation branches with the sets of parameters θ_s and θ_d, respectively. We use the standard softmax cross-entropy loss for segmentation and the inverse Huber loss for depth estimation [27]. Our total loss (Eqn. (1)) contains an additional scaling parameter, λ, which, for simplicity, we set to 0.5:

$$
\begin{aligned}
\mathcal{L}_{total}(I, G_s, G_d; \theta_b, \theta_s, \theta_d) &= (\lambda \cdot \mathcal{L}_{segm}(I, G_s; \theta_b, \theta_s) + \\
&\quad (1-\lambda) \cdot \mathcal{L}_{depth}(I, G_d; \theta_b, \theta_d)), \\
\mathcal{L}_{segm}(I, G) &= \frac{-1}{|I|} \sum_{i \in I} log(softmax(\tilde{y}_s)_{iG_i}), \\
\mathcal{L}_{depth}(I, G) &= \begin{cases} |\tilde{y}_d - G|, \text{ if } |\tilde{y}_d - G| \leq c \\ ((\tilde{y}_d - G)^2 + c^2)/(2c), \text{ otherwise,} \end{cases} \\
c &\stackrel{\text{def}}{=} 0.2 \cdot \max |\tilde{y}_d - G|,
\end{aligned}
\tag{1}
$$

where G_s and G_d denote ground truth segmentation mask and depth map, correspondingly; $(\cdot)_{ij}$ in the segmentation loss is the probability value of class j at pixel i.

C. Expert Labeling for Asymmetric Annotations

As one would expect, it is impossible to have all the ground truth sensory information available for each single image. Quite naturally, this poses a question of how to deal with a set of images $S = \{I\}$ among which some have an annotation of one modality, but not another. Assuming that one modality is always present for each image, this then divides the set S into two disjoint sets $S_1 = S_{T_1}$ and $S_2 = S_{T_1,T_2}$ such that $S = S_1 \cup S_2$, where T_1 and T_2 denote two tasks, respectively, and the set S_1 consists of images for which there are no annotations of the second task available, while S_2 comprises images having both sets of annotations.

Plainly speaking, there is nothing that prohibits us from still exploiting equation (1), in which case only the weights of the branch with available labels will be updated. As we show in our experiments, this leads to biased gradients and, consequently, sub-optimal solutions. Instead, emphasising the need of updating both branches simultaneously, we rely on an expert model to provide us with noisy estimates in place of missing annotations.

More formally, if we denote the expert model on the second task as E_{T_2}, then its predictions $\tilde{S}_1 = E_{T_2}(S_1)$ on the set S_1 can be used as synthetic ground truth data, which we will use to pre-train our joint model before the final fine-tuning on the original set S_2 with readily available ground truth data for both tasks. Here, we exploit the labels predicted by the expert network instead of logits, as storing a set of large 3-D floating point tensors requires extensive resources.

Note also that our framework is directly transferable to cases when the set S comprises several datasets. In Sect. V-B we showcase a way of exploiting all of them in the same time using a single copy of the model.

IV. Experimental Results

In our experiments, we consider two datasets, NYUDv2-40 [2], [3] and KITTI [47], [48], representing indoor and outdoor settings, respectively, and both being used extensively in the robotics community.

All the training experiments follow the same protocol. In particular, we initialise the classifier part using the weights pre-trained on ImageNet [49], and train using mini-batch SGD with momentum with the initial learning rate of $1e$-3 and the momentum value of 0.9. Following the setup of Light-Weight RefineNet [20], we keep batch norm statistics frozen. We divide the learning rate by 10 after pre-training on a large set with synthetic annotations. We train with a random square crop of 350×350 augmented with random mirroring.

All our networks are implemented in PyTorch [50]. To measure the speed performance, we compute 100 forward passes and report both the mean and standard deviation values, as done in [20]. Our workstation has 24GB RAM, Intel i5-7600 processor and a single GT1080Ti GPU card running CUDA9.0 and CuDNN7.0.

A. NYUDv2

NYUDv2 is an indoor dataset with 40 semantic labels. It contains 1449 RGB images with both segmentation and depth annotations, of which 795 comprise the training set and 654 - validation. The raw dataset contains more than $300,000$ training images with depth annotations. During training we use less than 10% (25K images) of this data. As discussed in Sect. III-C, we annotate these images for semantic segmentation using a teacher network (here, we take the pre-trained Light-Weight RefineNet-152 [20] that achieves 44.4% mean iou on the validation set). After acquiring the synthetic annotations, we pre-train the network

TABLE I – Results on the test set of NYUDv2. The speed of a single forward pass and the number of FLOPs are measured on 640×480 inputs. For the reported mIoU the higher the better, whereas for the reported RMSE the lower the better. (†) means that both tasks are performed simultaneously using a single model, while (‡) denotes that two tasks employ the same architecture but use different copies of weights per task

		Sem. Segm.	**Depth Estimation**		**General**		
Model	Regime	mIoU,%	RMSE (lin),m	RMSE (log)	Parameters,M	GFLOPs	speed,ms (mean/std)
†**Ours**	Segm,Depth	42.02	0.565	0.205	**3.07**	**6.49**	**12.8±0.1**
RefineNet-101 [41]	Segm	**43.6**	–	–	118	–	60.3 ± 0.5
RefineNet-LW-50 [20]	Segm	41.7	–	–	27	33	19.6 ± 0.3
Context [43]	Segm	40.6	–	–	–	–	–
†Sem-CRF+ [44]	Segm,Depth	39.2	0.816	0.314	–	–	–
‡Kendall and Gal [45]	Segm,Depth	37.3	**0.506**	–	–	–	150
Fast Res.Forests [46]	Segm	34.3	–	–	–	–	48.4
‡Eigen and Fergus [9]	Segm,Depth	34.1	0.641	0.214	–	–	–
Laina *et al.* [27]	Depth	–	0.573	**0.195**	63.6	–	55
†Qi *et al.* [33]	Depth,Normals	–	0.569	-	-	–	870

on the large set, and then fine-tune it on the original small set of 795 images.

Quantitatively, we are able to achieve 42.02% mean iou and 0.565m RMSE (lin) on the validation set (Table I), outperforming several large models, while performing both tasks in real-time simultaneously. More detailed results for depth estimation are given in Table II, and qualitative results are provided in Fig. 2.

TABLE II – Detailed results on the test set of NYUDv2 for the depth estimation task. For the reported RMSE, abs rel and sqr rel the lower the better, whereas for accuracies (δ) the higher the better

	Ours	Laina *et al.* [27]	Kendall and Gal [45]	Qi *et al.* [33]
RMSE (lin)	0.565	0.573	**0.506**	0.569
RMSE (log)	0.205	**0.195**	–	
abs rel	0.149	0.127	**0.11**	–
sqr rel	**0.105**	–	–	0.128
$\delta < 1.25$	0.790	0.811	0.817	**0.834**
$\delta < 1.25^2$	0.955	0.953	0.959	**0.960**
$\delta < 1.25^3$	**0.990**	0.988	0.989	**0.990**

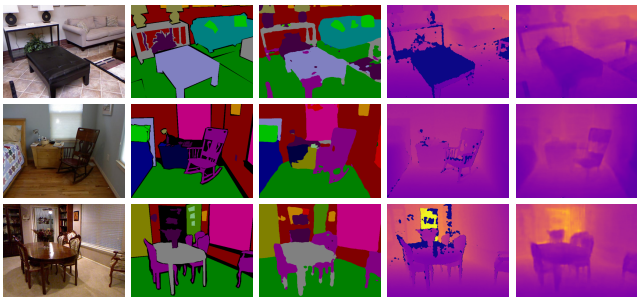

Fig. 2 – Qualitative results on the test set of NYUD-v2. The black and dark-blue pixels in 'GT-Segm' and 'GT-Depth' respectively, indicate pixels without an annotation or label

Ablation Studies. To evaluate the importance of pre-training using the synthetic annotations and benefits of performing two tasks jointly, we conduct a series of ablation experiments. In particular, we compare three baseline models trained on the small set of 795 images and three other approaches that make use of additional data - ours with noisy estimates from a larger model, and two methods, one by Kokkinos [10], where the gradients are being accumulated until a certain number of examples is seen, and one by Dvornik *et al.* [11], where the task branch is updated every time at least one example is seen.

The results of our experiments are given in Table III. The first observation that we make is that performing two tasks jointly on the small set does not provide any significant benefits for each separate task, and even substantially harms semantic segmentation. In contrast, having a large set of depth annotations results in valuable improvements in depth estimation and even semantic segmentation, when it is coupled with a clever strategy of accumulating gradients. Nevertheless, none of the methods can achieve competitive results on semantic segmentation, whereas our proposed approach reaches better performance without any changes to the underlying optimisation algorithm.

TABLE III – Results of ablation experiments on the test set of NYUDv2. *SD* means how many images have a joint pair of annotations - both segmentation (*S*) and depth (*D*); *task update frequency* denotes the number of examples of each task to be seen before performing a gradient step on task-specific parameters; *base update frequency* is the number of examples to be seen (regardless of the task) before performing a gradient step on shared parameters

	Annotations		**Update Frequency**		**Segm.**	**Depth**
Method	Pre-Training	Fine-Tuning	Task	Base	mIoU,%	RMSE (lin),m
Baseline (SD)	795SD	–	1	1	32.48	0.6328
Baseline (S)	795S	–	1	1	34.44	–
Baseline (D)	795D	–	1	1	–	0.6380
BlitzNet [11]	25405D + 795SD	795SD	1	1	34.82	0.5823
UberNet [10]	25405D + 795SD	795SD	10	30	35.88	0.5728
Ours	25405SD	795SD	1	1	**42.02**	**0.5648**

B. KITTI

KITTI is an outdoor dataset that contains 100 images semantically annotated for training (with 11 semantic classes) and 46 images for testing [48] without ground truth depth maps. Following previous work by [51], we keep only 6 well-represented classes.

Besides segmentation, we follow [26] and employ 20000 images with depth annotations available for training [47], and 697 images for testing. Due to similarities with the CityScapes dataset [52], we consider ResNet-38 [16] trained on CityScapes as our teacher network to annotate the training images that have depth but not semantic segmentation. In turn, to annotate missing depth annotations on 100 images with semantic labels from KITTI-6, we first trained a separate copy of our network on the depth task only, and then

TABLE IV – Results on the test set of KITTI-6 for segmentation and KITTI for depth estimation

		Sem. Segm.	Depth Estimation		General			
Model	Regime	mIoU,%	RMSE (lin),m	RMSE (log)	Parameters,M	Input Size	GFLOPs	speed,ms (mean/std)
Ours	Segm,Depth	**87.02**	**3.453**	0.182	**2.99**	1200x350	**6.45**	**16.9±0.1**
Fast Res.Forests [46]	Segm	84.9	–	–	–	1200x350	–	106.35
Wang et al. [51]	Segm	74.8	–	–	–	–	–	–
Garg [29]	Depth	–	5.104	0.273	–	–	–	–
Goddard [30]	Depth	–	4.471	0.232	31	512x256	–	35.0
Kuznietsov [31]	Depth	–	3.518	**0.179**	–	621x187	–	48.0

used it as a teacher. Note that we abandoned this copy of the network and did not make any further use of it.

After pre-training on the large set, we fine-tune the model on the small set of 100 examples. Our quantitative results are provided in Table IV, while visual results can be seen on Fig. 3. Per-class segmentation results are given in Table V. As evident, our approach outperforms other competing methods across a large set of metrics - both on semantic segmentation and depth estimation, while being light-weight and having low latency.

TABLE V – Detailed segmentation results on the test set of KITTI-6

Model	sky	building	road	sidewalk	vegetation	car	Total
Ours	85.1	**87.7**	**92.8**	**82.7**	86.1	**87.6**	**87.0**
Fast Res.Forests [46]	84.5	85.9	92.3	78.8	**87.8**	80.3	84.9
Wang et al. [51]	**88.6**	80.1	80.9	43.6	81.6	63.5	74.8

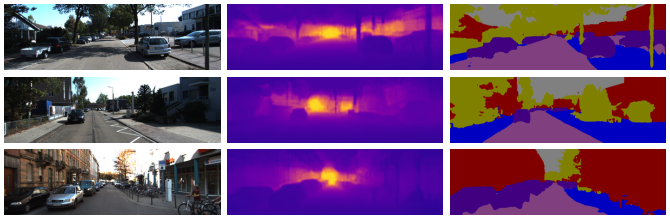

Image Pred-Depth Pred-Segm

Fig. 3 – Qualitative results on the test set of KITTI (for which only GT depth maps are available). We do not visualise GT depth maps due to their sparsity

V. Extensions

The goal of this section is to demonstrate the ease with which our approach can be directly applied in other practical scenarios, such as, for example, the deployment of a single model performing three tasks at once, and the deployment of a single model performing two tasks at once under two different scenarios - indoor and outdoor. As the third task, here we consider surface normals estimation, and as two scenarios, we consider training a single model on both NYUD and KITTI simultaneously without the necessity of having a separate copy of the same architecture for each dataset.

In this section, we strive for simplicity and do not aim to achieve high performance numbers, thus we directly apply the same training scheme as outlined in the previous section.

A. Single Model - Three Tasks

Analogously to the depth and segmentation branches, we append the same structure with two convolutional layers for surface normals. We employ the negative dot product (after normalisation) as the training loss for surface normals, and we multiply the learning rate for the normals parameters by 10, as done in [9].

We exploit the raw training set of NYUDv2 [2] with more than $300,000$ images, having (noisy) depth maps from the Kinect sensor and with surface normals computed using the toolbox provided by the authors. To acquire missing segmentation labels, we repeat the same procedure outlined in the main experiments - in particular, we use the Light-Weight RefineNet-152 network [20] to get noisy labels. After pre-training on this large dataset, we divide the learning rate by 10 and fine-tune the model on the small dataset of 795 images having annotations for each modality. For surface normals, we employ the annotations provided by Silberman *et al.* [2].

Our straightforward approach achieves practically the same numbers on depth estimation, but suffers a significant performance drop on semantic segmentation (Table VI). This might be directly caused by the excessive number of imperfect and noisy labels, on which the semantic segmentation part is being pre-trained. Nevertheless, the results on all three tasks remain competitive, and we are able to perform all three of them in real-time simultaneously. We provide a few examples of our approach on Figure 4.

TABLE VI – Results on the test set of NYUDv2 of our single network predicting three modalities at once with surface normals annotations from [2]. The speed of a single forward pass is measured on 640×480 inputs. Baseline results (with a single network performing only segmentation and depth) are in **bold**

Segm.	Depth		Surface Normals		General
mIoU,%	RMSE (lin),m	RMSE (log)	Mean Angle	Median Angle	speed,ms (mean/std)
38.66	0.566	0.209	23.95	17.74	13.4±0.1
42.02	**0.565**	**0.205**	–	–	**12.8±0.1**

B. Single Model - Two Datasets, Two Tasks

Next, we consider the case when it is undesirable to have a separate copy of the same model architecture for each dataset. Concretely, our goal is to train a single model that is able to perform semantic segmentation and depth estimation on both NYUD and KITTI at once. To this end, we simply concatenate both datasets and amend the segmentation branch to predict 46 labels (40 from NYUD and 6 from KITTI-6).

We follow the exact same training strategy, and after pre-training on the union of large sets, we fine-tune the model on the union of small training sets. Our network exhibits no difficulties in differentiating between two regimes

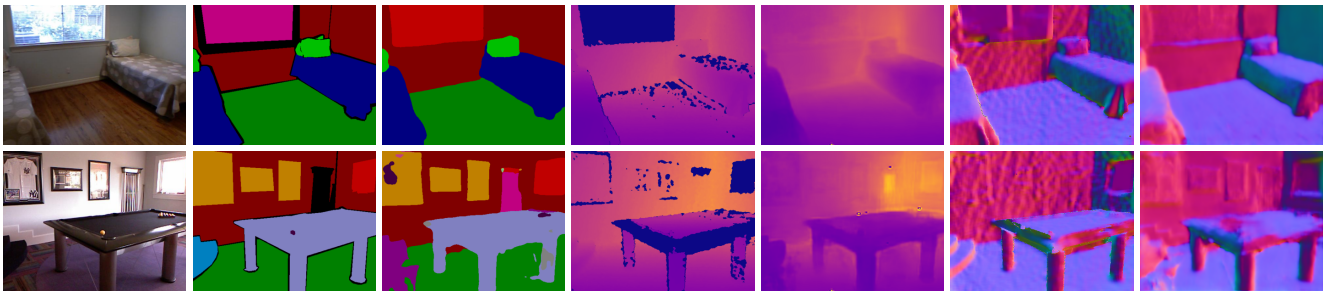

Fig. 4 – Qualitative results on the test set of NYUD-v2 for three tasks. The black pixels in the 'GT-Segm' images indicate those without a semantic label, whereas the dark blue pixels in the 'GT-Depth' images indicate missing depth values

(Table VII), and achieves results at the same level with the separate approach on each of the datasets without a substantial increase in model capacity.

TABLE VII – Results on the test set of NYUDv2, KITTI (for depth) and KITTI-6 (for segmentation) of our single network predicting two modalities on both datasets together. Baseline results (with separate networks per dataset) are in **bold**

NYUDv2			KITTI		
Segm.	**Depth**		**Segm.**	**Depth**	
mIoU,%	RMSE (lin),m	RMSE (log)	mIoU,%	RMSE (lin),m	RMSE (log)
38.76	0.59	0.213	86.1	3.659	0.190
42.02	**0.565**	**0.205**	**87.0**	**3.453**	**0.182**

C. Dense Semantic SLAM

Finally, we demonstrate that quantities predicted by our joint network performing depth estimation and semantic segmentation indoors can be directly incorporated into existing SLAM frameworks.

In particular, we consider SemanticFusion [1], where the SLAM reconstruction is carried out by ElasticFusion [53], which relies on RGB-D inputs in order to find dense correspondences between frames. A separate CNN, also operating on RGB-D inputs, was used by McCormac *et al.* [1] to acquire 2D semantic segmentation map of the current frame. A dense 3D semantic map of the scene is obtained with the help of tracked poses predicted by the SLAM system.

We consider one sequence of the NYUD validation set provided by the authors[1], and directly replace ground truth depth measurements with the outputs of our network performing depth and segmentation jointly (Sect. IV-A). Likewise, we do not make use of the authors' segmentation CNN and instead exploit segmentation predictions from our network. Note also that our segmentation network was trained on 40 semantic classes, whereas here we directly re-map the results into the 13-classes domain [54]. We visualise dense surfel-based reconstruction along with dense segmentation and current frame on Fig. 5. Please refer to the supplementary video material[2] for the full sequence results.

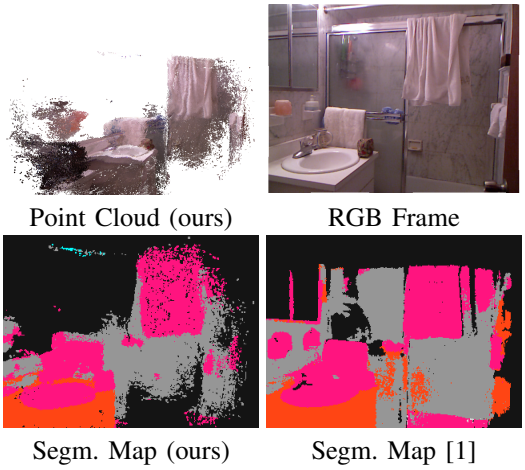

Fig. 5 – 3D reconstruction output using our per-frame depths and segmentation inside SemanticFusion [1]

VI. Conclusion

We believe that efficient and effective exploitation of visual information in robotic applications using deep learning models is crucial for further development and deployment of robots and autonomous vehicles. To this end, we presented a simple way of achieving real-time performance for the joint task of depth estimation and semantic segmentation. We showcased that it is possible (and indeed beneficial) to re-use large existing models in order to generate synthetic labels important for the pre-training stage of a compact model. Moreover, our method can be easily extended to handle more tasks and more datasets simultaneously, while raw depth and segmentation predictions of our network can be seamlessly used within available dense SLAM systems. As our future work, we will consider whether it would be possible to directly incorporate expert's uncertainty during the pre-training stage to acquire better results, as well as the case when there is no reliable expert available. Another interesting direction lies in incorporating findings of Zamir *et al.* [38] in order to reduce the total number of training annotations without sacrificing performance.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful and constructive comments. This research was supported by the Australian Research Council through the Australian Centre for Robotic Vision (CE140100016), the ARC Laureate Fellowship FL130100102 to IR, and the HPC cluster Phoenix at the University of Adelaide.

[1] https://bitbucket.org/dysonroboticslab/semanticfusion/overview

[2] https://youtu.be/qwShIBhaq8Y

REFERENCES

[1] J. McCormac, A. Handa, A. J. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *ICRA*, 2017. 1, 6

[2] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *ECCV*, 2012. 1, 3, 5

[3] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *CVPR*, 2013. 1, 3

[4] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. 1

[5] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, vol. abs/1602.07360, 2016. 1

[6] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," *CoRR*, vol. abs/1704.08545, 2017. 1, 2

[7] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," in *ICML*, 1993. 1

[8] J. Baxter, "A model of inductive bias learning," *J. Artif. Intell. Res.*, 2000. 1

[9] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *ICCV*, 2015. 2, 4, 5

[10] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *CVPR*, 2017. 2, 4

[11] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid, "Blitznet: A real-time deep network for scene understanding," in *ICCV*, 2017. 2, 4

[12] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," *CoRR*, vol. abs/1705.07115, 2017. 2

[13] L. Chen, Z. Yang, J. Ma, and Z. Luo, "Driving scene perception network: Real-time joint detection, depth estimation and semantic segmentation," in *WACV*, 2018. 2

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016. 2

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015. 2

[16] Z. Wu, C. Shen, and A. van den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *CoRR*, vol. abs/1611.10080, 2016. 2, 4

[17] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017. 2

[18] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *CoRR*, vol. abs/1606.02147, 2016. 2

[19] X. Li, Z. Liu, P. Luo, C. C. Loy, and X. Tang, "Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade," in *CVPR*, 2017. 2

[20] V. Nekrasov, C. Shen, and I. Reid, "Light-weight refinenet for real-time semantic segmentation," in *BMVC*, 2018. 2, 3, 4, 5

[21] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," *CoRR*, vol. abs/1801.04381, 2018. 2

[22] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *IJCV*, 2010. 2

[23] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *ICCV*, 2011. 2

[24] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in *ECCV*, 2014. 2

[25] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *ISMAR*, 2007. 2

[26] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *NIPS*, 2014. 2, 4

[27] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *3DV*, 2016. 2, 3, 4

[28] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *CVPR*, 2015. 2

[29] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *ECCV*, 2016. 2, 5

[30] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017. 2, 5

[31] Y. Kuznietsov, J. Stückler, and B. Leibe, "Semi-supervised deep learning for monocular depth map prediction," in *CVPR*, 2017. 2, 5

[32] T. Dharmasiri, A. Spek, and T. Drummond, "Joint prediction of depths, normals and surface curvature from rgb images using cnns," *arXiv preprint arXiv:1706.07593*, 2017. 2

[33] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "Geonet: Geometric neural network for joint depth and surface normal estimation," in *CVPR*, 2018. 2, 4

[34] C. Bucila, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *ACM SIGKDD*, 2006. 2

[35] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015. 2

[36] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *NIPS*, 2014. 2

[37] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *CoRR*, vol. abs/1412.6550, 2014. 2

[38] A. R. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *CVPR*, 2018. 2, 6

[39] J. Hoffman, S. Gupta, and T. Darrell, "Learning with side information through modality hallucination," in *CVPR*, 2016. 2

[40] Z. Li and D. Hoiem, "Learning without forgetting," *TPAMI*, 2017. 2

[41] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *CVPR*, 2017. 2, 4

[42] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *CVPR*, 2017. 3

[43] G. Lin, C. Shen, I. D. Reid, and A. van den Hengel, "Efficient piecewise training of deep structured models for semantic segmentation," *CoRR*, vol. abs/1504.01013, 2015. 4

[44] A. Mousavian, H. Pirsiavash, and J. Kosecka, "Joint semantic segmentation and depth estimation with deep convolutional networks," in *3DV*, 2016. 4

[45] A. Kendall and Y. Gal, "What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?" in *arXiv:1703.04977*, 2017. 4

[46] Y. Zuo and T. Drummond, "Fast residual forests: Rapid ensemble learning for semantic segmentation," in *CoRL*, 2017. 4, 5

[47] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *I. J. Robotics Res.*, 2013. 3, 4

[48] G. Ros, S. Ramos, M. Granados, A. Bakhtiary, D. Vazquez, and A. Lopez, "Vision-based offline-online perception paradigm for autonomous driving," in *WACV*, 2015. 3, 4

[49] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009. 3

[50] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017. 3

[51] S. Wang, S. Fidler, and R. Urtasun, "Holistic 3d scene understanding from a single geo-tagged image," in *CVPR*, 2015. 4, 5

[52] 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. 4

[53] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, 2016. 6

[54] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," *arXiv preprint arXiv:1301.3572*, 2013. 6