A Semi-Supervised Approach to Monocular Depth Estimation, Depth Refinement, and Semantic Segmentation of Driving Scenes using a Siamese Triple Decoder Architecture

John Paul T. Yusiong[1,2] and Prospero C. Naval, Jr.[1]
[1]Computer Vision and Machine Intelligence Group, Department of Computer Science
College of Engineering, University of the Philippines, Diliman, Quezon City, Philippines
[2]Division of Natural Sciences and Mathematics
University of the Philippines Visayas Tacloban College, Tacloban City, Leyte, Philippines
E-mail: jtyusiong@up.edu.ph; pcnaval@up.edu.ph

Keywords: Siamese triple decoder architecture, depth estimation and refinement, semantic segmentation, semi-supervised learning methods

Received: November 30, 2019

Depth estimation and semantic segmentation are two fundamental tasks in scene understanding. These two tasks are usually solved separately, although they have complementary properties and are highly correlated. Jointly solving these two tasks is very beneficial for real-world applications that require both geometric and semantic information. Within this context, the paper presents a unified learning framework for generating a refined depth estimation map and semantic segmentation map given a single image. Specifically, this paper proposes a novel architecture called JDSNet. JDSNet is a Siamese triple decoder architecture that can simultaneously perform depth estimation, depth refinement, and semantic labeling of a scene from an image by exploiting the interaction between depth and semantic information. A semi-supervised method is used to train JDSNet to learn features for both tasks where geometry-based image reconstruction methods are employed instead of ground-truth depth labels for the depth estimation task while ground-truth semantic labels are required for the semantic segmentation task. This work uses the KITTI driving dataset to evaluate the effectiveness of the proposed approach. The experimental results show that the proposed approach achieves excellent performance on both tasks, and these indicate that the model can effectively utilize both geometric and semantic information.

Povzetek: V članku je predstavljena izvirna metoda delno nadzorovanega učenja za raznovrstne vizualne naloge.

1 Introduction

Scene understanding is crucial for autonomous driving systems since it provides a mechanism to understand the scene layout of the environment [1, 2]. Scene understanding involves depth estimation and semantic segmentation, which facilitates the understanding of the geometric and semantic properties of a scene, respectively. Depth estimation and semantic segmentation address different areas in scene understanding but have complementary properties and are highly correlated.

For semantic segmentation, depth values help improve semantic understanding by enabling the model to generate more accurate object boundaries or differentiate objects having a similar appearance since these values encode structural information of the scene. On the other hand, for depth estimation, the semantic labels provide valuable prior knowledge to depict the geometric relationships between pixels of different classes and generate better scene layout [3, 4, 5, 6]. Thus, these two fundamental tasks in computer vision can be dealt with in an integrated manner under a unified framework that optimizes multiple objectives to improve computational efficiency and performance for both tasks from single RGB images. However, addressing depth estimation and semantic segmentation simultaneously where the two tasks can benefit from each other is non-trivial and is one of the most challenging tasks in computer vision given the peculiarities of each task and the limited information that can be obtained from monocular images.

Previous works jointly model these two tasks using traditional hand-crafted features and RGB-D images [7, 8]. However, the hand-crafted feature extraction process is quite tedious, and it generally fails to help achieve high accuracies while RGB-D image acquisition is a costly endeavor. To overcome the aforementioned issues, researchers employ a unified framework based on deep learning that enables these two tasks to enhance each other using single RGB images only, and this approach led to a significant breakthrough for both tasks [4, 5, 6, 9, 10, 11, 12]. Since these unified frameworks are based on the fully-supervised learning method, they require vast quantities of training images with per-pixel ground-truth semantic labels and depth measurements, and obtaining these ground-truths is non-trivial, costly, and labor-intensive. An alternative approach, as proposed by Ramirez *et al.* [13], is to integrate depth estimation and semantic segmentation into a unified framework using the semi-supervised learning method. The semi-supervised learning framework requires ground-truth semantic labels to provide supervisory signals for the semantic segmentation task, while for the depth estimation task, it employs geometry-based image reconstruction methods that utilize secondary information based on the underlying theory of epipolar constraints instead of requiring ground-truth depth measurements during training. In other words, addressing the problem of scene understanding assumes that both stereo image pairs and semantic information are available during training since this framework exploits the relationship between the geometric and semantic properties of a scene by performing semantic segmentation in a supervised manner and casting the depth estimation task as an image reconstruction problem in an unsupervised manner.

This paper presents another attempt towards addressing the joint inference problem involving depth estimation and semantic segmentation from a single image by proposing to train a novel architecture using a unified learning framework based on a semi-supervised technique. This paper introduces a novel Siamese triple decoder architecture with a disparity refinement module and a segmentation fusion module. The triple decoder architecture consists of one shared encoder and three parallel decoders. The disparity refinement module handles visual artifacts and blurred boundaries to generate better depth maps with no border artifacts around the image boundary while the segmentation fusion module generates the semantic segmentation map. In contrast, previous works apply a non-trainable post-processing heuristic during testing to refine the depth estimation outputs of the trained model [13, 14]. Essentially, the proposed method enables the model to simultaneously perform depth estimation, depth refinement, and semantic labeling of a scene from an image by exploiting the interaction between depth and semantic information in an end-to-end manner.

The main contributions of this work are the following:

1. It introduces a novel Siamese triple decoder architecture with a disparity refinement module and a segmentation fusion module, referred to as JDSNet, for depth estimation, depth refinement, and semantic segmentation.

2. It presents a unified framework for joint depth estimation with depth refinement and semantic segmentation from a single image based on a semi-supervised technique and trains JDSNet to simultaneously perform depth estimation, depth refinement, and semantic segmentation in an end-to-end manner using rectified stereo image pairs with ground-truth semantic labels as training data.

3. It describes a training loss function that optimizes these two tasks concurrently.

4. It demonstrates that the proposed method is capable of simultaneously addressing these two tasks that are mutually beneficial to both tasks. The experimental results prove that jointly solving these two tasks improves the performance of both tasks on various evaluation metrics.

The remainder of the paper is arranged as follows. Section 2 introduces the related works. Section 3 describes the proposed semi-supervised learning framework for simultaneous monocular depth estimation, depth refinement, and semantic segmentation. Section 4 discusses the experimental results using a standard benchmark dataset. Lastly, Section 5 concludes the paper.

2 Related work

This section focuses on the previous works that dealt with joint depth estimation and semantic segmentation where researchers attempted to develop better-suited models using different methods, such as traditional hand-crafted feature extraction techniques and deep learning-based techniques.

The earliest works [7, 8] show the feasibility of jointly modeling depth estimation and semantic segmentation from a single RGB image using the supervised learning method. However, they employ traditional hand-crafted features for these two tasks. The work of Ladicky *et al.* [7] is considered to be the first to jointly perform monocular depth estimation and semantic segmentation. Using properties of perspective geometry, they proposed an unbiased semantic depth classifier and considered both the loss from semantic and depth labels when training the classifier. They obtained results that outperformed previous state-of-the-art traditional methods in both the monocular depth and semantic segmentation domain. But, their model can only generate coarse depth and semantic segmentation maps because the predictions are based on local regions with hand-crafted features. Similarly, Liu *et al.* [8] carried out these two tasks in a sequential manner where they first performed semantic segmentation and then used the predicted semantic labels to improve the depth estimation accuracy. Specifically, they used Markov Random Field (MRF) models for depth estimation, where a multi-class image labeling MRF predicts the semantic class for every pixel in the image and uses the predicted semantic labels as priors to estimate depth for each class. By incorporating semantic features, they achieved excellent results with a simpler model that can take into account the appearance and geometry constraints.

Other researchers [6, 12, 13] use deep learning techniques for joint monocular depth estimation and semantic segmentation from a single image to improve the performance of each task. These works [6, 12] performed depth estimation and semantic labeling using the supervised learning method while Ramirez *et al.* [13] used the semi-supervised learning method.

Wang *et al.* [6] and Mousavian *et al.* [12] used deep network architecture to simultaneously perform depth estimation and semantic segmentation and used a Conditional Random Field (CRF) to combine the depth and semantic information. Specifically, Wang *et al.* [6] proposed a two-layer Hierarchical Conditional Random Field (HCRF), which employs two convolutional neural networks (CNNs) to extract local and global features and then these features are enhanced using CRF. Their proposed approach enabled them to obtain promising results in both the monocular depth and semantic segmentation domain. On the other hand, Mousavian *et al.* [12] introduced a multi-scale CNN to perform depth estimation and semantic segmentation and combined them using a CRF. As shown in their work, the proposed model achieved comparable results on monocular depth estimation but outperformed the state-of-the-art methods on semantic segmentation. A more recent work by Ramirez *et al.* [13] proposed to solve the joint inference problem using a semi-supervised learning method where they employed a deep network architecture that can be jointly optimized for depth estimation and semantic segmentation where ground-truth semantic labels are required for the semantic segmentation task while geometry-based image reconstruction methods are employed instead of ground-truth depth labels for the depth estimation task. However, the experimental results reveal that their model, which was jointly trained for depth prediction and semantic segmentation, only improved the depth estimation accuracy. Their model failed to obtain better results for semantic segmentation.

This work addresses past design issues to obtain significant improvements when simultaneously performing depth estimation and semantic segmentation using rectified stereo image pairs with ground-truth semantic labels as training data. Specifically, to produce better depth estimates and semantic labeling, the proposed method involves changing the essential building blocks of the network architecture and introducing a disparity refinement module and a segmentation fusion module to generate better quality depth maps and semantic segmentation maps.

3 Proposed method

This section describes the proposed method for simultaneous depth estimation, depth refinement, and semantic segmentation in a semi-supervised manner using rectified stereo image pairs (I_L, I_R) with ground-truth semantic labels seg^{gt} as training data. Since the training data does not have ground-truth depth labels, the right images I_R together with the predicted disparities D_{L1} are used to obtain supervisory signals for the depth estimation task based on the underlying theory of epipolar constraints during training. In short, the supervisory signal is generated by warping one view of a stereo pair into the other view using the predicted disparity maps. Figure 1 presents the semi-supervised framework for joint monocular depth estimation and semantic segmentation using JDSNet. JDSNet is the proposed Siamese triple decoder architecture with a disparity refinement module and a segmentation fusion module.

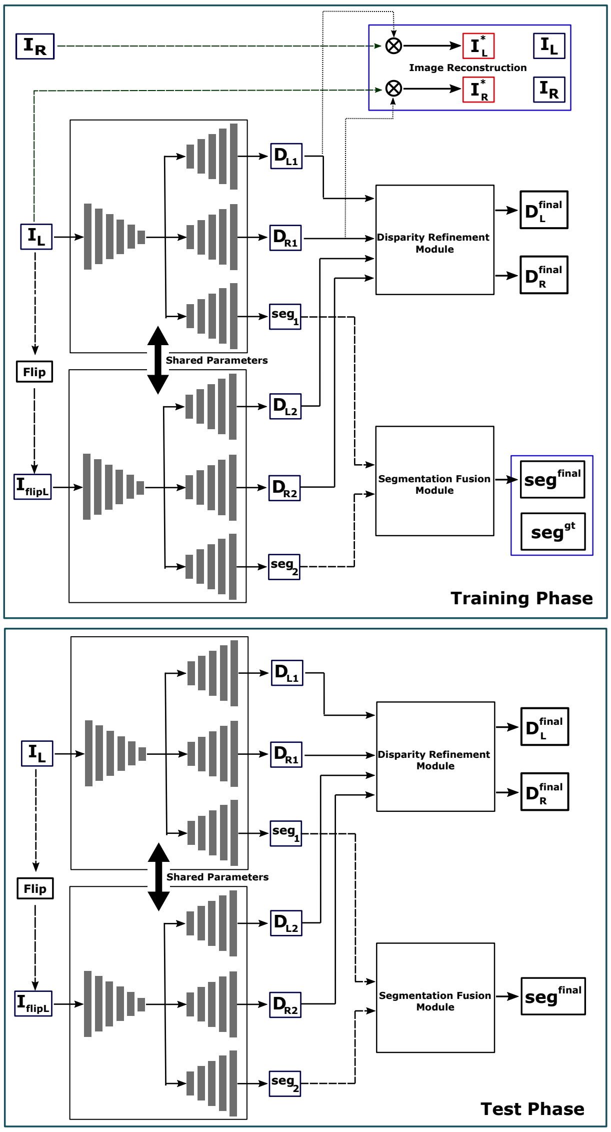

Figure 1: A semi-supervised framework for joint monocular depth estimation and semantic segmentation using JDSNet, the proposed Siamese triple decoder architecture.

3.1 Network architecture

The semi-supervised framework uses a Siamese architecture with the triple decoder network as the autoencoder. A Siamese architecture consists of two symmetrical structures and accepts two distinct images as inputs. An important feature of a Siamese architecture is that it uses two copies of the same network, and these two networks share weight parameters to process the two different inputs and generate two outputs. The original purpose of using a

Siamese architecture is for learning similarity representations, that is, to predict whether the two inputs are similar or not [15, 16]. However, in this study, the two outputs of the Siamese network are combined to produce the refined disparity maps through the disparity refinement module and a segmentation map through the segmentation fusion module.

JDSNet consists of two triple decoder networks that share weight parameters. It also has a disparity refinement module that enables the network to more effectively handle the visual artifacts and blurred boundaries while learning depth estimation. The disparity refinement module is the trainable version of the post-processing heuristic introduced by Godard *et al.* [14]. This module combines and refines the two pairs of depth maps. The segmentation fusion module combines the outputs from the two semantic segmentation decoders.

The Siamese triple decoder network receives the original left images I_L and the horizontally flipped version of the left input images I_{flipL} as inputs. With these images, the network is trained to predict depth maps, refine the predicted depth maps, and generate semantic segmentation maps.

The horizontally flipped version of the left input images I_{flipL} is necessary because in reconstructing the left images from the right images using the predicted disparities, there are pixels in the left images that are not present in the right images. Hence, no depth values can be predicted for these missing pixels. To overcome this limitation, the horizontally flipped version of the left input images I_{flipL} enables the network to predict the depth values of the occluded pixels, and by using the disparity refinement module, the predicted disparities from both inputs are combined to generate a refined disparity map.

A triple decoder network has a shared encoder and three parallel decoders that can be trained for depth estimation and semantic segmentation. The shared encoder is based on the encoder section of the AsiANet network architecture [17]. The encoded feature vectors are forwarded to the three parallel decoders: two depth decoders and one semantic segmentation decoder. The first depth decoder predicts the left disparity map and is constructed similar to the decoder section of AsiANet [17], while the second depth decoder that predicts the right disparity map and the semantic segmentation decoder are based on the ResNet50 decoders described in [13]. However, the last encoder block is modified due to hardware limitations where the number of output channels is reduced from 2048 to 1024. Also, unlike the previous works [13, 17], where a depth decoder generates two disparity maps when using rectified stereo image pairs as training data, each depth decoder in the proposed network generates a single disparity map.

The Siamese triple decoder network generates a pair of refined disparity maps $(D_L^{final},\ D_R^{final})$ at four different scales and a semantic segmentation map seg^{final} at full resolution only from the left image I_L. However, only the full resolution of the refined left disparity map D_L^{final} and semantic segmentation map are useful at test time.

3.1.1 Disparity refinement module

The disparity refinement module is based on the post-processing heuristic introduced by Godard *et al.* [14]. It is incorporated as a trainable component of the proposed Siamese triple decoder network rather than having a refinement step at test time since it decouples the refined disparity maps from the training. This design choice enables the network to simultaneously learn depth estimation and refine the predicted depth map in an end-to-end manner.

Essentially, the disparity refinement module performs three operations: horizontal flip operation, pixel-wise mean operation, and disparity ramps removal operation. The horizontal flip operation is performed on the disparity maps $(D_{L2},\ D_{R2})$ to generate $(D_{flipL},\ D_{flipR})$. Afterwards, the pixel-wise mean operation and the disparity ramps removal operation are performed on $(D_{L1},\ D_{flipL})$ and $(D_{R1},\ D_{flipR})$, respectively, to produce the refined disparity maps $(D_L^{final},\ D_R^{final})$.

3.1.2 Segmentation fusion module

The segmentation fusion module performs a horizontal flip operation on seg_2 to obtain seg_{flip}. It then adds the two layers seg_1 and seg_{flip} and forwards it to the softmax layer to output the probabilistic scores for each class and generate a semantic segmentation map seg^{final}.

3.2 Loss function

Training the proposed network relies on a loss function that can be expressed as a weighted sum of two losses, as defined in equation (1); a depth loss and a semantic segmentation loss, and the term is given by

$$L_{Total} = \alpha_{depth} L_{depth} + \alpha_{seg} L_{seg}, \tag{1}$$

where L_{depth} is the depth loss term, L_{seg} is the semantic segmentation loss term, and $\alpha_{depth},\ \alpha_{seg}$ are the loss weightings for each term.

3.2.1 Depth loss term

As defined in equation (2), L_{depth} is the sum of the depth losses at four different scales where L_s is the depth loss at each scale. L_s is a combination of three terms - appearance dissimilarity, disparity smoothness, and left-right consistency. This term is given by

$$\begin{aligned} L_{depth} &= \sum_{s=1}^{4} L_s, && (2) \\ L_s &= \alpha_{app} L_{app} + \alpha_{sm} L_{sm} + \alpha_{lr} L_{lr}, && (3) \\ L_{app} &= L_{app}^{left} + L_{app}^{right}, && (4) \\ L_{sm} &= L_{sm}^{left} + L_{sm}^{right}, && (5) \\ L_{lr} &= L_{lr}^{left} + L_{lr}^{right}, && (6) \end{aligned}$$

where L_{app} is the appearance dissimilarity term, L_{sm} is the edge-aware disparity smoothness term, L_{lr} is the left-right consistency term, and $\alpha_{app}, \alpha_{sm}, \alpha_{lr}$ are the loss weightings for each term. The depth loss term takes into account the left and right images where each component is in terms of the left images ($L^{left}_{app}, L^{left}_{sm}, L^{left}_{lr}$) and right images ($L^{right}_{app}, L^{right}_{sm}, L^{right}_{lr}$). However, this section provides details for the left components L^{left} only since the right components L^{right} are defined symmetrically.

The appearance dissimilarity term, as defined in (7), is a linear combination of the single-scale structural similarity (SSIM) term [18] and the L_1 photometric term. This term measures the quality of the synthesized target image by minimizing the pixel-level dissimilarity between the target image I and the synthesized target image I^*. This term is also widely used in previous studies [13, 14, 17] and it is given by

$$\begin{aligned} L^{left}_{app} = \frac{1}{N}\sum_{x,y}\omega\frac{1 - SSIM\left(I_L\left(x,y\right), I^*_L\left(x,y\right)\right)}{2} \\ + \left(1-\omega\right)\left\|I_L\left(x,y\right) - I^*_L\left(x,y\right)\right\| \end{aligned} \tag{7}$$

with a 3×3 box filter for the SSIM term and ω is set to 0.85 similar to [13, 14, 17]. The synthesized target left image I^*_L is obtained using a sampler from the spatial transformer network [19] that performs the bilinear interpolation. The sampler reconstructs the target left image I^*_L using the right image I_R and the predicted left disparity map D_{L1}.

The edge-aware disparity smoothness term, as defined in (8), regularizes the predicted disparities in spatially similar areas to ensure that the predicted disparities are locally smooth but can be sharp at the edges. This term is given by

$$\begin{aligned} {}^{left}_{sm} = & \frac{1}{N}\sum_{x,y}((|\partial_x D_{L2}(x,y)|e^{-|\partial_x I_{flipL}(x,y)|} \\ & + |\partial_y D_{L2}(x,y)|e^{-|\partial_y I_{flipL}(x,y)|}) \\ & + (|\partial_x D^{final}_L(x,y)|e^{-|\partial_x I_L(x,y)|} \\ & + |\partial_y D^{final}_L(x,y)|e^{-|\partial_y I_L(x,y)|})), \end{aligned} \tag{8}$$

where D^{final}_L is the refined left disparity map, D_{L2} is the second predicted left disparity map, and I_{flipL} is the horizontally flipped version of the left image I_L.

As described in [13, 14, 17], the left-right consistency term enforces consistency between the left and right disparities as defined in (9). This term is given by

$$\begin{aligned} L^{left}_{lr} = & \frac{1}{N}\sum_{x,y}|D^{final}_L(x,y) \\ & - (D_{R1}(x - D_{L1}(x,y), y))|, \end{aligned} \tag{9}$$

where D^{final}_L is the refined left disparity map, D_{R1} is the first predicted right disparity map, and D_{L1} is the first predicted left disparity map.

3.2.2 Semantic segmentation loss term

The semantic segmentation loss term, as defined in equation (10), is the standard cross-entropy loss between the predicted pixel-wise semantic labels seg^{final} and ground-truth pixel-wise semantic labels seg^{gt}. The semantic segmentation loss is computed using the left images only since these images have the corresponding ground-truth semantic labels at full image resolution. This term is given by

$$L_{seg} = -\sum_{i=1}^{N} P(seg^{gt}_i | seg^{final}_i), \tag{10}$$

where seg^{final}_i is the pixel-wise prediction for image I_i, seg^{gt}_i is the ground-truth semantic labels for image I_i, $P(y|x) = \sum_j p(y_j|x_j)$, and $p(y_j|x_j)$ is the probability of the ground-truth semantic label y_j at pixel j.

3.3 Datasets and evaluation metrics

Although the Cityscapes dataset [20] and KITTI dataset [21] contain a large number of training samples, the proposed semi-supervised learning framework for simultaneous depth estimation, depth refinement, and semantic segmentation requires rectified stereo image pairs with pixel-wise ground-truth semantic labels at training time. Hence, a subset of the Cityscapes dataset, which contains $2,975$ finely annotated images and the KITTI dataset consisting of 200 images with pixel-wise semantic ground-truth labels are used in this work.

Ramirez *et al.* [13] introduced a train/test split from the 200 images of the KITTI dataset for joint depth estimation and semantic segmentation. This dataset was split into 160 samples for the train set and 40 samples for the test set. The test set of 40 samples was used to quantitatively evaluate the proposed method given the distance range of 0-80 meters.

The standard evaluation metrics are used to evaluate the trained models quantitatively. The standard evaluation metrics for depth estimation measure the average errors, where lower values are better and accuracy scores where higher values are preferred [14, 22]. The six standard metrics for depth estimation are absolute relative difference (ARD), square relative difference (SRD), linear root mean square error (RMSE-linear), log root mean square error (RMSE-log), and the percentage of pixels (accuracy score) with thresholds (*t*) of 1.25, 1.25^2, and 1.25^3. These metrics are defined in Eq. (11) to Eq.(15).

$$ARD = \frac{1}{N}\sum\frac{|d^p_i - d^g_i|}{d^g_i} \tag{11}$$

$$SRD = \frac{1}{N}\sum\frac{||d^p_i - d^g_i||^2}{d^g_i} \tag{12}$$

$$RMSE - linear = \sqrt{\frac{1}{N}\sum||d^p_i - d^g_i||^2} \tag{13}$$

$$RMSE - log = \sqrt{\frac{1}{N}\sum||log(d^p_i) - log(d^g_i)||^2} \tag{14}$$

$$\delta < t = percent\,of\,d^p_i\ s.t.\ max\{\frac{d^p_i}{d^g_i}, \frac{d^g_i}{d^p_i}\} \tag{15}$$

d^g and d^p represent the ground-truth and estimated depth, respectively. N represents the number of pixels with valid depth value in the ground truth depth map.

On the other hand, the mean intersection over union (mIoU) is used to evaluate the semantic predictions of the model. It is the standard metric for segmentation tasks. The IoU measures the similarity between the intersection and union of the predicted pixel-wise semantic labels seg^{final} and ground-truth pixel-wise semantic labels seg^{gt}, and is calculated on a per-class basis and then averaged, as defined in (16). It is the ratio between the number of true positives (intersection) over the sum of true positives (TP), false positives (FP) and false negatives (FN) (union). This is given by

$$mIoU = \frac{1}{n_{cl}} \sum_{c} \frac{TP_c}{TP_c + FP_c + FN_c}, \tag{16}$$

where n_{cl} is the total number of classes and $c \in 0...n_{cl} - 1$.

Moreover, the pixel accuracy, as defined in (17), was also used to evaluate the performance of the model on the semantic segmentation task since the previous work [13] used this metric. This term is given by

$$pixel\ accuracy = \frac{1}{N} \sum_{c} TP_c, \tag{17}$$

where TP represents the true positives or correctly predicted pixels and N is the total number of annotated pixels.

4 Experiments

Tensorflow [23] was used to implement JDSNet. Training the network was performed on a single Nvidia GTX 1080 Ti GPU with 11 GB of memory. The training protocol was similar to [13, 14, 17] where the Adam optimizer [24] with $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-8}$ optimized the model for 50 epochs using the Cityscapes dataset and fine-tuned the model for another 50 epochs using the KITTI 2015 dataset by minimizing the training loss. For training and fine-tuning the model, the learning rate was initially set to $\lambda = 10^{-4}$ for the first 30 epochs and was reduced by half every 10 epoch until the process was completed. Moreover, the training phase involved using the same train/test split introduced in [13], resizing the input images to 256 by 512, using a batch size of 2, and performing data augmentation on the input images. The hyper-parameters have the following values: $\alpha_{depth} = 1.0, \alpha_{seg} = 0.1, \alpha_{app} = 1.0, \alpha_{lr} = 1.0$, and $\alpha_{sm} = 0.1/2^s$, where s is the down-sampling factor ranging from 0 to 3.

4.1 Results and discussion

This section discusses the results of the experiments conducted to evaluate the proposed method that simultaneously performs depth estimation, depth refinement, and semantic segmentation. The model was evaluated using the publicly available KITTI 2015 dataset [21] based on the test split introduced in [13]. Each test image has a corresponding ground-truth depth and semantic ground-truth labels.

The experiments involved training three different models:

1. Depth only model: $L_{Total} = L_{depth}$,

2. Semantic only model: $L_{Total} = L_{seg}$, and

3. Depth+Semantic model: Equation (1), which is the proposed training loss function.

In the depth only model, the semantic features are not considered during training. Hence, the model can only predict depth maps. In this setup, the two segmentation decoders and the segmentation fusion module are disabled. On the other hand, in the semantic only model, the depth features are not considered during training. Thus, the model can only generate semantic segmentation maps since the four depth decoders and the disparity refinement module are disabled. The main experiment involved training a depth+semantic model using the proposed method where both the semantic and depth features are considered during training.

Table 1 and Table 2 report the quantitative results. The experiment results were compared with the previous methods by directly using the results reported in [13]. These results reveal the effectiveness of the proposed method, which involved training the model to perform depth estimation, depth refinement, and semantic segmentation simultaneously.

As shown in Table 1, JDSNet is a better-suited model for depth estimation even when trained without any semantic information since it outperformed all previous models that were trained using both depth and semantic information based on the different evaluation metrics. The results also show further improvement when semantic information was considered in training JDSNet. Moreover, lower errors indicate that there are few outliers in the predicted depth maps.

A similar trend can be observed in Table 2, where JDSNet outperformed the previous models in terms of the semantic segmentation task when trained using both depth and semantic information. These results indicate that a good network design can significantly improve the performance of a model for both tasks, and including additional features during training can lead to better results. Specifically, simultaneously training the network for both tasks is more beneficial as the model can achieve better results than training a separate network for each task.

Although the results showed that the JDSNet-trained model using both depth and semantic information achieved high pixel accuracy rating, further validation was necessary since the pixel accuracy metric can be biased by imbalanced datasets. To overcome this limitation, the Jaccard index, also referred to as intersection-over-union, was employed. This evaluation metric takes into consideration

Method	Error Metric (Lower Is Better)				Accuracy Metric (Higher Is Better)		
	ARD	*SRD*	*RMSE (linear)*	*RMSE (log)*	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou *et al.* [25]	0.286	7.009	8.377	0.320	0.691	0.854	0.929
Mahjourian *et al.* [26]	0.235	2.857	7.202	0.302	0.710	0.866	0.935
GeoNet [27]	0.236	3.345	7.132	0.279	0.714	0.903	0.950
Godard *et al.* [14]	0.159	2.411	6.822	0.239	0.830	0.930	0.967
Ramirez *et al.* (ResNet50) [13]	0.143	2.161	6.526	0.222	0.850	0.939	0.972
Ramirez *et al.* (ResNet50+pp) [13]	0.136	1.872	6.127	0.210	0.854	0.945	0.976
Ours (JDSNet): Depth only	0.117	1.436	5.526	0.187	0.877	0.956	0.981
Ours (JDSNet): Depth+Semantic	**0.108**	**1.221**	**5.309**	**0.178**	**0.883**	**0.959**	**0.985**

Table 1: Monocular depth estimation results using the KITTI test split introduced in [13]. The **bold** values indicate the best results.

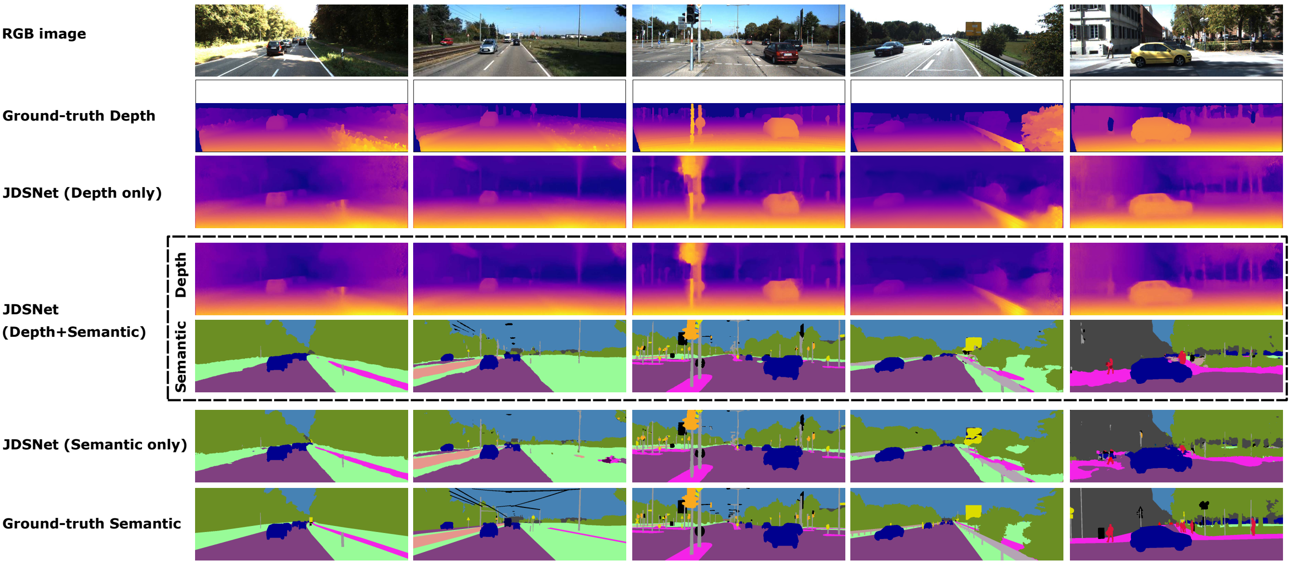

Figure 2: Qualitative results using the KITTI test split introduced in [13]. The ground-truth depth maps are interpolated for visualization purposes only. Best viewed in color.

Method	*PA*
Ramirez *et al.* (ResNet50) [13]: Semantic only	88.18%
Ramirez *et al.* (ResNet50) [13]: Depth+Semantic	88.19%
Ours (JDSNet): Semantic only	88.40%
Ours (JDSNet): Depth+Semantic	**89.57%**

Table 2: Semantic segmentation results using the KITTI test split introduced in [13]. PA means pixel accuracy. The **bold** values indicate the best results.

both the false positives and false negatives. Table 3 confirms that by incorporating depth information JDSNet performed better in the semantic segmentation task. For instance, when using both depth and semantic information, JDSNet was very effective in differentiating ambiguous pairs of classes, such as wall versus fence, sidewalk versus road, and motorcycle versus bicycle. It also achieved better results in terms of recognizing a person and segmenting distant objects and thin structures such as poles, traffic lights, and traffic signs.

The qualitative results, as shown in Figure 2, reveal that the proposed method generated depth maps that captures and preserves the general scene layout where thin structures are perceivable. Also, the disparity refinement module achieved a similar result to the post-processing heuristic that is performed during testing where the refined depth maps have no border artifacts on the image boundary. In addition, the results show that JDSNet can effectively perform semantic segmentation, as evidenced by its ability to capture the geometrical characteristics of the objects in the scene. For example, JDSNet was able to segment the traffic light in the third image even if it has a thin structure and an irregular shape.

5 Conclusion

This work has introduced a semi-supervised learning framework that simultaneously performs depth estimation, depth refinement, and semantic segmentation using rectified stereo image pairs with ground-truth semantic labels during training. The proposed architecture, referred to as

Method	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic light	Traffic sign	Vegetation	Terrain
JDSNet: Semantic only	90.77	47.13	72.86	16.71	11.70	31.73	13.96	19.79	86.38	**74.21**
JDSNet: Depth+Semantic model	**91.43**	**52.98**	**78.81**	**34.82**	**28.93**	**36.72**	**14.05**	**26.45**	**86.67**	74.08

Method	Sky	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle	mIoU
JDSNet: Semantic only	92.83	7.68	**3.09**	81.82	5.18	**0.00**	**5.04**	0.31	13.88	35.53%
JDSNet: Depth+Semantic model	**93.37**	**16.87**	0.74	**85.65**	**5.23**	**0.00**	1.51	**1.53**	**16.85**	**39.30%**

Table 3: Semantic segmentation results using the KITTI test split introduced in [13]. mIoU means mean intersection over union. The **bold** values indicate the best results.

JDSNet, is a Siamese triple decoder network architecture with a disparity refinement module and a segmentation fusion module that is capable of improving on the performance of both tasks by sharing the underlying features representations and utilizing both geometric and semantic information. Experiment results show that the proposed method achieved promising results on both depth estimation and semantic segmentation and outperformed previous methods.

References

[1] L. Chen, Z. Yang, J. Ma, and Z. Luo (2018) Driving Scene Perception Network: Real-time Joint Detection, Depth Estimation and Semantic Segmentation, *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, IEEE, pp. 1283–1291. https://doi.org/10.1109/WACV.2018.00145

[2] G. Giannone and B. Chidlovskii (2019) Learning Common Representation from RGB and Depth Images, *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, pp. 408–415. https://doi.org/10.1109/cvprw.2019.00054

[3] R. Cipolla, Y. Gal and A. Kendall (2018) Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 7482–7491. https://doi.org/10.1109/CVPR.2018.00781

[4] J. Liu, Y. Wang, Y. Li, J. Fu, J. Li, and H. Lu (2018) Collaborative Deconvolutional Neural Networks for Joint Depth Estimation and Semantic Segmentation, *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, vol. 29, no. 11, pp. 5655–5666. https://doi.org/10.1109/TNNLS.2017.2787781

[5] D. Sanchez-Escobedo, X. Lin, J. R. Casas, and M. Pardas (2018) Hybridnet for Depth Estimation and Semantic Segmentation, *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, pp. 1563–1567. https://doi.org/10.1109/ICASSP.2018.8462433

[6] Peng Wang, Xiaohui Shen, Zhe Lin, S. Cohen, B. Price, and A. Yuille (2015) Towards unified depth and semantic prediction from a single image, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 2800–2809. https://doi.org/10.1109/CVPR.2015.7298897

[7] L. Ladicky, J. Shi, and M. Pollefeys (2014) Pulling things out of perspective, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 89–96. https://doi.org/10.1109/CVPR.2014.19

[8] B. Liu, S. Gould, and D. Koller (2010) Single image depth estimation from predicted semantic labels, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 1253–1260. https://doi.org/10.1109/CVPR.2010.5539823

[9] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers (2016) Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture, *Proceedings of the Asian Conference on Computer Vision*, Springer, pp. 213–228. https://doi.org/10.1007/978-3-319-54181-5_14

[10] O. H. Jafari, O. Groth, A. Kirillov, M. Y. Yang, and C. Rother (2017) Analyzing modular CNN architectures for joint depth prediction and semantic segmentation, *Proceedings of the 2017 International Conference on Robotics and Automation*, IEEE, pp. 4620–4627. https://doi.org/10.1109/ICRA.2017.7989537

[11] V. Nekrasov, T. Dharmasiri, A. Spek, T. Drummond, C. Shen and I. Reid (2019) Real-Time Joint Semantic Segmentation and Depth Estimation Using Asymmetric Annotations, *Proceedings of the 2019 International Conference on Robotics and Automation*, IEEE, pp. 7101–7107. https://doi.org/10.1109/ICRA.2019.8794220

[12] A. Mousavian, H. Pirsiavash, and J. Košecká (2019) Joint Semantic Segmentation and Depth Estimation with Deep Convolutional Networks, *Proceedings of the 2016 Fourth International Conference on 3D Vision*, IEEE, pp. 611–619. https://doi.org/10.1109/3DV.2016.69

[13] P. Z. Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano (2018) Geometry meets semantic for semi-supervised monocular depth estimation, *Proceedings of the 14th Asian Conference on Computer Vision*, Springer, pp. 611–619. `https://doi.org/10.1007/978-3-030-20893-6_19`

[14] C. Godard, O. M. Aodha and G. J. Brostow (2017) Unsupervised Monocular Depth Estimation with Left-Right Consistency, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 6602–6611. `https://doi.org/10.1109/CVPR.2017.699`

[15] J. Bromley, I. Guyon, Y. LeCun, E. Sackinger, and R. Shah (1994) Signature verification using a siamese time delay neural network, *Proceedings of the Annual Conference on Neural Information Processing Systems*, pp. 737–744. `https://doi.org/10.1142/9789812797926_0003`

[16] G. Koch, R. Zemel, and R. Salakhutdinov (2015) Siamese neural networks for one-shot image recognition, *Proceedings of International Conference on Machine Learning*.

[17] J. P. Yusiong and P. Naval (2019) AsiANet: Autoencoders in Autoencoder for Unsupervised Monocular Depth Estimation, *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, IEEE, pp. 443–451. `https://doi.org/10.1109/WACV.2019.00053`

[18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli (2004) Image quality assessment: from error measurement to structural similarity, *IEEE Transactions on Image Processing*, IEEE, vol. 13, no. 4, pp. 600–612. `https://doi.org/10.1109/tip.2003.819861`

[19] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu (2015) Spatial transformer networks, *Proceedings of the Annual Conference on Neural Information Processing Systems*, pp. 2017–2025.

[20] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele (2016) The cityscapes dataset for semantic urban scene understanding, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 3213–3223. `https://doi.org/10.1109/CVPR.2016.350`

[21] Geiger, P. Lenz, and R. Urtasun (2012) Are we ready for autonomous driving? The kitti vision benchmark suite, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 3354–3361. `https://doi.org/10.1109/CVPR.2012.6248074`

[22] D. Eigen, C. Puhrsch and R. Fergus (2014) Depth map prediction from a single image using a multi-scale deep network, *Proceedings of the Annual Conference on Neural Information Processing Systems*, pp. 2366–2374.

[23] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.* (2016) Tensorflow: a system for large-scale machine learning, *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, USENIX Association, pp. 265–283.

[24] D. Kingma and J. Ba (2015) Adam: A method for stochastic optimization, *Proceedings of the International Conference on Learning Representations*.

[25] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe (2017) Unsupervised learning of depth and ego-motion from video, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 6612–6619. `https://doi.org/10.1109/CVPR.2017.700`

[26] R. Mahjourian, M. Wicke, and A. Angelova (2018) Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 5667–5675. `https://doi.org/10.1109/CVPR.2018.00594`

[27] Z. Yin and J. Shi (2018) GeoNet: Unsupervised learning of dense depth, optical flow and camera pose, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, pp. 1983–1992. `https://doi.org/10.1109/CVPR.2018.00212`