Interdependent Multi-task Learning for Simultaneous Segmentation and Detection

Mahesh Reginthala[1], Yuji Iwahori[2], M. K. Bhuyan[1], Yoshitsugu Hayashi[2], Witsarut Achariyaviriya[2] and Boonserm Kijsirikul[3]

[1]*Indian Institute of Technology Guwahati, 781039, India*

[2]*Chubu University, 487-8501, Japan*

[3]*Chulalongkorn University, Bangkok, 20330, Thailand*

Keywords: Multi-task Learning, Semantic Segmentation, Object Detection, Deep Learning.

Abstract: Lightweight, fast, and accurate deep-learning algorithms are essential for practical deployment in real-world use-cases. Semantic segmentation and object detection are the principal tasks of visual perception. A multi-task network significantly reduces the number of parameters compared to two independent networks running simultaneously for each task. Generally, multi-task networks have shared encoders and multiple independent task-specific decoders. Instead, we modeled our network to exploit the features from both encoder and decoder. We propose the multi-task network that performs both segmentation and detection with only 37.9 million parameters and inference time of 74 milliseconds on a consumer-grade GPU. This network performs two tasks with much fewer parameters and in much less inference time compared to each single task network.

1 INTRODUCTION

Convolutional neural networks (CNNs) have been remarkably successful in the field of computer vision over recent years. Visual perception is a crucial part of several upcoming breakthroughs in technologies like self-driving, robotics, health care, automation, and artificial intelligence. Semantic segmentation and object detection constitute a significant part of Visual perception. Semantic segmentation is required to understand the areal classes like road, vegetation, and sky. Whereas object detection helps us to understand countable classes like vehicles and humans. Enormous computational complexity and high inference times have been significant setbacks of these intensive tasks. Most of the real-world visual perception tasks necessitate both these tasks to be performed simultaneously on critical resource-constrained platforms.

It is evident that the initial layers of any encoder of a computer vision task have similar filters [to Gabor ones] independent of the task and decoder of a semantic segmentation network have all the pixel-level contextual information which is very helpful for an object detector to extract bounding boxes and class probabilities from those representations. This motivates us to build a single Multi-task learning model capable of performing the complex tasks of semantic segmentation and object detection simultaneously. Along with accuracy, we also concentrate on making our network light and the fast inference on a consumer-grade GPU.

In summary, the following are our important contributions:

- We present a novel MTL network that performs both semantic segmentation and object detection simultaneously with inference time and the number of parameters much less than a semantic segmentation or an object detection single task network.
- Usually, MTL networks have shared encoders and independent task-specific decoders; instead, we exploit the feature maps of segmentation decoder with rich semantic information. Thus, proposing a framework for an MTL network with interconnected decoders.
- We propose a scale aware training scheme for the trident block of a one-stage object detector with anchor boxes.
- We propose the training procedure for a highly interdependent MTL network.

Reginthala, M., Iwahori, Y., Bhuyan, M., Hayashi, Y., Achariyaviriya, W. and Kijsirikul, B.
Interdependent Multi-task Learning for Simultaneous Segmentation and Detection.
DOI: 10.5220/0008949501670174
In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2020)*, pages 167-174
ISBN: 978-989-758-397-1; ISSN: 2184-4313

2 RELATED WORKS

In general, Multi-task learning networks are categorized into hard parameter sharing and soft parameter sharing methods. Usually in hard parameter sharing, several task-specific decoders are used to make predictions by using a feature map generated from a single encoder similar to (Caruana, 1993) (Teichmann et al., 2016). This method is unlikely to overfit and found to be very good at generalization. Hard parameter sharing is widely used in MTL because of its computational advantages. In soft parameter sharing, every task will have its task-specific model with some degree of sharing parameters in between different models, for example (Misra et al., 2016). The similarity of parameters are improved by regularizing the distance between the parameters of models, as shown in (Duong et al., 2015).

2.1 Semantic Segmentation

Over these years, it is evident that CNNs are very good at semantic segmentation and classification tasks. For pixel-level prediction tasks like semantic segmentation, fully convolutional networks (FCNs) by (Long et al., 2015) introduced the end-to-end approach that maps the feature maps of a classification network to a dense prediction output. FCN (Long et al., 2015) modified VGG-16 (Simonyan and Zisserman, 2014) into an encoder-decoder architecture with skip connections. Conditional random fields (CRFs) are used on network output for better performance around object boundaries in DeepLab (Chen et al., 2016). For better performance across different scales of object's instances was initially achieved by training network at multiple rescaled versions or by fusing features from multiple parallel branches that take different image resolutions as shown in (Farabet et al., 2012) and (Long et al., 2015) respectively. These networks use pooling layers to increase the receptive field. Which are inefficient for a segmentation network. DeepLab (Chen et al., 2016) and PSPNet (Zhao et al., 2017) use dilated convolutions of different rates over multiple parallel branches and concatenates them. This enables them to increase the field-of-view of CNNs for multi-scale contextual information. The major setback of these approaches is computational complexity and hence, large inference time. Adapnet++ (Valada et al., 2019) addresses this issue by Cascading atrous convolutions to enlarge field-of-view efficiently. Several recent works like DANet (Fu et al., 2019) and GFF (Li et al., 2019a) use Gating and Attention mechanisms frequently used in recurrent networks like LSTM and GRU in various ways for tasks like semantic segmentation.

2.2 Object Detection

Object detectors based on deep learning can be classified into two categories one-stage detectors and two-stage detectors. It is commonly observed that two-stage detectors are good at accuracy, whereas one-stage detectors have faster inference times. Usually in two-stage detectors, the first stage proposes some regions of the image that are the potential to have an object. R-CNN (Girshick et al., 2014) uses Selective Search (Uijlings et al., 2013), SPPNet (He et al., 2015) uses spatial pyramid pooling, Fast R-CNN (Girshick, 2015) uses RoIPooling layers, Faster R-CNN (Ren et al., 2015) uses region proposal network (RPN) as their first stage. These proposed regions are forwarded to the second stage for refining the detected boundaries and classifying the object. For the second stage, R-CNN (Girshick et al., 2014) uses class-specific linear SVMs over the fixed-length feature vector of the warped region generated using a large convolutional neural network (CNN), Fast R-CNN (Girshick, 2015) uses fixed-size feature map mapped to a feature vector by fully connected layers (FCs), Faster R-CNN (Ren et al., 2015) uses Fast R-CNN detector module. On the other hand, one-stage methods YOLO (Redmon and Farhadi, 2017) and SSD (Liu et al., 2016) frames object detection as an optimized end-to-end regression problem to offsets of the predefined anchor boxes and class probabilities. Focal loss (Lin et al., 2018) addresses the issue of class imbalance common in one-stage detectors.

Large scale variations of object instances is a vital issue for object detectors. The following are different methods used to handle large scale variations, SSD (Liu et al., 2016) and MS-CNN (Cai et al., 2016) uses feature maps at different levels for making predictions at multiple scales. TDM (Shrivastava et al., 2016) and FPN (Lin et al., 2017) uses top-down pathway and lateral connections for more semantic representation and make predictions at multiple levels of the decoder. PANet (Liu et al., 2018) enhance the information flow between lower layers and topmost feature by bottom-up path augmentation. SNIP (Lee et al., 2018) selectively back-propagates the gradients of object instances of different sizes as a function of the image scale. M2Det (Zhao et al., 2019) uses alternating joint Thinned U-shape Modules and Feature Fusion Modules to extract more representative, multi-level multi-scale features. TridentNet (Li et al., 2019b) has a parallel multi-branch architecture in which each branch has dilated convolutions at different rates but with the same transformation parameters, which enables it to perform better over different scales.

(a) (b) (c)

Figure 1: These are the observations when the following modifications are performed on a U-Net like semantic segmentation network. a) Original prediction of semantic segmentation network. b) Prediction without skip connections, object boundaries are smoothened, and small objects like poles, distant persons are lost. This shows the importance of skip connections and feature maps from the top layers. c) Prediction with bottleneck activations and skip connections inverted. This shows the spatial correlation of feature maps at different levels of Deep CNNs.

3 PROPOSED NETWORK

In this section, we describe the overall architecture of the proposed network, which can perform simultaneous object detection and semantic segmentation. We detail our design criteria, reasons, and advantages of the proposed multi-tasking network. We then detail the Trident block for one-stage detectors with anchor boxes.

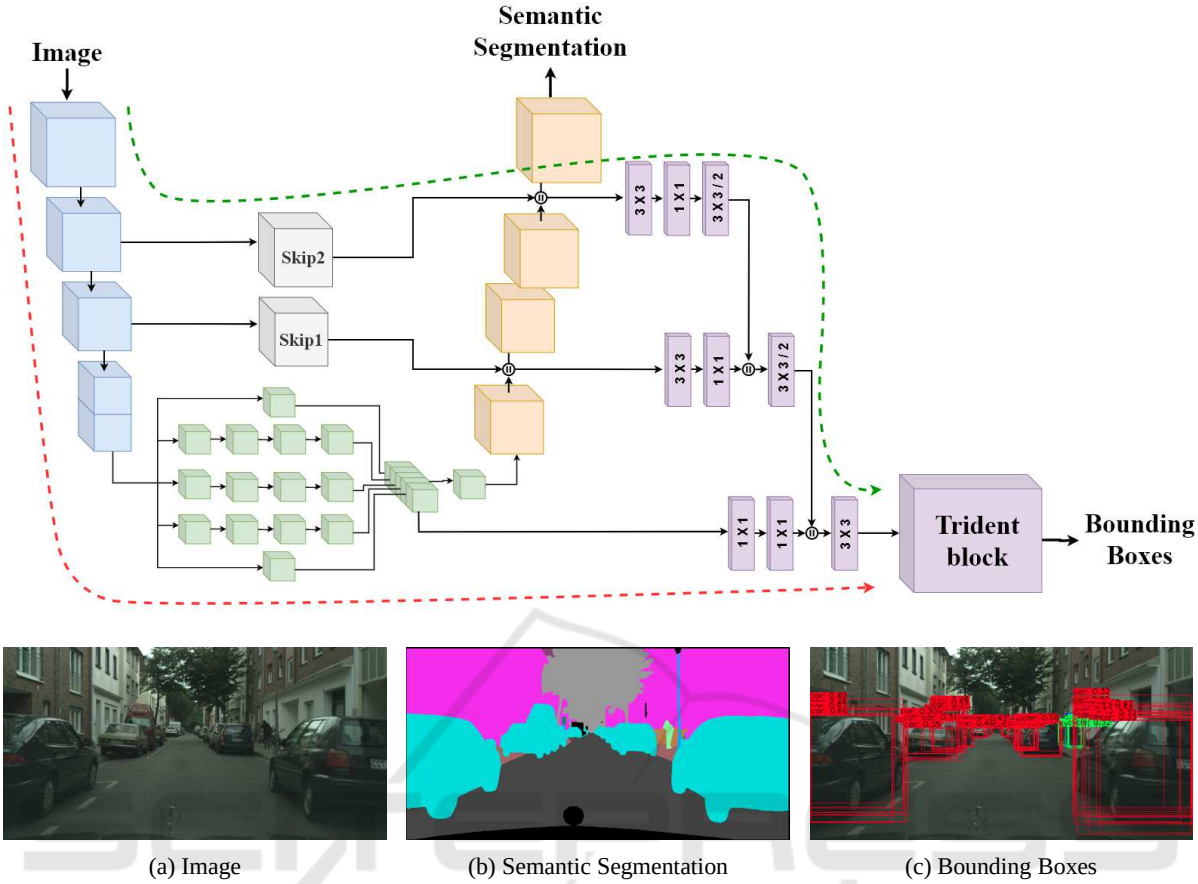
3.1 Semantic Segmentation

Our network is a fully convolutional image encoder - segmentation decoder - object detector design, as shown in the Figure 2. We adopt AdapNet++ (Valada et al., 2019) Architecture for the semantic segmentation part. AdapNet++ (Valada et al., 2019) is a computationally efficient semantic segmentation architecture with the Image encoder based on full preactivation ResNet-50 (He et al., 2016) with multiscale residual units at varying dilation rates in the last two blocks of the encoder. ResNet-50 (He et al., 2016) accommodates the sufficient deep contextual features in limited computational complexity. AdapNet++ (Valada et al., 2019) poses an efficient atrous spatial pyramid pooling (eASPP) module as the bottleneck of the network, which reduces the number of parameters required by over 87% compared to the originally proposed ASPP in DeepLabv3+ (Chen et al., 2018b). Its decoder consists of multiple deconvolutions and convolution layers with skip connections fusing feature maps from the encoder for the segmentation of small objects and object boundary refinement. Two auxiliary losses are used immediately after each upsampling stage to accelerate training and improve the gradient propagation in the network. These auxiliary losses also help us for coming forth object detector.

3.2 Object Detection

In object detection tasks, we need to do both localization and classification. The primary issue is the spatial information required for the localization is abundant at the top layers of the Deep CNNs, where bottom layers have rich contextual information, which is crucial for the classification of objects. On the other hand, we have large scale variations of object instances where smaller and simple instances are only found in top layers, and large and complex instances are found in the bottom layers, see Figure 1. Our proposed network addresses these issues by ensuring that feature maps from both the top and bottom layers are readily available for the object detection module.

We have two auxiliary softmax losses each after the first two up-convolutions in the segmentation decoder, which enforces the feature maps at that intermediate level to be more spatially similar to the ground truth labels. This feature maps concatenated with the feature maps from the skip refinement stage is forwarded to the object detection network, as shown in the Figure 2. We model our object detection network as a one-stage object detector similar to YOLO (Redmon and Farhadi, 2017) because of its simplicity and fast inference time compared to two-stage object detectors. In the object detection network, these feature maps are rapidly downsampled to the bottleneck resolution and concatenated with the feature maps from the bottleneck. We use only 3 X 3 and 1 X 1 convolutions in this rapid downsampling to preserve smaller and simple instance's information present in the top layers and also we use 3 X 3/2 convolution instead of Maxpool to preserve the spatial relativity. This rapid downsampling shortens the information path between lower layers and the topmost feature maps. Several Multi-Task Learning (MTL networks) (Teichmann et al., 2016), (Sistu et al., 2019) have shared encoders and multiple independent task-specific decoders. Instead, we try to exploit the rich semantic information in the segmen-



(a) Image (b) Semantic Segmentation (c) Bounding Boxes

Figure 2: Overview of our proposed MTL network. Encoder based on ResNet 50 (He et al., 2016) is depicted in blue, the bottleneck efficient atrous spatial pyramid pooling (eASPP) is depicted in green, The orange color part denotes the segmentation decoder, grey color blocks are the skip refinement stages and the plum color part represents the object detector.

tation decoder, which is highly useful for object detection. As we discussed before, high-resolution top layers are essential for object localization, but top layers of the encoder are low-level, fine-grained, shallow feature maps whereas top layers of the segmentation decoder are semantic, coarse-grained, deep feature maps which are more helpful for object detection. We boost the information flow similar to Path Aggregation Network (PANet) (Liu et al., 2018), as shown in the Figure 2 the dashed green line indicates the shortcut for low-level feature maps from top layers that are effective for accurate localization of instances whereas, the dashed red line goes through the whole encoder carries the deep-level feature maps that are valuable for classifying instances. Also, we are forwarding Multi-Level Features to the object detector by concatenating feature maps from both segmentation decoder and skip refinement stage.

Datasets like Cityscapes (Cordts et al., 2016) have large scale variations of classes like Cars and Person varies largely by the distance between the object and the camera. It is observed that some vehicle instances occupy a great part of the image. We use trident block originally proposed in TridentNet (Li et al., 2019b) for two-stage object detectors to handle this scale variation. It consists of multiple parallel branches each a stack of residual blocks with three convolutions of kernel size 1 X 1, 3 X 3 and 1 X 1 but each branch of trident block has different dilation rates for the 3 X 3 convolution as shown in the Figure 3. Weights are shared among these branches as it reduces the number of parameters, and it intends that the same transformation is applied at different spatial scales. In our experiments for Each branch of Trident block, we have 3 anchor boxes so the tensor is of shape 24 X 48 X [3 * (4 + 1 + number of classes)] for 4 bounding box offsets and 1 for object confidence prediction similar to YOLO (Redmon and Farhadi, 2017). 24 X 48 is the resolution of the bottleneck feature map. We use leaky rectified linear activation function in the object detection part.

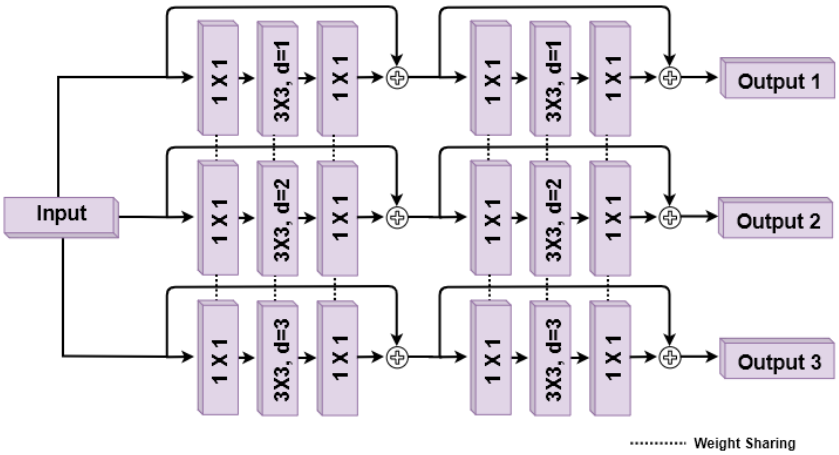


Figure 3: Illustration of the Trident block depicted in Figure 2. Trident block was initially proposed in TridentNet (Li et al., 2019b).

4 EXPERIMENTS AND RESULTS

In this section, we first give details about the dataset selection and dataset generation. Then, we describe our proposed scale-aware training of trident blocks for one-stage networks, and we also describe our proposed training procedure for an interdependent multi-tasking network like this. Finally, we detail the evaluation procedure and results.

4.1 Dataset

We need a dataset with both segmentation labels and bounding boxes for object detection. So, we trained and evaluated on publicly available Cityscapes dataset (Cordts et al., 2016). It is a very popular and highly challenging dataset containing images of complex urban scenes. We used only the provided 2875 finely annotated images for training, 500 are for validation. Bounding boxes are not directly provided in the original Cityscapes (Cordts et al., 2016) dataset, but instance-level annotations are provided. We modified the scripts given by Cityscapes (Cordts et al., 2016) dataset to extract bounding boxes from instance-level annotations for the classes Car/Truck/Bus, Person/Rider,Bicycle/Motorcycle bicycle. In our work, we first resized the images in the dataset to resolution 768 x 384. Then, we do data argumentation by randomly scaling (1 to 1.5), and we take random crops of resolution 768 x 384 followed by random horizontal flipping of the cropped image. We use Bilinear Interpolation and Nearest Neighbour Interpolation for rescaling images and segmentation labels, respectively. We ignored the bounding box if its center falls out of the randomly cropped image.

4.2 Training

For calculating segmentation loss (L_{seg}), we use the cross-entropy loss function for both main loss (L_{main}) and auxiliary losses (L_{aux1}, L_{aux2}). We use bilinear upsampling for the feature maps at each auxiliary loss branch to match the resolution with segmentation ground truth labels. We set the weights to $\lambda_1 = 0.6$ and $\lambda_2 = 0.5$.

$$L_{seg} = L_{main} + \lambda_1 * L_{aux1} + \lambda_2 * L_{aux2}. \quad (1)$$

In our work, We use three branches in the Trident block at dilation rates 1, 2, 3 for small instances, medium instances, large instances, respectively. We use three anchor boxes for each branch. Here, we propose a scale-aware training scheme for one-stage detectors to enhance scale awareness of every branch. We use k-means clustering on the training data set to determine the required nine anchor boxes similar to YOLO (Redmon and Farhadi, 2017). We sort those anchor boxes according to their area. Then we allot the three largest anchor boxes for the branch specialized for large instances with dilation rate 3. Similarly, we allot the three smallest anchor boxes for the branch specialized for smaller instances with dilation rate 1. Remaining three anchor boxes in the middle are allotted for the branch specialized for medium scale instances with dilation rate 2. We use focal loss (Lin et al., 2018) to calculate object detection loss (L_{det}). We use a weighted sum of individual losses for the two tasks to train this multi-tasking network.

$$L = W_{seg} * L_{seg} + W_{det} * L_{det} \quad (2)$$

$$(W_{seg} = 30 - 50, W_{det} = 1) \quad (3)$$

Table 1: Semantic segmentation and object detection results of the proposed network trained on the Cityscapes dataset (Cordts et al., 2016).

Metrics	Train	Validation
Road IoU	99.26	97.84
Traffic Sign IoU	84.07	69.84
Pedestrians IoU	87.87	74.25
Building IoU	96.36	90.91
Sky IoU	96.18	93.28
Vegetation IoU	95.73	91.66
Pole IoU	72.65	56.86
SideWalk IoU	95.34	83.91
Fence IoU	91.89	54.48
Rider/Cycle IoU	87.29	71.77
Car/Truck/bus IoU	96.60	93.07
Mean IoU (mIoU)	91.20	79.81
AP Car/Truck/Bus	71.92	64.17
AP Person/Rider	60.27	50.00
AP Bicycle/Motorcycle	57.86	37.10
Mean AP (mAP)	63.35	50.42

Table 2: Comparisons of semantic segmentation and object detection results with other MTL network evaluated on the Cityscapes dataset (Cordts et al., 2016).

Network	Segmentation (mIoU)	Detection (mAP)
Real-time Joint Object Detection and Semantic Segmentation Network for Automated Driving (Sistu et al., 2019)	55.55	23.55
Our proposed MTL network	79.81	50.42

Here, we propose the procedure for training a highly interdependent multi-tasking network that avoids plateau regions, longer training periods, imbalanced training of two tasks, and overfitting. We initialized the encoder of the network with the weights pre-trained on the ImageNet dataset (Deng et al., 2009), and He initialization (He et al., 2015) is used for initializing rest of the network. We use Adam optimizer with hyperparameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-10}$. We use dropout layers with probability 0.5 in block4 of the encoder and just before trident block. We use polynomial decay of the Learning rate with cyclic restarts for every 10K iterations.

1. First, we train the whole network for 30K iterations using an initial learning rate = 0.001 to stabilize the network from producing asymptotic numbers.
2. Secondly, We freeze object detector, and we train only encoder and segmentation decoder for 120k iterations with the initial learning rate = 0.001 and weight of detection loss function $W_{det} = 0$, $W_{seg} = 30 - 50$. In our network, object detector extracts feature maps segmentation decoder. So first, if the segmentation decoder is trained well, then It will give good feature maps, and the training of the object detector will be smooth later. Otherwise, object detector is taking many steps for training that leads to overfitting of the encoder.
3. Then, We freeze encoder, segmentation decoder, and we train the object detector for 120k iterations with the initial learning rate = 0.0001 and weight of segmentation loss function $W_{seg} = 0$, $W_{det} = 1$. Freezing segmentation decoder ensures that the additional object detection will not intervene in its segmentation task or any loss of segmentation accuracy.
4. Finally, we train the whole network for another 50k iterations with the learning rate starting from 0.0001.

4.3 Evaluation

We implemented our proposed multi-tasking network using TensorFlow (Abadi et al., 2015) deep learning library. We carried out the experiments on a system with one consumer-grade NVIDIA GeForce GTX 1080 Ti GPU. Per-class IoU and mean class IoU (Intersection over Union) were used as accuracy metrics for semantic segmentation, per-class average preci-

Table 3: Comparison Study of Single task segmentation networks (DPC (Chen et al., 2018a), DeepLabv3+ (Chen et al., 2018b), PSPNet (Zhao et al., 2017), HRNetV2-W48 (Wang et al., 2019), Mapillary (Bul et al., 2018)) vs Our proposed Multi-task network.

Network	Params(M)	mIoU
DPC	41.8	80.9
DeepLabv3+	43.5	79.6
PSPNet	56.3	80.9
HRNetV2-W48 (SOTA)	65.9	81.1
Mapillary	135.9	78.3
Our proposed MTL network	37.9	79.8

sion, and mean average precision (mAP) are the metrics used for evaluation. During inference, we use Non-maximal suppression with IoU threshold 0.55 to handle multiple object detections.

In Table 1, we summarize the results of our proposed MTL network. We trained our network on the Cityscapes dataset with 11 semantic segmentation classes and three object detection classes. We achieved semantic segmentation Mean Class Intersection over Union (mIoU) of 79.81 and object detection score of 50.42 Mean Average Precision (mAP) with only 37.9M parameters and inference time of 74ms on a consumer-grade GPU. Table 2 shows the accuracy leap compared to that recently proposed MTL network for joint object detection and semantic segmentation evaluated on the cityscapes dataset. Table 3 shows the computational effectiveness of our proposed MTL network. We are performing two tasks with the only 37.9M parameters, which are considerably less compared to other single task segmentation networks itself. This shows that our proposed network is the best and most accurate MTL network for simultaneous semantic segmentation and object detection with the right trade-off of performance, productivity, and computational complexity. It is making our proposed network as the most efficient way to perform visual perception tasks on resource-constrained environments and with faster inference times.

5 CONCLUSIONS

In this paper, we discussed the importance of computational lightness and quickness of visual perception algorithms. We examined the relationships between different layers and different features. Then, we proposed a multi-task learning framework for simultaneous semantic segmentation and object detection. We focused on exploiting the helpful feature maps from the decoder. We proposed a training scheme for interdependent MTL networks. We centered on designing a computationally efficient network to deploy on resource-constrained platforms. We evaluated and shared the results of the proposed network on the cityscapes dataset (Cordts et al., 2016). We discussed the effectiveness of our proposed network compared to other MTL network and single task networks.

ACKNOWLEDGEMENTS

This research is supported by SATREPS Project of JST and JICA: Smart Transport Strategy for Thailand 4.0 Realizing better quality of life and low-carbon society, by Japan Society for the Promotion of Science (JSPS) Grantin-Aid for Scientific Research (C) (17K00252) and by Chubu University Grant.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Bul, S. R., Porzi, L., and Kontschieder, P. (2018). In-place activated batchnorm for memory-optimized training of dnns. In *CVPR*, pages 5639–5647. IEEE Computer Society.

Cai, Z., Fan, Q., Feris, R. S., and Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *ECCV (4)*, volume 9908 of *Lecture Notes in Computer Science*, pages 354–370. Springer.

Caruana, R. (1993). Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 41–48. Morgan Kaufmann.

Chen, L.-C., Collins, M. D., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., and Shlens, J. (2018a). Searching for efficient multi-scale architectures for dense image prediction. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *NeurIPS*, pages 8713–8724.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018b). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., editors, *ECCV (7)*, volume 11211 of *Lecture Notes in Computer Science*, pages 833–851. Springer.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Duong, L., Cohn, T., Bird, S., and Cook, P. (2015). Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. pages 845–850, Beijing, China. Association for Computational Linguistics.

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2012). Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *ICML*. icml.cc / Omnipress.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., and Lu, H. (2019). Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154. Computer Vision Foundation / IEEE.

Girshick, R. (2015). Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 00, pages 580–587.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *IEEE International Conference on Computer Vision (ICCV 2015)*, 1502.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity Mappings in Deep Residual Networks. *arXiv e-prints*, page arXiv:1603.05027.

Lee, N., Ajanthan, T., and Torr, P. H. S. (2018). SNIP: Single-shot Network Pruning based on Connection Sensitivity. *arXiv e-prints*, page arXiv:1810.02340.

Li, X., Zhao, H., Han, L., Tong, Y., and Yang, K. (2019a). GFF: Gated Fully Fusion for Semantic Segmentation. *arXiv e-prints*, page arXiv:1904.01803.

Li, Y., Chen, Y., Wang, N., and Zhang, Z. (2019b). Scale-Aware Trident Networks for Object Detection. *arXiv e-prints*, page arXiv:1901.01892.

Lin, T.-Y., Dollr, P., Girshick, R. B., He, K., Hariharan, B., and Belongie, S. J. (2017). Feature pyramid networks for object detection. In *CVPR*, pages 936–944. IEEE Computer Society.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollar, P. (2018). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1.

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768. IEEE Computer Society.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. (2016). Ssd: Single shot multibox detector. volume 9905, pages 21–37.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Misra, I., Shrivastava, A., Gupta, A., and Hebert, M. (2016). Cross-stitch networks for multi-task learning. *CoRR*, abs/1604.03539.

Redmon, J. and Farhadi, A. (2017). Yolo9000: Better, faster, stronger. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ren, S., He, K., Girshick, R. B., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *NIPS*, pages 91–99.

Shrivastava, A., Sukthankar, R., Malik, J., and Gupta, A. (2016). Beyond Skip Connections: Top-Down Modulation for Object Detection. *arXiv e-prints*, page arXiv:1612.06851.

Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv e-prints*, page arXiv:1409.1556.

Sistu, G., Leang, I., and Yogamani, S. (2019). Real-time Joint Object Detection and Semantic Segmentation Network for Automated Driving. *arXiv e-prints*, page arXiv:1901.03912.

Teichmann, M., Weber, M., Zoellner, M., Cipolla, R., and Urtasun, R. (2016). MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving. *arXiv e-prints*, page arXiv:1612.07695.

Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., and Smeulders, A. W. M. (2013). Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171.

Valada, A., Mohan, R., and Burgard, W. (2019). Self-supervised model adaptation for multimodal semantic segmentation. *International Journal of Computer Vision*.

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., and Xiao, B. (2019). Deep High-Resolution Representation Learning for Visual Recognition. *arXiv e-prints*, page arXiv:1908.07919.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhao, Q., Sheng, T., Wang, Y., Tang, Z., Chen, Y., Cai, L., and Ling, H. (2019). M2det: A single-shot object detector based on multi-level feature pyramid network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:9259–9266.