5AUA0 Group 9 Project 2 Final Report

Alessandro Brugnera
1889958
a.brugnera@student.tue.nl

Sebastian Joseph
1874411
s.joseph1@student.tue.nl

Abstract

This report presents an approach for improving VipDeeplab which is a Depth Video Panoptic Segmentation network, by using part from another network: MonoDVPS. In particular the losses from MonoDVPS are added to the existing VipDeeplab losses in order to make it converge faster but also by keeping the architecture simpler than the MonoDVPS one. The experiments are conducted on the two dataset used in the original VipDeeplab paper: CityscapesDVPS and SemkittiDVPS. Future work involves tuning the weight of each loss with better methods. The proposed approach, although not perfect, opens the possibility for a solution that improves DVPS without adding complexity to the VipDeeplab network like has been done for MonoDVPS.

1. Introduction

The aim of this research is to improve the training process of a DVPS model. DVPS is a problem that includes several sub problems: segmentation, instance segmentation, depth estimation and temporal awareness
Segmentation and instance segmentation are very similar and convert the input image into an output one where each pixel is assigned to a class and to a defined instance of that class. Depth estimation is, as the name suggest, a process where, starting from one 2D image, an output is generated where each pixel represents the depth of that particular pixel in the original image. Temporal awareness is needed for video processing as it help in the previous tasks.
Those tasks were already solved in particular by one research, VIP-DeepLab [4], which demonstrated a good way to perform them. In this research, the goal is to reduce the training time of the VIP-Deeplab model by making it learn new characteristics. There is a strong correlation between depth and panoptic segmentation, as depth values must remain similar for pixels belonging to same object and changes when object changes. The VIP-Deeplab [4] implementation tries to utilise this relation during training by summing the semantic and depth losses to obtain the total loss used for backpropagation. Here, three additional losses specifically related to this relation are introduced to the training and the training behaviour is observed. The new proposed VIP-Deeplab implementation should try to utilise this relation during training to reach optimisation faster. The research will be done using the CityscapesDVPS and SemkittiDVPS datasets. As stated in the beginning, DVPS is a combination of multiple tasks and making the model learn more complex relations in the data should improve the overall performance of the model also.

2. Related Works

2.1. VIPDeepLab

VIP-DeepLab [4] is a video-based model that performs the DVPS task. It implements an encoder-decoder architecture that takes nth and n+1 frame and produce depth prediction and temporally consistent panoptic segmentation. The model predicts five outputs, depth value, center score, center regression, semantic segmentation and next frame regression, then combine these predictions to give depth and panoptic prediction. This model is an extension of the Panoptic-Deeplab that performs DVPS task just for a single frame.

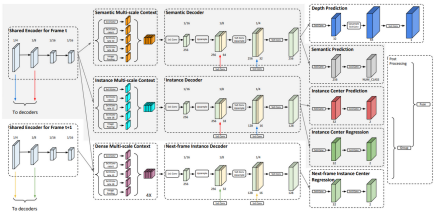

Figure 1. VIPDeepLab architecture [4]

2.2. MonoDVPS

MonoDVPS [3] is a model that uses a multi-task network to perform DVPS. The key change here is that they try to minimize the costs by using self-supervised and semi-supervised methods in order to use unlabeled video sequences for training. They also introduce new losses,

which are the thing that the current research took from them, and those losses are panoptic-guided and they also introduced the concept of a panoptic masking scheme for objects in movement which are there to prevent corruption in the training signal. The architecture is different from VIPDeepLab altough it takes a lot of inspiration from it.
The main problem encountered with this paper was the lack of details in the network architecture which lead to a lot of unclear design choices which in the end lead to a network which might become too different from the original and not work as expected. For this reason we tried to take the losses from this architecture which are well explained with mathematical formulas and combine them with the VIPDeepLab architecture.

Both VIPDeepLab and MonoDVPS use the same two datasets, introduced by VIPDeepLab. They are further discussed in the appropriate section.

3. Proposed Method

3.1. Method

A DVPS model has to extract multiple types of information from the image sequence in order to perform video panoptic segmentation and depth prediction effectively. This includes identifying instance centers, semantic segmentation, depth values, and next frame instance offset. The characteristics that the model learns are primarily based on the loss function used during backpropagation.

In the VIP-Deeplab model, individual losses are employed for each prediction but it only considers the true and predicted values of each prediction. Although the model sums all individual losses to calculate the total loss used for backpropagation, there is no loss that represents the mutual relation between predictions. As a result, the model takes longer to learn the relationships between different predictions, impeding its efficiency.

The MonoDVPS [3] model outperforms the VIP-Deeplab model significantly in the DVPS task. However, the MonoDVPS model is more complex and learns additional characteristics such as optical flow and pose estimation compared to VIP-Deeplab. It also implements extra losses to facilitate the learning of relationships between different predictions. In the MonoDVPS architecture, three panoptic-guided losses are introduced.

The first is the panoptic-guided smoothness loss, which aims to enforce similar depth values for pixels within the same panoptic segmentation. This loss penalizes abrupt changes or inconsistencies in depth between neighbouring pixels or regions.

$$L_{\text{pgs}} = \left|\partial x \bar{dt}\right| (1 - \partial x P_t) + \left|\partial y \bar{dt}\right| (1 - \partial y P_t)$$

where P_t represents the panoptic ground truth label, ∂P_t are the panoptic contours, and $\bar{dt}$ is the mean normalized inverse depth. For two adjacent pixels (p_0, p_1), we define $\partial x P_t(p_0, p_1)$ as the Iverson bracket:

$$\partial x P_t(p_0, p_1) = [P(p_0) \neq P(p_1)]$$

The second is the panoptic-guided edge discontinuity loss, which encourages a peak in the gradient of the disparity map at panoptic edges. This is particularly relevant when adjacent pixels have different panoptic segmentation.

$$L_{\text{ped}} = \partial x P_t e^{-|\partial x \bar{dt}|} + \partial y P_t e^{-|\partial y \bar{dt}|}$$

Lastly, the edge-aware smoothness loss which encourages the adjacent pixels to have similar depth values unless an edge is present in the image.

$$L_{\text{smooth}} = \partial x d_t e^{-|\partial x \bar{I} t|} + \partial y d_t e^{-|\partial y \bar{I} t|}$$

where I_t represents the image.
In this implementation of VIP-Deeplab, the above three depth-related losses are added. Unlike traditional depth losses that only consider the difference between the prediction and ground truth, these new losses encourage the model to consider panoptic segmentation while predicting the depth values. This should help to reduce the training time required

3.2. Data

In this paper, two different dataset have been used. One is called CityscapesDVPS [4] and the other SemKittiDVPS [4].
CityscapesDVPS has been partially created by the authors of VIPDeepLab. They started with CityscapesVPS which is a dataset derived from the original Cityscapes [2]. The original one contains images of city streets and a label for semantic segmentation, however it is missing the video panoptic annotations. Those are being proivded by the extended CityscapesVPS which includes several 30 frames video where every 5 frames there is one which is annotated for this new purpose. Additionally the authors of VIPDeepLab augmented this data even further by adding the depth map in the dataset. They did so by estimating the depth by looking at the stereo images provided in the original Cityscapes in addition to several methods which are not disclosed.
SemKittiDVPS [4] was also augmented by the authors of VIPDeepLab. In this case the original SemanticKitti [1], which is derived from the KITTI dataset, consists of 3D point clouds which are semantically annotated and each

cloud represents a frame. They projected the 3D point clouds into 2D planes and by doing so they used different methods to address the problems that came from the sensors positions, which can lead to some sensors not being able to see every point that other sensors may see and problems coming from the fact that sometimes thin objects which should be a whole piece are full of distant point coming from the background which bring noise to the label.

The Cityscapes dataset consists of images along with panoptic labels and depth labels, while the Semkitti dataset includes depth predictions, instance labels, and semantic labels. However, these labels cannot be directly used as the ground truth values for the prediction heads. Preprocessing steps are performed to extract the truth values for each prediction head.

3.3. Evaluation and experiments

In order to investigate whether these additional losses can expedite the learning process, we conducted experiments using four different models. Two models were trained on each of the two datasets: Cityscapes DVPS and Semkitti DVPS. By using both datasets, we aimed to observe if the effectiveness of the method is diminished when applied to sparse data.

Each model was trained for 20 epochs, and we examined the changes in loss for the training and validation images. If the losses are indeed improving the model, the individual depth loss should decrease more rapidly compared to the models without the new losses.

It is important to note that the semantic head and depth head utilize the same decoder as a feature extractor. Therefore, the addition of new losses is expected to assist the common decoder in reaching optimization more quickly, ultimately expediting the overall model optimization process.

In addition to that, the single losses are taken into consideration for evaluation. By looking at the individual losses it should be possible to determine more accurately whether the model is improving or not and at which rate it is doing so. This gives a better look at the training process as it permits an accurate analysis of the additional losses.

Overall the process of evaluation compares only similar models, which means it is comparing only models trained on the same dataset. This is needed in order to exclude any possible influence of the type of dataset used.

3.4. Results and discussion

The results can be subdivided into common and individual, depending on which of the two datasets was used for training. The main common result is that the additional depths did not improve the model and, as later explained, in one case made training worse and in the other they basically added no improvements. Another common result, as seen in the training graphs shown in the paper, is that the validation has got worse with the additional depths. More specifically it started oscillating for the trainings done with Cityscapes DVPS dataset and got down for the Semkitti DVPS dataset but this was due to the model being able to fool the loss so it is still to be considered as worse than the baseline.

When it comes to the individual results we can separate them into two sections, one per dataset. The trainings made with Cityscapes DVPS dataset, were not very different between each other. This in itself means that the additional losses did not have an impact on the training and therefore did not improve it. The only thing to be noted which is also a reason of why the training with the additional losses has to be avoided is that it made the validation very unstable, proving that there is overfitting during the training phase. The reason for this conclusion is that oscillating validation means random predictions, which is the case when an overfitted network is given an unseen sample. This can be observed clearly from the depth prediction of the model with new losses shown in Figure 3.

The same decoder branch is used by both depth prediction head and semantic head. The weights assigned to the new additional depth were high compared to the weights assigned to semantic loss and normal depth loss. Due to this, the model is giving high significance to the task of predicting the depth based on panoptic segmentation. The model architecture is designed to perform semantic segmentation. In order to predict panoptic segmentation, the model needs to understand the center score prediction and offset regression tasks. This model is trying to learn a task nearly impossible for it. As a result of it, the model is unable to perform either semantic prediction or depth prediction.

Semkitti DVPS is a dataset composed of sparse data and this made a significant impact on the losses. In particular there are 3 losses that are trying to smooth the prediction and make all the adjacent pixels the same value and just 1 loss, the edge loss, is trying to counteract this behavior. Due to the nature of the sparse data, between the labeled points there is a lot of unlabeled points which the losses are not ignoring, and since the impact of the 3 smoothing losses is way greater, the predictions become all flat and give the same value as the unlabeled data also to the labeled pixels. The different loss behaviour can be observed in the Figure 4

The possible reasons for all the problems as found out in the final stages of the research, are the wrongly tuned hyperparameters. All the losses described in this paper and used in the network, are given a weight which is a multiplication factor that gets applied to the losses when summing them to get the total loss. These factors if not tuned properly can make one or more losses take over other losses and this was found out when a very unbalanced weight was given to two of the two losses during the test. The weights used

for this research however were not given randomly. They were given empirically by following the rule for which all the losses should be of the same order of magnitude and possibly even equal, at least in the first epochs. This was a design choice that has been made but is likely that the magnitude of the losses needed to be adjusted by grouping the losses into their respective region of influence in the network and then adjusting the weights based on the sum of the losses per each group. However, it is still possible that even slightly unbalanced weights can lead to the same issue in the long term and therefore more research need to be put into it. Unfortunately, the problem was discovered too late and therefore, a better hyperparameter tuning is left as a suggestion for future researches which wants to work with this kind of improvements for this network. Instead of panoptic guided losses, semantic guided losses are more suitable for smaller models like VIP-Deeplab.

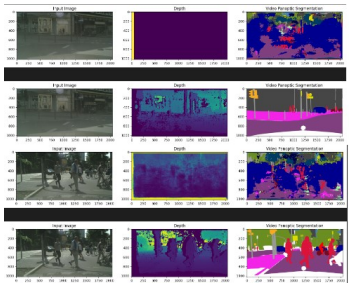

Figure 2. Prediction and truth for VIP-Deeplab model trained on CityScapeDVPS without new losses (top) and with new losses (bottom)

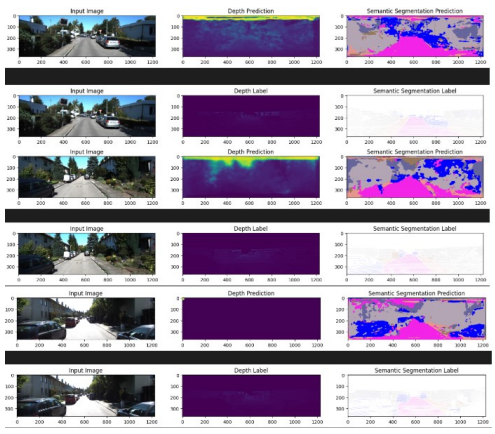

Figure 3. Prediction and truth for VIP-Deeplab model trained on SemKittiDVPS without new losses (top), with new losses after 2 epoch (middle) and with new losses after 10 epoch(bottom)

3.5. Conclusion

Although the answer to the main question of this research was not positive, meaning that the additional losses

Figure 4. Losses during training and evaluation for VIP-Deeplab model trained with SemKittiDVPS dataset and new losses (Total, Depth, Edge, P. G. Smoothness, Smoothness)

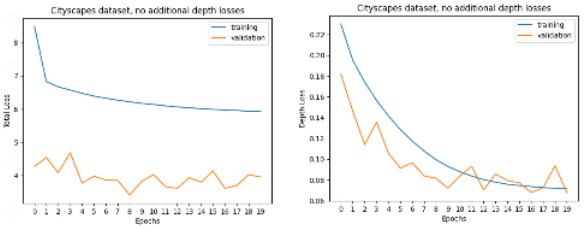

Figure 5. Losses during training and evaluation for VIP-Deeplab model trained with CityScapeDVPS dataset (Total, Depth)

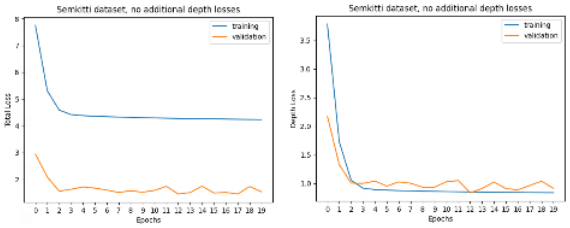

Figure 6. Losses during training and evaluation for VIP-Deeplab model trained with SemKittiDVPS dataset (Total, Depth)

Figure 7. Losses during training and evaluation for VIP-Deeplab model trained with CityScapeDVPS dataset and new losses (Total, Depth, Edge, P. G. Smoothness, Smoothness)

did not improve the overall model, it has to be noted that some important results were still found, such as the importance of the hyperparameters of a network and how they can affect it in positive or negative ways. This research also provided the future researchers with a possible solution to the problem of why it may have failed to demonstrate the premises. This solution may also then give the possibility to get a positive answer to the question of this research.

On top of that this research demonstrated the impact of sparse data like the one present in SemkittiDVPS dataset in contrast to continuous data present in the CityscapesDVPS dataset and presented the problem of having to create one architecture that can deal with both of these kind of data, focusing in particular on the losses of such architecture.

References

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss. Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset. *The International Journal on Robotics Research*, 40(8-9):959–967, 2021. 2

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016. 2

[3] Andra Petrovai and Sergiu Nedevschi. Monodvps: A self-supervised monocular depth estimation approach to depth-aware video panoptic segmentation, 2022. 1, 2

[4] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan L. Yuille, and Liang-Chieh Chen. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. *CoRR*, abs/2012.05258, 2020. 1, 2