Robust Semantic Video Segmentation through Confidence-based Feature Map Warping

Timo Sämann
timo.saemann@valeo.com
Valeo Schalter und Sensoren GmbH

Karl Amende
karl.amende@valeo.com
Valeo Schalter und Sensoren GmbH

Stefan Milz
stefan.milz@valeo.com
Valeo Schalter und Sensoren GmbH

Horst Michael Groß
horst-michael.gross@tu-ilmenau.de
Ilmenau University of Technology, Neuroinformatics and Cognitive Robotics Lab

ABSTRACT

One of the limiting factors when using deep learning methods in the field of highly automated driving is their lack of robustness. Objects that suddenly appear or disappear from one image to another due to inaccurate predictions as well as occurring perturbations in the input data can have devastating consequences. A possibility to increase model robustness is the use of temporal consistency in video data. Our approach aims for a confidence-based combination of feature maps that are warped from previous time stages into the current one. This enables us to stabilize the network prediction and increase its robustness against perturbations. In order to demonstrate the effectiveness of our approach, we have created a test data set with image perturbations such as image artifacts and adversarial examples in which we significantly outperform the baseline.

KEYWORDS

safe artificial intelligence, semantic segmentation, robustness against perturbations in input data, leverage temporal consistency, calibrating neural networks

ACM Reference Format:
Timo Sämann, Karl Amende, Stefan Milz, and Horst Michael Groß. 2019. Robust Semantic Video Segmentation through Confidence-based Feature Map Warping. In *Proceedings of ACM*. ACM, Kaiserslautern, Germany, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The realization of highly automated driving requires the intensive use of deep learning methods. One of the major challenges when using deep learning methods in the automotive industry are the high safety requirements for the algorithms. The use of black box solutions causes a potential safety risk and is therefore not permitted. For this reason, deep learning methods are needed that show comprehensible behaviour for us humans. In addition, the algorithms must be reliable and robust in the case of perturbations in the input data. These perturbations can be caused by sensor errors, external contamination of the sensors, overexposure or the occurrence of adversarial examples. Objects that suddenly appear or disappear from one frame to another due to inaccurate prediction or occurring perturbations can have disastrous consequences. These aspects receive less attention in the scientific community and are neglected in public data sets.

One way to achieve robustness against perturbations is to use temporal consistency in video data. The vast majority of previous deep neural networks have an independent single image prediction of the currently recorded scene, i.e. the same operations are performed for all input images and the information already received from the previous time step is discarded. For video processing in which there is a temporal continuity of the image content, the use of the information from previous time steps can overcome perturbations that would otherwise lead to miss-classification.

With our approach, we are able to overcome perturbations by incorporating the relevant information from earlier time steps into the current prediction. The idea is to combine the calculated feature maps from earlier time steps with the current feature map to compensate for shortcomings in the current prediction. This requires warping the feature maps from previous time steps t_{-1} to t_{-n} into the time stage t_0, where n is the number of previous time steps. Warping takes place via the optical flow, following the idea of [5]. According to our experiments, a naive combination of the complete feature maps does not always lead to an improvement of the results. There are two main reasons for this:

1. It is in the nature of things that frames from previous time steps are less relevant than the current frame. Objects that appear in the image for the first time, e.g. because they have been covered by another object, cannot be represented by warping.
2. The warping process depends on the quality of the optical flow. Especially objects with a low pixel density like pole where the optical flow is not precise enough suffer in quality.

Therefore, a confidence-based combination of feature maps is performed that significantly reduces these issues. The confidence map gives us a confidence value for each pixel in the image that estimates the confidence of the prediction. The confidence map is obtained by probabilities from softmax distributions, which we have calibrated to obtain a reliable confidence estimate. We have observe that the confidence maps have a relatively low value at the

---

ACM, October 2019, Kaiserslautern, Germany

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

areas in the image where we have inserted a perturbation, cf. [8]. Therefore, we use the confidence values as a measure of which areas of the feature maps t_{-1} to t_{-n} we combine with the feature map t_0. For the combination, a weighting is used that can be derived from the confidence values of the current and previous confidence maps. The areas of feature maps that have a higher confidence than the areas of the current feature map are combined. The combined feature map fm_new_{t0} then serves as the new feature map $fm_{t_{-1}}$.

To demonstrate the effectiveness of our approach, we use semantic video segmentation applied to two test data sets: One set of test data with artificially added perturbations, such as image artifacts, masking and adversarial pattern. And another one with the same images, but without any perturbations. We show that our approach not only significantly outperforms the perturbed data set but also slightly improves the baseline on the *clean* data set. Our approach is independent of the network architecture and does not require any further training or fine-tuning.

2 RELATED WORK

The focus of previous Deep Neural Networks (abbr. DNN) development has been on single image prediction. This means that the results and intermediate results of the DNN calculated with great effort are discarded after each image. Thus the processing of the input data takes place independently of each other. However, the application of many DNNs often involves the processing of images in a sequence, i.e. there is a temporal consistency in the image content between adjacent input images. The use of this consistency has been used in previous work in this area to increase quality and reduce computing effort. Furthermore, this approach offers the potential to improve the robustness of DNN prediction by incorporating this consistency as a priori knowledge into DNN development. The relevant work in the field of video prediction in the computer vision area differs essentially in two aspects:

1. DNNs are specially designed for video prediction. This usually requires training from the scratch and the presence of training data in a sequence.
2. A transformation from single prediction DNNs to video prediction DNNs takes place. Usually no training is required, i.e. the existing weights of the model can be used unchanged.

The first aspect often involves Conditional Random Field (abbr. CRF) and its variants. CRFs are known for their use as post a processing step in the prediction of semantic segmentation whose parameters are learned separately or jointly with the DNN [22][1][2]. They refine the prediction of DNNs based on image intensities or calculated superpixels. The idea is based on the assumption that image areas with similar pixel intensities or superpixels belong to the same object. The use of CRFs for semantic video prediction allows an estimation of how likely a current prediction is based on the previous one. This is performed at the pixel and supervoxel level in [18][12] respectively. An optimization of the feature space of the DNN or another classifier used as input for the CRF is performed in [11]. Another way to use spatiotemporal features is the include of 3D convolutions, which adds an additional dimension to the conventional 2D convolution layer. [16] use 3D convolution layers for video recognition tasks such as action and object recognition. [17] extend [16] by using a 3D deconvolution layer and use this for semantic video segmentation and optical flow prediction.

One further approach to use spatial and temporal characteristics of the input data is to integrate Long Short Term Memory (abbr. LSTM) [9][6], a variant of the Recurrent Neural Network (abbr. RNN). [4] integrates LSTM layers between the encoder and decoder of their Convolutional Neural Network for semantic segmentation. The significantly higher GPU memory requirements and computational effort are a disadvantage of this method. More recently [13] uses Gated Recurrent Unit [3], which generally requires significantly less memory. A disadvantage of the described methods is that sequential data for training must be available, which are often limited and show a lack of diversity.

The second aspect is more related to our approach and has the advantage that the approach is relatively model independent and transferable to other models. The authors of [15] found that the deep feature maps within the network change only slightly according to the change in image content in a video. Flat feature maps show larger differences with smaller input changes. This observation is used by the clockwork FCN presented in [15]. Flat feature maps are updated more frequently than deep feature maps. At the end of the network a fusion of flat and deep feature maps takes place. This process leads to less computational effort, since only parts of the network are calculated per input image. Reduced computational effort is accompanied by a partially significant reduction in output quality.

The authors of [5] first calculate the optical flow of the input images from time steps t_0 and t_{-1} and transform it into the so-called *transform flow*. This is used to transform the feature maps of the time step t_{-1}, so that an aligned representation to the feature map t_0 is achieved. The transformed feature maps from time step t_{-1} are then fused with the current feature maps from time step t_0. This procedure applied to the PSPNet [21] could improve the mean Intersection over Union (abbr. mIoU) of the Cityscapes data set from 79.4 to 80.6 %. The higher computational effort of this approach represents a considerable expenditure of time, depending on the image resolution.

To the best of the authors' knowledge, a confidence-based combination of feature maps from previous time steps has not yet been published.

3 PIPELINE

The entire pipeline of our approach is shown for a single time step t_0 in Fig. 1. As an example DNN we use the ENet architecture [14], a ResNet based network that has a very low runtime while providing a respectable quality. However, the model can be exchanged with any other architecture. The model was trained on an internal fisheye training data set with 17 classes. After the last layer, the number of feature maps is 17 (referenced as fm_{t0} in Fig. 1). The calculation of the Argmax[1] and the following coloring lead to the baseline of semantic segmentation for the current time step, which is in Fig. 1 referred as $Segmentation_{t0}$.

The confidence map of the current time step cm_{t0} is determined by the probabilities from softmax distributions. To obtain reliable confidence values, we calibrate the DNN by using temperature

---

[1] For every pixel, the index of the maximum value along the depth axis is determined.

scaling, which consists of a single value added to the softmax layer (see subsection 3.1). The confidence and feature maps are warped in a so-called *warp* module (see box with red border in Fig. 1), which is described in detail in subsection 3.2. This module requires the optical flow as well as the confidence and feature maps from the previous time steps.

This aligned confidence maps are processed in the so-called *thresh* module (see box with green border in Fig. 1) with threshold values and a weighting, which is described in subsection 3.3. In the *combine* module, the feature maps from the *warp* module are multiplied by the threshold confidence maps from the *thresh* module (see subsection 3.4). The output of the *combine* module are 17 feature maps, which are composed pixel by pixel from the feature maps of time steps t_0 to t_{-n}. The new confidence map is called cm_new_{t0} and the robust semantic segmentation $RobustSegmentation_{t0}$. Please note that all results in section 6 refer to $n = 2$.

3.1 Confidence Calibration

Confidence calibration describes the problem of predicting probability estimates that are representative of the true probability of correctness [7]. In other words, the aim of confidence calibration is to achieve the best possible consistency in predicting confidence and accuracy. For example, if the confidence of an image results in 90%, the accuracy of this image should also result in 90%. [7] has found that modern networks tend to be over-confidence in predicting confidences. The reason for the overconfidence of modern networks is the increased network capacity, the use of batch normalization and weight decay. A metric that indicates how well the network is calibrated is the Expected Calibration Error (ECE). To the best of our knowledge, the ECE metric has so far only been applied for image classification. In contrast to image classification, in semantic segmentation we do not calculate the gap between *acc* and *conf* per image but per pixel. This change requires an additional loop over all images that average the ECE. More formally, we describe the ECE for semantic segmentation as

$$\mathrm{ECE} = \sum_{l=1}^{L} \sum_{m=1}^{M} \frac{\|B_{m;l}\|}{n} \left\| \mathrm{acc}(B_{m;l}) - \mathrm{conf}(B_{m;l}) \right\| \tag{1}$$

with

$$\mathrm{acc}(B_{m;l}) = \frac{1}{\|B_{m;l}\|} \sum_{i \in B_{m;l}} \mathbf{1}(\hat{y}_i = y_i) \tag{2}$$

$$\mathrm{conf}(B_{m;l}) = \frac{1}{\|B_{m;l}\|} \sum_{i \in B_{m;l}} \hat{p}_i. \tag{3}$$

We designate L as the number of images, M as the number of interval bins in which the predictions are grouped[2], B as the number indices of pixels whose predictions (accuracy and confidence respectively) falls into the bin interval, n as the number of pixel per image, $\hat{y}$ as the prediction of the class, $\hat{p}$ as the prediction of the confidence, y as ground truth and i as the number of pixel per image and bin.

To calibrate the DNN we use temperature scaling, which consists of a single value added to the softmax layer. It was found in [7] that this type of calibration is the simplest and most effective at the same time. The extension provides for a division of the input of the softmax layer z with a scalar T (see Eq. 4). The optimal temperature scaling parameter was determined by Grid Search on our validation data set. This allowed us to reduce the ECE from 1.9 to 1.1. Relevant reference values from the literature could not be found. A direct comparison with ECE values from the task of image classification is not possible since the calculation is different.

$$\mathrm{softmax} = \frac{e^{\frac{z}{T}}}{\sum_i e^{\frac{z_i}{T}}} \tag{4}$$

3.2 Warp Module

The function of the *warp* module is to warp the feature or confidence maps from past time steps into the current time step in order to obtain a aligned representation. For warping we use the optical flow that we create with FlowNet2 [10]. Please note that we apply this model for fisheye camera images, although the model was trained on pinhole camera images. This leads to slightly worse results at the lateral areas of the image. To avoid jaggies during warping we use bilinear interpolation filters. The number of past time steps is variable and depends on the application. For the sake of simplicity, we decided on two time levels: t_{-1} and t_{-2}. The general case for the feature and confidence map warping is described in 5 and 6 respectively.

$$fm_{t0} = \mathrm{Warp}(fm_{(t-1,\cdots,t-n)}; opt_{(t-1,\cdots,t-n)}) \tag{5}$$

$$cm_{t0} = \mathrm{Warp}(cm_{(t-1,\cdots,t-n)}; opt_{(t-1,\cdots,t-n)}) \tag{6}$$

Here *opt* stands for optical flow, fm for feature map and n for the number of previous time steps.

3.3 Thresh Module

In the *thresh* module the confidence maps are processed in the first step with threshold values and in second step with a weighting. The resulting confidence maps $cm\_th_{(0-2)}$ can be considered as a mask used for multiplication with the feature maps fm_{0-2} in the *combine* module.

First, we have found experimentally that if the confidence values of cm_0 are above a certain threshold, it leads to better results when no combination with confidence maps from earlier time steps is performed (see section 5). Therefore, we set all pixels i, j from cm_0 with a confidence value above the threshold to 1, which we then call $\widetilde{cm}_0$. To take into account that frames from past time steps are less relevant, we assign a generally lower confidence to the earlier time steps, subtracting 10% or 20% of their confidence value from cm_1 and cm_2, which we denote with $\widetilde{cm}_1$ and $\widetilde{cm}_2$ respectively. This threshold causes fewer pixels to be combined from the t_{-2} time step than from the t_{-1} time step.

Second, the confidence values of $\widetilde{cm}_0$ are compared pixel by pixel to the confidence values of $\widetilde{cm}_1$ and $\widetilde{cm}_2$ for their size, see Eq. 10 and 11. For all confidence values of $\widetilde{cm}_0$, which are lower than the confidence values of $\widetilde{cm}_1$ or $\widetilde{cm}_2$, a weighting of the confidence values takes place. In contrast, all confidence values of $\widetilde{cm}_0$ that are greater than $\widetilde{cm}_1$ or $\widetilde{cm}_2$ are set to 1 in cm_th_0, see Eq. 7. Please note that we do not only take the pixels from the previous time step, we perform a weighting. As an example: Assuming one specific confidence pixel value in the current time step $\widetilde{cm}_0$ is 0.6 (60%) and

---

[2] The number of interval bins in our case is 5, which leads to a bin size of 20% (100%/5) each.

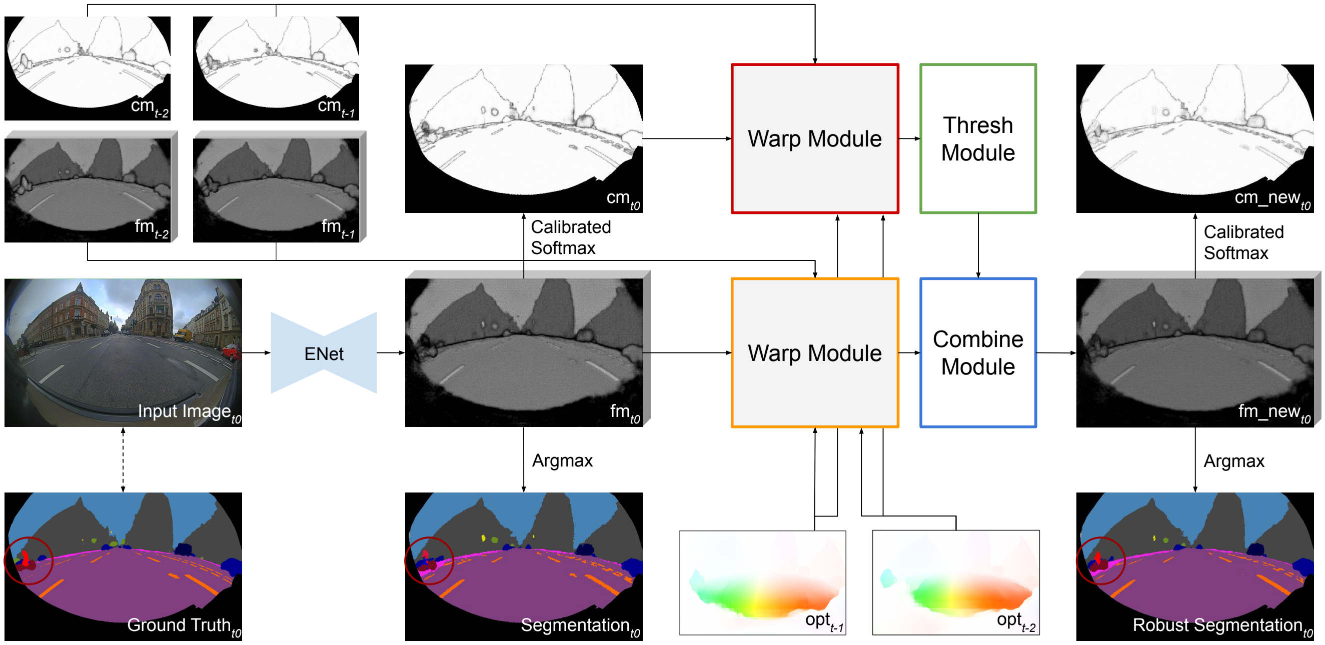

Figure 1: Overview of our pipeline. The abbreviations “cm”, “fm” and “opt” stand for confidence map, feature map and optical flow respectively. White areas in the cm mean a high and black a low confidence.

the corresponding value in the previous time step $\widetilde{cm}_1$ is 0.8 (80%), then a weighting of the two is performed. The value of $\widetilde{cm}_0$ is set to the calculated ratio $\frac{0.6}{0.6+0.8} \approx 0.43$ and $\widetilde{cm}_1$ to $\frac{0.8}{0.8+0.6} \approx 0.57$ or short: $1 - 0.43 = 0.57$, see Eq. 8 and 9. With this procedure we limit the influence of the previous time stages and increase the influence of the current one.

A visualization of the resulting masks cm_th_{0-2} can be found in Fig. 2. White means that these pixels are combined from the corresponding time step in the *combine* module. Black means the opposite. An addition of the masks cm_th_{0-2} would result in a fully white image. The calculation of the threshold confidence maps cm_th_{0-2} is listed in the following:

$$cm\_th_0^{i,j} = \begin{cases} \frac{\widetilde{cm}_0^{i,j}}{\widetilde{cm}_0^{i,j}+\widetilde{cm}_1^{i,j}}, & \text{if } cm\_mask_1^{i,j}, \\ \frac{\widetilde{cm}_0^{i,j}}{\widetilde{cm}_0^{i,j}+\widetilde{cm}_2^{i,j}}, & \text{if } cm\_mask_2^{i,j}, \\ 1, & \text{else} \end{cases} \tag{7}$$

$$cm\_th_1^{i,j} = \begin{cases} 1 - cm\_th_0^{i,j}, & \text{if } (\widetilde{cm}_1^{i,j} > \widetilde{cm}_2^{i,j}), \\ 0, & \text{else} \end{cases} \tag{8}$$

$$cm\_th_2^{i,j} = \begin{cases} 1 - cm\_th_0^{i,j}, & \text{if } (\widetilde{cm}_2^{i,j} > \widetilde{cm}_1^{i,j}), \\ 0, & \text{else} \end{cases} \tag{9}$$

with

$$cm\_mask_1^{i,j} = (\widetilde{cm}_1^{i,j} > \widetilde{cm}_0^{i,j}) \& (\widetilde{cm}_1^{i,j} > \widetilde{cm}_2^{i,j}) \tag{10}$$

$$cm\_mask_2^{i,j} = (\widetilde{cm}_2^{i,j} > \widetilde{cm}_0^{i,j}) \& (\widetilde{cm}_2^{i,j} > \widetilde{cm}_1^{i,j}) \tag{11}$$

3.4 Combine Module

The *combine* module contains a multiplication of the feature maps fm_{0-2} with the threshold confidence maps from the *thresh* module cm_th_{0-2} and a sum of the resulting feature maps which we call fm_new_{0-2}. The multiplication is pixel by pixel and can be seen as a weighted masking of the feature maps (see Eq. 12, 13 and 14). The pixelwise sum of the resulting feature maps fm_new_{0-2} gives the final feature map fm_new.

$$fm\_new_0 = fm_0 \cdot cm\_th_0 \tag{12}$$

$$fm\_new_1 = fm_1 \cdot cm\_th_1 \tag{13}$$

$$fm\_new_2 = fm_2 \cdot cm\_th_2 \tag{14}$$

4 DATA SET

In order to test our approach, we have created our own data set consisting of a sequence of 1200 images which we refer to as *clean* data set. The images were taken with a fisheye camera and show scenes from the downtown area of a German city. To generate ground truth data we use the Tu-Simple-DUC model [19], which we trained on a fisheye camera data set before. In order to test the robustness of our network, a second data set was created by adding perturbations to the first data set, which we refer to as *perturb* data set. The perturbations can be divided into 3 different categories: Random patterns (random changes of multiple color channels), real perturbations (e.g. caused by packet loss) and adversarial patterns (generated from [20]). Images to which a perturbation was assigned were selected at random. Furthermore, the perturbations were placed at random locations in the image and can occur up to 6 times per image. In addition, perturbations also occur over several

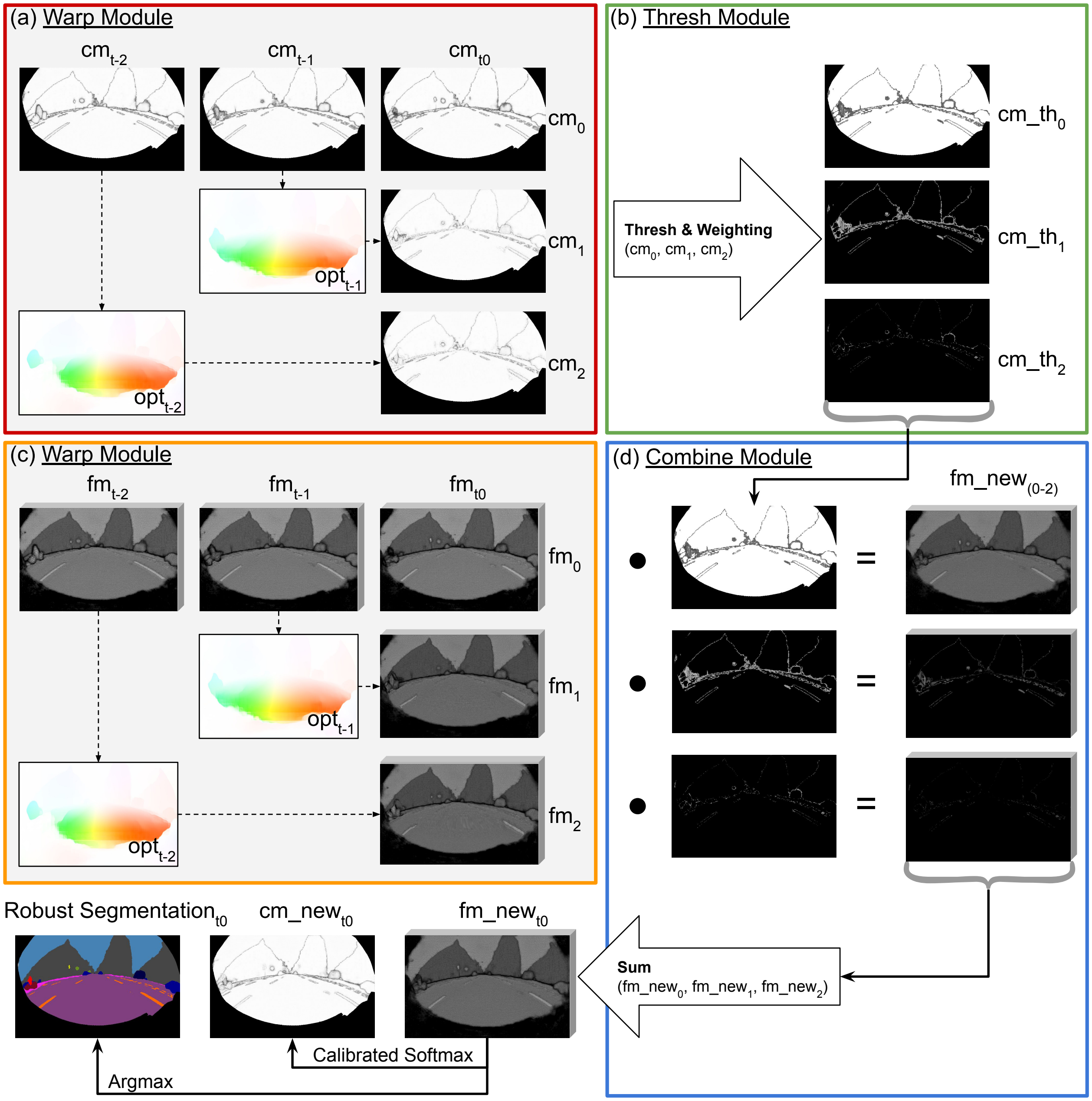

Figure 2: Detailed description of our pipeline. The colors of the framed modules correspond to the colors in Fig. 1. The abbreviations "cm", "fm" and "opt" stand for confidence map, feature map and optical flow respectively. White areas in the cm mean a high confidence and black a low one. For the "cm_th" images, white areas mean that they are combined and black areas mean that they are not combined.

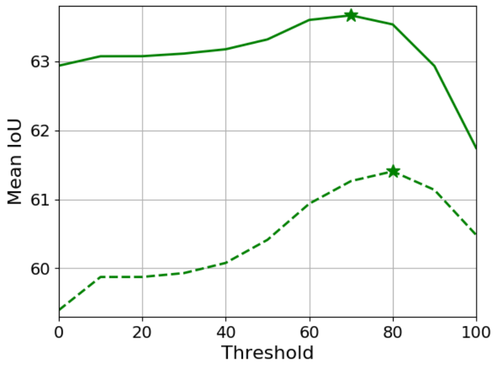

Figure 3: Comparison of the effect of different thresholds on the mean IoU. The solid and dashed curve indicates the evaluation on the *clean* and *perturb* validation data set respectively. The asterisk marks the threshold value that leads to the highest mean IoU value.

frames to evaluate robustness over a longer lasting perturbations. In total, 412 (33.67%) of 1200 images contain at least one added perturbation.

5 EXPERIMENTS

In our experiments we investigate the influence of different thresholds for $\widetilde{cm}_0$. This threshold determines the confidence values at which a combination with feature maps from previous time steps should be performed or not. If the predicted confidence is above this threshold, no combination takes place. A value of 0% can be equated with the baseline, i.e. no combination takes place. A value of 100% leads to a combination at all pixels. Fig. 3 shows the effect of the different thresholds on the mean IoU value for our validation data set. The solid curve describes the *clean* data set and the dashed curve the *perturb* data set. The asterisk marks the threshold value that leads to the highest mean IoU value. For the *clean* data set the threshold is 70% and for the *perturb* data set 80%. We use these values for the evaluation on the test data set, which we report in section 6. The course of the solid curve shows that a combination improves already from 0%, but declines relatively steeply from about 80%. A threshold of 100% leads to a deterioration of the mean IoU, from which it can be concluded that a naive combination of feature maps is not sufficient. The dashed curve has a similar curve to the solid curve, but rises more strongly and drops significantly flatter at the back. One explanation is that a combination has a much more positive effect in the *perturb* data set. Even a naive combination (threshold = 100%) leads to a significant improvement of the mean IoU value.

6 RESULTS

We evaluate our approach qualitatively and quantitatively on the basis of two data sets: One without added perturbation pattern, which we call *clean*, and one with which we call *pertub*, see section 4. For qualitative evaluation we use the mean intersection over union (mIoU) and the global accuracy. The mIoU for the *clean* data set could be clearly improved from 62.39% to 63.20%. The IoU values per class are listed in Table 1. Apart from the classes *pole, traffic light* and *rider*, the values have increased significantly. One reason for the deterioration of these classes can be found in the inaccurate optical flow. A correct warping of the class *pole* requires a very precise optical flow. The global accuracy, which indicates the percentage of pixels correctly classified, could be increased from 95.31% to 95.56%. Evaluated on our *perturb* data set the baseline worsens to a mIoU of 57.51% and a global accuracy of 93.87%. With our approach we achieve a significant increase of the mIoU from over 2.3% to 59.86% and a global accuracy of 94.61%. Due to the low confidence values at the locations of the perturbation patterns, these locations are used for combination. In this way, the negative effects of perturbations on prediction can be overcome or mitigated.

Fig. 4, 5, 6 and 7 show the qualitative results for the data set *clean* and *perturb*. Four images are viewed in consecutive time steps. (a) represents the input image, (b) the baseline and (c) our approach. With our approach we achieve a much more stable and robust prediction in Fig. 4 and 5. Please note that our approach generally looks much smoother than the baseline, although the resolution is exactly the same. Even more clearly, the improvement can be seen in Fig. 6 and 7 for our *perturb* data set. Please note the image caption for further information. Furthermore we uploaded a video showing the sequence of our *clean* data set with the corresponding baseline and robust segmentation. The stability of our robust segmentation becomes much clearer than on pictures.

Table 1: Quantitative results of our *clean* (abbr. "cle") and *perturb* data set (abbr. "per"). Comparison of the baseline (abbr. "Base") and our approach (abbr. "Ours"). All values are given in percent and indicate the IoU.

Classes	**Base-cle**	**Ours-cle**	**Base-per**	**Ours-per**
Road	95.66	**95.90**	94.59	**95.39**
Sidewalk	73.13	**74.06**	70.19	**72.42**
Building	92.36	**92.71**	89.97	**90.83**
Wall	64.96	**68.03**	43.57	**52.02**
Fence	20.26	**20.99**	17.72	**18.57**
Pole	**39.11**	38.16	**37.50**	36.74
Traffic light	**47.86**	47.26	**46.32**	45.50
Traffic sign	48.32	**49.98**	46.15	**48.08**
Vegetation	85.30	**85.86**	79.47	**81.26**
Terrain	31.90	**33.31**	23.87	**26.58**
Sky	96.10	**96.35**	94.46	**95.18**
Person	43.98	**45.40**	42.39	**44.21**
Rider	**40.13**	39.12	29.81	**37.22**
Car	86.75	**87.07**	82.46	**84.17**
Truck	81.84	**82.87**	73.92	**78.31**
Bicycle	46.16	**48.66**	45.03	**47.65**
Road markings	66.89	**68.73**	60.17	**63.49**
Mean IoU	62.39	**63.20**	57.51	**59.86**

7 CONCLUSION AND FUTURE WORK

Safety-critical applications require reliable and robust algorithms. We introduced an approach that allows a DNN for semantic image segmentation to leverage consistency in video data to make the prediction much more robust. With regard to suddenly occurring

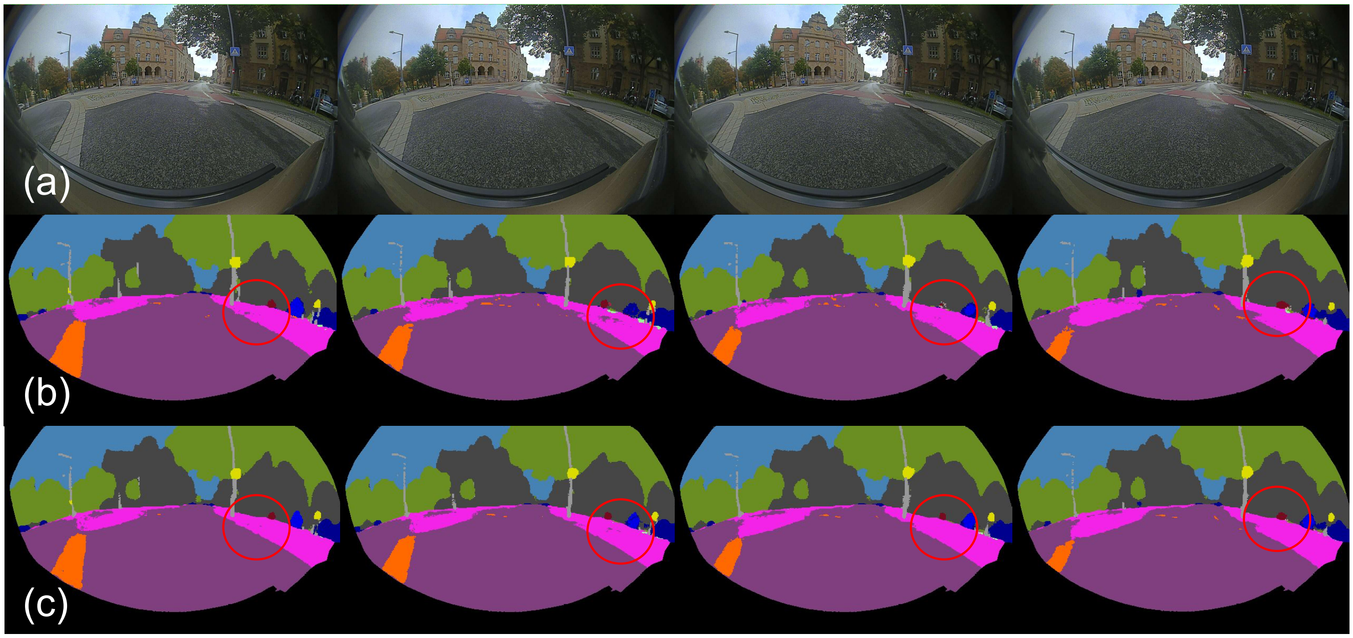

Figure 4: Qualitative results from our *clean* data set. 4 images are shown in consecutive time steps. (a) input image, (b) baseline, (c) our approach. It can be seen that the sidewalk in all pictures is much denser and has fewer holes. Furthermore, the baseline shows a class change between *motorcycle* and *car* in column 2, as well as the disappearance of *bicycle* in columns 3, which does not happen with our approach.

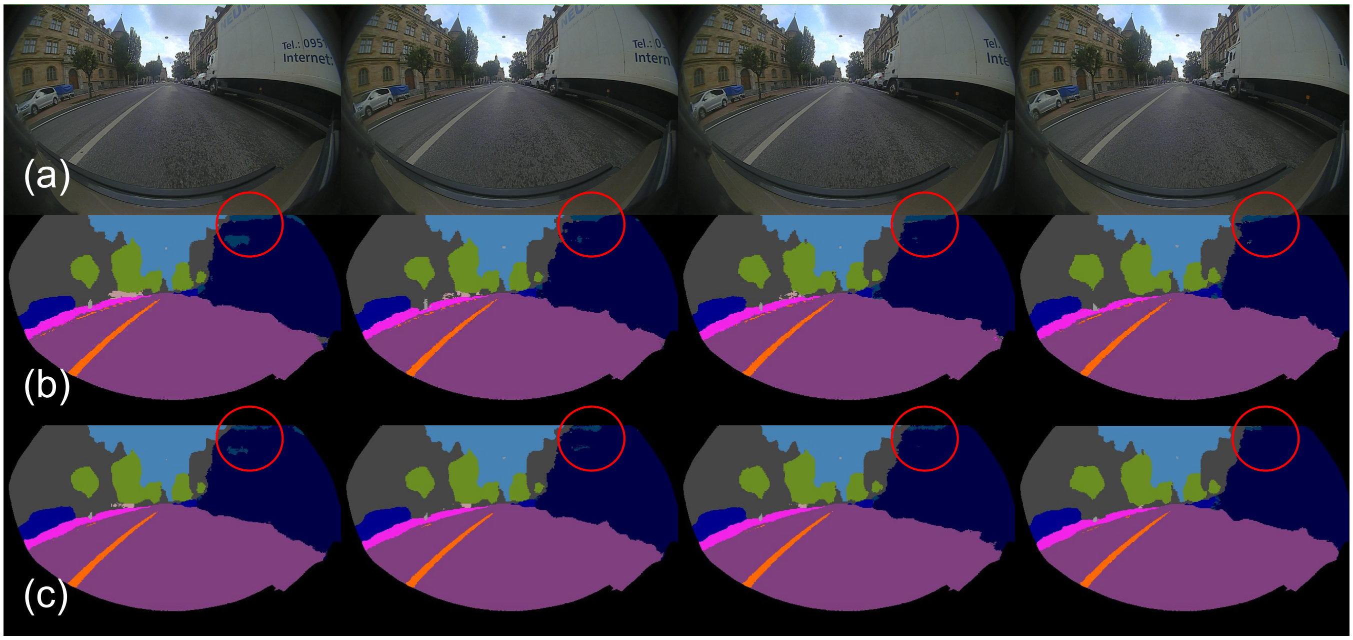

Figure 5: Qualitative results from our *clean* data set. 4 images are shown in consecutive time steps. (a) input image, (b) baseline, (c) our approach. The class *truck* is predicted much more stable compared to the baseline.

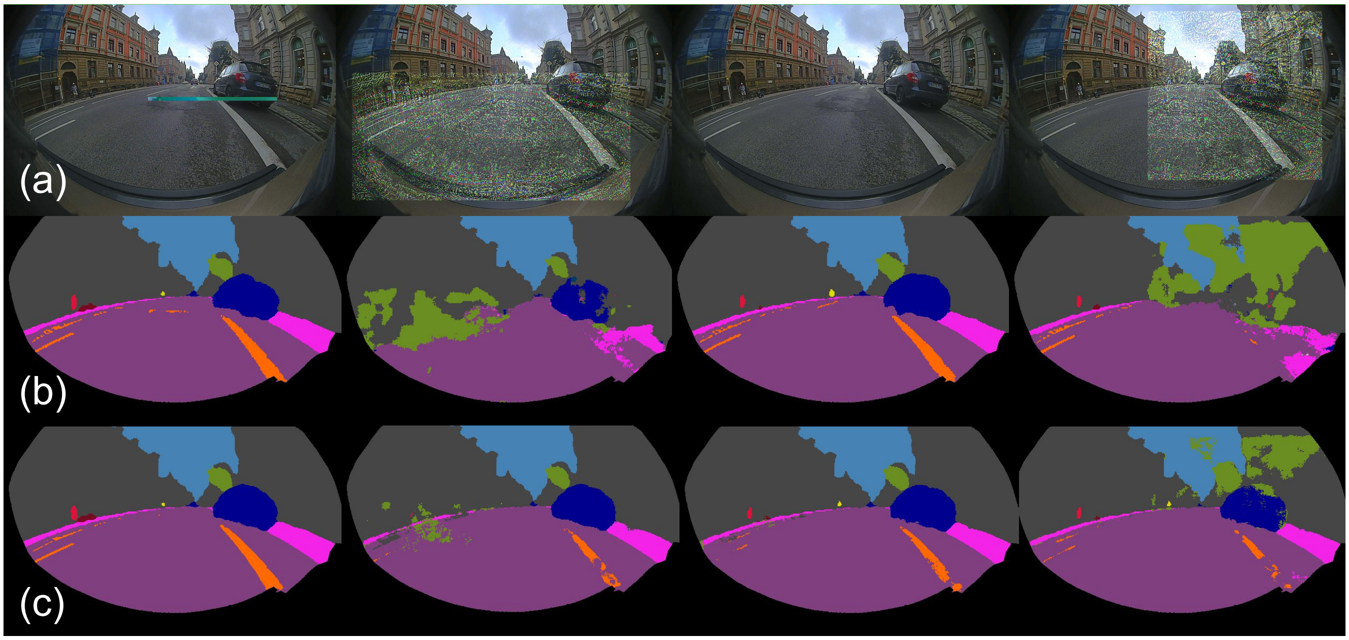

Figure 6: Qualitative results from our *perturb* data set. 4 images are shown in consecutive time steps. (a) input image, (b) baseline, (c) our approach. The perturbation pattern in columns 2 and 4 drastically destroys the prediction of the baseline, while our approach drastically reduces the influence of the perturbation on the prediction. Please note that the perturbation patterns in the image are amplified for visualization reasons.

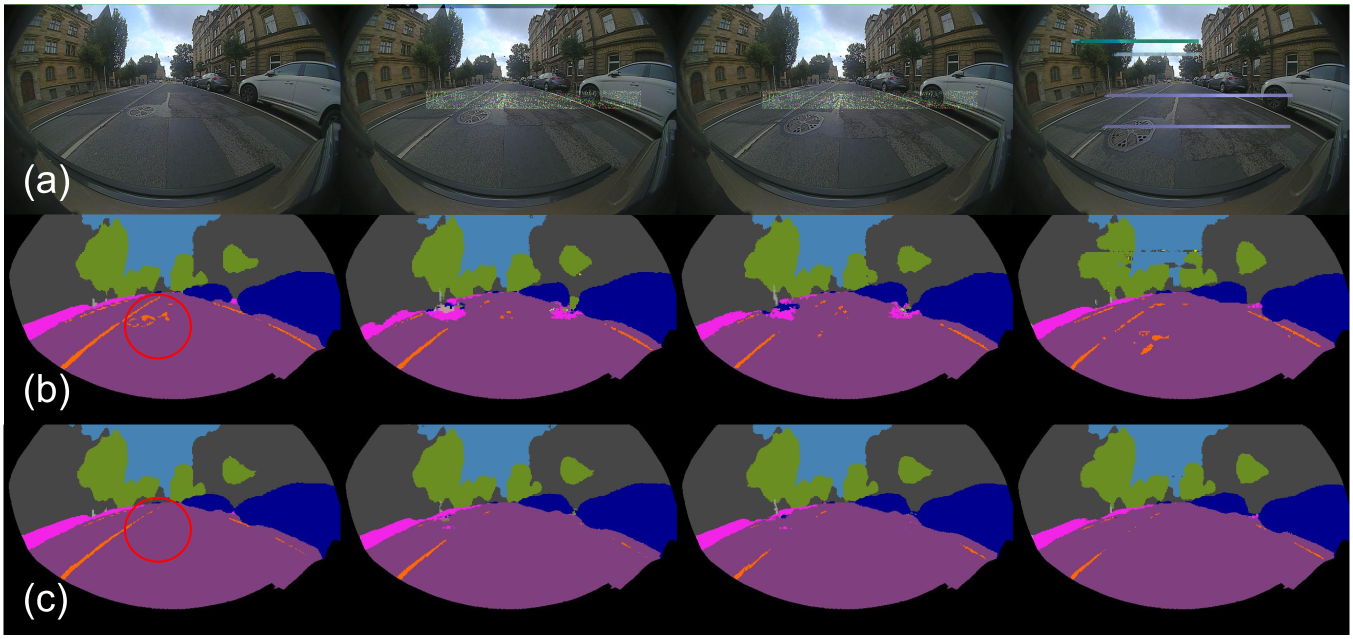

Figure 7: Qualitative results from our *perturb* data set. 4 images are shown in consecutive time steps. (a) input image, (b) baseline, (c) our approach. In the first column the class *road marking* is wrongly detected by the baseline. In the other columns it can be seen that the perturbations in the input data affect our approach much less. Please note that the perturbation patterns in the image are amplified for visualization reasons.

perturbations in the input data, our approach can drastically increase the robustness of the prediction. But even under normal conditions a more stable prediction can be achieved, which we have shown qualitatively and quantitatively. We see considerable potential for improvement in our approach through better uncertainty modeling. The knowledge of the exact localization of the image regions where the DNN is uncertain is a crucial point for the effectiveness of our approach. For this reason we plan to replace the calibrated probabilities from softmax distributions with different types of uncertainty modelling.

REFERENCES

[1] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. 2015. Material recognition in the wild with the materials in context database. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 3479–3487.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062* (2014).

[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[4] Mohsen Fayyaz, Mohammad Hajizadeh Saffar, Mohammad Sabokrou, Mahmood Fathy, Reinhard Klette, and Fay Huang. 2016. Stfcn: Spatio-temporal fcn for semantic video segmentation. *arXiv preprint arXiv:1608.05971* (2016).

[5] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. 2017. Semantic video cnns through representation warping. In *Proceedings of the IEEE International Conference on Computer Vision.* 4453–4462.

[6] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with LSTM. (1999).

[7] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70.* JMLR. org, 1321–1330.

[8] Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136* (2016).

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2462–2470.

[11] Abhijit Kundu, Vibhav Vineet, and Vladlen Koltun. 2016. Feature space optimization for semantic video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3168–3175.

[12] Buyu Liu, Xuming He, and Stephen Gould. 2015. Multi-class semantic video segmentation with exemplar-based object reasoning. In *2015 IEEE Winter Conference on Applications of Computer Vision.* IEEE, 1014–1021.

[13] David Nilsson and Cristian Sminchisescu. 2018. Semantic video segmentation by gated recurrent flow propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 6819–6828.

[14] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. 2016. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147* (2016).

[15] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. 2016. Clockwork convnets for video semantic segmentation. In *European Conference on Computer Vision.* Springer, 852–868.

[16] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision.* 4489–4497.

[17] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2016. Deep end2end voxel2voxel prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops.* 17–24.

[18] Subarna Tripathi, Serge Belongie, Youngbae Hwang, and Truong Nguyen. 2015. Semantic video segmentation: Exploring inference efficiency. In *2015 International SoC Design Conference (ISOCC).* IEEE, 157–158.

[19] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. 2018. Understanding convolution for semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV).* IEEE, 1451–1460.

[20] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. 2017. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE International Conference on Computer Vision.* 1369–1378.

[21] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2881–2890.

[22] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. 2015. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision.* 1529–1537.