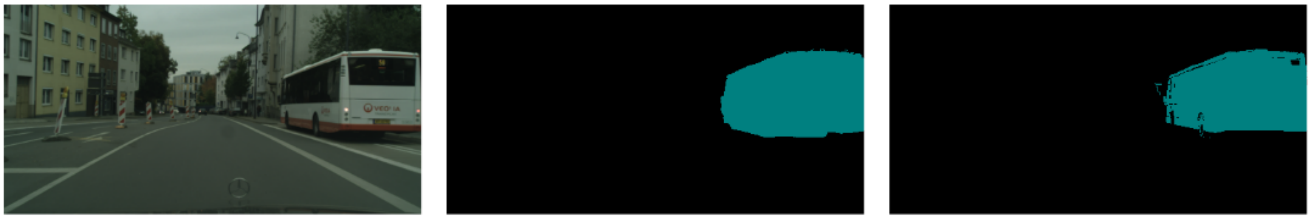
Fast Bilateral Solver for Semantic Video Segmentation

Max Wang
Stanford University
mwang07@stanford.edu

Shannon Kao
Stanford University
kaos@stanford.edu



Segmentation pipeline: input RGB image (left), CRFasRNN segmented output (center), per-pixel labels smoothed by fast bilateral solver (right).

Abstract

We apply the fast bilateral solver technique to the problem of real-time semantic video segmentation. While structured prediction by a dense CRF is accurate on video datasets, the performance is not adequate for real-time segmentation. We hope to utilize the efficient smoothing methodology from the fast bilateral solver within the video segmentation framework introduced by Kundu et al. *[9], improving the accuracy of the segmentation, maintaining temporal and spatial coherence, and achieving the speedups necessary for real-time performance.*

1. Introduction

The human visual system operates, by default, on real time video input. Artificial systems that hope to function in the physical world must be able to similarly process video-based input, in real time, in order to successfully navigate a task. This motivation fuels the current efforts on performing semantic video segmentation, and the challenges are twofold. First, accuracy becomes more difficult because predictions need to be consistent both spatially, within a frame, and temporally, across frames. Second, computational cost also increases, as video sequences are fundamentally more data-heavy than individual images. If AI systems are to function in the real world, they must be capable of overcoming these challenges for handling live video input. For certain applications such as the self-driving car, it is imperative that the segmentation be both extremely fast and extremely accurate.

Semantic image segmentation has seen a recent shift towards structured prediction as a means of achieving the required spatial coherence. The fully-connected, or dense CRF [6] explicitly enforces this label coherence by incorporating pairwise pixel potentials, along with the unary potentials for individual pixels, to encode contextual information about the image. Applying the dense CRF to existing image segmentation models [11] has demonstrable gains in accuracy [3].

In semantic video segmentation, we face the added challenge of temporal coherency, or consistency across frames. This is further complicated by the fact that both the scene and the camera itself may be in motion. Kundu *et al.* [9] introduce a technique which optimizes the dense CRF feature space such that the Euclidean distance between pixels in the feature space is a more accurate representation of their correspondence in the scene. Applying the dense CRF to video volumes significantly increases the accuracy of the semantic video segmentation techniques explored.

However, the feature space optimization proposed involves inference over a series of overlapping cliques, in order to calculate the pairwise potential between pixels. Kundu *et al.* [9] introduce a mean field approximation, Q, similar to the approximation used in the initial dense CRF model [6]. The fast bilateral solver [1] is recent breakthrough in edge-aware smoothing, which offers a bilateral-space optimization that is general, differentiable, and significantly faster than inference over the dense CRF. We propose a system that applies the techniques introduced in the fast bilateral solver to the space of semantic video segmentation. The fast bilateral solver offered an 8-10x speedup on state-of-the-art semantic image segmentation, and we believe that with full parallelization, this technique will result in a close to real-time semantic video segmentation system.

Our system will use the fast bilateral solver in place of the energy function introduced in Kundu *et al.* [9]s semantic video segmentation. This contains an approximation for both the pixel-specific unary and the pairwise potential containing contextual information. Then, as described in [1], we will use conjugate gradient descent methods to solve the matrix inversion required by the bilateral solver.

We fit this bilateral solver into the end-to-end framework introduced in [9]. The unary terms that are input into our system will be the output of a CNN, using weights and data obtained from [15]. While [15] proposes an RNN to refine the output of their initial segmentation, we will directly use the CNN output, relying on the bilateral solver to perform fast and accurate smoothing. We will be training and testing our system on the Cityscape dataset [4], which has segmented video frames for semantic segmentation, with pretrained weights obtained using the PASCAL VOC dataset.

2. Previous Work

Previous approaches to the semantic segmentation problem have centered around CNNs, with a more recent shift towards using CRFs and structured prediction. In this section we will review approaches to semantic segmentation, as well as the state of the art techniques for video segmentation. Finally, we will discuss the state of real-time models in the space of semantic segmentation.

2.1. Semantic Segmentation

The core goal of a semantic segmentation system is to match each pixel within an image to one of a set of predetermined semantic labels, representing objects. This problem is often posed as MAP inference in a CRF over individual pixels, or image patches [5, 8]. These CRF potentials are comprised of terms that can encode relationships between object classes, as well as smoothness terms to encourage label coherence.

Recent developments indicate that using a fully-connected CRF [6, 7], which considers the pairwise potentials of all pixel pairs across the image, rather than individual pixels or regions, offers significantly more accurate results.

2.2. Segmentation in Video

Semantic segmentation in video has always lagged a little behind static image segmentation, due to the increased complexity of the video medium. Approaches often focus almost exclusively on either temporal coherence [3, 12], while methods that tackle object-level coherence have limitations. Structure from motion models, for example, are only able to parse moving objects [2]. While there have been video segmentation methods that do track object-level, from the unified MRF model of Wang *et al.* [14] to the optical flow system developed by Lezama *et al.* [10], these systems only handle single-class foreground objects.

	Dense CRF	Fast Bilateral Solver
Image	[6]	[1]
Video	[9]	Our contribution

Figure 1. The current state of image and video segmentation. Krahenbuhl *et al.* [6] introduced the dense CRF technique, which Kundu *et al.* [9] applied to the space of video segmentation. Barron *et al.* [1] introduced the fast bilateral solver, which we will apply to real-time video segmentation.

Kundu *et al.* [9] applied structured prediction via the dense CRF to the video segmentation problem, with impressive results. Their feature-space optimization technique allows both accurate temporal coherence with overlapping cliques spanning blocks of frames, as well as spatial, object-level coherence within a frame. However, while the dense CRF models are accurate, they are nowhere near efficient enough to provide real-time semantic video segmentation.

2.3. Real-time Video Segmentation

In order to achieve the speedups necessary for real-time semantic segmentation, approaches often sacrifice a certain amount of accuracy. We believe that by applying the results of the fast bilateral solver [1] we will be able to produce a real-time semantic video segmentation with high-quality output, comparable to the dense CRF.

3. Technical Details

We have implemented an end-to-end system that, given a video sequence, outputs semantic labels for spatially and temporally coherent segments of the sequence. This system has two key stages:

1. Segmentation: the generation of the initial semantic label hypothesis and corresponding confidence, performed by CNN.
2. Smoothing: an edge-aware smoothing performed by the fast bilateral solver, to create the final, temporally and spatially coherent labeled output.

3.1. Semantic Video Segmentation

This work applies the same video segmentation algorithm introduced in the feature space optimization described by [9]. Their model is a set of overlapping cliques, each covering a block of frames within the video volume. Each block in the original model has a corresponding dense CRF

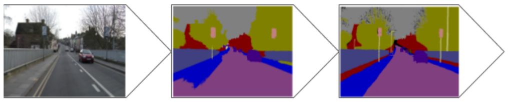


Figure 2. Simplification of the system pipeline. An input image is extracted from a video, segmented into semantic objects, then smoothed using the bilateral solver to achieve near ground-truth accuracy.

defined over it. Each pixel, then, has a Gibbs distribution $P(x|\mathbf{P})$ (Eq 1) and corresponding Gibbs energy $E(x|\mathbf{P})$ (Eq 2) where $\mathbf{P}$ is the set of pixels in the video and x is a mapping from $\mathbf{P}$ to a set of label assignments.

$$P(x|\mathbf{P}) = \frac{1}{Z(\mathbf{P})} \exp(-E(x \mid \mathbf{P})) \tag{1}$$

$$E(x|\mathbf{P}) = \sum_{\mathbf{p}} \phi^u_{\mathbf{p}}(x_{\mathbf{p}}) + \sum_{(\mathbf{p},\mathbf{q})\in\varepsilon} \phi^p_{\mathbf{p},\mathbf{q}}(x_{\mathbf{p}}, x_{\mathbf{q}}) \tag{2}$$

This energy equation (Eq 2) is comprised of two key components per pixel, a unary term $\phi^u_{\mathbf{p}}$ describing the cost of assigning a specific label to pixel $\mathbf{p}$, and a pairwise term $\phi^p_{\mathbf{p},\mathbf{q}}$ encoding the context of the pixel to enforce coherency.

The pairwise term is calculated over all pixels in the entire image, using the dense CRF formulation introduced by [6].

$$\phi^p_{\mathbf{p},\mathbf{q}}(x_{\mathbf{p}}, x_{\mathbf{q}}) = \mu(x_{\mathbf{p}}, x_{\mathbf{q}}) \sum_{m=1}^{M} w^m k^m(\mathbf{f}_{\mathbf{p}}, \mathbf{f}_{\mathbf{q}}) \tag{3}$$

As described in [9], the pairwise potential is computed using a label compatibility term, μ, sum over the neighboring pixels of the mixture weights w^m and the kernel of the feature vectors for pixels $\mathbf{p}$ and $\mathbf{q}$.

3.2. Fast Bilateral Solver

The fast bilateral solver introduced by Barron *et al.* [1] offers an efficient, appealing alternative to the fully-connected CRF. The bilateral solver can be formulated as an optimization problem, as shown below.

$$\min_x \frac{\lambda}{2} \sum_{i,j} \hat{W}_{i,j}(x_i - x_j)^2 + \sum_i c_i(x_i - t_i)^2 \tag{4}$$

It is comprised of two key components, similar to the Gibbs energy equation described above. The second sum expresses a confidence, c_i, in the label assignment, which is proportional to the confidence unary, $\phi^u_{\mathbf{p}}$, introduced in [9]. The first term applies a bistochiasized bilateral kernel, $\hat{W}_{i,j}$, to the Euclidean distance between two pixels.

$$\begin{aligned} W_{i,j} = \exp(&-\frac{\|[p^x_i, p^y_i]\| - \|[p^x_j, p^y_j]\|^2}{2\sigma^2_{xy}} \\ &- \frac{(p^l_i - p^l_j)^2}{2\sigma^2_l}) \\ &- \frac{\|[p^u_i, p^v_i]\| - \|[p^u_j, p^v_j]\|^2}{2\sigma^2_{uv}} \end{aligned} \tag{5}$$

Here, pi_i indicates pixel i in the RGB input, and its five-dimensional feature vector consisting of color (p^l_i, p^u_i, p^v_i) and position (p^x_i, p^y_i). The denominator of each term holds a σ value which controls the extent to which each feature influences the blur. This bilateral kernel then can be structured such that it is proportional to the pairwise potentials introduced in the previous energy equation [13].

By replacing the energy equation in [9] with the equation formulated in the fast bilateral solver, we can now express this problem as a quadratic optimization. This can be solved efficiently using matrix inversion, which we further optimize by implementing conjugate gradient descent.

3.3. Implementation

Our system takes as sequence of video frames. It then runs semantic segmentation on each individual frame, generating a confidence value per label, for each pixel in the frame. This vector of confidence values is then passed to the fast bilateral solver, which smooths the noisy input. The final output is a sequence of frames with smoothed semantic labeling for each object in the frame.

3.3.1 Video processing

Light initial pre-processing of the video frames is necessary: the system first reads in a frame, resizes it appropriately for the CRF as RNN model, then segments this frame image.

3.3.2 Semantic segmentation

Our semantic segmentation is based off of the work done in the CRF as RNN system [15].

We used Caffe and Python to accomplish the implementation of the CRF as RNN portion of the pipeline. We adapted a model that is pre-trained on the PASCAL dataset and modified it for usage on the Cityscapes dataset. Due to physical memory issues during training, we chose to focus primarily on the classes for cars, buses, and pedestrians. The Cityscapes dataset provides per-pixel ground truths that we used as an input to our evaluation system, in addition to the colored images.

When making a prediction, the CRF as RNN takes a colored image and provides a high dimensional vector that contain probabilities of each pixel with respect to each class label. This high dimensional probability vector is then used as the input to our bilateral solver.

We also attempted segmentation of other classes but the results were suboptimal. This is discussed in the results section below.

3.3.3 Fast bilateral solver

As seen in figure 3, the bilateral solver takes as input an RGB image, a target image, and a confidence matrix corresponding to the target image. The solver then performs a fast, edge-aware blur, resulting in the final, smoothed image in the bottom right quadrant of the figure.

We implemented the fast bilateral solver on the CPU, using conjugate gradient descent to perform the matrix inversion. The three inputs are, first, an RGB reference image, which guides the blur. Second, a target image, containing the values to be filtered. Lastly an image representing a confidence level per pixel in the target image. For this, we simply used a uniform confidence, as described in [1]. We first implemented blurring for depth superresolution; essentially a system that blurred a single depth value per pixel, as shown in figure 3. It was straightforward, then, to extend this to blur an N-dimensional vector of labels, where N is the number of semantic labels in the dataset. Each element of the vector is a confidence value for that specific label. This target image is exactly the output of the CRF as RNN system.

The output of this modified bilateral solver, then, is the the smoothed version of the initial label hypotheses. We take the argmax of this vector to obtain the predicted label for a particular pixel.

4. Experiments

We used the Cityscapes dataset [4] used by [9] for evaluation, so as to better compare the performance of the two systems. This dataset is primarily used for road scene understanding, and provides video input along with pixel-level semantic annotations of selected frames.

4.1. Dataset

The Cityscapes dataset is relatively new, and used primarily for outdoor scene understanding. It consists of driver's perspective of urban environments with roads, buildings, and outdoor objects. This dataset is particularly applicable, because the overarching end goal of this system is segmentation for the self-driving car. The dataset contains 2975 training images, 500 validation images, and 1525 test images. There are 30 semantic classes represented in total.

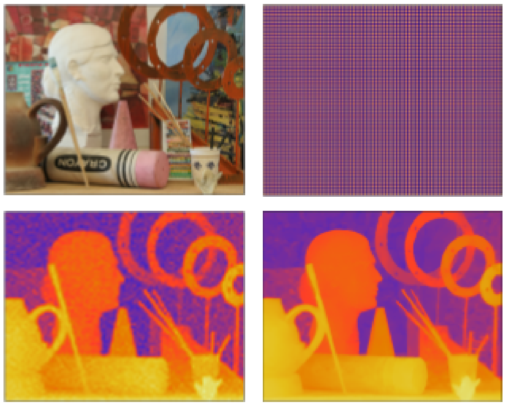


Figure 3. Bilateral solver. Top left: input image. Top right: confidence levels. Bottom left: bilateral filter output. Bottom right: bilateral solver output.

4.2. Segmentation

We use intersection over union (IoU) as the primary metric for evaluating the system. This is calculated on a per-label basis, and formulated as the intersection between the labelled pixels (true positives) divided by the union of labelled pixels (true positives + false positives + false negatives).

	CRF as RNN Only	With Fast Bilateral Solver
Car	0.502	0.536
Bus	0.597	0.425
People	0.111	0.169

We found that our system was generally comparable to the CRF as RNN in terms of accuracy, though slightly less so in the car and person categories. On visual examination of the results, it became evident that the bilateral solver would remove the labelling from areas within the object that had high color variation, such as car windows with reflections, or multi-colored clothing (figure 4). This greatly hurt our evaluation score, even though our results showed much sharper edges and clearer segmentation at contour boundaries.

4.3. Environment Labels

We initially focused on vehicles and pedestrians, finding that cars and buses had the best segmentation results, on visual inspection. We attempted to extend our scope to include environmental labels, such as buildings and roads. However, we found that the CRF as RNN often labeled most of the image as ”unknown” as the probabilities for other classes were much lower. This is why in the images we evaluated, cars and buses were most prominent.

We further analyzed this by disabling the ”unknown” label, forcing the network to return alternative labels. The

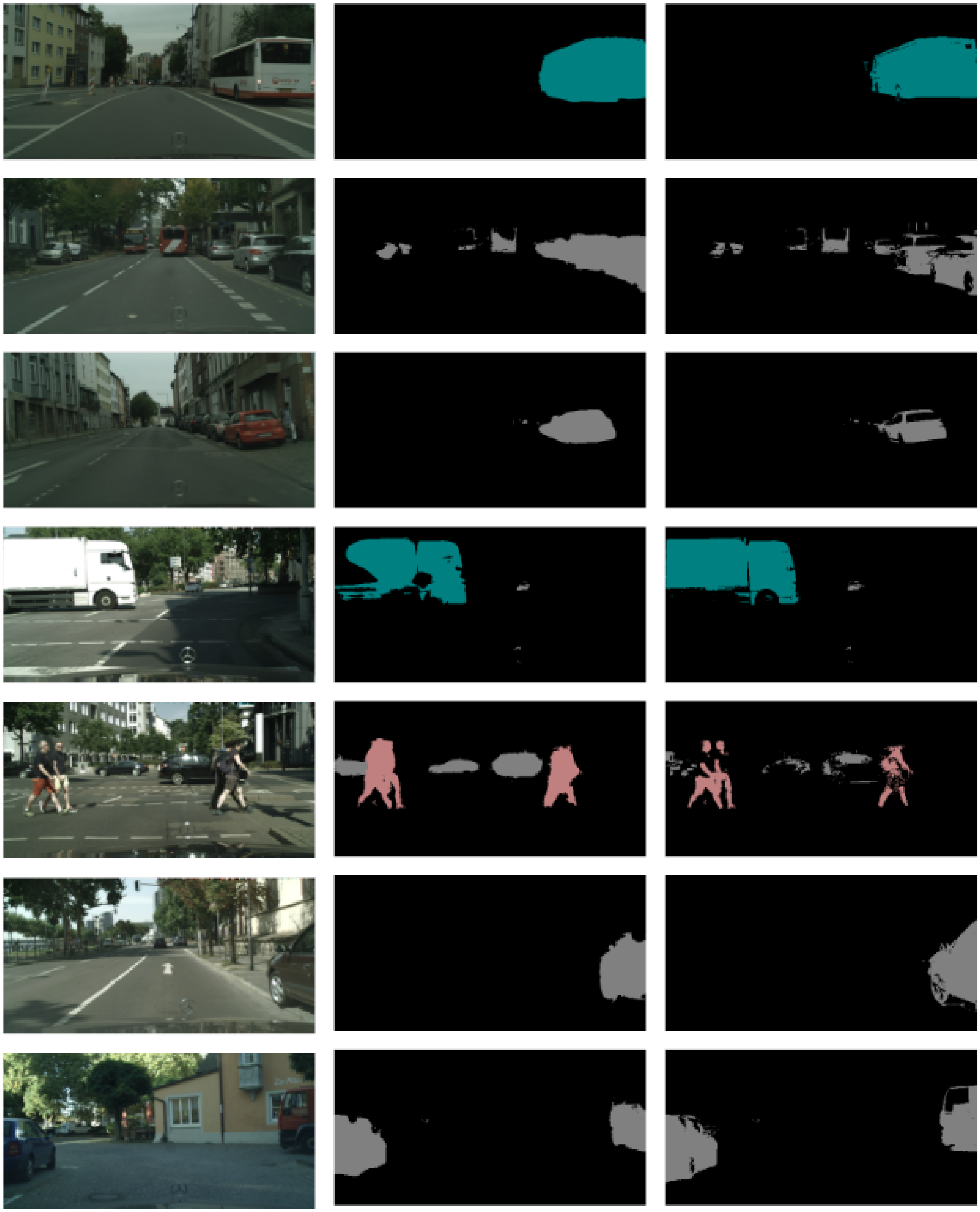


Figure 4. RGB input images (left), the output of the CRF as RNN (center), and the CRF as RNN output after the fast bilateral solver has been applied (right). On both car objects and person objects it's clear that the fast bilateral solver removes areas with high color variation (namely windows and clothing).

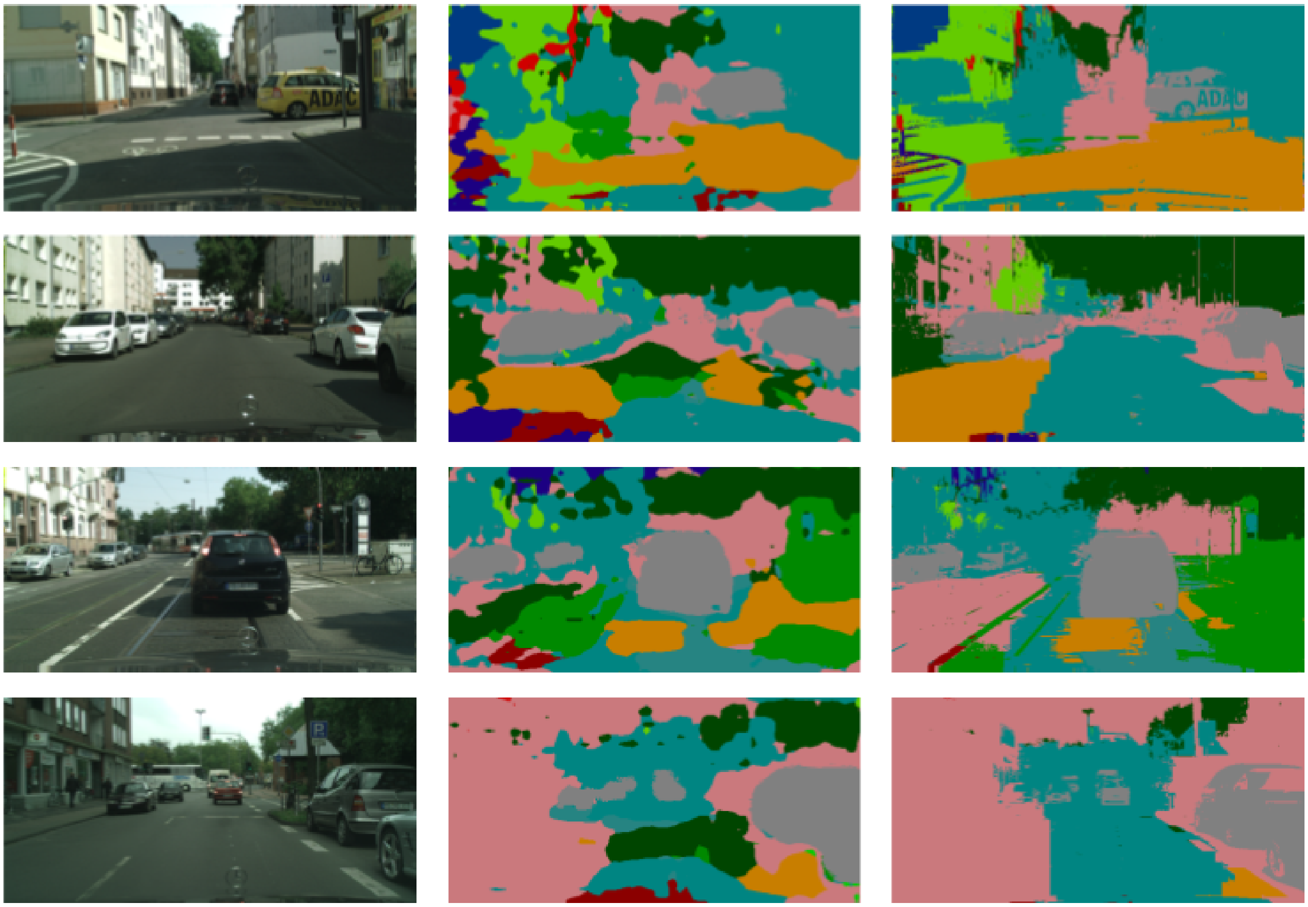


Figure 5. RGB input images (left), the multi class segmentation of CRF as RNN (center) and the multi-class segmentation of CRF as RNN and bilateral solver (right).

resulting data demonstrated that the network had very little confidence on the other classes. We suspect this was due to training being prematurely terminated on the server for exceeding hardware memory capacity. Its peak memory consumption was well over 10GB of RAM, which we did not have. This shortcoming in training means that the network was quite skilled at detecting classes in the PASCAL dataset, but not for classes that solely appeared in the Cityscapes dataset. Regardless, we ran the predictions without the ”unknown” label, as shown in figure 5. We noticed that even with the poor segmentation from the CRF as RNN, the bilateral solver was able to retrieve a much more accurate segmentation using the RGB image as reference. The system roughly understood which clusters of pixels should belong to the same class, perhaps due to an underlying understanding of pixel-to-object representation that exist in both the CRF as RNN and the fast bilateral solver portions. This tells us that we could get very accurate multi class segmentation results with further refinement.

Regarding the memory issue during trainig: We tried to migrate to a more suitable server but ran into other hardware issues regarding permissions.

4.4. Runtime

Although the system is not yet fully parallelized, we took some initial runtime metrics. This is useful both as a baseline for future iterations of this project, and for preliminary comparison with other system. The average time, per frame, to segment, then run the bilateral solver (including the time for image IO), was 2.967 seconds. We anticipate that further optimization and image IO removal can greatly decrease this runtime.

5. Future Work

This project is a proof-of-concept, which demonstrates that the proposed pipeline of segmentation, followed by the fast bilateral solver, is very promising and worth exploring further. There are several immediate next steps that can be taken to generate stronger results with this system.

5.1. End-to-end Training

We were unable to resolve our hardware issues in order to fully train the network, but end-to-end training is necessary to report accurate results for the finalized system, especially for multi-class segmentation. It may also resolve our issue of objects having high color variations. We believe that further refinement will leverage from the current understanding of representation and our system will become much better.

5.2. Optimization

In order to achieve fully real-time semantic video segmentation, an immediate next step is to parallelize the system. The current algorithm will be implemented in CUDA in order to achieve the speedup necessary for the required performance. In addition, a more integrated pipeline would remove the current, time-consuming intermediate steps that exist between the CRF as RNN portion and the fast bilateral solver portion of our system.

6. Conclusion

In conclusion, we find that using the fast bilateral solver as a more efficient alternative to the dense CRF in the semantic video segmentation pipeline has very promising initial results. The effectiveness of the fast bilateral solver is already evident in how clearly it defines the boundaries of objects even based solely on color data. We hope to continue this project and fully optimize this system for both accuracy and potential.

References

[1] J. T. Barron and B. Poole. The fast bilateral solver.

[2] G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. 2008.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs.

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.

[5] X. He, R. S. Zemel, and M. A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. 2004.

[6] P. Krahenbuhl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials, 2011.

[7] P. Krahenbuhl and V. Koltun. Parameter learning and convergent inference for dense random fields, 2013.

[8] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. 2005.

[9] A. Kundu, V. Vineet, and V. Koltun. Feature space optimization for semantic video segmentation. 2016.

[10] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues.

[11] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. 2015.

[12] O. Miksik, D. Munoz, J. A. Bagnell, and M. Hebert. Efficient temporal consistency for streaming video scene analysis.

[13] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 839–, Washington, DC, USA, 1998. IEEE Computer Society.

[14] C. Wang and N. Paragios. Segmentation, ordering and multi-object tracking using graphical models. 2009.

[15] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr. Conditional random fields as recurrent neural networks.