# Semantic Video Segmentation by Gated Recurrent Flow Propagation

David Nilsson[1] and Cristian Sminchisescu[1,2]
[1]Department of Mathematics, Faculty of Engineering, Lund University
[2]Institute of Mathematics of the Romanian Academy
{david.nilsson, cristian.sminchisescu}@math.lth.se

## Abstract

*Semantic video segmentation is challenging due to the sheer amount of data that needs to be processed and labeled in order to construct accurate models. In this paper we present a deep, end-to-end trainable methodology for video segmentation that is capable of leveraging the information present in unlabeled data, besides sparsely labeled frames, in order to improve semantic estimates. Our model combines a convolutional architecture and a spatio-temporal transformer recurrent layer that is able to temporally propagate labeling information by means of optical flow, adaptively gated based on its locally estimated uncertainty. The flow, the recognition and the gated temporal propagation modules can be trained jointly, end-to-end. The temporal, gated recurrent flow propagation component of our model can be plugged into any static semantic segmentation architecture and turn it into a weakly supervised video processing one. Our experiments in the challenging CityScapes and Camvid datasets, and for multiple deep architectures, indicate that the resulting model can leverage unlabeled temporal frames, next to a labeled one, in order to improve both the video segmentation accuracy and the consistency of its temporal labeling, at no additional annotation cost and with little extra computation.*

## 1. Introduction

Systems capable of computing accurate and temporally consistent semantic segmentations in video are central to scene understanding, being useful in applications in robotics, for instance grasping, or for autonomous vehicles where one naturally works with videos rather than single images, and high levels of precision are needed. Since the emergence of deep learning methods for image classification, the problem of semantic image segmentation has received increasing attention, with some of the most successful methods based on fully trainable convolutional architectures (CNN). Data for training and refining single frame, static models is now quite diverse [7, 29]. In contrast, fully trainable approaches to semantic video segmentation face the difficulty of obtaining detailed annotations for individual video frames, although datasets are emerging for the (unsupervised) video segmentation problem [11, 36, 27]. Therefore some of the existing approaches to semantic video segmentation [42, 43, 25] rely on single frame models with corresponding variables connected in time using random fields with higher-order potentials, and mostly prespecified parameters. Fully trainable approaches to video are rare. The computational complexity of video processing further complicated matters.

One possible approach to designing semantic video segmentation models in the long run can be to only label frames, sparsely, in video, as it was done for static datasets[7, 29]. Then one should be able to leverage temporal dependencies in order to propagate and aggregate information in order to decrease uncertainty during *both* learning and inference. This would require a model that can integrate spatio-temporal dependencies across video frames.

While approaches based on CNNs appear right, they are non-trivial to adapt to video segmentation due to the amount of data that needs to be processed for dense predictions. If video processing and temporal matching were to be learned without explicit components such as optical flow warping, one possibility would be to design a model based on 3D-convolutions, as used e.g. for action recognition[20, 3]. To our knowledge no such approach has been pursued for semantic video segmentation. Instead, we will take an explicit modeling approach relying on existing single-frame CNNs augmented with spatial transformer structures that implement warping along optical flow fields. These will be combined with adaptive recurrent units in order to learn to fuse the estimates from single (unlabeled) frames with the labeling information temporally propagated from nearby ones, properly gated based on their uncertainty. The proposed model is differentiable and end-to-end trainable.

## 2. Related Work

Our semantic video segmentation work relates to the different fields of semantic image segmentation, as well as,

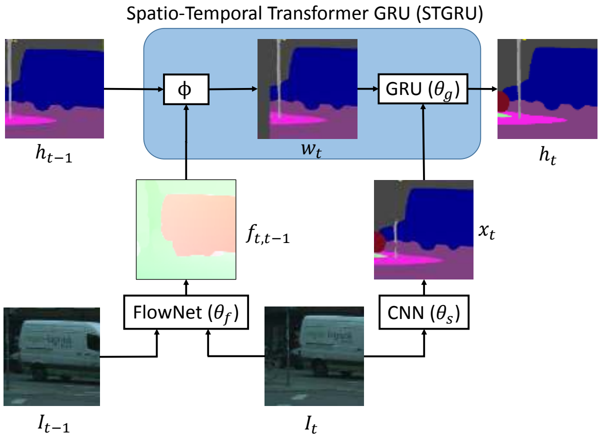

Figure 1. Overview of our Spatio-Temporal Transformer Gated Recurrent Unit (STGRU), combining a Spatial Transformer Network (§3.1) mapping $\phi$ for optical flow warping with a Gated Recurrent Unit (§3.2) to adaptively propagate and fuse semantic segmentation information over time. E.g. the pole is not accurately segmented by the static network ($x_t$), but combining $x_t$ with the segmentation at the previous timestep ($h_{t-1}$) gives a more accurate estimate $h_t$.

more remotely, to unsupervised video segmentation. We will here only briefly review the vast literature with some attention towards formulations based on deep architectures which represent the foundation of our approach.

Many approaches start with the network in [24, 39] and refine it for semantic segmentation. In [15] residual connections are used making it possible to increase depth substantially. [32] obtained semantic segmentations by turning a network for classification [39] into a dense predictor by computing segmentations at different scales and combining all predictions. The network was made fully convolutional. Another successful approach is to apply a dense conditional random field (CRF) [23] as a post-processing step on top of individual pixel or frame predictions. [4] use a fully convolutional network to predict a segmentation and then apply the dense CRF in post processing. [46] realized that inference in dense CRFs can be formulated as a fixed point iteration implementable as a recurrent neural network. Another successful approach is the deep architecture of [44] where max pooling layers are replaced with dilated convolutions. The network was extended by introducing a context module where convolutions with increasingly large dilation sizes are used.

Video segmentation has received significant attention starting from early methodologies based on temporal extensions to normalized cuts [38], random field models and tracking [42, 26], motion segmentation [34] or efficient hierarchical graph-based formulations [14, 43]. More recently, proposal methods where multiple figure-ground estimates or multipart superpixel segmentations are generated at each time-step, then linked through time using optical flow [27, 1, 35], have become popular.

The dense CRF of [23] has been used for semantic video segmentation [25, 41], most notably by [25] using pairwise potentials based on aligning the frames using optical flow. Along similar lines as our earlier version of this work[33], [10] independently present an end-to-end trainable system for semantic video segmentation that warps two-frame intermediate representations in a CNN. We differ in that we warp the segmentation outputs and we can use multiple frames forward and backward in time. In a similar fashion, [30] combines semantic segmentations by means of optical flow warping for body part segmentation in videos. In [19], video propagation is performed by filtering in a bilateral space instead of using optical flow to connect frames temporally. The temporal matching can also be performed using superpixels and optical flow, as in [16], where information in matched regions is pooled using Spatio-Temporal Data-Driven Pooling (STD2P). [21] use GANs [13] to first predict future video frames in an unsupervised manner and then use the learned features for semantic video segmentation. In [37] observe that intermediate representations in a CNN change slowly in video, and present a method to only recompute features when there is enough change, leading to significant speed-ups.

## 3. Methodology

A visual illustration of how our semantic video segmentation model aggregates information in adjacent video frames is presented in fig. 1. We start with a semantic segmentation at the previous time step, $h_{t-1}$ and warp it along the optical flow to align it with the segmentation at time $t$, by computing $w_t = \phi_{t-1,t}(h_{t-1})$ where $\phi$ is a mapping of labels along the optical flow. This is fed as the hidden state to a gated recurrent unit (GRU) where the other input is the estimate $x_t$ computed by a single frame CNN for semantic segmentation. The information contained in $w_t$ and $x_t$ has significant redundancy, as one expects from nearby video frames, but in regions where it is hard to find the correct segmentation, or where significant motion occurs between frames, they might contain complementary roles. The final segmentation $h_t$ combines the two segmentations $w_t$ and $x_t$ by means of learnt GRU parameters and should include segments where either of the two are very confident. Our model is end-to-end trainable and we can simultaneously refine the GRU parameters $\theta_g$, the parameters of the static semantic segmentation network $\theta_s$ and the parameters of the FlowNet $\theta_f$.

Our overall video architecture can operate over multiple timesteps both forward and backward with respect to the timestep $t$, say, where semantic estimates are obtained. The illustration of this mechanism is shown in fig. 2. In *training*, the model has the desirable property that it can rely only on sparsely labeled video frames, but can take advantage of the temporal coherency in the unlabeled video neighborhoods centered at the ground truth. Specifically, given an estimate of our static (per-image) semantic segmentation model at timestep $t$, as well as estimates prior and posterior to it, we can warp these using the confidence gated optical flow forward and backward in time (using the Spatio-Temporal Transformer Gated Recurrent Unit, STGRU, illustrated in fig. 1) towards timestep $t$ where ground truth information is available, then fuse estimates in order to obtain a prediction. The resulting model is conveniently differentiable. The loss signal will then be used to backpropagate information for training both the parameters of the gated recurrent units ($\theta_{gf}$, $\theta_{gb}$), the parameters of the (per-frame) semantic segmentation network ($\theta_s$) and the parameters of the FlowNet ($\theta_f$). In *testing* the network can operate either statically, per frame, or take advantage of video frames prior and (if available) posterior to the current processing timestep.

Given these intuitions we will now describe the main components of our model: the spatio-temporal transformer warping and the gated recurrent unit, and then describe implementation and training details.

### 3.1. Spatio-Temporal Transformer Warping

We will use optical flow as input to warp the semantic segmentation estimates across successive frames. We extend the spatial transformer network [18] to operate in the spatio-temporal video domain. Elements on a two-dimensional grid $x_{ij}$ will map to $y_{ij}$ according to

$$y_{ij} = \sum_{m,n} x_{mn} k(i + f^y_{ij} - m, j + f^x_{ij} - n), \qquad (1)$$

where $(f^x_{ij}, f^y_{ij})$ is the optical flow vector for the pixel at location $(i, j)$. We will use a bilinear interpolation kernel $k(x, y) = \max(0, 1 - |x|) \max(0, 1 - |y|)$. The mapping is differentiable and we can backpropagate gradients from $y$ to both $x$ and $f$. The sum contains only $4$ non-zero terms when using the bilinear kernel, so it can be computed efficiently. The methodology has been introduced earlier by us [33] and also independently in [17, 30, 10].

### 3.2. GRUs for Semantic Video Segmentation

To connect the probability maps for semantic segmentation at different timesteps, $h_{t-1}$ and $h_t$, we will use a modified convolutional version of the Gated Recurrent Unit [5]. In particular, we will design a gating function based on the flow, so we only trust the semantic segmentation values warped from $h_{t-1}$ at locations where the flow is certain. We also use gating to predict the new segmentation probabilities, taking into account if either $h_{t-1}$ or $x_t$ have high confidence for a certain class in some region of the image.

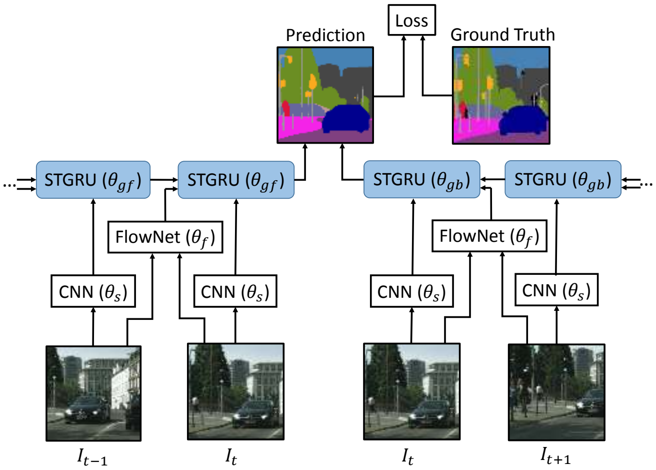

Figure 2. Illustration of our temporal architecture entitled Gated Recurrent Flow Propagation (GRFP) based on Spatio-Temporal Transformer Gated Recurrent Units (STGRU), illustrated in fig.1. The model can integrate both forward only and forward-backward calculations, under separate recurrent units with different parameters $\theta_{gf}$ (forward) and $\theta_{gb}$ (backward). Each of the forward and backward recurrent units have tied parameters across timesteps. The parameters of the semantic segmentation architecture ($\theta_s$) and FlowNet ($\theta_f$) are shared over time. The predictions from the forward model aggregated over frames $t-T, \ldots, t-1, t$ (in the above illustration $T = 1$) and (when available and desirable) backward model aggregated over frames $t + T, \ldots t + 1, t$ are fused at the central location $t$ in order to make a prediction that is compared against the ground truth available only at frame $t$ by means of a semantic segmentation loss function.

To adapt a generic GRU for semantic video segmentation, we first change all fully connected layers to convolutions. The hidden state $h_t$ and the input variable $x_t$ are no longer vectors but tensors of size $H \times W \times C$ where $H$ is the image height, $W$ is the image width and $C$ is the number of channels, corresponding to the different semantic classes. The input $x_t$ is normalized using softmax and $x_t(i, j, c)$ models the probability that label is $c$ for pixel $(i, j)$. We let $\phi_{t-1,t}(x)$ denote the warping of a feature map $x$ from time $t - 1$ to $t$, using optical flow given as additional input, as described in section 3.1. The proposed adaptation of the GRU for semantic video segmentation is

$$w_t = \phi_{t-1,t}(h_{t-1}) \qquad (2)$$
$$r_t = 1 - \tanh(|W_{ir} * (I_t - \phi_{t-1,t}(I_{t-1})) + b_r|) \qquad (3)$$
$$\tilde{h}_t = W_{xh} * x_t + W_{hh} * (r_t \odot w_t) \qquad (4)$$
$$z_t = \sigma(W_{xz} * x_t + W_{hz} * (r_t \odot w_t) + b_z) \qquad (5)$$
$$h_t = \text{softmax}(\lambda(1 - z_t) \odot r_t \odot w_t + z_t \odot \tilde{h}_t), \qquad (6)$$

where $W$ and $b$ denote trainable convolution weights, and biases, respectively. Instead of relying on a generic parametrization for the reset gate $r_t$, we use a confidence measure for the flow by comparing the image $I_t$ with the warped image of $I_{t-1}$. We also discard $\tanh$ when computing $\tilde{h}_t$ and instead use softmax in order to normalize $h_t$. We multiply with a trainable parameter $\lambda$ in order to compensate for a possibly different scaling of $\tilde{h}_t$ relative to the warped $h_{t-1}$ due to the convolutions with $W_{hh}$ and $W_{xh}$. Note that $h_{t-1}$ only enters when we compute the warping $w_t$ so we only use the warped $h_{t-1}$, i.e. $w_t$.

### 3.3. Implementation

For the static (per-frame) component of our model, we rely on a deep neural network pre-trained on the CityScapes dataset and fed as input to the gated recurrent units. We conducted experiments using the Dilation architecture [44], LRR [12] and PSP [45]. The convolutions in the STGRU were all of size 7 × 7. We use the standard log-likelihood loss for semantic segmentation

$$L(\theta) = -\sum_{i,j} \log p(y_{ij} = c_{ij}|I, \theta) \tag{7}$$

where $p(y_{ij} = c_{ij}|I, \theta)$ is the softmax normalized output of the STGRU, estimating the probability of the correct class $c_{ij}$ for the pixel at $(i, j)$. The recurrent network was optimized using Adam [22] with $\beta_1 = 0.95$, $\beta_2 = 0.99$ and learning rate $2 \cdot 10^{-5}$. Due to GPU memory constraints, the per-frame semantic segmentation CNN computations had to be performed one frame at a time with only the final output saved in memory. When training the system end-to-end the intermediate activations for each frame had to be recomputed. We used standard gradient descent with momentum for the experiments where the static networks or flow networks were refined, with learning rate $2 \cdot 10^{-12}$ and momentum $0.95$. Note that the loss was not normalized, hence the small learning rate. We used FlowNet2 [17] for all experiments unless otherwise stated.

**Default setup** The GRFP model we use is, unless otherwise stated, a forward model trained using 5 frames ($T = 4$ in fig. 2) where the parameters of the STGRU $\theta_{gf}$ and the parameters of the static segmentation CNN $\theta_s$ are refined, while the parameters of the FlowNet $\theta_f$ are frozen.

## 4. Experiments

We perform an extensive evaluation on the challenging CityScapes and CamVid datasets, where video experiments nevertheless remain difficult to perform due to the large volume of computation. We evaluate under two different perspectives, reflecting the relevant, key aspects of our method. First we evaluate semantic video segmentation. We will compare our method with other methods for semantic segmentation and show that by using temporal information we can improve segmentation accuracy over a network where the predictions are computed per frame and unlabeled video data is not used. In the second evaluation we use our method to compute semantic segmentations for all frames in a longer video. We will then compare its temporal consistency against the baseline method where the predictions are performed per frame. We will show quantitatively that our method gives a temporally more consistent segmentation compared to the baseline.

| Method | IoU cls | IoU cat |
|---|---|---|
| GRFP(PSP-MSc, FlowNet2) | 81.3 | 90.7 |
| PSP-MSc [45] | 80.9 | 90.5 |
| GRFP(PSP-SSc, FlowNet2) | 80.2 | 90.2 |
| PSP-SSc [45] | 79.7 | 89.9 |
| GRFP(LRR-4x, FlowNet2) | 73.6 | 88.3 |
| LRR-4x [12] | 72.5 | 87.8 |
| GRFP(Dilation10, FlowNet2) | 69.5 | 86.4 |
| Dilation10 [44] | 68.7 | 86.3 |

Table 1. Average class (cls) and category (cat) IoU on the CityScapes validation set for various single frame baselines we tried our model GRFP on. By using our video methodology we can see labelling improvements for all baselines we tried, showing that our method is applicable to many different single frame semantic segmentation CNNs. With SSc and MSc we mean single scale and multi scale testing, see [45] for details.

### 4.1. Semantic Video Segmentation

| Method | IoU cls | IoU cat |
|---|---|---|
| GRFP(PSP-Msc, FlowNet2) | 80.6 | 90.8 |
| NetWarp [10] | 80.5 | 91.0 |
| PSP-Msc [45] | 80.2 | 90.6 |
| PEARL [21] | 75.4 | 89.2 |
| GRFP(LRR-4x, FlowNet2) | 72.9 | 88.6 |
| LRR-4x [12] | 71.8 | 88.4 |
| Adelaide_context [28] | 71.6 | 87.3 |
| DeepLabv2-CRF [4] | 70.4 | 86.4 |
| GRFP(Dilation10, FlowNet2) | 68.1 | 86.6 |
| Dilation10 [44] | 67.1 | 86.5 |
| DPN [31] | 66.8 | 86.0 |
| FCN 8s [32] | 65.3 | 85.7 |

Table 2. Average class (cls) and category (cat) IoU on the CityScapes test set. We use Dilation10, LRR-4x and PSP as baselines, and we are able to improve the average class IoU with 1.0, 1.1 and 0.4 percentage points, respectively. Notice that our GRFP methodology proposed for video is applicable to most of the other semantic segmentation methods that predict each frame independently – they can all benefit from potential performance improvements at no additional labeling cost.

CityScapes [6] consists of sparsely annotated frames. Each labeled frame is the 20th frame in a 30 frame video

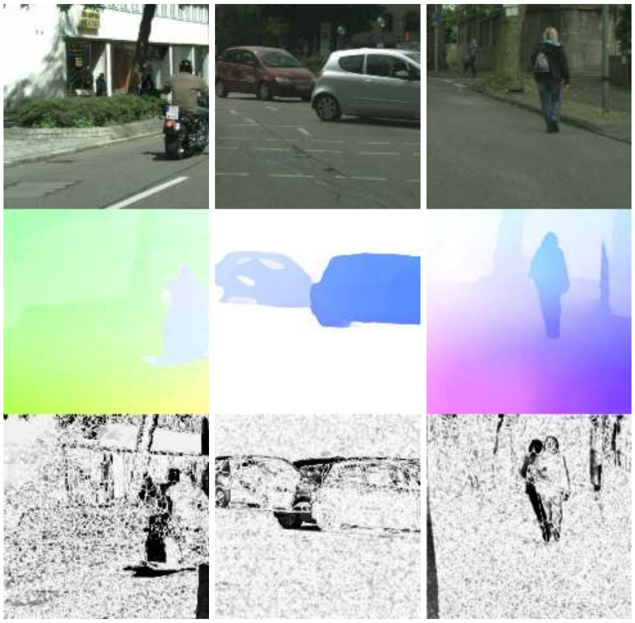

Figure 3. Illustration of the flow gating as estimated by our Spatio-Temporal Transformer Gated Recurrent Unit. We show three examples each containing a video frame, the optical flow and its confidence as estimated by our STGRU model. White regions indicate a confident flow estimate ($r_t = 1$) whereas black regions are uncertain ($r_t = 0$). Occluded regions are black, as expected.

| Frames | IoU cls | IoU cat |
|---|---|---|
| 1 | 68.8 | 86.3 |
| 2 | 69.2 | 86.4 |
| 3 | 69.4 | 86.4 |
| 4 | 69.5 | 86.4 |
| 5 | 69.5 | 86.4 |

Table 3. Average class (cls) and category (cat) IoU on the validation set of CityScapes when using a different number of frames for inference. The model was trained using 5 frames. Notice that using more than one frame improves performance which for this dataset, however, saturates beyond 4 frames.

snippet. There is a total of 2,975 labelled frames in the training set, 500 in the validation set and 1,525 in the test set. We use a forward model with 5 frames and apply the loss to the final frame. Notice however that due to computational considerations, while the STGRU unit parameters $\theta_{gf}$ were trained based on propagating information from 5 frames, the unary network parameters $\theta_s$ were refined based on back-propagating gradient from the 3 STGRU units closest to the loss. The images had size $512 \times 512$ in training, whereas in testing their size was increased to the full resolution $1024 \times 2048$ as more memory was available compared to the training setup.

We used Dilation10 [44], LRR [12] or PSP [45] as backend to our model. We obtain improved performance by using the proposed video methodology compared to the static per-frame baseline for all deep architectures used for static processing. We show the results on the validation set in table 1. In this experiment, we only refined the parameters of the GRU and not the parameters of the PSP network.

| Class | Dilation10 | GRFP(5) | GRFP(1) |
|---|---|---|---|
| Road | 97.2 | 97.3 | 97.1 |
| Sidewalk | 79.5 | 80.1 | 79.2 |
| Building | 90.4 | 90.5 | 90.4 |
| Wall | 44.9 | 50.6 | 46.8 |
| Fence | 52.4 | 53.3 | 53.0 |
| Pole | 55.1 | 55.3 | 55.2 |
| Traffic light | 56.7 | 57.5 | 56.7 |
| Traffic sign | 69.0 | 68.7 | 68.9 |
| Vegetation | 91.0 | 91.1 | 91.0 |
| Terrain | 58.7 | 59.6 | 58.7 |
| Sky | 92.6 | 92.7 | 92.5 |
| Person | 75.7 | 76.2 | 75.7 |
| Rider | 50.0 | 50.3 | 50.1 |
| Car | 92.2 | 92.4 | 92.2 |
| Truck | 56.2 | 57.4 | 55.8 |
| Bus | 72.6 | 73.9 | 72.8 |
| Train | 53.2 | 53.4 | 54.5 |
| Motorcycle | 46.2 | 48.8 | 46.3 |
| Bicycle | 70.1 | 71.0 | 70.4 |
| Average | 68.7 | 69.5 | 68.8 |

Table 4. Average class IoUs on the CityScapes validation set for the Dilation10 baseline, the GRFP model using 5 frames, GRFP(5), and the refined Dilation10 net that the GRFP learns, which is equivalent to GRFP(1).

In table 2 we show semantic segmentation results of our model on the CityScapes test set, along with the performance of a number of state of the art static semantic segmentation models.

We used our GRFP methodology trained using Dilation10, LRR-4x and PSP as baseline models and in all cases we show improved labelling accuracy. Notice that our methodology can be used with any semantic segmentation method that processes each frame independently. Since we showed improvements using all baselines, we can predict that other single-frame methods can benefit from our proposed video methodology as well.

In table 3 we show the mean IoU over classes versus the number of frames used for inference for our model based on Dilation10. One can see that under the current representation, *in inference*, not much gain is achieved by the forward model beyond propagating information from 4 frames. The results are presented in more detail in table 4 where we show the estimates produced by the pre-trained Dilation10 network and the per-frame Dilation network with parameters refined by our model, GRFP(1), as well as the results of our GRFP model operating over 5 frames GRFP(5). Notice that while the average of our GRFP(1) is almost identical

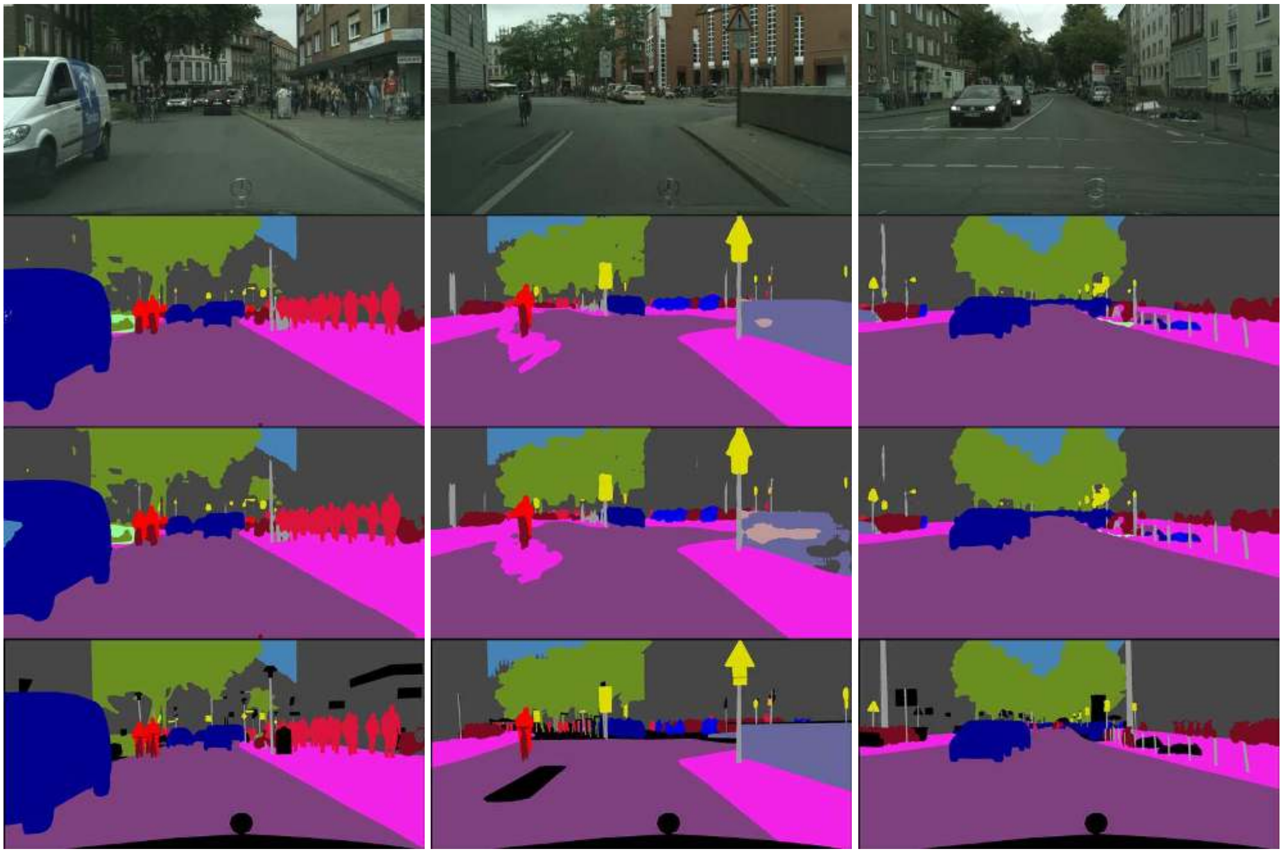

Figure 4. From top to bottom: the image, video segmentation by GRFP, static segmentation by Dilation10, and the ground truth. Notice the more accurate semantic segmentation of the car in the left example, the right wall in the middle example, and the left pole in the right example. For the first two examples the static method fails where the some object has a uniform surface over some spatial extent. The pole in the right image may be hard to estimate based on the current frame alone, but the inference problem becomes easier if earlier frames are considered, a property our GRFP model has.

| Method | Dilation10 | GRFP(5) | GRFP(1) | fwbw(a) | fwbw(b) | fwbw(c) | fwbw(d) | FlowNet2(e) |
|---|---|---|---|---|---|---|---|---|
| IoU | 68.7 | 69.5 | 68.8 | 69.5 | 69.6 | 69.6 | 69.8 | 69.5 |

Table 5. Average class IoU on the CityScapes validation set for various forward-backward models and for models refining FlowNet2 and training end-to-end. See Fig. 2 for how the parameters are defined. The first three models are described in more detail in table 4. **(a)** A forward-backward model evaluated with $T = 2$ using the same parameters for the backward STGRU as the best forward model GRFP(5) both in the forward direction and backward direction. We used $\theta_s$ and $\theta_{gf}$ as for GRFP(5) and set $\theta_{gb} = \theta_{gf}$. **(b)** as (a) but with $T = 4$. **(c)** We used $\theta_{gf}$ and $\theta_s$ from GRFP(5) but refined $\theta_{gb}$ independently. We used $T = 4$. **(d)** We refined all parameters $\theta_{gf}$, $\theta_{gb}$ and $\theta_s$ jointly and used $T = 4$. (**e**) We used the setting in GRFP(5) and trained the Dilation10 network, the flow network and the recurrent network jointly. It was evaluated in forward mode with $T = 4$.

to the one of the pre-trained Dilation10, the individual class accuracies are different. It is apparent that most of our gains come from contributions due to temporal propagation and consistency reasoning in our STGRU models.

Figure 4 shows several illustrative situations where our proposed GRFP methodology outperforms the single frame Dilation10 baseline. In particular, our method is capable to more accurately segment the car, the right wall, and the left pole. In all cases it is apparent that inference for the current frame becomes easier when information is integrated over a longer temporal window, as in GRFP. in fig. 3 we show several illustrative examples of the flow gating learned by our STGRU units. Notice that the areas our model learns to discard ($r_t = 0$) correspond to occluded regions.

**Combining forward and backward models.** In table 5 we show the accuracy on the CityScapes validation set for various settings where we used the forward and backward models and averaged the predictions using Dilation10. This joint model was described in fig. 2. The best results were

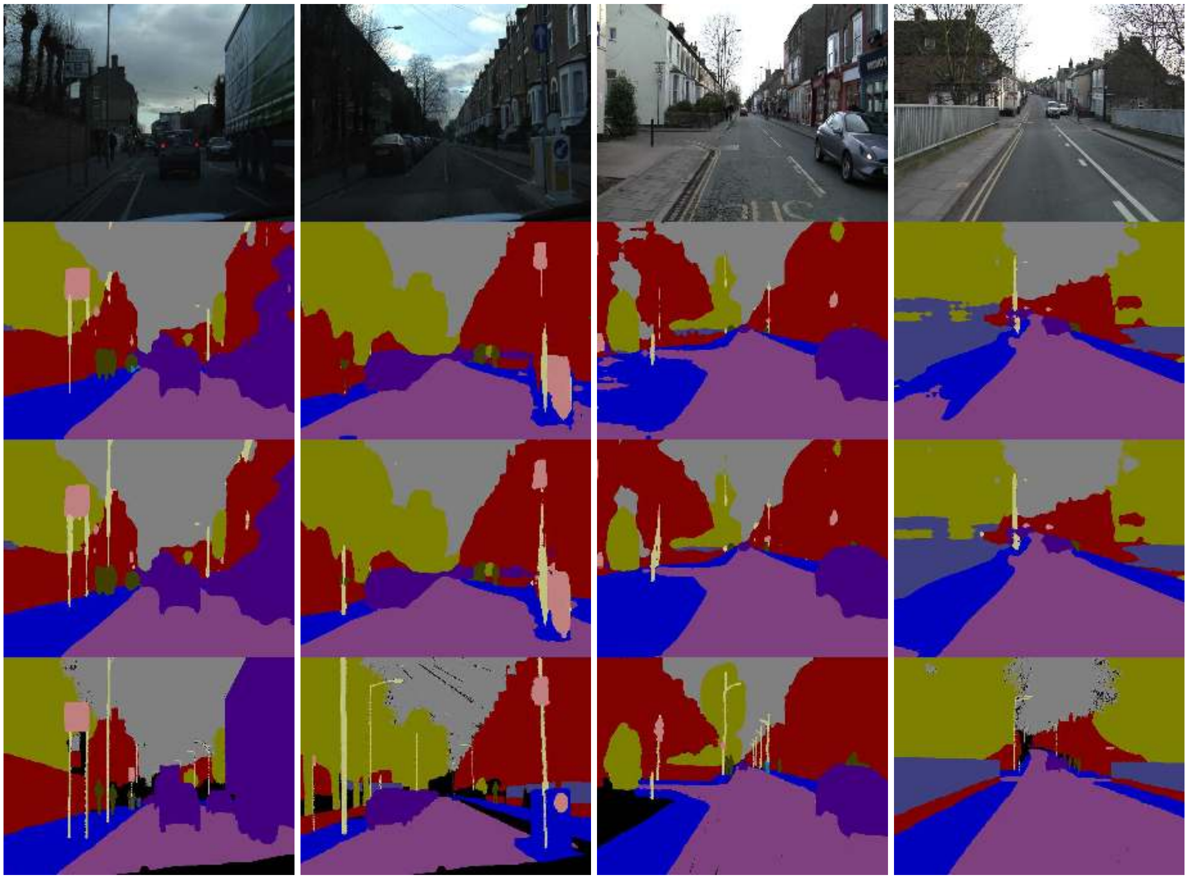

Figure 5. Qualitative examples from the CamVid test set. From top to bottom: the image, static segmentation by Dilation8, video segmentation by GRFP, and the ground truth. In the two examples to the left, notice that poles are better segmented by our video method. In the two right images, the sidewalks are better segmented using video.

| Method | IoU |
|---|---|
| Dilation8 [44] | 65.3 |
| FSO [25] | 66.1 |
| GRFP(Dilation8, FlowNet2) | 66.1 |
| VPN [19] | 66.7 |
| NetWarp [10] | 67.1 |

Table 6. Average IoU on the test set of CamVid for different video segmentation methods all based on the per-frame Dilation8 CNN. Note that out method GRFP obtains a higher score than the static Dilation8 model it is based on.

obtained when a forward-backward model was trained by averaging the predictions using 5 frames going forward (using $I_{t-4}, I_{t-3}, \ldots, I_t$) with the predictions using 5 frames going backward (using $I_{t+4}, I_{t+3}, \ldots, I_t$). Better results were obtained when the forward and backward models were trained jointly, and not independently.

**Joint training including optical flow.** To make our model entirely end-to-end trainable we also refine its optical flow component. In training, we jointly estimate all the component STGRU, Dillation10 and Flownet2 parameters. The model produced competitive results, see (e) in table 5, although the performance was not improved compared to models where we only refined the parameters of the static network and those of the STGRU. We note that the error signal passed to the FlowNet comes from a loss based on semantic segmentation. This is a very weak form of supervision for refining optical flow.

**CamVid** To show that our method is not limited to CityScapes we also provide additional experiments on the CamVid dataset [2]. This dataset consists of 4 videos that are annotated at 1 Hz. We follow the setup in [44, 25] where all images are downsampled to 480 × 640, and we use the same split with 367 training images, 100 validation images and 233 test images. We use our GRFP methodology with Dilation8 [44] as static network and FlowNet2 [17] as flow network. The results are shown in table 6. We can see that the segmentation accuracy is improved by using additional video frames as input, and our labeling results are on par with the state-of-the-art[25]. Qualitative examples are

| Method | Vid0 | Vid1 | Vid2 | Avg |
|---|---|---|---|---|
| GRFP(PSP-SSc, FlowNet2) | 88.15 | 90.99 | 85.64 | 88.26 |
| PSP-Ssc | 85.82 | 88.93 | 82.92 | 85.89 |
| GRFP(LRR-4x, FlowNet2) | 84.78 | 88.73 | 81.72 | 85.08 |
| LRR-4x | 80.74 | 86.22 | 77.88 | 81.61 |
| FSO [25] | 91.31 | 93.32 | 89.01 | 91.21 |
| GRFP(Dilation10, FlowNet2) | 84.29 | 88.87 | 81.96 | 85.04 |
| Dilation10 | 79.18 | 86.13 | 76.77 | 80.69 |

Table 7. Temporal consistency (%) for demo videos Stuttgart_00, Stuttgart_01 and Stuttgart_02 in the CityScapes dataset. Notice that our GRFP semantic video segmentation method achieves a more consistent solution than single frame baselines.

| | Dilation10 | LRR-4x |
|---|---|---|
| Segmentation module | 350 ms | 200 ms |
| FlowNet2/FlowNet1 | 300/40 ms | 300/40 ms |
| STGRU | 35 ms | 35 ms |

Table 8. Timing of the different components of out GRFP methodology for a Titan X GPU. We show improved segmentation accuracy and temporal consistency by incurring an additional runtime of 75 ms per frame if we use FlowNet1 or 335 ms per frame if we use FlowNet2.

shown in fig. 5.

**Temporal Consistency** We evaluate the temporal consistency of our semantic video segmentation method by computing trajectories in video using [40] and calculating for how many of the trajectories the labelling is the same in all frames, following the evaluation methodology in [25]. We use the demo videos provided in the CityScapes dataset, that are 600, 1100 and 1200 frames long, respectively. Due to computational considerations, we only used the middle 512 × 512 crop of the larger CityScapes images. The results are given in table 7 where improvements are achieved for all videos and for all methods compared to per-frame baselines. This can also be seen qualitatively in the videos provided in the supplementary material. There is significantly less flickering and noise when using the proposed GRFP semantic video segmentation methodology compared to models that rely on single-frame estimates. Note that while the temporal consistency is lower for our method compared to FSO, the run-time is significantly faster. Our method takes about 0.7 s per frame while FSO takes more than 10s per frame.

**Timing.** We report timings for the different components of our method when using a Titan X GPU. We report results for both FlowNet1 [9] and FlowNet2 [17]. Table 8 show timings per frame for the three main components of our framework: the static prediction, the flow, and the STGRU computations. We report the time to process (testing, not training) one frame in a video with resolution 512 × 512. With the proposed methodology we achieve both improved temporal consistency and labelling accuracy at an additional runtime cost of 335 ms per frame with FlowNet2 and 75 ms with FlowNet1.

| Method | IoU cls |
|---|---|
| GRFP(LRR-4x, FlowNet2) | 73.6 |
| GRFP(LRR-4x, FlowNet1) | 73.4 |
| GRFP(LRR-4x, Farnebäck) | 73.0 |
| LRR-4x | 72.5 |

Table 9. Assessing the robustness of our methodology w.r.t. optical flow quality. We report the average class IoU on the validation set of CityScapes. The optical flows computed using FlowNet1[9] or the method of Farnebäck[8] have lower accuracy than FlowNet2, yet our GRFP methodology can still improve the labelling accuracy over per-frame processing baselines.

**Effect of Low Optical Flow Quality.** To assess the robustness of our methodology to inaccurate optical flow modules we perform experiments where Flownet1[9] or the optical flow method of Farnebäck [8] were used at test time instead of (the most competitive) FlowNet2 for a GRFP model trained with LRR-4x and FlowNet2. The average end-point-error (EPE) on KITTI 2012 is 25.3 pixels for Farnebäck, 9.1 pixels for FlowNet1, whereas the best FlowNet2 model has an EPE of 1.8 pixels. Based on results in table 9 we conclude that our method can compensate and still improve over per-frame processing baselines, even when the optical flow has significantly lower quality then FlowNet2.

## 5. Conclusions

We have presented a deep, end-to-end trainable methodology for semantic video segmentation – including the capability to jointly refine the static recognition, optical flow and temporal propagation modules –, that is capable of taking advantage of the information present in unlabeled frames in order to improve estimates. Our model combines a convolutional architecture and a spatio-temporal transformer recurrent layer that learns to temporally propagate semantic segmentation information by means of optical flow, adaptively gated based on its locally estimated uncertainty. Our experiments on the challenging CityScapes and CamVid datasets, and for different deep semantic components, indicate that our resulting model can successfully propagate information from labeled video frames towards nearby unlabeled ones, in order to improve both the accuracy of the semantic video segmentation and the consistency of its temporal labeling, at no additional annotation cost, and with little supplementary computation.

**Acknowledgments:** This work was supported by the European Research Council Consolidator grant SEED, CNCS-UEFISCDI PN-III-P4-ID-PCE-2016-0535, the EU Horizon 2020 Grant DE-ENIGMA, and SSF.

## References

[1] D. Banica, A. Agape, A. Ion, and C. Sminchisescu. Video object segmentation by salient segment chain composition. In *ICCV, IPGM Workshop*, 2013. 2

[2] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. 7

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 2, 4

[5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 3

[6] 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. In *CVPR*, 2016. 4

[7] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. 1

[8] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. *Image analysis*, pages 363–370, 2003. 8

[9] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 8

[10] R. Gadde, V. Jampani, and P. V. Gehler. Semantic video cnns through representation warping. In *ICCV*, 2017. 2, 3, 4, 7

[11] F. Galasso, N. Nagaraja, T. Cardenas, T. Brox, and B.Schiele. A unified video segmentation benchmark: Annotation, metrics and analysis. In *ICCV*, 2013. 1

[12] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, 2016. 4, 5

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014. 2

[14] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient hierarchical graph-based video segmentation. In *CVPR*, 2010. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[16] Y. He, W.-C. Chiu, M. Keuper, and M. Fritz. Std2p: Rgbd semantic segmentation using spatio-temporal data-driven pooling. *CVPR*, 2017. 2

[17] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 3, 4, 7, 8

[18] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015. 3

[19] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. *CVPR*, 2017. 2, 7

[20] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. In *PAMI*, 2013. 1

[21] X. Jin, X. Li, H. Xiao, X. Shen, Z. Lin, J. Yang, Y. Chen, J. Dong, L. Liu, Z. Jie, et al. Video scene parsing with predictive feature learning. *ICCV*, 2017. 2, 4

[22] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[23] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 2

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[25] A. Kundu, V. Vineet, and V. Koltun. Feature space optimization for semantic video segmentation. In *CVPR*, 2016. 1, 2, 7, 8

[26] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *CVPR*, 2011. 2

[27] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *CVPR*, 2013. 1, 2

[28] G. Lin, C. Shen, I. Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 4

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[30] S. Liu, C. Wang, R. Qian, H. Yu, and R. Bao. Surveillance video parsing with single frame supervision. In *CVPR*, 2017. 2, 3

[31] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, 2015. 4

[32] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2, 4

[33] D. Nilsson and C. Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. *arXiv preprint arXiv:1612.08871*, 2016. 2, 3

[34] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *PAMI*, 2014. 2

[35] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013. 2

[36] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 1

[37] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell. Clockwork convnets for video semantic segmentation. In *Computer Vision–ECCV 2016 Workshops*, pages 852–868. Springer, 2016. 2

[38] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000. 2

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[40] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*, 2010. 8

[41] S. Tripathi, S. Belongie, Y. Hwang, and T. Nguyen. Semantic video segmentation: Exploring inference efficiency. In *SoC Design Conference (ISOCC), 2015 International*, pages 157–158. IEEE, 2015. 2

[42] D. Tsai, M. Flagg, A. Nakazawa, and J. M. Rehg. Motion coherent tracking using multi-label MRF optimization. *IJCV*, 2012. 1, 2

[43] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012. 1, 2

[44] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2, 4, 5, 7

[45] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *CVPR*, 2017. 4, 5

[46] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2