Gated Feedback Refinement Network for Dense Image Labeling

Md Amirul Islam, Mrigank Rochan, Neil D. B. Bruce, and Yang Wang
Department of Computer Science, University of Manitoba
Winnipeg, MB, Canada
{amirul, mrochan, bruce, ywang}@cs.umanitoba.ca

Abstract

Effective integration of local and global contextual information is crucial for dense labeling problems. Most existing methods based on an encoder-decoder architecture simply concatenate features from earlier layers to obtain higher-frequency details in the refinement stages. However, there are limits to the quality of refinement possible if ambiguous information is passed forward. In this paper we propose Gated Feedback Refinement Network (G-FRNet), an end-to-end deep learning framework for dense labeling tasks that addresses this limitation of existing methods. Initially, G-FRNet makes a coarse prediction and then it progressively refines the details by efficiently integrating local and global contextual information during the refinement stages. We introduce gate units that control the information passed forward in order to filter out ambiguity. Experiments on three challenging dense labeling datasets (CamVid, PASCAL VOC 2012, and Horse-Cow Parsing) show the effectiveness of our method. Our proposed approach achieves state-of-the-art results on the CamVid and Horse-Cow Parsing datasets, and produces competitive results on the PASCAL VOC 2012 dataset.

1. Introduction

In recent years, there have been rapid advances in deep learning applied to problems in computer vision. This has been met with a great deal of success, and has given rise to proliferation of significant variety in the structure of neural networks. Many current deep learning models apply a cascade comprised of repeated convolutional stages, followed by spatial pooling. Down-sampling by pooling allows for a very large pool of distinct and rich features, albeit at the expense of spatial resolution. For recognition problems, the loss of spatial precision is not especially problematic. However, dense image labeling problems (e.g. semantic segmentation) require pixel-level precisions. They typically involve a decoding process that gradually recovers a pixel level specification of categories. In some cases this *decoding* is done in one step [21]. While in other instances, both the encoding of patterns, and gradual recovery of spatial resolution are hierarchical. It is interesting to note that this mirrors the observed computational structure of human vision wherein space is abstracted away in favour of rich features, and recognition of patterns precedes their precise localization [10].

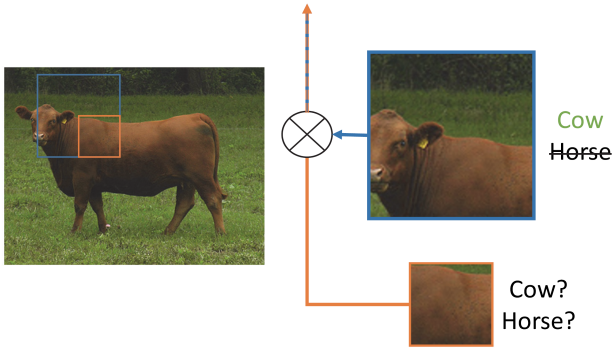


Figure 1. An illustration of the relationship between receptive field size across layers, and ambiguity that may arise. In this case, the larger (and more discriminative) receptive field (blue) resides at a deeper layer of the network, and may be of value in refining the representation carried by an earlier layer (orange) to resolve ambiguity and improve upon labeling performance.

Some models that have shown success for segmentation problems [1, 22] share a common structure involving stage-wise encoding of an input image, followed by stage-wise decoding to recover a per-pixel categorization. At an abstract level, this is reminiscent of a single network that involves a feedforward pass, followed by a recurrent pass from the top layer downward where additional computation and refinement ensues. There are tangible distinctions though, in that decoding is typically driven only by information flow that satisfies solving a specific labeling problem, and that all decoding may be informed only by the representation carried by the highest encoder layer.

At the deepest stage of encoding, one has the richest possible feature representation, and relatively poor spatial resolution from a per-neuron perspective. While spatial resolution may be poor from a per-neuron perspective, this

does not necessarily imply that recovery of precise spatial information is impossible. For example, a coarse coding strategy [13, 7] may allow for a high degree of precision in spatial localization but at the expense of the diversity of features encoded and involved in discrimination. An important implication of this, is that provided the highest layer does not require the power to precisely localize patterns, a much richer feature level representation is possible.

Information carried among earlier layers of encoding do have greater spatial locality, but may be less discriminative. Given that there is an extant representation of image characteristics at every layer, it is natural to assume that value may be had in leveraging earlier encoding representations at the decoding stage. In this manner, spatial precision that may be lost at deep layers in encoding may be gradually recovered from earlier representations. This removes some of the onus on deeper layers to represent highly discriminative characteristics of the image, while simultaneously facilitating precise localization. This intuition appears in the model we propose, as seen in connections between encoder layers and decoder layers in our network. This implies the shift in responsibility among encoding layers, and the associated discriminative power or capacity deeper in the network.

If one were to label categories within the image using only early layers, this may be problematic, especially in instances where local parts are ambiguous. The re-use of information from earlier encoder layers at the decoding stage is weakened by their lack of discrimination. For example, if one assumes reliance on convolution, and unpooling (which involve a fixed set of weights) to recover information and ultimately assign labels, this implies that any ambiguous representations are necessarily involved in decoding, which may degrade the quality of predictions. For example, while a convolutional layer deep within the network may provide strong discrimination between a cow and a horse, representations from earlier layers may be specific to animals, but express confidence for both. If this confidence is passed on to the decoding stage, and a fixed scheme for combining these representations is present, this contributes to error in labeling. This observation forms the motivation for the most novel and important aspect of our proposed model and this intuition is illustrated in Fig. 1. While information from early encoding layers may be of significant value to localization, it is sensible to filter this information such that categorical ambiguity is reduced. Moreover, it is natural to use deeper, more discriminative layers in filtering information passed on from less discriminative, but more finely localized earlier layers.

The precise scheme that achieves this is discussed in detail in the remainder of this paper. We demonstrate that a high degree of success may be achieved across a variety of benchmarks, using a relatively simple model structure in applying a canonical gating mechanism that may be applied to any network comprised of encoder and decoder components. This is also an area in which parallels may be drawn to neural information processing in humans, wherein more precisely localized representations that may be ambiguous are modulated or gated by higher-level features, iteratively and in a top-down fashion [24].

2. Background

In this section, we describe background most relevant for our proposed model.

Encoder-Decoder Architecture: Our model (Fig. 2) is based on the deep encoder-decoder architecture (e.g. [1, 22]) used for dense image labeling problems, such as semantic segmentation. The encoder network extracts features from an image and the decoder network produces semantic segmentation from the features generated by the encoder network. The encoder network is typically a CNN with alternating layers of convolution, pooling, non-linear activation, etc. The output of each convolution layer in the encoder network can be interpreted as features with different receptive fields. Due to spatial pooling, the spatial dimensions of the feature map produced by the encoder network are smaller than the original image. The decoder network will then enlarge the feature map using upsampling and unpooling in order to produce the final semantic segmentation result. Many popular CNN-based semantic segmentation models fall into this encoder-decoder framework, e.g. FCN [21], SegNet [1], DeconvNet [22].

Skip Connections: In a standard encoder-decoder architecture, the feature map from the top layer of the encoder network is used as the input for the decoder network. This feature map contains high-level features that tend to be invariant to "nuisance factors" such as small translation, illumination, etc. This invariance is crucial for certain high-level tasks such as object recognition, but is not ideal for many dense image labeling tasks (e.g. semantic segmentation) that require precise pixel-wise information, since important relationships may be abstracted away. One possible solution is to use "skip connections" [12, 21]. A skip connection directly links an encoder layer to a decoder layer. Since the bottom layers in the encoder network tend to contain precise pixel-wise information, the skip connections allow this information to be directly passed to the decoder network to produce the final segmentation result.

3. Gated Feedback Refinement Network

In this section, we describe our proposed *Gated Feedback Refinement Network* (G-FRNet) for the dense image labeling problem.

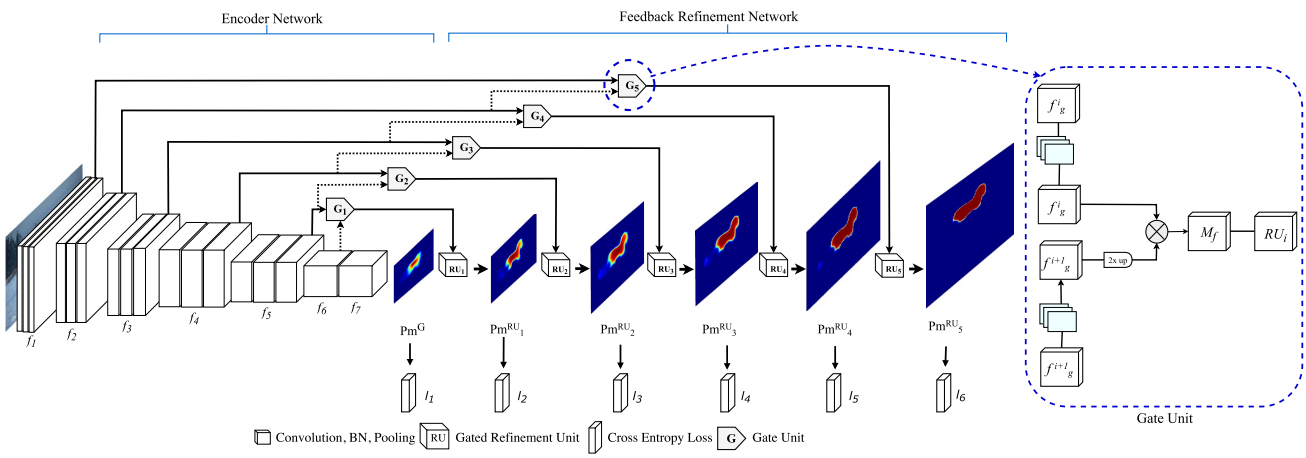


Figure 2. Overview of our Gated Feedback Refinement Network (G-FRNet). We use feature maps with different spatial dimensions produced by the encoder (f_1, f_2,..., f_7) to reconstruct a small (i.e. coarse) label map Pm^G. The decoder progressively refines the label map by adding details from feature maps in the encoder network. At each stage of the decoding, a refinement unit (RU_1, RU_2,..., RU_5) produces a new label map with larger spatial dimensions by taking information from the previous label map and encoder layers as inputs (denoted by the edge connecting G_i and RU_i). The main novelty of the model is that information from earlier encoder layers passes through a gate unit before being forwarded to the decoder. We use standard 2x bilinear upsampling on each class score map before passing it to the next stage refinement module. We also use down-sampled ground-truth label maps to provide supervision (l_1, l_2, ..., l_6) at each decoding stage.

3.1. Network Overview

Our G-FRNet is inspired by the encoder-decoder architecture [22, 1, 14] for dense image labeling. An overview of the G-FRNet architecture is shown in Fig. 2. Our encoder network is based on the VGG-16 network [25] while removing the softmax and fully connected layers in VGG-16. Following [22, 3, 21], we add two convolution layers $conv6$ and $conv7$ at the end of encoder. For an input image I, the encoder network produces 7 feature maps $(f_1, f_2, ..., f_7)$ with decreasing spatial resolution. The feature map f_7 obtained from $conv7$ has smaller spatial dimensions than the input image. We obtain the coarse prediction map Pm^G by applying a 3×3 convolution on f_7 where we set the number of output channels equal to the number of possible labels. In other words, Pm^G is an $h \times w \times C$ map where C is the number of classes. Pm^G corresponds to confidence used in predicting each spatial position as one of the C classes. Since Pm^G has smaller spatial dimensions than the input image, it only carries a coarse labeling of the image. Although we can directly upsample Pm^G (e.g. using bilinear interpolation) to match the input image size, the upsampled label map will not be very precise since the finer image details (e.g. boundaries and fine structure) are missing in Pm^G. In order to obtain a more accurate label map, we use the decoder network to progressively enlarge the label map while including finer details in label predictions. Note that we use Pm to denote prediction (or label) map throughout the paper.

We propose a Feedback Refinement Network (FRN) which forms our decoder network. Following previous work on skip connections [21, 14], FRN leverages feature maps from encoder layers to provide the finer details needed for producing an enlarged label map. For example, in order to obtain an enlarged label map Pm^{RU_1}, we can use the information from the encoder layer f_5. The conventional way of doing this is to use skip connections that directly connect two layers in a network, i.e. an encoder layer to an decoder layer. For example, in the network architecture of Fig. 2, a traditional skip connection might connect f_5 with Pm^{RU_1}. Although this allows the network to pass finer detailed information from the early encoder layers to the decoder, it may degrade the quality of predictions. As mentioned earlier, the categorical ambiguity in early encoder layers may be passed to the decoder.

The main novelty of our work is that we use a gating mechanism to modulate the information being passed via the skip connections. For example, say we want to have a skip connection to pass information from the encoder layer f_5 to the decoder layer Pm^{RU_1}. Instead of directly passing the feature map f_5, we first compute a gated feature map G_1 based on f_5 and an encoder layer above (i.e. f_6 in Fig. 2). The intuition is that f_6 contains information that can help resolve ambiguity present in f_5. For instance, some of the neurons in f_6 might fire on image patches that look like an animal (either cow or horse). This ambiguity about categories (cow vs. horse) cannot be resolve by f_5 alone since the receptive field corresponding to this encoder layer might not be large or discriminative enough. But the encoder layer (e.g. f_6) above may not be subject to these limitations and provide unambiguous confidence for the correct category. By computing the gated feature map from f_5 and f_6, categorical ambiguity can be filtered out before reaching the

decoding stage. Fig. 1 provides an example of categorical ambiguity.

The gated feature map from G_1 contains information about finer image details. We then combine it with the coarse label map Pm^G to produce an enlarged label map Pm^{RU_1}. We repeat this process to produce progressively larger label maps (Pm^{RU_1}, Pm^{RU_2}, Pm^{RU_3}, Pm^{RU_4}, Pm^{RU_5}).

We describe in detail how the gating feature is composed (Sec. 3.2) and how we compute the enlarged label map at one stage in the decoder (Sec. 3.3) in the following sections.

3.2. Gate Unit

Previous work [23] proposed refinement across different levels by combining convolution features from earlier layers. Instead of combining convolution features with coarse label maps directly, we introduce gate units to control the information passed on. The gate units are designed to control the information passed on by modulating the response of encoder layers for each spatial region in a top-down manner. Fig. 2 (right) illustrates the architecture of a gate unit.

The gate unit takes two consecutive feature map f_g^i and f_g^{i+1} as its input. The features in f_g^i are of high-resolution with smaller receptive fields (i.e. small context), whereas features in f_g^{i+1} are of low-resolution with larger receptive fields (i.e. large context). A gate unit combines f_g^i and f_g^{i+1} to generate rich contextual information. Other approaches which use refinement process straight away combines convolution features (using skip connections [21]) with coarse label maps through concatenation to generate a new label map. In this case, it is less likely that the model take full advantage of the contribution of higher resolution feature maps if they carry activation that is ambiguous with respect to class. As a result, skip connections alone have inherent limits in discerning missing spatial details. Therefore, unlike skip connections we first obtain a gated feature map before passing on the higher resolution encoding to the refinement unit.

We now explain how we obtain a gated feature map from a gate unit. The two input feature maps f_g^i and f_g^{i+1} have different spatial dimensions and channel dimensions. A sequence of operations is carried out on f_g^i and f_g^{i+1} followed by a element-wise product. Firstly, we apply a 3×3 convolution with batch normalization and ReLU to both feature maps. After these operations, let c_g^i and c_g^{i+1} be the number of channels in f_g^i and f_g^{i+1} such that $c_g^i = c_g^{i+1}$. f_g^{i+1} is then upsampled by a factor of 2 to produce a new feature map $f_{g'}^{i+1}$ whose spatial dimensions match f_g^i. We obtain the i^{th} stage gated (from gate G_i in Fig. 2) feature map M_f from the element-wise product between f_g^i and $f_{g'}^{i+1}$. Finally, the resultant feature map M_f is fed to the gated refinement unit (see Sec. 3.3). The formulation of obtaining a gated feature map M_f from gate unit G_i can be written as follows:

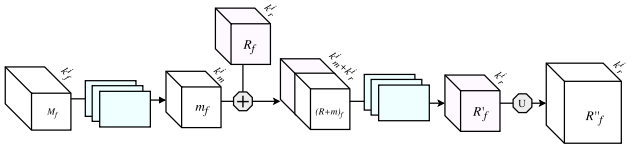


Figure 3. Detailed overview of a Gated Refinement Unit. The refinement unit is unfolded here for i^{th} stage. The refinement module (similar to [14]) is composed of convolution, batch normalization, concatenation, and upsampling operations.

$$v_i = T_f(f_g^{i+1}), u_i = T_f(f_g^i), M_f = v_i \otimes u_i \quad (1)$$

where T_f denotes the transformation function comprised of sequence of operations mentioned and $\otimes$ denotes element-wise product.

3.3. Gated Refinement Unit

Fig. 3 shows in detail the architecture of our gated refinement unit (see RU in Fig. 2). Each refinement unit RU^i takes a coarse label map R_f with channel k_r^i (generated at $(i-1)^{th}$ stage of the FRN) and gated feature map M_f as its input. RUs learn to aggregate information and generate a new label map R'_f with larger spatial dimensions through the following sequence of operations: First, we apply a 3×3 convolution followed by a batch normalization layer on M_f to obtain a feature map m_f with channel k_m^i. In our model configuration, $k_m^i = k_r^i = C$ where C is the number of possible labels. Next, m_f is concatenated with the prior stage label map R_f, producing feature map $(R+m)_f$ with $k_m^i + k_r^i$ channels. There are two reasons behind making $k_m^i = k_r^i$. First, the channel dimension of the feature map obtained from the encoder is typically very large (i.e. $c_g^i \gg k_r^i$). So directly concatenating R_f with a feature map containing a larger number of channels is computationally expensive. Second, concatenating two feature maps having a large difference in the number of channels risks dropping signals from the representation with fewer layers. Finally, the refined label map R'_f with k_r^i channels is generated by applying a 3×3 convolution on $(R+m)_f$ feature map. Note that R'_f is the i^{th} stage prediction map. The prediction map R'_f is upsampled by a factor of 2 and fed to the next stage $(i+1)^{th}$ gated refinement unit. These operations can be summarized as follows:

$$m_f = \mathbb{C}_{3\times3}(M_f), \gamma = m_f \oplus R_f, R'_f = \mathbb{C}_{3\times3}(\gamma) \quad (2)$$

where $\mathbb{C}(.)$ and $\oplus$ refer to convolution and concatenation respectively.

3.4. Stage-wise Supervision

Our network produces a sequence of label maps with increasing spatial dimensions at the decoder stage, although

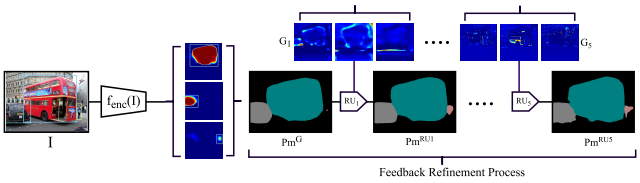


Figure 4. Visualization of hierarchical gated refinement scheme. The refinement process integrates higher-frequency details with the lower resolution label map at each stage. Class-wise activation maps for each gate are shown as heatmaps.

we are principally interested in the label map at the last stage of the decoding. Label maps produced at earlier stages of decoding might provide useful information as well and allow for supervision earlier in the network. Following [14], we adopt the idea of deep supervision [18] in our network to provide stage-wise supervision on predicted dense label maps. In more specific terms, let $I \in \mathbb{R}^{h \times w \times d}$ be a training sample with ground-truth mask $\eta \in \mathbb{R}^{h \times w}$. We obtain k resized ground-truth maps $(R_1, R_2, ..., R_k)$ by resizing η. We define a loss function l_i (pixel-wise cross entropy loss is used) to measure the difference between the resized ground-truth $R_i(\eta)$ and the predicted label map at each stage of decoding. We can write these operations as follows:

$$l_k = \begin{cases} \xi\Big(R_i(\eta), Pm^G\Big) & i = 1 \\ \xi\Big(R_i(\eta), Pm^{RU_i}\Big) & otherwise \end{cases} \quad (3)$$

where ξ denotes cross-entropy loss. The loss function in our network is the summation of cross-entropy losses (i.e. $loss(I) = \sum_{k=1}^{6} \ell_k$) at various stages of refinement network. The network is trained using back-propagation to optimize this loss.

Fig. 4 illustrates the effectiveness of the gated refinement scheme. We can see that the refinement scheme progressively improves the spatial details of dense label maps. It also shows that the top convolution layer (conv7 in our encoder network) can predict a coarse label map without capturing finer image details. The feedback refinement network is able to recover missing details (e.g. the boundaries of the bus and the car) in the coarse label map.

4. Experiments

In this section, we first discuss some implementation details (Sec. 4.1). Then we present experimental results on three challenging dense labeling benchmark datasets: Cambridge Driving Labeled Video (CamVid) (Sec. 4.2), PASCAL VOC 2012 (Sec. 4.3), and Horse-Cow Parsing (Sec. 4.4).

4.1. Implementation Details

We have implemented our network using Caffe [15] on a single Titan X GPU. Pre-trained VGG-16 [25] parameters are used to initialize the convolution layers in the encoder network (i.e. $conv1$ to $conv5$ layer). Other convolution layers' parameters are randomly assigned based on Xavier initialization. Randomly cropped patches of size $(h_{min} \times w_{min})$ are fed into the network. We set $(h_{min} \times w_{min})$ to 320×320 for Pascal VOC and 360×480 for CamVid and Horse-Cow parsing datasets. For the PASCAL VOC 2012 dataset, we normalize the data using VGG-16 mean and standard deviation. We employ pixel-wise cross entropy loss (with equal weights) as the objective function to be optimized for all the semantic categories. For the CamVid dataset, since the classes are not balanced, we use weighted cross entropy loss following previous work [1]. The weights are computed using the class balancing technique proposed in [6].

During testing, our network can take an image at its original size, as all the gated refinement modules can handle an input of any size. The network therefore produces dense predictions at the original resolution for each test image.

4.2. CamVid

The Cambridge-driving Labeled Video (CamVid) dataset [2] consists of 701 high resolution video frames extracted from a video footage recorded in a challenging urban setting. Ground-truth labels are annotated according to one of 32 semantic categories. Following [17, 1, 28], we consider 11 larger semantic classes (road, building, sky, tree, sidewalk, car, column-pole, fence, pedestrian, bicyclist, and sign-symbol) for evaluation. We split the dataset into training, validation, and test sets following [26]. Finally, we have 367 training images, 100 validation images, and 233 test images. In order to make our experimental settings comparable to previous works [17, 32, 28, 1], we downsample the images in the dataset by a factor of 2 (i.e. 480×360).

Table 1 shows the results of our model and comparisons with other state-of-the-art approaches on this dataset, demonstrating that we achieve state-of-the-art results on this dataset. For each method, we report the category-wise IoU score and mean IoU score. LRN [14] outperforms SegNet [1] by more than 11% (in terms of mean IoU) while our approach (i.e. G-FRNet) achieves an accuracy gain of 6% when compared with DeepLab [3] and by almost 2% over Dilation [32] and FSO [17].

Fig. 5 shows some qualitative results on this dataset. We can see that our model is especially accurate for challenging object categories, such as column-pole, side-walk, bicyclist, and sign-symbols compared to [17].

4.3. PASCAL VOC 2012

PASCAL VOC 2012 [8] is a challenging dataset for semantic segmentation. This dataset consists of 1,464 training images and 1,449 validation images of 20 object classes (plus the background class). There are 1,456 test images for which

Method	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Bicyclist	**mIoU**
SegNet [1]	68.7	52	87	58.5	13.4	86.2	25.3	17.9	16.0	60.5	24.8	50.2
Spatial-temporal DPN [20]	80.6	73.1	91.4	77.9	40	90.8	43.9	29.2	16	71.9	47.9	60.25
DeepLab-LargeFOV [3]	81.5	74.6	89.0	82.2	42.3	92.2	48.4	27.2	14.3	75.4	50.1	61.6
Dilation [32]	82.6	76.2	89.9	84.0	46.9	92.2	56.3	35.8	23.4	75.3	55.5	65.29
Dilation + FSO [17]	**84.0**	**77.2**	91.3	**85.7**	49.8	92.6	**59.3**	37.6	16.9	76.2	56.8	66.11
Dilation + FSO – DiscreteFlow [17]	**84.0**	77.2	91.3	85.6	**49.9**	92.5	59.1	37.6	16.9	76.0	57.2	66.12
LRN [14]	78.6	73.6	76.4	75.2	40.1	91.7	43.5	41.0	30.4	80.1	46.5	61.7
G-FRNet	82.5	76.8	**92.1**	81.8	43.0	**94.5**	54.6	**47.1**	**33.4**	**82.3**	**59.4**	**68.0**

Table 1. Quantitative results on the CamVid dataset [2]. We report per-class IoU and mean IoU for each method. Our approach achieves the state-of-the-art results on this dataset. Note that the improvements on smaller and finer objects are particularly pronounced for our model.

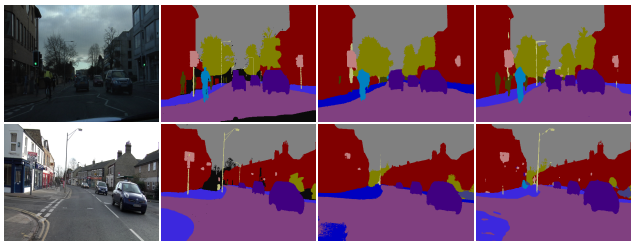


Figure 5. Qualitative results on the CamVid dataset. G-FRNet is capable of retaining the shape of smaller and finer object categories (e.g. column-pole, side-walk, bicyclist, and sign-symbols) accurately compared to FSO [17].

Method	Mean IoU (%)
DeepLab-MSc-CRF-LargeFOV [3]	68.7
FCN [21]	61.3
OA-Seg + CRF [31]	70.3
DeconvNet [22]	67.1
Attention [5]	71.4
DeepLabv2 [4]	77.7
LRN [14]	62.8
G-FRNet	68.7
G-FRNet + CRF	71.0
G-FRNet-Res101 + CRF	**77.8**

Table 2. Comparison of different methods on PASCAL VOC 2012 validation set. Note that DeconvNet [22] result is taken from [31].

ground-truth labels are not publicly available. We obtained results on the test set by submitting our final predictions to the evaluation server. Following prior work [3, 21, 1], we augment the training set with extra labeled PASCAL VOC images from [11]. In the end, we have 10,582 labeled training images.

In Table 2, we compare our results on the validation set with previous works. G-FRNet + CRF achieves best result with 71.0% mean IoU accuracy compared to encoder-decoder based architecture ([22, 31, 21]). When we switch to a base model that exhibits stronger performance (e.g. ResNet-101 [4] instead of VGG) our model G-FRNet-Res101 + CRF achieves 77.8% mean IoU which is very competitive compared to recent ResNet based state-of-the-art methods. Table 3 shows quantitative results of our method on the test set. We achieve very competitive performance compared to other baselines. LRN [14] achieves 64.2% mean IoU which outperforms FCN [21] and SegNet [1]. Our proposed approach G-FRNet improves the mean IoU accuarcy by 4%. Many existing works (e.g. [3, 22, 4, 5]) use a CRF model [16] as a postprocessing to improve the performance. When we apply CRF on top of our final prediction (G-FRNet + CRF), we further improve the mean IoU to 70.4% on the test set. G-FRNet-Res101 (with CRF) further improves the performance and yields 79.3% mean IoU on test set which is very competitive compared to existing state-of-the-art approaches. Fig. 6 shows qualitative results on the PASCAL VOC 2012 validation set. In recent years, many semantic segmentation methods have been proposed based on PASCAL VOC 2012 which are increasingly more precise in terms of IoU measure, and also introduce significant additional model complexity. However, there are only few recent methods [22, 1] that use a simpler encoder-decoder architecture for this problem, and it is most natural to compare our approach directly with this related family of models. Unlike other baseline methods, we obtain these results without employing any performance enhancing techniques, such as using object proposals [22] and multi-stage training [22]. It is worth noting that while the proposed model is shown to be highly capable across several datasets, a deeper ambition of this paper is to demonstrate the power of basic information routing mechanisms provided by gating in improving performance. The encoder-decoder based architecture provides a natural vehicle for this demonstration. It is expected that a wide variety of networks that abstract away spatial precision in favor of a more complex pool of features may benefit from installing similar logic.

method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	**mIoU**
FCN-8s [21]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
SegNet [1]	74.5	30.6	61.4	50.8	49.8	76.2	64.3	69.7	23.8	60.8	54.7	62.0	66.4	70.2	74.1	37.5	63.7	40.6	67.8	53.0	59.1
DeconvNet[22]	87.8	41.9	80.6	63.9	67.3	88.1	78.4	81.3	25.9	73.7	61.2	72.0	77.0	79.9	78.7	59.5	78.3	55.0	75.2	61.5	70.5
DeepLab [3]	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7	71.6
Dilation [32]	91.7	39.6	87.8	63.1	71.8	89.7	82.9	89.8	37.2	84.0	63.0	83.3	89.0	83.8	85.1	56.8	87.6	56.0	80.2	64.7	75.3
Attention [5]	**93.2**	41.7	88.0	61.7	74.9	92.9	84.5	90.4	33.0	82.8	63.2	84.5	85.0	87.2	85.7	60.5	87.7	57.8	84.3	68.2	76.3
LRR [9]	92.4	45.1	**94.6**	65.2	75.8	95.1	**89.1**	92.3	**39.0**	85.7	**70.4**	88.6	89.4	**88.6**	86.6	**65.8**	86.2	57.4	85.7	**77.3**	79.3
DeepLabv2 [4]	92.6	**60.4**	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	**92.1**	87.0	**87.4**	63.3	88.3	**60.0**	86.8	74.5	**79.7**
LRN [14]	79.3	37.5	79.7	47.7	58.3	76.5	76.1	78.5	21.9	67.7	47.6	71.2	69.1	82.1	77.5	46.8	70.1	40.3	71.5	57.4	64.2
G-FRNet	84.8	39.6	80.3	53.9	58.1	81.7	78.2	78.9	28.8	75.3	55.2	74.7	75.5	81.9	79.7	51.7	76.3	43.2	80.1	62.3	68.2
G-FRNet + CRF	87.7	42.9	85.4	51.6	61.0	82.9	81.7	81.6	29.1	79.3	56.1	77.6	78.6	84.6	81.6	52.8	79.0	45.0	82.1	64.1	70.4
G-FRNet-Res101	91.4	44.6	91.4	**69.2**	**78.2**	**95.4**	88.9	**93.3**	37.0	**89.7**	61.4	**90.0**	91.4	87.9	87.2	63.8	**89.4**	59.9	**87.0**	74.1	79.3

Table 3. Quantitative results in terms of mean IoU on PASCAL VOC 2012 test set. Note that G-FRNet-Res101 includes CRF.

Figure 6. Qualitative results on PASCAL VOC 2012 validation set.

4.4. Horse-Cow Parsing Dataset

To further confirm the value and generality of our model for dense labeling problems, we evaluate our model on object parts parsing dataset introduced in [29]. This dataset contains images of horse and cow images only, which are manually selected from the PASCAL VOC 2010 benchmark [8] based on most observable instances. The task is to label each pixel according to whether this pixel belongs to one of the body parts (head, leg, tail, body). We split the dataset following [29] and obtain 294 training images and 227 test images.

Table 4 shows the performance of our models and comparisons with other baseline methods. The proposed G-FRNet architecture outperforms all the baselines in terms of mean IoU. The superior performance achieved by our model indicates that integrating gate units in the refinement process is very effective in capturing complex contextual patterns within images which play a critical role in distinguishing and segmenting different localized semantic parts of an instance.

4.5. Ablation Analysis

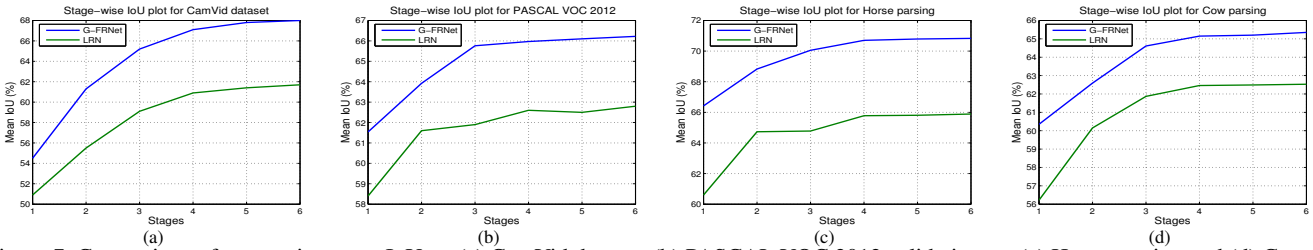
In this section, we investigate the contribution of each proposed component of the network by leaving out one or more components. We first perform a controlled study to isolate the effect of gate units. Then we include the gate units and train the network on all the datasets. Fig. 7 shows the stage-wise performance of G-FRNet and LRN [14]. From this analysis, it is clear that the inclusion of gate units not only improves the overall performance of the network, but also achieves performance gains at each stage of the feedback refinement network.

5. Discussion

From the qualitative results shown in Fig. 5 and Fig. 6 , we can see that our predictions are more precise and semantically meaningful than the baselines. For example, smaller regions (e.g. tail) in the horse-cow parsing dataset and thinner objects (e.g. column-pole, pedestrian, sign-symbol) in the CamVid dataset can be precisely labeled by G-FRNet. G-FRNet is also capable of efficiently handling categories that are similar in visual appearance (e.g. horse and cow). Regions with similar appearance (e.g. body parts of horse and cow) can be discriminated by the global contextual guidance via the gate units. The local boundaries for different semantic regions are preserved using the low-frequency information from earlier layers. Fig. 8 shows that prediction quality progressively improves with each successive stage of refinement. In coarse-level predictions, the network is only able to identify some parts of objects or semantic categories. With each stage of gated refinement, missing parts of the object are recovered and mislabeled parts are corrected. Fig. 9 shows comparison between different methods in terms of the total number of model parameters and mean IoU (%) on PASCAL VOC 2012 dataset. Although our model has only 12 to 25 percent of the number of parameters of other state-of-the-art methods (FCN [21] and DeconvNet [22]), it achieves very competitive performance. This shows the

	Horse						Cow					
Method	Bkg	head	body	leg	tail	IoU	Bkg	head	body	leg	tail	IoU
SPS- Guidance [27]	76.0	55.0	52.4	46.8	37.2	50.3	69.7	57.6	62.7	38.5	11.8	48.03
HC [12]	85.71	57.30	77.88	51.93	37.10	61.98	81.86	55.18	72.75	42.03	11.04	52.57
JPO [30]	87.34	60.02	77.52	58.35	51.88	67.02	85.68	58.04	76.04	51.12	15.00	57.18
DeepLab-LargeFoV [3]	87.44	64.45	80.70	54.61	44.03	66.25	86.56	62.76	78.42	48.83	19.97	59.31
LG - LSTM [19]	89.64	**66.89**	84.20	60.88	42.06	68.73	89.71	68.43	82.47	53.93	19.41	62.79
LRN [14]	90.11	53.23	81.57	56.50	48.03	65.89	90.30	64.41	81.52	53.44	23.03	62.53
G-FRNet	**91.79**	60.44	**84.37**	**64.07**	**53.47**	**70.83**	**91.48**	**69.26**	**84.10**	**57.58**	**24.31**	**65.35**

Table 4. Comparison of object parsing performance with state-of-the-art methods on Horse-Cow parsing dataset [29]. Note that LRN [14] does not report results on this dataset.



(a) (b) (c) (d)

Figure 7. Comparison of stage-wise mean IoU on (a) CamVid dataset; (b) PASCAL VOC 2012 validation set (c) Horse parsing and (d) Cow parsing dataset between LRN [14] and proposed network G-FRNet.

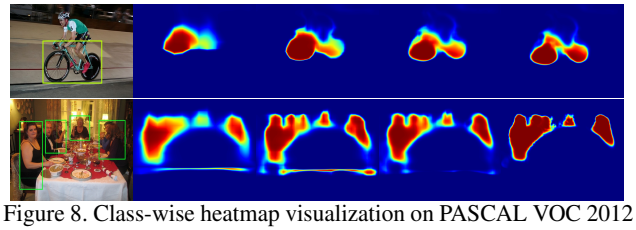


Figure 8. Class-wise heatmap visualization on PASCAL VOC 2012 validation set images after each stage of refinement. Interestingly, the network gradually aligns itself more precisely with semantic labels, while correcting initially mislabeled regions. The rightmost column shows the heatmap of the final prediction layer.

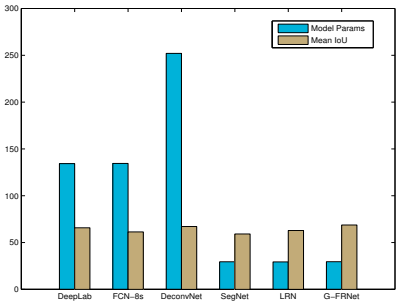


Figure 9. Analysis on the number of model parameters (in millions) and the mean IoU (%) on PASCAL VOC 2012 validation set for different methods. The rightmost method is our proposed model, which achieves best performance, even with considerably fewer parameters, and a more parsimonious model structure.

efficiency of the proposed model despite its simplicity and also the broader value of the proposed gating mechanism. Additionally, the value of the gating mechanism is demonstrated in each of the experiments, with its strengths evident in both the qualitative and quantitative results. The LRN method uses the upper layer feature map alone. We reported the result of LRN for all datasets. It is clear that the proposed gating mechanism in G-FRNet significantly improves performance compared with LRN.

6. Conclusion

We have presented a novel end-to-end deep learning framework for dense image labeling deemed a gated feedback refinement network. Our model uses an encoder-decoder architecture to progressively produce finer resolution dense labeling. The gate units in our model are able to effectively modulate signals passed forward from encoding, in order to resolve ambiguity. Our experimental results on several challenging datasets demonstrate that the proposed model performs either comparable to, or significantly better than state-of-the-art approaches. In addition, experimental results based on ablation analysis reveal generality in the value of coarse-to-fine gated refinement. A wide range of CNNs may benefit from these simple architectural modifications, given that gated refinement combines naturally with a wide array of canonical neural network architectures.

Acknowledgment

This work was supported by NSERC and the University of Manitoba Research Grants Program (URGP). We gratefully acknowledge the support of the NVIDIA Corporation GPU Grant Program.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. *TPAMI*, 2017. 1, 2, 3, 5, 6, 7

[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. 5, 6

[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. In *ICLR*, 2015. 3, 5, 6, 7, 8

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016. 6, 7

[5] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 6, 7

[6] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 5

[7] C. W. Eurich and H. Schwegler. Coarse coding: calculation of the resolution achieved by a population of large receptive field neurons. *Biological cybernetics*, 76(5):357–363, 1997. 2

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 5, 7

[9] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *ECCV*, 2016. 7

[10] K. Grill-Spector and N. Kanwisher. Visual recognition as soon as you know it is there, you know what it is. *Psychological Science*, 16(2):152–160, 2005. 1

[11] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 6

[12] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 2, 8

[13] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart. Distributed representations. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. 1986. 2

[14] M. A. Islam, S. Naha, M. Rochan, N. Bruce, and Y. Wang. Label refinement network for coarse-to-fine semantic segmentation. *arXiv:1703.00551v1*, 2017. 3, 4, 5, 6, 7, 8

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014. 5

[16] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*. 2011. 6

[17] A. Kundu, V. Vineet, and V. Koltun. Feature space optimization for semantic video segmentation. In *CVPR*, 2016. 5, 6

[18] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AISTATS*, 2015. 5

[19] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan. Semantic object parsing with local-global long short-term memory. In *CVPR*, 2016. 8

[20] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang. Deep learning markov random field for semantic segmentation. *arXiv:1606.07230*, 2016. 6

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2, 3, 4, 6, 7

[22] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 1, 2, 3, 6, 7

[23] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016. 4

[24] J. T. Serences and S. Yantis. Selective visual attention and perceptual coherence. *Trends in cognitive sciences*, 10(1):38–45, 2006. 2

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 5

[26] P. Sturgess, K. Alahari, L. Ladicky, and P. H. Torr. Combining appearance and structure from motion features for road scene understanding. In *BMVC*, 2009. 5

[27] S. Tsogkas, I. Kokkinos, G. Papandreou, and A. Vedaldi. Deep learning for semantic part segmentation with high-level guidance. *arXiv:1505.02438*, 2015. 8

[28] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville. Reseg: A recurrent neural network-based model for semantic segmentation. In *CVPR Workshops*, 2016. 5

[29] J. Wang and A. L. Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *CVPR*, 2015. 7, 8

[30] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Joint object and part segmentation using deep learned potentials. In *ICCV*, 2015. 8

[31] Y. Wang, J. Liu, Y. Li, J. Yan, and H. Lu. Objectness-aware semantic segmentation. In *ACMMM*, 2016. 6

[32] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 5, 6, 7