Gated-SCNN: Gated Shape CNNs for Semantic Segmentation

Towaki Takikawa, David Acuna, Varun Jampani, Sanja Fidler
NVIDIA, University of Waterloo, University of Toronto, Vector Institute
ICCV 2019

Project Website: https://nv-tlabs.github.io/GSCNN/
Github: https://github.com/nv-tlabs/GSCNN

Presenter: Runfa Li
Jacobs School of Engineering
Presentation Date: 05/14/2020
• Motivation & Objective
• Methodology
• Experiment details and analysis
• Advantages and disadvantages
• Takeaway
• Question
MOTIVATION

Problem:

- Information are all processed together inside a deep CNN

Potential solution:

- Incorporating additional connectivity helps the different types of information to flow across different scales of network depth 😐
- Disentangling these representations by design can lead to a more natural and effective recognition pipeline 😇

Figure 1: We introduce Gated-SCNN (GSCNN), a new two-stream CNN architecture for semantic segmentation that explicitly wires shape information as a separate processing stream. GSCNN uses a new gating mechanism to connect the intermediate layers. Fusion of information between streams is done at the very end through a fusion module. To predict high-quality boundaries, we exploit a new loss function that encourages the predicted semantic segmentation masks to align with ground-truth boundaries.
OBJECTIVE

- Obtain semantic mask, boundaries
- Small, thin objects

Figure 1: We introduce Gated-SCNN (GSCNN), a new two-stream CNN architecture for semantic segmentation that explicitly wires shape information as a separate processing stream. GSCNN uses a new gating mechanism to connect the intermediate layers. Fusion of information between streams is done at the very end through a fusion module. To predict high-quality boundaries, we exploit a new loss function that encourages the predicted semantic segmentation masks to align with ground-truth boundaries.
Figure 2: GSCNN architecture. Our architecture constitutes of two main streams. The regular stream and the shape stream. The regular stream can be any backbone architecture. The shape stream focuses on shape processing through a set of residual blocks, Gated Convolutional Layers (GCL) and supervision. A fusion module later combines information from the two streams in a multi-scale fashion using an Atrous Spatial Pyramid Pooling module (ASPP). High quality boundaries on the segmentation masks are ensured through a Dual Task Regularizer.

- Regular Stream
- Shape Stream
- Fusion Module
SHAPE STREAM

\[ t \in \{0, 1, \cdots, m\} \quad t = 0, 1, 2 \quad s_t: \text{boundaries from shape stream} \]
\[ r_t: \text{features from regular stream} \]

Attention Map:
\[ \alpha_t = \sigma(C_{1 \times 1}(s_t || r_t)) \quad \alpha_t \in \mathbb{R}^{H \times W} \]

GCL Operation:
\[ \hat{s}_{t(i,j)} = (s_t \odot w_t)(i,j) \]
\[ = (s_{t(i,j)} \odot \alpha_{t(i,j)}) + s_{t(i,j)}^T w_t \]
SHAPE FEATURE OUTPUT FROM SHAPE STREAM

Figure 6: Example output of shape stream fed into the fusion module.
FUSION MODULE

- Input:
  - Region features
  - Shape features
- Output:
  - Refined segmentation

\[ f = p(y|s, r) = \mathcal{F}_\gamma(s, r) \in \mathbb{R}^{K \times H \times W} \]

(K: number of classes)

\[ r \in \mathbb{R}^{C \times \frac{H}{m} \times \frac{W}{m}} \]

\[ s \in \mathbb{R}^{H \times W} \]

(Atrous spatial pyramid pooling)

Fusion Module

segmentation loss
dualtask loss
WHY USE ASPP?

(ATROUS SPATIAL PYRAMID POOLING)

• output need same size as input, enlarge receptive field

• Upsample with pooling layers (lose resolution, not learnable)

• Atrous/Dilated convolution reduce lose resolution, learnable
ATROUS/DILATED CONVOLUTION

Stride = 1,
padding = 0
Kernel size = 3
Receptive field = 7*7
dilated rate = 2

Receptive field = [(dilated rate - 1) * (kernel size + 1) + kernel size]
ATROUS SPATIAL PYRAMID POOLING (ASPP)

Problem: Cannot use in different scales

Advantage: Preserve multi-scale contextual information

• Edge bce loss
• Segmentation loss
• Dualtask loss
JOINT MULTI-TASK LEARNING

• Jointly supervise segmentation and boundary map prediction
DUAL TASK REGULARIZER

• Boundary-level regularization:

\[
\zeta \in \mathbb{R}^{H \times W} \\
\zeta = \frac{1}{\sqrt{2}} \left\| \nabla \left( G \ast \arg\max_{k} p(y^k r, s) \right) \right\| \\
\mathcal{L}_{\text{reg}}^{\theta, \phi, \gamma} = \lambda_3 \sum_{p^+} |\zeta(p^+) - \hat{\zeta}(p^+)|
\]

(p+ contains the set of all non-zero pixel coordinates)

• Boundary pixels are penalized when there is a mismatch with GT boundaries

• Avoid non-boundary pixels to dominate

\[
\mathcal{L}^{\theta, \phi, \gamma} = \mathcal{L}_{\text{reg}}^{\theta, \phi, \gamma} + \mathcal{L}_{\text{reg-}}^{\theta, \phi, \gamma}
\]

• Semantic-level regularization:

\[
\mathcal{L}_{\text{reg-}}^{\theta, \phi, \gamma} = \lambda_4 \sum_{k,p} \mathbb{1}_{s} \log p(y^k p | r, s)
\]

\[
\mathbb{1}_{s} = \left\{ 1 : s > \text{thr} \right\}
\]

(threshold = 0.8)
• **Baseline:** DeepLabV3+

• **Dataset:** Cityscapes, 2975 training, 500 validation, 1525 test images, 30 classes, 20000 additional coarse annotation. 19 Classes used, not use coarse data, generate the ground truth boundaries using existed method. (D. Acuna, A. Kar, and S. Fidler. Devil is in the edges: Learning semantic boundaries from noisy annotations. In CVPR, 2019)

• **Implementation details:**

  Training on an NVIDIA DGX Station using 8 GPUs
  training resolution = 800×800
  training epochs = 230
  batch_size = 16
  learning_rate = 1e-2
• Standard mIoU (for region features)
• F-score along boundaries (for boundary features)
• Distance-based mIoU (for varying distance evaluation)
STANDARD IOU

Table 1: Comparison in terms of IoU vs state-of-the-art baselines on the Cityscapes val set.

- 2% improvement
- Great improvements for small objects: motorcycles, traffic signs, traffic lights. Thin objects: poles.
\[ F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \]

\[ \beta = 1 \]

\[ \mathcal{F} = \frac{2P_c R_c}{P_c + R_c} \]

\( P_c \): contour-based precision
\( R_c \): contour-based recall


<table>
<thead>
<tr>
<th>slack in distance</th>
<th>3 pixels</th>
<th>5 pixels</th>
<th>9 pixels</th>
<th>12 pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>threshold</td>
<td>0.00088</td>
<td>0.001875</td>
<td>0.00375</td>
<td>0.005</td>
</tr>
</tbody>
</table>

(Use validation set)
### BOUNDARY F-SCORE

| Thrs | Method      | road  | s.walk | build. | wall   | fence  | pole   | t-light | t-sign | veg   | terrain | sky   | person | rider | car   | truck | bus | train | motor | bike | mean |
|------|-------------|-------|--------|--------|--------|--------|--------|---------|--------|-------|---------|-------|--------|-------|-------|-------|------|------|-------|-------|------|------|
|      |             | 92.3  | 80.4   | 87.2   | 59.6   | 53.7   | 83.8   | 75.2    | 81.2   | 90.2  | 60.8    | 90.4  | 80.4   | 91.9  | 82.8  | 81.0  | 78.5 | 78.7  | 91.6  | 81.8 | 78.0  | 81.8 |
| 12px | DeepLabV3+  | 92.2  | 81.7   | 87.9   | 59.6   | 54.3   | 87.1   | 82.3    | 84.4   | 90.9  | 61.1    | 91.9  | 80.4   | 92.6  | 92.6  | 90.0  | 90.0 | 94.6  | 79.1  | 82.2 | 80.1  |       |
|      | Ours        |       |        |        |        |        |        |         |        |       |         |       |        |       |       |       |      |       |       |      |       |       |
| 9px  | DeepLabV3+  | 91.2  | 78.3   | 84.8   | 58.1   | 52.4   | 82.1   | 73.7    | 79.5   | 87.9  | 59.4    | 89.5  | 74.7   | 76.8  | 90.0  | 80.5  | 78.1 | 86.6  | 92.5  | 81.0 | 75.4  | 78.7 |
|      | Ours        | 91.3  | 80.1   | 86.0   | 58.5   | 52.9   | 86.1   | 81.5    | 83.3   | 89.0  | 59.8    | 91.1  | 79.1   | 81.5  | 91.5  | 80.5  | 78.1 | 89.7  | 94.4  | 81.0 | 80.4  | 80.7 |
| 5px  | DeepLabV3+  | 88.1  | 72.6   | 78.1   | 55.0   | 49.1   | 77.9   | 69.0    | 74.7   | 81.0  | 55.8    | 86.4  | 69.0   | 71.9  | 85.4  | 79.4  | 77.0 | 85.4  | 92.1  | 79.4 | 68.4  | 74.7 |
|      | Ours        | 88.7  | 75.3   | 80.9   | 55.9   | 49.9   | 83.6   | 78.6    | 80.4   | 83.4  | 56.6    | 88.4  | 75.4   | 77.8  | 88.3  | 79.4  | 77.0 | 88.9  | 94.2  | 76.9 | 75.1  | 77.6 |
| 3px  | DeepLabV3+  | 83.7  | 65.1   | 69.7   | 52.2   | 46.2   | 72.0   | 62.8    | 67.7   | 71.8  | 52.0    | 80.9  | 61.5   | 66.4  | 78.8  | 78.2  | 75.8 | 83.9  | 91.7  | 77.9 | 60.9  | 69.7 |
|      | Ours        | 85.0  | 68.8   | 74.1   | 53.3   | 47.0   | 79.6   | 74.3    | 76.2   | 75.3  | 53.1    | 83.5  | 69.8   | 73.1  | 83.4  | 78.2  | 75.8 | 88.0  | 93.9  | 75.1 | 68.5  | 73.6 |

Table 2: Comparison vs baselines at different thresholds in terms of boundary F-score on the Cityscapes val set.

- 4% improvement
DISTANCE-BASED EVALUATION IN TERMS OF IOU

Why use this metric?

- The global IoU metric does not well reflect the accuracy for small(distant) objects

How to implement this?

- Crop factor : C
- Crop C pixels from the top and bottom
- Crop 2 × C pixels from the left and right

Figure 3: Illustration of the crops used for the distance-based evaluation.

Figure 4: Predictions at diff. crop factors.
The gap in performance between GSCNN and DeeplabV3+ increases from 2% at crop factor 0 (i.e. no cropping) to close to 6% at crop factor 400.

GSCNN performs better for small objects.
COMPARISON OF ERRORS IN PREDICTION

- Poles misclassified as people in Deeplab-v3+
- People are sharper in GSCNN
ADVANTAGE & DISADVANTAGE

Advantages:

• Sharper predictions around object boundaries and significantly boosts performance on thinner and smaller objects.

• Can be easily incorporated into existing networks for performance improvements

Disadvantages

• 21 better works (3 not use coarse data)

• Complex training scheme

• Need to approximate gradients

\[ \zeta = \frac{1}{\sqrt{2}} \| \nabla (G * \arg \max_k p(y^k | r, s)) \| \]

(Gumbel softmax trick)
TAKEAWAY

• A new gating mechanism is derived to connect the intermediate layers, which deactivates low-level information, and focus on high-level information

• To predict high quality boundaries, a new loss function is derived to encourage the predicted semantic segmentation masks to align with ground-truth boundaries

• GSCNN achieves impressive performance on predicting small and thin objects
What role does the GCL play for GSCNN? Why use the distance-based mIoU as the metric?
THANKS