1. Introduction

• Task & General Approaches:
  Semantic Segmentation

• Existing Works:
  Increase the model capacity to achieve promising results
  High runtime complexity

<table>
<thead>
<tr>
<th>Backbone Network</th>
<th>Speed (ms)</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG16</td>
<td>5.7</td>
<td></td>
</tr>
<tr>
<td>ResNet-101</td>
<td>7.1</td>
<td></td>
</tr>
<tr>
<td>Inception-ResNet</td>
<td>9.0</td>
<td></td>
</tr>
<tr>
<td>Layer Cascade</td>
<td>14.7</td>
<td></td>
</tr>
<tr>
<td>Layer Cascade (fast)</td>
<td>23.6</td>
<td></td>
</tr>
</tbody>
</table>

• Motivation:
  Partition all pixels into three sets by classification confidence

  ![Classification Confidence Image]

  Nearly 40% easy region

  • Our Idea:
    Treats a single deep model as a cascade of several sub-models
  Earlier sub-models are trained to handle easy and confident regions
  Feed-forward harder regions to the next sub-model for processing

2. Approach

• Turning Inception-ResNet into Deep Layer Cascade (LC)

  ![Deep Layer Cascade Diagram]

• Region Convolution

  ![Region Convolution Diagram]

  • Stage-wise Label Distribution

  ![Stage-wise Label Distribution Graph]

  Stage-1 handles the easy regions (background)
  Stage-2 focuses more on the foreground than stage-1 does
  Stage-3 further focuses on harder classes

5. Overall Performance

<table>
<thead>
<tr>
<th>Epoch</th>
<th>Loss</th>
<th>mIoU</th>
<th>Running Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.3</td>
<td>59.4</td>
<td>52.2</td>
</tr>
<tr>
<td>2</td>
<td>2.1</td>
<td>66.9</td>
<td>62.1</td>
</tr>
<tr>
<td>3</td>
<td>1.9</td>
<td>73.9</td>
<td>72.1</td>
</tr>
<tr>
<td>4</td>
<td>1.7</td>
<td>79.9</td>
<td>82.1</td>
</tr>
</tbody>
</table>

3. Experiments

• Ablation Study on Probability Thresholds $\rho$
  $\rho$ controls percentage of easy and hard regions

  ![Ablation Study Diagram]

• Comparisons with DeepLab & SegNet
  ![Comparisons Graph]

4. Stages’ Outputs

• Performance and Speed Trade-off
  ![Performance Graph]

5. Conclusion

• LC adopts a “difficulty-aware” learning paradigm
• LC accelerates both training and testing by the usage of region convolution. It is capable of running in real-time.
• LC is an end-to-end trainable framework that jointly optimizes the feature learning for different regions.

![Conference on Computer Vision and Pattern Recognition Image]