Attention Pyramid Module for Scene Recognition

Zhinan Qiao, Xiaohui Yuan, Chengyuan Zhuang, Abolfazl Meyarian
Department of Computer Science and Engineering, University of North Texas
Outline

- Background
- Attention Pyramid Module (APM)
  - Attention Pyramid
  - Scale Dependency
  - Scale Aggregation
- Conclusion
Background

Scene Recognition:

Scenery images often represent a complex view that includes multiple objects at different scales and a complicated background.
Introduction

Related work & Limitations:

Conventional multi-scale scene classification methods commonly follow a four-step pipeline:

2. Extracting features.
3. Concatenating or summing the features.
4. Making the final prediction.

Limitation:

Each level of the pyramid requires to train a separate network, these methods often face expensive computation cost, especially when the number of levels of the pyramid increases.
Details of our Attention Pyramid Module:

- **Attention Pyramid:**
  Scale-aware Attention Map Extraction.
- **Scale Dependency:**
  Learning the Weight of Scales.
- **Scale Aggregation:**
  Aggregating Re-weighted Scale-aware Attention Maps.
APM

APM: Implementation and Efficiency

APM introduces an extremely modest parameter increase.

For example, for the benchmark CNN model ResNet-50, the three-scale APM only adds 48 parameters ($16 \times 3$) to the network, which is negligible compared with the 24.26 million parameters of ResNet-50.

The schema of a residual block (left) and an APM embedded residual block (right).
**APM**

**Classification Results on Places365**

**Settings:** We implement APM using PyTorch. The optimizer is SGD, with a 0.9 momentum and a 1e−4 weight decay. The batch size is 256 and the learning rate is initialized as 0.1. We set the epoch size to 100 and divide the learning rate by 10 at the 30th, 60th, and 90th epochs. All experiments are conducted on a server with 8 Tesla V100 GPUs.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1</td>
<td>Top-5</td>
<td>Params</td>
</tr>
<tr>
<td>Vanilla</td>
<td>53.693</td>
<td>83.778</td>
<td>11.36 M</td>
</tr>
<tr>
<td>APM</td>
<td>54.978</td>
<td>84.786</td>
<td>11.37 M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ResNet-50</th>
<th>Top-1</th>
<th>Top-5</th>
<th>GFLOPs</th>
<th>Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>SK-ResNet-50 [14]</td>
<td>56.142</td>
<td>86.274</td>
<td>4.18</td>
<td>24.85 M</td>
</tr>
<tr>
<td>APM-ResNet-50</td>
<td>56.707</td>
<td>86.597</td>
<td>4.13</td>
<td>24.29 M</td>
</tr>
</tbody>
</table>

**Highlight:** Our proposed module with the vanilla ResNet50 improves the performance by 3.54% top-1 classification accuracy, whereas almost no additional computations are introduced.
The influence of Scales of APM

<table>
<thead>
<tr>
<th>Model</th>
<th>ResNet-18</th>
<th>ResNet-50</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1</td>
<td>Top-5</td>
</tr>
<tr>
<td>Vanilla</td>
<td>53.693</td>
<td>83.778</td>
</tr>
<tr>
<td>- $L_1$</td>
<td>54.523</td>
<td>84.838</td>
</tr>
<tr>
<td>- $L_2$</td>
<td>54.636</td>
<td>84.978</td>
</tr>
<tr>
<td>- $L_3$</td>
<td>54.540</td>
<td>84.767</td>
</tr>
<tr>
<td>$L_1 + L_2 + L_3$</td>
<td><strong>54.978</strong></td>
<td><strong>84.786</strong></td>
</tr>
</tbody>
</table>

**Highlight:** The best result, in terms of top-1 accuracy, for ResNet18 and ResNet-50 is achieved when all three scales, L1, L2, and L3 are involved.
**APM**

**Ablation Study**

<table>
<thead>
<tr>
<th></th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
<th>✓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pyramid Attention (Non-learnable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Batch Normalization</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sigmoid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pyramid Attention (Learnable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td><strong>Top-1</strong></td>
<td>54.767</td>
<td>54.896</td>
<td>55.318</td>
<td>55.477</td>
<td>56.162</td>
</tr>
<tr>
<td><strong>Top-5</strong></td>
<td>84.932</td>
<td>84.942</td>
<td>85.499</td>
<td>85.592</td>
<td>86.129</td>
</tr>
</tbody>
</table>

**Highlight:** 1) We can observe that using APM alongside all the components except the non-learnable module leads to achieving the best result.

2) Switching a non-learnable multiscale module to APM achieved a 0.97% top-1 accuracy improvement.
Conclusion

• We have presented a simple yet effective module, called APM for building attention pyramids inside benchmark networks and further assisting the task of scene recognition.

• The APM can be combined with any existing backbone architectures in a plug-and-play manner with marginal computation increase.

• We also experimentally demonstrated that our APM is more parameter efficient while achieving better performance against state-of-the-art attention modules.