Object-Level Context Modeling For Scene Classification with Context-CNN

Syed Ashar Javed$^1$ and Anil Kumar Nelakanti$^2$

$^1$IIIT Hyderabad, $^2$Amazon

{ashar.javed@research.iiit.ac.in, annelaka@amazon.com}

1. Introduction

The task of classifying a scene requires assimilation of complex, inter-connected information about the objects and the context surrounding their presence. Although deep CNN based models provide a decent baseline for scene classification, vanilla CNNs by design, are not suitable for capturing contextual knowledge like the complex interaction of objects in a scene. More sophisticated approaches from the recent literature either involve multiple networks with high number of parameters trained for weeks or models involving components which are learned separately which limits the effectiveness of the complete learning system due to the need to fuse these components.

In this work, we propose the Context-CNN model which encodes object-level context using object proposals and LSTM units on top of a CNN which extracts deep image features. This architecture attempts to bridge the semantic gap in scenes by modeling object-object and scene-object relationships within an easily-implementable, end-to-end trained system.

Our model builds on earlier work before deep learning took off where context was explicitly modeled in the form of semantic context (object co-occurrence), spatial context and scale context [2]. But unlike these approaches, our model can take into account the semantic context of a set of objects instead of a pair, does not involve separate terms for the classifier probability and context probability which are difficult to fuse and is end-to-end learned. We benchmark the model on the LSUN dataset [3] which contains 10 million images across 10 categories. The Context-CNN model achieves an accuracy of 89.03% on the validation set which makes it one of the top performing models on this dataset. Additionally, it only uses 2% of the dataset to converge to this score. We also compare our base network with variations of our model which aim to verify the source of performance gain in comparison to vanilla CNNs. Additionally, we also analyse the CNN and LSTM features and perform experiments to highlight the context modeling capacity and the discriminative capacity of the model.

2. Context-CNN model

Our model (see Figure 1) uses a pre-trained VGG16 network to extract CNN features. The input size of the images are fixed at $512 \times 512$ and the last convolutional layer produces feature maps of size $32 \times 32$. Bounding boxes are extracted using edge boxes [4] and the feature maps of these object boxes are passed through an RoI pooling layer [1] to generate a fixed size vector of size $7 \times 7$ per feature map. These object vectors are passed as input to two subsequent layers of LSTM units in decreasing order of their confidence score with increasing time steps. The output of all time steps are concatenated to build the final feature vector and fed into the dense layers and then through a softmax layer for prediction. A shortened functional form of the LSTM unit can be summarised as:

$$(c_t, h_t) = LSTM(x_t, h_{t-1}, c_{t-1}, W)$$

Thus, with each passing time step, the LSTM reads in an individual object feature vector and updates its memory. This memory helps the model capture scene context by relating objects occurring in that given scene and distinguishing it from other scenes. The discriminative capacity of the network improves as the LSTM receives more information with increasing time steps.

For more experiments, see the complete paper here: [https://arxiv.org/pdf/1705.04358.pdf](https://arxiv.org/pdf/1705.04358.pdf)

A part of this work was done while the authors were at Cube26, data science lab, New Delhi.
3. Experiments & results

We train and test our model on the LSUN dataset. The best performing variant of our model achieves an accuracy of 89.03% which is among the best results for this dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIAT_MMLAB</td>
<td>91.61</td>
</tr>
<tr>
<td>Google</td>
<td>91.20</td>
</tr>
<tr>
<td>SJTU-ReadSense(ensemble)</td>
<td>90.43</td>
</tr>
<tr>
<td>TEG Rangers(ensemble)</td>
<td>88.70</td>
</tr>
<tr>
<td><strong>Our model</strong></td>
<td><strong>89.03</strong></td>
</tr>
</tbody>
</table>

Table 1. Evaluation on the LSUN dataset

We further compare the base model against three other variations as shown in Figure 3. The 1\textsuperscript{st} variation highlights the importance of the information obtained from the high confidence score objects fed in earlier time steps. The 2\textsuperscript{nd} variation highlights the difference in performance between fully connected units versus LSTM units. The 3\textsuperscript{rd} is simply a VGG16 model for comparison. Note that even though both VGG16 and Context-CNN share the same convolution layers, our model outperforms a VGG16 network by 5.6% with 8 million fewer parameters.

<table>
<thead>
<tr>
<th>Model Variation</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-CNN base model</td>
<td>89.03</td>
</tr>
<tr>
<td>Context-CNN with last time step</td>
<td>87.34</td>
</tr>
<tr>
<td>Context-CNN with LSTM replaced</td>
<td>85.47</td>
</tr>
<tr>
<td>VGG16</td>
<td>83.41</td>
</tr>
</tbody>
</table>

Table 2. Model comparison

Figure 4. t-SNE visualisation: In (a), each data point is a CNN feature vector of a single bounding box obtained from the RoI pooling layer. (b), (c) and (d) show the output feature vectors from the 1\textsuperscript{st}, 5\textsuperscript{th} and 10\textsuperscript{th} time step of the LSTM respectively. (See Figure 2 for the names of all classes and their ID). The plot clearly shows how the discriminative ability of the features of the object bounding boxes change across the CNN and the various time steps of the LSTM.

4. Analysis and visualisation

We visualise features obtained from the CNN and compare it with features obtained from various time steps of the LSTM using t-SNE (see Figure 4).

We also use occlusion to evaluate the significance of objects in the scene in Figure 2. The importance of a bounding box is measured by the reduction in the softmax score of the correct class if the bounding box was obscured and the corresponding object occluded. The most significant bounding box is the one that leads to maximum reduction in the softmax score.

References