GPU Boosted Deep Learning in Real-time Face Alignment

Peiyi Li1 Xiaowei Liao2,1 Yong Xu2 Haibin Ling3,1 Chunyuan Liao1
1Meitu HiScene Lab, HiScene Information Technologies, Shanghai, China
2School of Computer Science and Engineering, South China University of Technology
3Computer and Information Sciences Department, Temple University, Philadelphia, PA USA

Introduction

• Detection and aligning faces in photos are critical in intelligent applications. However, this is far from enough for modern application scenarios.

• For an example shown above, for automatic facial image beautification, accurate localization of facial landmarks often determines the quality of the performance.

Face Alignment (FA)

• Being critical for face analysis, FA has been studied extensively in recent years. For academia, research work among this line is challenging when face images have extreme poses, lighting, expressions, and occlusions etc. Besides, FA is also a fundamental component in all face analysis algorithms mentioned above. For industry, once having these facial key point locations, many impossible applications becomes reachable. A robust FA algorithm is in great demand.

Deep Learning & CNN

• Research in Deep Learning (DL) has made great progresses in FA. Implementation of Convolutional Neural Networks (CNN) has fulfilled this demand, and thus made it available to some applications.

• Training a CNN requires massive computing time, which is hardly achievable without GPU technology. Specifically, CNN training can run in parallel on neurons, and the training procedure is identical and independent on different neurons. The procedure is thus natural and perfect for GPU technology.

GPU Server

• We develop our CNN model based on the DL framework Caffe [5]. Once finalized the CNN structure, we feed tons of human-labelled facial images to train the model on a GPU server cluster with two nodes connected by Infinite Band. Each node has four NVIDIA K-40 GPU on its own.

Neural Network Structure

image (1 × 80 × 80) ▼ dropout
conv-relu (20 × 80 × 80) ▼ dropout
conv-relu (20 × 80 × 80) ▼ dropout
ave-pooling (20 × 40 × 40) ▼ dropout
conv-relu (40 × 40 × 40) ▼ dropout
conv-relu (40 × 40 × 40) ▼ dropout
ave-pooling (40 × 20 × 20) ▼ dropout
conv-relu (60 × 20 × 20) ▼ dropout
conv-relu (60 × 20 × 20) ▼ dropout
ave-pooling (60 × 10 × 10) ▼ dropout
conv-relu (80 × 10 × 10) ▼ dropout
conv-relu (80 × 10 × 10) ▼ dropout
ave-pooling (80 × 5 × 5) ▼ dropout
conv-relu (80 × 3 × 3) ▼ dropout
fc-relu (256) ▼ fc(2)

• We use rectified linear units (ReLU) as the activation function [2] except for the last fully connected layer, which uses the identity function instead.

• Dropout [2] is an element-wise operation. During training, each unit in a dropout layer randomly set the output as zero to avoid over fitting in the fc layer. Dropout ratio is set to 0.01.

Experiment & Results

• On the IBUG public benchmark dataset (http://ibug.doc.ic.ac.uk/home), our method performs the best among state-of-the-art algorithms [3, 4].

• We report our results on the same metrics as [3, 4], mean error. The mean error is measured by the distances between estimated landmarks and the ground truths, and normalized with respect to the inter-ocular distance [3] or outer-ocular distance [4]. Please see Table 1.

Table 1. Average mean error accuracy comparison.

<table>
<thead>
<tr>
<th></th>
<th>Mid-ocular</th>
<th>Inter-ocular</th>
<th>Outer-ocular</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCDCN [3]</td>
<td>-</td>
<td>5.54%</td>
<td>-</td>
</tr>
<tr>
<td>CNN Cascades [4]</td>
<td>-</td>
<td>-</td>
<td>3.88%</td>
</tr>
<tr>
<td>Ours</td>
<td>5.60%</td>
<td>5.56%</td>
<td>3.96%</td>
</tr>
<tr>
<td>Ours (Shifted Average)</td>
<td>5.46%</td>
<td>5.43%</td>
<td>3.86%</td>
</tr>
</tbody>
</table>

Figure 1. Workflow of the proposed FA algorithm (left), and the structure of the proposed CNN model, including 11 convolutional layers (conv) [1, 2], 4 average pooling layers, and 2 fully connected (fc) layers.

References: