Efficient Object Detection on GPUs using MB-LBP features and Random Forests

Shalini Gupta, Nvidia
Problem overview

- Accurate and real-time object (face) detection on the GPU
Applications

Smart photography  Human-computer interaction
Windowed approach

Object/non-object pattern classifier

Most popular algorithms:
• Viola and Jones, 2004.
Existing solution - Features

- Multi-block Local Binary Pattern (MB-LBP) features

![Diagram](image)

**MB-LBP code**

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>0</th>
<th>01011010</th>
<th>MB-LBP code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>01011010</td>
<td>MB-LBP code</td>
</tr>
</tbody>
</table>

**Average of block**

**Threshold**

- \[(x_0, y_0)\]

- Multi-block Local Binary Pattern (MB-LBP) features

- Multi-block Local Binary Pattern (MB-LBP) features
Existing solution - Classifier

- Adaptive boosting cascaded classifier

Sub-window

Stage 1

Stage 2

Stage 3

Rejected sub-windows

Only 3x speedup on GPU

More stages ~15-20
Proposed solution

- MB-LBP features + Random Forest Classifier

Independent decision trees that vote
Analogous to a committee of decision makers that don’t talk to each other.
Why random forests?

- Well suited for GPUs
  - Massively parallel
  - Same amount of computation for each pixel
- Previous work
  - Face detection with HAAR features (Belle 2008)
  - Face Recognition (Belle 2008, Ghosal 2009)
  - Expression recognition (Fanelli et al., 2012)
- Fast training
- Possible to add recognition on top of detection
- Online learning
Random forest training

- Train multiple independent decision trees
  - Each tree trained on a random subset of data selected via bagging

Randomly picked subset of features determine each split

Each feature represents a possible split
Randomly picks features 1, 5 & 6
Feature 1 is better than 5 & 6 so is chosen for the split.
Training data

Positive cases (~47K faces)

20x24 rotated and mirrored near frontal upright faces

Negative cases (~50K non-faces)

Randomly selected from 10K images
Feature Selection

- All 5796 MB-LBP features
  - Slow training
  - Lower accuracy
- Feature selection based on repeatability
  - Rejected features selected < 6 times in ~1K trees
  - 2135 features selected
  - Improved accuracy
Bootstrapping

Positive Cases \rightarrow \text{Train} \rightarrow \text{Find false positives} \rightarrow \text{Append} \rightarrow \text{Negative Cases}

Up to five stages of bootstrapping improved accuracy.
Classifier Parameters

- Ordered decisions
- Increasing number of features randomly selected for a split
- 32 total trees
- Tree depth of 5
GPU (CUDA) Detector

CPU
- Convert to gray
- Resize
- Integral image

GPU
- MB-LBP features
- RF classifier

CPU
- Non-maxima suppression

>95% of computation
CUDA Kernel

- Shared memory
- 8 x 32 threads process 256 pixels (1 pixel/thread)
- Thread block

Bank conflicts

Decision trees in cache

- Trees stored in BFS order as fixed height full binary trees
- No execution branching while computing trees
Optimizations

- For large images, skip every other pixel - 30% faster

- Reducing bank conflicts by increased bank size and increased registers

- 16 bit integral instead of 32 bit

- Borders and small images on CPU

- Memcopy and kernel temporal overlap
Non-maxima suppression

Final confidence = \( \text{avg}(\text{confidence}) + \frac{(\text{no. of windows})}{50} \)

- improves accuracy
Accuracy

Measured on the FDDB dataset - 2845 images containing 5171 faces

Hard cases
# Performance (GK107 vs. core i7 - 3.0 GHz)

Image size 640 x 480

<table>
<thead>
<tr>
<th></th>
<th>MB-LBP + Random Forest</th>
<th>MB-LBP + Cascaded AdaBoost</th>
<th>Haar + Cascaded AdaBoost (Viola and Jones)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU (i7) single core</td>
<td>471</td>
<td>117</td>
<td>200</td>
</tr>
<tr>
<td>GPU (GK107)</td>
<td>22</td>
<td>42</td>
<td>100</td>
</tr>
<tr>
<td>Speed up</td>
<td>21.4</td>
<td>2.7</td>
<td>2</td>
</tr>
</tbody>
</table>

Image size 1280 x 960

<table>
<thead>
<tr>
<th></th>
<th>MB-LBP + Random Forest</th>
<th>MB-LBP + Cascaded AdaBoost</th>
<th>Haar + Cascaded AdaBoost (Viola and Jones)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU (i7) single core</td>
<td>1752</td>
<td>526</td>
<td>1250</td>
</tr>
<tr>
<td>GPU (GK107)</td>
<td>95</td>
<td>175</td>
<td>425</td>
</tr>
<tr>
<td>Speed up</td>
<td>18.4</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
GPU utilization (GK107)

- 95% global efficiency, 5% overhead of loads from shared
- 99.6% occupancy
- IPC ~3
- Further speedup needs algorithmic changes
Conclusion

- MB-LBP features + random forest classifiers for object detection
- Feature selection technique
- Optimized GPU (CUDA) detector implementation
- Highly portable to GPUs (20x speedup)
Questions?