A 58.6mW Real-Time Programmable Object Detector with Multi-Scale Multi-Object Support Using Deformable Parts Model on 1920x1080 Video at 30fps

Amr Suleiman, Zhengdong Zhang, and Vivienne Sze

Massachusetts Institute of Technology
Why Object Detection?

- UAV/Drones
- Surveillance/Recognition
- ADAS
- Self-driving cars
Object Detection System Requirements

High Image Resolution

- (480x270)
- HD (1920x1080)

Real-time & Low Latency

Low Power
Outline

• Detection with Deformable Parts Models (DPM)
• Chip Architecture
• Main Contributions
• Chip Specifications and Comparisons
• Summary
General Object Detection Methodology

 Localization (Where?)

 Classification (True or False?)
Localization: 3D Search

Localization (Where?)

Nearby/large objects

Far/small objects

Classification (True or False?)

Image Pyramid

Minimum detectable distance

Maximum detectable distance
Classification with DPM Templates

Localization (Where?)

Classification (True or False?)

Trained **DPM** templates on HOG feature images

Dot product

**Person** (high score)

HOG: Histogram of Oriented Gradients
How Does DPM Work?

DPM Score = RootScore + ∑_{i=1}^{8} \max_{dx,dy}(PartScore_i(dx, dy) − DeformCost_i(dx, dy))
Detection Accuracy

Average precision (AP) = Area under curve 

\[ 0 \leq \text{AP} \leq 1 \]
Deformable Parts are More Accurate

Detecting parts enhances the accuracy by 2x
Measured on INRIA person dataset*

Challenge: DPM has 35x more computation compared to without parts (rigid body) detection

*[http://pascal.inrialpes.fr/data/human/]
Outline

• Detection with Deformable Parts Models (DPM)
• Chip Architecture
• Main Contributions
• Chip Specifications and Comparisons
• Summary
12-level Feature Pyramid

HOG Features Pyramid

1920x1080 image

2.7x data expansion for HD (1920x1080) images

e.g.: Min. = 2.36 m  Max. = 26 m

Minimum detectable distance

Maximum detectable distance
2 Programmable Detectors

DPM Model

- Root Weights
- Parts Weights
- Deformation

Programmable DPM model with a maximum template size of 128x128 pixels
Outline

• Detection with Deformable Parts Models (DPM)
• Chip Architecture
• Main Contributions
• Chip Specifications and Comparisons
• Summary
Optimizations for Energy Efficiency

Goal: Reducing the parts classification overhead

Methods:
1) Reduce the number of classifications  
   (Pruning & Vector Quantization)
2) Reduce the cost of each classification  
   (Basis Projection)
Method 1
Reduce the number of classifications
Parts Classification in Region of Interests

Feature Pyramid Generation → Root Classification

High scores → 8 Parts Classification → Deformation

Low scores → Discard

ROIs
Parts Classification in Region of Interests

Classification Pruning Savings

Discard 97% of the feature pyramid

10x classification power reduction
Feature Storage for Parts Classification

- Store features for reuse by parts to avoid re-computation

**Vector Quantization**

16x reduction in memory size (520 KB vs. 32 KB)

2x reduction in overall chip area
Method 2
Reduce the cost of each classification
Multiplication by Zero Can be Skipped

Classification = Dot product

- Dot product → 3 K multiplications
- HD image → 88 M multiplications
- HD pyramid → 235 M multiplications

With more zero weights:
- Fewer multiplications
- Smaller weights memory size and BW
Project the Classification to a Sparse Space

\[
\langle H, W \rangle = \left\langle H, \sum_d \alpha_d S_d \right\rangle = \sum_d \alpha_d \langle H, S_d \rangle = \sum_d \alpha_d P_d = \langle P, \alpha \rangle
\]

\( H \): HOG features  \hspace{1cm} \( W \): Template weights  

\( S \): Basis vectors  \hspace{1cm} \( P \): Projected features  

\( \alpha \): Projected weights

Histogram of Templates Weights (W)  

7% zeros

Histogram of Projected Weights (\( \alpha \))  

56% zeros
Project the Classification to a Sparse Space

\[
\langle H, W \rangle = \left( H, \sum_d \alpha_d S_d \right) = \sum_d \alpha_d \langle H, S_d \rangle = \sum_d \alpha_d P_d = \langle P, \alpha \rangle
\]

Feature Basis Projection Savings

- 56% fewer multiplications
- 34% smaller weights memory BW
- 43% less power
Overall Optimizations Savings

Total 2-Detector System Savings

- **5x** less power
- **3.6x** smaller memory

Detection Accuracy *(mAP)*:  \( \Delta = -4.8\% \)

*\( \text{mAP: mean Average Precision, on PASCAL VOC2007 with 20 classes} \)
Outline

• Detection with Deformable Parts Models (DPM)
• Chip Architecture
• Main Contributions
• Chip Specifications and Comparisons
• Summary
Chip Die Photo and Specifications

<table>
<thead>
<tr>
<th>Technology</th>
<th>65nm CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chip size</td>
<td>4.0 x 4.0 mm²</td>
</tr>
<tr>
<td>Logic gates</td>
<td>3283 kgates</td>
</tr>
<tr>
<td>SRAM</td>
<td>280.1 KB</td>
</tr>
<tr>
<td>Supply</td>
<td>0.77 – 1.11 V</td>
</tr>
<tr>
<td>Frequency</td>
<td>62.5 – 125 MHz</td>
</tr>
<tr>
<td>Frame rate</td>
<td>30 – 60 fps</td>
</tr>
<tr>
<td>Resolution</td>
<td>1920x1080</td>
</tr>
<tr>
<td>Power</td>
<td>58.6 – 216.5 mW</td>
</tr>
<tr>
<td>Energy/pixel</td>
<td>0.94 – 1.74 nJ</td>
</tr>
</tbody>
</table>

Two detectors, 97% pruning.
### Energy Scalability

<table>
<thead>
<tr>
<th>Enabling <strong>one</strong> detector</th>
<th>Enabling <strong>two</strong> detectors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>34.7 mW</strong></td>
<td><strong>49.4 mW</strong> (97% pruning)</td>
</tr>
</tbody>
</table>

#### No parts vs. With parts

- **No parts**
  - Feature Generation: 34.7 mW
  - Root: 0 mW
  - Feature Storage: 0 mW
  - Parts: 0 mW

- **With parts**
  - Feature Generation: 40.3 mW
  - Root: 0 mW
  - Feature Storage: 0 mW
  - Parts: 8.6 mW

- **1-detector power**: 15% classification & 25% feature storage
- **Adding an extra detector increases power by only 19%**
Detection Examples with DPM Chip

- Live video feed
- 1920x1080
- 30fps
- Detecting pedestrians

- Fixed frames
- 1920x1080
- Detecting cars & pedestrians
### Comparison with ASIC Object Detectors

<table>
<thead>
<tr>
<th></th>
<th>JSPS 2014</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>65 nm</td>
<td>65 nm</td>
</tr>
<tr>
<td>Chip Size (mm$^2$)</td>
<td>4.2×2.1</td>
<td>4.0×4.0</td>
</tr>
<tr>
<td>Voltage</td>
<td>0.7V</td>
<td>0.77V</td>
</tr>
<tr>
<td>Resolution</td>
<td>1920x1080</td>
<td>1920x1080</td>
</tr>
<tr>
<td>#Object Classes</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Frame rate</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Multi-scale</td>
<td>No</td>
<td>12 levels</td>
</tr>
<tr>
<td>Deformable Parts</td>
<td>No</td>
<td>8 parts</td>
</tr>
<tr>
<td>Accuracy (AP)</td>
<td>0.166</td>
<td>0.80</td>
</tr>
<tr>
<td>Power (mW)</td>
<td>84</td>
<td>58.6</td>
</tr>
<tr>
<td>Energy (nJ/pixel)</td>
<td>1.35</td>
<td>0.94</td>
</tr>
</tbody>
</table>

*INRIA person dataset

- 4.7x more accurate
- 30% less energy
Summary

- A 58.6mW object detection accelerator that processes 1920x1080 videos at 30 fps
  - Uses deformable parts for 2x increase in accuracy
  - Two programmable object detectors supporting 12 scales

- Pruning, vector quantization and feature basis projection reduce the DPM classification cost
  - Reduce power by 5x and memory size by 3.6x

- This accelerator enables object detection to be as energy-efficient as video compression at < 1nJ/pixel

Acknowledgement
DARPA, Texas Instruments and TSMC University Shuttle