Automotive deep learning journey
*From driver assistance to driver replacement*

Mohamed Moustafa, PhD

moustafa@aucegypt.edu
moustafa@ieee.org
3 stops

- Pedestrian/vehicle detection
- driver’s distraction
- Autonomous steering
Pedestrian Detection: Haar cascade

Haar-like features with AdaBoost (Viola & Jones)

- AdaBoost can learn a strong classifier $H(x)$ based on a linear combination of $T$ weak classifiers, $h_t(x)$:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

318 msec/frame
TPR: 80.6%
29-stage cascade

Reject Non-pedestrian Sub-window

3
Pedestrian Detection: CoHoG features

The co-occurrence matrices are calculated for each region with various offsets \((x, y)\) and the paired orientations \((i, j)\) are voted into the corresponding component of the co-occurrence matrix \(C_{x,y}(i,j)\).

Watanabe et al., "Co-occurrence Histograms of Oriented Gradients for Pedestrian Detection", PSIVT 2009

1 min/frame
TPR: 92.1%
Dim. Of 34,704 for 3x6 regions
Pedestrian Detection: Haar + CoHoG

1st stage: Joint Haar features with AdaBoost, extracts pedestrian candidates (12 stages cascade).

2nd stage: CoHOG with L-SVM classifier to verify each proposed candidate from 1st classifier (3x6 small regions).

Faster but slightly less accurate than standalone CoHOG.

297 msec/frame
TPR: 91.9%
205 times faster

Leithy et al., “Multi-Cascade of Complementary Features for Fast and Accurate Pedestrian Detection,” Transactions on Computer Vision and Applications DOI: 10.2197/ipsjtcva.4.30
Pedestrian Detection: Haar + CoHoG

Sample from the INRIA dataset

(a) Extract possible pedestrian candidates by simple Haar features. (b) Verify each candidate by CoHOG descriptors.

On average, 89 candidates (only 0.07% of the total samples)
Pedestrian Detection: Multi-Cascade

- Integrating Full and Upper Body Detectors:
  - Full body detector: our proposed cascade ‘Haar + CoHOG’.
  - Upper body detector: the same cascade, with different dimensions, extracts possible upper parts of the fully and partially visible pedestrians.
  - Faster & more accurate than standalone CoHOG.

Leithy et al., “Multi-Cascade of Complementary Features for Fast and Accurate Pedestrian Detection,” Transactions on Computer Vision and Applications DOI: 10.2197/ipsjtcva.4.30

749 msec/frame
TPR: 94.7%
81x faster
Pedestrian Detection: Multi-Cascade

Daimler Chrysler Test set

INRIA Person Test set

Leithy et al., “Multi-Cascade of Complementary Features for Fast and Accurate Pedestrian Detection,” Transactions on Computer Vision and Applications DOI: 10.2197/ipsjtcva.4.30
### Pedestrian Detection: Multi-Cascade

<table>
<thead>
<tr>
<th>Method</th>
<th>Total Time/Image (116129 ROIs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoHOG (Watanabe)</td>
<td>61</td>
</tr>
<tr>
<td>Divided-CoHOG</td>
<td>23.46</td>
</tr>
<tr>
<td>CoHOG-FPGA</td>
<td>0.61</td>
</tr>
<tr>
<td>Boosted-CoHOG</td>
<td>2.03</td>
</tr>
<tr>
<td>Haar+CoHOG (Proposed)</td>
<td><strong>0.297</strong></td>
</tr>
<tr>
<td>Haar+CoHOG F/U (Proposed)</td>
<td>0.749</td>
</tr>
<tr>
<td>PLS</td>
<td>39.65</td>
</tr>
<tr>
<td>HOG &amp; L-SVM</td>
<td>69.82</td>
</tr>
<tr>
<td>Riemann. Manifold</td>
<td>34.84</td>
</tr>
<tr>
<td>PID &amp; RK-SVM</td>
<td>36.29</td>
</tr>
<tr>
<td>Multiple Instance</td>
<td>35.83</td>
</tr>
<tr>
<td>Deformable Part Model</td>
<td>5.63</td>
</tr>
</tbody>
</table>

CoHOG-related

Proposed

Other methods

Part-based

---

Leithy et al., “Multi-Cascade of Complementary Features for Fast and Accurate Pedestrian Detection,” Transactions on Computer Vision and Applications DOI: 10.2197/ipsjtcva.4.30
Pedestrian Detection: Multi-Cascade

Leithy et al., “Multi-Cascade of Complementary Features for Fast and Accurate Pedestrian Detection,” Transactions on Computer Vision and Applications DOI: 10.2197/ipsjtcva.4.30
Pedestrian Detection: Deep Learning

Pedestrian Detection: recent survey

Vehicle Detection: CNN is dominating

- DetectNet
- MSCNN [https://github.com/zhaoweicai/mscnn](https://github.com/zhaoweicai/mscnn)
- RRC [https://github.com/xiaohaoChen/rrc_detection](https://github.com/xiaohaoChen/rrc_detection)

Inference visualization
Distracted Driver Dataset

- We collected the first publicly-available dataset for distracted driver postures.
- We had 31 participants (22 males and 9 females) from 7 different countries (and different ethnicities and skin colors): Egypt, Germany, Canada, Uganda, Palestine, and Morocco.
- Videos were shot in 4 different cars: Proton Gen2, Mitsubishi Lancer, Nissan Sunny, and KIA Carens.
- We extracted 17,308 frames distributed over the following classes: Drive Safe, Talk Passenger, Text Right, Drink, Talk Left, Text Left, Talk Right, Adjust Radio, Hair & Makeup, and Reach Behind.

Distracted Driver Dataset samples

Distracted Driver Dataset samples

Distracted Driver Dataset Distribution

Distracted Driver Classification

- Trained an AlexNet to classify the 10-class distracted driver problem.
- Also trained hand and face detection networks
- Used an ensemble of classifiers to improve the result. We calculated the ensemble fusion weights using a genetic algorithm.
- Our AlexNet (or, ensemble of two AlexNet’s on original and skin-segmented images) could run in real-time.

Distracted Driver Classification Ensemble

### Distracted Driver Classification Results

<table>
<thead>
<tr>
<th>Model</th>
<th>NLL loss</th>
<th>Average Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet on full images</td>
<td>0.3909</td>
<td>93.65%</td>
</tr>
<tr>
<td>AlexNet on Face segmented areas</td>
<td>1.0516</td>
<td>84.28%</td>
</tr>
<tr>
<td>AlexNet on Hands segmented areas</td>
<td>0.6186</td>
<td>89.52%</td>
</tr>
<tr>
<td>AlexNet GA Ensemble</td>
<td>0.1575</td>
<td>95.98%</td>
</tr>
</tbody>
</table>

The hands seem to have more weight in posture recognition than the face.

Distracted Driver Classification Results

Auto steering: noise filtering

- Noisy steering recording via CAN bus (~100Hz) + a camera stream (25Hz)
- Asynchronized
- Filtering: Smoothing then sampling steering recordings
Auto steering: C-LSTM

- CNN is an expert feature extractor
- LSTM to learn temporal aspects of driving
- Sliding window technique during learning

Auto steering: Angular encoding

- Direct regression of steering angle output is not stable.
- Encoding the angle in a sin wave is more reliable.

\[ Y_i = \sin \left( \frac{2\pi(i - 1)}{N - 1} - \frac{\phi \pi}{2\phi_{\max}} \right), \quad 1 \leq i \leq N \]

Auto steering: Angular encoding

## Auto steering: Regression Results

<table>
<thead>
<tr>
<th>CNN Network</th>
<th>Performance</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RMSE [Degrees]</td>
<td>Whiteness [Degrees / Time unit]</td>
</tr>
<tr>
<td>Simple CNN [6]</td>
<td>23.30</td>
<td>65.8</td>
</tr>
<tr>
<td>Inception V3</td>
<td>18.67</td>
<td>43.9</td>
</tr>
<tr>
<td>Resnet 152</td>
<td>17.77</td>
<td>39.1</td>
</tr>
</tbody>
</table>

## Auto steering: Classification vs. Regression Results

<table>
<thead>
<tr>
<th>CNN Network</th>
<th>CNN Performance</th>
<th></th>
<th>C-LSTM Performance</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>17.77</td>
<td>39.1</td>
<td>16.01</td>
<td>9.7</td>
</tr>
<tr>
<td>Classification, using NLL</td>
<td>18.70</td>
<td>54.1</td>
<td>17.84</td>
<td>10.0</td>
</tr>
<tr>
<td>Classification by, sine wave fitting</td>
<td>17.44</td>
<td>43.9</td>
<td>14.93</td>
<td>8.2</td>
</tr>
</tbody>
</table>

Auto steering: Sample results

Comma.ai dataset

Steering in GTA

Going forward

- Add more cameras
- Integrate auto steering with path planning
- Consider newer deep network architectures
- Combine supervised with semi-supervised models
- Use more simulation/graphics to augment data or as a safe testbed where it is easier to generate ground truth.