Motivation

- Declining awareness for human drivers
- Monitor the driver and assess readiness to take control
- Analyze driver hands
Related Work

- Simply localizing hands in 2D does not inform us about crucial information like 3D distance to different control elements inside a vehicle cabin
- Heavily dependent on camera view used for training
- Retraining new views would require devoting considerable efforts toward ground truth annotation
- Lack of sufficiently labeled depth data
Objectives

- Convolutional Neural Network capable of detecting and segmenting driver hands from depth images
- Localize hands in 3D, calculate distance of hand to other elements
- Identify objects held by the driver
- Labeling based on chroma-keying
Chroma-keying for Instance Mask Acquisition

- Separating specific foregrounds from a common background based on color hues
- Generate instance masks by Chroma-keying registered RGB images
- Subject wears green gloves and red wrist bands, soft lights added
- Pair RGB camera and depth camera
Depth Images

- In-painted Depth Images only for input of HandyNet
- One-to-one correspondence between every pixel in depth and RGB images
- Ensure valid supervision between depth and RGB images when training
Advantages of Labeling Approach

- Hypothetical Privacy Issues
- Unaffected by lighting
- Calculation of hand location and distance
- 3D point location for pixel \((x,y)\) of depth image:

\[
X = \frac{(x - c_x)}{f_x} \cdot d, \quad Y = \frac{(y - c_y)}{f_y} \cdot d, \quad Z = d,
\]
Instance Mask Acquisition Algorithm:

Algorithm 1 Pseudocode for obtaining instance masks for a given sequence by chroma-keying

**Input:** \( \{rgb_i, depth_{ij} \}_{i=1}^N \)

**Output:** \( \{inst\_masks_j\}_{j=1}^m \)

> binary mask for each hand instance in every frame

for \( i = 1 \) to \( N \) do

\( y_i^{red} \leftarrow rgb_i^{red}(c_i, 1) \) \>
red channel

\( y_i^{green} \leftarrow rgb_i^{green}(c_i, 2) \) \>
green channel

\( y_i^{blue} \leftarrow rgb_i^{blue}(c_i, 3) \) \>
blue channel

\( Y_i^{lum} \leftarrow 0.3 \cdot y_i^{blue} + 0.59 \cdot y_i^{green} + 0.11 \cdot y_i^{red} \) \>
relative luminance

\( \text{masks} \leftarrow (Y_i^{lum} \geq \text{threshold}) \)

\( \text{mask}_j \leftarrow \text{OCL(masks)} \)

> connected component labeling

/* The block below is for cases where two instances might be merged in 3D, but are disjoint in 3D/*

for each \( \text{mask}_j \in \{\text{mask}_1\} \) do

\( \text{depth\_pixels} \leftarrow depth_{ij}(\text{mask}_j) \)

> depth pixels corresponding to each mask

\( \text{opt\_thresh} \leftarrow \text{otsu}() \)

> get optimal threshold using Otsu’s method

\( \text{mask}_k \leftarrow \{\text{depth\_pixels} \geq \text{opt\_thresh}\} \)

\( \text{mask}_k \leftarrow \{\text{depth\_pixels} < \text{opt\_thresh}\} \)

\( \{\text{mask}_j\} \leftarrow \{\text{mask}_j\} \cup \{\text{mask}_k\} \)

\( \{\text{mask}_j\} \leftarrow \{\text{mask}_j\} \setminus \{\text{mask}_k\} \)

end for

/* The block below is for cases where a hand might be partially occluded resulting in disjoint regions */

**Require:** For each element in \( \{\text{mask}_j\} \), the area in pixels is known.

\( \text{inst\_masks}_k \leftarrow \{\} \)

while \( \{\text{mask}_j\} \neq \emptyset \) do

\( \text{mask}_z \leftarrow \) largest mask \( \in \{\text{mask}_j\} \)

if \( \text{Area} (\text{mask}_z) \leq 20 \) then

\( \{\text{mask}_j\} \leftarrow \{\text{mask}_j\} \setminus \{\text{mask}_z\} \)

continue end if

for each \( \text{mask}_k \in \{\text{mask}_j\} \) do

\( \text{dist} \leftarrow \text{Distance\_3d}(\text{mask}_z, \text{mask}_k) \)

> distance is calculated between centroids using \( eq. 1 \)

if \( \text{dist} \leq 7 \) cm then

\( \{\text{mask}_j\} \leftarrow \{\text{mask}_j\} \setminus \{\text{mask}_k\} \)

\( \{\text{mask}_f\} \leftarrow \{\text{mask}_f\} \cup \{\text{mask}_k\} \)

\( \text{mask}_z \leftarrow \text{mask}_z \setminus \{\text{mask}_k\} \)

> combine \( \text{mask}_k \) and \( \text{mask}_z \)

end if

end for

\( \text{inst\_masks}_k \leftarrow \text{inst\_masks}_k \cup \{\text{mask}_z\} \)

end while

end for
Handheld Object Labels

- Identify objects in driver hands
- Each hand instance labeled
- Subject holds same object in same hand for captured sequence

Table 1: List of handheld object classes and types of objects included in each class.

<table>
<thead>
<tr>
<th>Class ID</th>
<th>Class Label</th>
<th>Objects Included</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no object</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>smartphone</td>
<td>cellphones, smartphones</td>
</tr>
<tr>
<td>2</td>
<td>tablet</td>
<td>iPad, Kindle, tablets</td>
</tr>
<tr>
<td>3</td>
<td>drink</td>
<td>cups, bottles, mugs, flasks</td>
</tr>
<tr>
<td>4</td>
<td>book</td>
<td>newspapers, books, sheets of paper</td>
</tr>
</tbody>
</table>

Algorithm 2 Pseudocode for labeling handheld objects for a given sequence

Require: The driver holds the same object using the same hand for the entire duration of the sequence

/* Note that only one instance per frame is assigned label; other instances are assigned 0 corresponding to no handheld object. This is valid by the requirement stated above. */

Input: \( \{ \text{inst}_m \}_{m=1}^M \), where
\[ \text{inst}_m = \{ \text{mask}_1, \ldots, \text{mask}_M \}, \]

- binary mask for each hand instance in every frame
- \( \text{label} \in \{1, 2, 3, 4\} \)
- label for the object used in the sequence
- \( m_1 \in \{1, \ldots, M\} \),
- user provided instance associated with label for the first frame

Output: \( \{\text{o}_i\}_{i=1}^N \)
- object label for each hand instance in every frame

1. \( \text{o}_1 \leftarrow \{0\}_M \) /* initialize all instances with zeros */
2. \( \text{o}_1(m_1) \leftarrow \text{label} /* assign label to instance holding object */
3. \( \text{last} \leftarrow \text{mask}_{-1}^{m_1} /* store last instance holding object */

/* Find instance in current frame closest to last known instance holding object */

4. for \( i \leftarrow 2 \) to \( N \) do
5. \( \min_{\text{dist}} \leftarrow \infty \)
6. for \( j \leftarrow 1 \) to \( M \), do
7. \( \text{cur} \leftarrow \text{Distance}(\text{mask}_i, \text{last}) /* distance is calculated between centroids using eq. 1 */
8. if \( \text{cur} \leq \min_{\text{dist}} \) then
9. \( m_i \leftarrow j \)
10. \( \min_{\text{dist}} \leftarrow \text{cur} \)
11. end if
12. end for
13. \( \text{o}_i \leftarrow \{0\}_M \) /* initialize all instances with zeros */

/* The following condition handles cases where the hand holding the object is temporarily occluded */

14. if \( \min_{\text{dist}} \leq 1\text{cm} \) then
15. \( \text{last} \leftarrow \text{mask}_{-1}^{m_1} \)
16. \( \text{o}_1(m_1) \leftarrow \text{label} /* assign label */
17. end if
18. end for
Training and Testing Methodology
HandyNet Architecture

- Based on Mask R-CNN
- Region Proposal Network: ResNet-50 with feature pyramids
- Prediction: Mask head retained, bounding box regression and classification head split
- Classification head receives slightly larger Region of Interest (RoI+) than other two heads
- RoI+ definition:

\[
\begin{align*}
x' &= x - \alpha \cdot w, \quad y' = y - \alpha \cdot h, \\
w' &= (1 + 2\alpha) \cdot w, \quad h' = (1 + 2\alpha) \cdot h,
\end{align*}
\]
Mask R-CNN
HandyNet Implementation

- RoI is positive if IoU of 0.5 or more; negative otherwise
- Input images fed without resizing
- Configuration account for average number and scale of hand instances
- Training schedule:
  1. 40 epochs for RPN
  2. 40 epochs for 4th stage RPN and task specific heads
  3. 40 epochs for entire network

<table>
<thead>
<tr>
<th>Input image size</th>
<th>424 × 512</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch size</td>
<td>6</td>
</tr>
<tr>
<td><strong>RPN$^1$ anchor scales</strong></td>
<td>{16, 32, 64, 128}</td>
</tr>
<tr>
<td><strong>RPN$^1$ anchor aspect ratios</strong></td>
<td>{0.5, 1, 2}</td>
</tr>
<tr>
<td>Number of anchors per image used for RPN$^1$ training</td>
<td>64</td>
</tr>
<tr>
<td>Number of RoIs per image retained for training the heads</td>
<td>20</td>
</tr>
<tr>
<td>Ratio of positive RoIs per image</td>
<td>0.1</td>
</tr>
<tr>
<td>RoIs retained post NMS during training</td>
<td>100</td>
</tr>
<tr>
<td>RoIs retained post NMS during inference</td>
<td>50</td>
</tr>
</tbody>
</table>

$^1$ Regional Proposal Network
Analysis and Results

- Training and Validation set: chroma-keying
- Test set: combined chroma-keying data with real world drive data
- Metrics: Average Precision (AP), AP50, AP75, APs, and APm
- Evaluated using Mask-IoU

<table>
<thead>
<tr>
<th>Split</th>
<th>Number of Unique Drivers</th>
<th>Number of Frames</th>
<th>Number of Hand Instances</th>
<th>Fraction of Naturalistic Driving Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>7</td>
<td>128,317</td>
<td>219,369</td>
<td>0.2</td>
</tr>
<tr>
<td>Validation</td>
<td>1</td>
<td>6,897</td>
<td>13,525</td>
<td>0.0</td>
</tr>
<tr>
<td>Testing</td>
<td>2</td>
<td>36,497</td>
<td>69,794</td>
<td>0.9</td>
</tr>
</tbody>
</table>
Analysis and Results

- Cross validation to determine $\alpha$
- Saturation point at $\alpha = 0.5$
- Class agnostic and class sensitive evaluation

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>AP</th>
<th>AP$_{50}$</th>
<th>AP$_{75}$</th>
<th>AP$_{S}$</th>
<th>AP$_{M}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>28.8</td>
<td>48.7</td>
<td>26.0</td>
<td>24.7</td>
<td>43.3</td>
</tr>
<tr>
<td>0.1</td>
<td>28.9</td>
<td>49.1</td>
<td>26.5</td>
<td>25.1</td>
<td>43.4</td>
</tr>
<tr>
<td>0.2</td>
<td>29.2</td>
<td>49.3</td>
<td>27.1</td>
<td>25.4</td>
<td>43.8</td>
</tr>
<tr>
<td>0.3</td>
<td>29.9</td>
<td>49.8</td>
<td>27.3</td>
<td>26.2</td>
<td>43.9</td>
</tr>
<tr>
<td>0.4</td>
<td>30.0</td>
<td>49.7</td>
<td>27.7</td>
<td>28.6</td>
<td>44.0</td>
</tr>
<tr>
<td>0.5</td>
<td><strong>30.6</strong></td>
<td><strong>51.8</strong></td>
<td><strong>28.0</strong></td>
<td>29.4</td>
<td><strong>44.9</strong></td>
</tr>
<tr>
<td>0.6</td>
<td>30.5</td>
<td>51.7</td>
<td>27.8</td>
<td><strong>29.6</strong></td>
<td>44.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type of evaluation</th>
<th>AP</th>
<th>AP$_{50}$</th>
<th>AP$_{75}$</th>
<th>AP$_{S}$</th>
<th>AP$_{M}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>class agnostic</td>
<td>42.9</td>
<td>83.3</td>
<td>40.4</td>
<td>34.7</td>
<td>50.8</td>
</tr>
<tr>
<td>class sensitive</td>
<td>30.3</td>
<td>51.2</td>
<td>27.9</td>
<td>28.5</td>
<td>44.0</td>
</tr>
</tbody>
</table>
Results on Driving Data

Distance to wheel: 0.01m
Object in hand: None

Distance to wheel: 0.17m
Object in hand: Smartphone
Advantages and Disadvantages

Advantages:

● Depth images as input
● Quick labeling
● Mask R-CNN state of the art
● Successfully identify most defined objects

Disadvantages:

● Different hand held objects, grasping patterns
  ○ Gather more diverse data
  ○ Change Camera View
  ○ Incorporate Temporal Information
● Train from scratch takes 52 hours
● Proper chroma-key lighting
Key Takeaway

Because of the chroma-keying approach for instance masking, sufficiently labeled training data can be compiled quickly and efficiently, and specific hand-related instance tasks such as detection, localization, and identification of handheld objects can be undertaken.
Question:

Compare the HandyNet architecture to its parent Mask R-CNN specifically in terms of its two stages. Why does the region of interest need to be larger for the classification head?
Thanks for Listening! Any Questions?