Multi-view Target Transformation for Pedestrian Detection

Wei-Yu Lee, Ljubomir Jovanov, and Wilfried Philips
TELIN-IPI, Ghent University-imec, Gent, Belgium

1. Implementation Details

As similar as [4], we downsample the input image \( I_s \) from 1080 \( \times \) 1920 to \( H = 720, W = 1280 \), and the extracted feature maps of the single-view images \( F_s \) are with downsampled size \( H_f = 90 \) and \( W_f = 160 \) from ResNet-18 [3]. After the ROI alignment [2], for each pedestrian, we get the pooled size \( s = 9 \) with the channel number \( C = 128 \). Then, the encoder is a single fully connected layer with output dimension \( 128 \). Hence, the \( \hat{F}_{l, l} \in \mathbb{R}^{128} \). The projected ground plane size \( H_g = 120 \) and \( W_g = 250 \) for Wildtrack [1] dataset. For MultiviewX [4], \( H_g = 120 \) and \( W_g = 250 \). For better understanding, we show the pseudo-code of our proposed method in Alg. 1 to illustrate the whole process.

References


Algorithm 1: Multi-view Target Transformation

| Input | Input images from \( N \) cameras: \( I_s \), Single-view predicted bounding box \( B_s \) |
| Output | Estimated occupancy maps \( O \) |

Extract the features maps \( F_s \) from the feature extractor ResNet-18(\( I_s \))

// Step 1: Single-view detection
for \( i \)-th camera view do

\( B_i = \) DetectionHead(\( F_i \))

// Step 2: Pedestrian feature extraction
for \( i \)-th camera view do

Extract the pedestrian features \( F_{p,i} \) by using the predicted bounding boxes \( B_i \)

\( F_{p,i} = \) ROI\_align(\( F_i, B_i \) \( \in \mathbb{R}^{s \times s \times C} \) for \( l \)-th pedestrian in \( F_{p,i} \) do

\( \hat{F}_{l, l} = \) Encoder(\( F_{l, l} \) \( \in \mathbb{R}^{1 \times 1 \times C} \))

end

// Step 3: Meta feature maps
Follow the size of \( F_p \) to create new tensors filled with zeros \( M_f \)
Insert each pedestrian features \( \hat{F}_{p,i} \) into the corresponding foot point

// Step 4: Perspective transformation
Concatenate extracted feature maps \( F_s \) and meta feature maps \( M_f \)
\( F_{sf} = \) concat(\( F_s, M_f \))
Apply Eq(1) to the concatenated feature maps to get the projected feature maps \( \tilde{F}_{sf} \)

// Step 5: Occupancy map
Overlap the projected feature maps \( \tilde{F}_{sf} \) from size \( (N, H_g, W_g, 2C) \) to \( (N \times 2C, H_g, W_g) \)
Predict the occupancy map by the ground plane heat map generator \( G_h \)
\( O = G_h(\tilde{F}_{sf}) \)