Supplementary Material: Two-level Data Augmentation for Calibrated Multi-view Detection

Martin Engilberge* Haixin Shi* Zhiye Wang Pascal Fua
EPFL, Lausanne, Switzerland
firstname.lastname@epfl.ch

1. Geometric Augmentation as Homography

In order to augment the homography projecting features on the ground plane, we need to represent the different geometric augmentations as homography. In this section we list the homography corresponding to the different type of geometric augmentation. We assume the augmentation is applied on an image of dimensionality $w \times h$

Flipping Flipping operations can be achieved with a homography by inverting the dimension of flipping, and shifting the image by that dimension size.
For horizontal flipping it reads

$$\mathsf{H}_{\text{Hflip}} = \begin{bmatrix} -1 & 0 & w \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

where w is the width of the image.
For vertical flipping it reads

$$\mathsf{H}_{\text{Vflip}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & h \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where h is the height of the image

Resized Crop A resized cropping operation is equivalent to a translation followed by a rescaling (zooming). Given a crop of dimensionality $w_c \times h_c$ with its top left corner coordinates (x_c, y_c) in the original image space and the crop final dimension after resize $w_r \times h_r$ the homography for that cropping operation reads as

$$\mathsf{H}_{\text{Crop}} = \begin{bmatrix} \frac{w_r}{w_c} & 0 & -x_c \frac{w_r}{w_c} \\ 0 & \frac{h_r}{h_c} & -y_c \frac{h_r}{h_c} \\ 0 & 0 & 1 \end{bmatrix}. \tag{3}$$

Affine Transformation Affine transformation consists of a combination of rotation, scaling, shearing, and translation. The homography for the affine transformation reads as:

$$\mathsf{H}_{\text{Affine}} = \mathsf{TRS} \tag{4}$$

T is the matrix responsible for translation

$$\mathsf{T} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

where t_x, t_y are the translation factor on the x and y dimension respectively.
R is the rotation matrix which applies a rotation of the angle θ it reads as

$$\mathsf{R} = \begin{bmatrix} cos(\theta) & sin(\theta) & 0 \\ -sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{6}$$

S is the matrix responsible for scaling and shearing,

$$\mathsf{S} = \begin{bmatrix} s_x & h & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{7}$$

where s_x, s_y are the scaling factor on the x and y dimension respectively. h is the shearing factor.

Perspective Transformation The perspective transformation is similar to the affine one with two additional degrees of freedom. It reads as follows

$$\mathsf{H}_{\text{Perspective}} = \mathsf{H}_{\text{Affine}}\mathsf{L}. \tag{8}$$

With L containing the two new degrees of freedom l_x, l_y the perspective distortion along the x and y dimension respectively.

$$\mathsf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_x & l_y & 1 \end{bmatrix} \tag{9}$$

In practice the torchvision homography doesn't sample each parameter of the perspective augmentation independently. Instead they sample four points which correspond to the image corner after applying the perspective augmentation. The perspective homography matrix is then generated through optimization, using the 4 pairs of corner matching points. In that case a single parameter is used to control perspective distortion, controlling how the corner points are sampled.

2. Rescaling in view augmentation

The augmentation is applied on the original image of dimension $w \times h$ however the ground plane projection is applied in the feature space. The ResNet feature extractor reduce the spatial dimension by a factor of eight, resulting in features with a spatial dimension of $\frac{w}{8} \times \frac{h}{8}$. When augmenting the ground plane projection matrix, we need to account for this change of scale.

In the main paper and for simplicity's sake, we wrote the view augmentation as $\mathsf{T}'_v = \mathsf{H}_v^{-1}\mathsf{T}_v$.

To account for the change of dimensionality above we define H_v^{-1} as follows $\mathsf{H}_v^{-1} = R_s H_{v_{\text{aug}}}^{-1} R_s^{-1}$.

Where $H_{v_{\text{aug}}}^{-1}$ is the inverse of one of the augmentation matrix defined in Section 1 and R_s is used to cancel the effect of the downscaling of the feature extractor

$$\mathsf{R}_s = \begin{bmatrix} 8 & 0 & 0 \\ 0 & 8 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

3. Code

The source code of the model and training is provided in the following repository: `https://github.com/cvlab-epfl/MVAug`