ALF-PoseNet: An Attention-Based Lightweight Fruit Pose Estimation Network

Supplementary Material

1. ALF-PoseNet (model only) code

In this section, we provide a compact PyTorch listing of ALF-PoseNet. Listing 1 shows how we keep an HRNet backbone for high-resolution feature extraction and replace the HRNet head with our lightweight CBAM–Deconv upsampling head. The head applies channel-and-spatial attention (CBAM), projects features with a 1×1 convolution, upsamples with three transposed-convolution stages, and outputs K heatmaps at a fixed resolution. This head is a drop-in replacement and can be paired with HRNet-W18/W32/W40/W48.

2. Inference Details

At inference, the pipeline begins by applying a detector (Faster R-CNN in our case) to the input image I. The detector identifies each fruit instance x_j and returns an axis-aligned bounding box parameterized as (x, y, w, h) in the image coordinates. Only detections that exceeded a 75% confidence threshold are retained. These bounding boxes are then used to crop and normalize each instance before passing it to the pose estimator. The overall inference process is illustrated in Figure 1.

Keypoint-coordinate transform. For each detected strawberry, we extract an aspect-ratio preserving crop from the original RGB image and resize it to a fixed resolution $S \times S$ (here $S = 256$), followed by zero-padding if necessary. For each crop we stores a metadata tuple $\mathcal{M} = (x_1, y_1, x_2, y_2, s, p_x, p_y)$, where (x_1, y_1, x_2, y_2) denotes the crop window in the original image,I, s is the isotropic scaling factor applied to this window, and (p_x, p_y) are the horizontal and vertical paddings added after resizing.

Given a keypoint in image coordinates $\mathbf{k}^{\text{img}} = (x^{\text{img}}, y^{\text{img}})$, its location in the normalized crop coordinate system $\mathbf{k}^{\text{crop}} = (x^{\text{crop}}, y^{\text{crop}})$ is computed as-

$$x^{\text{crop}} = (x^{\text{img}} - x_1)\, s + p_x \tag{1}$$

$$y^{\text{crop}} = (y^{\text{img}} - y_1)\, s + p_y \tag{2}$$

Conversely, predicted keypoints in crop coordinates are mapped back to the original image frame via the inverse transformation:

$$x^{\text{img}} = \frac{x^{\text{crop}} - p_x}{s} + x_1 \tag{3}$$

$$y^{\text{img}} = \frac{y^{\text{crop}} - p_y}{s} + y_1 \tag{4}$$

The pose network output a set of K heatmaps of size $H_m \times W_m$ per crop. For each keypoint k, we first apply a small Gaussian filter and then select the location of the maximum response (u_k, v_k) on the corresponding heatmap. This index is converted to a continuous crop-space coordinate by-

$$x_k^{\text{crop}} = \left(u_k + \tfrac{1}{2}\right) \frac{S}{W_m} \tag{5}$$

$$y_k^{\text{crop}} = \left(v_k + \tfrac{1}{2}\right) \frac{S}{H_m} \tag{6}$$

which is subsequently mapped to image coordinates using the inverse transform above (equations 3 and 4). This procedure ensures that all evaluation metrics are computed consistently in image coordinates, while the network operates on normalized crops.

3. Additional qualitative results

Figure 2 and 3 provide additional qualitative examples of single-fruit crops (Fig. 2), ALF-PoseNet produces compact, well-centered heatmap peaks around fine-grained landmarks, whereas the HRNet baseline often yields diffuse or slightly shifted responses under challenging cues such as reflections, textured surfaces, or low-contrast pedicels.

In multi-fruit scenes (Fig. 3), where fruits may be tightly clustered or partially occluded, ALF-PoseNet maintains precise localization and clear instance separation, avoiding the cross-assignment errors frequently observed with HRNet.

These examples visually confirm the quantitative gains, showing sharper peaks, fewer off-target activations, and more reliable placement of all six fruit keypoints, even in cluttered or occluded conditions.

```python
class ALFHead(nn.Module):
    def __init__(self, in_ch, K, hm=64, W=256, p=0.10):
        super().__init__(); self.hm = hm; self.cbam = CBAM(in_ch)
        self.proj = nn.Sequential(nn.Conv2d(in_ch, W, 1, bias=False), nn.BatchNorm2d(W),
                                  nn.ReLU(True), nn.Dropout2d(p))
        self.up = nn.Sequential(*sum(([nn.ConvTranspose2d(W, W, 4, 2, 1, bias=False),
                                       nn.BatchNorm2d(W), nn.ReLU(True), nn.Dropout2d(p)]
                                      for _ in range(3)), []))
        self.pred = nn.Conv2d(W, K, 1)
    def forward(self, F_in):
        y = self.pred(self.up(self.proj(self.cbam(F_in))))
        return y if y.shape[-1] == self.hm else F.interpolate(y, (self.hm, self.hm), mode="bilinear",
            align_corners=False)

class ALFPoseNet(nn.Module):
    def __init__(self, backbone_name="hrnet", K=6, hm=64, W=256, p=0.10, pretrained=True):
        super().__init__(); import timm
        self.backbone = timm.create_model(backbone_name, pretrained=pretrained, features_only=True,
            out_indices=[-1])
        in_ch = self.backbone.feature_info.channels()[-1]
        self.head = ALFHead(in_ch, K, hm=hm, W=W, p=p)
    def forward(self, x):
        F_last = self.backbone(x)[0] # (B,C,H',W')
        return self.head(F_last) # (B,K,hm,hm)
```

Listing 1. ALF-PoseNet: HRNet backbone with CBAM–Deconv upsampling head (compact and model only).

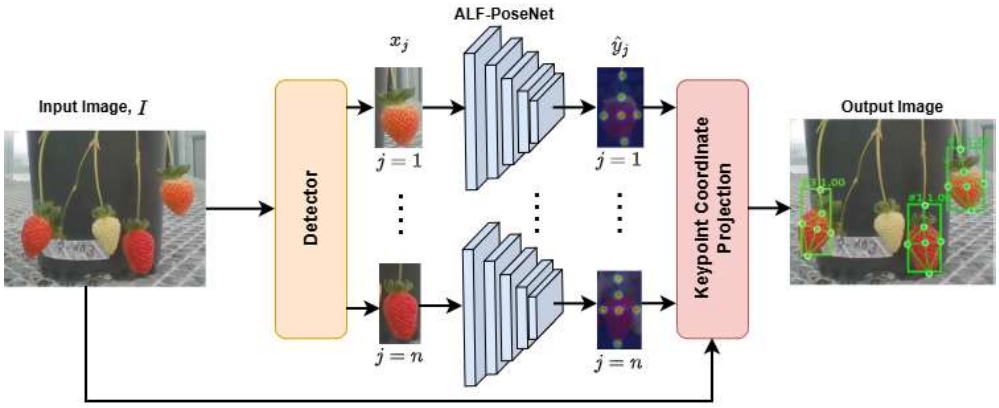

Figure 1. The inference pipeline of ALF-PoseNet for multi-instance pose estimation operates on each detected ripe strawberry in the *test* dataset. For every detection, a normalized crop is extracted, and its geometric parameters are stored. The pose network then predicts low-resolution heat maps in the crop coordinate system. The peak response in each heatmap is mapped to a continuous keypoint location within the crop, which is subsequently back-projected to the original image frame using the stored crop metadata. This ensured that all pose estimation metrics are computed consistently in the original image coordinates.

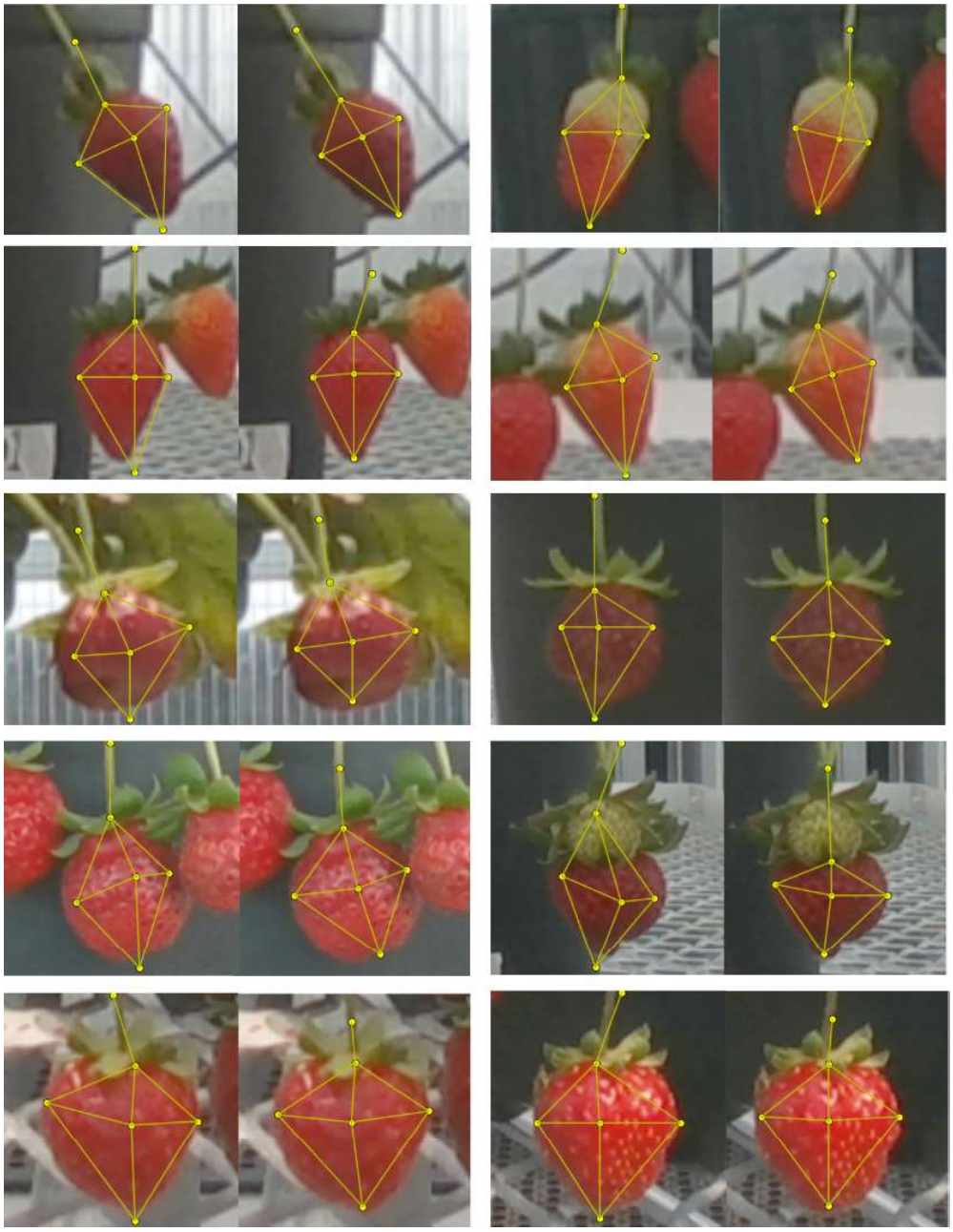

Figure 2. Additional qualitative results on the validation set of our strawberry dataset. Each example shows HRNet predictions (left) and ALF-PoseNet predictions (right). ALF-PoseNet produces accurate and well-centered heatmaps, even for strawberries with irregular or uneven shapes.

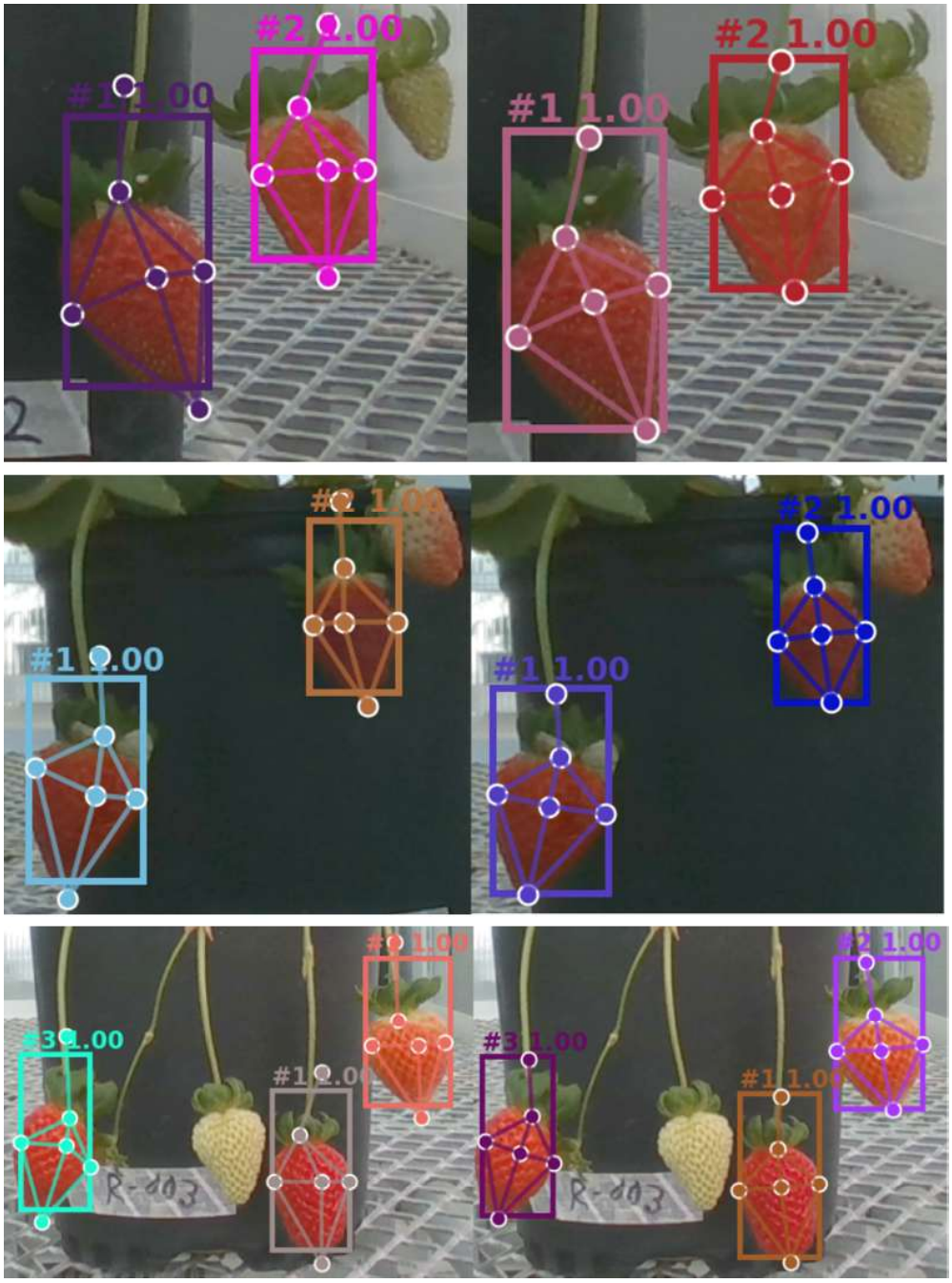

Figure 3. Additional qualitative results on the test set of the strawberry dataset at inference. Each example shows HRNet predictions (left) and ALF-PoseNet predictions (right), illustrating the improvement in keypoint localization achieved by ALF-PoseNet.