Notes for DeepFashion2

Article

Match R-CNN

Match R-CNN is built based on Mask R-CNN. We use https://github.com/facebookresearch/Detectron as framework. As can be seen from the Diagram of MatchR-CNN in our paper, you only need add the MN branch since FN and PN are provided in Mask R-CNN.

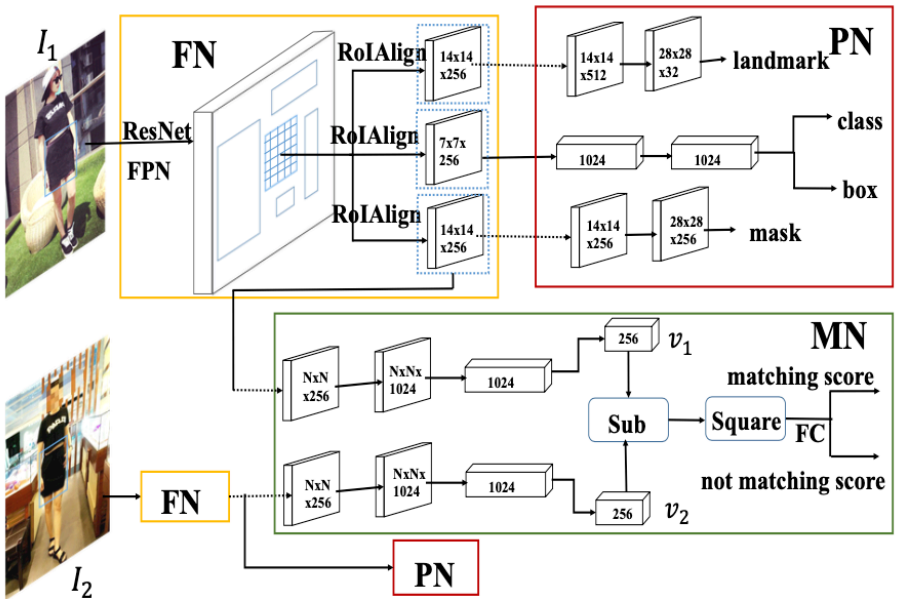

Figure 4. **Diagram of Match R-CNN** that contains three main components including a feature extraction network (FN), a perception network (PN), and a match network (MN).

Structure

Each image is passed through three main components including a Feature Network (FN), a Perception Network (PN), and a Matching Network (MN).

In the first stage, FN contains a ResNet-FPN backbone, a region proposal network (RPN) and RoIAlign module. An image is first fed into ResNet50 to extract features, which are then fed into a FPN that uses a top-down architecture with lateral connections to build a pyramid of feature maps. RoIAlign extracts features from different levels of the pyramid map.

In the second stage, PN contains three streams of networks including landmark estimation, clothes detection, and mask prediction as shown in Fig.4. The extracted RoI features after the first stage are fed into three streams in PN separately. The clothes detection stream has two hidden fully-connected (fc) layers, one fc layer for classification, and one fc layer for bounding box regression. The stream of landmark estimation has 8 'conv' layers and 2 'deconv' layers to predict landmarks. Segmentation stream has 4 'conv' layers, 1 'deconv' layer, and another 'conv' layer to predict masks.

In the third stage, MN contains a feature extractor and a similarity learning network for clothes retrieval. The learned RoI features after the FN component are highly discriminative with respect to clothes category, pose, and mask. They are fed into MN to obtain features vectors for retrieval, where v1 and v2 are passed into the similarity learning network to obtain the similarity score between the detected clothing items in I1 and I2. Specifically, the feature extractor has 4 'conv' layers, one pooling layer, and one fc layer. The similarity learning network consists of subtraction and square operator and a fc layer, which estimates the probability of whether two clothing items match or not.

Loss Functions

Loss Functions. The parameters Θ of the Match R-CNN are optimized by minimizing five loss functions, which are formulated as $\min_{\Theta} \mathcal{L} = \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{box} + \lambda_3 \mathcal{L}_{pose} + \lambda_4 \mathcal{L}_{mask} + \lambda_5 \mathcal{L}_{pair}$, including a cross-entropy (CE) loss $\mathcal{L}_{cls}$ for clothes classification, a smooth loss [4] $\mathcal{L}_{box}$ for bounding box regression, a CE loss $\mathcal{L}_{pose}$ for landmark estimation, a CE loss $\mathcal{L}_{mask}$ for clothes segmentation, and a CE loss $\mathcal{L}_{pair}$ for clothes retrieval. Specifically, $\mathcal{L}_{cls}$, $\mathcal{L}_{box}$, $\mathcal{L}_{pose}$, and $\mathcal{L}_{mask}$ are identical as defined in [6]. We have $\mathcal{L}_{pair} = -\frac{1}{n}\sum_{i=1}^{n}[y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i)]$, where $y_i = 1$ indicates the two items of a pair are matched, otherwise $y_i = 0$.

Benchmarks

Clothes Detection. This task detects clothes in an image by predicting bounding boxes and category labels. The evaluation metrics are the bounding box's average precision $\mathrm{AP}_{\mathrm{box}}$, $\mathrm{AP}_{\mathrm{box}}^{\mathrm{IoU}=0.50}$, and $\mathrm{AP}_{\mathrm{box}}^{\mathrm{IoU}=0.75}$ by following COCO [11].

Landmark Estimation. This task aims to predict landmarks for each detected clothing item in an each image. Similarly, we employ the evaluation metrics used by COCO for human pose estimation by calculating the average precision for keypoints $\mathrm{AP}_{\mathrm{pt}}$, $\mathrm{AP}_{\mathrm{pt}}^{\mathrm{OKS}=0.50}$, and $\mathrm{AP}_{\mathrm{pt}}^{\mathrm{OKS}=0.75}$, where OKS indicates the object landmark similarity.

Segmentation. This task assigns a category label (including background label) to each pixel in an item. The evaluation metrics is the average precision including $\mathrm{AP}_{\mathrm{mask}}$, $\mathrm{AP}_{\mathrm{mask}}^{\mathrm{IoU}=0.50}$, and $\mathrm{AP}_{\mathrm{mask}}^{\mathrm{IoU}=0.75}$ computed over masks.

Results

	scale			occlusion			zoom-in			viewpoint			overall
	small	moderate	large	slight	medium	heavy	no	medium	large	no wear	frontal	side or back	
$\mathrm{AP}_{\mathrm{mask}}$	0.634	**0.700**	0.669	**0.720**	0.674	0.389	**0.703**	0.627	0.526	0.695	**0.697**	0.617	0.680
$\mathrm{AP}_{\mathrm{mask}}^{\mathrm{IoU}=0.50}$	0.831	**0.900**	0.844	**0.900**	0.878	0.559	**0.899**	0.815	0.663	0.829	**0.886**	0.843	0.873
$\mathrm{AP}_{\mathrm{mask}}^{\mathrm{IoU}=0.75}$	0.765	**0.838**	0.786	**0.850**	0.813	0.463	**0.842**	0.740	0.613	0.792	**0.834**	0.732	0.812

Table 4. **Clothes segmentation** of Mask R-CNN [6] on different validation subsets, including scale, occlusion, zoom-in, and viewpoint. The evaluation metrics are $\mathrm{AP}_{\mathrm{mask}}$, $\mathrm{AP}_{\mathrm{mask}}^{\mathrm{IoU}=0.50}$, and $\mathrm{AP}_{\mathrm{mask}}^{\mathrm{IoU}=0.75}$. The best performance of each subset is bold.

Implementation

In our experiments, each training image is resized to its shorter edge of 800 pixels with its longer edge that is no more than 1333 pixels. Each minibatch has two images in a GPU and 8 GPUs are used for training. For minibatch size 16, the learning rate (LR) schedule starts at 0.02 and is decreased by a factor of 0.1 after 8 epochs and then 11 epochs, and finally terminates at 12 epochs. This scheduler is denoted as 1x. Mask R-CNN adopts 2x schedule for clothes detection and segmentation where '2x' is twice as long as 1x with the LR scaled proportionally. Then It adopts s1x for landmark and pose estimation where s1x scales the 1x schedule by roughly 1.44x. Match R-CNN uses 1x schedule for consumer-to-shop clothes retrieval. The above models are trained by using SGD with a weight decay of 10^{-5} and momentum of 0.9.

Mask R-CNN Segmentation