1 Methodology

1.1 Architecture

The overall architecture of our method is depicted in Figure 1, which contains three main components: a CNN backbone to extract low-level feature representation of a video frame; a Transformer encoder to capture long-range dependencies within a video frame globally; a Head to realize the actual downstream task, predicting heatmaps of hand keypoints. It is similar to the architecture proposed in [15], which have achieved impressive performance in human pose estimation task. Inspired by it, we want to further explore the performance of Transformer on hand keypoint detection task.

Backbone. We choose two common CNNs as the backbone: ResNet [5] and HRNet [10], which are pre-trained on ImageNet [3]. Given the video frame input $x_{frame} \in R^{3\times H_0\times W_0}$, the CNN backbone generates a corresponding feature map $f \in R^{d\times H\times W}$, in which $d = 256$ and $H, W = \frac{H_0}{8}, \frac{W_0}{8}$.

Transformer encoder. The feature map $f \in R^{d\times H\times W}$ output by the CNN backbone is flattened into a sequence $s \in R^{d\times HW}$ first, since the Transformer encoder expects a sequence as input. The Transformer encoder contains N uniform layers, and each layer consists of two sub-layers as [13]. The first sub-layer is a multi-head self-attention, and the second sub-layer is a position-wise feed-forward network. Also, residual connection and layer normalization are employed to each sub-layer.

Head. The final prediction is computed by a Head module which outputs keypoints heatmaps and each heatmap is used to predict one keypoint location. The Head first converts the encoder output sequence $t = R^{d\times HW}$ to initial feature map shape $z \in R^{d\times H\times W}$. Then the Head reduce the channel of z to k by 1×1 convolution and generate k keypoints heatmaps $h \in R^{k\times\widetilde{H}\times\widetilde{W}}$, where $\widetilde{H} \times \widetilde{W} = H_0/\lambda, W_0/\lambda$. λ is the downsampling ratio which is empirically set to 4.

1.2 Heatmap Regression

Heatmap regression is widely used in the keypoint detection task [11, 14, 2, 1]. It was first introduced in [11] and rapidly became the mainstream method for keypoint detection. It uses Mean Squared Error(MSE) to minimize the distance between the predicted heatmap and the ground-truth heatmap. Also, heatmap regression includes two processes: the encoding process and the decoding process [17]. The encoding process refers to transforming the ground-truth keypoints to heatmaps and the decoding process refers to transforming the predicted heatmaps into the final keypoint coordinates in the original image space. The heatmap follows a 2D Gaussian distribution as:

$$G(x, y) = exp^{-\frac{(x-u)^2+(y-v)^2}{2\sigma^2}}$$

where (u, v) is the ground-truth keypoint coordinate of a joint, (x, y) specifies a pixel location in the heatmap, and σ is a fixed spatial variance.

1.3 Loss Function

We use pixel-wise Mean Squared Error as the loss function to calculate the gap between the predicted heatmaps and the ground-truth heatmaps:

$$L(G, \hat{G}) = \sum_{i=1}^{K}\sum_{y=1}^{\widetilde{H}}\sum_{x=1}^{\widetilde{W}}(G_i(x_i, y_i) - \hat{G}_i(x_i, y_i))^2/K/\widetilde{H}/\widetilde{W}$$

where K is the number of keypoints, $\widetilde{H}$ and $\widetilde{W}$ are the height and width of a heatmap.

1.4 Activation Maximization

Activation Maximization [4, 9] is a method to find out the location where the heatmap is maximally activated. In our task, the activation maximum location is the keypoint location, which is computed by:

$$\hat{m} = argmax(\hat{G})$$

where $\hat{G}$ is the heatmap Gaussian distribution and $\hat{m}$ is the final predicted keypoint location.

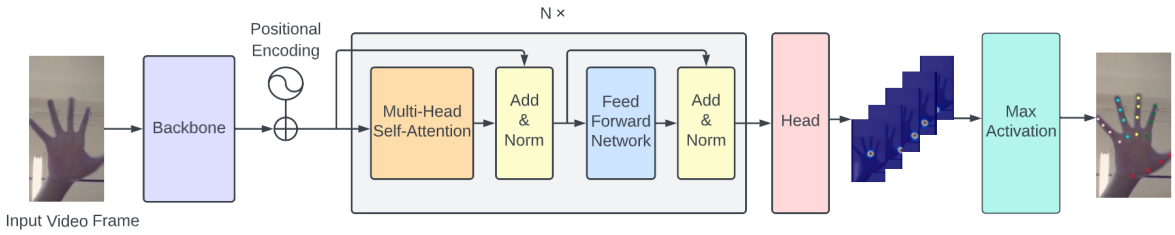

Figure 1: The architecture.

2 Experiments

Technical details. We train the model with Adam optimizer [8] and cosine annealing learning rate decay scheduler. The learning rate decays from 10^{-4} to 10^{-5}. We adopt mini-batch mode and batch size of 20. The backbone is ImageNet-pretrained ResNet model [5] or HRNet model [10], which retains only the initial parts for faster training. We resize all the video frames into 256×192 resolution proportionally. Since most video frames' size is not proportional to the target size because of mobile phone configuration, paddings are added to the left and right of the shorter frame side. Position encodings are added to the inputs of the Transformer encoder layers to retain positional information. We use 2D sine fixed position encoding as the default position encoding. The Transformer is trained with a 0.1 dropout and ReLU activation function.

2.1 Results

We use Percentage of Correct Keypoints (PCK) as the evaluation metric. If the predicted keypoint lies within a distance threshold σ of its ground-truth location, the estimation is considered to be correct. We set σ to $\alpha \times max(h, w)$ as [12, 6], where α is a fraction and (h, w) is the output heatmap size. The smaller the value of α, the stricter the evaluation metric will be. In our experiment, we set the α to 0.03. For a particular keypoint i, we estimate it by PCK and approximate it on the testing set T as:

$$PCK_{\alpha}^{i} = \frac{\sum_{T} \delta(\|x_i - y_i\|_2 < \sigma)}{\mid T \mid}$$

where x_i is the i-th predicted keypoint and y_i is its ground-truth location on a video frame.

To measure the smooth predictions, we follow the related works [7, 16] to adopt *acceleration error*. For 2D datasets, this measures the average difference between ground-truth acceleration and predicted acceleration of each joint in $pixel/frame^2$.

References

[1] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial posenet: A structure-aware convolutional network for human pose estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1221–1230, 2017.

[2] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[4] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[6] Junguang Jiang, Yifei Ji, Ximei Wang, Yufeng Liu, Jianmin Wang, and Mingsheng Long. Regressive domain adaptation for unsupervised keypoint detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6780–6789, 2021.

[7] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5614–5623, 2019.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[10] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019.

[11] Jonathan J Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[12] Shubham Tulsiani and Jitendra Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[14] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11802–11812, 2021.

[16] Ailing Zeng, Lei Yang, Xuan Ju, Jiefeng Li, Jianyi Wang, and Qiang Xu. Smoothnet: a plug-and-play network for refining human poses in videos. In *European Conference on Computer Vision*, pages 625–642. Springer, 2022.

[17] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.