Weighted Feature Pooling Network in Template-Based Recognition

Zekun Li, Yue Wu, Wael Abd-Almageed and Prem Natarajan
zekunl@isi.edu, yue_wu@isi.edu, wamageed@isi.edu , pnataraj@isi.edu
Information Sciences Institute

1. Introduction

Many computer vision tasks are template-based learning tasks in which multiple instances of a specific concept (e.g. multiple images of a subject's face) are available at once to the learning algorithm. The template structure of the input data provides an opportunity for generating a robust and discriminative unified template-level representation that effectively exploits the inherent diversity of feature-level information across instances within a template. In contrast to other statistical feature pooling and neural-network based aggregation methods, we propose a new technique to dynamically predict weights that consider factors such as noise and redundancy in assessing the importance of image-level features and use those weights to appropriately aggregate the features into a single template-level representation.

2. Motivation

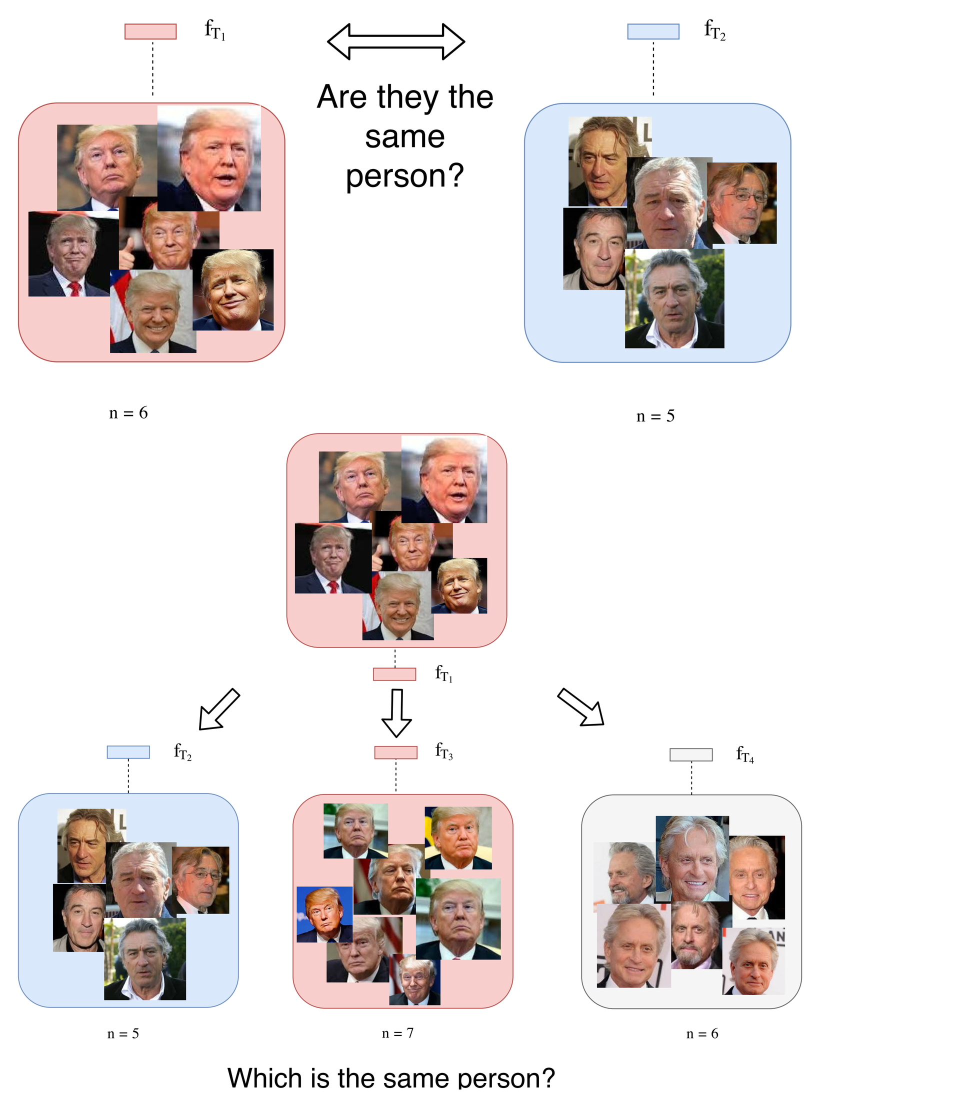

Figure 1: Template-based face verification and searching

Problem: How to construct discriminative template-level feature f_T given image-level features $\{f_{x_1}, f_{x_2}, ... f_{x_n}\}$?

3. Weighted Feature Pooling Network Structure

One of the main design goals of WFPN is to easily integrate (as a plug and play module) into existing network architectures, without changes to the underlying network. The input of WFPN is a template, and output depends on the specific task. The feature extraction block and task-specific block in this model are initialized from their corresponding base models. Built upon the base model, WFPN uses a weight predictor $\mathcal{P}$ to predict the importance of image-level features, and a fusion layer $\mathcal{M}$ takes predicted weights together with image features to compute the final template representation.

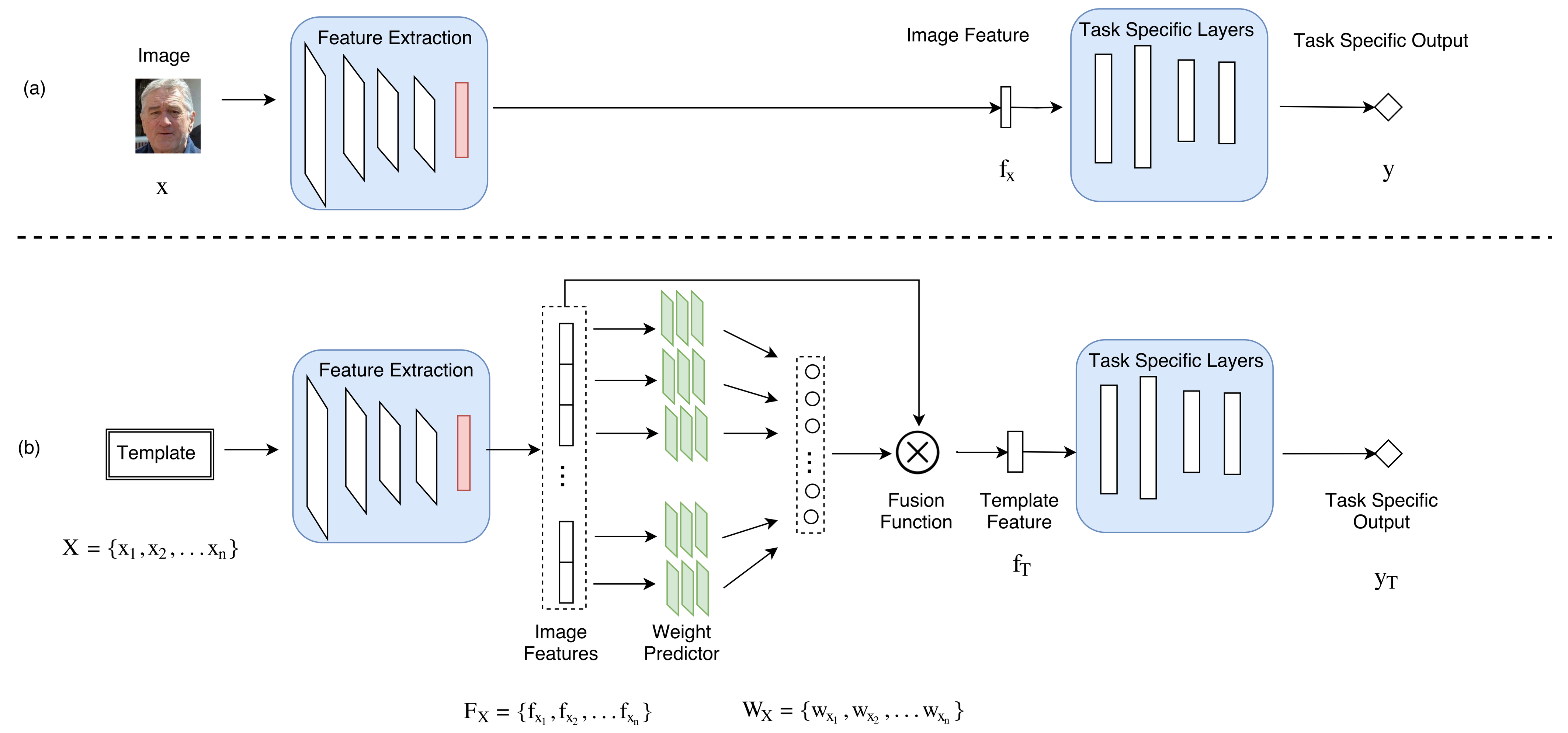

Figure 2: WFPN can be constructed upon an arbitrary image-based learning network. (a) Image-based learning network (base model). (b) Template-based weighted feature pooling network (WFPN). The tiny red block in feature extractor indicates split node. Weight predictor networks (green layers) have the same structure and shared weights.

4. Weight Predictor

The weight predictor looks at each image feature individually and predicts a value that expresses the absolute significance of this feature. A weight predictor has L fully-connected layers can be formally defined as below.

$$a_{x_i}^{(1)} = W^{(1)} f_{x_i} + b^{(1)} \quad (1)$$

$$g_{x_i}^{(1)} = max\{0, a_{x_i}^{(1)}\} \quad (2)$$

$$a_{x_i}^{(2)} = W^{(2)} g_{x_i}^{(1)} + b^{(2)} \quad (3)$$

$$g_{x_i}^{(2)} = max\{0, a_{x_i}^{(2)}\} \quad (4)$$

$$\ldots \quad (5)$$

$$a_{x_i}^{(L)} = W^{(L)} g_{x_i}^{(L-1)} + b^{(L)} \quad (6)$$

$$w_{x_i} = \frac{exp(a_{x_i}^{(L)})}{\sum_j exp(a_{x_j}^{(L)})} \quad (7)$$

Notice that $W^{(L)}$ is of shape $(1 \times D)$ where D is the dimension of $g_{x_i}^{(L-1)}$. This constraint ensures that w_{x_i} is scalar.

5. WFPN for Digit Recognition

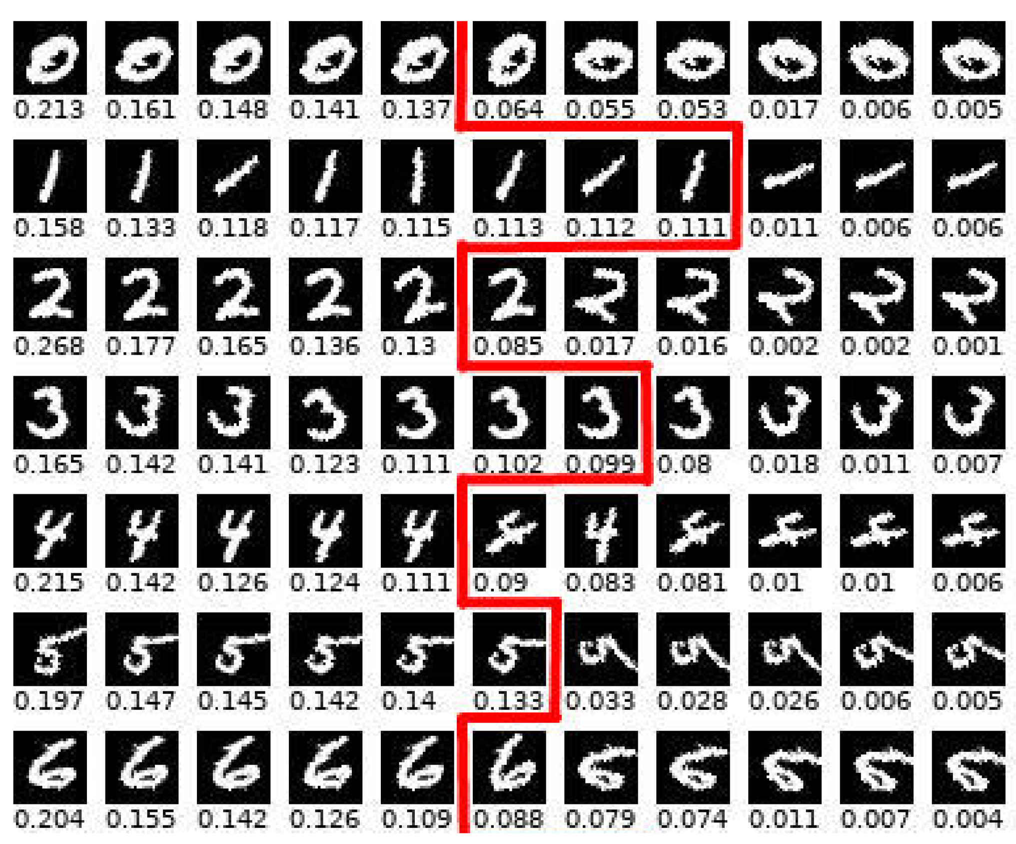

Figure 3: Predicted weights on MNIST. (Red line splits the images of weights above and below the average)

6. WFPN for Face Recognition

Figure 4: Low resolution images have smaller weights

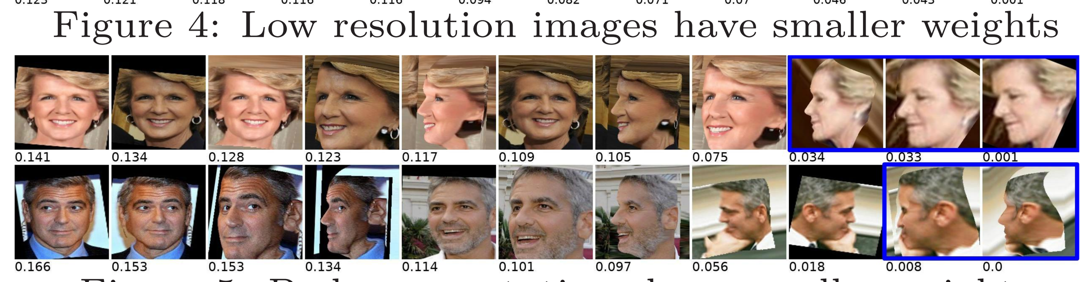

Figure 5: Bad augmentations have smaller weights

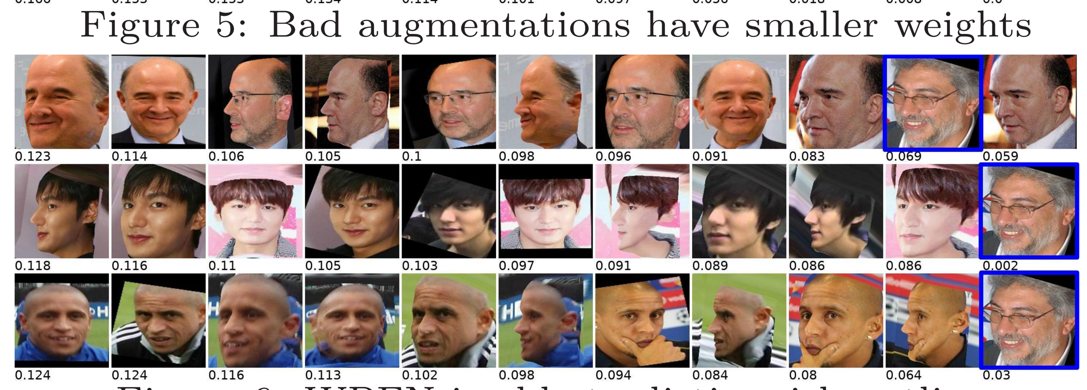

Figure 6: WPFN is able to distinguish outliers

7. WFPN for Object Recognition

ACC(%)	img.	tsize	avg.	vote	[1]	ours
Res20	91.42	3	91.47	90.06	91.64	**92.49**
		5	92.09	91.59	92.09	**93.17**
		8	91.88	90.98	92.71	**93.17**
Res18	89.41	3	89.80	88.85	89.69	**90.02**
		5	90.66	89.97	90.36	**90.89**
		8	90.47	89.40	90.63	**90.84**
VGG16	93.59	3	93.69	93.32	93.54	**93.84**
		5	93.88	93.62	**94.11**	93.92
		8	93.66	93.39	93.88	**94.02**

Table 1: WFPN on **CIFAR10**. WFPN consistently outperforms baselines and NAN with diff. base network structures

8. WFPN for Activity Recognition

	Two-Strm [2]	Conv-TS [3]	TSN [4]
RGB	73.0	82.61	84.5
Flow	83.7	86.25	87.2
Joint	88.0	90.62	92.0
	HiddenTSN [5]	I3D [6]	Ours
RGB	85.7	92.2*	**94.1**
Flow	86.3	94.7*	**96.3**
Joint	92.5	96.0*	**97.8**

Table 2: Comparison of WFPN and other methods on UCF101 dataset. (WFPN initialized from imagenet + kinetics pretrained weights, *: Keras implementation)

9. References

[1] Jiaolong Yang, Peiran Ren, Dong Chen, Fang Wen, Hongdong Li, and Gang Hua. Neural aggregation network for video face recognition. *arXiv preprint arXiv:1603.05474*, 2016.

[2] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[3] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[4] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[5] Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander G Hauptmann. Hidden two-stream convolutional networks for action recognition. *arXiv preprint arXiv:1704.00389*, 2017.

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733. IEEE, 2017.

10. Acknowledgements

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA 2014- 14071600010.