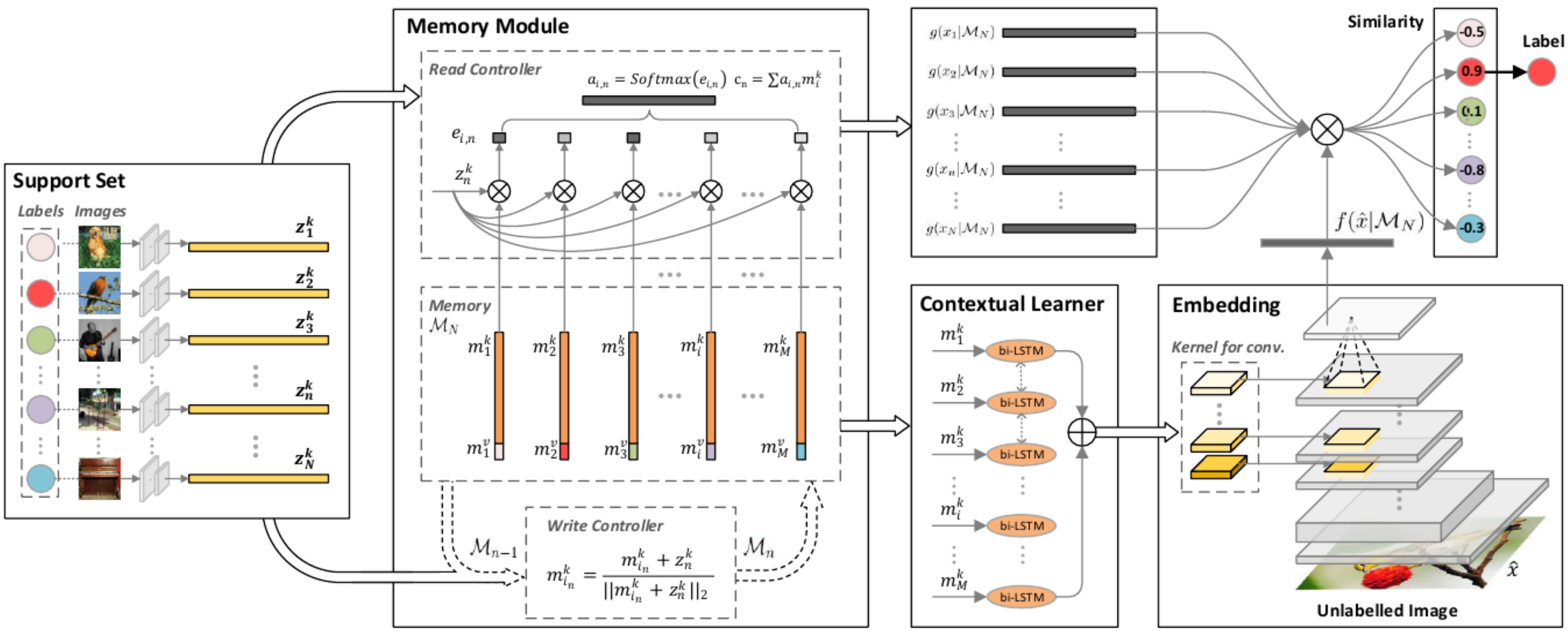
One-shot Learning Application

黄江雷

Memory Matching Networks for One-Shot Image Recognition*

Qi Cai †, Yingwei Pan †, Ting Yao ‡, Chenggang Yan §, and Tao Mei ‡
† University of Science and Technology of China, Hefei, China
‡ Microsoft Research, Beijing, China
§ Hangzhou Dianzi University, Hangzhou, China
{cqcaiqi, panyw.ustc}@gmail.com, {tiyao, tmei}@microsoft.com, cgyan@hdu.edu.cn



Memory Matching Networks for One-Shot Image Recognition*

Table 1. Mean accuracy (%) ± CIs (%) of our MM-Net and other state-of-the-art methods on Omniglot dataset.

Model	5-way Accuracy		20-way Accuracy	
	1-shot	5-shot	1-shot	5-shot
SN [16]	97.3	98.4	88.2	97.0
MN [31]	98.1	98.9	93.8	98.5
MANN [27]	82.8	94.9	—	—
SM [14]	98.4	99.6	95.0	98.6
Meta-N [21]	98.95	—	97.00	—
MAML [9]	98.7 ± 0.4	$\mathbf{99.9 \pm 0.1}$	95.8 ± 0.3	$\mathbf{98.9 \pm 0.2}$
MM-Net	$\mathbf{99.28 \pm 0.08}$	99.77 ± 0.04	$\mathbf{97.16 \pm 0.10}$	$\mathbf{98.93 \pm 0.05}$

Table 2. Mean accuracy (%) ± CIs (%) of our MM-Net and other state-of-the-art methods on *mini*ImageNet dataset.

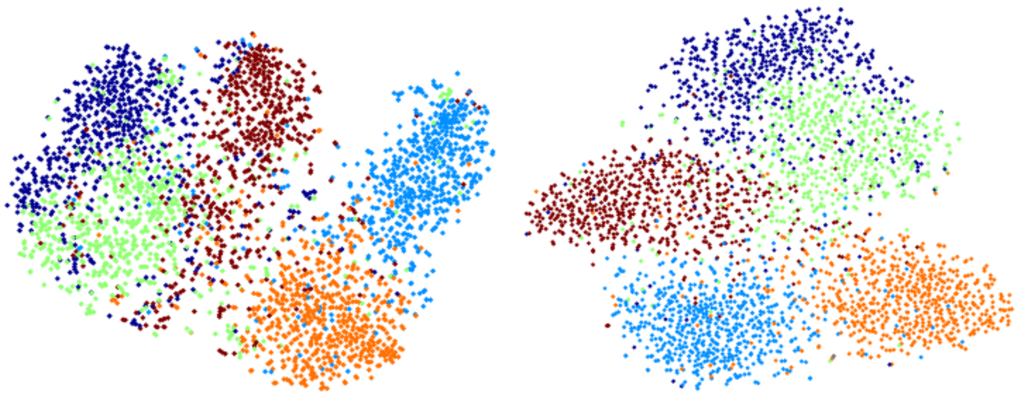
Model	5-way Accuracy	
	1-shot	5-shot
MN [31]	43.40 ± 0.78	51.09 ± 0.71
MN-FCE [31]	43.56 ± 0.84	55.31 ± 0.73
ML-LSTM [25]	43.44 ± 0.77	60.60 ± 0.71
MAML [9]	48.70 ± 1.84	63.11 ± 0.92
Meta-N [21]	49.21 ± 0.96	—
MM-Net$^-$	52.74 ± 0.45	65.82 ± 0.37
MM-Net	$\mathbf{53.37 \pm 0.48}$	$\mathbf{66.97 \pm 0.35}$

Memory Matching Networks for One-Shot Image Recognition*

Table 3. Mean accuracy (%) of MM-Net by varying training strategies for 5-way k-shot image recognition task ($k \in \{1, 2, 3, 4, 5\}$) on *mini*ImageNet.

Train	**Test**				
	1-shot	2-shot	3-shot	4-shot	5-shot
1-shot	**52.74**	57.53	59.31	60.02	60.33
2-shot	52.68	**59.14**	62.11	63.39	63.92
3-shot	51.67	58.48	**62.21**	64.03	65.40
4-shot	51.44	58.56	62.12	**64.48**	65.77
5-shot	51.09	58.03	61.80	64.14	**65.82**
Mixed k-shot	52.83	59.88	63.31	65.32	66.71
Mixed C-way k-shot	**53.37**	**59.93**	**63.35**	**65.49**	**66.97**

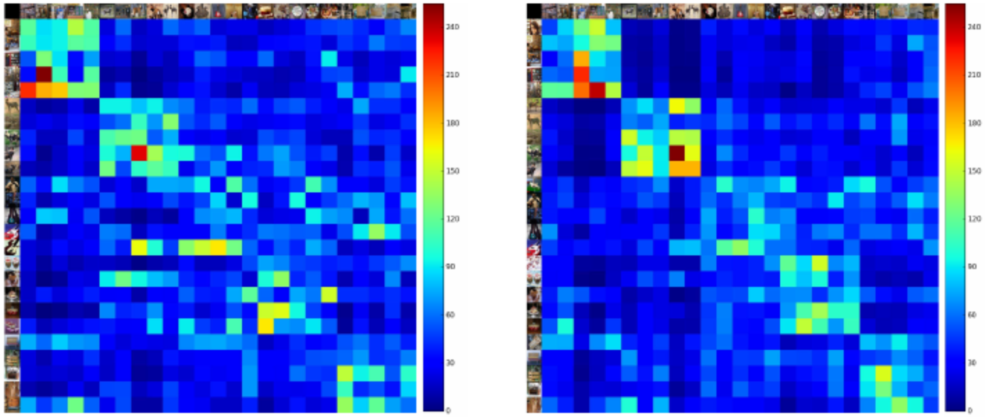
Memory Matching Networks for One-Shot Image Recognition*



(a) MN [31] (b) MM-Net

Figure 3. Image representation embedding visualizations of MN and our MM-Net on *mini*Imagenet using t-SNE [19]. Each image is visualized as one point and colors denote different classes.

Memory Matching Networks for One-Shot Image Recognition*



(a) MN [31] (b) MM-Net

Figure 4. Similarity matrix of MN and our MM-Net on *miniImagenet* (vertical axis: 5 labelled images per class in support set; horizontal axis: 5 unlabelled test images per class). The warmer colors indicate higher similarities.

Learning feed-forward one-shot learners

Luca Bertinetto*
Torr Vision Group
University of Oxford
luca@robots.ox.ac.uk

João F. Henriques*
Visual Geometry Group
University of Oxford
joao@robots.ox.ac.uk

Jack Valmadre*
Torr Vision Group
University of Oxford
jvlmdr@robots.ox.ac.uk

Philip H. S. Torr
Torr Vision Group
University of Oxford
philip.torr@eng.ox.ac.uk

Andrea Vedaldi
Visual Geometry Group
University of Oxford
vedaldi@robots.ox.ac.uk

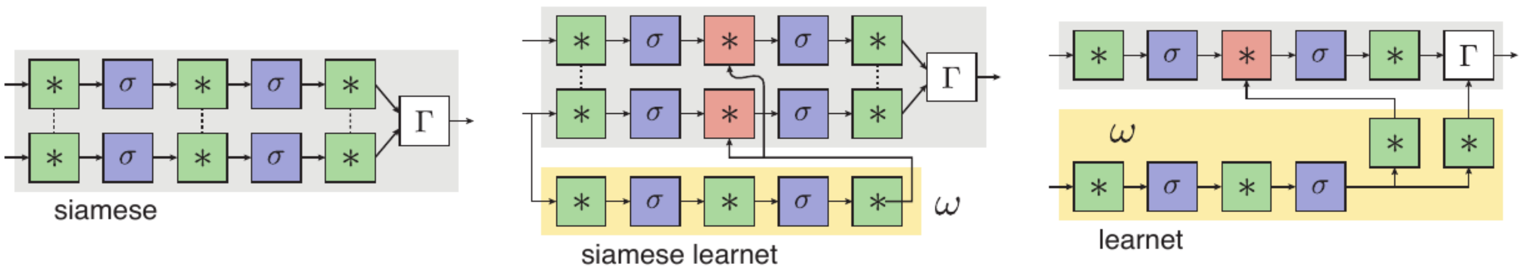


Figure 2: Our proposed architectures predict the parameters of a network from a single example, replacing static convolutions (green) with dynamic convolutions (red). The siamese learnet predicts the parameters of an embedding function that is applied to both inputs, whereas the single-stream learnet predicts the parameters of a function that is applied to the other input. Linear layers are denoted by $*$ and nonlinear layers by σ. Dashed connections represent parameter sharing.

Learning feed-forward one-shot learners

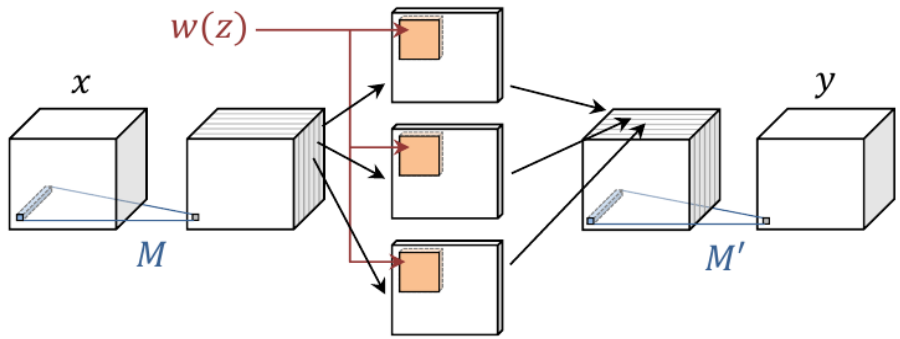


Figure 1: Factorized convolutional layer (eq. (8)). The channels of the input x are projected to the factorized space by M (a 1×1 convolution), the resulting channels are convolved independently with a corresponding filter prediction from $w(z)$, and finally projected back using M'.

$$y = w(z)x + b(z).$$

$$y = M' \operatorname{diag}(w(z)) Mx + b(z).$$

$$y = W * x + b,$$

$$y = M' * w(z) *_d M * x + b(z),$$

Learning feed-forward one-shot learners

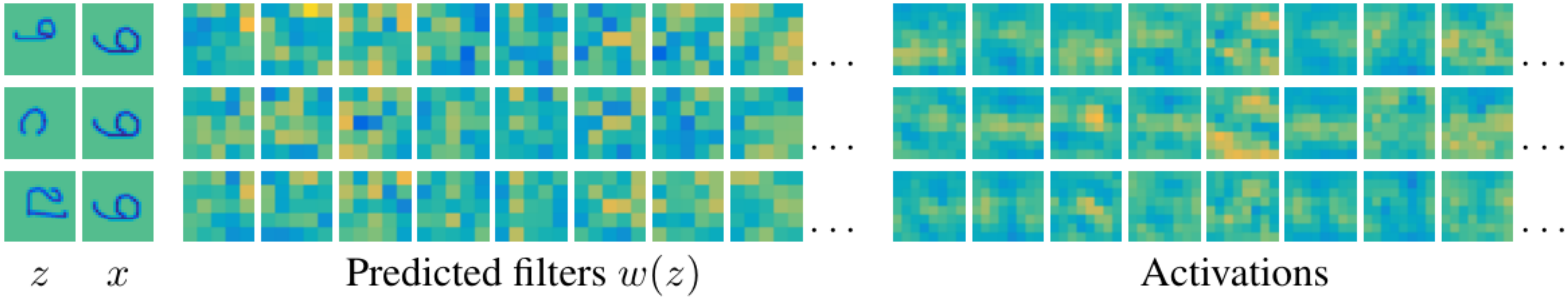


Figure 3: The predicted filters and the output of a dynamic convolutional layer in a single-stream learnet trained for the OCR task. Different exemplars z define different filters $w(z)$. Applying the filters of each exemplar to the same input x yields different responses (although in typical operation, the network defined by a single exemplar is applied to many other inputs). Best viewed in colour.

	Inner-product (%)	Euclidean dist. (%)	Weighted ℓ^1 dist. (%)
Siamese (shared)	48.5	37.3	41.8
Siamese (unshared)	47.0	41.0	34.6
Siamese (unshared, factorized)	48.4	–	33.6
Siamese learnet (shared)	51.0	39.8	31.4
Learnet	43.7	36.7	**28.6**

Table 1: Error rate for character recognition in foreign alphabets (chance is 95%).

Learning feed-forward one-shot learners

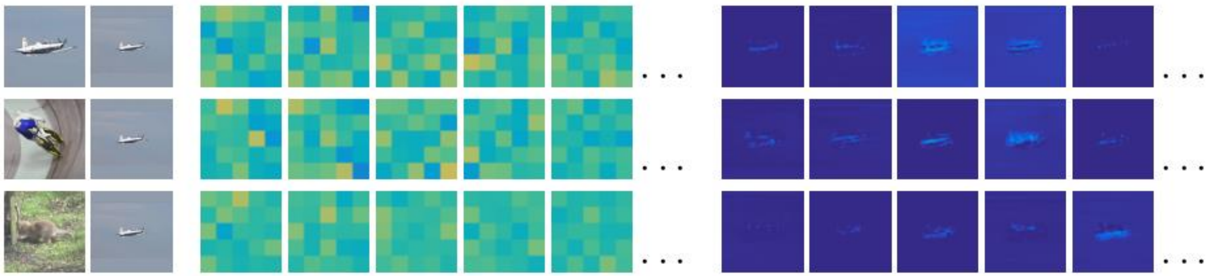


Figure 4: The predicted filters and the output of a dynamic convolutional layer in a siamese learnet trained for the object tracking task. Best viewed in colour.

Method	Accuracy	Failures
Siamese (φ=B)	0.465	105
Siamese (φ=B; unshared)	0.447	131
Siamese (φ=B; factorized)	0.444	138
Siamese learnet (φ=B; ω=A)	**0.500**	**87**
Siamese learnet (φ=B; ω=B)	0.497	93
DAT [17]	0.442	113
SO-DLT [21]	0.540	108

Method	Accuracy	Failures
Siamese (φ=C)	0.466	120
Siamese (φ=C; factorized)	0.435	132
Siamese learnet (φ=C; ω=A)	0.483	**105**
Siamese learnet (φ=C; ω=C)	**0.491**	106
DSST [2]	0.483	163
MEEM [22]	0.458	107
MUSTer [6]	0.471	132

Table 2: Tracking accuracy and number of tracking failures in the VOT 2015 Benchmark, as reported by the toolkit [10]. Architectures are grouped by size of the main network (see text). For each group, the best entry for each column is in bold. We also report the scores of 5 recent trackers.