SUPPLEMENTARY MATERIAL FOR THE PAPER
Factorized Higher-Order CNNs with an Application to Spatio-Temporal Emotion Estimation

Jean Kossaifi*† NVIDIA

Antoine Toisoul* Samsung AI Center

Adrian Bulat Samsung AI Center

Yannis Panagakis† University of Athens

Timothy Hospedales Samsung AI Center

Maja Pantic† Imperial College London

In these supplementary materials, we give additional details on the method, as well as results on other tasks (image classification on CIFAR-10, where we compare with the most similar architecture to ours, MobileNet v2, and gesture estimation from videos).

1. Dimensional model of affect

Discrete emotional classes are too coarse to summarize the full range of emotions displayed by humans on a daily basis. This is the reason why finer, dimensional affect models, such as the valence and arousal are now favoured by psychologists [5]. In this circumplex, which can be seen in Figure 1, the valence level corresponds to how positive or negative an emotion is, while the arousal level explains how calming or exciting the emotion is.

A visualization of the prediction of valence and arousal of our model can be seen in Figure 2, along with some representative frames.

2. Automatic rank selection

Using the automatic rank selection procedure detailed in the method section, we let the model learn end-to-end the rank of each of the factorized higher-order convolutions.

In Figure 3, we show the number of parameters set to zero by the network for a regularization parameter of 0.01 and 0.05. The lasso regularization is an easy way to automatically tune the rank. We found that on average 8 to 15% of the parameters can be set to zero for optimal performance. In practice, about 1 million parameters were removed thanks to this regularization.

In Table 1 we report the number of parameters of spatio-temporal baselines and compare it to our CP factorized model. Besides having less parameters, our approach has the advantage of having a very low number of temporal parameters which facilitates the training on spatio-temporal data once it has been pretrained on static data.

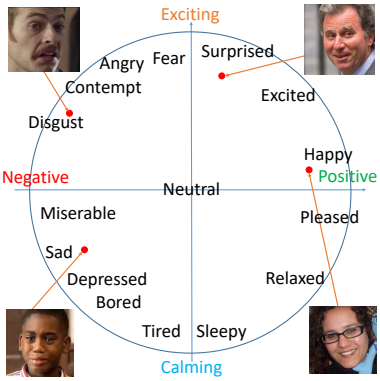

Figure 1: The valence and arousal circumplex. This dimensional model of affect covers the continuous range of emotions displayed by human on a daily basis. The images are taken from the AffectNet dataset [4]

Table 1: **Number of parameters optimized to train the temporal model**

Network	Total # parameters	# parameters removed with LASSO	# parameters optimized for video
ResNet18-(2+1)D	31M	-	31M
ResNet-18-3D	33M	-	33M
Ours [$\lambda = 0.01$]	11M	0.7	0.24
Ours [$\lambda = 0.05$]	11M	1.3M	0.24

*Joint first authors.

†Jean Kossaifi, Yannis Panagakis and Maja Pantic were with Samsung AI Center, Cambridge and Imperial College London.

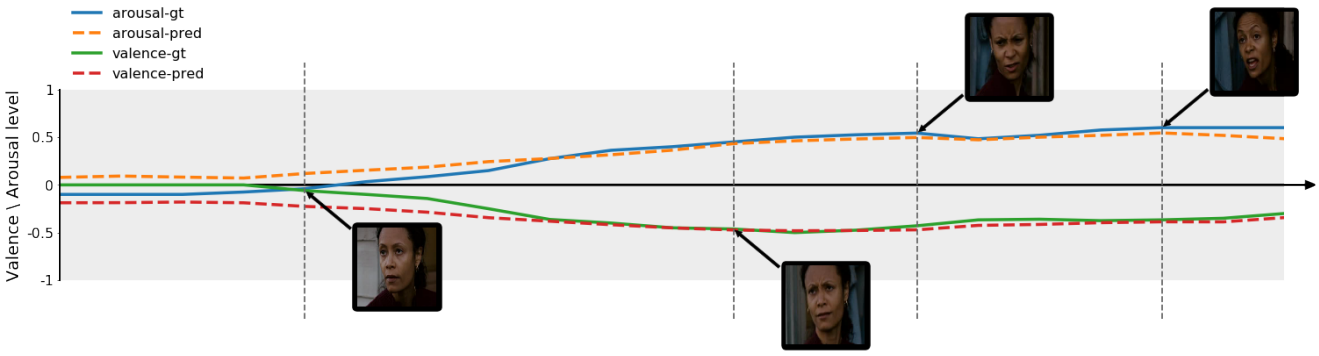

Figure 2: **Evolution of the ground-truth (*gt*) and predicted (*pred*) levels of valence and arousal** as a function of time, for one of the test videos of the *AFEW-VA* dataset.

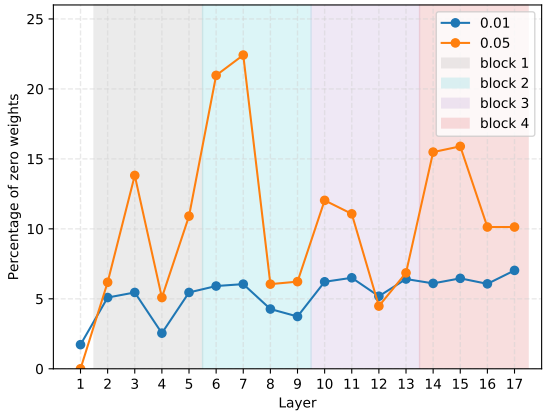

Figure 3: Sparsity induced by the automatic rank selection at each layer of the network (ResNet-18 backbone).

3. Results on LSEMSW

LSEMSW [2] is the Large-scale Subtle Emotions and Mental States in the Wild database. It contains more than $175,000$ static images annotated in terms of subtle emotions and cognitive states such as helpless, suspicious, arrogant, etc. We report results on LSEMSW in table. 2.

Table 2: Results on the LSEMSW database

Method	**Accuracy**
ResNet 34 [2]	28.39 %
Ours	**34.55%**

4. Initialization of transduction factors

When performing transduction, the added temporal factors to the CP convolutions are initialized to a constant value of one. This corresponds to performing static prediction for each frame. In a first step, only these factors are optimized while the spatial factors and the weights of the CP convolutions are kept unchanged. Then in a second step every parameter of the network is optimized. This prevents the new transducted factors from pulling the existing parameters out of their local mimimum which would cause the network to forget the knowledge learnt on the static case (*catastrophic forgetting*). The full training with transduction is summarized in Figure 4.

5. Loss function

Our goal is to maximize the correlations coefficients PCC and CCC, as these metrics are the most commonly employed in continuous affect estimation. However minimizing the RMSE also helps maximizing the correlations as it gives a lower error in each individual prediction. Our regression loss function reflects this by incorporating three losses :

$$\mathcal{L} = \frac{\alpha}{\alpha+\beta+\gamma}\mathcal{L}_{RMSE} + \frac{\beta}{\alpha+\beta+\gamma}\mathcal{L}_{PCC} + \frac{\gamma}{\alpha+\beta+\gamma}\mathcal{L}_{CCC}.$$

$$\mathcal{L}_{RMSE} = \text{RMSE}_{valence} + \text{RMSE}_{arousal} \quad (1)$$

$$\mathcal{L}_{PCC} = 1 - \frac{\text{PCC}_{valence} + \text{PCC}_{arousal}}{2} \quad (2)$$

$$\mathcal{L}_{CCC} = 1 - \frac{\text{CCC}_{valence} + \text{CCC}_{arousal}}{2} \quad (3)$$

The coefficients α, β and γ are shake-shake regularization coefficients [1] taken randomly in the range $[0; 1]$ following a uniform distribution. These coefficients allow the network to minimize each loss without focusing on one of

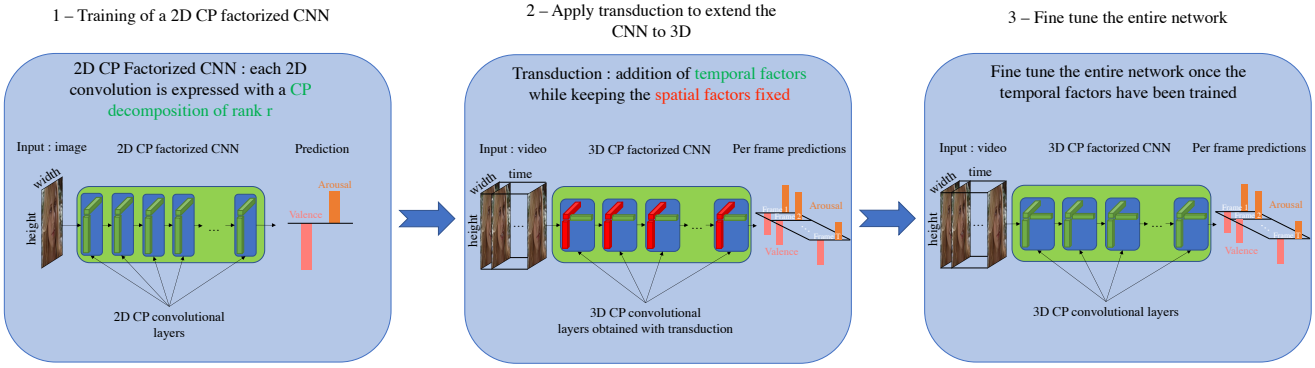

Figure 4: Overview of our method. **(Left)** We start by training a 2D CNN with our proposed factorized convolutional block, on static images. **(Middle)** We then apply transduction to that model to extend it from the static to the spatio-temporal domain. The pretrained spatial factors are kept fixed. **(Right)** Once the temporal factors have been trained, we fine tune them along with the spatial factors in order to improve the final performance.

them while leaving others out. Note that on AffectNet discrete classes of emotions are available. We therefore further jointly perform a regression of the valence and arousal values as well as a classification of the emotional class. The classification is done by adding a cross entropy to the loss function.

6. Results on CIFAR-10

While in the paper we focus on affect estimation, we report here results on a traditional image classification dataset, CIFAR 10.

CIFAR-10 [3] is a dataset for image classification composed of 10 classes with $6,000$ images which, divided into 5000 images per class for training and 1000 images per class for testing, on which we report the results.

We used a MobileNet-v2 as our baseline. For our approach, we simply replaced the full MobileNet-v2 blocks with ours (which, in the 2D case, differs from MobileNet-v2 by the use of two separable convolutions along the spatial dimensions instead of a single 2D kernel). We kept all the parameters the same for all experiments to allow for fair comparison and reported performance averaged across 3 runs. The standard deviation was 0.033 for MobileNet-v2 and 0.036 for our approach. We optimized the loss using stochastic gradient descent with a mini-batch size of 128, starting with a learning rate of 0.1, decreased by a factor of 10 after 150 and 250 epochs, for a total of 400 epochs, with a momentum of 0.9. For MobileNet-v2, we used a weight decay of $4e^{-5}$ and $1e^{-4}$ for our approach. Optimization was done on a single NVIDIA 1080Ti GPU.

We compared our method with a MobileNet-v2 with a comparable number of parameters, Table 3. Unsurprisingly, both approach yield similar results since, in the 2D case, the two networks architectures are similar. It is worth noting that our method has marginally less parameters than MobileNet-v2, for the same number of channels, even though that network is already optimized for efficiency.

Table 3: **Results on the CIFAR-10 dataset**

Network	**# parameters**	**Accuracy (%)**
MobileNet-v2	2.30M	94
Ours	2.29M	94

7. Results on gesture estimation

In this section, we report additional results on the Jester dataset. We compare a network that employs regular convolutional blocks to the same network that uses our proposed higher-order factorized convolutional blocks.

20BN-Jester v1 is a dataset[1] composed of $148,092$ videos, each representing one of 27 hand gestures (e.g. *swiping left*, *thumb up*, etc). Each video contains a person performing one of gestures in front of a web camera. Out of the $148,092$ videos $118,562$ are used for training, $14,787$ for validation on which we report the results.

For the 20BN-Jester dataset, we used a convolutional column composed of 4 convolutional blocks with kernel size $3 \times 3 \times 3$, with respective input and output of channels: $(3, 64)$, $(64, 128)$, $(128, 256)$ and $(256, 256)$, followed by two fully-connected layers to reduce the dimensionality to 512 first, and finally to the number of classes. Between each convolution we added a batch-normalisation layer, non-linearity (ELU) and $2 \times 2 \times 2$ max-pooling. The full architecture is graphically represented in Figure 5 of the architecture detailed in the paper, for clarity.

[1]Dataset available at https://www.twentybn.com/datasets/jester/v1.

Figure 5: **Architecture of our 3D convolutional network**. We employed the same architecture for both our baseline and our approach, where the only difference is the 3D convolutional block used (3D Conv B): for the baseline a regular 3D conv, and for our method, our proposed HO-CP conv-S. Each convolution is followed by a batch-normalisation, non-linearity (ELU) and a max pooling (over $2 \times 2 \times 2$ non-overlapping regions).

For our approach, we used the same setting but replaced the 3D convolutions with our proposed block and used, for each layer, $6 \times n_{\text{input-channels}}$ for the rank of the HO-CP convolution. The dataset was processed by batches of 32 sequences of RGB images, with a temporal resolution of 18 frames and a size of 84×84. The loss is optimized by mini-batches of 32 samples using stochastic gradient descent, with a starting learning-rate of 0.001, decreased by a factor of 10 on plateau, a weight decay of $1e^{-5}$ and momentum of 0.9. All optimization was done on 2 NVIDIA 1080Ti GPUs.

Table 4: **Results on the 20BN-Jester Dataset**

Network	#conv parameters	Accuracy (%)	
		Top-1	Top-5
3D-ConvNet	2.9M	83.2	97.0
HO-CP ConvNet (**Ours**)	1.2M	83.8	97.4
HO-CP ConvNet-S (**Ours**)	1.2M	**85.4**	**98.6**

Results for 3D convolutional networks For the 3D case, we test our Higher-Order CP convolution with a regular 3D convolution in a simple neural network architecture, in the same setting, in order to be able to compare them. Our approach is more computationally efficient and gets better performance as shown in Table 4. In particular, the basic version without skip connection and with RELU (emphHO-CP ConvNet) has 1.7 million less parameters in the convolutional layers compared to the regular 3D network, and yet, converges to better Top-1 and Top-5 accuracy. The version with skip-connection and PReLU (*HO-CP ConvNet-S*) beats all approaches.

8. Algorithm for our CP convolutions

We summarize our efficient higher-order factorized convolution in algorithm 1.

Algorithm 1 Higher-order CP convolutions

Require: Input activation tensor $\mathcal{X} \in \mathbb{R}^{C \times D_0 \times \cdots \times D_N}$
CP kernel weight tensor $\mathcal{W} = [\![\mathbf{U}^{(T)}, \mathbf{U}^{(C)}, \mathbf{U}^{(K_0)}, \cdots, \mathbf{U}^{(K_N)}]\!]$
Skip connection weight matrix $\mathbf{U}^{(S)} \in \mathbb{R}^{O \times C}$
Ensure: Efficient N-D convolution on $\mathcal{X}$
Efficient $1D$ convolution on the channels: $\mathcal{H} \Leftarrow \mathcal{X} \times_0 \mathbf{U}^{(C)}$
for i:=1 **to** $N-2$ **do**
$\mathcal{H} \Leftarrow \mathcal{H} \star_i \mathbf{U}^{(K_i)}$ (1–D conv along the i$^{\text{th}}$mode)
$\mathcal{H} \Leftarrow$ PReLU $(\mathcal{H})$ or ReLU $(\mathcal{H})$ [optional]
$\mathcal{H} \Leftarrow$ Batch-Norm $(\mathcal{H})$ [optional]
end for
Efficient $1D$ convolution from the rank to the output number of channels: $\mathcal{H} \Leftarrow \mathcal{H} \times_1 \mathbf{U}^{(T)}$
if Skip-connection **then**
return $\mathcal{H} + \mathcal{X} \times_0 \mathbf{U}^{(S)}$
else
return $\mathcal{H}$
end if

References

[1] Xavier Gastaldi. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017. 2

[2] Guosheng Hu, Li Liu, Yang Yuan, Zehao Yu, Yang Hua, Zhihong Zhang, Fumin Shen, Ling Shao, Timothy Hospedales, Neil Robertson, et al. Deep multi-task learning to recognise subtle facial expressions of mental states. In *ECCV*, pages 103–119, 2018. 2

[3] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 3

[4] Ali Mollahosseini, Behzad Hasani, and Mohammad H Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, 10(1):18–31, 2017. 1

[5] J.A. Russell, Bachorowski J.A, and J.M. Fernandez-Dols. Facial and vocal expressions of emotions. *Annu. Rev. Psychol.*, 54:329–349, 2003. 1