IDIAP RESEARCH REPORT

Multiscale Facial Expression Recognition using Convolutional Neural Networks

Beat Fasel [a]

IDIAP–RR 02-52

November 2002

PUBLISHED IN
Proceedings of ICVGIP 2002, Ahmedabad, India

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[a] IDIAP - Institut Dalle Molle d'Intelligence Artificielle Perceptive Rue du Simplon 4, CP592 - 1920 Martigny, Switzerland Beat.Fasel@idiap.ch

IDIAP Research Report 02-52

Multiscale Facial Expression Recognition using Convolutional Neural Networks

Beat Fasel

November 2002

published in
Proceedings of ICVGIP 2002, Ahmedabad, India

Abstract. Automatic face analysis has to cope with pose and lighting variations. Especially pose variations are difficult to tackle and many face analysis methods require the use of sophisticated normalization procedures. We propose a data-driven face analysis approach that is not only capable of extracting features relevant to a given face analysis task, but is also robust with regard to face location changes and scale variations. This is achieved by deploying convolutional neural networks. We show that the use of multi-scale feature extractors and whole-field feature map summing neurons allow to improve facial expression recognition results, especially with test sets that feature scale, respectively, translation changes.

1 Introduction

Many automatic facial expression analysis approaches presented in the literature need some kind of manual intervention during training, such as the construction of face models or during testing due to necessary initialization procedures, such as the precise localization of facial features, in order to perform reliably. Several data-driven face analysis methods have been described in the literature and comprise among others neural network-based approaches and PCA-based methods. However, numerous data-driven face analysis approaches need accurate face normalization preprocessing stages. In this paper, we propose multi-scale convolutional neural network (CNN)[4] based approaches. CNNs, as well as the similar neocognitrons [2], are bio-inspired hierarchical multi-layered neural network approaches that model to some degree characteristics of the human visual cortex and encompass scale and translation invariant feature detection layers. Convolutional neural networks have been successfully applied for character recognition [5], object detection [5] and more specifically for the task of face recognition [3].

2 Convolutional Neural Networks

Figure 1 shows the architecture of the convolutional neural networks we trained for the task of facial expression recognition. Its layers alternate between convolution layers with feature maps $ConvLay^i_{k,l}$

$$ConvLay^i_{k,l} = g(I^i_{k,l} \otimes W_{k,l} + B_{k,l}) \tag{1}$$

and non-overlapping sub-sampling layers with feature maps $SubsLay^i_{k,l}$

$$SubsLay^i_{k,l} = g(I \downarrow^i_{k,l} w_{k,l} + \Theta b_{k,l}) \tag{2}$$

where $g(x) = \tanh(x)$ is a sigmoidal activation function, B, respectively b the biases, W and w the weights, $I^i_{k,l}$ the i'th input and $I \downarrow^i_{k,l}$ the down-sampled i'th input of the neuron group k of layer l. Θ is a matrix whose elements are all one and $\otimes$ denotes a 2-dimensional convolution. Note that upper case letters represent matrices, while lower case letters denominate scalars. We obtained good results by choosing receptive fields sizes of 5×5 pixels for the groups of neurons in the first feature extraction layer and 15×15 pixels in the third feature extraction layer, respectively 2×2 pixels for the receptive fields of the sub-sampling layers. The learned weights of the convolutional layers allow for problem-at-hand dependent feature extraction, whereas the sub-sampling layers increase the invariance of the object of interest's location dependence. Weight sharing allows to significantly reduce the number of free parameters, which in turn improves the generalization ability [4]. For example, the number of neuron interconnections in the feature extraction layers of the network architecture shown in Figure 1 is 3367308, while the number of weights amounts only to 1902 and the number of neurons to 47787. The number of neuron interconnections in the MLP is however 59826 with the same number of weights and a mere 6 neurons. This shows that even though the feature extraction layers are huge with regard to the number of neuron interconnections, only a few weights need to be trained. Most weights have to be adapted in the MLP connected to the feature layers.

Face images I_{in} at the input of the CNNs were not pose-normalized, but only global lighting changes were addressed by removing the mean value $\overline{I_{in}}$ of the images contained in the training base. In order to increase the learning speed, we normalized also the variances of the input variables by dividing them by their standard deviation σ_{in} of the images of the training set: $I_{norm} = \frac{I_{in} - \overline{I_{in}}}{\sigma_{in}}$. No attempts were taken to reduce image dimensionality by using e.g. holistic PCA as demonstrated in [3]. Instead, we relied on the kernels of the feature extraction layers to perform decorrelation of the input data. Holistically applied PCA without using sophisticated pose normalization procedures would attempt to represent pose information, which is not desired, as there are too many pose variations present in natural face images (due to translation, rotation and scale).

We distinguish three different convolutional layer *(ConvLay)* types: Simple feature extraction convolutional layers *(CoSi)* that contain neuron groups, which operate on a single input feature map

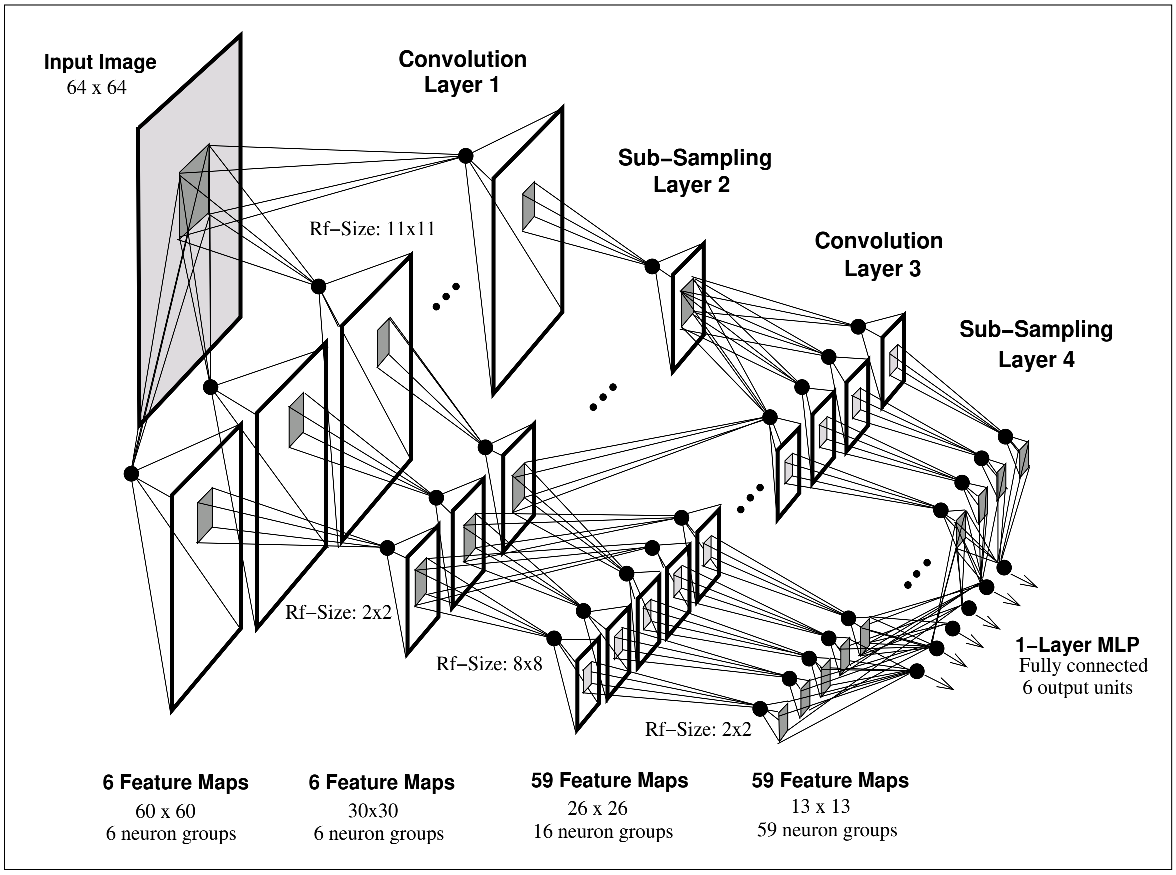

Figure 1: Depicted is the architecture of a 5-layer convolutional neural network (with 2 feature extraction, 2 sub-sampling and one fully connected MLP layer), which we applied for translation invariant facial expression analysis. Note that the larger dots represent groups of identical neurons. The above shown network can be described by the following notation: A6x11x11-B6x2x2-C16x8x8-B-59x2x2-mpl1. For an explanation see also Table 1.

and have as output also a single feature map. Secondly, simple weight-sharing convolutional layers *(CoSiSh)* that contain groups of neurons, which operate like those of simple feature extraction layers, however, share weights amongst themselves. We thus allow here not only for weight sharing over space, but also over different feature extractor groups. Thirdly, complex network layers *(CoCo)* contain neuron groups with different weights per input map, while featuring a single output map. Finally, summer neuron layers *(SuNe)* sum up incoming signals into a singular output value, see also Figure 2.

Using *SuNe*-type layers in combination with convolution-based feature extraction layers *(CoSi)*, we obtain an increased invariance to translation. *SubsLay*-type sub-sampling layers allow for a certain invariance to shearing and local feature deformation, while *CoCo*-type layers integrate small and simple features (5×5) into complex features of a size (15×15). The latter correspond, when taking preceding sub-sampling layers into account, to about the facial area that is of interest for facial action recognition. Finally, *SuNe*-type layers allows for scale-invariant feature extraction, when combined with a preceding multi-scale simple feature extraction layer with receptive fields of different sizes, e.g. 5×5, 7×7 and 9×9.

Training of our CNNs was achieved in a supervised manner by using the standard back-propagation algorithm, adapted for convolutional neural networks. The weight and bias deltas for the feature extraction kernels in the convolutional layers of simple type *(CoSi)* described in equation 1 are

$$\Delta W_{t,k}^{CoSi} = l_R \sum_{i=1}^{F} (I_i^L \otimes D_i^H) + m_R \Delta W_{t-1,k}^{CoSi} \tag{3}$$

$$\Delta B_{t,k}^{CoSi} = l_R \sum_{i=1}^{F} D_i^H + m_R \Delta B_{t-1,k}^{CoSi} \tag{4}$$

while the weight and bias deltas for the sub-sampling layers *(SubsLay)* described in equation 2 are as follows

$$\Delta w_{t,k}^{Subs} = l_R \sum_{i=1}^{F} \sum_{m=1}^{M_i} \sum_{n=1}^{N_i} (I \downarrow_i^L \times D_i^H) + m_R \Delta w_{t-1,k}^{Subs} \tag{5}$$

$$\Delta b_{t,k}^{Subs} = l_R \sum_{i=1}^{F} \sum_{m=1}^{M_i} \sum_{n=1}^{N_i} D_i^H + m_R \Delta b_{t-1,k}^{Subs} \tag{6}$$

where I_i^L is the input image i, $I \downarrow_i^L$ a down-sampled version of the input image i of the lower layer L, D_i^H is the error delta coming from the higher layer H. $\otimes$ denotes a 2-dimensional convolution and $\times$ a component-vise matrix multiplication. F is the number of connected input feature maps of the current neuron group k, M_i and N_i the number or rows, respectively columns of the feature map i. l_R is the learning rate and m_R the moment rate.

The way error deltas D_i for the current layer and neuron group i are computed depends on whether the upper layer is a convolutional layer of type *(CoSi,CoSiSh,CoCo)*

$$D_i = (\sum_{j=1}^{G} infl(D)_{ij}^H \otimes rot180(W_{ij}^H)) \times (1 - \Phi_i \times \Phi_i) \tag{7}$$

a sub-sampling layer *(SubsLay)*

$$D_i = D \uparrow_i^H w_i^H \times (1 - \Phi_i \times \Phi_i) \tag{8}$$

or the MLP

$$D_i = resh_i(d_1^{MLP} \times w_1^{MLP}) \times (1 - \Phi_i \times \Phi_i) \tag{9}$$

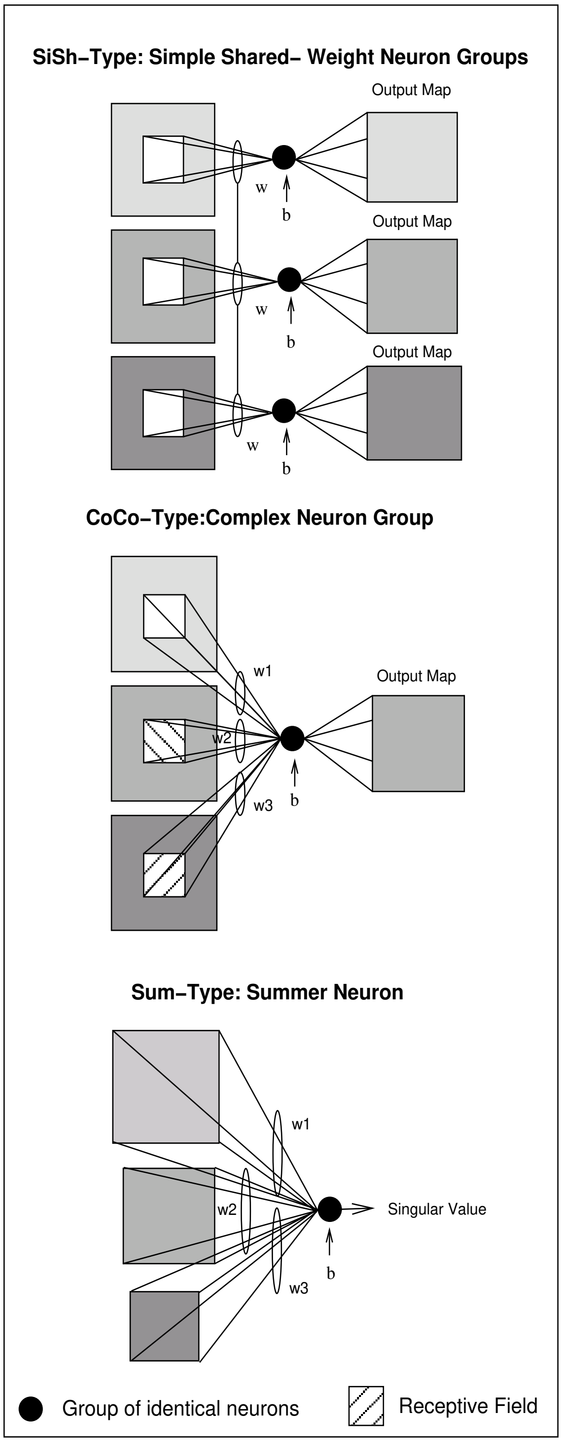

Figure 2: Above is given an illustration of convolutional simple weight-sharing groups *(CoSiSh)* as well as a convolutional non-sharing complex neuron group *(CoCo)* and a non-sharing summer neuron *(SuNe)*.

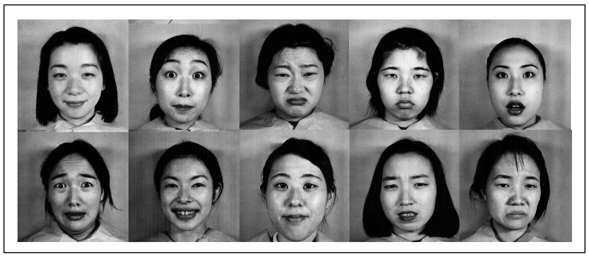

Figure 3: Sample images of the employed JAFFE facial expression database [6]. Note slight variations of the head position, scale and rotation.

Hereby, Φ_i is the output map of the current layer, $infl(D)$ in equation 7 corresponds to a D inflated by zeroes - the matrix D is padded by surrounding zeroes, leading to a total matrix size of $(\#rows_D + 2 * rfSize - 2, \#cols_D + 2 * rfSize - 2)$, where rfSize is the size of the receptive fields or weights (here supposed to be square). Furthermore, $rot180(W_{ij}^H)$ corresponds to the weight matrix ij of the higher layer H and G is the number of neuron groups in the upper layer connected to the current feature map. Finally, $D \uparrow_i^H$ in equation 8 represents the up-sampled delta of the higher layer and $resh_i(d_1^{MLP} \times w_1^{MLP})$ in equation 9 the i'th transformation of the delta vector times the weight vector, of the first layer of the MLP classifier into matrix form. Hereby, the operation $resh_i()$ stands for reshaping of the i'th input vector into matrix form.

3 Experiments and Results

We tested our neural network architectures on two different database sets. Database set 1 consists of the JAFFE facial expression database [6], which contains posed emotional facial expression images of 10 Japanese female subjects (6 different emotion and neutral face displays), see Figure 3. The expressed emotions correspond to the 6 primary or basic emotions postulated by Ekman and Friesen [1] and possess each a distinctive content together with a unique facial expression. They seem to be universal across human ethnicities and cultures and comprise happiness, sadness, fear, disgust, surprise and anger.

The grayscale images originally of size 256×256 pixels were reduced in scale to 64×64 pixels in order to lower the information content that has to be learned by the networks. We used a total of 140 images to train our neural networks and 70 images for testing. The images were labeled into $6 + 1 = 7$ emotion classes. We found that the employed database is too small to allow for a good generalization with regard to the recognition of individual expressions. Dividing the database for example into 7 subjects for training and 3 subjects for testing, yielded average correct recognition results as low as 30%. Therefore, the same subjects were used for both the train and test set. The facial expression images contained in the JAFFE database feature some head pose variations with regard to scale, out-of-plane and in-plane head rotations as well as shifts. In order to demonstrate the capability of CNNs to cope with situations, where head pose variations come into play, we artificially increased the JAFFE database by shifting the images (up, down, left, right), zooming in and out as well as rotating the images both in clockwise and counter-clock wise directions, leading to a second test set of $8 \times 70 = 560$ images that were labeled in the same way as the fist test set of 70 images, i.e. images were tagged according to the prevalent emotional display only. Note that only affine transformations were applied to the test images. See Figure 4 for a few examples of the database set 2.

Table 1 lists the facial expression recognition results obtained on the afore mentioned database sets. Network 1-3 are convolutional neural networks and network 2 and 3 employ multi-scale receptive fields. Network 4 is a 3-layered Multi-layer Perceptron (MLP) for comparison with the CNNs. Network 1 is similar in structure to the one shown in Figure 1. Note that all convolutional networks score better than the MLP with regard to translation invariance. Network 2 and 3 score also better with regard

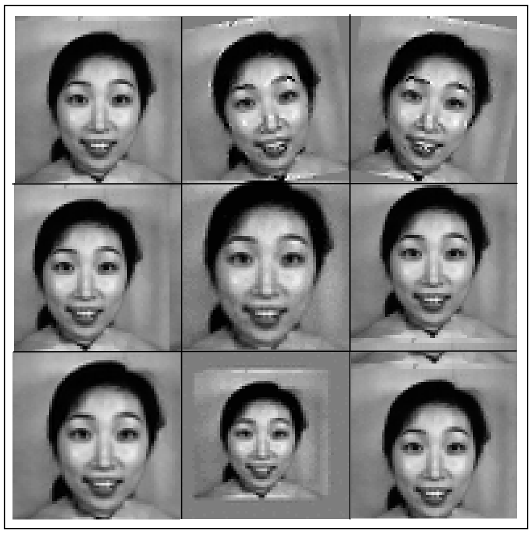

Figure 4: Above are shown some examples of database set 2 that was artificially extended by introducing translational shifts (up, down, left, right), zoom in and out, as well as clockwise and counter-clockwise rotations.

CNN Network Architectures layer type - rf. sizes - classifier type	**Correct Recognition** **Set 1** (140/70)	**Correct Recognition** **Set 2** (140/560)
(1) A12*5x5-B12*2x2-C12*15x15-... B12*2x2-mlp2	80.0%	T38.2% R45.0% S47.1%
(2) A12*5x5-12*7x7-12*9x9-B36*2x2-... S4*15x15-4*15x15-4*15x15-B12*2x2-mlp2	72.9%	T36.8% R42.1% S50.0%
(3) A1*1x1-1*5x5-1*9x9-... S36*5x5-B36*3x2-C18*15x15-D18-mlp2	82.9%	T49.6% R54.3% S59.3%
(4) MLP3 (100-20-6)	88.6%	T25.0% R38.6% S36.4%

Table 1: Facial expression recognition results obtained by using three different convolutional neural network architectures as well as an MLP. Following notation was employed for the network architectures: **A** stands for simple feature extraction layers, **S** for simple shared feature extraction layers, **B** for sub-sampling layers, **C** for complex layers and **D** for summer neuron layers. Notation used to describe recognition results: **T** stands for translated, **R** for in-plane rotated and **S** for scaled face images of the test set 2.

to scale invariance. Furthermore, it is also interesting to note that the CNNs improved invariance to in-plane rotation. This is probably due to the sub-sampling layers. All CNNs gave slightly lower recognition results on test set 1, when compared to the MLP. Test set 1 features only few pose variations and thus the full connectivity of the MLP might be an advantage.

Unfortunately, we cannot compare our facial expression recognition results with the ones Lyons and Akamatsu [6] obtained on the same database, as they computed facial expression similarities using semantic values stemming from human ratings, resulting in a mixture of facial expressions per analyzed face, while we used one category per facial expression.

4 Conclusions

In this paper we focused on adjusting the architecture of convolutional neural networks in order to allow for an increased invariance with regard to translation and scale changes without relying on huge databases for learning affine transformations of human faces. We were able to improve results with regard to scale and translation changes by using multi-scale simple feature extractor layers in combination with weight-sharing feature extraction layers, respectively, translation independence was increased by using summer layers in combination with convolutional features extractors. The employed CNN architectures recognized facial expressions also in the presence of in-plane pose variations without requiring extensive pose normalization or feature tracking initialization procedures. The only assumptions we made was, that the input image is coarsely centered around a single face to be analyzed. Also, no face segmentation is required with our approach. Further work has to be done in order to improve network inherent in- and out-of-plane rotation invariance.

5 Acknowledgments

Thanks go to the SNSF (Swiss National Science Foundation) for funding this project within the framework of IM(2) under project number 21-54000.98.

References

[1] P. Ekman and W. V. Friesen. Constants Across Cultures in the Face and Emotion. *Journal of Personality and Social Psychology*, 17(2):124–129, 1971.

[2] Fukushima K. Neocognitron: A Self-Organizing Neural Network for a Mechanism of Pattern Recognition Unaffected by Sift in Position. *Biol Cybern*, 36:193–202, 1980.

[3] Steve Lawrence, C. Lee Giles, A.C. Tsoi, and A.D. Back. Face recognition: A convolutional neural network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.

[4] Y. LeCun and Y. Bengio. Convolutional networks for images, speech, and time-series. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, 1995.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[6] Lyons M., Akamatsu S., Kamachi M., and Gyoba J. Coding Facial Expressions with Gabor Wavelets. In *Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 200–205, April 1998.

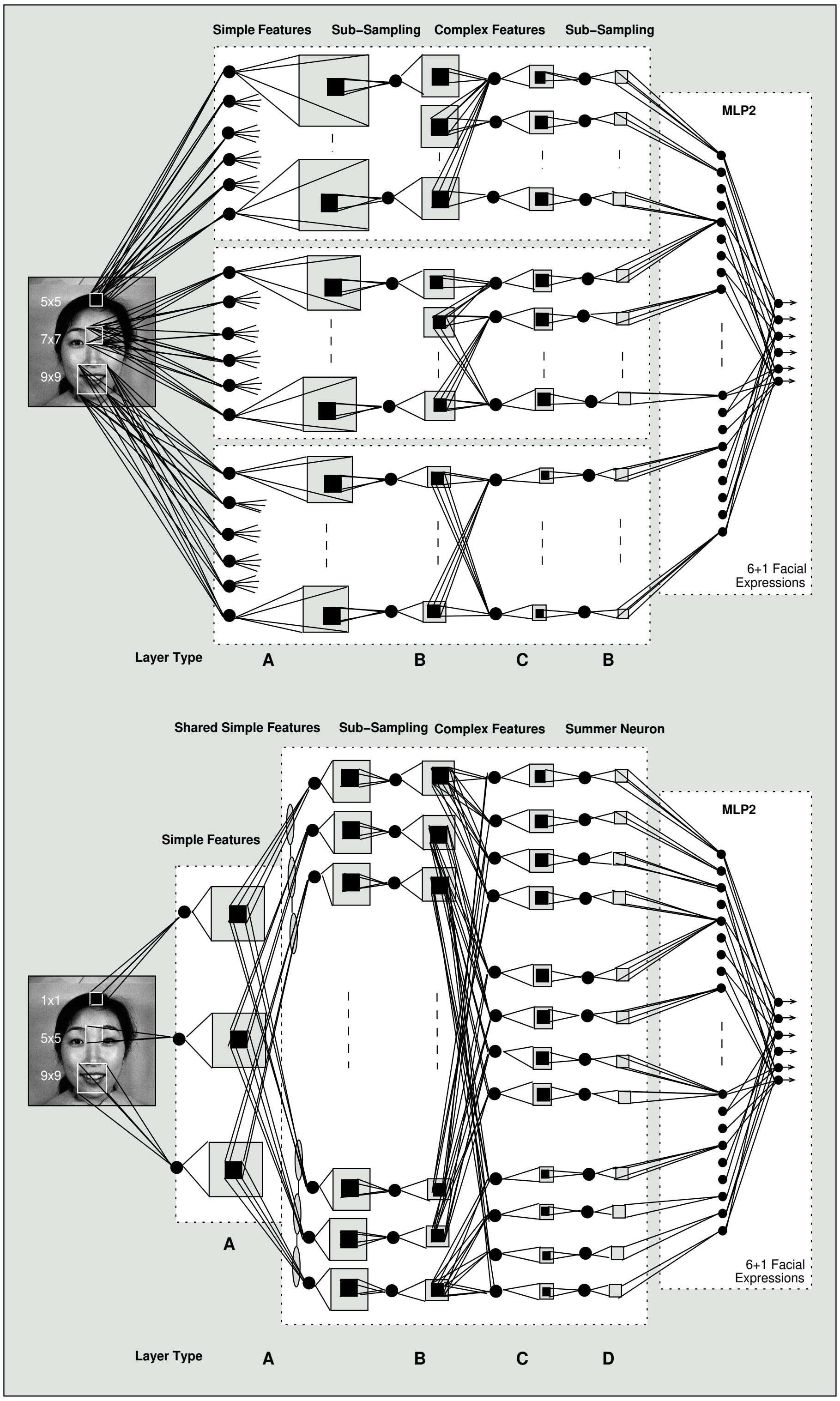

Figure 5: Shown above are two convolutional neural networks that use multi-scale receptive fields. The network on top corresponds to the network 2 in Table 1, while the network on the bottom corresponds to the network 3. The three highlighted sub-regions of the network 2 operate at different resolutions and there is no connection in-between them. Integration of the signals only occurs in the MLP classifier. Note also that the complex neurons in the third network layer integrate information stemming from several input feature maps into single output maps. Furthermore, network 3, shown on the bottom, extracts the same features at different scales by using shared convolutional simple features extractors (CoSiSh) and also employs summer neurons in the last feature extraction layer in order to improve translation independence.