MicroExpNet: An Extremely Small and Fast Model For Expression Recognition From Frontal Face Images

İlke Çuğu, Eren Şener, Emre Akbaş
Department of Computer Engineering, Middle East Technical University
06800 Ankara, Turkey
{cugu.ilke, sener.eren}@metu.edu.tr, emre@ceng.metu.edu.tr

Abstract

This paper is aimed at creating extremely small and fast convolutional neural networks (CNN) for the problem of facial expression recognition (FER) from frontal face images. To this end, we employed the popular knowledge distillation (KD) method and identified two major shortcomings with its use: 1) a fine-grained grid search is needed for tuning the temperature hyperparameter and 2) to find the optimal size-accuracy balance, one needs to search for the final network size (i.e. the compression rate). On the other hand, KD is proved to be useful for model compression for the FER problem, and we discovered that its effects gets more and more significant with the decreasing model size. In addition, during the search for a network architecture, we hypothesized that translation invariance achieved using max-pooling layers would not be useful for the FER problem as the expressions are sensitive to small, pixel-wise changes around the eye and the mouth. However, we have found an intriguing improvement on generalization when max-pooling is used. Experiments are made on two widely-used FER datasets, CK+ and Oulu-CASIA. Our smallest model (MicroExpNet), obtained using knowledge distillation, is less than 1MB in size and works at 1851 frames per second on an Intel i7 CPU. Despite being less accurate than the state-of-the-art, MicroExpNet still provides significant insights on the creation of a micro architecture for the FER problem.

1. Introduction

Expression recognition from frontal face images is an important aspect of human-computer interaction and has many potential applications, especially in mobile devices. Face detection models have long been deployed in mobile devices, and relatively recently, face recognition models are also being used, e.g. for face based authentication. Arguably, one of the next steps is the mobile deployment of

facial expression recognition models. Therefore, creating small and fast models is an important goal. In order to have an idea about the current situation, we looked at the size and runtime speeds of two representative, currently state-of-the-art models, namely PPDN [40] and FN2EN [5]. In terms of the number of total parameters in the network, both models are in the order of millions (PPDN has 6M and FN2EN has 11M). In terms of speed, both models run at $9 - 11$ms per image on a GTX 1050 GPU, however, on an Intel i7 CPU, while PPDN takes 57.18 ms, FN2EN takes 96.08 ms (further details in Tables 6 and 7).

The central question that motivated the present work was how much we could push the size and speed limits so that we end up with a compact expression recognition model that still works reasonably well. To this end, we focused only on frontal face images and first explored training a large model on two widely used benchmark FER datasets, CK+ [24] and Oulu-Casia [39], by simply using the Inception_v3 [36] model. Then, using the "knowledge distillation" (KD) method [13], we were able to create a family of small and fast models. In the KD method, there is a large, cumbersome model called the *teacher* (Inception_v3 in our case) and a relatively much smaller model called the *student*. The student is trained to "mimic" the softmax values of the teacher via a *temperature* hyperparameter (see Eq. 2). We have experimented on four student networks with different sizes, and name the smallest one as **MicroExpNet** which is 100x smaller in size and has 335x fewer parameters compared to its teacher.

We found two major shortcomings of the KD method. First, the temperature hyperparameter does not seem to have any meaningful relation with the accuracy of the student model. We found that the accuracy fluctuates between low and high values as temperature is swept across a wide range. In order to find a high-accuracy temperature, one needs to do a fine-grained grid search. Second, in the KD method, the final student model size (i.e. the compression rate) is given as input. Therefore, to find the optimal size-accuracy balance, one needs to search for the size, too.
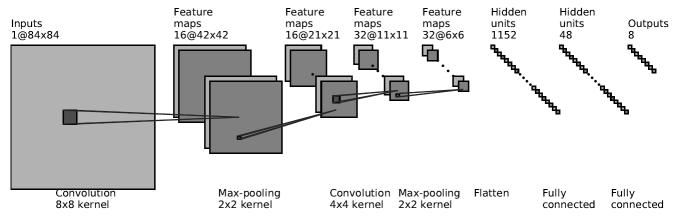
Figure 1. The architecture of MicroExpNet, our smallest (65K parameters, 0.88 MB in size) and fastest (1851 FPS on Intel i7 CPU) model.

We also hypothesized that invariance to translation achieved using max-pooling layers would not be useful for the FER problem as the expressions are sensitive to small, pixel-wise changes around the eye and the mouth. In our experiments, we found that this is **not** the case. On the contrary, the best results are obtained when we added a max pooling layer after each convolutional layer. However, we also found a strange phenomena when we separated the validation and training sets via random splits. In the literature, FER datasets are separated into **subject-independent** splits for validation & training. In this setting, the trained model is tested with the photographs of the subjects it did not see while training. However, for random splits, model may or may not see a test subject during training. Although each photograph is numerically different, random splits transform the FER problem to a memorization problem. We validate this proposition with our empirical analysis in Table 2. For random split, max-pooling degrades the performance whereas for subject-independent split it improves the performance.

Overview. Our findings raise three important questions:

1. **Is information loss essential for generalization?** We show that, considering it as a tool for information loss where numerically dominant values suppress the others, max-pooling improves the classification performance. However, when the problem is transformed into a memorization challenge via a simple change in the selected method for the separation of validation & training sets for the same dataset, having no max-pooling layer yields the best results.

2. **Is a small model more open to supervision than a cumbersome model?** We show that the effect of KD (compared to training from scratch) increases as the network size gets smaller. To the best of our knowledge, this kind of analysis has not been done before.

Whether this effect is specific to the FER problem is yet to be seen (left as future work).

3. **Why is the change in *temperature* hyperparameter of KD results in severe fluctuations of classification performance in a stochastic way?** We show that, whether a random or a subject-independent split is used, the classification accuracy fluctuates between low and high values as temperature is swept across a wide range.

In order to support our propositions, we provide standard classification performance comparison for CK+ and Oulu-CASIA datasets; parameter count comparison; runtime speed comparison; max pooling vs. no pooling analysis; extensive temperature and size combinations to provide answers to "what if" questions.

2. Related work

2.1. Facial expression recognition (FER)

We categorize the previous work as image (or frame) based and sequence based. While image-based method analyse individual images independently, sequence-based methods exploit the spatio-temporal information between frames.

Image based. There are three groups of work. Models that use 1) hand-crafted features (HCFs), 2) deep representations, and 3) both. Our work falls into the second group.

We do not focus on HCF models [3, 8, 33, 41] here because they are obsolete (with the emergence of deep models) and in general, they do not achieve competitive results. These works typically extract Local Binary Patterns (LBP) [27], SIFT [33] or Gabor features and use SVM [4] or AdaBoost [9] on top of these features, as the classifier.

Deep representations learned from face images are the main ingredients of [23, 25, 40, 5, 16]. Liu *et al.* [23]

proposed a loopy boosted deep belief network framework for feature learning, then used them in an AdaBoost classifier. Mollahosseini *et al*. [25] introduced an inception network for FER. Their model is much larger compared to ours considering the two large fully connected layer at the end of their network. Zhao *et al*. [40] proposed a peak-piloted GoogLeNet [35] model which uses both peak and non-peak expression images during training. Training peak and non-peak images in pairs naturally requires their proposed back-propagation algorithm which adds complexity to implementation compared to our work. FN2EN [5] employs a multi staged model production for FER. First, they train convolutional layers by mimicking [2] a pre-trained FaceNet [28]. Then, they append a fc layer to the model for retraining. Recently, Kim *et al*. [16] introduced a deep generative contrastive model for FER. They combined encoder-decoder networks and CNNs into a unified network that simultaneously learns to generate, compare, and classify samples on a dataset.

Finally, [19] form a hybrid approach. They train CNNs with both the original input images and 3D mappings of local binary patterns [27], then finalize via fine-tuning.

Sequence-based. We can categorize sequence based facial expression classifiers in the same three groups as in the case of image-based classifiers.

We do not focus on HCF based sequence models [12, 10, 31, 32] for the same reasons with the image-based case.

Deep representations are the core ingredient of [22, 6]. Liu *et al*. [22] proposed a manifold modeling of videos based on representations gathered via learned spatio-temporal filters. Kahou *et al*. [6] fused CNNs with recurrent neural networks (RNNs). CNN is used on static images to gather high-level representations which are then used by the RNN training.

Jung *et al*. [15] proposed a hybrid approach via two deep models. First, a 3D-CNN to extract the temporal appearance features from image sequences. Second, a fully connected model which captures geometrical information about the motion of the facial landmark points.

2.2. Model size reduction

FitNets. Romero *et al*. [29] built their FitNets using the "knowledge distillation" method to produce deep and thin student networks with comparable or better performance compared to the teacher. They built student networks that are thinner but deeper than their teacher by training some layers of the student beforehand with the teacher's supervision for better initialization. They trained the whole student network using knowledge distillation to finalize their model. They applied their model to object recognition, handwriting recognition and face recognition where the FitNet failed to outperform the state-of-the-art solutions, but achieved su-

perior performance against its teacher. To the best of our knowledge, we are the first to apply knowledge distillation to the facial expression recognition (FER) problem. In addition, we choose a model that is much shallower than the teacher and avoid any pre-training of the student to prevent increasing the complexity of the overall training procedure. Another important point is that, Romero *et al*. did not give much information on the selection of the temperature parameter, in which we do a systematic analysis.

SqueezeNets. Iandola *et al*. [14] proposed a CNN with no fully connected layers to reduce the model size, and preserved the classification performance via their fire modules. Like FitNets, they also did not test their model on FER.

3. Methodology

3.1. Knowledge distillation

Knowledge distillation was introduced by Hinton et al. [13] in 2015. The main idea is to have a cumbersome network called *the teacher* to supervise the training of a much smaller network called *the student* via soft outputs. The algorithm is as follows: first, a large teacher network is trained for the task using an empirical loss calculated with respect to one-hot vector of true labels. Then, a much smaller student network is trained using both one-hot vectors of the true labels and the softmax outputs (Eq. 2) of the teacher network. The aim is to increase the information about the target classes by introducing uncertainty into probability distributions. Since these distributions contain similarity information on different classes, Hinton *et al*. further used this similarity information coming from the teacher to correctly classify a target class intentionally removed from the training set of the student. Additionally, in order to prevent the teacher's strong predictions to dominate the similarity information, softmax logits(z_i) of the teacher are softened using a hyperparameter called temperature denoted as T in Eq. 1.

Formally, let p_t be the softened output of the teacher's softmax, z_i be the logits of the teacher, p_s be the hard and p'_s be the soft output of the student's softmax, v_i be the logits of the student, λ be the weight of distillation, y be the ground truth labels, N be the batch size and function $\mathcal{H}$ refers to the cross-entropy. Then:

$$p_t = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}}, \quad p'_s = \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}, \quad p_s = \frac{e^{v_i}}{\sum_j e^{v_j}} \quad (1)$$

and the loss becomes

$$\mathcal{L} = \lambda\left(\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}(p_t, p'_s)\right) + (1-\lambda)\left(\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}(y, p_s)\right). \quad (2)$$

3.2. Network architectures

In this work, we use two convolutional networks, namely the teacher and the student. The teacher is deep and large whereas the student is shallow and small. There are several versions of the student network having different number of parameters. We call our smallest network as MicroExpNet.

Teacher network. We use the Inception_v3 [36] network as the teacher for its proven record of success on classification tasks [30].

Student network. Our student network has a very simple architecture: two convolutional layers and a fully connected layer with rectified linear unit (ReLU) [26] as the activation function and a final fully connected layer as a bridge to the softmax. Due to our detailed examination about using max-pooling vs no-pooling, described in Section 4.1, we decided to have pooling layers after each convolutional layer. Next, we squeezed the student network by reducing the size of its last fully connected layer to have a fairly compact CNN, and we used the knowledge distillation method [13] to keep the high performance. We created four student models, from largest to smallest: M, S, XS and XXS, to determine the most suitable size-performance balance for our final proposal. Table 1 presents the architectures of these four models. We compare their classification performances in sections 4.2 and 4.3, speeds in Section 4.6, and memory requirements in Section 4.5.

3.3. Implementation

CK+ & Oulu-CASIA. For each image in CK+, we apply the Viola Jones [37] face detector, and for each image in Oulu-CASIA we use the already cropped versions. All images are converted to grayscale. Then, in order to augment the data, we extract 8 crops (4 from each corner and 4 from each side) from an image with dimensions of 84x84 for students and 256x256 for the teacher. There is no difference on hyperparameter selections for the trainings on CK+ and Oulu-CASIA. As done in previous work, we report the average 10-fold cross validation (CV) performance. For both the teacher and students, trainings are finalized after 3000 epochs.

Teacher Network. We employ a Inception_v3 trained on the 1.28 million training images of ImageNet [30], and fine-tune it on FER datasets. The base learning rate is set as 10^{-4} and remained constant through iterations, mini-batch size is 64, and the optimization algorithm is Adam [17].

Vanilla & Student Networks. We have the same hyper-parameters across all of the different model sizes for both vanilla and student trainings. "Vanilla" training means that

Model	# of Parameters	Architecture
M	900920	conv1 - kernel: 8x8 , stride: 2 pool1 - kernel: 2x2 , stride: 2 conv2 - kernel: 4x4 , stride: 2 pool2 - kernel: 2x2 , stride: 2 fc1 - in: 1152 fc2 - in: 768 softmax - 8
S	232184	conv1 - kernel: 8x8 , stride: 2 pool1 - kernel: 2x2 , stride: 2 conv2 - kernel: 4x4 , stride: 2 pool2 - kernel: 2x2 , stride: 2 fc1 - in: 1152 fc2 - in: 192 softmax - 8
XS	120728	conv1 - kernel: 8x8 , stride: 2 pool1 - kernel: 2x2 , stride: 2 conv2 - kernel: 4x4 , stride: 2 pool2 - kernel: 2x2 , stride: 2 fc1 - in: 1152 fc2 - in: 96 softmax - 8
XXS	65000	conv1 - kernel: 8x8 , stride: 2 pool1 - kernel: 2x2 , stride: 2 conv2 - kernel: 4x4 , stride: 2 pool2 - kernel: 2x2 , stride: 2 fc1 - in: 1152 fc2 - in: 48 softmax - 8

Table 1. Architectures of the student networks from largest to smallest.

the network is trained from scratch without any teacher guidance. Weights and biases are initialized using Xavier initialization [11]. Network architectures are implemented via Tensorflow [1]. Adam [17] optimizer is adopted as the optimization algorithm. The base learning rate is set as 10^{-4}, dropout [34] is 0.5, mini-batch is 64 and the weight of the distillation λ is 0.5 (see Section 3.1) for all student models. Selected model sizes are 900K, 232K, 121K and 65K parameters respectively, which are produced by decreasing the size of the $fc1$ layer (see Table 1). Training operations are finalized after 3000 epochs for all models and the XXS student model is denoted as **MicroExpNet**. Empirical results are given in Table 4 and Table 5, note that for student networks we only put the best performers across different temperatures (selected using cross-validation). Furthermore, student models are used in temperature selection tests (for detailed explanation see Section 4.4). The results we report for these models are obtained by averaging the 10-fold cross validation performances.

Model	CK+	Oulu-CASIA	Model	CK+	Oulu-CASIA	10-fold split
Candidate_v_M	97.93%	97.68%	**Candidate_v_{XS}**	**93.41%**	**88.73%**	Random
Candidate_$p1_M$	**97.99%**	**97.79%**	Candidate_$p1_{XS}$	91.85%	80.16%	
Candidate_$p2_M$	97.41%	96.64%	Candidate_$p2_{XS}$	86.84%	77.88%	
Candidate_$p12_M$	97.39%	97.47%	Candidate_$p12_{XS}$	88.07%	77.04%	
Candidate_v_S	96.65%	92.95%	**Candidate_v_{XXS}**	**81.91%**	**73.64%**	
Candidate_$p1_S$	**96.73%**	**93.22%**	Candidate_$p1_{XXS}$	69.05%	52.99%	
Candidate_$p2_S$	94.09%	88.61%	Candidate_$p2_{XXS}$	77.74%	66.84%	
Candidate_$p12_S$	94.39%	88.72%	Candidate_$p12_{XXS}$	78.52%	61.71%	
Candidate_v_M	81.23%	60.87%	Candidate_v_{XS}	77.14%	53.73%	Subject-independent
Candidate_$p1_M$	**81.57%**	**62.46%**	Candidate_$p1_{XS}$	77.14%	53.41%	
Candidate_$p2_M$	78.77%	60.21%	Candidate_$p2_{XS}$	78.42%	57.51%	
Candidate_$p12_M$	79.95%	60.53%	**Candidate_$p12_{XS}$**	**79.78%**	**57.54%**	
Candidate_v_S	79.73%	58.18%	Candidate_v_{XXS}	71.36%	44.33%	
Candidate_$p1_S$	**81.25%**	**59.49%**	Candidate_$p1_{XXS}$	67.04%	34.04%	
Candidate_$p2_S$	78.75%	57.37%	Candidate_$p2_{XXS}$	76.91%	54.62%	
Candidate_$p12_S$	79.71%	57.25%	**Candidate_$p12_{XXS}$**	**78.44%**	**55.03%**	

Table 2. The effect of max-pooling. Classification performances of the candidate models for 1000 epochs of training. $p1$ indicates that there is only one max pooling layer after conv1, $p2$ indicates that there is only one max pooling layer after conv2, $p12$ indicates that each conv layer is followed by a max pooling layer, and v indicates that there is no pooling layer at all. The smaller the network, the more max-pooling degrades the performance for **random split** whereas the opposite holds for **subject-independent split**.

4. Experiments

4.1. Max. Pooling vs. No Pooling Analysis

Facial expressions are located mostly on eyes and mouth [7], and they form only a small fraction of a frontal face image. The idea is to capture these subtle indicators of an emotion by preserving the pixel information across layers. Therefore, our starting point was a CNN with no pooling layers. However, in order to validate our intuition, we build three variations containing max pooling layers for each student. All pooling layers have 2x2 filters with stride 2. All hyperparameters mentioned at Section 3.3 apply to these variations as well. We call them candidate expression networks. These candidates are explained in Table 2.

From the results in Table 2, we draw the following conclusions. When models are large enough, the provided capacity for learning dominates pooling effects. For instance, for the size M, classification performances of candidates are very close to each other. For size S, poolings in later layers drops the performance but early pooling is still the most profitable. After this point (XS and XXS), we begin to see an interesting difference between the results of random and subject-independent split experiments. For random split, we see the advantage of not having any pooling layers with significant gains in performance. Since the trained candidates see the same subjects in both training and test (for ≈ 80% of the subjects), although the images are numerically different, we think that the resemblance transforms the FER problem to a memorization challenge. Hence, the information loss caused by the pooling layers drops the performance. On the contrary, for subject-independent split,

test subjects are not seen during the training and we see the advantage of having pooling layers. It is also interesting to observe that the second pooling layer seems to be a much critical point of improvement than the first pooling layer. Nevertheless, combining these observations with our intention to reduce the model size, we decided to employ the architecture with two pooling layers as the foundation of our student networks.

Note that adding a pooling layer drops the number of parameters, thus prevents a proper performance comparison. Therefore, we did two modifications to increase the model size, in order to make it a fair comparison. First, when we add a pooling layer after the first convolutional layer, we decrease the stride of the first conv layer from 4 to 2. This directly recovers all parameters that has been lost. Second, when we add a pooling layers after the second convolutional layer, we increase the number of outputs of the first fully connected layer by 3-fold. This results in having slightly less parameters than the original one (CandidateExpNet$_v$).

4.2. The CK+ dataset

CK+ is a widely used benchmark database for facial expression recognition. This database is composed of 327 image sequences with eight emotion labels: anger, contempt, disgust, fear, happiness, sadness, surprise and neutral. There are 123 subjects. As done in previous work, we extract last three and the first frames of each expression sequence when images are labeled. When unlabeled, we only extracted first frames as neutral. The total number of images is 1574 (see at Table 3), which is split into 10 folds. We report results for 3000 epochs for training throughout

	Anger	Contempt	Disgust	Fear	Happy	Sad	Surprise	Neutral	All
CK+	135	54	177	75	207	84	249	593	1574
Oulu-CASIA	240	-	240	240	240	240	240	-	1440

Table 3. The number of images per expression classes in Ck+ and Oulu-CASIA.

this section.

Training in Isolation. We evaluate the pre-trained Inception_v3 via fine-tuning on CK+. Then, we train four models, namely VanillaExpNet$_M$, VanillaExpNet$_S$, VanillaExpNet$_{XS}$, and VanillaExpNet$_{XXS}$, from scratch. At this stage, we did not employ knowledge distillation. For all models, we used 3000 epochs for training, and the classification performances are shown in Table 4. Although Zhao *et al*. [40] seem to achieve better performance than Inception_v3 (in Table 4), they use only 6 emotion categories, whereas we use all of the 8 emotion categories. In the light of these results, we choose Inception_v3 as the teacher for the knowledge distillation stage.

Training with Supervision. We evaluate four students, namely StudentExpNet$_M$, StudentExpNet$_S$, StudentExpNet$_{XS}$, and StudentExpNet$_{XXS}$, via knowledge distillation on CK+. At this stage, we use the teacher's supervision to improve the learning. As explained in Section 3.3, we need to tune the *temperature* for each student since it is regarded as correlated with model size. Therefore, we conducted an extensive experiment on classification performances for a wide range of *temperatures*. The results are reported in Figure 2. According to these results, fluctuations between performances are increased while models are getting smaller. Consequently, it suggests that large networks are more tolerant to the changes in the temperature. This observation also holds for random split case as shown in Figure 3.

Best performers, regarding their average classification performances for 10-fold cross validation, across different temperatures are then used for performance comparison in Table 4. Our findings (see Fig. 4) show that knowledge distillation can be used to gain back some of the performance lost by decreasing the model size.

4.3. The Oulu-CASIA dataset

Oulu-CASIA has 480 image sequences taken under *dark, strong, weak* illumination conditions. In this experiment, as also done in previous work, we used only videos with *strong* condition captured by a VIS camera. In total, there are 80 subjects and six expressions: anger, disgust, fear, happiness, sadness, and surprise. Similar to CK+, the first frame is always neutral while the last frame has the peak expression. All studies we have encountered on Oulu-CASIA database

Method	Accuracy	# of Classes
CSPL [41]	89.9%	Six Emotions
3DCNN-DAP [21]	92.4%	
Inception [25]	93.2%	
AdaGabor [3]	93.3%	
STM-ExpLet [22]	94.2%	
LOMo [32]	95.1%	
LBPSVM [8]	95.1%	
BDBN [23]	96.7%	
DTAGN [15]	97.3%	
FN2EN [5]	98.6%	
DCN [38]	98.9%	
PPDN [40]	99.3%	
AU-Aware [20]	92.1%	Eight Emotions
FN2EN [5]	96.8%	
GCNet [16]	97.3%	
TeacherExpNet	**97.6%**	
VanillaExpNet$_M$	78.8%	
VanillaExpNet$_S$	78.6%	
VanillaExpNet$_{XS}$	77.2%	
VanillaExpNet$_{XXS}$	75.3%	
StudentExpNet$_M$	83.1%	
StudentExpNet$_S$	83.6%	
StudentExpNet$_{XS}$	83.7%	
MicroExpNet	**84.8%**	

Table 4. Average classification performances of different methods on the CK+ dataset using subject-independent splits.

use only the last three frames of the sequences, so we also use the same frames. Therefore, the total number of images is 1440. As in the earlier studies, a 10 fold CV is performed, and the split is subject independent. We report results for 3000 epochs for training throughout this section.

Training in isolation. The same approach taken for CK+ is employed for Oulu-CASIA. The classification performances are shown in Table 5. According to the table, Inception_v3 performs on par with the state-of-the-art solutions whereas our vanilla models failed to achieve competitive results.

Training with supervision. The same explanations on students for CK+ also apply to Oulu-CASIA experiments.

Method	Accuracy
HOG 3D [18]	70.63%
AdaLBP [39]	73.54%
STM-ExpLet [22]	74.59%
Atlases [12]	75.52%
DTAGN [15]	81.46%
LOMo [32]	82.10%
PPDN [40]	84.59%
GCNet [16]	86.39%
FN2EN [5]	87.71%
TeacherExpNet	**85.83%**
VanillaExpNet$_M$	56.81%
VanillaExpNet$_S$	55.53%
VanillaExpNet$_{XS}$	54.67%
VanillaExpNet$_{XXS}$	56.71%
StudentExpNet$_M$	63.81%
StudentExpNet$_S$	62.01%
StudentExpNet$_{XS}$	61.76%
MicroExpNet	**62.69%**

Table 5. Average classification performances of different methods on the Oulu-CASIA dataset.

The results are reported in Figure 5 from which, we can observe a similar fluctuating behavior as seen in the CK+ experiments. Once again, we can see that large networks are more tolerant to the changes in the temperature than the smaller ones. In addition, as in CK+ experiments, this observation also holds for random split case as shown in Figure 6.

Best performers across different temperatures are then used for performance comparison in Table 5. We can still observe that the student models perform better than vanilla models (which are trained from scratch without any teacher supervision) for facial expression recognition.

4.4. Temperature analysis

Temperature is a tool to enforce the uncertainty of the teacher network to emerge. This uncertainty may be used as similarity information between different classes to enhance the training. However, there is no formulation for selecting the most effective temperature; it is set empirically. Hence, we did a grid search for temperatures of [2, 4, 8, 16, 20, 32, 64] with 10-fold cross validation across all of our student networks using both CK+ (see Figure 2) and Oulu-CASIA (see Figure 5) datasets using a subject-independent train & validation split. Moreover, we did a grid search for a random train & validation split as well (see Figures 3 and 6).

According to the results, smaller models are more prone to temperature changes in general, and performances for a given temperature seem rather stochastic. However,
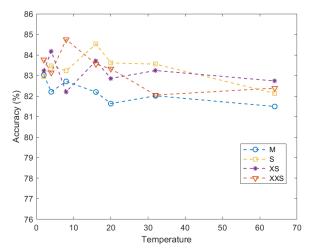

Figure 2. Classification performances of the student networks across different temperatures on the CK+ dataset using **subject-independent splits**.
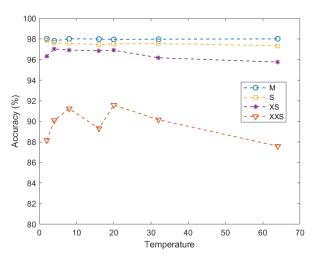

Figure 3. Classification performances of the student networks across different temperatures on the CK+ dataset using **random splits**.

large models show different characteristics for random split case and subject-independent split case. When subject-independent split is used, we observe fluctuations in performance for all models regardless of their size. Whereas for random split case, large models have relatively stable performances. Nevertheless, when calibrated adequately, KD improves the overall FER performance as it can be seen at figures 4 and 7 for subject-independent split case.

4.5. Model size analysis

One of the most important benefits of a small neural network is its modest need for memory space. Table 6 shows the comparison of the model sizes in megabytes. Our ul-
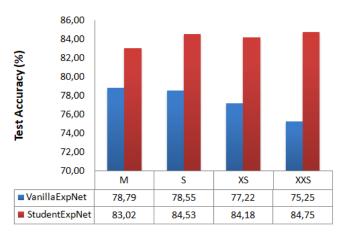
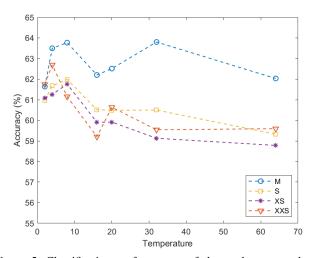Figure 4. The effect of supervision on CK+ for 3000 epochs of training.

	M	S	XS	XXS
■ VanillaExpNet	78,79	78,55	77,22	75,25
■ StudentExpNet	83,02	84,53	84,18	84,75

Figure 6. Classification performances of the student networks across different temperatures on the Oulu-CASIA dataset using **random splits**.

Figure 5. Classification performances of the student networks across different temperatures on the Oulu-CASIA dataset using **subject-independent splits**.

	M	S	XS	XXS
■ VanillaExpNet	56,81	55,53	54,67	56,70
■ StudentExpNet	63,80	61,99	61,76	62,69

Figure 7. The effect of supervision on Oulu-CASIA for 3000 epochs of training.

timate facial expression recognition model MicroExpNet takes less than 1 MB to store which is 100x smaller than our teacher network (Inception_v3). In addition, MicroExpNet has 335x fewer parameters than the teacher.

4.6. Model speed analysis

Another important benefit of a small neural network is its speed. In order to measure the speed, we ran each model for 1000 times with single input image and measure the average run time. Table 7 shows the comparison of the elapsed times to process one image in milliseconds. According to the table, MicroExpNet achieves the best performance by classifying the facial expression in an image in less than 1 ms on an Intel i7-7700HQ CPU. Also, it can be seen that all of the students achieved speeds that are well above the requirements of real-time processing. Ultimately,
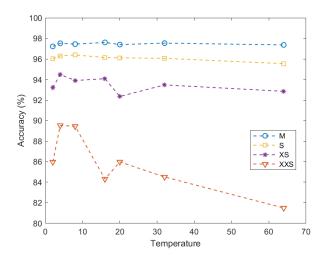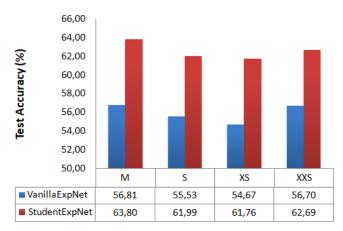
Model	# of Parameters	Size
TeacherExpNet	21.8M	88.13 MB
FN2EN [5]	11M	42.42 MB
PPDN [40]	6M	23.93 MB
StudentExpNet$_M$	900K	10.88 MB
StudentExpNet$_S$	232K	2.91 MB
StudentExpNet$_{XS}$	121K	1.52 MB
MicroExpNet	**65K**	**0.88 MB**

Table 6. Memory requirements of different FER models.

our final facial expression recognition model, when compared to our teacher network Inception_v3, MicroExpNet is 234x faster on Intel i7-7700HQ CPU, and 85x faster on

Model	i7-7700HQ	GTX1050	Tesla K40
TeacherExpNet	124.22 ms	83.25 ms	-
FN2EN [5]	96.08 ms	23.81 ms	13.09 ms
PPDN [40]	57.18 ms	9.12 ms	13.11 ms
StudentExpNet$_M$	0.89 ms	1.13 ms	1.74 ms
StudentExpNet$_S$	0.78 ms	1.08 ms	1.69 ms
StudentExpNet$_{XS}$	0.63 ms	0.97 ms	1.63 ms
MicroExpNet	**0.53 ms**	**0.97 ms**	**1.52 ms**

Table 7. Average per-image running times of different FER models.

GTX1050 GPU.

5. Conclusion

We presented an extensive analysis on the creation of a micro architecture, called the MicroExpNet, for facial expression recognition (FER) from frontal face images.

From our experimental work, we have drawn the following conclusions. (1) Translation invariance achieved via max-pooling and knowledge distillation method improves the FER performance especially when the network is small. (2) We showed that a simple change in the approach taken for the separation of train & validation sets results in drastic changes in the problem definition, and thus in the performance observations (3) "Knowledge distillation"'s effect gets more prominent as the network size decreases. If this effect is generalizable to other problems/datasets is yet to be seen in future work. (4) The temperature hyperparameter (in knowledge distillation) should be tuned carefully for optimal performance. Especially when the network is small, the final performance fluctuates with temperature.

Availability

Our codes are available at GitHub.

Acknowledgement

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016. 4

[2] J. Ba and R. Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014. 3

[3] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 568–573. IEEE, 2005. 2, 6

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 2

[5] H. Ding, S. K. Zhou, and R. Chellappa. Facenet2expnet: Regularizing a deep face recognition net for expression recognition. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 118–126. IEEE, 2017. 1, 2, 3, 6, 7, 8, 9

[6] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal. Recurrent neural networks for emotion recognition in video. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 467–474. ACM, 2015. 3

[7] P. Ekman and E. L. Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997. 5

[8] X. Feng, M. Pietikäinen, and A. Hadid. Facial expression recognition based on local binary patterns. *Pattern Recognition and Image Analysis*, 17(4):592–598, 2007. 2, 6

[9] Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995. 2

[10] D. Ghimire and J. Lee. Geometric feature-based facial expression recognition in image sequences using multi-class adaboost and support vector machines. *Sensors*, 13(6):7714–7734, 2013. 3

[11] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010. 4

[12] Y. Guo, G. Zhao, and M. Pietikäinen. Dynamic facial expression recognition using longitudinal facial expression atlases. In *Computer Vision–ECCV 2012*, pages 631–644. Springer, 2012. 3, 7

[13] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 3, 4

[14] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 3

[15] H. Jung, S. Lee, J. Yim, S. Park, and J. Kim. Joint fine-tuning in deep neural networks for facial expression recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2983–2991, 2015. 3, 6, 7

[16] Y. Kim, B. Yoo, Y. Kwak, C. Choi, and J. Kim. Deep generative-contrastive networks for facial expression recognition. *arXiv preprint arXiv:1703.07140*, 2017. 2, 3, 6, 7

[17] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[18] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, pages 275–1. British Machine Vision Association, 2008. 7

[19] G. Levi and T. Hassner. Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 503–510. ACM, 2015. 3

[20] M. Liu, S. Li, S. Shan, and X. Chen. Au-aware deep networks for facial expression recognition. In *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops on*, pages 1–6. IEEE, 2013. 6

[21] M. Liu, S. Li, S. Shan, R. Wang, and X. Chen. Deeply learning deformable facial action parts model for dynamic expression analysis. In *Asian Conference on Computer Vision*, pages 143–157. Springer, 2014. 6

[22] M. Liu, S. Shan, R. Wang, and X. Chen. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1749–1756, 2014. 3, 6, 7

[23] P. Liu, S. Han, Z. Meng, and Y. Tong. Facial expression recognition via a boosted deep belief network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1805–1812, 2014. 2, 6

[24] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 94–101. IEEE, 2010. 1

[25] A. Mollahosseini, D. Chan, and M. H. Mahoor. Going deeper in facial expression recognition using deep neural networks. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–10. IEEE, 2016. 2, 3, 6

[26] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 4

[27] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996. 2, 3

[28] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015. 3

[29] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014. 3

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 4

[31] A. Sanin, C. Sanderson, M. T. Harandi, and B. C. Lovell. Spatio-temporal covariance descriptors for action and gesture recognition. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 103–110. IEEE, 2013. 3

[32] K. Sikka, G. Sharma, and M. Bartlett. Lomo: Latent ordinal model for facial analysis in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5580–5589, 2016. 3, 6, 7

[33] K. Sikka, T. Wu, J. Susskind, and M. Bartlett. Exploring bag of words architectures in the facial expression domain. In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 250–259. Springer, 2012. 2

[34] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014. 4

[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 3

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 1, 4

[37] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 4

[38] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. From facial expression recognition to interpersonal relation prediction. *arXiv preprint arXiv:1609.06426*, 2016. 6

[39] G. Zhao, X. Huang, M. Taini, S. Z. Li, and M. Pietikäinen. Facial expression recognition from near-infrared videos. *Image and Vision Computing*, 29(9):607–619, 2011. 1, 7

[40] X. Zhao, X. Liang, L. Liu, T. Li, Y. Han, N. Vasconcelos, and S. Yan. Peak-piloted deep network for facial expression recognition. In *European Conference on Computer Vision*, pages 425–442. Springer, 2016. 1, 2, 3, 6, 7, 8, 9

[41] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. N. Metaxas. Learning active facial patches for expression analysis. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2562–2569. IEEE, 2012. 2, 6