Image based Static Facial Expression Recognition with Multiple Deep Network Learning

Zhiding Yu
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
yzhiding@andrew.cmu.edu

Cha Zhang
Microsoft Research
One Microsoft Way
Redmond, WA 98052
chazhang@microsoft.com

ABSTRACT

We report our image based static facial expression recognition method for the Emotion Recognition in the Wild Challenge (EmotiW) 2015. We focus on the sub-challenge of the SFEW 2.0 dataset, where one seeks to automatically classify a set of static images into 7 basic emotions. The proposed method contains a face detection module based on the ensemble of three state-of-the-art face detectors, followed by a classification module with the ensemble of multiple deep convolutional neural networks (CNN). Each CNN model is initialized randomly and pre-trained on a larger dataset provided by the Facial Expression Recognition (FER) Challenge 2013. The pre-trained models are then fine-tuned on the training set of SFEW 2.0. To combine multiple CNN models, we present two schemes for learning the ensemble weights of the network responses: by minimizing the log likelihood loss, and by minimizing the hinge loss. Our proposed method generates state-of-the-art result on the FER dataset. It also achieves 55.96% and 61.29% respectively on the validation and test set of SFEW 2.0, surpassing the challenge baseline of 35.96% and 39.13% with significant gains.

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications—*computer vision, signal processing*; I.4.m [**Image Processing and Computer Vision**]: Miscellaneous

Keywords

Facial Expression Recognition; Convolutional Neural Network; Multiple Network Learning; EmotiW 2015 Challenge

1. INTRODUCTION

Automatically perceiving and recognizing human emotions has been one of the key problems in human-computer interaction. Its associated research is inherently a multidisciplinary enterprise involving a wide variety of related fields, including computer vision, speech analysis, linguistics, cognitive psychology, robotics and learning theory, etc [38]. A computer with more powerful emotion recognition intelligence will be able to better understand human and interact more naturally. Many real world applications such as commercial call center and affect-aware game development also benefit from such intelligence.

Possible sources of input for emotion recognition include different types of signals, such as visual signals (image/video), audio, text and bio signals. For vision based emotion recognition, a number of visual cues such as human pose, action and scene context can provide useful information. Nevertheless, facial expression is arguably the most important visual cue for analyzing the underlying human emotions. Despite the continuous research efforts, accurate facial expression recognition under un-controlled environment still remains a significant challenge. Many early facial recognition datasets [23, 36, 3, 14, 27, 4, 24, 35] were collected under "lab-controlled" environments where subjects were asked to artificially generate certain expressions [8]. Such deliberate behavior often results in different visual appearances, audio profiles as well as timing [38], and is therefore by no means a good representation of natural facial expressions [8]. On the other hand, recognizing facial expressions in the wild can be considerably more difficult due to the visually varying and sometimes even ambiguous nature of the problem. Other adverse factors may include poor illumination, low resolution, blur, occlusion, as well as cultural/age differences.

Recent advances in emotion recognition focus on recognizing more spontaneous facial expressions. The Acted Facial Expressions in the Wild (AFEW) dataset [8] and the Static Facial Expressions in the Wild (SFEW) dataset [11] were collected to mimic more spontaneous scenarios and contain 7 basic emotion categories. The video clips of AFEW are extracted from movies, while SFEW is a static subset of AFEW. The idea is that movies, although not truly spontaneous, at least provide facial expressions in a much more natural and versatile way than lab-controlled datasets. This year's Emotion Recognition in the Wild (EmotiW) 2015 Grand Challenge [2] consists two sub-challenges based on AFEW 5.0 and SFEW 2.0 respectively. Both datasets present ever more difficulties than many conventional ones as a result of their more spontaneous characteristics. While a number of hand-crafted features such as Local Binary Patterns on Three Orthogonal Planes (LBP-TOP) [40], Pyramid Histogram of Oriented Gradients (PHOG) [26] and Local Quantized Patterns (LPQ) [5] were proven to work well

ICMI 2015, November 9–13, 2015, Seattle, WA, USA.
© 2015 ACM. ISBN 978-1-4503-3912-4/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2818346.2830595.

on conventional datasets, they obtain significantly lower performances on these two datasets [8].

Deep convolutional neural network has recently yielded excellent performance in a wide variety of image classification tasks [1, 17, 28, 33, 30]. The careful design of local to global feature learning with convolution, pooling and layered architecture renders very strong visual representation ability, making it a powerful tool for facial expression recognition. In this paper, we focus ourselves on the task of image based static facial expression recognition on SFEW with deep CNNs. Our main contributions can be summarized as follows: 1. We propose a CNN architecture that achieves excellent emotion recognition performance. 2. We propose a data perturbation and voting method that further increases the recognition performance of CNN considerably. 3. We propose two novel constrained optimization frameworks to automatically learn the network ensemble weights by minimizing the loss of ensembled network output responses. Our best submission, achieved with the above methods, reaches 61.29% overall accuracy on the SFEW test set, surpassing the baseline of 39.13% with a significant gain of 21.6%. The proposed framework also achieves the state-of-the-art performance on FER dataset.

2. RELATED WORKS

A number of methods on AFEW were proposed in the past two EmotiW Challenges [10, 9]. Several popular approaches such as multiple kernel learning [29, 7], multiple feature fusion [20] and score-level fusion [32, 21] were reported useful in boosting the recognition performance. Ionescu *et al.* [15] presented a local learning approach to improve bag of words model for image based facial expression recognition. Other works include [19], which proposed a facial expression recognition framework through manifold modeling of videos based on a mid-level representation.

Facial expression and emotion recognition with deep learning methods were reported in [16, 34, 22, 18, 21]. In particular, Tang [34] reported a deep CNN jointly learned with a linear support vector machine (SVM) output. His method achieved the first place on both public (validation) and private data on the FER-2013 Challenge [13]. Liu *et al.* [18] proposed a facial expression recognition framework with 3D-CNN and deformable action parts constraints in order to jointly localizing facial action parts and learning part-based representations for expression recognition. In addition, Liu *et al.* [21] included the pre-trained *Caffe* CNN models to extract image-level features. Finally, the work by Kahou *et al.* [16] is probably the most related to our proposed method. Their method respectively trained a CNN for video and a deep Restricted Boltzmann Machines (RBM) for audio. “Bag of mouth” features are also extracted to further improve the performance. Two large datasets: the Toronto Face Dataset and the Google dataset were combined to train the CNN network. The Google dataset happens to be the very dataset provided to FER-2013 and therefore our method shares part of the training set with [16]. Despite such coincidence, our proposed learning strategy differs from [16] significantly. First, [16] only used the AFEW training data to train the aggregator SVM, while we choose to pre-train our CNN model on external data and fine-tune on the SFEW training data. Fine-tuning proved to be crucial in boosting the classification performance on SFEW, as it increases the accuracy on validation set from 45% to 53%, a significant gain. Second, the ensemble weights of different models in [16] is determined with random search, while our work proposes to automatically learn the ensemble weights through optimizing certain loss functions.

3. FACE DETECTION

The SFEW dataset contains labeled movie frames. While it is possible to directly extract features at frame-level, locating faces benefits the recognition task and the face detector performance is highly correlated with the recognition accuracy. Although the face alignment results provided by EmotiW using Mixtures of Trees (MoT) [41] are accurate under many challenging scenarios, they contain an unignorable amount of missing or false positive faces. Therefore, we ensemble multiple state-of-the-art face detectors to ensure the detection accuracy. Our final face detection module consists of three detectors: the joint cascade detection and alignment (JDA) detector from [6], the Deep-CNN-based (DCNN) detector from [39] and MoT. Before face detection, all input movie frames are resized to 1024 × 576 pixels in order to restore their original aspect ratio.

JDA is able to return detected faces with very high alignment accuracy and detection precision. As a result we put this detector on the first layer of the detection module. A slight drawback, however, is that JDA’s detection recall is unsatisfactory for profile faces. The DCNN-based detector shows excellent detection performance for non-frontal and even profile faces. Under the wild environment of SFEW, it is a very good complement to JDA. For any frame with multiple detections, the largest face is returned. This strategy generally works well except in very occasional cases where the largest face is not intended for emotion recognition. Fig. 1 gives some examples of detection results using both detectors. The first two examples show that JDA gives slightly better localizations than DCNN. The third shows a more difficult case where DCNN complemented JDA. Finally, the last example shows a mistakenly returned face under multiple detections. The left larger face is returned while the right face should be the actual focus.

In rare cases where both JDA and DCNN fail, we include MoT as the last step of the detection hierarchy. An overview diagram of the modules is shown in Fig. 2. Table 1 illustrates the number of correctly detected faces on the SFEW test set using single detectors as well as two cascade combinations. Significantly boosted results are obtained by cascading different detectors. Out of the 372 SFEW test frames, 371 faces are correctly detected by the proposed cascade. Note that “JDA+DCNN” and “JDA+DCNN+MoT” are denoted as “1+2” and “1+2+3” for short.

Table 1: Number of correct detections on SFEW test set using different detectors and cascades.

	JDA	DCNN	MoT	1+2	1+2+3
Det #	333	358	352	363	**371**

4. FACE PREPROCESSING

Face preprocessing proves to be a crucial step for good recognition performance. It helps to remove irrelevant noise and unifies all faces to the same domain. Since we decide to pre-train our deep network model on FER, the detected

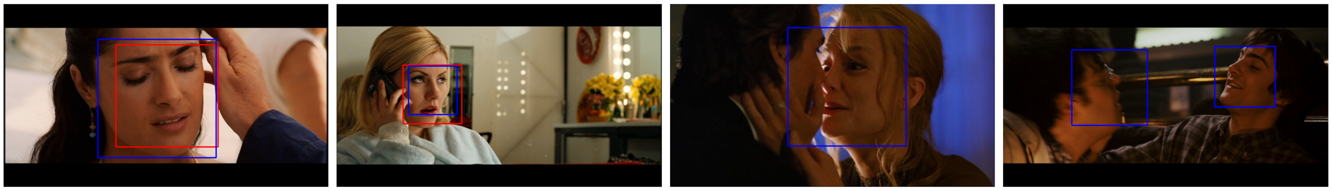


Figure 1: Examples of face detections by JDA (red) and DCNN (blue).

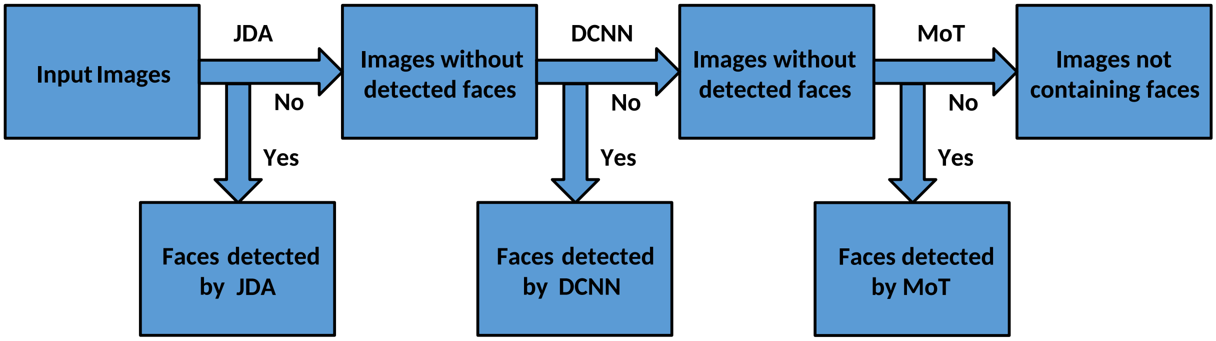


Figure 2: The system diagram of the proposed face detection module on SFEW 2.0.

faces on SFEW are all resized to 48 × 48 and are transformed to grayscale, which is exactly the same as the FER data.

Both the face images from SFEW and FER datasets are then preprocessed with standard histogram equalization, followed by a linear plane fitting to remove unbalanced illumination. Finally, the image pixel values after plane fitting are normalized to a zero mean and unit variance vector.

5. THE PROPOSED CNN MODEL

We train the deep network models based on our own C++ and Cuda implementation of a 7 hidden layer CNN. The architecture and parameters of our CNN model has been designed to optimize its performance on facial expression recognition tasks. In the rest part of this section we will describe the details of the proposed CNN model.

5.1 The Basic Network Architecture

An overview of the network architecture is shown in Fig. 3. The network contains five convolutional layers, three stochastic pooling layers and three fully connected layers. We adopted stochastic pooling [37] instead of max pooling for its good performance given limited training data. Unlike max pooling which chooses the maximum response, stochastic pooling randomly samples a response based on the probability distribution obtained by normalizing the responses. The fully connected layers contains dropout [31], another mechanism for randomization. These statistical randomness reduces the risk of network overfitting.

The input to the network are the preprocessed 48 × 48 faces. Both the second and the third stochastic pooling layers include two convolutional layers prior to pooling. The filter step height and width for all convolutional layers are both set to 1. The nonlinear mapping functions for all convolutional layers and fully connected layers are set as rectified linear unit (ReLU) [25]. For stochastic pooling layers, the window sizes are set to 3 × 3 and the strides are both set to 2. This makes the sizes of response maps reduce to half after each pooling layer.

The last stage of the network includes a softmax layer, followed by a negative log likelihood loss defined as:

$$\mathcal{L} = -\sum_{i=1}^{N} \log P(y_i|\mathbf{x}_i), \tag{1}$$

where N is the total number of training examples. $\mathbf{x}_i$ is the ith training sample, y_i is the label of $\mathbf{x}_i$, and $P(y|\mathbf{x}_i)$ is the network output response on the yth class category given $\mathbf{x}_i$. The network is trained using the adaptive subgradient method [12] with a batch size of 128 examples.

5.2 Generating Randomized Perturbation

While FER contains more than 35000 labeled samples which is considerably larger than SFEW, the classification performance can be further improved if we randomly perturb the input faces with additional transforms. The random perturbation essentially generates additional unseen training samples and therefore makes the network even more robust to deviated and rotated faces.

A similar method is reported in [16] where the authors generate perturbed training data by feeding their network with randomly cropped and flipped 40 × 40 face images from the original ones. Due to the difficult and wild nature of SFEW, the detected faces may contain a wide variety of different poses, cropped scales and deviations. To cover them as much as possible in training, we consider a much more comprehensive set of perturbations through the following randomized affine image warping:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} c & 0 \\ 0 & c \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & s_1 \\ s_2 & 1 \end{bmatrix} \begin{bmatrix} x - t_1 \\ y - t_2 \end{bmatrix} \tag{2}$$

where θ is the rotation angle randomly sampled from three different values: $\{-\frac{\pi}{18}, 0, \frac{\pi}{18}\}$. s_1 and s_2 are the skew parameters along x and y directions and are both randomly sampled from $\{-0.1, 0, 0.1\}$. c is a random scale parameter

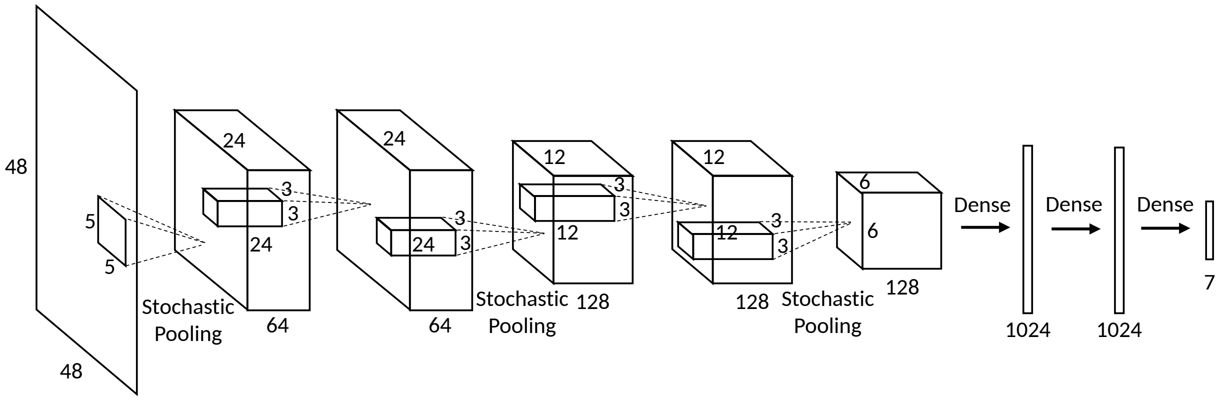


Figure 3: Network architecture of the proposed basic convolutional neural network.

defined as $c = 47/(47 - \delta)$, where δ is a randomly sampled integer on $[0, 4]$. t_1 and t_2 are two translation parameters whose values are sampled from $\{0, \delta\}$ and are coupled with c. In reality one generates the warped image with the following inverse mapping:

$$\left[\begin{array}{c} x \\ y \end{array} \right] = \mathbf{A}^{-1} \left[\begin{array}{c} x' \\ y' \end{array} \right] + \left[\begin{array}{c} t_1 \\ t_2 \end{array} \right], \tag{3}$$

where $\mathbf{A}$ is the composition of the skew, rotation and scale matrices. The input ($x' \in [0, 47], y' \in [0, 47]$) are the pixel coordinates of the warped image. Eq. (3) simply computes an inverse mapping to find the corresponding (x, y). As the computed mappings mostly contain non-integer coordinates, bilinear interpolation is used to obtain the perturbed image pixel values. For pixels mapped outside the original image, we take pixel value of its mirrored position. Finally, the input training faces are also randomly flipped to further introduce additional robustness. The top row of Fig. 4 gives 6 examples non-perturbed faces while the bottom row shows their corresponding randomly perturbed faces.

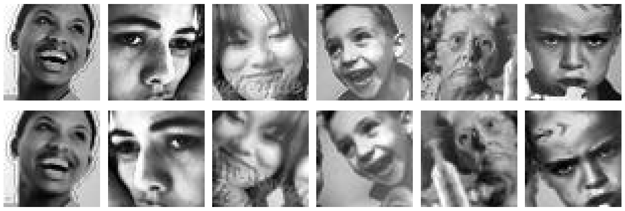


Figure 4: Examples of perturbed face with the proposed affine warping strategy.

5.3 Learning and Voting with Perturbation

With the perturbation on training set, the loss function of our network is modified to consider all perturbations:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{p=1}^{P} \log P(y_i|\mathbf{x}_i^p), \tag{4}$$

where P is the total number of perturbations. $\mathbf{x}_i^p$ is $\mathbf{x}_i$ with the pth perturbation configuration. In practice, one does not need to truly extend the training set with perturbations. Instead, the 128 samples in each batch are randomly perturbed among the P possible configurations.

An additional crucial improvement in our method is to output the response of each test image as an averaged voting of responses from all the perturbed samples:

$$P(y|\mathbf{X}_i) \triangleq \frac{1}{P}\sum_{p=1}^{P} P(y|\mathbf{x}_i^p), \tag{5}$$

where $\mathbf{X}_i \triangleq \{\mathbf{x}_i^p | p = 1, ..., P\}$. We have considered other voting strategies such as majority voting where the final label prediction is based on counting the predictions of all perturbations. Overall, averaging output response seem to render the best performance. In our experiment, voting often gives a consistent gain of roughly $2 - 3\%$. Conceptually, the test CNN architecture can be illustrated as Fig. 5.

5.4 Network Pre-training on FER

We pre-train our CNN model on the combined FER dataset formed by train, validation and test set. The initial network learning rate is set to 0.005 while the minimum learning rate is set to 0.0001. Each training epoch contains $\lceil N/128 \rceil$ number of mini batches, with the samples randomly selected from the training set and with random perturbation.

The loss and trained network parameters of each epoch are recorded. If there is an increase of training loss with more than 25% or more than 5 consecutive times of loss increase, the learning rate is reduced by half and the previous network with the best loss is reloaded. We found the network hardly overfits due to stochastic pooling and dropout. Thus after all epochs are finished, we select the network from the epoch with the best training accuracy as our final pre-trained model.

5.5 Network Fine-tuning on SFEW

The pre-trained CNN model on FER dataset gives around 45% of accuracy on the SFEW validation set without voting. While both datasets contain the same set of facial expression classes, we noticed that there exist certain level of dataset biases. Domain adaptation, therefore, is necessary for better recognition performance. Our proposed strategy is to fine-tune our network on the SFEW training set.

We adopt the same perturbation and voting strategy, as well as the network learning framework respectively described in Section 5.3 and Section 5.4. To overcome overfitting, we freeze the parameters of all the convolutional layers and only allow the update of parameters at the fully connected layers.

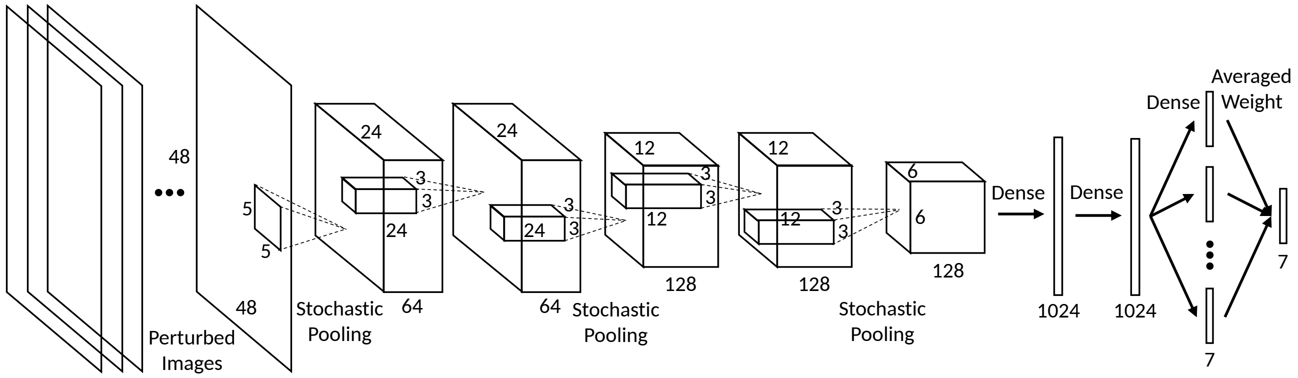


Figure 5: The improved *test* CNN architecture with random perturbations and voting.

We also observe that a slightly larger learning rate helps to reduce the risk of trapping at a local minima and benefits the fine-tuning performance. As a result the initial network learning rate is increased to 0.02.

6. MULTIPLE NETWORK LEARNING

On top of the single CNN model, we present a multiple network learning framework to further enhance the performance. A common way to ensemble multiple networks is to simply average the output responses. We observe that random initialization not only leads to varying network parameters, but also renders diverse network classification abilities for different data. In this case, ensemble with averaged weight is probably sub-optimal as voting is conducted without any discrimination. A better way is to adaptively assign different weights to each network such that the ensembled network responses complement each other.

To learn the ensemble weights $\mathbf{w}$, we independently train multiple differently initialized CNNs and output their training responses. A loss is defined on the weighted ensemble response, with $\mathbf{w}$ optimized to minimize such loss. At testing, the learned $\mathbf{w}$ is also used to compute the ensembled test response. In this paper, we consider the following two optimization frameworks:

6.1 Optimal Ensembled Log Likelihood Loss

The first multiple network learning framework seeks to minimize the following ensembled log likelihood loss:

$$\begin{aligned} &\min_{\mathbf{w}} -\sum_{i=1}^{N}\log\sum_{k=1}^{K}P_k(y_i|\mathbf{X}_i)w_k+\lambda\sum_{k=1}^{K}w_k^2\\ &\text{s.t.}\ \sum_{k=1}^{K}w_k=1\\ &\qquad w_k\geq 0,\ \forall k \end{aligned} \tag{6}$$

In the objective function, N is the number of training samples, and K is the number of networks. $P_k(y|\mathbf{X}_i)$ is the kth network output response on the yth category given the set of perturbed samples $\mathbf{X}_i$. An l_2 norm regularizer is imposed on the ensemble weights such that the weights are not concentrated on very few networks and the ensemble does not overfit. λ is determined by maximizing the validation accuracy. To maintain a probabilistically meaningful ensembled output response, a convex combination constraint is also imposed on $\mathbf{w}$.

6.2 Optimal Ensembled Hinge Loss

Another considered objective is the following hinge loss:

$$\begin{aligned} &\min_{\mathbf{w}}\sum_{i=1}^{N}\sum_{y\neq y_i}\left[1-\frac{\sum_{k=1}^{K}(P_k^{i,y_i}-P_k^{i,y})w_k}{\gamma}\right]_+ +\lambda\sum_{k=1}^{K}w_k^2\\ &\text{s.t.}\ \sum_{k=1}^{K}w_k=1\\ &\qquad w_k\geq 0,\ \forall k \end{aligned} \tag{7}$$

where $P_k^{i,y}\triangleq P_k(y|\mathbf{X}_i)$. The intuition is that the ensembled output response corresponding to ground truth should be larger than others with a margin γ. With the hinge loss, any case where the response difference is larger than γ will not introduce any penalty. Again, both γ and λ are determined with respect to the accuracy on validation set.

We could have also included the validation loss in our objective to potentially generate better results. However, we decide to strictly adhere to the definition of validation and only use it to determine the fine-tune epoch number.

7. EXPERIMENTAL RESULTS

We conduct a comprehensive set of experiments on both FER and SFEW. The following section reports the performance of our proposed methods on these two datasets.

7.1 Experiment on FER

We first conduct experiment on the FER dataset with single network model. The dataset contains 28709 training images, 3589 validation(public) images and 3589 test(private) images. Fig. 6 shows the training and test accuracies with respect to the number of epochs during training. Note we show the testing accuracy curves of both voting and non-voting (no perturbation at testing) based methods. Clearly, voting with perturbations has a constant gain. The performance of multiple network learning and baselines are shown in Fig. 7, where "Single" refers to the average accuracy of 6 randomly initialized single CNN models (with voting). "Average" refers to averaged ensemble of these networks. The results of FER-2013 Champion are also listed. The proposed multiple network learning is also based on learning with same single CNN models. For the log likelihood loss framework, sub-sampling is conducted 10 times with the sampling rate set to 0.1. λ is set to 280. For the hinge loss framework, γ and λ respectively equals to 0.3 and 7000. The learned network weights are shown in Table 2.

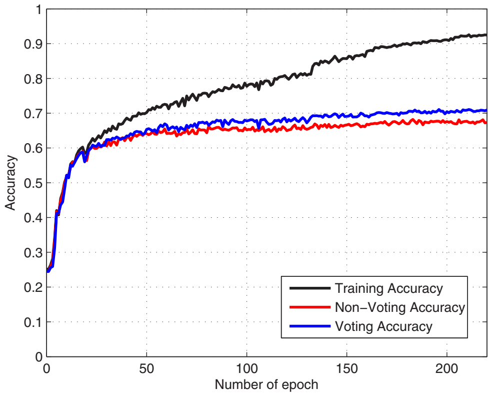


Figure 6: The training and testing accuracy curves on FER dataset.

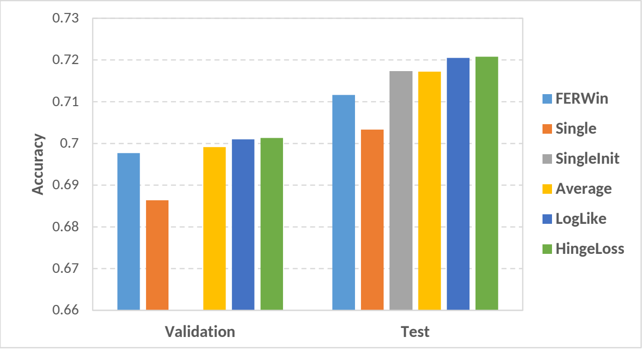


Figure 7: Classification accuracies of different methods on the FER validation and test set.

Both the proposed ensemble frameworks have surpassed the FER-2013 winner and the average ensemble. Although randomly initialized single model gives slightly worse performance than the champion, we happen to observe that a simple initialization with a previously trained network (without skewing) gives another boost surpassing the champion. Given the observation, we expect that our method can achieve even better results with re-trained networks.

7.2 EmotiW 2015 Results

Fig. 8 shows the fine-tuning accuracy curves of both voting and non-voting based methods on the SFEW validation set. The CNN model is first pre-trained with the combined FER dataset. Again one could see that voting based method constantly outperforms non-voting based method. Finally, We test the proposed multiple network learning on SFEW dataset. Fig. 9 shows both the validation and the test accuracies of our methods and several baselines. In addition, Table 3 shows the corresponding accuracy numbers. In our EmotiW submissions, we mainly experimented with the following baselines: 1. Single CNN model (Single) with random perturbation and voting; 2. Average ensemble with bagging (Average1) where each single CNN model is randomly initialized, pre-trained with randomly sub-sampled FER combined set, and then fine-tuned on SFEW; 3. Average ensemble (Average2) where each single CNN model is trained similar to 2 except without sub-sampling on FER; 4. SVM ensemble (SVM) where each single CNN model is the same as 3 and an SVM is trained and tested on the concatenated network output responses.

Table 2: Learned ensemble weights for each network.

	N#1	N#2	N#3	N#4	N#5	N#6
LL	0.2171	0.2481	0.2943	0	0.1119	0.1286
HL	0.2308	0.2345	0.2805	0	0.1068	0.1473

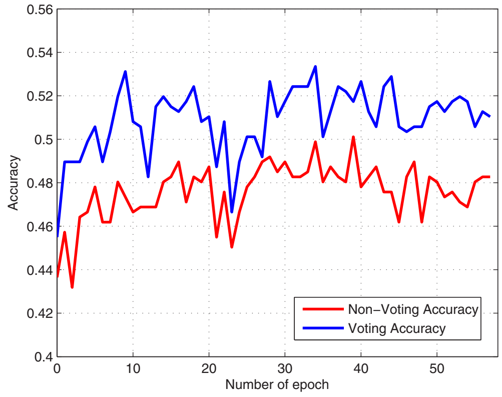


Figure 8: The fine-tuning accuracies of voting and non-voting based methods on SFEW validation set.

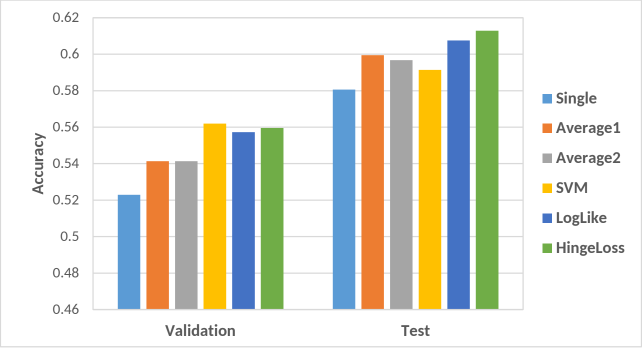
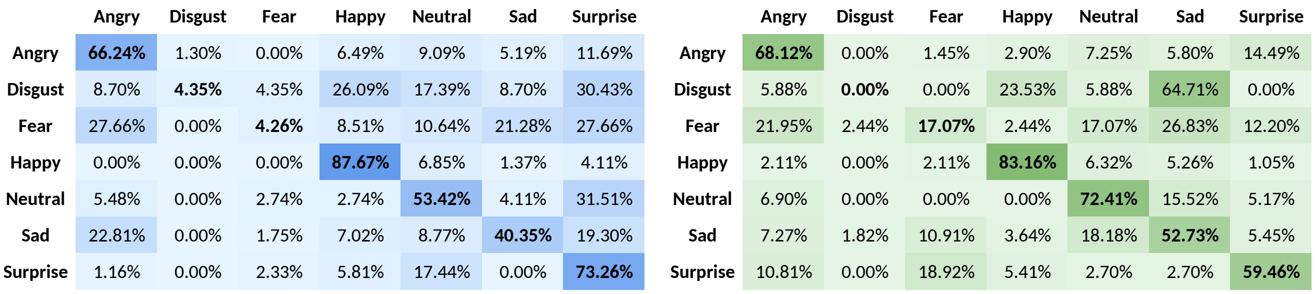


Figure 9: Classification accuracies of different methods on SFEW validation and test set.

Table 3: Classification accuracies (%) of different methods on SFEW validation and test set.

Acc	Single	Avg1	Avg2	SVM	LL	HL
Val	52.29	54.13	54.13	**56.19**	55.73	55.96
Test	58.06	59.95	59.67	59.14	60.75	**61.29**

The proposed two ensemble frameworks again achieve the best performance, with respectively 60.75% and 61.29% accuracy on the test set. In the experiment, λ in the log likelihood loss (denoted as "LL") framework is set to 600, while γ and λ in the hinge loss (denoted as "HL") framework are set to 0.1 and 400, all based on validation. Fig. 10 and 11 respectively shows the confusion matrices of both frameworks.



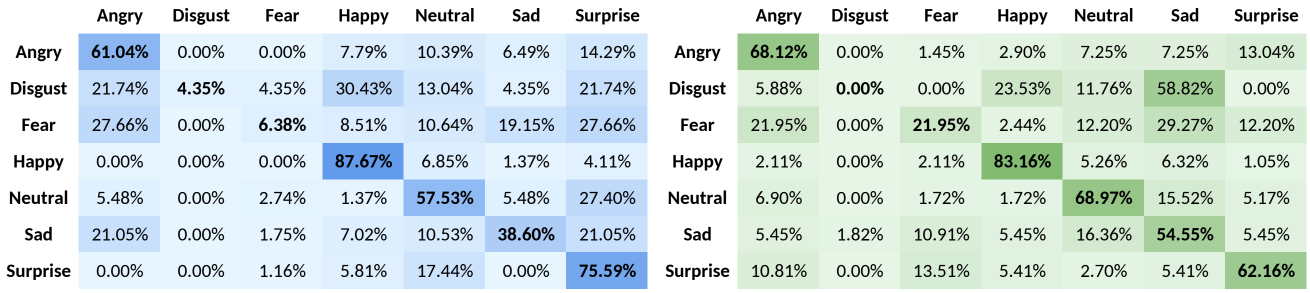
	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Angry	**66.24%**	1.30%	0.00%	6.49%	9.09%	5.19%	11.69%
Disgust	8.70%	**4.35%**	4.35%	26.09%	17.39%	8.70%	30.43%
Fear	27.66%	0.00%	**4.26%**	8.51%	10.64%	21.28%	27.66%
Happy	0.00%	0.00%	0.00%	**87.67%**	6.85%	1.37%	4.11%
Neutral	5.48%	0.00%	2.74%	2.74%	**53.42%**	4.11%	31.51%
Sad	22.81%	0.00%	1.75%	7.02%	8.77%	**40.35%**	19.30%
Surprise	1.16%	0.00%	2.33%	5.81%	17.44%	0.00%	**73.26%**

(a) Validation set

	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Angry	**68.12%**	0.00%	1.45%	2.90%	7.25%	5.80%	14.49%
Disgust	5.88%	**0.00%**	0.00%	23.53%	5.88%	64.71%	0.00%
Fear	21.95%	2.44%	**17.07%**	2.44%	17.07%	26.83%	12.20%
Happy	2.11%	0.00%	2.11%	**83.16%**	6.32%	5.26%	1.05%
Neutral	6.90%	0.00%	0.00%	0.00%	**72.41%**	15.52%	5.17%
Sad	7.27%	1.82%	10.91%	3.64%	18.18%	**52.73%**	5.45%
Surprise	10.81%	0.00%	18.92%	5.41%	2.70%	2.70%	**59.46%**

(b) Test set

Figure 10: Confusion matrices of the optimal log likelihood loss ensemble framework on SFEW.



	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Angry	**61.04%**	0.00%	0.00%	7.79%	10.39%	6.49%	14.29%
Disgust	21.74%	**4.35%**	4.35%	30.43%	13.04%	4.35%	21.74%
Fear	27.66%	0.00%	**6.38%**	8.51%	10.64%	19.15%	27.66%
Happy	0.00%	0.00%	0.00%	**87.67%**	6.85%	1.37%	4.11%
Neutral	5.48%	0.00%	2.74%	1.37%	**57.53%**	5.48%	27.40%
Sad	21.05%	0.00%	1.75%	7.02%	10.53%	**38.60%**	21.05%
Surprise	0.00%	0.00%	1.16%	5.81%	17.44%	0.00%	**75.59%**

(a) Validation set

	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Angry	**68.12%**	0.00%	1.45%	2.90%	7.25%	7.25%	13.04%
Disgust	5.88%	**0.00%**	0.00%	23.53%	11.76%	58.82%	0.00%
Fear	21.95%	0.00%	**21.95%**	2.44%	12.20%	29.27%	12.20%
Happy	2.11%	0.00%	2.11%	**83.16%**	5.26%	6.32%	1.05%
Neutral	6.90%	0.00%	1.72%	1.72%	**68.97%**	15.52%	5.17%
Sad	5.45%	1.82%	10.91%	5.45%	16.36%	**54.55%**	5.45%
Surprise	10.81%	0.00%	13.51%	5.41%	2.70%	5.41%	**62.16%**

(b) Test set

Figure 11: Confusion matrices of the optimal hinge loss ensemble framework on SFEW.

8. CONCLUSIONS

In this paper, we have proposed a deep convolutional neural network based facial expression recognition method, with multiple improved frameworks to further boost the performance. Our proposed method achieves excellent results on both FER and SFEW dataset, indicating the considerable potential of our facial expression recognition method.

9. REFERENCES

[1] Cuda-convnet Google code home page. https://code.google.com/p/cuda-convnet/.

[2] The Third Emotion Recognition in The Wild (EmotiW) 2015 Grand Challenge. http://cs.anu.edu.au/few/emotiw2015.html.

[3] T. Bänziger and K. R. Scherer. Introducing the geneva multimodal emotion portrayal (gemep) corpus. *Blueprint for affective computing: A sourcebook*, pages 271–294, 2010.

[4] M. S. Bartlett, G. C. Littlewort, M. G. Frank, C. Lainscsek, I. R. Fasel, and J. R. Movellan. Automatic recognition of facial actions in spontaneous expressions. *Journal of multimedia*, 1(6):22–35, 2006.

[5] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408. ACM, 2007.

[6] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *European Conference on Computer Vision (ECCV)*, 2014.

[7] J. Chen, Z. Chen, Z. Chi, and H. Fu. Emotion recognition in the wild with feature fusion and multiple kernel learning. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 508–513. ACM, 2014.

[8] A. Dhall et al. Collecting large, richly annotated facial-expression databases from movies. 2012.

[9] A. Dhall, R. Goecke, J. Joshi, K. Sikka, and T. Gedeon. Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 461–466. ACM, 2014.

[10] A. Dhall, R. Goecke, J. Joshi, M. Wagner, and T. Gedeon. Emotion recognition in the wild challenge 2013. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 509–516. ACM, 2013.

[11] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 2106–2112, 2011.

[12] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

[13] I. J. Goodfellow, D. Erhan, P. L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, et al. Challenges in representation learning: A report on three machine learning contests. In *Neural Information Processing*, pages 117–124. Springer, 2013.

[14] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010.

[15] R. T. Ionescu, M. Popescu, and C. Grozea. Local learning to improve bag of visual words model for facial expression recognition. In *Workshop on Challenges in Representation Learning, ICML*, 2013.

[16] S. E. Kahou, C. Pal, X. Bouthillier, P. Froumenty, Ç. Gülçehre, R. Memisevic, P. Vincent, A. Courville, Y. Bengio, R. C. Ferrari, et al. Combining modality specific deep neural networks for emotion recognition in video. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 543–550. ACM, 2013.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

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

[19] M. Liu, S. Shan, R. Wang, and X. Chen. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1749–1756. IEEE, 2014.

[20] M. Liu, R. Wang, Z. Huang, S. Shan, and X. Chen. Partial least squares regression on grassmannian manifold for emotion recognition. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 525–530. ACM, 2013.

[21] M. Liu, R. Wang, S. Li, S. Shan, Z. Huang, and X. Chen. Combining multiple kernel methods on riemannian manifold for emotion recognition in the wild. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 494–501. ACM, 2014.

[22] P. Liu, S. Han, Z. Meng, and Y. Tong. Facial expression recognition via a boosted deep belief network. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1805–1812. IEEE, 2014.

[23] 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.

[24] G. McKeown, M. F. Valstar, R. Cowie, and M. Pantic. The semaine corpus of emotionally coloured character interactions. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 1079–1084. IEEE, 2010.

[25] 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.

[26] V. Ojansivu and J. Heikkilä. Blur insensitive texture classification using local phase quantization. In *Image and signal processing*, pages 236–243. Springer, 2008.

[27] M. Pantic, M. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 5–pp. IEEE, 2005.

[28] 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*, pages 1–42, 2014.

[29] K. Sikka, K. Dykstra, S. Sathyanarayana, G. Littlewort, and M. Bartlett. Multiple kernel learning for emotion recognition in the wild. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 517–524. ACM, 2013.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[32] B. Sun, L. Li, T. Zuo, Y. Chen, G. Zhou, and X. Wu. Combining multimodal features with hierarchical classifier fusion for emotion recognition in the wild. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 481–486. ACM, 2014.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, 2015.

[34] Y. Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013.

[35] A. J. Toole, J. Harms, S. L. Snow, D. R. Hurst, M. R. Pappas, J. H. Ayyad, and H. Abdi. A video database of moving faces and people. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):812–816, 2005.

[36] F. Wallhoff. Facial expressions and emotion database. *Technische Universität München*, 2006.

[37] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013.

[38] Z. Zeng, M. Pantic, G. Roisman, T. S. Huang, et al. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):39–58, 2009.

[39] C. Zhang and Z. Zhang. Improving multiview face detection with multi-task deep convolutional neural networks. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 1036–1041. IEEE, 2014.

[40] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):915–928, 2007.

[41] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.