Estimation of Affective Level in the Wild With Multiple Memory Networks

Jianshu Li [1,2] Yunpeng Chen [1] Shengtao Xiao [1] Jian Zhao [1]
Sujoy Roy [2] Jiashi Feng [1] Shuicheng Yan [1] Terence Sim [1]

[1]National University of Singapore [2] SAP Innovation Center Network Singapore

{jianshu, chenyunpeng, xiao_shengtao, zhaojian90}@u.nus.edu sujoy.roy@sap.com
{elefjia, eleyans}@nus.edu.sg tsim@comp.nus.edu.sg

Abstract

This paper presents the proposed solution to the "affect in the wild" challenge, which aims to estimate the affective level, i.e. *the valence and arousal values, of every frame in a video. A carefully designed deep convolutional neural network (a variation of residual network) for affective level estimation of facial expressions is first implemented as a baseline. Next we use multiple memory networks to model the temporal relations between the frames. Finally ensemble models are used to combine the predictions from multiple memory networks. Our proposed solution outperforms the baseline model by a factor of* 10.62% *in terms of mean square error (MSE).*

1. Introduction

The "affect in the wild" challenge is a facial expression estimation challenge for estimating the valence and arousal values of the faces in videos. Unlike previous datasets which ascribe video-level valence-arousal annotations, the uniqueness of the dataset used in the challenge is the availability of frame-level annotation of valence-arousal for faces in natural ("in the wild") videos. This opens up novel avenues for understanding and modeling facial expressions "in the wild" in terms of valence and arousal. In this challenge, facial expression at a frame has context and hence can be modeled as a continuous predicate instead of a single isolated value. It also allows for understanding how facial expressions transform over time, which is useful for estimating more subtle facial expressions. Some examples of the faces with valence and arousal values are illustrated in Figure 1.

In our proposed solution, we use the major face in the video as the input to estimate its affective level in terms of valence and arousal values. Face based emotion recognition is a well-formulated problem and there are numerous methods attempting to solve it [2, 9, 7, 6]. Traditional face emotional analysis casts the emotion recognition problem into a classification problem [10, 8]. Given an input face image, the model aims to predict the correct emotion category of the face, such as "happy", "sad", "angry", "surprise", "disgust" and "neutral". Deep neural networks have been used to predict the class of the emotion of human faces and they have demonstrated excellent performance over traditional emotion recognition models. For the task at hand, the goal is to estimate the valence and arousal values for the face. In this work we model the task as a regression problem and adopt a single deep neural network to predict both the valence and arousal levels of human faces.

Our proposed solution also models the temporal information between video frames. For affective computing in videos, the contextual information provides useful cues for affective estimation of the current frame. To model the temporal information, we use a bi-directional long short-term memory network (BDLSTM) [4, 12]. As the number of frames in each video is very large, ranging from a few thousands to tens of thousands, the temporal relationship between all the frames is hard to model. So we use a sliding window approach on the video frames. We set a predefined length of input frames, and use a bi-directional LSTM with the predefined length as the model. The bi-directional LSTM is used to perform a sliding window on the whole video sequence, and to predict the value of affective level for all the frames in the sequence. In such a manner, the prediction of the current frame is affected by the nearby frames in both forward and backward directions, and the temporal information in a short term is considered when predicting the faces in the sequence.

Contributions: 1) We propose a solution to the "affect in the wild" challenge, which contains a single deep neu-

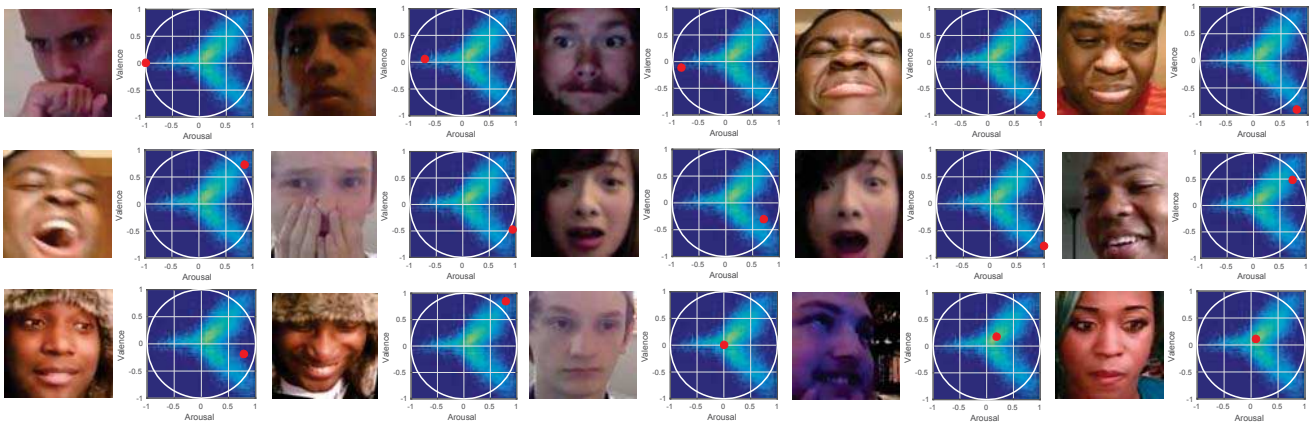

Figure 1. Examples of faces with annotated affective levels. For each pair, the left side shows the face, and the right side shows the corresponding affective levels in terms of valence and arousal values, as indicated by the red dots. The rendered background indicates the overall distribution of valence and arousal values.

ral network to learn discriminative face features and a bi-directional LSTM to model the temporal information between video frames. 2) The proposed solution achieves performance boost on the Aff-Wild dataset from the challenge compared with the baseline method.

2. The Proposed Approach

This section describes the proposed approach for the problem of dense prediction of affective level for faces in every frame in a video. The proposed framework breaks the problem into two sub-problems and solves each sub-problem separately, making it a two-stages approach. In the first stage, we consider how to learn and extract discriminative deep feature representations from each frame. In the second stage, we consider how to utilize the temporal information to leverage the power feature aggregation to achieve accurate dense affective level prediction.

Deep Face Feature Learning The aim of this stage is to extract rich and diverse facial features for the following affective level prediction as illustrated in Figure 2. The richness and diversity of the learned feature representation is critical for the following stage since the extracted features serve as the input to the following stage. In this part, we propose a novel convolutional neural network (CNN) architecture based on Collective Residual Unit (CRU) [13]. The novel architecture is called CRU-Net-56-tiny, which is a modified version of the CRU-Net-56 proposed in [13]. The new proposed CRU-Net-56-tiny is lighter than CRU-Net-56 with only half the model size. The significantly smaller model size reduces the number of learnable parameters and greatly alleviates the over-fitting problem. The time consumption for training the network is also much less than the original network.

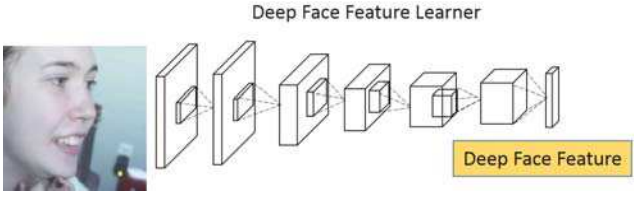

Figure 2. A deep face feature learner realized by a convolutional neural network. The input is an aligned face and the output is the deep face feature learned by the deep face feature learner.

The proposed CNN is designed based on the most recently proposed CRU-Net [13]. Different from the CRU-Nets proposed in [13], the CRU-Net-56-tiny only has half the size of ResNet-50 [5] but enjoys higher generalization

stage	output	**CRU-Net-56 (32×4d @×14)**	**CRU-Net-56-tiny (32×4d @×14)**
conv1	112x112	7 × 7, 64, stride 2	7 × 7, 64, stride 2
conv2	56x56	3 × 3 max pool, stride 2	3 × 3 max pool, stride 2
		[1×1, 128 3×3, 128, R=32 1×1, 256] × 3	[1×1, 256 3×3, 256, R=32 1×1, 256] × 3
conv3	28×28	[1×1, 256 3×3, 256, R=32 1×1, 512] × 4	[1×1, 512 3×3, 512, R=32 1×1, 256] × 4
conv4	14×14	[1×1, 640 3×3, 640, R=640 1×1, 640 1×1, 1024] × 6	[1×1, 512 3×3, 512, R=512 1×1, 1024 1×1, 256] × 6
conv5	7×7	[1×1, 1024 3×3, 1024, R=32 1×1, 2048] × 3	[1×1, 1024 3×3, 1024, R=32 1×1, 512] × 4
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params		$\mathbf{25.5} \times 10^6$	$\mathbf{12.4} \times 10^6$

Table 1. The overall architecture of our proposed deep CNN. Our new proposed CRU-Net-56-tiny has a less number of parameters than CRU-Net-56. Thus it has increased generalization ability and alleviates the over-fitting problem.

ability than the ResNet-50 (22.9% v.s. 24.0% [1] Top-1 error). The motivation of building such a CRU-Net-56-tiny is to alleviate the over-fitting problem. Table 1 shows the detailed network architecture of the proposed CRU-Net-56-tiny. Compared with the original CRU-Net-56, we propose to use thinner shortcut. The number of channels in the first shortcut keeps unchanged, while the next stage has a much less number of channels. As a result, the overall number of learnable parameters has been reduced to less than half the size of the vanilla CRU-Net-56. Based on experimental results, we find such modification only increases 1% Top-1 error rate on the ImageNet [11] classification task compared with the vanilla CRU-Net-56 (22.9% v.s.21.9%), but enjoys a lighter network with a much less chance of overfitting on the training set. We adopt the new proposed CRU-Net-56-tiny on the dataset in the challenge to learn the discriminative representations of faces for the task of affective level prediction.

Feature Aggregation with BDLSTM For feature aggregation, we use the Bi-directional Long Short-Term Memory (BDLSTM) network. As a special case of bidirectional recurrent neural networks, BDLSTM enjoys the privilege of an increased amount of input information. Different from multilayer perceptron and time delayed neural networks, where the input data need to have fixed lengths, BDLSTM can take in a data sequence of arbitrary lengths. BDLSTM can also access the data both in the past and in the future, *i.e.* the previous and future frames in a video sequence, to make the prediction of the current frame. The basic idea of BDLSTM is to connect two hidden layers of opposite directions to a combined feature space, based on which the prediction of the current frame is made. BDLSTM is useful when the context of the input in the current frame is needed. For affective computing in videos, context information is important, as the affective level of one frame is affected by adjacent frames both in the near past and in the near future.

The structure of the BDLSTM for face affective level prediction in videos is shown in Figure 3. The input features to the BDLSTM are extracted from the deep neural network in the first stage, *i.e.* CNN features. The output of the BDLSTM are the aggregated features of the input considering the nearby frames. Within BDLSTM, there are two streams of information flow, one in the forward direction and the other one in the backward direction. Given a sequence of input CNN features, the LSTM unit in the forward direction scans the sequence from left to right and performs feature encoding accordingly. Specifically, the LSTM will selectively remember important features from the inputs and forget less important features with its internal input gate and forget gate, respectively. The LSTM also stores useful features within its internal memory cell and produces the output with an output gate. In a similar manner the LSTM in the backwards direction scans the input sequence in the reverse order and the aggregation of features is also performed in the reverse order. Finally the features of the LSTM in both forward and backward directions are combined so that complimentary information is obtained as the aggregated feature. The aggregation of features is densely done for every frame in the sequence, generating a sequence of aggregated features. Based on the sequence of aggregated features, affective levels for each frame in the sequence are predicted using a regression model.

Since a video is long, ranging from several thousand frames to tens of thousands frames, we cannot simply feed the whole video as a single sequence to BDLSTM. The long-term dependency of the affective levels is very weak, *i.e.* the affective level of the current frame is not strongly related to the affective level of a frame a thousand frames later. So a long sequence of input does not make sense in this task. Therefore, we segment each video into multiple short sequences with overlapping, where each sequence is used to train the BDLSTM. The multiple sequences are generated in a sliding window fashion. To achieve this, we only need to define a length of the sequence, also known as window length L, and the step length between the starting point of each sequence, which is also called stride S. With a pre-defined window length and a stride, we can segment a video into multiple sequences. The process is illustrated in Figure 4. Since the videos in this task only contain a long continuous shot, no shot segmentation is needed to cut the video into continuous clips before using the sliding window method to cut the video into sequences.

During training stage, we randomly shuffle the videos. For each video, we feed all the sequences generated in the video to train the BDLSTM. During the testing stage, we use the same window length as the training stage and use a stride of 1 to densely scan through the video to make predictions for each frame. In this manner, each frame will have multiple predictions in different sliding windows. More explicitly, the number of predictions of the frames is equal to the window length, except for the beginning and ending frames, which will have less predictions. The final prediction of one frame is the average prediction of all the sliding widows containing that frame.

Feature Ensemble with Multiple Memory Networks So far we have modeled each single frame with a deep CNN and the temporal information within a relatively short range with a BDLSTM. Since the task of affective level prediction in video frames is a challenging task, we employs multiple memory networks, *i.e.* BDLSTM, to perform the task. More specifically we use memory networks with different memory cells and different input sequence lengths.

For ensemble of results from multiple BDLSTM, we use both prediction level ensemble and feature level ensemble.

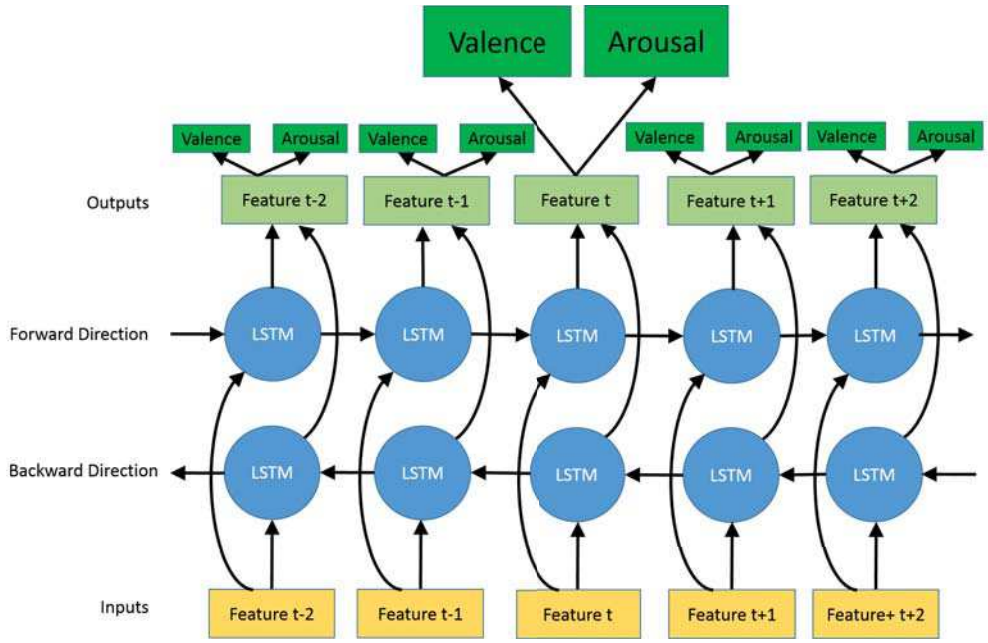

Figure 3. The overall structure of BDLSTM used in our approach. The BDLSTM has two streams of information flows in forward and backward directions. The input to BDLSTM is a sequence of deep face features and the output is a sequence of aggregated face features. The output feature of one frame aggregates features of the frames both before and after the current frame.

For prediction level ensemble, the prediction results for the same frame from different memory networks are averaged as the final prediction. For feature level ensemble, we proceed as follows.

For one BDLSTM with memory cell size C and input sequence length L, we can generate the feature of the current frame by considering the forward stream and the backward stream within the BDLSTM. Each of the stream will provide a feature vector with dimension C and totally a feature vector of length $2C$ is produced. Also the current frame is processed by L sliding windows (here we ignore the beginning and ending frames, which are contained by less than L sliding windows), so when concatenating all the features from the sliding windows, the total feature length will be $2CL$. For the beginning and ending frames, we repeat the features of length $2C$ to fill up a length of $2CL$ by repeating each of the $2C$ features in a circular manner. With the above described method, we convert each input frame to a feature vector of length $2CL$ as the representation of the input frame under one BDLSTM model.

When dealing with multiple BDLSTM models with different cell sizes and input lengths, we use concatenation to fuse all the features from each BDLSTM. So the total feature length will be

$$|f_{fusion}| = \sum_{m=1}^{M} 2C_m L_m, \tag{1}$$

where f_{fusion} is the fused feature and M is the number of total models to be fused. Based on the fused feature f_{fusion}, we can use regression models to predict the valence and arousal values of the current frame.

3. Experiments

In this section, we firstly introduce the experimental setting. Then we show experimental results and analysis of the results.

3.1. Experimental Setting

3.1.1 Dataset

In this paper, all the experiments are performed on Aff-Wild, the dataset provided by the challenge [14, 15]. The Aff-Wild dataset is meant for analyzing continuous emotion dimensions, *i.e.* valence and arousal. It contains about 300 videos annotated with regard to valence and arousal all captured "in-the-wild" and the main source is Youtube videos.

Among all the videos, 252 videos are provided for training and the remaining for testing. We randomly choose 15% of the videos, *i.e.* 38 videos as the validation set and the remaining 214 videos are used as the training set. Totally there are 841,000 frames in the training set and 127,000 frames in the validation set. We train models with the training set and report performance on the validation set. We also report the performance on the test set based on the validation set.

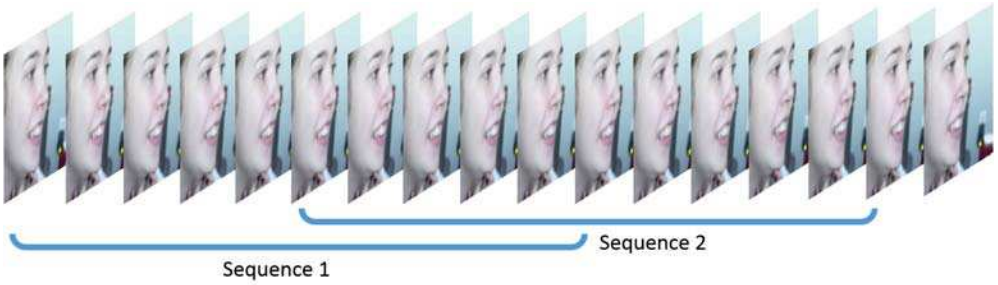

Figure 4. An illustration of separating sequences from the videos. In the figure a window length of 11 and a stride of 5 are used, *i.e.* $L = 11$ and $S = 5$. The deep face features of every face in one sequence are used as input to the BDLSTM model.

3.1.2 Implementation Details

We perform pre-processing of the videos in the following manner. First we use the provided face bounding boxes, which are detected faces filtered with the tracking algorithm, to crop the faces from the frames in each video. For all the cropped faces, we use a face landmark detector to detect the facial landmarks on the faces. Then the detected facial landmarks are used to align the faces. The inputs into the deep face feature extractor are the aligned faces. The size of the aligned face is 128×128, and a random crop of 112×112 is performed during the training stage. Random mirroring of the input face is also performed during training. During testing, the center crop is used and no mirroring is applied.

For the deep face feature extractor, we use deep neural network CRU-Net-56-tiny. The network is pre-trained on ImageNet [11] and then adapted to FER2013 dataset [3]. The first one is a large scale image dataset for image classification task and the latter one is a facial emotion recognition dataset. Then the network is fine-tuned with the training set of the challenge dataset with 20 epochs, with the first 10 epoch at learning rate 10^{-3}, next 5 epochs at learning rate 10^{-4}, next 3 epochs at learning rate 10^{-5} and final 2 epochs at learning rate 10^{-6}. After fine-tuning, the network is used to extract features from input faces and the extracted features are vectors with dimension 512.

For the multiple memory networks, we use BDLSTM with different sizes of cell memory $C = 250$, $C = 50$, $C = 25$ and $C = 16$. For input length L we use $L = 11$ and $L = 5$. The stride S for training is fixed as half of the input length, *i.e.* $S = 5$ and $S = 2$, respectively. The regression model which produces the output from the feature of BDLSTM is realized by a neural network with one fully connected layer with two output units. The two output units have no activations, and one unit is for prediction of valence and the other is for prediction of arousal. The memory networks are trained with RMSprop optimizer for 200 epochs. The initial learning rate is 0.02 and reduced by a factor of 10 when the validation MSE does not decrease for 10 epochs.

For feature level ensemble of multiple memory networks, we realize the regression model with a multilayer perceptron. Similar to the regression model in BDLSTM, the regression model for ensembled features also has two output units with no activations. The regression model is also trained with RMSprop optimizer for 200 epochs.

3.1.3 Evaluation Metric

In the affective computing task, we use Mean Square Error (MSE) and Concordance Correlation Coefficient (CCC) to evaluate the performance of the prediction of valence and arousal values. For a video sequence with N frames, we have a sequence of ground truth annotations for valence values $y_n^v, \forall n = 1, 2, \cdots, N$ and arousal values $y_n^a, \forall n = 1, 2, \cdots, N$. For each video sequence, we also have a sequence of predictions for valence $\hat{y}_n^v$ and a sequence of predictions for arousal $\hat{y}_n^a$.

With these notations, the MSEs for both valence and arousal are defined as

$$\begin{aligned} MSE^v &= \frac{1}{N}\sum_{n=1}^{N}(y_n^v - \hat{y}_n^v)^2, \\ MSE^a &= \frac{1}{N}\sum_{n=1}^{N}(y_n^a - \hat{y}_n^a)^2. \end{aligned} \tag{2}$$

The MSE will measure how much the predictions are deviated from the respective ground truth values. A smaller MSE indicates better performance.

The CCC for valence is defined as

$$CCC^v = \frac{2\sigma_{y^v\hat{y}^v}}{\sigma_{y^v} + \sigma_{\hat{y}^v} + (\mu_{y^v} - \mu_{\hat{y}^v})^2}, \tag{3}$$

where the mean is computed as

$$\begin{aligned} \mu_{y^v} &= \frac{1}{N}\sum_{n=1}^{N} y_n^v, \\ \mu_{\hat{y}^v} &= \frac{1}{N}\sum_{n=1}^{N} \hat{y}_n^v, \end{aligned} \tag{4}$$

the variance is computed as:

$$\sigma_{y^v} = \frac{1}{N}\sum_{n=1}^{N}(y_n^v - \mu_{y^v})^2,$$
$$\sigma_{\hat{y}^v} = \frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n^v - \mu_{\hat{y}^v})^2, \quad (5)$$

and finally the covariance is calculated as

$$\sigma_{y^v\hat{y}^v} = \frac{1}{N}\sum_{n=1}^{N}(y_n^v - \mu_{y^v})(\hat{y}_n^v - \mu_{\hat{y}^v}). \quad (6)$$

The CCC for arousal is defined in a similar way by replacing the superscript $(\cdot)^v$ by superscript $(\cdot)^a$:

$$CCC^a = \frac{2\sigma_{y^a\hat{y}^a}}{\sigma_{y^a} + \sigma_{\hat{y}^a} + (\mu_{y^a} - \mu_{\hat{y}^a})^2}. \quad (7)$$

The CCC measures how the distribution of the predictions matches the distribution of the ground truth values. A larger CCC value indicates better match of them and thus better performance.

3.2. Results and Analysis

During the experiments, we find that there is always severe over-fitting of the network on the training data. Although there are about 800,000 images in the training set, they are only from 214 videos. The number of images is much larger than the number of identities. Each video focuses on the affective level of only one person. We find that the network can easily achieve very low MSE (0.02) on the training set and only obtain about 0.1 MSE on the validation set. We have tried various models with different techniques to reduce the over-fitting so that the performance can be improved on the validation and testing sets.

For performance comparison and validation, we use a single value metric, which is the average of the MSE for valence and arousal. For the validation set, we also provide additional evaluation metrics described in the previous section. Since our model considers faces as input to predict the affective level, for the frames with no faces, the model will not produce predictions and we use linear interpolation to guess the mission values. So the results for the frames with faces and all the frames are different, but the trend is mostly the same. In the experiments we use label FACE to indicate that the results are obtained from only the frames with faces and we use label ALL to indicate that the results are obtained from all the frame.

3.2.1 Baseline Deep CNN

We test the performance on the baseline deep CNN directly. The results are shown in Table 2. For the results in Table 2, the MSE is calculated with face frames only. We can see that the baseline CNN achieves low MSE on the training set and high MSE on the validation set. From the table we conclude that the baseline CNN learns useful information for affective level prediction on the training set. Thus the features extracted from the baseline CNN are discriminative for affective level prediction.

	Train MSE	Validation MSE
CNN	0.0218	0.1067

Table 2. MSEs for the baseline CNN model.

3.2.2 BDLSTM on Top of CNN

To ameliorate the over-fitting in base CNN, we use a BDLSTM to model the temporal information on top of the base CNN. We use the decapitated baseline CNN model to extract face features. Each extracted face feature is a 512 dimensional vector. The BDLSTM model is built based on the sequence of the face feature vectors.

To start with, we use a BDLSTM with memory cell size $C = 250$ and input length $L = 11$. The results are shown in Table 3. We can see that the gap between the training and validation MSE becomes smaller. However, the gap is still considerably large. We use the pre-trained model without fine-tuning, which is trained on FER2013 dataset for emotion recognition task, to extract another set of 512 dimensional face feature vectors. Based on this set of face features, the BDLSTM model gets the performance as shown in the second row (labeled as BDLSTM No FT). Here the training and validation MSE is quite close, but both are quite high. Under this case, the BDLSTM fails to learn useful information from the second set of 512 dimensional features. Then we concatenate the two 512 dimensional face feature vectors to form a long 1024 dimensional face feature vector, based on which the BDLSTM achieves the performance in the third row (labeled as BDLSTM FT+No FT). The final result is better than the original BDLSTM trained with the fine-tuned CNN feature. The full results of the last model are shown in Table 4.

	Train MSE	Validation MSE
BDLSTM	0.03423	0.09842
BDLSTM No FT	0.098882	0.107055
BDLSTM FT+No FT	0.025454	0.098333

Table 3. MSEs for BDLSTMs with different input deep face features.

	Val. MSE	Aro. MSE	Val. CCC	Aro. CCC
BDLSTM-FACE	0.111205	0.085461	0.156441	0.290109
BDLSTM-ALL	0.117331	0.085551	0.173021	0.305441

Table 4. Full results for BDLSTM. FACE means the results are calculated on the frames with face, and ALL denotes the results are calculated on all the frames. Val. MSE means the mean square error for valence and Aro. CCC means the concordance correlation coefficient for arousal.

3.2.3 Multiple BDLSTMs

Then we vary the settings to test different BDLSTMs with the same set of 512 dimensional features extracted from the base CNN model. We reduce the memory cell from 250 to 50, 25 and 16, and reduce the input length from 11 to 5. The training and validation MSE for these settings are reported in Table 5. We can see that some BDLSTM with small memory cell size achieves better MSE on the validation set and worse MSE on the training set. We can see with smaller memory size, the over-fitting problem is partially addressed in that the training performance decreases while the validation performance increases.

	Train MSE	Validation MSE
BDLSTM C=250, L=11	0.03423	0.09842
BDLSTM C=50, L=11	0.03003	0.096982
BDLSTM C=50, L=5	0.02732	0.101284
BDLSTM C=25, L=5	0.03559	0.0964594
BDLSTM C=16, L=5	0.04305	0.098990

Table 5. MSEs for BDLSTMs with different cell memory sizes and input lengths.

3.2.4 Prediction Level Ensemble

For the various models above, we choose the models with validation MSE smaller than 0.1 and perform a prediction level ensemble to merge the results. The merged results are averaged over all the results from each of the chosen models. The performance of the merged results is shown in Table 6. In this table, the mean affective value, *i.e.* the mean of valence and arousal values, for the case of face frames (FACE) is 0.0956345, which is lower than all the models in previous subsections.

	Val. MSE	Aro. MSE	Val. CCC	Aro. CCC
P. Ensemble-FACE	0.106248	0.085021	0.135649	0.179434
P. Ensemble-ALL	0.112563	0.085950	0.160917	0.186296

Table 6. MSEs for prediction level ensemble.

3.2.5 Feature Level Ensemble

For the chosen model with validation MSE smaller than 0.1, we also perform a feature level ensemble to merge the results. We extract the features generated from the various BDLSTM models and fuse the features to further perform the task of affective level prediction. For some models, the length of the extracted feature is too long, *i.e.* the LSTM models with $C = 250$ and $L = 11$ will result in a feature length of $2CL$, or 5500. We exclude these models during the ensemble process. We train separate regression models based on fused features of all the models. Specifically, we train two multilayer perceptrons, one with 1 hidden layer of 32 units and the other with 2 hidden layers of 32 units. The results for the feature level ensemble are shown in Table 7.

	Val. MSE	Aro. MSE	Val. CCC	Aro. CCC
MLP1-FACE	0.112767	0.084258	0.209685	0.261834
MLP1-ALL	0.118834	0.084636	0.222690	0.270243
MLP2-FACE	0.111141	0.084448	0.166286	0.281198
MLP2-ALL	0.117270	0.084574	0.175925	0.292242

Table 7. MSEs for feature level ensemble.

Both the regression models achieve an MSE of lower than 0.1 on the validation set. A prediction level ensemble is performed for all the models in Section 3.2.4 and the two regression models above. The final results on validation set are shown in Table 8. In this table, the mean affective value for all the face frames is 0.095369, which is the lowest MSE achieved on the validation set. Compared with the the MSE of the baseline CNN model, 0.1067, the relative improvement is about 10%.

	Val. MSE	Aro. MSE	Val. CCC	Aro. CCC
F. Ensemble-FACE	0.106630	0.084108	0.149588	0.203219
F. Ensemble-ALL	0.112930	0.084873	0.171364	0.210785

Table 8. MSEs for the final ensemble results.

3.3. Testing Results

The results for the testing data are shown in Table 9. In the table the results of the single best model, prediction level ensemble and feature level ensemble are shown. The testing results are calculated over all the frames in videos. The corresponding validation results are also shown in the same table for comparison.

4. Conclusion and Future Directions

In this paper, we proposed a solution to the affect in the wild challenge. The proposed solution consists of a carefully designed deep face feature learner to learn discriminative features for affective levels and multiple memory

Test set	Val. MSE	Aro. MSE	Val. CCC	Aro. CCC
Single Model	0.134358	0.088	0.19635	0.21417
P. Ensemble	0.132	0.0887	0.1655	0.1938
F. Ensemble	0.132158	0.0881	0.177	0.2126

Val Set	Val. MSE	Aro. MSE	Val. CCC	Aro. CCC
Single Model-FACE	0.108081	0.084837	0.165066	0.194015
Single Model-ALL	0.114326	0.085927	0.186267	0.199779
P. Ensemble-FACE	0.106248	0.085021	0.135649	0.179434
P. Ensemble-ALL	0.112563	0.085950	0.160917	0.186296
F. Ensemble-FACE	0.106630	0.084108	0.149588	0.203219
F. Ensemble-ALL	0.112930	0.084873	0.171364	0.210785

Table 9. Testing results and the corresponding validation results.

networks for feature aggregation. Prediction level and feature level ensemble were shown to be effective in improving the performance of affective level prediction. The final model of the proposed approach outperforms the baseline CNN model by a factor of 10% on the validation set. One direction of further advancing the field of affective level estimation is a well-established dataset. As can be observed from Figure 1, the distribution of the valence and arousal values in the current dataset is very biased, even if the values are obtained by averaging the ratings from several human raters. A real world dataset, or a synthesized one, that has a nearly uniform distribution will be of great benefit in the field of affective level estimation.

Acknowledgment

This work was partially funded by National Research Foundation of Singapore. The work of Jiashi Feng was partially supported by National University of Singapore startup grant R-263-000-C08-133 and Ministry of Education of Singapore AcRF Tier One grant R-263-000-C21-112.

References

[1] Resnet training in torch, 2016. 3

[2] A. Dhall, R. Goecke, J. Joshi, J. Hoey, and T. Gedeon. Emotiw 2016: Video and group-level emotion recognition challenges. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 427–432. ACM, 2016. 1

[3] 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 *International Conference on Neural Information Processing*, pages 117–124. Springer, 2013. 5

[4] A. Graves, N. Jaitly, and A.-r. Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE, 2013. 1

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[6] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, et al. Emonets:multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 10(2):99–111, 2016. 1

[7] P. Khorrami, T. Paine, and T. Huang. Do deep neural networks learn facial action units when doing expression recognition? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 19–27, 2015. 1

[8] G. Levi and T. Hassner. Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In *Proc. ACM International Conference on Multimodal Interaction (ICMI)*, November 2015. 1

[9] J. Li, S. Roy, J. Feng, and T. Sim. Happiness level prediction with sequential inputs via multiple regressions. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 487–493. ACM, 2016. 1

[10] 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. 1

[11] 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. 3, 5

[12] M. Sundermeyer, R. Schlüter, and H. Ney. Lstm neural networks for language modeling. In *Interspeech*, pages 194–197, 2012. 1

[13] C. Yunpeng, J. Xiaojie, K. Bingyi, F. Jiashi, and Y. Shuicheng. Sharing residual units through collective tensor factorization in deep neural networks. *arXiv preprint arXiv:1703.02180*, 2017. 2

[14] S. Zafeiriou. Aff-wild: Valence and arousal in-the-wild challenge. 2017. 4

[15] S. Zaferiou. The menpo facial landmark localisation challenge. In *IEEE Conference on Computer Vision and Pattern Recognition-Workshops (CVPRW)*, volume 1, 2017. 4