Learning Spatio-Temporal Features with Partial Expression Sequences for on-the-Fly Prediction

Wissam J. Baddar, Yong Man Ro
Image and Video Systems Lab., Electrical Engineering,
KAIST, South Korea
{wisam.baddar,ymro}@kaist.ac.kr

Abstract

Spatio-temporal feature encoding is essential for encoding facial expression dynamics in video sequences. At test time, most spatio-temporal encoding methods assume that a temporally segmented sequence is fed to a learned model, which could require the prediction to wait until the full sequence is available to an auxiliary task that performs the temporal segmentation. This causes a delay in predicting the expression. In an interactive setting, such as affective interactive agents, such delay in the prediction could not be tolerated. Therefore, training a model that can accurately predict the facial expression ”on-the-fly” (as they are fed to the system) is essential. In this paper, we propose a new spatio-temporal feature learning method, which would allow prediction with partial sequences. As such, the prediction could be performed on-the-fly. The proposed method utilizes an estimated expression intensity to generate dense labels, which are used to regulate the prediction model training with a novel objective function. As results, the learned spatio-temporal features can robustly predict the expression with partial (incomplete) expression sequences, on-the-fly. Experimental results showed that the proposed method achieved higher recognition rates compared to the state-of-the-art methods on both datasets. More importantly, the results verified that the proposed method improved the prediction frames with partial expression sequence inputs.

Introduction

Facial expressions are a channel in which humans use to non-verbally convey and communicate their internal states, emotions and intentions (Littlewort et al. 2011). Recently, recognizing and interpreting facial expressions have attracted researchers in computer vision, affective computing and human computer interaction fields (Lee, Baddar, and Ro 2016). This is mainly attributed to the multitude of potential applications, such as emotionally intelligent interactive agents, fatigue measurement or even lie detection (Ekman and Rosenberg 1997).

The recent success of deep learning in various computer vision tasks has influenced researchers to investigate the utilization of deep learning techniques in facial expression recognition (FER) (Liu et al. 2014; 2013; 2015; Jung et al. 2015; Khorrami, Paine, and Huang 2015; Kim et al. 2016). A FER method influenced by the concept of action units has been introduced in (Liu et al. 2013; 2015). Spatial features were learned from local textural patterns, namely ”micro-action-patterns”, using a convolutional neural network (CNN). The learned spatial features were then used to classify the facial expression (Liu et al. 2013; 2015). In (Mollahosseini, Chan, and Mahoor 2015), the authors proposed adopting deeper CNN models by utilizing the concept of inception layers, to improve the FER performance in expressive face images. However, those methods relied only on the spatial features, and the facial expression dynamics were not utilized, which could limit the model performance at non-apex frames or frames of subtle expression.

Inspired by the dynamic nature of facial expressions (Jiang, Valstar, and Pantic 2011), researchers investigated spatio-temporal features to capture the dynamics of facial expressions in video sequences. In (Jung et al. 2015), appearance and geometric features were learned from expressive face sequences through a fusion of a 3D CNN and a multi-layer perceptron. The authors in (Liu et al. 2014), proposed using a 3D CNN to learn spatio-temporal features from deformable facial action parts. The large number of parameters (weights and biases) in 3D CNNs that needs to be learned and the large computational complexity makes the learning processes challenging (Prasoon et al. 2013; Roth et al. 2014) and increases the prediction time. To reduce the complexity, sampling a small number of frames or temporal normalization is usually used. This could cause a loss of information in the facial expression dynamics (Kim et al. 2017). Added to that, the aforementioned methods require the full sequence to be available at the prediction time in order to work efficiently, and cannot operate on-the-fly.

Recurrent neural networks (RNNs), in particular long short-term memory (LSTM) have been popular in encoding temporal and spatio-temporal features (Goodfellow, Bengio, and Courville 2016). They can provide a solid framework for on-the-fly prediction as described in (Eyben et al. 2010; Ringeval et al. 2015). However, when training the networks in (Eyben et al. 2010; Ringeval et al. 2015), the authors utilized dense labeled data (frame level labeling), which improves the prediction in an on-the-fly setting. On the other hand, The authors in (Kim et al. 2017) utilized LSTMs to encode the spatio-temporal dynamics of facial expressions from ”weakly” labeled sequences (video level labels). This improved the prediction at pre-segmented sequences but did the authors did not verify their method in an on-the-fly (interactive) scenario. Our experiments show, that relying on the weakly labeled data only, a delay in the prediction of the expression could occur (refer to Figure 4).

In this work, we proposed a method for learning spatio-temporal features for on-the-fly prediction with partial (incomplete) sequences. The proposed method facilitates the LSTMs on-the-fly prediction framework, and improves the partial prediction, such that the prediction delay problem can be resolved. To the best of our knowledge, this is the first work that estimates expression intensities, and utilizes them as dense labels to improve the expression prediction on-the-fly. As such, features can be obtained from partial (incomplete) sequences as they are fed to the model. The contributions of this paper are summarized as follows:

1. Obtaining densely labeled (frame level labeled) video sequences could be very costly and burdensome. Especially, in areas like FER, where labeling each frame can be subjective and time consuming [20]. On the other hand, weakly labeled data (video level labeled) is much easier to obtain. In this paper, we propose roughly generating dense labels (frame level labels) by estimating the expression intensity. The estimated expression intensities can be used in order to regulate the spatio-temporal feature training, such that prediction can be performed on-the-fly, even before the full sequence is available (using partial sequences).
2. We devise a method for learning spatio-temporal features, which can be readily used for prediction on-the-fly with partial sequences. To overcome the prediction delay problem and improve the expression prediction with partial (incomplete) expression sequences, the estimated expression intensities are used to regularize the features learning with a novel objective function construed of three objective terms. The first objective term minimizes the expression sequence classification error. The other two objective terms utilize the estimated expression intensity, and cluster the spatio-temporal feature to induce early prediction with partial sequences. As a result, the proposed method can perform prediction with partial (incomplete) sequences on-the-fly.

Proposed Method

Figure 1 shows an overview of the proposed method. The proposed spatio-temporal feature learning consists of facial expression intensity estimation and feature learning with estimated expression intensity constraints. In the facial expression intensity estimation part, the trained CNN model (Lee and Ro 2015) is used to obtain the spatial features from each frame in the training sequences. Using the obtained features, the expression intensity of each frame is estimated with respect to the apex frame. The estimated expression intensities are used as dense labels when learning the spatio-temporal features via the LSTM.

The spatio-temporal feature learning part focuses on encoding the facial expression dynamics, by learning spatio-temporal features via a regularized LSTM. Three objective terms are devised to guide the LSTM training. The objective terms are: (1) minimizing expression sequence classification error, (2) minimizing the expression intensity prediction error and (3) clustering spatio-temporal features obtained at each frame, by the LSTM, into perceived expression clusters. As a result, the proposed method can generate discriminative spatio-temporal features that improve the recognition of facial expression at the sequence level. Moreover, due to utilizing the expression intensity constraints, the LSTM can correctly predict the facial expressions at intermediate non-apex frames (with partial expression sequences). This allows the proposed method to be executed without requiring auxiliary tasks (such as expression transition detection, and expression segment detection (Ding et al. 2016)) or delays until the prediction is valid. We detail the proposed spatio-temporal feature learning in the following subsections.

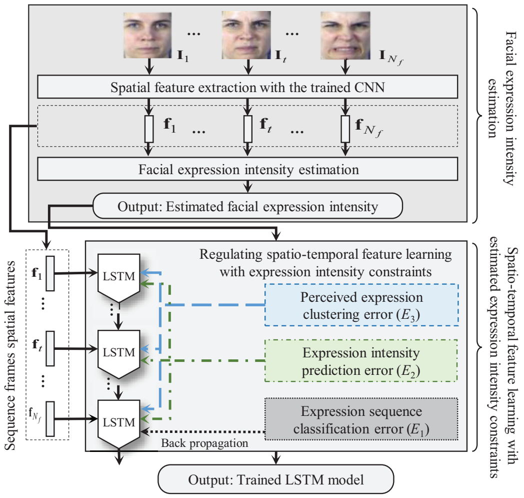

$\mathbf{I}_1$
$\mathbf{I}_t$
$\mathbf{I}_{N_f}$
Spatial feature extraction with the trained CNN
$\mathbf{f}_1$
$\mathbf{f}_t$
$\mathbf{f}_{N_f}$
Facial expression intensity estimation
Output: Estimated facial expression intensity
Facial expression intensity estimation
Sequence frames spatial features
LSTM
Regulating spatio-temporal feature learning with expression intensity constraints
Perceived expression clustering error (E_3)
Expression intensity prediction error (E_2)
Expression sequence classification error (E_1)
Back propagation
Output: Trained LSTM model
Spatio-temporal feature learning with estimated expression intensity constraints

Figure 1: Overview of the proposed method for learning spatio-temporal features with partial expression sequences for on-the-fly prediction.

Facial expression intensity estimation for regulating spatio-temporal feature learning

In facial expression video sequence datasets, it is often the case that one label is given to a video sequence. However, the frames of video sequences change from a neutral expression frame to apex frame (peak expression) (Zhao et al. 2011). Sometimes the sequence ends at a neutral expression frame (Pantic et al. 2005). In each video sequence, the time required for expression transition (e.g., onset to apex) is different. And expression intensity varies at each frame. This makes it difficult to manually label each frame with expression intensity or give hard labels that divide the expression sequence frames.

In this paper, we propose estimating the expression intensity at each frame with respect to the apex frame of the expression sequence. Spatial features are obtained from each cropped face region using CNN (Lee and Ro 2015). Then, the cosine similarity is calculated between the spatial feature of the current frame and the apex frame by:

Table 1: The network architecture for learning spatio-temporal features with partial expression sequences for on-the-fly prediction. Note that N_f is the number of frames in a sequence and N_c is the number of expression classes. *F and L are abbreviations for the fully connected layer and the LSTM layer, respectively. **F_{CNN} is the frame spatial feature, which is the last fully connected layer of the CNN network (Kim et al. 2017).

Type	Layer*	Input shape	Output shape
Input	Input	$N_f \times (64 \times 64 \times 3)$	$N_f \times (64 \times 64 \times 3)$
Spatial features	F^{**}_{CNN}	$N_f \times (64 \times 64 \times 3)$	$N_f \times (512 \times 1)$
Spatio-temporal features	L1 Ff Fc	$N_f \times (512 \times 1)$ $N_f \times (512 \times 1)$ $N_f \times (512 \times 1)$	$N_f \times (512 \times 1)$ $N_f \times 1$ $N_c \times 1$

$$cossimilarity = \frac{\mathbf{f}_t \cdot \mathbf{f}_{apex}}{\|\mathbf{f}_t\| \, \|\mathbf{f}_{apex}\|}, \tag{1}$$

where $\mathbf{f}_t$ is the spatial feature vector obtained from the frame at time t, $\mathbf{f}_{apex}$ is the spatial feature vector obtained from the apex frame and $\|\cdot\|$ denotes the L2 norm operation. After the cosine similarity is obtained for all the frames of the sequence, min-max normalization is performed on the cosine similarities. As a result, the expression intensity at the apex frame is set to 1, and the expression intensity of the frame with a neutral face is set to 0. In case of a neutral expression sequence, all expression intensities are assumed to be zero.

Spatio-temporal feature learning with estimated expression intensity constraints

Table 1 summarizes the overall architecture of the proposed network. To learn the spatio-temporal features with estimated expression intensity constraints, the spatial features are extracted from the sequence frames via the learned CNN model and fed to the LSTM. An objective function with three objective terms is devised to learn the LSTM network parameters.

The first objective term of the objective function (E_1) is devised for minimizing the expression sequence classification error at the expression output layer of the network (F_c in Table 1). To that end, a cross-entropy error (Golik, Doetsch, and Ney 2013) is utilized to enforce learning a discriminative feature, which is defined as:

$$E_1 = -\sum_{i,c} y_{i,c} \text{log} \widehat{y}^{(F_c)}_{i,c}, \tag{2}$$

where c is the expression class index,$y_{i,c}$ is the expression class ground truth of the i-th sequence (1 if c is the correct class and 0 otherwise), and $\widehat{y}^{(F_c)}_{i,c}$ is the predicted probability that the sequence belongs to the class c calculated at the output layer (F_c in Table 1).

To perform back-propagation, the gradients of the objective term E_1 for the layer F_c are written as follows:

$$\begin{gathered} \frac{\partial E_1}{\partial \mathbf{b}^{(F_c)}} = \sum_i \left(\widehat{y}^{(F_c)}_i - y^{(F_c)}_i \right) \\ \text{and} \frac{\partial E_1}{\partial \mathbf{W}^{(F_c)}} = \mathbf{h}^{(L_1)} \left(\frac{\partial E_1}{\partial \mathbf{b}^{(F_c)}} \right)^T, \end{gathered} \tag{3}$$

where $\mathbf{b}^{(F_c)}$ and $\mathbf{W}^{(F_c)}$ denote the biases and weights of the layer F_c, respectively.$\mathbf{h}^{(L_1)}$ denotes the output of the LSTM layer (L_1 in Table 1), $\mathbf{y}_i = [y_{i,1}, y_{i,2}, ..., y_{i,c}, ...y_{i,N_c}]^T$ is the expression ground truth vector of the $i-th$ training sample,$y_{i,c}$ if and only if the true class of the sample is c, and $\widehat{y}^{(F_c)}_i$ denotes the predicted probability calculated at the layer F_c with a Softmax function. As shown in 3, the gradient with respect to the weight is dependent on the gradient with respect to the bias. Therefore, the gradients with respect to the bias are sufficient to be described in the objective term. The gradients of the other layers can also be computed by the same back-propagation algorithm (Bishop 1995).

The cross-entropy loss of the objective term for minimizing the expression sequence classification error (E_1) is calculated at the end of a sequence. As a result, the optimization highly affects the spatio-temporal feature (L_1 output from Table 1) at the last frame of sequence (N_f-th frame). However, the expression intensity variations of expression sequences result in different spatio-temporal features ($_1$ output) at each frame. Hence, miss-classification could occur when prediction is performed at early stages of the sequence (at non-apex frames with a partial expression sequence input). The prediction of the correct expression could be delayed until the end of the sequence (apex, or near apex frames). To mitigate the aforementioned delay problem, we propose the second objective term (E_2), which minimizes expression intensity prediction error. To that end, Euclidian loss is used to enforce learning discriminative features at each frame, which is defined as:

$$E_2 = \frac{1}{2} \sum_{i,c} \left\| \widehat{\mathbf{L}}^{(F_f)}_{c,i} - \mathbf{L}_{c,i} \right\|^2_2, \tag{4}$$

where $\mathbf{L}_{c,i}$ is a vector of expression intensity labels for the $i-th$ sequence of the $c-th$ class (calculated by the estimation method described in section 2.1), and $\widehat{\mathbf{L}}^{F_f}_{c,i}$ is a vector of the predicted expression intensities at each frame.

The gradient of the second objective term (E_2) in 4, with respect to the bias of the last LSTM layer (L_1 in Table 1) can be written as:

$$\begin{gathered} \frac{\partial E_2}{\partial \mathbf{b}^{(L_1)}} = \sum_{c,p,i} g'(e_2) \left(\widehat{\mathbf{L}}^{(F_f)}_{c,i} - \mathbf{L}_{c,i} \right) \circ \sigma'(\mathbf{h}^{(L_1)}), \\ \text{where } e_2 = \left\| \mathbf{L}_{c,i} - \widehat{\mathbf{L}}^{(F_f)}_{c,i} \right\|^2_2, \end{gathered} \tag{5}$$

where '$\circ$' denotes the Hadamard product,$\mathbf{b}^{(L_1)}$ is the bias of the LSTM layer (L_1 in Table 1), $\sigma(.)$ is the activation function, and $\mathbf{h}^{(L_1)} = \mathbf{W}^{(l)}_o [\mathbf{h}^{(l)}_{t-1}, \mathbf{h}^{(l-1)}_t] + \mathbf{b}^{(l)}_o]$ (Hochreiter and Schmidhuber 1997).

When expression sequences transition from a neutral state to the apex. Humans would perceive the first number of frames as neutral frames, and start perceiving the expression at later stage. Hence, the learned spatio-temporal features are supposed to predict neutral expression at the beginning of the sequence, and predict the expression at later frames. To improve the perceived expression prediction at early stages of the expression sequence (prediction with partial expression sequence inputs) a third objective term (E_3) is devised. For each expression, the spatio-temporal features from the LSTM (L_1 output in Table 1) are obtained from all frames of all the expression sequences of that class. Then, the features for certain class (e.g., feature from smile sequences) are clustered into 2 classes, namely, perceived neutral and perceived expression class (e.g., smile, disgust, etc.). K-means clustering (Duda, Hart, and Stork 2012) is utilized for the clustering. Figure 2 (left) illustrates an example of sequences after k-means clustering. As shown in the figure, the number of frames perceived as neutral or expressive class, can be different from sequence to sequence. However, because the k-means cluttering is non-supervised, the perceived expressive frames features and the perceived neutral frames features can be obtained automatically. Note that in the case of neutral sequences, all frames are assigned to the perceived neutral expression.

To induce early prediction at non-apex frames (when the prediction is performed with partial expression sequence inputs), E_3 clusters the spatio-temporal features, obtained from the sequence frames, into perceived expression clusters (i.e., frames perceived as neutral and frames perceived as the expression class). It minimizes the intra-class variation due to expression intensity variations or subject appearance variations. The objective term E_3 is illustrated in Figure 2 (right). It can be written as:

$$E_3 = \frac{1}{2} \sum_{c,p,i} g\left(\|\mathbf{f}_{c,p,i} - \mathbf{m}_c\|_2^2 - d_c^2 \right), \tag{6}$$

where $\mathbf{f}_{c,p,i}$ is the spatio-temporal feature of the i-th expression sequence of class c and obtained from LSTM layer (L1 in Table 1) at the p-th frame of the sequence. $\mathbf{m}_c$ is the mean feature vector of spatio-temporal features with the perceived class c. d_c is half the distance between $\mathbf{m}_c$ and $\mathbf{m}_0$, where $\mathbf{m}_0$ is the mean of the perceived neutral frames. The function $g(\omega)$ is a smoothed approximation of $[\omega]_+ = \max(0, \omega)$ defined as (Mignon and Jurie 2012):

$$g(\omega) = \frac{1}{\beta} log\left(1 + \exp(\beta\omega)\right), \tag{7}$$

where β is a sharpness parameter.

For simplicity when obtaining the gradient of the third objective term (E_3) in (7), the mean feature vector $\mathbf{m}_c$ is assumed to be a constant vector. Thus the gradient of E_3 with respect to the bias of the LSTM layer (L_1 in Table 1) can be written as:

$$\frac{\partial E_3}{\partial \mathbf{b}^{(L_1)}} = \sum_{c,p,i} g'(e_3)\left(\mathbf{f}_{c,p,i} - \mathbf{m}_c\right) \circ \sigma'(\mathbf{h}^{(L_1)}), \quad \text{where } e_3 = \|\mathbf{m}_c - \mathbf{f}_{c,p,i}\|_2^2 - (d_c)^2. \tag{8}$$

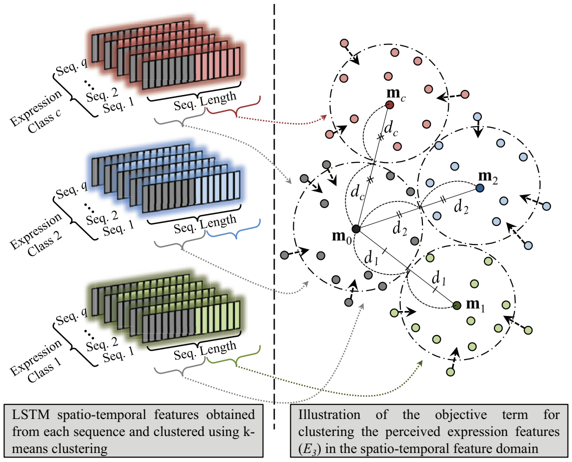


Figure 2: Illustration of the objective term for clustering the spatio-temporal features (L_1 output from Table 1) obtained at each frame. The features are clustered into perceived expression clusters (i.e., frames perceived as neutral and frames perceived as the expression class). The left part of the figure illustrates the LSTM features obtained at each frame in expression sequences. The features are clustered (using the k-means clustering) as a perceived expressive class. The features clustered as perceived expression classes are shown in color (red, blue and green). The features clustered as neutral (perceived neutral frames) are shown in dark gray. The right side of the figure illustrates the perceived expression clusters in the feature space. Each dot represents the perceived expression class. The circle enclosing the colored dots represents the cluster of a certain expression class. (best viewed in color)

Training the LSTM with the proposed objective terms

The training process was performed in three steps detailed in Algorithm 1. The training starts with the spatial features obtained from each frame in the training sequences, expression sequence labels and the estimated expression intensities. For each epoch, the network is first trained with the objective terms E_1 and E_2. In this paper, we assumed that both objective terms contribute equally to updating the LSTM weights (i.e., each loss was given the same weight). In the second training step, features are obtained from all frames of all the sequence of the training set. K-means clustering is then utilized to obtain the mean of perceived expressions in each expression class (see Figure 2). Finally, the obtained mean and distance values are fixed and utilized to update the LSTM weights according to objective term E_3.

Experiments

Experimental setup

To verify the effectiveness of the proposed method, experiments were conducted on two datasets. The face region

Algorithm 1: Pseudo code for training the LSTM with the proposed objective terms

```
Input: X: Spatial features from expression sequences
       Y:the facial expression sequences labels
       L: the estimated expression intensities
Output: Trained LSTM model
LSTM parameter initialization;
while epoch $<N_{epochs}$ do
    while batch $<M_{batches}$ do
        Feedforward ($\mathbf{X}$(batch),$\mathbf{Y}$(batch),$\mathbf{L}$(batch));
        Get gradient for $E_1 + E_2$ & update weights
    end
    for expression class in (1:C) do
        $\mathbf{f}_c$ = Get class sequences features;
    end
    for expression class in (1:C) do
        $\mathbf{m}_c$ = K-means($\mathbf{f}_c$ , K = 2);
        $\mathbf{d}_c$ = Calculate distances;
    end
    while batch $<M_{batches}$ do
        Feedforward ($\mathbf{X}$(batch),$\mathbf{m}_c$(batch),$\mathbf{d}_c$(batch));
        Get gradient for $E_3$ & update weights
    end
end
```

was detected and facial landmark detection was performed (Asthana et al. 2014) on each frame. The face region was automatically cropped and aligned based on the eye landmarks (Tian 2004). The construction of the utilized MMI and Oulu-CASIA datasets was performed as follows:

1. MMI dataset (Pantic et al. 2005): A total of 205 expression sequences with frontal faces were collected from 30 subjects. Each expression sequence was labeled with one of the six basic expressions classes (i.e., angry, disgust, fear, happy, sad, and surprise). In the MMI dataset, an expression sequence was recorded from onset to apex to offset. The indexes of the apex frames were located manually (Lee and Ro 2015). To unify the experiments on both datasets (MMI and Oulu CASIA), only onset to apex frames were used. For a practical FER system, neutral sequences are required to be included during the training. To that end, neutral sequences of the subject were simulated by manually collecting frames from the beginning and the end of the expressive sequences. As such, sequences of cropped neutral expression faces were generated. As total of 38 natural expression sequences were generated and added the total to the MMI sequences.

2. Oulu-CASIA dataset (Zhao et al. 2011): Sequences of the six basic expressions were collected from 80 subjects under three illumination conditions. For the experiments, a total of 480 image sequences were collected from sequences captured with a visible light camera under normal illumination conditions. For each subject, the basic expression sequence was captured from a neutral face until the expression apex. Similar to the MMI dataset, 80 neutral expression sequences were simulated by manually collecting natural frames from different sequences of the same subject.

All the experiments in this paper were conducted in a subject independent manner, such that the subjects in the training set were excluded from the test set. In particular, experiments on the MMI dataset set were performed with a leave-one-subject-out (LOSO) cross validation (Lee, Baddar, and Ro 2016; Liu et al. 2014; Kim et al. 2017; Lee et al. 2014), while 10-fold cross validation (Jung et al. 2015) was used for the experiments conducted on the Oulu-CASIA dataset.

To avoid overfitting, due to the limited number of samples in the utilized datasets (MMI and Oulu-CASIA), data augmentation was performed during the network training (Khorrami, Paine, and Huang 2015; Kim et al. 2017). For the CNN training, 54 augmentation variations of each expressive image were obtained by: (1) horizontal flipping of the sequence frames, (2) rotating the frames between the angles $[-5^\circ, 5^\circ]$ with an increment of 1°, (3) translating the frames along $[\pm 3, \pm 3]$ pixels in the x and y axis with 1 pixel increments, and (4) scaling the frames with scaling factors of 0.90, 0.95, 1.05 and 1.10.

For training the LSTM, each frame in the sequence was augmented similar to the aforementioned augmentation processes for training the CNN. From each augmented sequence two temporal augmentations were performed by selecting even and odd frames from each sequence. As a result, 108 augmentations of the expression sequences were generated and used to train the LSTM network.

The learning and implementation of the CNN and LSTM network (shown in Table 1) was done using TensorFlow. For the activation function, rectified linear unit (ReLU) was used in all layers except the layer FCNN, in which sigmoid activation was utilized in order to bound the LSTM input features and insure the LSTM learning stability. In this paper, CNN initial learning rate was set to 0.0001 for both the MMI and the Oulu-CASIA datasets, and the training was performed for 30 epochs. For the LSTM, the learning rate was set to 0.0001, and the learning rate was reduced by a factor of 10 every 10 epochs. The LSTM training was conducted for 50 epochs.

Effectiveness of the proposed method compared with previous state-of-the-art and existing methods on sequence-level prediction

To demonstrate the effectiveness of the proposed method, the FER performance of the proposed method was compared to previously reported state-of-the-art and existing methods (Liu et al. 2014; 2013; 2015; Jung et al. 2015). The experiment was conducted under LOSO cross validation. From previously reported methods, static (frame-based) features (i.e., AURF (Liu et al. 2013) and AUDN (Liu et al. 2015)) were evaluated at the apex frames. Spatio-temporal feature based methods (i.e., 3D CNN-DAP (Liu et al. 2014), DTAGN (Jung et al. 2015) and CNN+LSTM (Kim et al. 2017)) were evaluated at the sequence level. The comparative recognition rates are shown in Table 2. As shown in the table, the proposed method outperformed existing state-of-the-art FER methods. Specifically, the proposed method showed better recognition rates compared to the deep learning based methods with spatial features (AURF and AUDN).

Table 2: FER performance comparisons with existing FER methods on the MMI dataset in terms of recognition rate.*6 expression classes represent the basic expressions (i.e., angry, disgust, fear, happy, sad, and surprise).*7 expression classes represent the basic expressions + neutral

Method	Expression Classes*	Input	Recognition rate(%)
3D CNN-DAP (Liu et al. 2014)	6	sequence	63.4
DTAGN (Jung et al. 2015)	6	sequence	70.24
AURF (Liu et al. 2013)	7	static	69.88
AUDN (Liu et al. 2015)	7	static	75.85
CNN+LSTM (Kim et al. 2017)	6	sequence	78.61
CNN+LSTM (Kim et al. 2017)	7	sequence	76.64
Proposed method($E_1 + E_2$)	7	sequence	**78.29**
Proposed method($E_1+E_2+E_3$)	7	sequence	**78.96**

Table 3: FER performance comparisons with existing FER methods on the Oulu-CASIA dataset in terms of recognition rate. Note that all these methods use sequences as input.

Method	Expression Classes*	Recognition rate(%)
LBP-TOP (Zhao and Pietikainen 2007)	6	68.13
HOG 3D (Klaser, Marszaek, and Schmid 2008)	6	70.63
AdaLBP (Zhao et al. 2011)	6	73.54
Atlases (Guo, Zhao, and Pietikinen 2012)	6	75.52
Dis-ExpLet (Liu et al. 2016)	6	76.65
DTAGN (Jung et al. 2015)	6	81.46
CNN+LSTM (Kim et al. 2017)	7	78.21
Proposed method($E_1 + E_2$)	7	**82.15**
Proposed method($E_1 + E_2 + E_3$)	7	**82.86**

This can be attributed to the efficient encoding of the expression dynamics in the proposed spatio-temporal features learning. Moreover, by utilizing the estimated expression intensity in the proposed objective function, the proposed method achieved a superior performance to methods that encoded the spatio-temporal dynamics with a 3D CNN (3D CNN-DAP, DTAGN and CNN+LSTM).

Table 3 shows the recognition rate comparisons with state-of-the-art spatio-temporal feature based methods on the Oulu-CASIA dataset. The experiment was conducted with 10-fold subject-independent cross validation. As shown in the table, the results indicate that the proposed method is more effective than previous state-of-the-art spatio-temporal feature based methods.

Evaluating the proposed method for on-the-fly prediction with partial sequence inputs

In this experiment, we investigate the effectiveness of the proposed objective terms in improving the prediction on-the-fly with partial sequence inputs. For each sequence, prediction was performed at each frame using the learned spatio-temporal features of the partial expression sequence extending from the beginning of sequence until the frame in question). After the prediction was performed on all frames of the sequences (i.e., all partial sequences included in that sequence), recognition rate was calculated.

As previously mentioned, each expression sequence is labeled with one expression, but the sequence intensity changes from zero (at onset) to one at the apex frame. Hence, generating labels at the frame-level is necessary to evaluate the recognition rate at different expression intensities. To that end, the expression intensity was utilized to generate frame-level labels as shown in Figure 3. With a given expression intensity threshold (for example expression intensity = 0.2, as shown in the figure), all frames with expression intensity above that threshold are assumed as expressive images of that label (shown in green). On the other hand, all frames below that expression intensity threshold are considered as perceived natural frames (shown in pink). Accordingly, the recognition rate at that intensity threshold was calculated.

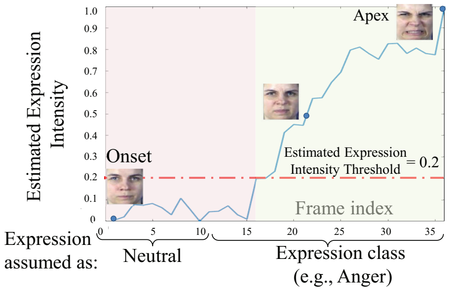


Figure 3: Generating frame level (dense) labels at different expression intensities to validate the FER performance at different facial expression intensities.

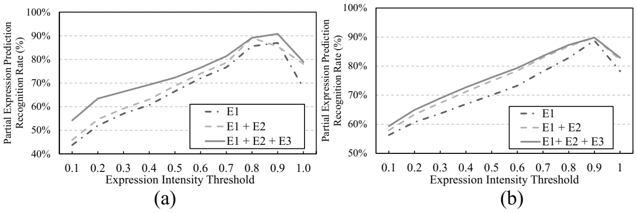
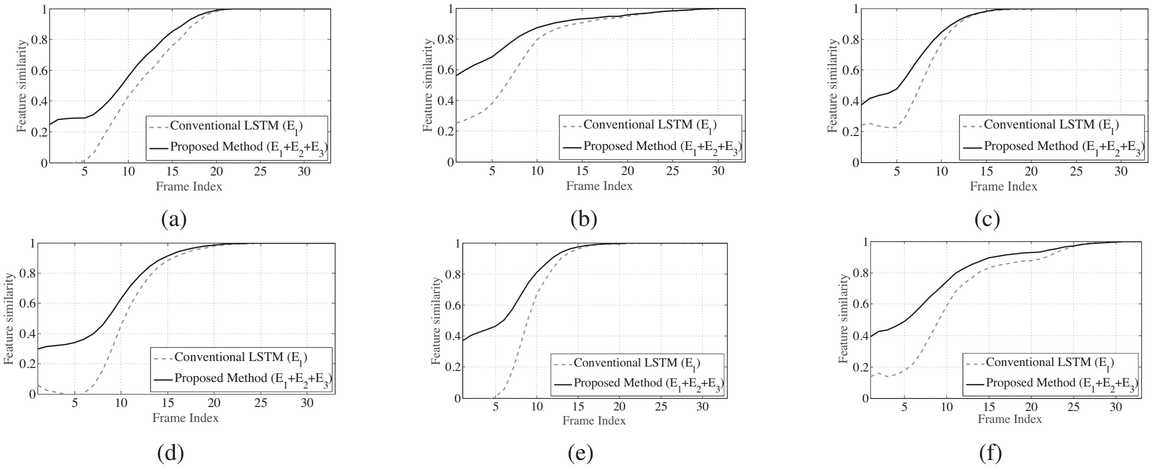


Figure 4: FER performance at different expression intensity thresholds. (a) MMI Dataset (b) Oulu Dataset.

Figure 4 shows a comparison between the obtained recognition rates for three different models. The first model was trained with a conventional LSTM (i.e. using the objective term E_1 only). The other two models were trained by incrementally adding the objective terms (i.e., one model was trained with $E_1 + E_2$, and $E_1 + E_2 + E_3$, respectively). As shown in the figure, the proposed objective terms significantly improve the prediction performance compared to the conventional LSTM method in terms of the recognition rate. More importantly, it can be seen that most of the improvement occurs in expression intensity range [0.3, 0.7]. These expression intensities can be considered subtle non-apex frames. It should also be noted that the prediction was only performed using a partial expression sequence (which extends from the beginning of the expression until that frame). This means that the proposed method can be readily applied to on-the-fly (interactive) applications. As such, the prediction can be performed after each frame is fed to the system, by considering the LSTM state of previous frames.



(a) (b) (c)

(d) (e) (f)

Figure 5: Effect of the proposed method on the feature simiarity between intermediate partial sequence features and the full sequence features. The figure shows the average cosine similarity values between the obtained features at each frame on-the-fly (partial sequence inputs) and the apex frame features (full sequence). Each figure represents a different expression: (a) anger (b) disgust, (c) fear, (d) happiness, (e) sadness and (f) surprise.

Notice the performance degradation at the expression intensity threshold = 1. At that point the prediction is performed with a full sequence. The number of expression sequences is significantly smaller than the number of frames, which justifies the resulting in the degradation in the prediction shown in the Figure4.

Effect of the proposed method on the learned spatio-temporal features

In this experiment, Oulu-CASIA dataset was utilized to analyze the effect of the proposed method on the learned features. To improve the prediction with partial sequences, the proposed objective function should increase the feature similarity between intermediate frame features (with partial sequence inputs) and the apex frame (with the full sequence). To validate that assumption, cosine similarity was calculated between the spatio-temporal features (L_1 output from Table 1) obtained from each frame and the frame obtained at the last frame (apex frame). Note that the features were obtained by feeding the LSTM the frames leading to the frame in question, to mimic the on-the-fly prediction process. The average of the features similarities from all the sequences of the same expression class are plotted in Figure 5. As the figure shows, the proposed method increases the features similarity between the intermediate frames and the apex frame compared to the conventional LSTM. This verifies that the learned model can predict the expression more accurately at earlier frames, alleviating the prediction delay problem.

Conclusion

In this paper, we proposed a new spatio-temporal feature learning method, which allow prediction with partial sequences on-the-fly. To achieve that, we proposed utilizing an estimated expression intensity to generate frame-level (dense) labels, which can be used to regulate the training of an LSTM with a novel objective function. The purpose of the proposed objective function is to generate features that could robustly predict the expression with partial (incomplete) expression sequences, on-the-fly. This can solve the delay problem that could occur when using an LSTM on-the-fly prediction framework, with weakly labeled data (one label per expression sequence). Comprehensive experiments were conducted on two facial expression datasets. The experimental results showed that the proposed method improved the FER performance on sequence level prediction. The results also verified that the proposed method improved the prediction with partial expression sequence inputs at non-apex frames. These results indicate that the proposed method can solve the expression prediction delay problem and can be deployed in an on-the-fly scenario (i.e., interactive environments).

Acknowledgment

This work was supported by the Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No. 2017-0-01778, Development of Explainable Human-level Deep Machine Learning Inference Framework)

References

Asthana, A.; Zafeiriou, S.; Cheng, S.; and Pantic, M. 2014. Incremental face alignment in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, 1859–1866. IEEE.

Bishop, C. M. 1995. *Neural networks for pattern recognition*. Oxford university press.

Ding, X.; Chu, W.-S.; De la Torre, F.; Cohn, J. F.; and Wang, Q. 2016. Cascade of tasks for facial expression analysis. *Image and Vision Computing* 51:36–48.

Duda, R. O.; Hart, P. E.; and Stork, D. G. 2012. *Pattern classification*. John Wiley & Sons, second edition edition.

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

Eyben, F.; Wllmer, M.; Graves, A.; Schuller, B.; Douglas-Cowie, E.; and Cowie, R. 2010. On-line emotion recognition in a 3-d activation-valence-time continuum using acoustic and linguistic cues. *Journal on Multimodal User Interfaces* 3(1):7–19.

Golik, P.; Doetsch, P.; and Ney, H. 2013. Cross-entropy vs. squared error training: a theoretical and experimental comparison. In *INTERSPEECH*, 1756–1760.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. Deep learning. book in preparation for mit press. *URL¡ http://www. deeplearningbook. org*.

Guo, Y.; Zhao, G.; and Pietikinen, M. 2012. *Dynamic facial expression recognition using longitudinal facial expression atlases*. Springer. 631–644.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Jiang, B.; Valstar, M. F.; and Pantic, M. 2011. Action unit detection using sparse appearance descriptors in space-time video volumes. In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, 314–321. IEEE.

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

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

Kim, B.-K.; Roh, J.; Dong, S.-Y.; and Lee, S.-Y. 2016. Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. *Journal on Multimodal User Interfaces* 1–17.

Kim, D. H.; Baddar, W.; Jang, J.; and Ro, Y. M. 2017. Multi-objective based spatio-temporal feature representation learning robust to expression intensity variations for facial expression recognition. *IEEE Transactions on Affective Computing*.

Klaser, A.; Marszaek, M.; and Schmid, C. 2008. A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, 275: 1–10. British Machine Vision Association.

Lee, S. H., and Ro, Y. M. 2015. Partial matching of facial expression sequence using over-complete transition dictionary for emotion recognition. *transaction on affective computing*.

Lee, S. H.; Baddar, W. J.; and Ro, Y. M. 2016. Collaborative expression representation using peak expression and intra class variation face images for practical subject-independent emotion recognition in videos. *Pattern Recognition*.

Lee, S. H.; Plataniotis, K.; Konstantinos, N.; and Ro, Y. M. 2014. Intra-class variation reduction using training expression images for sparse representation based facial expression recognition. *Affective Computing, IEEE Transactions on* 5(3):340–351.

Littlewort, G.; Whitehill, J.; Wu, T.; Fasel, I.; Frank, M.; Movellan, J.; and Bartlett, M. 2011. The computer expression recognition toolbox (cert). In *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, 298–305. IEEE.

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

Liu, M.; Li, S.; Shan, S.; Wang, R.; and Chen, X. 2014. *Deeply learning deformable facial action parts model for dynamic expression analysis*. Springer. 143–157.

Liu, M.; Li, S.; Shan, S.; and Chen, X. 2015. Au-inspired deep networks for facial expression feature learning. *Neurocomputing* 159:126–136.

Liu, M.; Shan, S.; Wang, R.; and Chen, X. 2016. Learning expressionlets via universal manifold model for dynamic facial expression recognition. *IEEE Transactions on Image Processing* 25(12):5920–5932.

Mignon, A., and Jurie, F. 2012. Pcca: A new approach for distance learning from sparse pairwise constraints. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2666–2672. IEEE.

Mollahosseini, A.; Chan, D.; and Mahoor, M. H. 2015. Going deeper in facial expression recognition using deep neural networks. *arXiv preprint arXiv:1511.04110*.

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

Prasoon, A.; Petersen, K.; Igel, C.; Lauze, F.; Dam, E.; and Nielsen, M. 2013. *Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network*. Springer. 246–253.

Ringeval, F.; Eyben, F.; Kroupi, E.; Yuce, A.; Thiran, J.-P.; Ebrahimi, T.; Lalanne, D.; and Schuller, B. 2015. Prediction of asynchronous dimensional emotion ratings from audiovisual and physiological data. *Pattern Recognition Letters* 66:22–30.

Roth, H. R.; Lu, L.; Seff, A.; Cherry, K. M.; Hoffman, J.; Wang, S.; Liu, J.; Turkbey, E.; and Summers, R. M. 2014. *A new 2.5 D representation for lymph node detection using random sets of deep convolutional neural network observations*. Springer. 520–527.

Tian, Y.-l. 2004. Evaluation of face resolution for expression analysis. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on*, 82–82. IEEE.

Zhao, G., and Pietikainen, M. 2007. 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.

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