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, fatigue measurement or even lie detection (Ekman 1997), the utilization of deep learning techniques in facial expression recognition (FER) (Liu et al. 2014; 2013; 2015) has influenced researchers to investigate 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 expression 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.

Recent 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 not improve in pre-segmented sequences.
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 regulate 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.

**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:
where $\| \cdot \|$ sequence, all expression intensities are assumed to be zero. With a neutral face is set to 0. In case of a neutral expression frame is set to 1, and the expression intensity of the frame similarities. As a result, the expression intensity at the apex sequence, min-max normalization is performed on the cosine similarity is obtained for all the frames of the se-

<table>
<thead>
<tr>
<th>Type</th>
<th>Layer*</th>
<th>Input shape</th>
<th>Output shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td></td>
<td>$N_f \times (64\times64 \times 3)$</td>
<td>$N_f \times (64\times64 \times 3)$</td>
</tr>
<tr>
<td>Spatial features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$F_{CNN}$</td>
<td></td>
<td>$N_f \times (64\times64 \times 3)$</td>
<td>$N_f \times (512 \times 1)$</td>
</tr>
<tr>
<td>Spatio-temporal</td>
<td>$L1$</td>
<td>$N_f \times (512 \times 1)$</td>
<td>$N_f \times (512 \times 1)$</td>
</tr>
<tr>
<td></td>
<td>$F_f$</td>
<td>$N_f \times (512 \times 1)$</td>
<td>$N_f \times 1$</td>
</tr>
<tr>
<td></td>
<td>$F_c$</td>
<td>$N_c \times 1$</td>
<td></td>
</tr>
</tbody>
</table>

$cossimilarity = \frac{f_t \cdot f_{apex}}{\|f_t\| \|f_{apex}\|}$, (1)

where $f_t$ is the spatial feature vector obtained from the frame at time $t$, $f_{apex}$ 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:

$$E_1 = - \sum_{i,c} y_{i,c} \log \hat{g}_{i,c}^{(F_c)}$$

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 $\hat{g}_{i,c}^{(F_c)}$ 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:

$$\frac{\partial E_1}{\partial b^{(F_c)}} = \sum_i (\hat{g}_{i}^{(F_c)} - y_{i,c}^{(F_c)}) $$

and

$$\frac{\partial E_1}{\partial W^{(F_c)}} = h^{(L_1)} (\frac{\partial E_1}{\partial b^{(F_c)}})^T,$$ (3)

where $b^{(F_c)}$ and $W^{(F_c)}$ denote the biases and weights of the layer $F_c$, respectively $h^{(L_1)}$ denotes the output of the LSTM layer ($L_1$ in Table 1), $y_i = [y_{i,1}, y_{i,2}, ..., y_{i,c}, ..., y_{i,N_c}]^T$ is the expression ground truth vector of the $i$-th training sample, and $\hat{g}_{i}^{(F_c)}$ 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 ($L$ in Table 1) can be written as:

$$E_2 = \frac{1}{2} \sum_{i,c} \left\| \hat{\ell}_{c,i}^{(F_f)} - \ell_{c,i}^{(F_f)} \right\|_2^2,$$ (4)

where $\ell_{c,i}$ is a vector of expression intensity labels for the $c-th$ expression of the $i-th$ frame. However, the expression intensity variations of expression sequences result in different spatio-temporal features 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:

$$E_2 = \frac{1}{2} \sum_{c,i} \left\| \ell_{c,i}^{(F_f)} - \hat{\ell}_{c,i}^{(F_f)} \right\|_2^2$$

where $\ell_{c,i}$ is a vector of expression intensity labels for the $c-th$ sequence of the $c-th$ class (calculated by the estimation method described in section 2.1), and $\hat{\ell}_{c,i}^{(F_f)}$ 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:

$$\frac{\partial E_2}{\partial b^{(L_1)}} = \sum_{c,p,i} e_{2} (\hat{\ell}_{c,i}^{(F_f)} - \ell_{c,i}^{(F_f)}) \circ \sigma'(h^{(L_1)}),$$ (5)

where $e_{2} = \left\| \ell_{c,i}^{(F_f)} - \hat{\ell}_{c,i}^{(F_f)} \right\|_2^2$, where ‘$\circ$’ denotes the Hadamard product, $b^{(L_1)}$ is the bias of the LSTM layer ($L_1$ in Table 1), $\sigma(\cdot)$ is the activation function, and $h^{(L_1)} = W_o [h_{t-1}^o, h_t^o] + b_o^l$ (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 (L₁ 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 clustering 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:

\[
E_3 = \frac{1}{2} \sum_{c,p,i} g \left( \| f_{c,p,i} - m_c \|^2_2 - d_c^2 \right),
\]

where \( f_{c,p,i} \) 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. \( m_c \) is the mean feature vector of spatio-temporal features with the perceived class \( c \). \( d_c \) is half the distance between \( m_c \) and \( m_0 \), where \( m_0 \) is the mean of the perceived neutral frames. The function \( g(\omega) \) is a smoothed approximation of \( |\omega|^\beta = \max(0, \omega) \) defined as (Mignon and Jurie 2012):

\[
g(\omega) = \frac{1}{\beta} \log (1 + \exp(\beta \omega)),
\]

where \( \beta \) is a sharpness parameter.

For simplicity when obtaining the gradient of the third objective term \( E_3 \) in (7), the mean feature vector \( m_c \) is assumed to be a constant vector. Thus the gradient of \( E_3 \) with respect to the bias of the LSTM layer (L₁ in Table 1) can be written as:

\[
\frac{\partial E_3}{\partial b^{(L_1)}} = \sum_{c,p,i} g'(\epsilon_3) (f_{c,p,i} - m_c) \odot \sigma'(h^{(L_1)}),
\]

where \( \epsilon_3 = \| m_c - f_{c,p,i} \|^2_2 - (d_c)^2 \).

**Figure 2**: Illustration of the objective term for clustering the spatio-temporal features (L₁ 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
2. Oulu-CASIA dataset (Zhao et al. 2011): Sequences of the

1. MMI dataset (Pantic et al. 2005): A total of 205 expres-

and Oulu-CASIA datasets was performed as follows:

automatically cropped and aligned based on the eye land-

(Asthana et al. 2014) on each frame. The face region was
detected and facial landmark detection was performed

ally (Lee and Ro 2015). To unify the experiments on

fear, happy, sad, and surprise). In the MMI dataset, an

subjects. Each expression sequence was labeled with one

sion sequences with frontal faces were collected from 30

initial learning rate was set to 0.0001 for both the MMI and

tion was utilized in order to bound the LSTM input features

in all layers except the layer FCNN, in which sigmoid activa-
tion function, rectified linear unit (ReLU) was used

augmentations of the expression sequences were generated

even and odd frames from each sequence. As a result, 108
augmentations of the expression sequences were generated

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 land-
marks (Tian 2004). The construction of the utilized MMI

1. MMI dataset (Pantic et al. 2005): A total of 205 expres-

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 man-

ually (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 se-
quences 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 frames 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 un-
der three illumination conditions. For the experiments, a
total of 480 image sequences were collected from se-
quences captured with a visible light camera under nor-
mal illumination conditions. For each subject, the basic
expression sequence was captured from a neutral face un-
til 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.

were simulated by manually collecting frames from the begin-
ning and the 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 frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.

were simulated by manually collecting frames from the begin-
ning and the end of the expressive sequences. As such, sequences of
cropped neutral expression frames were generated. As total
of 38 natural expression sequences were generated and
added the total to the MMI sequences.
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

<table>
<thead>
<tr>
<th>Method</th>
<th>Expression Classes*</th>
<th>Input</th>
<th>Recognition rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D CNN-DAP (Liu et al. 2014)</td>
<td>6 sequence</td>
<td>63.4</td>
<td></td>
</tr>
<tr>
<td>DTAGN (Jung et al. 2015)</td>
<td>6 sequence</td>
<td>70.24</td>
<td></td>
</tr>
<tr>
<td>AURF (Liu et al. 2013)</td>
<td>7 static</td>
<td>69.88</td>
<td></td>
</tr>
<tr>
<td>AUDN (Liu et al. 2015)</td>
<td>7 static</td>
<td>75.85</td>
<td></td>
</tr>
<tr>
<td>CNN+LSTM (Kim et al. 2017)</td>
<td>6 sequence</td>
<td>78.61</td>
<td></td>
</tr>
<tr>
<td>CNN+LSTM (Kim et al. 2017)</td>
<td>7 sequence</td>
<td>76.64</td>
<td></td>
</tr>
<tr>
<td>Proposed method(E1 + E2)</td>
<td>7 sequence</td>
<td>78.29</td>
<td></td>
</tr>
<tr>
<td>Proposed method(E1+E2+E3)</td>
<td>7 sequence</td>
<td>78.96</td>
<td></td>
</tr>
</tbody>
</table>

This can be attributed to the efficient encoding of the expression dynamics in the proposed spatio-temporal features. 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 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.
Figure 5: Effect of the proposed method on the feature similarity 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 Figure 4.

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 (L1 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


