Detecting Manipulated Facial Videos: A Time Series Solution

Zhewei Zhang, Can Ma, Meilin Gao, Bowen Ding

The Third Research Laboratory,
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

zhangzhewei@iie.ac.cn
Manipulated Video Detection Problems

Challenges:

- Huge datasets with various manipulate generation methods
- Poor robustness on unseen cases, slow inference speed

Related Works: Facial Forgery Detection Method

- [Pan et al., 2007] Eye-Blinking Model
- [Afchar et al., 2018] CNN-Based Model
- [Ferrara et al., 2012] Color-based Model
- [Li et al., 2018] LSTM-CNN Model
Our approaches: FA/DFA-LSTM
Simple yet Effective

\[ I \in \mathbb{R}^{n \times 136}, \] in which \( n \) denotes the window length of LSTM. Note that \( n \) also equals to one slice of the input sequence.
Feature Project: Dense Face Alignment

3D Morphable Model: \[ S = \bar{S} + A_{shp} \alpha_{shp} + A_{exp} \alpha_{exp} \]

Texture Model: \[ T = \bar{T} + B_{shp} \beta_{shp} \]

Projected 3D facial vertices: \[ V(p) = s \cdot P_r \cdot R(\bar{S} + A_{shp} \alpha_{shp} + A_{exp} \alpha_{exp}) + t_x \]

Parameters: \[ p = [s, R, t_x, \alpha_{shp}, \alpha_{exp}] \]

PAF: 3D vertex \(\Rightarrow\) 64 \(\times\) 64 \(\times\) 2 feature anchors

Figure 4: Illustrations of different feature types. (a) raw sequence image. (b) 3D vertices mask of a facial surface. (c) PAF of a facial surface. (d) PAC of a facial surface.
DFA-LSTM

\[ J(W, Z) = \| X - W Z \|_F^2 \]

\[ Z \in \mathbb{R}^{L \times N}, \quad W \in \mathbb{R}^{D \times L} \]

\[
\begin{align*}
\Sigma &= 1/N \sum_{i=1}^{N} x_i x_i^T \\
\Sigma w_i &= \lambda_i w_i.
\end{align*}
\]

\( W_i \), \( 0 \leq i \leq N \) is computed as an eigenvector of the covariance matrix \( \Sigma \) with the eigenvalue \( \lambda_i \)

Loss Function:
\[ \mathcal{L}(\hat{y}_i, y_i) = -w_i[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] \]
Experimental Results

ROC curve for forgeries detection method on FaceForensics++ dataSet.
Compression Level: 23 and 40
### Experimental Results

#### Inference Speed and Accuracy Test

<table>
<thead>
<tr>
<th>Methods</th>
<th>CPU</th>
<th>GPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meso-4[Afchar et al., 2018]</td>
<td>43.251ms</td>
<td>16.344ms</td>
</tr>
<tr>
<td>XceptionNet[Chollet, 2017]</td>
<td>166.021ms</td>
<td>38.130ms</td>
</tr>
<tr>
<td>LRCN[Li et al., 2018]</td>
<td>38.260ms</td>
<td>13.974ms</td>
</tr>
<tr>
<td>CNN-Eye[Kim et al., 2017]</td>
<td>18.362ms</td>
<td>6.611ms</td>
</tr>
<tr>
<td>FA-LSTM</td>
<td><strong>16.125ms</strong></td>
<td><strong>5.314ms</strong></td>
</tr>
<tr>
<td>DFA-LSTM</td>
<td>37.320ms</td>
<td>10.253ms</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$C_{23}$</td>
<td>$C_{40}$</td>
</tr>
<tr>
<td>FA-LSTM</td>
<td>0.825</td>
<td>0.689</td>
</tr>
<tr>
<td>FA-LSTM + Attention</td>
<td><strong>0.879</strong></td>
<td><strong>0.721</strong></td>
</tr>
<tr>
<td>Meso-4[Afchar et al., 2018]</td>
<td>0.830</td>
<td>0.702</td>
</tr>
</tbody>
</table>
Experimental Results

Training Convergence Analysis for DFA/FA LSTM Method

(a) FA-LSTM, BatchSize=8
(b) DFA-LSTM, BatchSize=8
(c) FA-LSTM, BatchSize=1
(d) DFA-LSTM, BatchSize=1
Experimental Results

- Visual Effect Demo for Facial Forgery Detection

Example sequences of failure detectioncases. #1, #3, #5: Misclassification samples of FA-LSTM generated by DeepFake. #2, #4, #6: Improved results of #1 after introducing the attention layer. #1, #2 are forged samples, whereas #3 is an original sample.
THANKS