CapsuleNet for Micro-Expression Recognition

Nguyen Van Quang, Jinhee Chun, Takeshi Tokuyama
Graduate School of Information Sciences, Tohoku University, Sendai, Japan

Abstract—Facial micro-expression recognition has attracted researchers in terms of its objectiveness to reveal the true emotion of a person. However, the limited number of publicly available datasets on micro-expression and its low intensity of facial movements have posed a great challenge to training robust data-driven models for recognition task. In 2019, Facial Micro-Expression Grand Challenge combines three popular datasets, i.e., SMIC, CASME II, and SAMM into a single cross-dataset which requires the generalization of proposed method on a wider range of subject characteristics. In this paper, we propose a simple yet effective CapsuleNet for micro-expression recognition. The effectiveness of our proposed methods was evaluated on the cross-database micro-expression benchmark using the Leave-One-Object-Out cross-validation. The experiments show that our method achieved superiorly higher results than the baseline method (LBP-TOP) provided and other state-of-the-art CNN models.

I. INTRODUCTION

A Micro-Expression (ME) is observed as a brief and involuntary facial movement when a person experiences an emotion but tries to hide their genuine underlying emotion. The normal facial expressions, also known as macro-expressions, last 1/2 to 4 seconds involving large areas of facial movements [9]. Meanwhile, Matsumoto and Hwang [10] suggest that micro-expressions occur in a fraction of a second (usually from 1/5 to 1/25 of a second) in small local areas. The brevity and low amplitude of facial movements make micro-expressions challenging to recognize in real-time by human eyes and even by experienced experts. Unlike macro-expressions, it is difficult for people to fake their micro-expressions. Therefore, micro-expressions play a crucial role in understanding human’s underlying emotion, which powers various applications such as criminal interrogation [8], national security [5], and deceit detection [6], [7].

Due to the spontaneously induced characteristic of micro-expression, very few well-established datasets have been introduced, constraining the development of micro-expression research area. The number of samples in these datasets is usually too small to train robust micro-expression classifiers. In the second Micro-Expression Grand Challenge (MEGC), three spontaneous micro-expression datasets, including SAMM [2], CASME II [3], and SMIC [4], are integrated into a single cross-database with the motivation of increasing the number of samples as well as the diversity of subject characteristics. Although there have been no standard protocols of collecting and labeling the data, it is generally acceptable that MEs can be categorized into seven main emotions: happiness, surprise, contempt, anger, fear, sadness, and disgust. The previous MEGC challenge used the objective class labels following the proposal in [1]. However, the number of classes in the MEGC challenge is simplified into three labels, namely, negative, positive, and surprise in order to reduce the ambiguity among datasets. The task of this MEGC challenge is to classify which of three categories an ME sequence belongs to with the ME cross-database.

Computational automatic analysis on micro-expression has recently attracted interest from researchers. Hand-crafted feature engineering methods were widely employed using from local binary pattern, histogram to optical flow as features to train traditional machine learning models [3], [14]–[20], [23]. Convolutional Neural Network (CNN) gaining huge success on visual tasks were also applied successfully to ME recognition problem as feature extractors or classifiers [13], [24], [25]. The adoption of CNN networks replaces the manual feature engineering procedure by automatically finding a good feature representation for images. However, CNNs have still struggled to represents the part-whole relationships between the entity and its parent in the image which is described to be powerful against spatial variations and adversarial attacks. To address the limitations of CNNs, Sabour et al. [29] quite recently introduced Capsule Networks (CapsuleNet), in which a capsule is a group of neurons that represent various properties of entities in a vector rather than a scalar. CapsuleNet find the part-whole relationships via an agreement routing mechanism. CapsuleNet have shown superiority over CNNs for digit [29] and object [30], [31] recognition.

Inspired by the recent success of capsule models on image recognition compared to traditional neural networks, we propose a CapsuleNet for micro-expression recognition using only apex frames. We would like to examine the capability of CapsuleNet to figure out the part-whole relationships and be trained effectively on only small datasets like micro-expression recognition task. The unweighted average recall score and unweighted F1 scores our model obtained in this challenge show that our method outperforms the provided LBP-TOP baseline method and the other state-of-the-art CNN models as well. In the full framework for ME recognition, we first apply the preprocessing step to detect unknown apex frames from frame sequences and extract the facial area from those apex frames. We, in turn, forward the cropped facial images into the CapsuleNet to perform classification. Our implementation for ME CapsuleNet is published at the github link: https://github.com/quangdtsc/megc2019. To the best of our knowledge, this is the first work which applies...
the idea of CapsuleNet on micro-expression recognition.

II. RELATED WORK

A. Hand-crafted approaches

Since the introduction of spontaneous ME datasets, several works were benefited from handcrafted feature engineering techniques such as Local Binary Pattern with three Orthogonal Planes (LBP-TOP) [20]. Yan et al. [3] utilized LBP-TOP to extract features, fitting into an SVM classifier to perform the recognition. Wang et al. [14] adopted LBP-Six Intersection Points (LBP-SIP) to refine the features. Li et al. [15] combined LBP-TOP, Histogram of Oriented Gradients, and Histogram of Image Gradient Orientation together, yielding single feature vectors to perform the recognition.

Besides, some techniques in video analysis were also adopted to capture the temporal information. X.Li et al. [16] used the Temporal Interpolation Model to sample uniformly the image frames from the ME sequence. Optical Flow techniques were also used as a good feature descriptor in several works. Shreve et al. [23] introduced an optical strain, the derivative of optical flow, as a feature descriptor which were used later in [17], [18]. Liu et al. [21] proposed Main Directional Mean Optical-flow (MDMO) to compute facial movements while Xu et al. [22] applied Facial Dynamics Map (FDM) and [19] used Bi-Weighted Oriented Optical Flow.

B. Deep learning approaches

Deep learning techniques have recently been applied to micro-expression recognition task. In their early work, Patel et al. [27] utilized Convolutional Neural Network (CNN) trained on macro-expression databases as a feature extractor. The extracted features are later forwarded to the genetic algorithm before fitting into traditional classifiers. Peng et al. [25] proposed Dural Temporal Scale Convolutional Neural Network (DSTCNN) which was designed with shallow architecture to prevent overfitting on small micro-expression datasets. Very recently, Khor et al. [24] introduced an enriched version of Long-term Recurrent Convolutional Network which consists of spatial feature extractor and a temporal module to capture the temporal information. Peng et al. [13] applied transfer learning from macro-expression datasets to micro-expression recognition task on a ImageNet-pretrained ResNet10 model.

C. Capsule Networks

After the introduction of the CapsuleNet idea, more works have been done to examine the capability of CapsuleNet in various research areas. A CapsuleNet for brain tumor recognition designed by Afshar et al. [33] exceeds the performance of CNN networks. Jaiswal et al. [34] introduced Generative Adversarial Capsule Networks (CapsuleGAN) which out-performs CNN-based GAN at modeling image distribution on the MNIST dataset. For sentiment classification task in natural language processing, Wang et al. [32] proposed a RNN-Capsule architecture with state-of-the-art results.

These above works trigger the motivation for our work to examine whether we can apply successfully CapsuleNet to micro-expression recognition or not.

III. OUR PROPOSED METHOD

In this section, we present a complete framework for micro-expression recognition which adopts CapsuleNet architecture as the main component. The framework first detects and preprocesses the apex frames from the ME sequences if not provided before generating the ME predictions. We adopt transfer learning mechanism to initialize the pretrained weights on ImageNet for the first convolutional layers while training the network. Figure 1 summarizes the proposed framework with the preprocessing module and classification module.

A. Preprocessing

The ME sequence starts with an onset frame recording the neutral expression, and ends with an offset frame when the subject returns to the neutral expression. Meanwhile, the apex frame of ME sequence indicates the highest change in pixel intensities when the ME occurs. Several works [13], [19] show that apex frames provide rich information enough for ME recognition task. However, the apex frames are only annotated in SAMM and CASME II datasets while omitted in the SMIC dataset. In the preprocessing module, we first locate the apex frame of each ME, and segment the facial area out from the located apex frame.

We apply an open-source facial toolkit to obtain 68 landmarks of the face in each frame of ME sequence. Following [28], we define 10 regions on the face based on the detected landmarks which represent for facial areas where muscle movements occur very frequently. The size of each cell is estimated by half of the mouth width heuristically. 10 regions were depicted in Figure 2.

To decide which one in the sequence is the apex frame, we compute the absolute pixel differences between the current frame with the onset and offset frames in the ten regions. To reduce the noise of environment, we normalize the summation of two difference by dividing it with the difference between the considered frame and its consecutive frame. Finally, we obtain the per-pixel average value for each frame in the ME sequence. The apex frame should indicate the peak of intensity differences with the onset and offset frame of the sequence. Therefore, we approximately select the frame with the highest per-pixel value of $M_i$. The variation of the mean of $M_i$ is demonstrated in Figure 3.

$$f(frame_i, frame_{j}) = \frac{|frame_i - frame_j| + 1}{|frame_i - frame_j| + 1} \quad (1)$$

$$M_i = f(frame_i, frame_{onset}) + f(frame_i, frame_{offset}) \quad (2)$$

After detecting the apex frame (on SMIC dataset only), we crop the facial area on the apex frames of the ME sequences based on the facial landmarks, which later are fitted into the CapsuleNet for performing classification. We sumarize our preprocessing module in the Figure 4.

https://github.com/ageitgey/face_recognition
B. CapsuleNet

We adopt CapsuleNet architecture, which aims to perform micro-expression recognition on the apex frame of its ME sequence. As illustrated in Figure 2, the architecture takes apex frames as input images. The cropped facial images of apex frames obtained are converted to color images and resized to the shape $[224, 224, 3]$. The input shape is extremely larger than the digit input shape in [29]. Since the facial movements of micro-expressions are low in intensity, we will lose a lot of information if we resize the input image into smaller shape. Therefore, we feed the input images into the first convolutional layer and three convolutional blocks of ResNet18 model [37] to transform pixel intensities into local features of shape $[28, 28, 256]$. We avoid to use larger ResNet versions like ResNet50 or ResNet101 to reduce the possibility of overfitting on small datasets. The initial weights of the model were set by the pretrained weights of ResNet18 respective layers on ImageNet. This design helps to reduce the exponential number of parameters while allows transferring the knowledge learned from object recognition on huge dataset like ImageNet.

These features, in turn, are passed into the primary capsule layer to obtain primary capsules (denoted as PrimaryCaps) by multiplying with convolutions. Each of primary capsule encapsulates the information and characteristics of an entity in a vector of neurons instead of a scalar value in CNNs. The activity of neurons in a capsule describes the instantiation parameters of that entity. We denote $i$ as a capsule at the primary capsule layer and $j$ as a capsule at the output capsule layer. The activation of the capsule $j$ is determined based on the activations of all capsules in the primary layer. The coupling coefficient $c_{ij}$ measures the agreement between capsule $i$ and capsule $j$ which is computed by routing softmax as follows:
where \( b_{ij} \) is denoted as the log probability of whether capsule \( i \) at primary capsule layer should be coupled with parent capsule \( j \). \( b_{ij} \) is initially set to 0 and \( c_{ij} \) is iteratively refined via the dynamic routing process which we refer readers to [29] for more details. Then, input to capsule \( j \) (denoted as \( x_j \)) is computed as:

\[
x_j = \sum_i c_{ij} W_{ij} u_i
\]

In the output capsule, its length represents the probability that a certain entity exists while its orientation is forced to represents the properties of the entity. To ensure the length of the output vectors between the interval \([0,1]\) while its orientation keep unchanged, a non-linearity squash function is applied. The final output of capsule \( j \) i.e. \( v_j \) is computed using the squash function as below:

\[
v_j = \frac{||x_j||^2 x_j}{1 + ||x_j||^2 ||x_j||}
\]

In our CapsuleNet, the output capsules (denoted as MeCaps) are used only for the purpose of micro-expression recognition but not for reconstruction of the input images. The additional sub-net of reconstruction in [29] was added as a regularization for the overall network with input digit images of size \([28 \times 28]\). However, we remove this reconstruction part from the network since the input image size is much larger, \([224 \times 224]\), and the number of training dataset is quite small.

C. Network Optimization

We use the margin loss as an objective function for training our network:

\[
L\text{net} = \sum_k L_k^{\text{margin}}
\]

where \( L_k^{\text{margin}} \) is the margin loss for the respective ME \( k \). If an image contains an ME \( k \), we force the length corresponding capsule to be long while we expect its length to be short when there is no ME in the image. To satisfy such conditions, we employ the following margin loss for each ME \( k \):

\[
L_k^{\text{margin}} = T_k \max(0, m^+ - ||v_k||)^2 + \lambda_k (1 - T_k) \max(0, ||v_k|| - m^-)^2
\]

where \( T_k = 1 \) if ME \( k \) exists in the image and \( T_k = 0 \) if \( k \) does not exist in the image. \( \lambda_k \) specifies the effect of losses obtained when the ME is present or absent in the image. Finally, margin loss is provided to be zero if \( ||v_k|| > m^+ \) when \( T_k = 1 \) and \( ||v_k|| < m^- \) when \( T_k = 0 \).

IV. EXPERIMENTS

A. Experimental setup

In our architecture, after passing input of size \([224, 224, 3]\) to the first convolutional block, we obtain a tensor of shape \([20, 20, 256]\). In the PrimCaps layer, we have 32 channels of convolutional 8D capsules where each capsule has 8 convolutional neuron units with a \([9 \times 9]\) kernel. Capsules in \([6 \times 6]\) grid share weights and we obtain \([6 \times 32]\) 8D capsule outputs. Each capsule in PrimCaps is connected to each capsule in MeCaps layer by a weight matrix \( W_{ij} \) of size \([16 \times 8]\). There are 3 capsules in the final outputs MeCaps. The margin loss \( L_k^{\text{margin}} \) is defined such that \( m^+ = 0.9, m^- = 0.1 \) and \( \lambda_k = 0.5 \) as suggested in [29]. During optimization we used Adam optimizer with a learning rate 0.0001 and decaying learning rate weight 0.9 in 20 epochs. We used 3 iterations of dynamic routing. The data augmentation was applied with some well-known transformations including: resizing, random cropping, mirroring, rotation and color jittering.

B. Baseline

A method from [12] using LBP-TOP was reimplemented as the first choice for the baseline for the MEGC challenge 2019. LBP-TOP descriptor first proposed by Zhao et al [20] is a well-known feature descriptor in micro-expression representation. In LBP-TOP method, frames in ME sequences were divided into \([5 \times 5]\) non-overlapping blocks with LBP-TOP parameters: \( RXY, RXT, RYT \) = \([1, 1, 1]\), number of neighboring points \( P = 4 \) for all planes, and \( TIM = 10 \).

We also compare our proposed model with two state-of-the-art network architectures for object recognition, namely, ResNet (ResNet18 version) [36] and VGG (VGG11 version) [37]. Two baseline models were modified by replaced the last fully connected layer with 1000 category outputs by another fully connected layer with 3 category outputs. Except for the last layer, the initial weights of the model were set by the pretrained weights on ImageNet. The multi-label cross entropy is employed as the loss function. The training procedure is performed with the same learning rate, epochs, and optimizers as training our proposed model.

C. Datasets

The cross-database of the 2019 challenge comprises of 3 popular spontaneous datasets as follows:

SMIC database [4] includes 164 micro-expressions from 16 subjects. Each ME was recorded at the speed of 100 fps and labeled with three general emotion labels: positive, negative and surprise. Recently, a new version of the database, SMIC-E, was published, which also contains some non-expression frames before and after the labeled microframes.

CASME II [3] is a comprehensive spontaneous micro-expression database containing 247 video samples, elicited from 26 Asian participants with an average age of 22.03 years old. The videos in this database showed a participant
evoked by one of five categories of micro-expressions: Happiness, Disgust, Repression, Surprise, Others.

**SAMM**, The Spontaneous Actions and Micro-Movements, [2] is a newer database of 159 micro-movements (one video for each) induced spontaneously from a demographically diverse group of 32 participants with a mean age of 33.24 years, and an even male-female gender split. Originally intended for investigating micro-facial movements, the SAMM was induced based on the 7 basic emotions.

Both the CASME II and SAMM databases have much in common: They are recorded at a high speed frame rate of 200 fps. Meanwhile, SMIC only record at the speed frame rate of 100 fps. To avoid the clutter and complication when combining all three datasets together, the challenge introduces a simplified class protocol for assigning labels for each sample as SMIC. There are altogether 68 subjects (16 from SMIC, 24 from CASME II, 28 from SAMM) after the databases are consolidated based on the new generic classes. Since the cross-database is too small to suffice the network, we enriched the datasets by acquiring the apex frame in ME sequence and its 4 neighbor frames as the training dataset. The resampling technique was also applied to reduce the effect of imbalance in the training dataset.

![Table I: The label summary on each dataset and cross-database (3DB-combined).](attachment:image.png)

**Evaluation metrics.** Both Holdout-Database Evaluation (HDE) and Composite Database Evaluation (CDE) were used to evaluate the effectiveness of recognition methods in the last year MEGC challenge. However, HDE procedure is not a wise choice since it leads to combinatorial explosion of many permutations for train-test partitions from the cross-database. Following [12], we use Leave-One-Subject-Out (LOSO) cross-validation as a CDE method to report the performance on ME recognition. The real world situation of people from a wide range of backgrounds including ethnicity, gender emotional sensitivities which were recorded in different settings would be considered with LOSO evaluation method. Furthermore, it also leads to subject-independent evaluation. Since the cross-database is apparently imbalanced, the recognition performance is evaluated with two balanced metrics:

**Unweighted F1-score** (UF1) is also commonly known as macro-averaged F1-score. We first calculate all the True Positives (TP	extsubscript{c}), False Positives (FP	extsubscript{c}) and False Negatives (FN	extsubscript{c}) over all k folds of LOSO by each class c (of C classes), compute their respective F1-scores, and the final balanced F1-score is determined by averaging the per-class F1-scores as follows:

\[
F1\textsubscript{c} = \frac{2TP\textsubscript{c}}{2TP\textsubscript{c} + FP\textsubscript{c} + FN\textsubscript{c}} \\
UF1 = \frac{1}{C} \sum_{c} F1\textsubscript{c} 
\]

**Unweighted Average Recall** (UAR), or also known as balanced accuracy of the system. The per-class accuracy scores UAR	extsubscript{c} are first calculated, and we average all Acc	extsubscript{c} by the number of classes to obtain the final UAR score as below:

\[
Acc\textsubscript{c} = \frac{TP\textsubscript{c}}{N\textsubscript{c}} \\
UAR = \frac{1}{C} \sum_{c} Acc\textsubscript{c} 
\]

Both UF1 and UAR give us a fair evaluation on how well the model performs on all the classes rather than biasing on only few certain classes.

**V. RESULTS AND DISCUSSION**

We report the unweighted F1 and recall scores of our proposed model with the baseline provided by the challenge as well as the baselines we set up in Table II. Table II compares the UF1 and UAR scores on Full cross-database, and on separate parts including SMIC, CASME II and SAMM between the baselines and our proposed model. Our model obtained the UF1 score of 0.6512, and the UAR of 0.6498. Apparently, our model performance is the best among those of the baselines. The micro-expression performance of the proposed methods outperforms the state-of-the-art CNN networks, ResNet18 and VGG11 with large margins, approximately about 10%, which confirms the effectiveness of CapsuleNet architecture. Rather than using all the frames in ME sequence like LBP-TOP method to extract features, our model utilizes only a single apex frame of each ME as the input data. However, it still gained the 6.5% higher of UF1 and compared with LBP-TOP method.

Figure 5 shows the LOSO confusion matrix of our proposed method. The recall rate of negative class is the highest (0.780) among the other classes since the negative samples are dominant in the cross-database. However, the recall rates of the two remaining classes are also acceptable with 0.596 and 0.575 respectively. This indicates the efficiency of the resampling technique for our CapsuleNet against imbalance-data effect.

**Ablative study** The additional layers added to the original CapsuleNet resolves the computational complexity. Although the choice of the additional layers could be quite diverse, we prefer the layers from state-of-the-art networks like ResNet and VGG to transfer the knowledge learned from bigger datasets like ImageNet with transfer learning technique. The layers extracted from ResNet18 are described as our proposed architecture above while the layers extracted from VGG11 are the first 3 convolutional layers of VGG11...
This work was supported by JSPS KAKENHI Grant Number 15H02665, 17K0002.

REFERENCES


VI. CONCLUSIONS

This work introduced a complete framework with a CapsuleNet for micro-expression recognition using only apex frames. The additional design is the key to reduce the computational complexity of the CapsuleNet and improve the generalization on small micro-expression datasets. Also, CapsuleNet exploits the knowledge from apex frames only without heavy and complicated computations when using all the frames in micro-expression sequence. Experimental results show the effectiveness proposed method which outperforms the LBP-TOP baseline and several powerful CNN models in micro-expression recognition.

VII. ACKNOWLEDGEMENTS

ACKNOWLEDGEMENTS

TABLE II: The LOSO cross-validation performances for our proposed methods and the baselines.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Cross-database</th>
<th>UAR (Full)</th>
<th>Unweighted F1</th>
<th>Unweighted Average Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>LBP-TOP</td>
<td>0.5885</td>
<td>0.5791</td>
<td>0.2000</td>
<td>0.5280</td>
</tr>
<tr>
<td>VGG11</td>
<td>0.5264</td>
<td>0.5392</td>
<td>0.3461</td>
<td>0.3558</td>
</tr>
<tr>
<td>ResNet18</td>
<td>0.5392</td>
<td>0.5459</td>
<td>0.3576</td>
<td>0.3602</td>
</tr>
<tr>
<td>Our CapsuleNet</td>
<td>0.6520</td>
<td>0.6506</td>
<td>0.5820</td>
<td>0.5877</td>
</tr>
</tbody>
</table>

TABLE III: The LOSO cross-validation performances for the ablative study.

<table>
<thead>
<tr>
<th>Method</th>
<th>Scores</th>
<th>Unweighted F1</th>
<th>Unweighted Average Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapsuleNet used VGG11</td>
<td>0.6130</td>
<td>0.6260</td>
<td></td>
</tr>
<tr>
<td>CapsuleNet used ResNet18</td>
<td>0.6520</td>
<td>0.6506</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5: Confusion matrix on cross-database with the LOSO cross-evaluation method. The number in each cell indicates the number of predictions.

<table>
<thead>
<tr>
<th>Confusion Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>192 30 24</td>
</tr>
<tr>
<td>33 65 11</td>
</tr>
<tr>
<td>25 9 46</td>
</tr>
</tbody>
</table>

TABLE III: The LOSO cross-validation performances for the ablative study.


