Capsule Network based End-to-end System for Detection of Replay Attacks

Meidan Ouyang, Rohan Kumar Das, Jichen Yang, Haizhou Li

Department of Electrical and Computer Engineering,
National University of Singapore, Singapore
meidan.ouyang@u.nus.edu.sg, {rohankd, eleyji, haizhou.li}@nus.edu.sg

Abstract

Automatic speaker verification systems are prone to various spoofing attacks. The convolutional neural networks are found to be effective for detection of spoofing attacks. However, they lack spatial information and relationship of low-level features with the pooling layer. On the other hand, capsule networks use vectors to record spatial information and the probability of presence simultaneously. They are known to be effective for detection of forged images and videos. In this work, we study capsule networks for replay attack detection. We consider different input features to capsule network and study on recent ASVspoof 2019 physical access corpus. The studies suggest the proposed capsule network based system performs effectively and the performance is comparable to state-of-the-art single systems for replay attack detection.

Index Terms: replay attack detection, capsule network, ASVspoof 2019, anti-spoofing

1. Introduction

Biometric recognition techniques use physiological characteristics or behavioral traits of humans to recognize an individual. Speech is one among such traits, which is used in automatic speaker verification (ASV) systems [1–4]. As we continue to improve the robustness of ASV under various conditions, ASV systems become more vulnerable to spoofing attacks, which can be mistakenly considered as a channel or noise variation [5, 6]. There are four broad categories of such attacks, which are impersonation, text-to-speech (TTS), voice conversion (VC) and replay [7]. Impersonation refers to mimicking the target speaker voice, which is a behavioural attack and ASV systems are less vulnerable to such attacks [8]. The TTS and VC based attacks are logically derived attacks, which need the attacker to have knowledge of implementing such systems [9]. In contrast to these attacks, the replay attacks can be easily realized by pre-recorded samples of the target speakers that poise an imminent threat. Therefore, we focus on studying replay attack detection in this work.

The rising importance of spoofing attack detection has led to organization of ASVspoof1 challenge series [10]. The second edition ASVspoof 2017 projected the threat of replay attacks in uncontrolled setup [11]. The latest edition ASVspoof 2019 was devoted to detect spoofing attacks derived by the state-of-the-art VC and TTS systems in one track as well as replay attacks generated in a simulated controlled setup in another track [12]. This setup of replay is chosen to analyze such attacks more carefully in contrast to the previous edition.

The earlier attempts for spoofing attack detection consider robust hand-crafted features such as cochlear filter cepstral coefficient and instantaneous frequency [13], linear frequency cepstral coefficients (LFCC), subband spectral flux coefficients and spectral centroid frequency coefficients [14]. Subsequently, the long-term constant-Q transform (CQT) based constant-Q cepstral coefficients (CQCC) [15, 16] and other features [17–21] emerged as very effective front-ends for detection of spoofing attacks. The recent explorations study deep features and deep learning systems apart from strong hand-crafted front-end features [22–32].

The results of ASVspoof 2019 showed that the spoofing countermeasures using neural networks in either front-end or back-end performed well [12]. Some of those used deep feature extractors to represent utterance level embeddings for robust detection [22, 23]. However, most of the systems used deep learning system as a classifier for processing various handcrafted features [24–30]. Among these works, the end-to-end systems developed using convolutional neural network (CNN), light CNN (LCNN), gated recurrent unit (GRU) were reported as some of the well performing systems [26, 27, 29]. These end-to-end systems have the advantage of setting the hyperparameters of feature extraction as well as classification mutually and simultaneously for effective learning. Therefore, we are interested in developing robust end-to-end systems for spoofing attack detection.

Capsule networks are novel deep learning models that capture spatial information by use of vectors and the probability of presence simultaneously [33]. For instance, the locations of the eyes and noses can be known for face detection task. The capsule networks have been studied for various speech processing applications like command and emotion recognition [34, 35]. Recently, they are used for detection of forged images and videos created from replay attacks using printed images or recorded videos to computer generated videos [36]. The studies showed that the capsule networks outperform other state-of-the-art systems for identifying such attacks [36].

We believe the spatial information captured by capsule networks could be useful to describe the artifacts for detection of replay speech in a similar way as forged images and videos. The replay speech differs from the genuine speech due to the effects of recording, playback devices and background environment that gets added during attack. Therefore, spatial information in terms of relative position of these effects along time-frequency axis by using spectrogram as an input to capsule networks can reflect additional artifacts that discriminate them from the genuine speech. With this motivation, we study the capsule networks for detection of replay attacks using ASVspoof 2019 physical access corpus.

The remainder of the paper is organized as follows. Section 2 describes the details of capsule networks used for replay attack detection. In Section 3, we mention the experiments for the studies, followed by their results and analysis in Section 4. Finally, the work is concluded in Section 5.

1http://www.asvspoof.org/
2. Capsule Network for Replay Detection

2.1. Theory

Capsule network with dynamic routing was proposed in [33]. It was suggested that human visual system establishes a coordinate frame, and the difference of coordinate frames greatly changes human cognition. In other words, when people recognize objects, the coordinate frame is involved in the recognition process, which is dominated by the concept of space. Therefore, the capsule represents an entity by a group of neurons instead of a single neuron.

In general, the capsule networks use the high dimensional vectors to replace scalar neurons, which are used by neural networks such as CNNs and deep neural networks (DNNs). They consider the length of the vector to represent the probability of the entity’s existence, and the direction of the vector to represent the instantiation parameters (some graphical properties of the entity). A non-linear squashing function is applied as the activation function of capsule network to ensure that the length of short vectors map to almost zero and that of long vectors transforms to a length slightly below one.

Considering \( v_j \) and \( s_j \) as the output and input vectors, respectively, of capsule \( j \), the vector weighted sum of all the capsules from the previous layer transfer to the current layer, capsule \( j \) is

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

To obtain \( s_j \), one needs to first calculate ‘prediction vector’ \( \hat{u}_{j|i} \), as

\[
    \hat{u}_{j|i} = W_{j|i} u_i
\]

where \( W_{j|i} \) is the weighted matrix between capsule \( i \) in previous layer and capsule \( j \) in current layer, \( u_i \) is the output of previous layer. Then,

\[
    s_j = \sum_i c_{ij} \hat{u}_{j|i}
\]

where \( c_{ij} \) represents coupling coefficients that are calculated by the iterative dynamic routing process, shown as Figure 1, which determine the way of propagation between two layers. The \( c_{ij} \) can be computed by applying softmax function to \( b_{ij} \) as

\[
    c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}
\]

where \( b_{ij} \) represent the log prior probabilities that capsule \( i \) should by coupled to capsule \( j \). The \( b_{ij} \) is initially set to 0, which means the output in previous layer is routed to all possible capsules in the current layer equally. Then, \( v_j \) in Equation (1) can be obtained to update \( b_{ij} \) by

\[
    b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} v_j
\]

The output capsule is then computed with updated \( b_{ij} \) for \( r \) number of routing times. Finally, \( v_j \) at the final iteration is returned as the output of capsule \( j \) in the current layer.

Generally, the margin loss is used as a loss function to train the model.

\[
    L_c = T_c \max(0, m^+ - \|v_c\|)^2 + \lambda(1 - T_c) \max(0, \|v_c\| - m^-)^2
\]

where \( T_c \) is the margin loss of class \( c \), and \( v_c \) is a final output capsule in class \( c \). \( T_c = 1 \) when the target class is \( c \), \( m^+ \) and \( m^- \) are the margins, and \( \lambda \) decreases the weight of the loss for the absent classes. Further, the sum of the losses of all classes is the total loss.

2.2. Architecture

As in other speech applications, the input of capsule networks can come from some convolutional layers to present the latent information [34, 35]. The output of capsule network can be regarded as the final classification results directly or further processed by additional dense layers. We study both the configurations of the capsule networks and they are shown in Figure 2. The pipeline without additional dense layers is referred to as CapsNet, and the one with an additional dense fully-connected (FC) layers is referred to as CapsNetFC hereon in this work.

The convolutional layers block shown in Figure 2 has 4 convolutional layers. Each of them has the same kernel size of \( 3 \times 7 \) and stride of \( 2 \times 2 \), except the final convolutional layer with the stride of \( 1 \times 1 \). In addition, all convolutional layers are followed by batch normalization (BatchNorm) and rectified linear unit (ReLU) layers. We summarize the details of the convolutional layers in Table 1.

We now present the capsule layers of Figure 2 in Figure 3 considering short-time Fourier transform (STFT) spectrogram as input to the convolutional layers of Figure 2. The STFT spectrogram \( 120 \times 1025 \) transforms to \( 15 \times 129 \) as the output
of convolutional layers. It can be seen that another convolutional layer with strides of $2 \times 2$ and filter size of $3 \times 7$ is used alone in capsule layers without any batch normalization layers or activation layers, for adapting the capsule shape. Then the output is transformed into $8320 \times 16$, which can be considered as $8320$ input capsules and each of $16$ dimensions. The routing algorithm is employed to update coupling coefficients with three iterations, while the weighted matrix is updated by backpropagation. As in Figure 2, CapsNet output layer has two capsules, bonafide capsule and spoof capsule, where the length of each capsule is the final score for each class.

The fully-connected layers are used with the output of CapsNet to have another dense model CapsNetFC. In this case, the output of capsule layers block, containing both class and length output of capsule layers block, is fed to two fully-connected layers. Each layer has dropout with rate of $0.5$ to reduce overfitting, and ReLU activation is applied for the first fully-connected layer and lastly Softmax layer is used to obtain final results. Table 2 shows the details of fully-connected layers block.

Table 2: Summary of fully-connected layers block.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dropout</td>
<td>0.5</td>
</tr>
<tr>
<td>ReLU</td>
<td></td>
</tr>
<tr>
<td>Dropout</td>
<td>0.5</td>
</tr>
<tr>
<td>Softmax</td>
<td>-</td>
</tr>
</tbody>
</table>

The training set of ASVspoof 2019 physical access corpus contains larger number of spoofed utterances than that of bonafide utterances. Therefore, we considered equal number of spoofed utterances randomly in each training epoch to balance with the number of bonafide utterances. In addition, all the models are trained with AMSGrad optimizer. The initial learning rate is set as $0.001$, weight decay $\lambda = 1 \times 10^{-3}$, and momentum is set as $0.9$. The loss function for CapsNetFC is cross-entropy. On the other hand, CapsNet uses margin loss as given in Equation (6), where $m^+ = 0.9$, $m^- = 0.1$ and $\lambda = 0.5$.

Figure 3: Block diagram showing detailed architecture of capsule layers block. The output of convolutional layers are are fed to the input capsule layer and the output capsule layer has two capsules that are obtained by dynamic routing.

Table 3: Summary of ASVspoof 2019 physical access corpus.

<table>
<thead>
<tr>
<th>Subset</th>
<th>#Speakers</th>
<th>#Utterances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>8, 12</td>
<td>5,400, 48,600</td>
</tr>
<tr>
<td>Development</td>
<td>4, 6</td>
<td>5,400, 24,300</td>
</tr>
<tr>
<td>Evaluation</td>
<td>21, 27</td>
<td>18,090, 116,640</td>
</tr>
</tbody>
</table>

3.2. Experimental Setup

The studies in this work are carried out with different inputs to the capsule network. We consider STFT and long-term CQT spectrograms. The studies in [27] showed the higher resolution features significantly outperformed lower resolution features. Therefore, we consider the same high resolution features with 2048 STFT bins each. Further, each utterance along time axis is set as around $2.4$ s duration to maximize the information. The zero padding is performed for the shorter ones and longer ones are made fixed length by selecting required duration randomly. In case of CQT, the parameters are set following those by the original authors [15, 16].
We are now interested to compare the performance of the capsule network based anti-spoofing system to some of the other known systems on ASVspoof 2019 physical access corpus. The ASVspoof 2019 baseline systems developed using CQCC and LFCC features with Gaussian mixture model (GMM) classifier are considered first for this comparison [12]. Additionally, results of some of the well performing single systems such as CNN, GRU and residual networks (ResNet) are also considered. Table 5 shows the performance comparison of capsule network based end-to-end system to the some of the other single systems discussed. We find that the proposed capsule network performs much better than the two baselines of ASVspoof 2019 challenge. Further, they outperform CNN and ResNet based systems and perform comparable GRU based systems. We further extend the studies to evaluate the influence on the distance of recording devices from the speaker and the quality of replay devices. These settings are represented by two letters, where the first one denotes the distance of the recording device from speaker (A: 10-50 cm, B: 50-100 cm, C: >100cm) and the second one stands for the quality of replay devices (A: perfect, B: high, C: low) as per the ASVspoof 2019 protocol. We also compare the proposed CapsNetFC system to one of the ASVspoof 2019 baseline (CQCC-GMM) for this study. Table 6 shows the results for this comparison under different replay attack settings. It suggests that quality of the replay devices has the most influence on identifying replay attacks as comparatively very high t-DCF and EERs are observed for the AA, BA and CA configurations. Additionally, the proposed CapsNetFC based system is less sensitive to quality of replay device as well as distance range than the CQCC baseline system. This shows the potential of the proposed system for real-world applications.

### 4. Results and Analysis

We study the capsule network based system discussed in previous section on ASVspoof 2019 physical access corpus. Table 4 shows the results of capsule network configurations CapsNet and CapsNetFC based systems for different inputs. Comparing STFT and CQT spectrogram input based results for both the systems, we observe that the former performs better. Further, CapsNetFC outperforms CapsNet based system, showing the gain achieved due to having a denser architecture. The log spectrograms of the STFT and CQT are also investigated for both systems. We find the use of log spectrograms help in most of the cases to obtain an improved results, which is most dominant for STFT based spectrograms.

### 5. Conclusion

This work focuses on studying capsule network based end-to-end systems for detection of replay attacks. The spatial information captured by the capsule network is expected to have relevant artifacts for replay speech that cannot be captured by conventional neural networks. We developed capsule network based end-to-end systems using different inputs to evaluate the detection of replay attacks on ASVspoof 2019 physical access corpus. The studies suggested that the proposed system outperforms the two baseline systems of the challenge and is comparable to some of the well performing single systems. This highlights the usefulness of spatial information in detection of replay attacks. The future work will focus on investigating capsule networks for synthetic speech detection as well as generalized countermeasures for unknown spoofing attacks.

### 6. Acknowledgements

This research work is partially supported by Programmatic Grant No. A1687b0033 from the Singapore Government’s Research, Innovation and Enterprise 2020 plan (Advanced Manufacturing and Engineering domain), Human-Robot Interaction Phase 1 (Grant No. 192 25 00054) by the National Research Foundation, Prime Minister’s Office, Singapore under the National Robotics Programme. This work is also part of a collaboration with Kriston AI Lab, China in 2020.