Gait Recognition Using Convolutional Long Short-Term Memory Neural Networks

Enes Dursun
Istanbul Technical University
dursun18@itu.edu.tr

***Abstract*—The human gait has become another biometric trait used in security systems because it is unique to each person and can be recognized at a distance. Machine learning becomes a useful tool in analyzing such data that signifies behavioral characteristics of humans. Recently, deep learning based on convolutional neural network has achieved great state-of-the-art performance in many fields such as image classification, semantic analysis, and biometric recognition. Therefore, a deep convolutional neural network is one of the most advanced machine learning technique that has the ability to approximate complex non-linear functions from high-dimensional input data in a hierarchical process, such as gait recognition. To model a gait sequence, the LSTM recurrent neural network is naturally adopted. Convolutional LSTM, by combining convolution operation with LSTM architecture, has shown to be powerful with spatiotemporal sequences. The experiments were made on a popular but challenging dataset in terms of cross-view gait recognition.**

Keywords—gait recognition, convolutional neural networks, long short-term memory, recurrent neural networks, deep learning.

I. Introduction

Human motion analysis is necessary in various areas of computer science such as biometrics, computer graphics and games industry. Gait recognition is a biometric technique that is used in order to determine the identity of humans based on the style and the manner of their walk.

From a computer vision point of view, gait recognition could be seen as a particular case of human action recognition. While deep convolutional neural networks (CNNs) have shown a great success in single-label image classification, it is important to note that real world has a time dimension. Human brain does not start thinking from scratch every second. As we read a book our brain understands each word based on your understanding of previous words. Recurrent neural networks address this issue.

Since it requires a sequential data to recognize a gait, we need to use Recurrent Neural Networks. LSTM has been proven to be a powerful model for sequential data prediction. However, the traditional fully connected LSTM (FC-LSTM) cannot efficiently catch the spatial correlation within the data. Meanwhile, convolutional neural network has shown its effectiveness of capturing the spatial correlation with sparsity of connection and parameter sharing.

In this paper we will use a Convolutional Neural Network to extract features from gait silhouettes and use these features to classify humans with Long Short-Term Memory classifier.

II. Related Work

Gait recognition has been studied widely in the last 30 years. The majority of the state-of-the-art approaches to gait recognition are based on hand-crafted features of body motion: pose estimations or just silhouettes. One of the most popular descriptors based on silhouette is Gait Energy Image (GEI, Han and Bhanu, 2006), the average image of the binary silhouette masks of the subject over gait cycle. In (Chen and Liu, 2014) the modification of GEI is proposed-frame difference energy image. Instead of averaging all normalized silhouettes over the gait cycle, they take the difference between every pair of consecutive frames and combine it with denoised GEI.

One of the most challenging tasks of gait recognition is handling multi-view angle problem, i.e., when the view angle of the probe gait is not the same as that of the gallery gait. Zheng et al. [4] proposed a robust, easy-to-implement, and rapid method that transforms the feature of the gait in the probe view into that of the gait in the gallery view.

Another approach to gait recognition is based on deep learning and does not use handcrafted features. All features are trained inside the neural network on their own. Convolutional neural networks are now very popular in different problems concerned with video recognition and achieve high results. Recent advances in deep learning, especially recurrent neural network (RNN) and Long Short-Term Memory (LSTM) models [5] [7], provide some useful insights of how to tackle this problem. According to the philosophy underlying the deep learning approach, if we have a reasonable end-to-end model and sufficient data for training it, we are close to solving the problem. Recent approaches advanced from 2D image classification to 3D video classification. Karpathy et al. [1] use a multi-resolution, fovea architecture applying 3D convolutions on different time frames of a video. Donahue et al. [2] developed a hybrid architecture, concatenating a CNN with a Long Short-Term Memory (LSTM) network, where the CNN embeds the single frames into feature vectors and the LSTM classifies sequences of these vectors. Similar to [1], Tran et al. [3] designed a CNN using 3D convolutions, but with a deeper structure fully exploiting spatio-temporal features for video classification.

III. Preliminaries

CNNs have proved to have an outstanding performance as regards solving computer vision problems thanks to their ability to learn spatial features. However, they do not encode temporal characteristics explicitly. An example of a machine learning technique that learns temporal features is a Long Short-Term Memory Network. Combining them gives us a network that can do both tasks.

A. LSTM

Long Short-Term Memory Networks are a special kind of RNN, capable of learning long-term dependencies. RNNs have units whose output is connected not only to the next layer but also to the unit itself as an input.

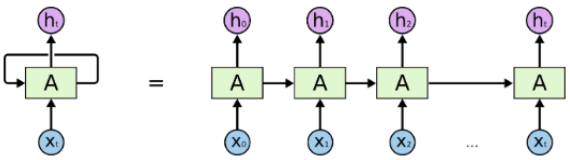

Figure 1 An unrolled recurrent neural network

For general-purpose sequence modeling, LSTM, as a special RNN structure has proven stable and powerful for modeling long-range dependencies in various previous studies [5] [6]. LSTMs expand this model by including a new element in the neurons called the memory cell c_t. The purpose of this is to act as an accumulator of previous inputs x_{t-1}, thus enabling the unit to remember the information it has already processed. Through the use of gates, the unit can decide whether it should accumulate the new input x_t, in the memory cell c_t and also whether it should forget the previous state c_{t-1} and should propagate the memory cell c_t to the output h_t (hidden state). Multiple LSTMs can be stacked and temporally concatenated to form more complex structures.

The key equations of the implementation used are shown in Eqn.1.

$$
\begin{aligned}
i_t &= \sigma(W_i x_t + b_{ii} + U_i h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_f x_t + b_{if} + U_f h_{t-1} + b_{hf}) \\
g_t &= tanh(W_g x_t + b_{ig} + U_g h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_o x_t + b_{io} + U_o h_{t-1} + b_{ho}) \\
c_t &= f_t \circ c_{t-1} + i_t \circ g_t \\
h_t &= o_t \circ tanh\,(c_t)
\end{aligned}
$$

Eqn. 1 Implementation equations of LSTM

B. Convolutional LSTM

Convolutional LSTM is an extension of LSTM networks designed to handle spatio-temporal data. This architecture is appropriate for problems that have a temporal structure in their input such as the order of images in a video or words in a text.

ConvLSTM can be viewed as a convolutional layer embedded with an LSTM cells. Compared with conventional LSTM, ConvLSTM replaces the matrix vector multiplication with the convolution operation, which significantly reduces the number of parameters to learn while capturing the spatial relation between voxels more efficiently.

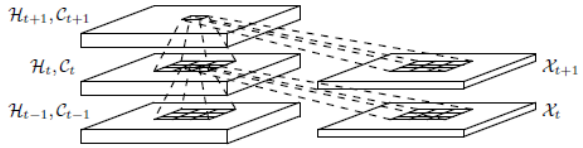

Figure 2 Inner structure of ConvLSTM

The formulas of ConvLSTM are shown in Eqn. 2. Here, we denote X_t as the input tensor, H_t as the hidden state tensor and C_t as the cell state tensor.

To ensure that the states have the same number of rows and same number of columns as the inputs, padding is needed before applying the convolution operation. If we perform zero-padding (which used in this work) on the hidden states, we are actually setting the state of the outside world to zero and assume no prior knowledge about the outside.

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_t + W_{ci} \circ C_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_t + W_{cf} \circ C_{t-1} + b_f) \\
o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_t + W_{co} \circ C_{t-1} + b_o) \\
C_t &= f_t \circ C_{t-1} + i_t \circ tanh(W_{xc} * X_t + W_{hc} * H_t + b_g) \\
H_t &= o_t \circ tanh(C_t)
\end{aligned}
$$

Eqn. 2 Implementation equations of ConvLSTM

IV. Experiments

A. Dataset

To evaluate our method, we used the CASIA-B [8] gait dataset. CASIA Dataset is a large database consisting 124 subjects in total, 10 carrying conditions for each subject, (6 normal walk, 2 carrying a bag and 2 wearing a coat) taken from 11 different angles (0°, 18°, 36°, …, 180°). For training 5 normal, 1 carrying and 1 clothed subject are selected. And the remaining 3 conditions are used in test set.

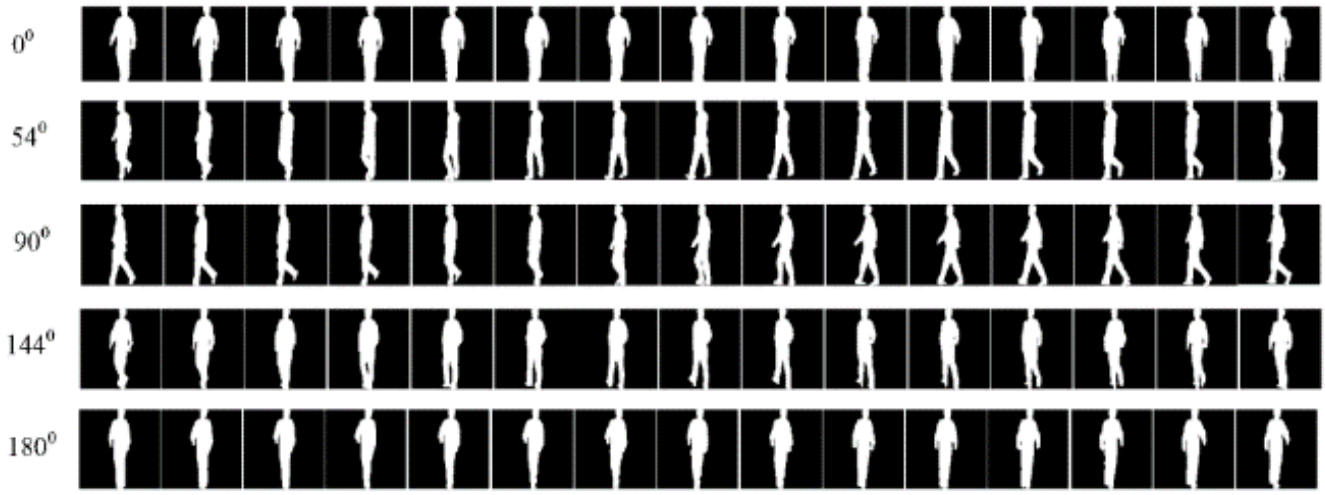

Figure 3 Silhouettes of gaits of one person from various view angles

Since each original sequence will probably have a different length and ConvLSTM requires a fixed size input, each sequence is limited to be 50 frames. To ensure that the states have the same number of frames as the inputs, 2-D zero-padding is performed. Each frame has been resized to 32x32 to lower the computational load.

Under these conditions our training data shape is (868, 50, 32, 32, 11) with 868 being 7 carrying conditions of 124 subjects, 50 is frame number followed by 32x32 image size and 11 angles as 11 different channels. Test data shape is (372, 50, 32, 32, 11) in the same manner.

B. Experimental Setup

In this model, ConvLSTM was used to capture the temporal dependency in the dataset. It not only can establish the timing relationship like LSTM, but also obtain local spatial features like CNN. We used Keras with our work, which is known for its simplicity in creating neural networks. Keras is a high-level API capable of running on top of TensorFlow.

In our network, two different types of architecture is used. First type is the ConvLSTM layer integrated with batch normalization, max pooling and dropout layers. Other one is the fully-connected architecture, which involves dense layers, activations and dropout layers. For the transition in between them, a flatten layer is used. With many trial runs, the best rates of learning have achieved with 2 ConvLSTM layers and 2 fully-connected layer followed by the output layer.

Due to max pooling layers having (2, 2) pooling sizes, image size is halved by height and weight. Our input images are 32x32 pixels and taking into consideration of the parameter size, somewhere between 4-10 pixels are ideal for the fully-connected layers. Thus, our network has two max pooling, hence two convolutional LSTM layers. Before going for the fully-connected layers, multi-dimensional data needs to be single-dimensional, meaning reshaping the data in a 1-D array. Figure 4 shows the entire network that has used in this work.

Looking into some of the hyper-parameters, as the activation function for convolutional and dense layers Rectified Linear Units known as relu's are used. For the LSTM layers Hyperbolic Tangent known as tanh's are used. Output layer has the Softmax activation function. In each layer L2 Regularization is used in the form kernel, bias and recurrent regularizers defined in the Keras API. Adam optimizer is used with 10^{-4} learning rate and 10^{-5} decay rate. Dropout rate is 0.5 for the entire network.

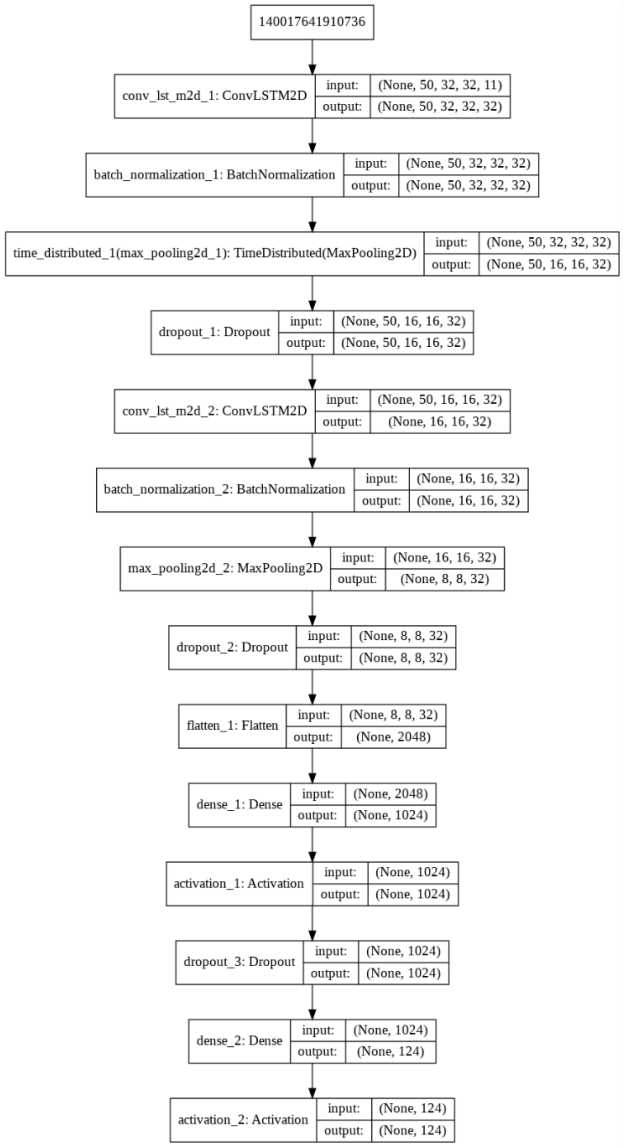

Figure 4 Summary of the model

As can be seen in Figure 5, training accuracy is around 95 percent and the loss is going down as intended. In the test runs, our network has very good test accuracy, over 90 percent.

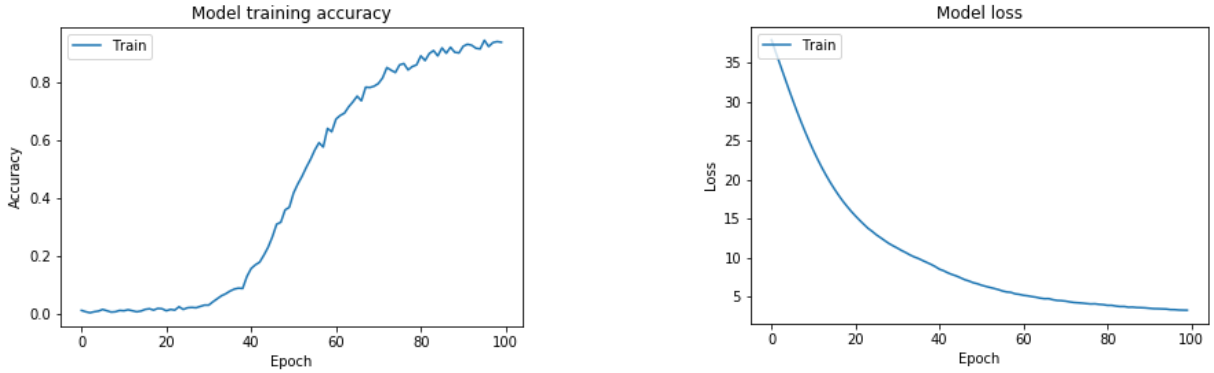

Figure 5 Training accuracy and training loss

V. Conclusion

In this paper, we present a novel feature learning method for gait recognition. A Convolutional LSTM based gait recognition method is studied, with an empirical evaluation in terms of pre-processing approaches and network architectures. The obtained results demonstrate that by using the discriminative power of the deep learning methods, we can achieve good recognition results using unsupervised learning. Comprehensive experiments on the CASIA-B dataset demonstrate the effectiveness of our method for gait recognition.

References

[1] A. Karpathy et al., "Large-scale video classification with convolutional neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1725–1732.

[2] *J. Donahue et al., "Long-term recurrent convolutional networks for visual recognition and description," arXiv preprint arXiv:1411.4389, 2014.*

[3] *D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3d: generic features for video analysis," arXiv preprint arXiv:1412.0767, 2014.*

[4] *S. Zheng, J. Zhang, K. Huang, R. He, and T. Tan, "Robust view transformation model for gait recognition," in 2011 18th IEEE International Conference on Image Processing (ICIP), Sept 2011, pp. 2073–2076.*

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

[6] *R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In ICML, pages 1310–1318, 2013.*

[7] *J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In CVPR, 2015.*

[8] *Yu, S.; Tan, D.; and Tan, T. 2006. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In ICPR, volume 4, 441–444.*