Learning Temporal Features Using LSTM-CNN Architecture for Face Anti-spoofing

Zhenqi Xu Shan Li Weihong Deng
Beijing University of Posts and Telecommunication
No 10, Xitucheng Road, Haidian District, Beijing, PR China
{xuzhenqi,ls1995,whdeng}@bupt.edu.cn

Abstract

Temporal features is important for face anti-spoofing. Unfortunately existing methods have limitations to explore such temporal features. In this work, we propose a deep neural network architecture combining Long Short-Term Memory (LSTM) units with Convolutional Neural Networks (CNN). Our architecture works well for face anti-spoofing by utilizing the LSTM units' ability of finding long relation from its input sequences as well as extracting local and dense features through convolution operations. Our best model shows significant performance improvement over general CNN architecture (5.93% vs. 7.34%), and hand-crafted features (5.93% vs. 10.00%) on CASIA dataset.

1. Introduction

Face anti-spoofing has addressed increasing attentions because of its importance for face recognition systems. Diverse spoofing methods, including warped photo attacks, cut photo attacks, video attacks, *etc*., make this problem difficult. Figure 1 shows sample images from CASIA dataset [20]. Only images in the first column are real.

This problem can be seen as a binary classification problem giving input faces. The temporal features which describes the faces' dynamic structure is helpful for face anti-spoofing. In this work, we propose a new architecture to learn temporal features from video sequences.

Learning a global description for the videos' temporal evolution is important for face anti-spoofing. Recurrent neural networks (RNN) are usually used for learning patterns from such time sequences. But the general RNN architecture may suffer from optimization problem of exponential decay of gradient information [8]. Thus, we implement a recurrent neural network with Long Short Term Memory (LSTM) [7] units. The LSTM units can discover long-range temporal relationships from the input sequences by using input gates, output gates, forget gates to control modifying, accessing and storing the internal states. We put the LSTM layer above a convolutional neural network (CNN) architecture, by doing this, we can leverage the local and dense property from convolution operation and learn the temporal structure by storing information in LSTM units.

(a) (b) (c) (d)

Figure 1. Sample images from CASIA dataset. (a)Genuine faces. (b)Warped photo attacks. (c)Cut photo attacks. (d)Video attacks.

Our contributions can be summarized as follows:

1) We explore deep learning architectures for face anti-spoofing, and find that these end-to-end features perform better than hand-crafted features by a large margin.

2) We propose LSTM-CNN architecture to learn the temporal structure from videos, and show that temporal features is helpful for face anti-spoofing.

3) We confirm that the background information is also useful for distinguishing the genuine attempts from fake attacks.

We have done experiments on CASIA dataset, and achieve state-of-art performance.

2. Related works

To solve face anti-spoofing problem, traditional methods are developing hand-crafted features, then feeding them to a binary classifier, usually Support Vector Machines (SVM) or Linear Discriminant Analysis (LDA). The performance of these methods mainly depend on the discriminative fea-

978-1-4799-6100-9/15/$31.00 ©2015 IEEE

tures devised.

The traditional methods can be roughly categorized into three groups: texture-based, motion-based and multi-spectral-based. Besides, one may integrate different features to improve performance either in feature level or score level [18, 14]. The texture-based features contains LBP [13], HOG [11], DoG [17], *etc*. Pereira *et al*. [4] used LBP-TOP to extract spatial and time domain features from three orthogonal planes. For motion-based features, eye blinking [16] or lip movement [10] are used for face anti-spoofing. Marsico *et al*. [6] also explore movements based on 3D projective invariants. Though real face and fake face may look similar under visible light, they may be distinguished under other spectrum [14]. The multi-spectral-based features can be then extracted.

Different from traditional methods, deep neural networks can extract powerful end-to-end features directly from raw data. This kind of deep representation are discriminative, generalized well if training data is sufficient, and has been proved efficient in many other vision fields. Yang *et al*. [19] explored CNN architecture to extract features and use SVM to get the label of the input image.

Most of these works use one single image [11, 17, 13, 19] or stacked images as input, and they can not model the temporal structure in image sequences well. The idea of extracting temporal feature for face anti-spoofing is to treat it as video classification, which receives a sequence of images and classifies them into real attempts or fake attacks. LBP-TOP [4] is one of such methods. But it's limited because of splitting the input sequences into several planes. The temporal structure also exists in pixels that not in the same plane. Yang *et al*. [19] stacks several images as input for CNN architecture, and get no better results than using a single image. The CNN architecture itself can not extract temporal features.

To extract temporal features from video sequences, Baccouche *et al*. [1] apply LSTM units on top of Scale-invariant Feature Transform (SIFT) features and a Bag-of-words (BoW) representation for action classification. In [15], CNN architecture with various pooling methods and LSTM-CNN architecture are used for video classification, they also integrated raw pixels with optical flow images to capture more motion features.

3. Method

Instead of using single image, we treat face anti-spoofing as video classification problem. We put a LSTM layer on top of CNN, thus to extract temporal relationship from different video frames. The input of our model is the sequence of video frames $(x_1, x_2, ..., x_n)$, and the output is a binary number y indicating whether the input sequences are real. To achieve this, we model the conditional probability of the

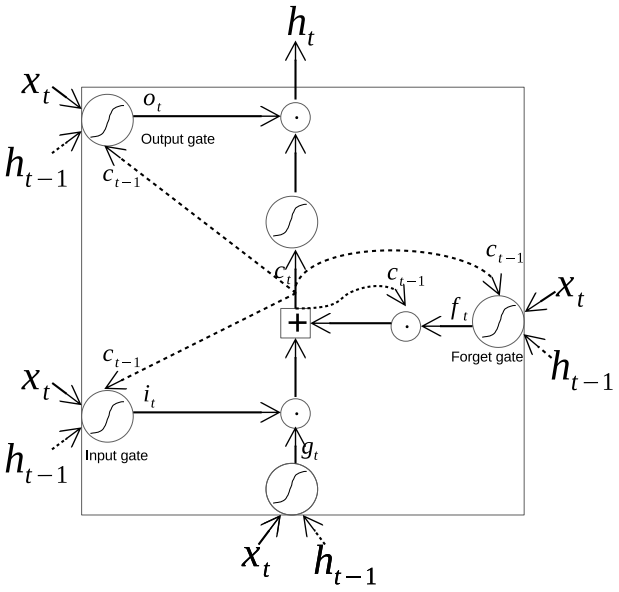


Figure 2. LSTM unit illustration. Each circle with a curve means a non-linear transformation. Circle with a dot is element-wise product operation. Square with a plus sign is element-wise plus operation. The dashed arrow means connection from the last time step.

output y giving the input $(x_1, x_2, ..., x_n)$ *i.e.*

$$p(y|x_1, x_2, ..., x_n) \tag{1}$$

3.1. LSTM units

The LSTM units [8] are known having the ability to learn long range dependency from the input sequences. Each LSTM unit have a memory cell(c_t) to store the internal state and three additional gates to control the behaviour between the memory cell, the input(x_t, with t denote the time step) and the output(h_t). The three gates are input gate(i_t), output gate(o_t) and forget gate(f_t), see Figure 2. We used LSTM units with peephole connection described in [7]:

$$\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) && (2)\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) && (3)\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) && (4)\\
g_t &= \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) && (5)\\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t && (6)\\
h_t &= o_t \odot \tanh(c_t) && (7)
\end{aligned}$$

Where σ operator is sigmoid function, g_t is the non-linear transformation of inputs, W and b are model parameters.

Equ. 6 shows that the forget gate controls how much the last internal state contributes to the internal state now. If i_t equals 0, the last internal state is forgotten. The input gate controls how much the input influence the internal state

by multiplying the cell's non-linear transformation of inputs g_t. The cell memory is updated by adding these two parts together. Equ. 7 shows that the output gate decides how much the internal state to transfer to the unit output. Outputs and cell memories from last time step are connected to the three gates, making the LSTM unit to capture the temporal relation from the input sequences, which is important for face-anti-spoofing.

3.2. LSTM-CNN architecture

For comparison, we firstly implement a CNN architecture, which contains two convolutional layer with max pooling after each, one fully connected layer, one dropout layer, and one softmax layer to predict the probability whether an input is real or fake, see Figure 3. We find that the use of dropout layer can significantly prevent over-fitting thus improving performance.

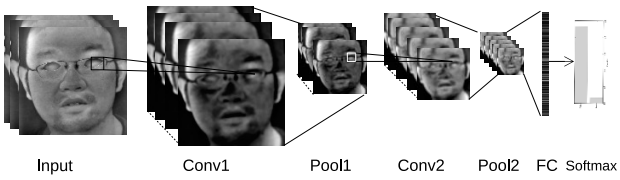


Figure 3. CNN architecture we use for comparison.

To learn the temporal structure from input sequences, we put a LSTM layer between the fully connected layer and softmax layer, which forms our LSTM-CNN architecture. Different from CNN architecture, there are connections between nodes in LSTM layer. This architecture can be seen as a deep architecture through time steps with the CNN parts sharing the same parameters, see Figure 4. The total parameters is a little greater than CNN architecture, but with more powerful representation ability. We only use one LSTM layer, and find it performs well for face anti-spoofing problem. Adding more layers doesn't help a lot.

When dealing with video classification, the architecture in [15] predicts a class at each time step, then uses pooling methods to get the final answer. Here, we stack all the outputs each time steps, and classify them with a single softmax layer since the number of our input frames is fixed. The way we use is simple and efficient for face anti-spoofing.

Our architecture learns end-to-end features from video sequences. The features are extracted layer by layer, from simple representation to complex concepts. The bottom convolutional layers can extract features locally and densely (while the images are locally relevant). The fully connected layer recombines the representations learned by convolutional layer and reduces the dimension. Then, the LSTM layer learns temporal structure from these representations. The final result was obtained by softmax layer.

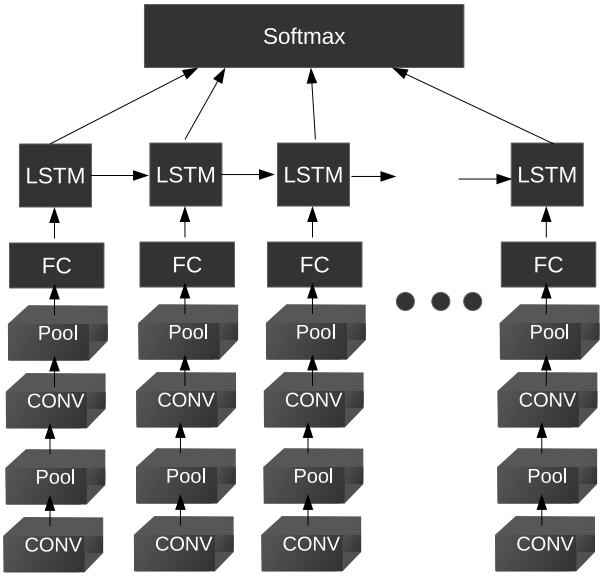


Figure 4. LSTM-CNN architecture unrolled in time.

3.3. Training

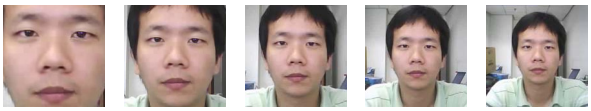
We use caffe toolbox [9] to optimize our model on a computer with a Nvidia K40c GPU card. Our two models share almost the same optimization parameters. We use Stochastic Gradient Decent (SGD) with a momentum of 0.9 and use the truncated BPTT [7] to compute gradient for LSTM layer. The learning rate is 0.001 for several iterations and when the loss seems no descending, we lower the learning rate by a factor of 10 to stabilize the parameters. To reduce over-fitting, we use L2 norm to penalize the parameters with a weight decay of 0.01. The mini-batch we use is 100 for CNN architecture and 32 for LSTM-CNN architecture, since the LSTM-CNN architecture consumes more memory.

4. Experiments

We have done multiple experiments on CASIA dataset [20]. This dataset contains totally 50 subjects (20 for training, and 30 for testing). For each subject, 12 videos are provided under three video qualities and four kinds of videos: the real attempt, warped photo attacks, cut photo attacks and video replay attacks.

4.1. Data preprocessing

The raw data in the video contains a lot of background information, and varies in size and length. To make them available for our model to train, we first detect face locations using a common face detector Viola-Jones in OpenCV. The face detector may fail to find faces in some video frames. For CNN architecture, we crop faces based on the bound-



(a) 1.0 (b) 1.4 (c) 1.8 (d) 2.2 (e) 2.4

Figure 5. Faces in varies scales.

ing box and simply remove the bad faces. For LSTM-CNN architecture, we treat a number of video frames as a sample and extract overlapping samples from videos. The bounding box of a sample is the minimum bounding box that contains all the face regions of its frames.

Following [19], we also explore the use of background information for classifying. We scale the face bounding box by factors of $\{1.0, 1.4, 1.8, 2.2, 2.6\}$, see Figure 5. In the processing of scaling, the bounding box may exceed the border of video frames, and we fill these pixels with zero. All of the images are resized into $128 * 128$ before feeding into neural networks.

4.2. Model details

The CNN architecture contains two convolutional layers, the first one has 48 filters, and the second one has 96 filters. All filters are square of 3 pixels, and the stride is 1 pixel. The pooling layer after each convolutional layer is max pooling with size of 2 pixels and stride of 2 pixels. The fully connected layer has 1000 neurons, and each activation is set to 0 with a probability of 0.5 during training. The non-linear function we use for these three layers is rectified linear units (ReLU) [12]. For LSTM-CNN architecture, we put the LSTM layer after the fully connected layer. The LSTM layer has 30 internal cells for each time step. And we have experiments with time steps of $\{3, 5, 7, 9\}$.

4.3. Model Selection

The CASIA dataset only gives train set and test set. To make fully using of the data, we split the train set into five folds. Note that the test set and the train set have no common subjects, and so the validation fold. Another concern is that each fold should contain the similar ratio of real samples and fake ones to prevent the unbalance problem.

We train models on these five folds, and using the average loss of validation folds to tune the model parameters. And then using the best parameter to retrain a model on the whole train set. The final result is reported on the test set using the model.

4.4. Evaluation criteria

To make fair comparison with other anti-spoofing methods, we report two kinds of evaluation criteria, one is Half Total Error Rate (HTER) on test fold using the average Equal Error Rate (EER) on validation folds to set threshold, the other is EER on test set. We use the probability $p(y = 1|x)$ from the softmax output as our score.

Table 1. Basic results(%) on CASIA dataset.

Method	EER(val)	HTER(test)	EER(test)
Correlation [3]	26.65	30.33	—
LBP [2]	16.00	18.17	—
DoG[20]	—	—	17.00
LBP-TOP [5]	—	—	10.00
CNN(ours)	6.09	7.34	6.20
LSTM-CNN(ours)	**5.04**	**5.93**	**5.17**

4.5. Results

The basic results can be summarised in Table 1. We can see that our CNN architecture and LSTM-CNN architecture can learn a good discriminant representation of the input faces. The best HTER of our CNN architecture is 7.34% (EER is 6.20%), and 5.9% (EER is 5.17%) for our LSTM-CNN model, which is a large improvement compared to the hand-crafted features. By extracting temporal features from input sequences, our LSTM-CNN model performs better than general CNN model.

We also explore how the areas of background information influence the performance of our models with scales of $\{1.0, 1.4, 1.8, 2.2, 2.6\}$. The result details is showed on Table 2. We can see that the background information is useful for face anti-spoofing.

Table 2. Test HTER(%) on differernt scales and different time steps. LSTM-CNN(n) means LSTM-CNN architecture with n input images.

Scale	1.0	1.4	1.8	2.2	2.6
CNN	8.97	7.89	7.42	7.86	7.34
LSTM-CNN(3)	8.85	7.51	6.94	7.35	11.47
LSTM-CNN(5)	8.85	7.34	6.80	6.34	10.93
LSTM-CNN(7)	8.81	7.52	**5.93**	6.11	9.25
LSTM-CNN(9)	8.90	7.62	6.25	6.81	9.56

The HTER of our models is robust to the score threshold, see Figure 6. The score threshold doesn't influence HTER much. It shows that the deep neural network method tends to make strong predictions for whether the input is real or fake.

5. Conclusions

We propose a deep neural network architecture for face anti-spoofing. By putting a LSTM layer above a convolutional architecture, our model can extract features locally and densely as well as exploring the temporal structure from

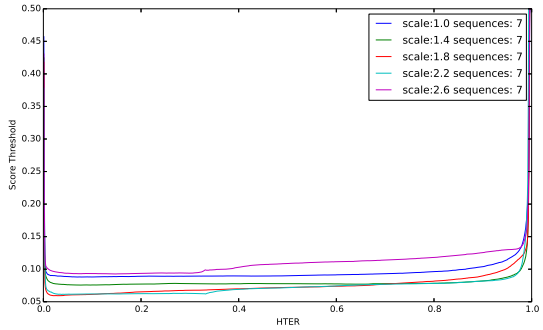


Figure 6. HTER vs. Score Threshold.

input sequences. Our model is different from most of the existing methods, whose input is a single image. The experiments on CASIA dataset show that the LSTM-CNN architecture performs better than general CNN architecture and traditional hand-crafted features.

6. Acknowledgements

This work was partially sponsored by supported by the NSFC (National Natural Science Foundation of China) under Grant No. 61375031, No. 61573068, No. 61471048, and No.61273217, the Fundamental Research Funds for the Central Universities under Grant No. 2014ZD03-01, This work was also supported by the Beijing Higher Education Young Elite Teacher Program, and the Program for New Century Excellent Talents in University.

References

[1] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In *Artificial Neural Networks–ICANN 2010*, pages 154–159. Springer, 2010. 2

[2] I. Chingovska, A. Anjos, and S. Marcel. On the effectiveness of local binary patterns in face anti-spoofing. In *Biometrics Special Interest Group (BIOSIG), 2012 BIOSIG-Proceedings of the International Conference of the*, pages 1–7. IEEE, 2012. 4

[3] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel. Can face anti-spoofing countermeasures work in a real world scenario? In *Biometrics (ICB), 2013 International Conference on*, pages 1–8. IEEE, 2013. 4

[4] T. de Freitas Pereira, A. Anjos, J. M. De Martino, and S. Marcel. Lbp- top based countermeasure against face spoofing attacks. In *Computer Vision-ACCV 2012 Workshops*, pages 121–132. Springer, 2013. 2

[5] T. de Freitas Pereira, J. Komulainen, A. Anjos, J. M. De Martino, A. Hadid, M. Pietikäinen, and S. Marcel. Face liveness detection using dynamic texture. *EURASIP Journal on Image and Video Processing*, 2014(1):1–15, 2014. 4

[6] M. De Marsico, M. Nappi, D. Riccio, and J.-L. Dugelay. Moving face spoofing detection via 3d projective invariants. In *Biometrics (ICB), 2012 5th IAPR International Conference on*, pages 73–78. IEEE, 2012. 2

[7] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with lstm recurrent networks. *The Journal of Machine Learning Research*, 3:115–143, 2003. 1, 2, 3

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1, 2

[9] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014. 3

[10] K. Kollreider, H. Fronthaler, M. I. Faraj, and J. Bigun. Real-time face detection and motion analysis with application in liveness assessment. *Information Forensics and Security, IEEE Transactions on*, 2(3):548–558, 2007. 2

[11] J. Komulainen, A. Hadid, and M. Pietikainen. Context based face anti-spoofing. In *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, pages 1–8. IEEE, 2013. 2

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, volume 1, page 4, 2012. 4

[13] J. Määttä, A. Hadid, and M. Pietikainen. Face spoofing detection from single images using micro-texture analysis. In *Biometrics (IJCB), 2011 international joint conference on*, pages 1–7. IEEE, 2011. 2

[14] S. Marcel, M. S. Nixon, and S. Z. Li. *Handbook of biometric anti-spoofing: trusted biometrics under spoofing attacks*. Springer, 2014. 2

[15] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *arXiv preprint arXiv:1503.08909*, 2015. 2, 3

[16] G. Pan, L. Sun, Z. Wu, and S. Lao. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 2

[17] B. Peixoto, C. Michelassi, and A. Rocha. Face liveness detection under bad illumination conditions. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 3557–3560. IEEE, 2011. 2

[18] R. Tronci, D. Muntoni, G. Fadda, M. Pili, N. Sirena, G. Murgia, M. Ristori, and F. Roli. Fusion of multiple clues for photo-attack detection in face recognition systems. In *Biometrics (IJCB), 2011 International Joint Conference on*, pages 1–6. IEEE, 2011. 2

[19] J. Yang, Z. Lei, and S. Z. Li. Learn convolutional neural network for face anti-spoofing. *arXiv preprint arXiv:1408.5601*, 2014. 2, 4

[20] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li. A face antispoofing database with diverse attacks. In *Biometrics (ICB), 2012 5th IAPR international conference on*, pages 26–31. IEEE, 2012. 1, 3, 4