ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Graph for Transformer Feature: A New Approach for Face Anti-Spoofing

Quoc-Huy Trinh[1,2,3],Trong-Hieu Nguyen Mau[1,2,3], Xuan-Mao Nguyen [2], Minh-Van Nguyen [1,3], Hai-Dang Nguyen [1,3]

1- University of Science, VNU-HCM, Ho Chi Minh city, Vietnam
2- VNG Corporation, Ho Chi Minh city, Vietnam
3- Software Engineering Laboratory

Abstract. Face recognition is popular nowadays, however, Face anti-spoofing (FAS) poses a significant challenge for recognition systems due to the threat of external attacks. While many deep learning methods have been proposed to address this issue, they often face challenges in industry settings. Experiments found that patch extraction modules, such as the Vision Transformer and Swin Transformer, are effective for FAS in single images and perform well in industrial environments. From this point, we propose a model that leverages Transformer features and Graph Neural Networks to learn global information and identify correlations between patch features, which are critical for FAS.

1 Introduction

Face Recognition is a popular system nowadays, which has been in several products [1]. Moreover, applications based on face recognition are used widely for authentication as timekeeping, authentication for shopping, and authentication for security in buildings or offices, which leads to many problems related to security occurring if using face recognition. On the other hand, one of the most detrimental is using spoof media for cheating on face recognition systems [1]. Face Anti-Spoofing (FAS) is the method proposed to prevent these attacks from human factors to cheat on the authentication of the face recognition system [2].

Graph Neural Network is the famous geometric method to find the present information in Deep Learning [3]. A graph of nodes is used for the classification or related task. In fact, a graph can present the correlation of the features [3] in various nodes that are connected which is very important in Face Anti-Spoofing. Because the attack feature is sometimes implicit, some Deep Learning methods that use Convolution Neural Networks try to capture the correlation information by each window slicing, but the feature output is also difficult for classification. From that point, our idea is to use a Graph to represent the feature of each patch of an image, and the Transformer encoder with Patches embedding is the reasonable method for extracting patches from images and also extracting the information from them.

In this paper, our method focuses on inference on a single image, which is different from the State of the Art while the input of those methods is the video. By inference on a single image, the model is easier for deployment in the industrial. The main contribution is listed below:

- Introduce Graph Neural Network application for Transformer features.
- Evaluate the effectiveness of Graph correlation on the Face Anti-Spoofing task.
- A new approach to Face Anti-Spoofing problems by using Graph Neural Network, which is efficient but has an acceptable computation cost.

2 Graph for Transformer Feature

2.1 Patches Embedding

Patches Embedding [4] is a technique to divide the input image into smaller patches. The images with size $N \times N \times C$ are split into multiple patches with size $patches\_size \times patches\_size \times C$ for each patch. These patches are then flattened and embedded into a sequence of tokens. In our implementation, the patch size is 16.

2.2 Transformer Encoder

Our Transformer Encoder is constructed based on the Transformer encoder from the Vision Transformer [5]. It takes in a sequence of tokens with dimensions $B \times N \times C$, where B is the batch size, N is the number of tokens, and C is the channel of the token projection. The output feature size is $B \times N \times C_{new}$, which captures global information and long-range dependencies within the sequence. Each token is treated as a key, query, and value to compute Scale Dot-Product Attention [6], and the attention maps of all tokens are fused and concatenated to form a sequence of attention maps. A skip connection is used to concatenate the output with the long-range dependency information, and the Multi-Head Dot-Product Attention formula (shown in formula.1) with the attention weight $W \in R$ is applied to create the output of the self-attention mechanism.

$$MHA(q, k, v) = [head_0, head_1, ..., head_{n-1}]W \tag{1}$$

** MHA is the Multi-Head Attention*

The process of calculating each $head_i$ using the Scale Dot-Product Attention (as shown in formula.2) [6] involves the use of input vectors q, k, v, (in this case $q = k = v$ for Self-Attention of each token), and the number of tokens in the sequence, denoted by n. The output of the attention is then passed through a Layer Normalization layer and a Fully Connected layer with the Gelu activation function to create a token sequence that can capture the global information of the sequence.

2.3 Graph Correlation Layer

Graph Structure [3] is required for a Graph Neural Network layer [3]. Graph structure can be created manually by human knowledge, but in some cases - like

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

ours, the graph structure is not defined. To solve this problem, our idea is to use Self-Attention (originating from Scale Dot Product Attention) to construct the attention map $N \times N$ (with N representing the number of rows and columns) to present the graph structure. The formula.2 illustrates the construction of the graph structure matrix.

$$Graph\_Structure\_Matrix = Attention(q, k, v) = Softmax(\frac{qk^T}{\sqrt{d_k}})V \tag{2}$$

In our implementation, we use the scale dot product as a self-attention mechanism, where the input tensor is denoted by X. We obtain the query, key, and value vectors by multiplying X with weight matrices $W_q, W_k, W_v \in R$ in the attention mechanism. Specifically, $q = X \times W_q$, $k = X \times W_k$, and $v = X \times W_v$. We assume that the input dimensions of q, k, and v are all d_k, and we apply a scale layer by dividing the dot product formula $q.k = \sum_{i=0}^{d_k-1} q_i k_i$ by $\sqrt{d_k}$ to scale the values to variance = 1.

2.4 Graph Neural Network Module

In order to represent the correlation between the tokens, we utilize the Graph Correlation Layer which takes the output sequences from the Transformer encoder and constructs a graph input ($G_{input} \in R^{N \times N}$) to carry the correlation information. The input sequence is considered as a graph, represented by an adjacency matrix ($A \in R^{N \times C}$). The output of this layer is a graph correlation matrix updated ($G_{update} \in R^{N \times 1}$) that represents the correlation information of each token in the graph.

To create the correlation matrix, each graph is formed by taking the dot product of the adjacency matrix with a weight matrix ($W \in R$) that represents the weight of the correlation in the graph. Formula.3 provides an illustration of how the correlation output is constructed from the graph and the graph structure matrix.

$$G_{update} = W \times (G_{input} \cdot A) \tag{3}$$

During the update weights phase, the graph correlation is updated to reflect the new graph structure and the updated correlation information between the tokens. This process allows for the extraction of implicit correlation information from each patch of the image.

2.5 General architecture

Our architecture, depicted in Fig.1, begins by dividing an input image into n patches (in our implementation, n is 100). Each of these patches is treated as a token and fed into a Transformer Encoder, which includes Positional Embedding

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

to preserve position information. The resulting token sequences are then passed to a Graph Neural Network module, which treats each token as a node in a graph. The Graph Correlation output is combined with the encoded patches output from the Transformer by using a skip connection. This combination of correlation and global information is used to determine whether an image is real or fake. We scale our model with several depths, our best model is just about 10.84 million parameters.

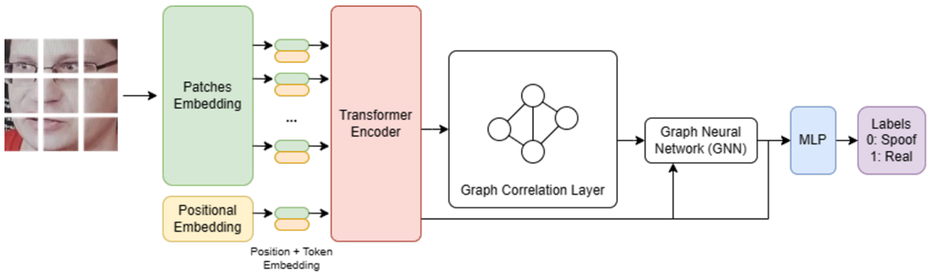

Fig. 1: Architecture of Graph for Transformer

3 Experiment

3.1 Implementation details

Our dataset for training includes two classes, which are classified as either spoof (0) or real (1). To produce relevant benchmark results, we train our model independently on different datasets. Input images first are normalized in the range [0,1], then Our models are trained with augmentations and optimized using the Adam optimization algorithm with a learning rate of 1e-4. To prevent the model from getting stuck in a local cost minimum due to imbalanced data, we apply the Cosine Annealing learning rate schedule during experiments. Training and inference are performed on a single NVIDIA Tesla V100 16GB with a batch size of 64 and 50 epochs. Additionally, we employ early stopping and reduced learning rates to improve results. After several experiments, the best-performing model was achieved after 2 hours of training.

3.2 Result

In Face Anti-Spoofing evaluation, we use Equal Error Rate (EER), Attack Presentation Classification Error Rate (APCER), and Area Under the Curve (AUC) as metrics to compare the performance of our model with other state-of-the-art models, following the benchmark of CelebA Spoof dataset [7]. EER is the threshold value at which the False Accepted Rate (FAR) and the False Rejected Rate (FRR) intersect. FAR is the ratio of false positives to the sum of false positives and true negatives, while FRR is the ratio of false negatives to the sum of true positives and false negatives.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

The Equal Error Rate (EER) is a common evaluation metric used in Face Anti-Spoofing. It is calculated by finding the point where the False Accepted Rate (FAR) and Face Rejection Rate (FRR) are equal on the threshold. The False Accepted Rate is the proportion of fake samples that are classified as genuine among all the fake samples, while the False Rejected Rate is the proportion of genuine samples that are classified as fake among all the genuine samples. The FAR is calculated by dividing the number of false positives by the sum of false positives and true negatives, and the FRR is calculated by dividing the number of false negatives by the sum of true positives and false negatives

Qualitative Results: We evaluated our proposed method on two datasets, CelebA Spoof [7] and LCC FASD [8], and achieved competitive results comparable to State of the Art models. On the CelebA Spoof dataset, our method achieved an AUC score of 0.9882, an EER score of 0.17, and an APCER score of 6, which were similar to MN3 large and better than MN3 small, but lower than AE Net. On the LCC FASD dataset, our method demonstrated comparable performance with an EER score of 0.8, an AUC score of 0.95, and an APCER score of 1.1. These results demonstrate the potential of our model to detect various attacks, including those originating from smartphones and electronic devices.

Model	EER (%)	AUC	APCER (%)
Ours	**12.23**	**0.833**	**8.9**
MN3 large [8]	16.13	0.921	17.26
AE Net (2020) [8]	20.91	0.868	12.52
MN3 small [8]	18.7	0.889	14.79

Table 1: Results on LCC FASD dataset

Model	EER (%)	AUC	APCER (%)
Ours	**1.7**	**0.9882**	**0.6**
MN3 large [9]	2.26	0.998	1.21
AE Net (2020) [9]	1.1	0.9988	0.23
PTA-LLL (2022) [10]	None	0.9785	2.53
MN3 small [9]	3.84	0.994	1.47

Table 2: Results on Celeb A Spoof dataset

4 Conclusion

In conclusion, our method tackles Face Anti-Spoofing by combining correlation information from the graph and global information with long-range dependencies from the Transformer encoder. This approach achieves results comparable to State of The Art methods and offers a unique perspective by capturing the correlation information from image patches, which holds valuable information but is often implicit. Our method improves liveness detection by enhancing the correlation of image patches. Additionally, this approach opens up new possibilities for Face Anti-Spoofing by finding the difference in correlation between real and spoof images through the graph, a reasonable and explicit method compared to traditional methods such as Convolution Neural Networks or other high-computational cost approaches.

Acknowledgement

This research is supported by research funding from the Faculty of Information Technology, University of Science, Vietnam National University - Ho Chi Minh City.

References

[1] Zhongyuan Wang et al. Masked face recognition dataset and application. *arXiv preprint arXiv:2003.09093*, 2020.

[2] Ajian Liu et al. Cross-ethnicity face anti-spoofing recognition challenge: A review. *IET Biometrics*, 10(1):24–43, 2021.

[3] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 2008.

[4] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022.

[5] Dosovitskiy et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Ashish Vaswani et al. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[7] Yuanhan Zhang et al. Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations. In *Computer Vision–ECCV 2020: 16th European Conference, Proceedings, Part XII 16*, pages 70–85. Springer, 2020.

[8] Denis Timoshenko et al. Large crowdcollected facial anti-spoofing dataset. *2019 Computer Science and Information Technologies (CSIT)*, pages 123–126, 2019.

[9] Yuanhan Zhang et al. Celeba-spoof challenge 2020 on face anti-spoofing: Methods and results. *arXiv preprint arXiv:2102.12642*, 2021.

[10] Kostiantyn Khabarlak. Post-train adaptive mobilenet for fast anti-spoofing. *arXiv preprint arXiv:2207.13410*, 2022.