Multi dimensional feature fusion based on histogram covariance matrix features and spectrogram

August 25, 2024

1 Team details

- Team name: despairingjuvenile
- leader kaggle username: despairingjuvenile
- 565027448@qq.com
- just leader: despairingjuvenile Email: 565027448@qq.com
- https://github.com/hopelessjuvenile/multi-ffdv
- SCAU, Guangzhou, Guangdong Province, China

2 Solution details

2.1 Base information

- Solution name: Multi dimensional feature fusion based on histogram covariance matrix features and spectrogram
- Validation score: 0.9997227206, rank: 20
- Final testing score: 0.6512, rank: 22

2.2 Method

The overall framework of the solution is as follows:

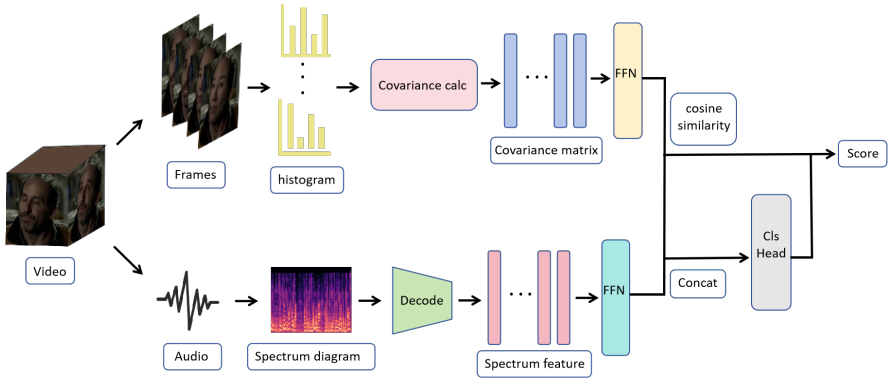

Figure 1: Overall framework of solutions

In the input stage of the model, the input video will be divided into video frames and audio, and the audio will be extracted into mel spectrograms, while the video frames will be counted as a histogram matrix H_v. For spectrograms, ResNet18 is used for feature extraction to obtain preliminary spectral image features S_{mel}.

The histogram matrix will be calculated as a covariance matrix $C_v \in \mathbb{R}^{n\times n}$, which is symmetric, the calculation method is as follows:

$$C_v = \frac{1}{n-1}\left(H_v - \mu_{H_v}\right)\left(H_v - \mu_{H_v}\right)^T \tag{1}$$

Where μ_{H_v} represents the mean of H_v, with each row representing the covariance between the i -th frame and the remaining n-1 frames. Considering the need to extract the top-k frames with the smallest covariance (frames with the possibility of forgery), it is necessary to sum the C_v in the column dimension, sort them in reverse order, record the index of the smallest top-k value, and finally obtain the covariance matrix $C_v^k \in \mathbb{R}^{k\times n}$ with the smallest top-k value, representing the preliminary forgery score corresponding to the video, the calculation formula of C_v^k as follows:

$$C_v^k = C_v[argmin_k(sum(C_v)_{dim=1})] \tag{2}$$

S_{mel} and C_v^k both use a feedforward neural network to obtain S'_{mel} and $C_v^{k'}$, respectively, for dimension alignment and preparation for feature fusion.

Feature fusion is divided into two branches. The first branch is the fusion of classification heads, where S_{mel} and C_v^k perform a simple concatenating and feed it into the classification head to obtain a preliminary classification score. The other branch calculates the cosine similarity Cos_f between S'_{mel} and $C_v^{k'}$.

In the feature fusion stage, S'_{mel} and $C_v^{k'}$ are not aligned in the time dimension, but consider the degree of matching between video and audio globally. Finally, the cosine similarity and classification score are added by weight to obtain the final forgery score. The calculation formula of i-th sample as follows:

$$Score^{(i)} = MLP\left(\left[S_{mel}^{\prime(i)}, C_v^{k\prime(i)}\right]\right) + \alpha\left(\frac{S_{mel}^{\prime(i)} \cdot C_v^{k\prime(i)}}{\left\|S_{mel}^{\prime(i)}\right\|\left\|c_v^{k\prime(i)}\right\|}\right) \tag{3}$$

Where α is an adjustable hyperparameter weight.

3 Multi-dimensional Facial Forgery Detection Analysis

3.0.1 Solution analysis

The effectiveness of video features on deepfakes is largely positively correlated with the sampling rate. Low sampling rates carry the risk of avoiding forged frames, while extracting visual features at high sampling rates can result in significant resource consumption. Therefore, it is particularly important to find methods with high sampling rates and low resource consumption.

Histogram statistics of video frames is a good method to save computational resource consumption. A frame can be represented by a vector of only $C_f \in \mathbb{R}^{1\times256}$. For n frames of video, a complete video can be represented by $C_v \in \mathbb{R}^{n\times256}$ without using convolution or other projection methods. However, its disadvantage is that it reflects less feature information and requires reasonable utilization of limited histogram feature information.

For videos with forgery, their forgery can be roughly classified into one category, which is the modification of faces in different ways. This modification will affect the distribution of characteristic pixels to a certain extent, and this impact will be intuitively reflected in the histogram. For audio and video, the possibility of forgery is largely reflected in the correlation between audio and video.

In order to find the differences between the histograms corresponding to different video frames, we calculated the covariance matrix of the histogram matrix corresponding to the video. The covariance was used to calculate the differences between the histograms of different frames. The larger the covariance, the more positive the correlation between frames. The closer the covariance is to 0, the linear independence between frames. The negative and smaller the covariance, the negative correlation between frames.

3.0.2 Features / Data representation

As mentioned above, a covariance matrix is used to explore facial differences between video frames. The effectiveness of this feature needs to be discussed here. By visualizing the covariance size between different frames in the same video, we can explore whether small covariance sums correspond to fake frames, as shown in the following figure:

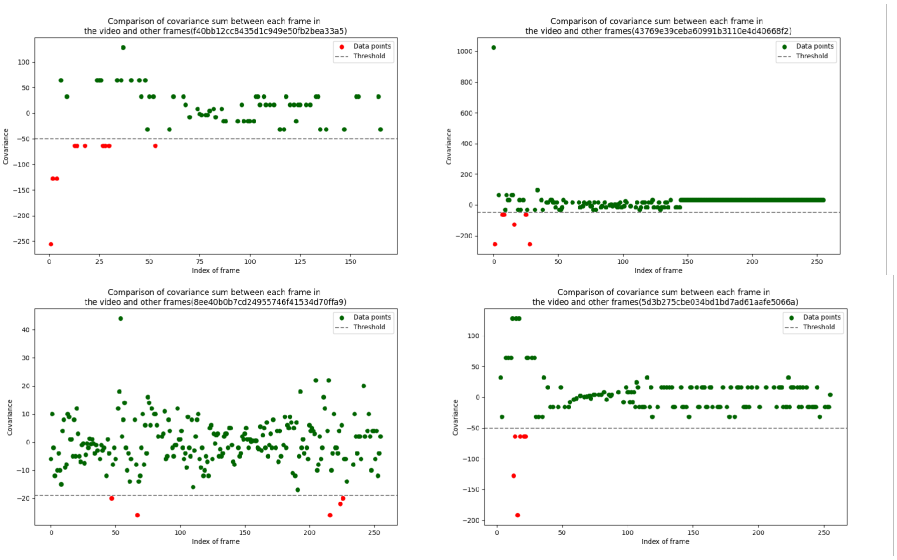

Figure 2: Visualization of covariance between different frames in individual forgery examples

The above figure selects four forgery examples, each point representing the sum of covariance between each frame and the remaining frames in the covariance matrix of the video. The selected forgery frames (red dots) in the covariance matrix are visualized through threshold segmentation. It can be seen that the selected fake frame regions for the covariance matrix are correct. However, due to the low resolution of the multi-ffdv dataset, the pixel stability is relatively small, which may cause some pixel disturbances in low-resolution situations. The threshold of the covariance matrix for each video segmentation is also different, and the model needs to learn the distribution of forged frame covariance.

3.0.3 Data Fusion Strategies

The preliminary video and audio features mentioned above will go through two fusion branches after passing through the feedforward neural network. The first is the classification head of concatenation, and the second is the calculation of cosine similarity between video and audio. The feature dimension changes as follows:

The preliminary spectrogram features have a dimension of (n, 512) and covariance features of (n, k, k), and are summed up in the last dimension to obtain a dimension of (n, k). (n) is the number of samples, and the dimensions of the two features are equal after passing through two feedforward neural networks, both of which are (n, 128), where 128 is the dimension output by the network.

Put the concatenated video and audio features into the first fusion branch

classification header. The dimension of the overlapped video and audio features is (n, 256). After passing through the classification header, it becomes (n, 2). The cosine similarity of the other fusion branch calculates the similarity between the two features, resulting in a dimension of (n, 1). Finally, the cosine similarity is added to the classification score by weight through broadcasting to obtain the final score.

3.0.4 Learning strategy

The first step in the learning process of the model is to save the video and audio features. The video is saved as a Numpy array of (k, k) and saved to a. npy file, while the audio is pre saved as a. jpg file with the spectrogram. The feature saving in the first step does not involve any parameter calculations, only the calculation of covariance matrix and spectrogram. Then, each training session can directly call the pre saved video and audio feature files, which can improve training efficiency.

3.0.5 Training description

The model was trained on RTX3060, Intel (R) Xeon (R) CPU E5-2696 v3 processor, and 14G memory. The feature file size of the multi ffdv dataset is shown in Table.1.

Table 1: Different sub dataset feature folder sizes

Mode	video	audio
Transet	23.9GB	5.04GB
Valset	7.92GB	1.65GB
Testset	17.59GB	3.71GB

After extracting the features of the spectrogram and covariance matrix, the average training time for each epoch is 446 seconds, the model size is only 44M (file size).

During the training, the influence of α parameters on the results was adjusted in the validation set, and the influence of k value of covariance matrix top k on the results was also verified, as shown in Table.2, Table.3.

Table 2: The impact of different α values on the resultss

α	0.5	1.0	2.0	5.0
AUC	0.9892	0.9997	0.9945	0.9942

Table 3: The influence of different top k values on the results

k	64	96	128
AUC	0.9812	0.9913	0.9997

4 Global Method Description

For feature extraction of spectrograms, a pre-trained model of ResNet18 ImageNet-1K was used to obtain preliminary features. During the training process, a certain amount of data augmentation was applied to the spectrogram, and the spectrogram was randomly cropped and standardized with a probability of 0.5. For the characteristics of the covariance matrix, data augmentation is also required. The input covariance matrix is perturbed with a probability of 0.5, and the formula for matrix perturbation is as follows:

$$C_v^{k+} = \left[\left((C_v^k + (\gamma R_{cov})(\gamma R_{cov})^T) + (C_v^k + (\gamma R_{cov})(\gamma R_{cov})^T)^T\right)\right]/2 \quad (4)$$

Where γ is the perturbation parameter, R_{cov} is the random number matrix, and C_v^k is the covariance matrix of the input.