Appending Adversarial Frames for Universal Video Attack

Zhikai Chen
Xi'an Jiaotong University
zhikai_chen@outlook.com

Lingxi Xie
Huawei Noah's Ark Lab
198808xc@gmail.com

Shanmin Pang
Xi'an Jiaotong University
pangsm@xjtu.edu.cn

Yong He
Xi'an Jiaotong University
hy0275@stu.xjtu.edu.cn

Qi Tian
Huawei Noah's Ark Lab
tian.qi1@huawei.com

A. Supplementary Material

This Supplementary Material provides additional algorithm details and more experimental results.

- In Sec. A.1, we provide model details in attacking phase (*e.g. the parameters of attack setting*).
- In Sec. A.2, we provide more training details for reproducing video classification models.
- In Sec. A.4, we provide the result of ablation study about appending position. We tried to insert frames in the front, middle and end, and we obtain very similar results, which are shown in Tab. 5.
- In Sec. A.3, we provide an illustration of the attacked video generated by our method, which is shown in Fig. 5.
- In Sec. A.5, we provide the results of A^{2}F under different adversarial frames to prove that the attack performance of our method does not influence by the pattern of adversarial frames, which are shown in Tab. 6.
- In Sec. A.6, we provide the experiment results of targeted attack, which are shown in Tab. 7.
- In Sec. A.7, we provide additional results on UCF-101 dataset under a special spatial mask attack setting to demonstrate that we can generate more imperceptible adversarial examples, which are shown in Tab. 8 and the corresponding visualization results are shown in Fig. 6.
- In Sec. A.8, we provide the results of the transferability of perturbations across models with A^{2}F-AM on HMDB-51, which are shown in Tab. 9.

A.1. Attacking Details

The max number of iterations for A^{2}F-AV and A^{2}F-AM is $10 \times N$ (the number of testing videos) and $5 \times K$ (the number of ensemble models), respectively. While the max number of iterations for all the other evaluated methods is 20 and the threshold ϵ for the magnitude of adversarial perturbations is 0.001. For A^{2}F-FS, the feature extractor to measure the distance of internal layer representation is ResNet50. The step size of perturbation $\epsilon = 0.01$.

A.2. Training Details

The evaluated models are trained on a workstation with 4 Titan-X GPUs (each 11GB memory). At the data preprocessing stage, the input frames of videos are resized to 224×224 and the value range is transformed from [0,255] to [0,1]. We only use the first 28 frames of videos for model training and evaluation. We randomly divide the trimmed videos into a training set and test set, where the ratio of the number of training examples and the number of test examples is $9:1$. We use one-hot encoding to represent different classes. There are some differences among models:

- C3D: This model used in our paper contains 3D convolutional layers, and followed by batch normalization layer with RELU activation. We set 0.2 as a dropout rate to avoid overfitting and set the learning rate to 1e-4. The batch size is 16 and we trained it 15 epochs totally.
- CNN+LSTM: This model contains two parts. The first part is a normal 2D convolutional network (ResNet50) in our paper. Then we use the LSTM model to copy with the temporal domain. Due to the restriction of memory, we set batch size 5 here and use 40 epochs totally to train it.
- I3D-ResNet: The base model is ResNet50 and the batch size is 16. We train it 15 epochs totally.
- I3D-Inception: This model is the same as I3D-ResNet. The difference is that the base model is the Inception model. Due to the low rate of convergence, we load the pre-trained model and fine-tuned it at the target dataset.
- ResNet3D: This model is similar to C3D, and its base

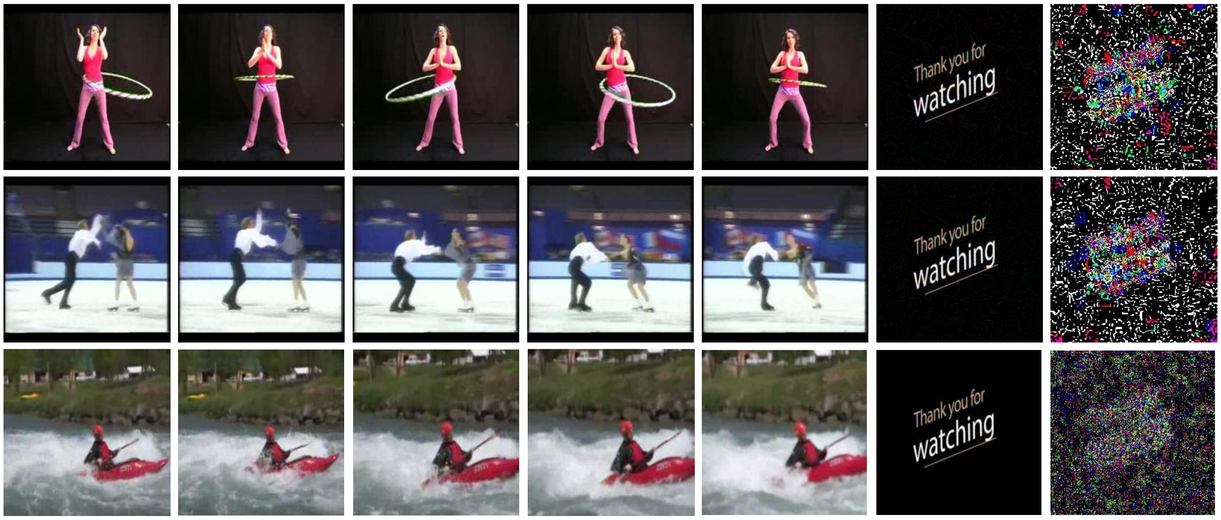

Figure 5. Three adversarial videos are generated by A^{2}F with Resnet3D. Top-5 columns are original videos and the sixth column is their corresponding adversarial frames(the adversarial frame is already attacked by adding perturbations shown in the last column). The last column shows the perturbations, in which the amplitudes are enlarged by 255 times for better visualization.

model is ResNet50. We load the 2D parameters pretrained on ImageNet for the ResNet50.

- P3D: We use P3D63 in our experiment. The batch size is 10 here and we train it for 15 epochs.

A.3. An illustration of Attacked Videos

We first apply our attack method on a single network. We appending a communal frame with adding the perturbation. Figure. 5 shows three example adversarial videos generated by A^{2}F.

A.4. Attack performance with Appending Position

There are 28 frames for each original video, and for extensive evaluation, we append adversarial frames in six different positions shown in Tab. 5. As it shows, our method is robust to appending positions. That is, the accuracy of models as well as the fooling rate are almost the same with the appending frames in different positions.

A.5. A^{2}F under Different Adversarial Frames

The goal of this section is to investigate the relationship between the adversarial frames and attack performance of A^{2}F. Those different adversarial frames are shown in the first row of Fig. 6, which defined as 'TFW1', 'TFW2', and 'TFW3', respectively. They represent different patterns. The left figure is a zoom in the version of the adversarial frame we mentioned before, the middle is the adversarial frame with a white background which is the reverse of the former adversarial frame and the right one is an adversarial frame with different font size and different font style. These new appending frames are shown in very different patterns and the attack performance from these frames almost can cover all possible of existing adversarial frames because we use different backgrounds and different fonts. See Tab. 6 for the performance of A^{2}F under different adversarial frames on UCF-101. There is nearly no performance gap between different adversarial frames, and therefore it is easy to conclude that the patterns of ending frames will not necessarily influence the performance of A^{2}F.

A.6. Targeted Attack

We evaluate the performance of our method attack specified target labels. We separately choose 5 different labels as our target attack labels from UCF-101 and HMDB-51. With each target label, we set other categories as the testing set. Then, we randomly choose one video for each test class, and thus, we obtain new testing set with 100 videos in UCF-101 and another testing set with 50 videos in HMDB-51. The experimental results are shown in Tab. 7. As we can see, our method is inferior to BA in terms of FR for some cases. This is mainly because the targeted attack needs a specific gradient direction to proceed, so the constrained attack direction may not help the adversarial attack accurately to find the right way to achieve a specific classification region. However, with the same or nearly the same fooling rate, A^{2}F usually has smaller AAP than BA, which further demonstrates A^{2}F can help to generate smaller adversarial perturbations in basic attack settings.

A.7. Special Spatial Mask Attack

As we mentioned before, we can construct arbitrary shape perturbations by changing shapes of spatial masks to generate a more imperceptible perturbation. Empirically

Table 5. Appending position w.r.t accuracy (%) and fooling rate (%) of different video models.

Models	0		5		10		15		20		24	
	ACC	FR	ACC	FR	ACC	FR	ACC	FR	ACC	FR	ACC	FR
I3D-ResNet	57.7	100	57.7	100	57.7	100	57.7	100	57.7	100	57.7	100
I3D-Inception	94.9	100	94.9	100	94.9	100	94.9	100	94.9	100	94.9	100
CNN+LSTM	34.5	100	31.5	100	34.5	100	35.6	100	34.0	100	32.5	100
C3D	50.9	94.2	50.9	100	50.9	100	50.9	100	50.9	100	50.9	100
ResNet3D	83.7	96.8	83.7	96.8	83.7	97.4	83.7	98.4	83.7	96.8	83.7	94.2
P3D	58.8	100	58.8	100	58.8	100	58.8	100	58.8	100	58.8	100

Table 6. Performance of A^2F under different adversarial frames on UCF-101.

Target Model	TFW1		TFW2		TFW3	
	FR (%)	AAP	FR (%)	AAP	FR (%)	AAP
I3D-ResNet	100	0.04	100	0.04	100	0.05
I3D-Inception	100	0.09	99.5	0.09	99.5	0.10
CNN+LSTM	100	0.02	98.9	0.02	100	0.02
C3D	95.2	0.15	96.3	0.15	97.3	0.17
ResNet3D	97.4	0.09	96.8	0.09	97.3	0.10
P3D	100	0.02	100	0.02	100	0.02

Table 7. Comparison of BA and A^2F for targeted attack.

Target Model	Methods	UCF-101		HMDB-51	
		FR (%)	AAP	FR (%)	AAP
I3D-ResNet	BA	97.6	0.29	**97.8**	0.31
	A^2F	**97.7**	**0.17**	**97.8**	**0.14**
I3D-Inception	BA	**84.6**	0.23	**96.8**	0.27
	A^2F	27.4	**0.08**	40.2	**0.08**
CNN+LSTM	BA	**61.6**	0.23	**55.8**	0.27
	A^2F	53.2	**0.07**	42.4	**0.07**
C3D	BA	**97.9**	0.30	**97.8**	0.31
	A^2F	83.8	**0.26**	95.0	**0.22**
Resnet3D	BA	**98.1**	0.28	**98.0**	0.30
	A^2F	**98.1**	**0.15**	**98.0**	**0.13**
P3D	BA	**98.0**	0.22	**97.8**	0.26
	A^2F	97.8	**0.07**	**97.8**	**0.08**

Table 8. Performance of A^2F with special perturbation under different adversarial frames on UCF-101. The percentage in brackets is the spatial rate of the spatial mask. For instance, 'TFW1 (16%)' denotes that the pixels of the text region occupy 16% in the adversarial frame 'TFW1'.

Target Model	TFW1 (16%)		TFW2 (16%)		TFW3 (11%)	
	FR (%)	AAP	FR (%)	AAP	FR (%)	AAP
I3D-ResNet	74.7	0.001	82.4	0.001	84.8	0.001
I3D-Inception	1.5	0.003	4.0	0.003	2.0	0.002
CNN+LSTM	88.6	0.001	87.8	0.001	84.8	0.001
C3D	28.8	0.003	6.8	0.004	25.5	0.002
ResNet3D	14.3	0.004	17.6	0.003	19.4	0.003
P3D	91.4	0.003	86.2	0.001	86.5	0.001

speaking, the perturbation adding to the abundant texture areas can make the perturbation more imperceptible, so we make the mask that filters the background and keeps the font to be attacked. We visualize three examples of adversarial frames and their corresponding spatial masks, adversarial frames as well as spatial perturbations (as shown in Fig. 6). See Tab. 8 for more details in attacking a fixed model with a specific spatial perturbation.

A.8. Experimental Results on HMDB-51

We provide the results of the transferability of perturbations across models with A^2F-AM on UCF-101 as mentioned in Sec. 4.5. The experimental results are shown in Tab. 9, which indicates the attack transferability across models on HMDB-51 dataset remains robust and powerful.

Table 9. Comparison of BA and A^{2}F-AM in transferability across models on HMDB-51 dataset. The first column indicates we use the Leave-One-Out ensemble method that excludes one model to produce perturbations. For instance,' −I3D-ResNet' means the corresponding ensemble model excludes I3D-ResNet. The numbers in the 3-8 columns are the fooling rates (%) for each attacked model.

Models	Method	I3D-ResNet	ResNet3D	P3D	I3D-Inception	C3D	CNN+LSTM
−I3D-ResNet	BA	2.1	91.5	100	6.1	64.0	63.0
	A^{2}F-AM	**19.1**	95.7	100	2.0	58.0	61.7
−ResNet3D	BA	100	2.1	100	12.2	64.0	67.4
	A^{2}F-AM	100	**6.4**	100	6.1	58.0	61.7
−P3D	BA	100	87.2	15.2	6.1	64.0	60.4
	A^{2}F-AM	100	93.6	**93.5**	2.0	58.0	60.0
−I3D-Inception	BA	100	87.2	91.3	**0.0**	64.0	54.2
	A^{2}F-AM	100	93.6	100	**0.0**	48.0	60.4
−C3D	BA	100	87.2	87.0	90.0	0.0	54.2
	A^{2}F-AM	100	93.6	100	4.1	**4.0**	62.5
−CNN+LSTM	BA	100	87.2	100	6.1	64.0	36.7
	A^{2}F-AM	100	93.6	100	2.0	58.0	**44.7**

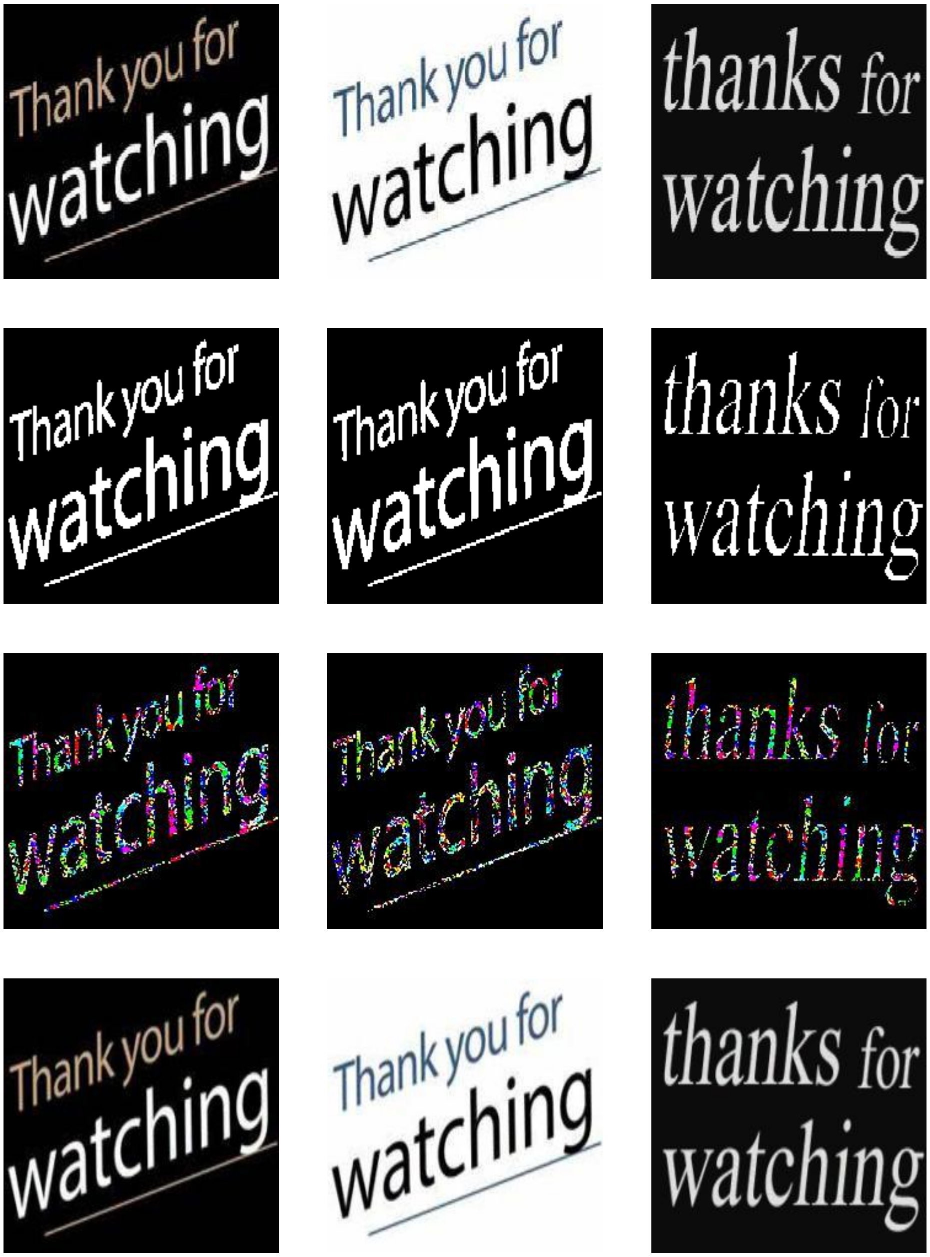

Figure 6. Three examples of adversarial frames and its corresponding spatial mask, adversarial frames and spatial perturbations. The first row is adversarial frames without perturbation. The second row is the corresponding spatial mask for filtering the background to make the attack focus on the font. The third row represents the corresponding perturbation with a certain font spatial mask (amplify with $\times 255$ for better visualize). The last row is the final adversarial frame appending to original videos.