PAT: Pseudo-Adversarial Training For Detecting Adversarial Videos

Nupur Thakur, Baoxin Li
Arizona State University
{nsthaku1, baoxin.li}@asu.edu

Abstract

Extensive research has demonstrated that deep neural networks (DNNs) are prone to adversarial attacks. Although various defense mechanisms have been proposed for image classification networks, fewer approaches exist for video-based models used in security-sensitive applications like surveillance. In this paper, we propose a novel yet simple algorithm called Pseudo-Adversarial Training (PAT), to detect the adversarial frames in a video without requiring knowledge of the attack. Our approach generates 'transition frames' that capture critical deviation from the original frames and eliminate the components insignificant to the detection task. To avoid the necessity of knowing the attack model, we produce 'pseudo perturbations' to train our detection network. Adversarial video detection is then achieved through the use of the detected frames. Experimental results on UCF-101 and 20BN-Jester datasets show that PAT can detect the adversarial video frames and videos with a high detection rate. We also unveil the potential reasons for the effectiveness of the transition frames and pseudo perturbations through extensive experiments.

1. Introduction

Deep neural networks (DNNs) have proven to be excellent learners for various tasks like image classification and video action recognition. Recent studies have also shown the vulnerability of DNNs to adversarial attacks [5, 7, 18, 29]. A typical adversarial attack example is when the input is altered subtly, leading to misclassification by the DNN. This has drawn a lot of attention as DNNs are being used for applications like surveillance [28], autonomous vehicles [20, 37], facial recognition [13, 23], etc., where resilience to adversarial attacks is of utmost importance.

While the current literature documents extensive research related to adversarial learning in image-based applications, [1–3, 11, 18, 27, 36]. the adversarial vulnerability of video-based DNNs remains a less explored area. This poses a pressing practical challenge, as DNNs are widely deployed in various video-analysis tasks [4, 6, 19, 24, 31, 32]. Although the video-based DNNs are more difficult to attack due to the additional temporal dimension, [33] showcased the susceptibility of video action recognition models to adversarial attacks.

An adversarial video is a video with one or more adversarial frames. The attacks on video models can be categorized into two types - 1) sparse attack where either minimum number of frames are perturbed or minimum amount of perturbations are added to each frame like [33], 2) dense attack where perturbations are added to all the frames like in [12].

There are plenty of defenses designed to defend image attacks [15, 17, 21, 25, 30]. However, in general, they cannot be directly applied to videos as they do not take temporal information into account. Often, defenses for videos need to be computationally efficient, in order to handle the volume of data. The temporal redundancy present in videos can lead to a significant waste of computation if one simply applies an image-based approach on a frame-by-frame basis. To overcome these challenges, we suggest that the detection of adversarial videos will be a better alternative as compared to techniques involving intensive training, changes to network parameters or reconstructing the frames [14].

In this paper, we propose a novel defense strategy, **P**seudo-**A**dversarial **T**raining (**PAT**), for video action recognition networks that can detect the adversarial frames without any prior knowledge of the perturbations. It is called 'pseudo' because the network is not trained on the actual perturbations. The detection network is trained on transition frames and pseudo perturbations to detect the perturbed frames. The transition frames, constructed from the neighboring frames, make the otherwise finely blended perturbations noticeable. The pseudo perturbations mimic the actual perturbations without being generated by any actual attack algorithm. They enable the network to learn about potential deviations from authentic frames, without the need to know specific attack models.

We summarize our contributions as follows:

- We propose a new technique, **P**seudo-**A**dversarial **T**raining (**PAT**) to detect the adversarial frames in a video. This strategy does not need prior knowledge of

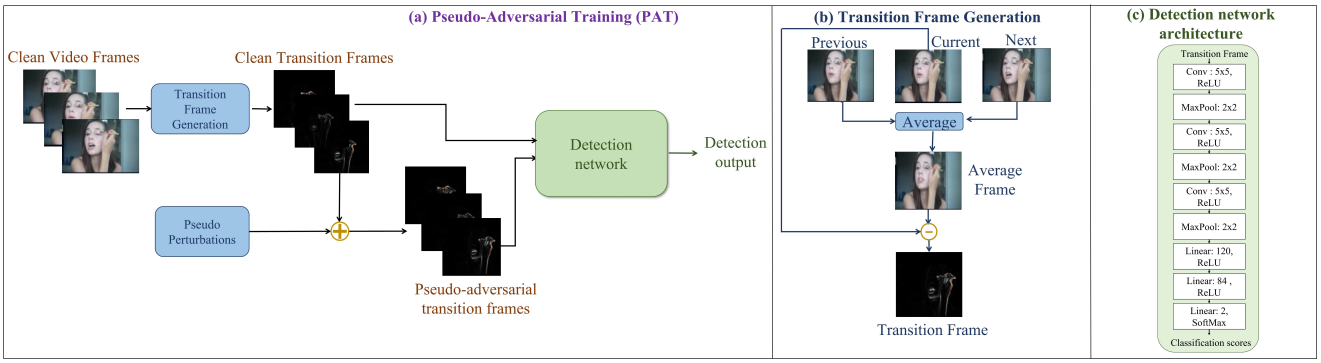

Figure 1. Overview of the Pseudo-Adversarial Training (PAT) approach. (a) shows the key components - transition frames generation, pseudo-perturbation generation and training the detection network using the clean transition frames and pseudo-adversarial transition frames. (b) shows transition frame generation. (c) shows the detection network architecture.

the attack scheme or the added perturbations and can defend both the sparse as well as the dense attacks.

- We define and use the transition frames so that the detection network can focus on perturbations that are more relevant to the detection of the adversarial frames rather than other elements in the frame.
- We also propose to generate pseudo perturbations and use them to train our detection network. These perturbations are generated such that the network can learn to handle varying perturbations due to adversarials.

2. Related Work

The fact that the video action recognition models are vulnerable to adversarial attacks was first studied in [33]. A video action recognition model aims to predict the label for an input video. Several deep learning models are used for this task like CNN+LSTM [4] etc. Such networks learn spatial and temporal information from the video input. [33] used $l_{2,1}$ optimization loss to generate the perturbations. They achieved temporal sparsity and a high fooling ratio using a temporal mask and propagation of the perturbations to the consecutive frames.

[22] makes use of a neural network to generate perturbations that are rich in detail and sparse. As these perturbations are highly specific, they are created for each video separately. [12] is a white-box attack that uses GAN structure to generate 'circular dual universal perturbations', with the discriminator being the target network. A post-processor is added between the generator and discriminator to perform a circular shift on the perturbations.

[10] is a black-box attack exploiting the pre-trained image models and black-box optimization techniques to minimize the queries to the target model and the search space. [34] is another black-box attack that takes a heuristic approach to key frames and regions to add the perturbations.

The focus on the spatial information of the images in the image-based defenses makes them non-optimal for the video-based models. [35] designed a defense for video-based models, where the optical flow is used to generate the current frame based on the previous. The video is then passed through the target network to check for temporal consistency.

[9] also uses the concept of temporal consistency to determine the perturbed frames in a video. They use a well-trained network to determine the labels of each frame and a frame is considered adversarial if its label is different from its adjacent frames. [14] presents a defense method that replaces the batch normalization layers in the action recognition network with their module named MultiBN. They adversarially train this modified network to defend against adversarial attacks.

PAT avoids the complex tasks of frame reconstruction and optical flow estimation by using the 'transition frames', which are computed by a much simpler process. PAT neither introduces any hyperparameters for detection (like that in [9]) nor requires retraining the target classifier (like that in [14]) and hence reduces the overhead of learning.

3. Methodology

3.1. Problem Definition

Without loss of generality, we consider video action recognition models in this work but it can be easily extended to other video-related tasks. Let X_1, X_2,...., X_t, X_{t+1} be the image frames of a video with X_t being the target frame. Let D be the classifier model and the prediction output of X_t be Y_t ($D(X_t) = Y_t$). An attacker aims to generate an adversarial frame X_t^* by adding a small perturbation ϵ to the target frame X_t such that $D(X_t^*) = Y_t^*$, where Y_t^* is the adversarial target output. Our aim is to detect the adversarial frame X_t^* without any knowledge of the attacking algorithm besides the two frames - X_{t-1} and X_{t+1}. Also,

there is no definite knowledge if X_{t-1} and X_{t+1} are clean or perturbed frames.

3.2. Pseudo-Adversarial Training (PAT)

The Pseudo-Adversarial Training (PAT) strategy consists of three key components: transition frame generation, pseudo perturbation generation and training our detection network to detect the adversarial frames, which are elaborated below and the overall framework is shown in Fig. 1.

3.2.1 Transition Frames Generation

A video is a sequence of frames depicting a (dynamic) scene. The temporal component plays an important role in video-related tasks. Assuming a reasonable frame rate and small and continuous motion, an approximate reconstruction of the frame is possible using the nearby frames. PAT leverages these facts to compute 'transition frames', which are used to capture the underlying dynamics across the frames while ignoring portions non-relevant to this detection task. Consider three consecutive frames X_{t-1}, X_t and X_{t+1}. We define 'motion' M_1 between X_{t-1} and X_t as $M_1 = X_t - X_{t-1}$. The term 'motion' is used in this narrow sense throughout this paper, and it is supposed to capture underlying dynamics. Similarly, the motion M_2 between X_t and X_{t+1} is $M_2 = X_{t+1} - X_t$. Now, the motion between M_1 and M_2 is given by -

$$M_2 - M_1 = X_{t+1} - X_t - X_t + X_{t-1} \tag{1}$$

Eq. 1 reduces to the transition frame equation (Eq. 2). These equations show how the transition frame can capture the motion from the three frames used to create it. It is generated using two simple operations only - average and difference, making it computationally inexpensive.

$$X_t^{tr} = \left(\frac{X_{t-1} + X_{t+1}}{2}\right) - X_t \tag{2}$$

There are two special cases - the first and the last frame of the video. The first frame does not have the previous frame and the last frame does not have the next frame. Thus, we replace the average frame for first and last frames by the second frame and the second last frame in the video respectively.

Fig. 2 shows the sample original frames, their average and the transition frames for the current frames. It is clear from the transition frame that it gets rid of most of the static background. It only contains the main object and the motion around it. Elimination of the passive elements does not hamper the detection process as they are not relevant to our detection task. As a result, the perturbations become visible prominently in the transition frames.

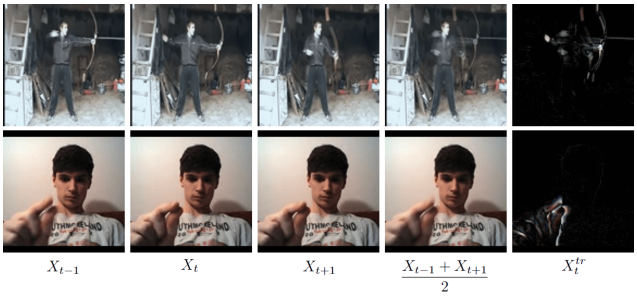

Figure 2. Sample original frames (first 3 columns), the average frame (fourth column) and the transition frame (fifth column) from UCF-101 (first row) and 20-BN Jester (second row) datasets.

3.2.2 Pseudo Perturbations Generation

For training the detection network, we need clean and adversarial samples. As there is no prior knowledge of the perturbations being added or the attack algorithm, we propose a way to generate on-the-go pseudo perturbations. These perturbations are not actual perturbations as they are not generated using an attack strategy. But, when added to the transition frames, they are enough to let the network learn about the actual perturbations in the video frames.

To generate the pseudo perturbations, we use a varying magnitude of Gaussian noise by changing the standard deviation σ of the Gaussian distribution. We generate Gaussian noise mask $X_n \sim \mathcal{N}(0,\, \sigma)$ where $\sigma \sim \mathcal{U}(0.0001,\, 0.05)$. It is of the same shape as that of the transition frame (Eq. 3, where $X_{p_{adv}}^{tr}$ represents the pseudo-adversarial transition frame and X_t^{tr} represents the target transition frame).

We chose this particular range for the value of σ because it covers a wide variety of magnitude. For the values below 0.0001, the noise does not make any significant impact on the image. The values above 0.05 completely disrupt the transition frame and turn it into complete noise. For every transition frame, a different value of σ is picked randomly. This varying noise mask added to the transition frames is essential to train the detection network such that it can identify a variety of perturbations.

$$X_{p_{adv}}^{tr} = X_t^{tr} + X_n \tag{3}$$

3.2.3 Training the detection network

We use a convolutional neural network (architecture shown in Fig.1 (c)) for binary classification as the detection network. The transition frames for the clean class are obtained from the original videos. For the adversarial class, the transition frames are obtained by adding the pseudo perturbations to the clean transition frames. Our approach is summarized in Algorithm 1. The adversarial video detection can be done using the detected adversarial frames (details

are in Section 4.2).

Algorithm 1: PAT: Pseudo-Adversarial Training

Input : Training Videos $\mathbf{X}$, Labels y_c and y_{adv}, Test Videos $\mathbf{X_{test}}$, Detection Network D
Output: Predicted Labels $\hat{y}$
for *each epoch* $e = 1, 2, ...$ **do**
  while *Training* **do**
    for *each video frame* $X_t = 1, 2, ...$ **do**
      $X_t^{tr} = \left(\frac{X_{t-1}+X_{t+1}}{2}\right) - X_t$;
      Generate $\sigma \sim \mathcal{U}(0.0001,\, 0.05)$;
      Generate $X_n \sim \mathcal{N}(0,\, \sigma)$;
      $X_{p_{adv}}^{tr} = X_t^{tr} + X_n$;
      Train D using cross-entropy loss;
    end
  end
  while *Testing* **do**
    for *each video frame* $X_{test_t} = 1, 2, ...$ **do**
      $X_{test_t}^{tr} = \left(\frac{X_{test_{t-1}}+X_{test_{t+1}}}{2}\right) - X_{test_t}$;
      $\hat{y} = D(X_{test_t}^{tr})$;
    end
  end
end

4. Experiments and Results

In this section, we start with a discussion of the experimental settings and the metrics used for evaluation. We also discuss the two attack baselines used to evaluate our technique. Lastly, we present the results of PAT on two datasets - UCF-101 [26] and Jester [16] dataset.

4.1. Experimental Settings

4.1.1 Datasets and Target Networks

We chose UCF-101 [26] and Jester [16] dataset to show that our approach works for both coarse-grained and fine-grained action recognition data. We use split 1 of UCF-101 and validation set of 20BN-Jester for our experiments.

The target network for the UCF-101 dataset is the CNN+LSTM classifier. The pre-trained ResNet152 is followed by a layer of LSTM and three fully-connected layers (final classification using SoftMax) and yields a test accuracy of 91.09%. The target network for the Jester dataset is a C3D classifier. We fine-tune the model of depth 18 pre-trained on Kinetics dataset available on GitHub repository [8]. Validation accuracy is 90.33%. The input frame resolution is 112×112 for both datasets. The sequence length is 40 frames and 16 frames for UCF-101 and Jester datasets respectively.

4.1.2 Attack Baselines

We consider two attacks to evaluate our approach. Sparse Adversarial Perturbations [33] perturbs only a small percentage of frames from the entire video using an $l_{2,1}$ optimization loss and temporal mask. As the perturbations are sparse, we refer to it as 'sparse attack' for the rest of the paper. For our experiments, we perturb pre-determined 22.5% (9 out of 40 frames) and 20% (4 out of 16 frames) of the total frames for UCF-101 and Jester datasets respectively. The other attack [12] perturbs all the frames in a video using a generative model. As all the frames are perturbed, we refer to it as 'dense attack'.

We chose these two baselines as they are strong attacks and represent two very different types of attacks - videos with only one or a few frames perturbed and videos with all the frames perturbed. With these attacks, we show that our method can detect adversarial frames containing a variety of perturbations.

4.1.3 Evaluation metrics

Two evaluation metrics are used in our experiments: 1) Frame Detection Rate -

$$FDR = \frac{\sum_{i=0}^{N} D(X_i) = Y_i}{N} \tag{4}$$

where X_i is the input frame, $Y_i \in \{0, 1\}$ is the ground truth for frame detection, D is Detection network and N is the total number of frames; 2) Video Detection Rate -

$$VDR = \frac{\sum_{i=0}^{M} \left(\prod_{j=0}^{N} D(X_{ij})\right) = Y_i}{M} \tag{5}$$

where X_{ij} is the j^{th} frame in i^{th} video, $Y_i \in \{0, 1\}$ is the ground truth for video detection, D is Detection network, N is the number of frames and M is the number of videos.

We also calculate Area under Receiver Operating Characteristic (ROC), shortly known as AUC on UCF-101 dataset for comparison purposes. Higher the Area under Curve (AUC), better is the capability of the model to distinguish between the two classes.

4.2. Adversarial Detection Results

Table 1 summarizes our results for adversarial detection on both the datasets. From the FDR column, it is clear that PAT detects adversarial frames from both types of attacks with high accuracy. This demonstrates that PAT can detect different types and magnitude of perturbations without having any prior knowledge about them. Also, PAT works well for both datasets showing that it can handle coarse-grained and fine-grained classification data. This makes it a very promising approach for detecting adversarial frames. Based on the detected adversarial frames, the adversarial videos

can be detected. The frame detection rate obtained by PAT is enough to detect most of the adversarial videos with ease.

Table 1. Adversarial Frame detection rate (FDR) and video detection rate (VDR) for PAT on UCF-101 and 20BN-Jester datasets for different attacks.

Attack Algorithm	FDR	VDR
UCF-101		
Sparse Attack	83.62%	92.86%
Dense Attack	83.46%	88.25%
20BN-Jester		
Sparse Attack	75.9%	80.7%

The sparse attack showed that even if one frame is perturbed, a success rate of 60% is achieved. So, even if one adversarial frame is detected in the entire video sequence, the video can be categorized as an adversarial one. However, to accommodate the scenario of having false positives, we use a threshold of 3 adversarial frames i.e we consider a video to be adversarial when atleast 3 frames are classified as adversarial by the detection network. See Table 1 for the adversarial VDR results for PAT.

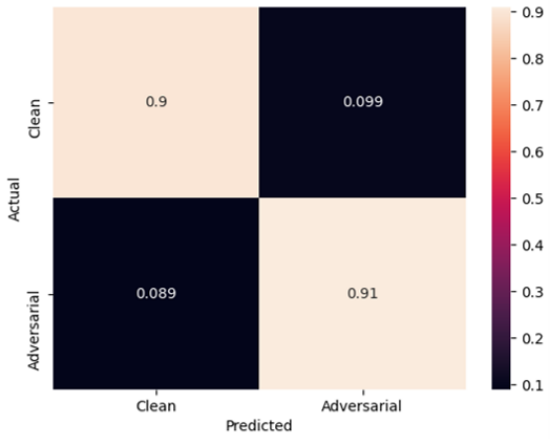

Figure 3. Confusion matrix of PAT for the adversarial video detection when the adversarial videos are generated by both sparse and dense attack.

We tried different values of this threshold to determine an adversarial video. We found empirically that a value of 3 for the threshold maintained a balance between false positives and false negatives. Higher values of threshold led to a higher number of adversarial videos being misclassified as clean which can pose a threat to the network. On the other hand, lower values of threshold led to a higher number of clean videos being classified as adversarial, which is not desirable too.

Fig. 3 shows the confusion matrix for detecting adversarial video using PAT on UCF-101 dataset. The adversarial videos contain a mix of videos generated by sparse and dense attacks. The high true positives and true negatives along with low misclassification of videos for both the classes indicate the ability of PAT to detect adversarial videos without any prior knowledge of perturbations.

Table 2. Comparison of PAT with other defenses (AUC) on UCF-101 dataset. The 1^{st} column denotes the defense mechanism. The last 3 columns denote different adversarial attacks. The second-last row corresponds to training the detection network on clean and adversarial videos generated by both the attacks.

Defense	Sparse	Dense	Sparse+Dense
Temporal+Spatial [9]	-	-	77%
AdvIT [35]	97%	-	-
PAT (Adversarial data)	93.4%	99.8%	99.8%
PAT	**97.6%**	**94.1%**	**94.2%**

We also evaluate PAT using Area under Receiver Operating Characteristic curve (AUC) for comparing with other baselines. Table 2 shows the AUC results and the ROC curve for PAT is displayed in Fig. 4. This curve indicates that our approach has a high capability of differentiating between clean and adversarial videos. PAT achieves an AUC of 94.2% on clean and adversarial (contains both sparse and dense attacks) videos.

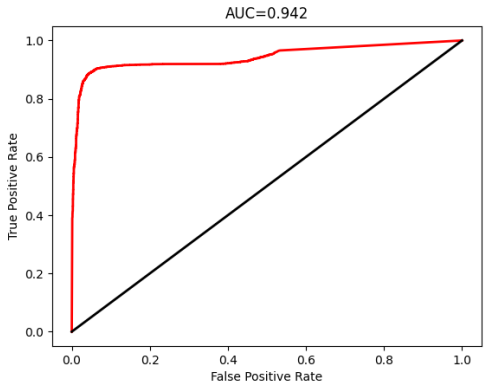

Figure 4. ROC Curve under different thresholds for performance of proposed method, PAT on clean and adversarial videos.

In Table 2, the first row determines the performance of PAT when it is trained using the clean and adversarial videos. It is not surprising that the performance on both the attacks is high in this case as the network is aware of the perturbations. PAT can achieve almost as good performance (94.2% AUC which is just $\sim 5\%$ lower) as the first case in Table 2, without actually knowing the real pertur-

bations. Our method also outperforms [9] and [35] with an improvement of approximately 17% and 0.6% respectively.

5. Ablation Study

In this section, we analyze the effect of the two major components of the PAT - the transition frames and the pseudo perturbations and present the run-time analysis of PAT on the UCF-101 dataset.

5.1. Input Frames

(a) (b)

Figure 5. (a) An adversarial frame from a UCF-101 dataset video (b) its corresponding transition frame.

Factors like color, texture and background add complexity to a frame. The attackers take advantage of such components to blend in the added perturbations. This is where the transition frames play a major role in bringing the perturbations into the light. The transition frames, calculated using Eq. 2, eliminate the passive components in a frame and focus on the objects and their motion only. As a result, the perturbations have no way to blend in and therefore are clearly visible.

In Fig. 5 (a), the perturbations do not stand out while being prominently visible in the transition frame (b). As insignificant components (for our detection task) are removed in transition frames, the perturbations are easily visible and processed by the network. This also helps in keeping the detection network architecture small, reducing the training & inference time.

The 1^{st} row of Table 3 shows the detection rate when original frames are used to train the detection network instead of the transition frames. The detection rate is almost equal to a random guess for both attacks. This is because the original frames have a lot of information irrelevant for the detection task which acts as a perfect disguise for the perturbations. On the contrary, the transition frame keeps only relevant elements and has a higher adversarial frame detection rate.

5.2. Pseudo Perturbations

In PAT, the pseudo adversarial transition frames are generated using the Eq. 3. The standard deviation σ is drawn from a uniform distribution and is different for every frame for each training epoch. This is crucial so that PAT can learn to handle adversarial input with different perturbations.

Table 3. Adversarial FDR showing the importance of key components of PAT using UCF-101. Sparse and Dense are the two attack baselines. *σ varies between 0.0001-0.05.

Frame Type	σ	**Sparse**	**Dense**
Original	Varying*	52.31%	51.12%
Transition	0.0001	63.98%	49.6%
Transition	0.01	80.12%	64.65%
Transition	Varying*	**83.62%**	**83.46%**

With varying standard deviation, the network learns from a different version of the same transition frame during every epoch. We observed in some cases of fixed σ, the network overfits at some point of time. For example, for $\sigma = 0.01$, the performance on the sparse attack is close to the best case FDR but the performance on the dense attack is poor. Thus, to achieve good performance on both the attacks, the transition frames and the varying standard deviation of Gaussian noise are both essential.

5.3. Run-time Analysis

To showcase that PAT is computationally inexpensive, we empirically measure the running time for our detection process using Nvidia Titan XP GPU. We use a mix of both clean and adversarial (sparse and dense attacks) videos to determine the average detection time for PAT. Our approach takes 0.01 seconds on average to determine whether a video is adversarial, which is extremely low overhead to the existing action recognition systems.

5.4. Conclusion

We proposed a novel approach, PAT to detect the adversarial frames in a video efficiently and keep the video-based networks secure from different types of attacks. We achieve good detection rate without having any access to the attack or the perturbations, which is generally the case in real-world applications. Our experiments on UCF-101 and Jester datasets demonstrate that the approach is highly accurate in detecting the adversarial input produced by different attacks. We also show the detection of adversarial video based on the PAT detected frames. Furthermore, we demonstrated the importance of transition frames and the varying Gaussian noise to generate pseudo perturbations in achieving a good frame detection rate.

Acknowledgement: The work was supported in part by a grant from ONR. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of ONR.

References

[1] Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017. 1

[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 1

[3] Yuzhen Ding, Nupur Thakur, and Baoxin Li. Advfoolgen: Creating persistent troubles for deep classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 142–151, 2021. 1

[4] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2625–2634, 2015. 1, 2

[5] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 1

[6] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016. 1

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1

[8] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. 4

[9] Xiaojun Jia, Xingxing Wei, and Xiaochun Cao. Identifying and resisting adversarial videos using temporal consistency. *arXiv preprint arXiv:1909.04837*, 2019. 2, 5, 6

[10] Linxi Jiang, Xingjun Ma, Shaoxiang Chen, James Bailey, and Yu-Gang Jiang. Black-box adversarial attacks on video recognition models. In *Proc. ACM Intl. Conf. on Multimedia*, pages 864–872, 2019. 2

[11] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1

[12] Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V Krishnamurthy, Amit K Roy Chowdhury, and Ananthram Swami. Adversarial perturbations against real-time video classification systems. *arXiv preprint arXiv:1807.00458*, 2018. 1, 2, 4

[13] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 1

[14] Shao-Yuan Lo and Vishal M Patel. Defending against multiple and unforeseen adversarial videos. *arXiv preprint arXiv:2009.05244*, 2020. 1, 2

[15] Jiajun Lu, Theerasit Issaranon, and David Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 446–454, 2017. 1

[16] Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The jester dataset: A large-scale video dataset of human gestures. In *Proc. IEEE Intl. Conf. on Computer Vision Workshops*, pages 0–0, 2019. 4

[17] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147, 2017. 1

[18] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 1

[19] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 1

[20] Peter Ondruska and Ingmar Posner. Deep tracking: Seeing beyond seeing using recurrent neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 1

[21] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. IEEE, 2016. 1

[22] Roberto Rey-de Castro and Herschel Rabitz. Targeted nonlinear adversarial perturbations in images and videos. *arXiv preprint arXiv:1809.00958*, 2018. 2

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 815–823, 2015. 1

[24] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 1

[25] Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017. 1

[26] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 4

[27] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019. 1

[28] Swathikiran Sudhakaran and Oswald Lanz. Learning to detect violent videos using convolutional long short-term memory. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017. 1

[29] Nupur Thakur, Yuzhen Ding, and Baoxin Li. Evaluating a simple retraining strategy as a defense against adversarial attacks. *arXiv preprint arXiv:2007.09916*, 2020. 1

[30] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 1

[31] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019. 1

[32] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 3978–3987, 2019. 1

[33] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. Sparse adversarial perturbations for videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8973–8980, 2019. 1, 2, 4

[34] Zhipeng Wei, Jingjing Chen, Xingxing Wei, Linxi Jiang, Tat-Seng Chua, Fengfeng Zhou, and Yu-Gang Jiang. Heuristic black-box adversarial attacks on video recognition models. In *AAAI*, pages 12338–12345, 2020. 2

[35] Chaowei Xiao, Ruizhi Deng, Bo Li, Taesung Lee, Benjamin Edwards, Jinfeng Yi, Dawn Song, Mingyan Liu, and Ian Molloy. Advit: Adversarial frames identifier based on temporal consistency in videos. In *Proc. IEEE Intl. Conf. on Computer Vision*, pages 3968–3977, 2019. 2, 5, 6

[36] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. 1

[37] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 669–677, 2016. 1