Enhancing Robustness in Video Recognition Models: Sparse Adversarial Attacks and Beyond

Ronghui Mu[b], Leandro Marcolino[a], Qiang Ni[a], Wenjie Ruan[b,*]

[a]*School of Computing and Communications, Lancaster University, Lancaster, UK*
[b]*Department of Computer Science, University of Liverpool, Liverpool, UK*

Abstract

Recent years have witnessed increasing interest in adversarial attacks on images, while adversarial video attacks have seldom been explored. In this paper, we propose a sparse adversarial attack strategy on videos (DeepSAVA). Our model aims to add a small human-imperceptible perturbation to the key frame of the input video to fool the classifiers. To carry out an effective attack that mirrors real-world scenarios, our algorithm integrates spatial transformation perturbations into the frame. Instead of using the l_p norm to gauge the disparity between the perturbed frame and the original frame, we employ the structural similarity index (SSIM), which has been established as a more suitable metric for quantifying image alterations resulting from spatial perturbations. We employ a unified optimisation framework to combine spatial transformation with additive perturbation, thereby attaining a more potent attack. We design an effective and novel optimisation scheme that alternatively utilises Bayesian Optimisation (BO) to identify the most critical frame in a video and stochastic gradient descent (SGD) based optimisation to produce both additive and spatial-transformed perturbations. Doing so enables DeepSAVA to perform a very sparse attack on videos for maintaining human imperceptibility while still achieving state-of-the-art performance in terms of both attack success rate and adversarial transferability. Furthermore, built upon the strong perturbations produced by DeepSAVA, we design a novel adversarial training framework to improve the robustness of video classification models. Our intensive experiments on various types of deep neural networks and video datasets confirm the superiority of DeepSAVA in terms of attacking performance and efficiency. When compared to the baseline techniques, DeepSAVA exhibits the highest level of performance in generating adversarial videos for three distinct video classifiers. Remarkably, it achieves an impressive fooling rate ranging from 99.5% to 100% for the I3D model, with the perturbation of just a single frame. Additionally, DeepSAVA demonstrates favorable transferability across various time series models. The proposed adversarial training strategy is also empirically demonstrated with better performance on training robust video classifiers compared with the state-of-the-art adversarial training with projected gradient descent (PGD) adversary.

Keywords: Deep Learning, Adversarial Robustness, Action Recognition, Adversarial Training, Video Classification.

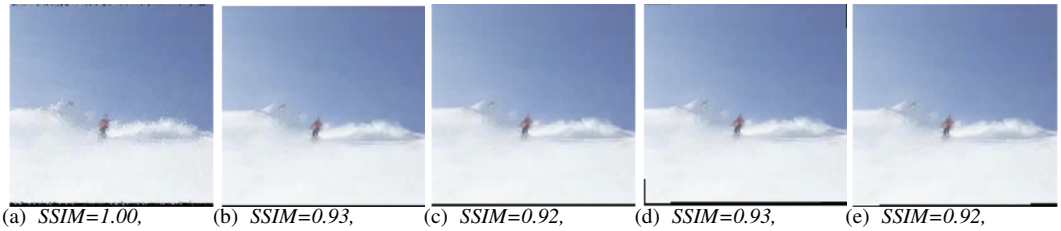
Highlights

- Perform sparse adversarial attacks on video models, aiming to perturb only a small number of frames while achieving a high attack success rate.
- Capture a wide range of adversarial examples by combining additive and spatial transformation perturbations.
- Use Structural Similarity Index (SSIM) instead of l_p-norm to maintain human perception during the attack process.
- Apply Bayesian Optimisation to nominate the most critical frame to perturb.
- Propose a new adversarial training method based on a combination perturbation generator.

1. Introduction

Deep Neural Networks (DNNs) have shown impressive performance in a variety of fields in recent years,

*Corresponding author
Email addresses: ronghui.mu@liverpool.ac.uk (Ronghui Mu), l.marcolino@lancaster.ac.uk (Leandro Marcolino), q.ni@lancaster.ac.uk (Qiang Ni), w.ruan@trustai.uk (Wenjie Ruan)



(a) *SSIM=1.00,* $l_{1,2}$*=0.00* (b) *SSIM=0.93,* $l_{1,2}$*=3.23* (c) *SSIM=0.92,* $l_{1,2}$*=3.39* (d) *SSIM=0.93,* $l_{1,2}$*=3.29* (e) *SSIM=0.92,* $l_{1,2}$*=3.40*

Figure 1: Comparison of SSIM and $l_{1,2}$ norm distance for: (a) Original image. [(b)-(e)] Perturbed images: (b) Noise. (c) Zooming out spatial scaling + noise. (d) Counterclockwise rotation 5 ° + noise. (e) Counter-clockwise rotation 5 ° + spatial scaling with zooming + noise. SSIM values for (b) and (d) are identical, whereas $l_{1,2}$ increases when an imperceptible rotation is introduced. This indicates that SSIM is less sensitive to rotation and can potentially result in a stronger adversarial attack with spatial transformation perturbation.

including image classification (Shen et al., 2017), text analysis (Mittal et al., 2020), speech recognition (Fohr et al., 2017), and object detection (Fohr et al., 2017). Despite their enormous success, extensive research has shown that DNNs are vulnerable to adversarial attacks (Szegedy et al., 2014; Carlini and Wagner, 2017b; Wang et al., 2022), appearing as adding nonrandom small and imperceptible perturbations on inputs that cause DNNs to give incorrect predictions. The increasing adoption of DNN models in various domains with security demands, such as facial authentication (Mohammadi et al., 2017), autonomous driving (Lu et al., 2017; Wu et al., 2023), and robotics (Yim et al., 2007), has sparked a heightened interest in the investigation of adversarial examples. Investigating adversarial examples can benefit the community in increasing awareness of safety risks in the system and also providing support for the construction of more robust DNNs (Xie et al., 2017; Huang et al., 2012).

In real-world scenarios, video often serves as an intrinsic input modality for the vision systems operating, such as those used in autonomous driving (Liao et al., 2020), medical care (Romeo et al., 2020) and traffic monitoring (Bas et al., 2007). Since video classifiers are built upon DNNs, adversarial examples can have detrimental effects on the model's performance. To this end, investigating adversarial samples on videos is urgently needed. By now, the adversarial attack and defence strategies primarily concentrate on image-related tasks, and the adversarial robustness of deep learning models on videos has not yet been comprehensively explored[1].

Nevertheless, videos are different from static images in that they contain a sequential data structure that changes dynamically over time. As a result, attack strategies designed for images cannot be applied directly to videos. Existing work on attacking video models can be divided into two types: dense attack and sparse attack. The dense attack is to perturb all frames of a video (Pony et al., 2021) which may result in a high fooling rate, but it is time-consuming and may also compromise human imperceptibility. Sparse attacks, on the other hand, were proposed by Wei et al. (2019). They demonstrated that the inherent characteristics of the video classification model facilitate the propagation of adversarial perturbations across different frames. Consequently, they leverage the temporal structure of videos to select only a subset of frames for the attack, which is a more efficient and reasonable approach, reducing the overall perturbation required.

To achieve such *sparse* adversarial attack, the perturbation should be powerful and the perturbed example should resemble a real-world instance as close as possible. According to the image attacks with spatial transformation perturbation (Xiao et al., 2018b; Wang et al., 2023b; Zhang et al., 2023), perturbing the positions of pixels can improve perceptual realism and make it locally smooth. Hence, we introduce a new term in the loss function for optimising both additive and spatial transformation perturbation to generate adversarial examples effectively. However, current video attack strategies all adopt the l_p-norm metric to measure the fidelity of the perturbed examples. Although the l_p norm is effective in capturing noise contamination, it is sensitive to natural-occurring transformations such as rotation, spatial shift, and scaling (Zhou Wang and Bovik, 2009). Taking Figure 1 as an example, where the original video frame is modified by different types of perturbation: additive Gaussian noise, spatial scaling, and slight rotation, and both $l_{1,2}$ and structural similarity (SSIM) are given. In Figure 1(b) and (d), one frame has only noise added, while the other has a small rotation and noise added. The results demonstrate that the SSIM values of the two

[1]In Table 1, we list all current adversarial video attacks as exhaustively as possible.

frames are identical, whereas the $l_{1,2}$ norm varies. This indicates that when using $l_{1,2}$-norm even a negligible spatial transformation perturbation can significantly amplify the adversarial distance, which constrain it to capture certain spatial transformations that naturally occur in real-world scenarios, such as camera shaking, vibration, or rotation, thereby limiting the effectiveness of the attack. On the other hand, *SSIM exhibits lower sensitivity towards these small modifications*, thus better aligning with human perception to constrain the amount of spatial transformation. Furthermore, the Image Quality Assessment community has demonstrated that SSIM is a superior alternative signal fidelity measure compared to the l_p-norm in applications where human perceptual criteria matter (Zhou Wang and Bovik, 2009). Consequently, in this paper, utilising an SSIM-based loss function is more suitable for constraining the distance between adversarial and clean videos, thereby improving the efficiency and efficacy of spatial-transformed perturbation.

Additionally, addressing the challenge of choosing the subset of frames for an extremely sparse attack is a complex task. In previous work, Wei et al. (2020) initially introduced a heuristic method to rank the importance of video frames, identifying the keyframes for potential attacks. Subsequently, Yan et al. (2022) utilised reinforcement learning techniques to learn the impact of various frame selection strategies and, in turn, determine the crucial frames to target. However, both these works are under the black-box setting, which needs many queries to access the model output and make decisions. As for under white box setting, we propose to implement the Bayesian Optimisation mechanism to select the most critical frames under the combined perturbation. We design an alternating optimisation strategy that can effectively identify the key frames via BO and then initiate additive and spatial-transformed perturbations on the selected key frames by stochastic gradient descent (SGD) based optimiser. Such an alternating process happens in each iteration of the optimisation until key frames are found.

To improve the robustness of the model in the adversary environment, various adversarial defensive methods have been proposed recently. The adversarial defence primarily aims to improve the neural network's accuracy for data that is perturbed by adversarial attacks. To mitigate this problem, existing approaches are mainly focused on adversarial training and certified defences. The adversarial training attempts to increase the robustness of the model by incorporating adversarial perturbation during the training process. As far as we know, from the empirical result, the most effective adversarial training method is the adversarial training with projected gradient descent (PGD) adversary (Wong et al., 2020). As for the certified defences, it is supposed to give a certified bound for the lowest accuracy under specific adversarial attacks. However, compared with adversarial training, there are still gaps in their performance (Ren et al., 2020). As a result, studying adversarial defences can help us to defend better against different adversarial threats (He et al., 2017). In security-critical systems, such as autonomous driving (Deng et al., 2020) and object detection (Zhang and Wang, 2019), adversarial defences serve as the essential blocks to ensure the trained models are reliable enough.

Overall, this paper introduces DeepSAVA, a Sparse Adversarial Video Attack for Deep neural networks, and proposes a novel adversarial training method to enhance the robustness of video models against strong attacks. DeepSAVA can capture a wide range of adversarial instances, encompassing noise contamination and various spatial transformations. It can achieve a sparse attack by perturbing only a few frames of a video, while still achieving a state-of-the-art attack success rate. Additionally, it exhibits strong adversarial transferability across various recurrent models compared to baseline methods. In summary, the contributions of this paper can be summarised as follows:

- DeepSAVA is the first work to combine additive and spatial-transformed perturbation for video attacks. With a proper SSIM-based constraint, we can produce strong perturbations combined with additive and spatial transformation. Such combined perturbation enables DeepSAVA to achieve successful attacks by just perturbing one frame and be effective across diverse types of DNNs.
- This paper is also the first work that uses Bayesian optimisation (BO) to identify the most critical frames of the video in attacks. We introduce an innovative alternating optimisation strategy for pinpointing the crucial frame under the combined perturbation, resulting in an improved following rate compared to the baselines.
- Based upon our novel perturbation generator, we propose a new adversarial training method to improve the robustness of the video classification models. We perform extensive experiments on different models to evaluate the effectiveness of our algorithm. The results confirm that the new design adversarial training could improve the robustness against various attacks.

The flow chart of our method is illustrated in Figure 2. We release the DeepSAVA code and adversarial train-

Similarity metric	Flickering l_p	RL l_p	Heuristic l_1	Append l_∞	BlackBox l_∞	GAN-based l_p	Sparse $l_{2,1}$	Gradient $l_{2,1}$	Ours SSIM
Spatial-transformed perturbation	✗	✗	✗	✗	✗	✗	✗	✗	✓
Additive Perturbation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Identify Key Frames	✗	✓	✓	✗	✗	✗	✗	✓	✓
Transferability Study	✗	✗	✗	✓	✓	✗	✓	✗	✓
Sparse Attack	✗	✓	✓	✗	✗	✗	✓	✗	✓
Adversarial Training	✗	✗	✗	✗	✗	✗	✗	✗	✓

Table 1: Comparison with related works (Flickering (Pony et al., 2021), RL (Yan et al., 2022), Heuristic (Wei et al., 2020), Append (Chen et al., 2021), BlackBox (Jiang et al., 2019), GAN-based (Li et al., 2019), and Sparse Attack (Wei et al., 2019)) and Gradient-based (Xu et al., 2022) in different aspects.

ing[2][3] and generated adversarial videos across multiple models[4].

The paper's structure can be summarised as follows: In Section 2, we provide an overview of previous research on video action recognition models, adversarial attacks in images and videos, and adversarial defense. Section 3 is dedicated to presenting our framework and methodology. Initially, we address the definition of the attack problem and then describe the implementation of the sparse adversarial attack using an alternating optimisation strategy. We also detail the adversarial training algorithm aimed at enhancing the robustness of our video recognition model. In Section 4, we present the results of our experiments, which include comparisons with a baseline, an ablation study, transferability assessments, and the results of adversarial training. Finally, in Section 5, we discuss the limitations of our proposed method and offer suggestions for future work within the community.

2. Related Work

Video Action Recognition Models: The video classification task focusses mainly on action recognition (Kong and Fu, 2018). Previous studies on video classification using DNNs are developed in two ways: using 2D or 3D-based convolution neural networks (CNN). Since CNNs have obtained state-of-the-art performance in image classification, Karpathy et al. (2014) first proposed to use 2D CNN to classify each frame of the video. Szegedy et al. (2015) then developed the Inception-v3, which is commonly used as a baseline classification model. As 2D-CNNs use incomplete video information, some work added layers containing temporal information, such as LSTM, to integer CNN features extracted over time, which is referred to as CNN+LSTM model (Nguyen et al., 2015; Donahue et al., 2017). As for 3D CNNs (Tran et al., 2015), it can learn temporal features from videos by entering all frames in three dimensions directly. Carreira and Zisserman (2017) proposed a two-stream inflated 3D CNN (I3D) to build the 2D kernel first and then merge the pooling layer and kernel into a 3D network. By pre-training the I3D on Kinetics Dataset, it could reach state-of-the-art performance on recognising UCF101 and HMDB51 action video datasets.

Adversarial attack on images: The adversarial attack on images has been explored extensively recently. Szegedy et al. (2014) first proposed adding visually imperceptible noise on the images to mislead pre-trained CNNs to give the wrong prediction label. Goodfellow et al. (2015) proposed to use of a gradient-based approach, the fast gradient sign method (FGSM), to generate adversarial examples. DeepFool (Moosavi-Dezfooli et al., 2016) is then proposed to find the minimal perturbation by iteratively linearising the loss function. Other gradient-based optimisation algorithms to generate perturbation were also proposed (Carlini and Wagner, 2017b; Liu et al., 2017; Tanay and Griffin, 2016; Xiao et al., 2018a; Yin et al., 2022). These works mentioned above only apply additive perturbation to pixels. Some works (Xiao et al., 2018b; Wong et al., 2019; Laidlaw and Feizi, 2019; Laidlaw et al., 2020; Jordan et al., 2019) use a functional perturbation which is a non-additive-only perturbation such as spatial transformation. These perturbations slightly modify the location of pixels. Some work such as (Jordan et al., 2019; Zhao et al., 2020; Gragnaniello et al., 2021) also uses other types of metrics such as SSIM to quantify human perception, but none of them explored SSIM-guided spatial transformation. For more details on adversarial attacks, please refer to our recent survey (Huang et al., 2020) and

[2] https://github.com/TrustAI/DeepSAVA

[3] A preliminary version of this paper has been published at the 32nd British Machine Vision Conference 2021 (Mu et al., 2021).

[4] https://www.youtube.com/channel/UCBDswZC2QhBhTOMUFNLchCg

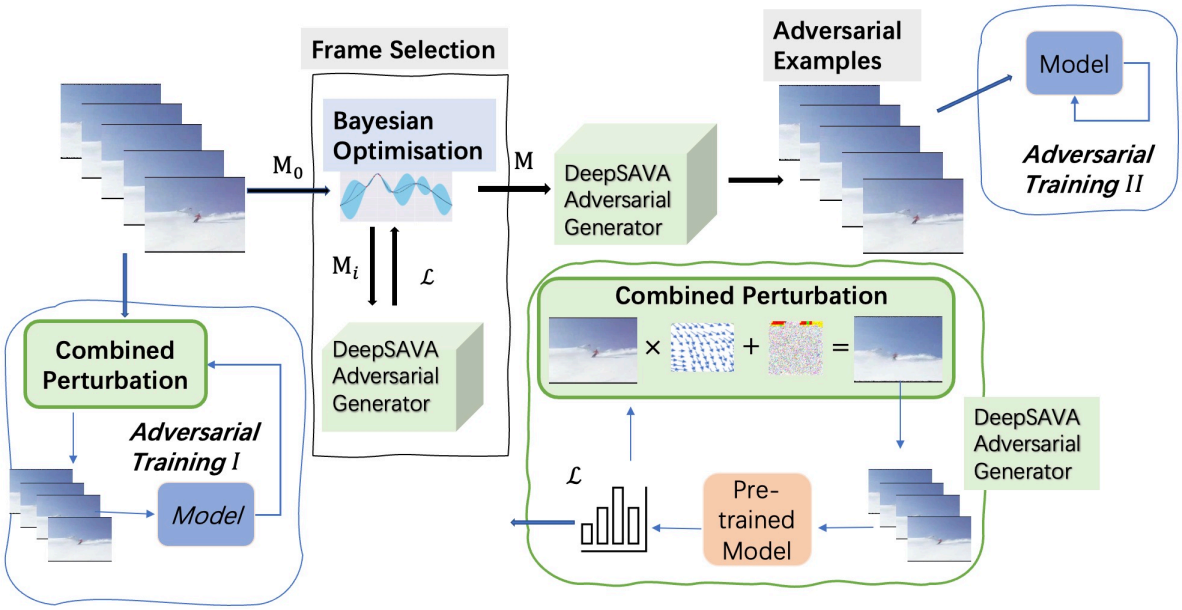


Figure 2: Overview of DeepSAVA: The key frame mask indicator M is alternately identified using Bayesian optimisation (BO), which takes M_0 as input and iteratively updates M_i by interacting with the adversarial generator to obtain the prediction loss ($\mathcal{L}$). The adversarial generator incorporates additive and spatial transformation perturbations to generate adversarial examples for the selected frame. Two adversarial training approaches are demonstrated. Adversarial Training I utilises original videos as input and embeds combined perturbations during the training process, while Adversarial Training II directly uses the generated adversarial examples as input to train the model.

tutorial (Ruan et al., 2021; Berthier et al., 2021).

Adversarial attack on videos: Wei et al. (2019) claimed that they are the first to attack videos. Instead of attacking each frame of a video, they apply additive perturbations on randomly selected frames and use $l_{2,1}$ norm to guide the gradient-based optimisation and evaluated the performance on the CNN+LSTM model. Li et al. (2019) used a GAN network to generate offline universal perturbations for each frame. Chen et al. (2021) proposed to append a noise frame to the end of videos, which is obtained based on all videos. However, adding an additional frame is not imperceptible to humans, and in some scenarios, it may appear abnormal to observe an extra noisy frame. Pony et al. (2021) applied flickering temporal perturbations on each frame to generate universal perturbations for the I3D model, but they did not consider to apply sparse attack on videos. Jiang et al. (2019) was the first to propose a black-box approach to attack videos, which also performs attack on each frame. In the field of black-box sparse attacks, several studies have introduced diverse algorithms aimed at determining the appropriate frame for executing an attack. Wei et al. (2020) proposed to use a heuristic method, which measures the model's outputs in the event that a certain frame is eliminated in order determine its importance. Therefore, to select the key frame, the connection between the input sample's features and the model outputs is taken into account. However, if the frame is eliminated directly from the input video, the output may be significantly altered, leading to low accuracy and time consuming.

Then, Yan et al. (2022) used a reinforcement learning algorithm to select the key frames to perform a black-box attack. However it requires more queries to train the agent to select the key-frame, which is not suitable for the white-box attack. Therefore, we proposed a novel white-box sparse attack framework, DeepSAVA Mu et al. (2021), which is the first work to perform white-box sparse attack on videos with keyframe selection technology. We employ an alternating optimisation strategy to combine the Bayesian optimisation and Adam optimiser to select the keyframes. Xu et al. (2022) proposed to use a gradient-based map to identify crucial features in all frames, selecting frames containing these features as keyframes. In contrast, our approach employing Bayesian optimisation is more general and can be applied to any models. Existing methods exclusively use additive perturbation based on l_p-norm distance, whereas we employ spatial transformation based on the SSIM metric for our approach, leading to a more powerful at-

tack. Table 1 presents a comparison of our method with existing works on video attacks across six aspects.

Adversarial defences: The adversarial defence is proposed to improve the robustness of the model, which aims to reduce the power of adversarial examples. Goodfellow et al. (2015) first proposed to inject adversarial examples into the training dataset to retrain the model. However, this method is time-consuming, and the achieved robustness of the retrained model relies on the power of the injected adversarial examples. Then, Madry et al. (2018) first build the structure of adversarial training by adding multi-step projected gradient descent (PGD) in the training stage, which is considered the most effective way for adversarial defences (Athalye et al., 2018). Zhang et al. (2019) then proposed the method to establish a trade-off between model accuracy and robustness by adding an additional loss term of adversarial examples in the training stage.

There is a rising number of works proposed to achieve adversarial defences, in addition to adversarial training, some works attempt to adopt other methods, such as detection techniques (Metzen et al., 2017; Feinman et al., 2017; Carlini and Wagner, 2017b,a), provable defences (Katz et al., 2017; Sinha et al., 2018; Wong and Kolter, 2018; Raghunathan et al., 2018; Jin et al., 2022), and preprocessing algorithms (Guo et al., 2018; Buckman et al., 2018; Song et al., 2018). From the empirical perspective, adversarial training with PGD (Madry et al., 2018) adversary still appears to engage the most robust performance against a wide range of adversarial attacks (Madry et al., 2018; Li et al., 2020; Wang et al., 2023a). As existing adversarial defences mainly focused on images, and there is no work to improve the robustness of the video classification model against spatial-transformed perturbations, thus, in this paper, we adapt the existing adversarial training methods on images to videos to improve its robustness on both spatial transformation and additive perturbation.

3. Methodology

3.1. Attack Problem Definition

The video classifier is defined as $J(\cdot;\boldsymbol{\theta})$, with a set of pretrained weights $\boldsymbol{\theta}$. We define the input to this classifier as a clean video, represented by $\mathbf{X} = (x_1, x_2, ..., x_T) \in \mathbb{R}^{T\times W\times H\times C}$, where each dimension represents a different aspect of the video frames: T is the total number of frames (the video's length), while W, H, and C denote the width, height, and the number of channels (such as color channels) in each frame, respectively.

The aim of our study is to create an 'adversarial video', denoted as $\hat{\mathbf{X}}$, from the original video $\mathbf{X}$. This transformation involves two main steps: first, altering the spatial properties of the video with a transformation function $\mathcal{S}$ (called a spatial transformer), and second, introducing a certain amount of noise (or disturbance) in the frame.

The objective function, which guides the creation of this adversarial video, is formulated as follows:

$$\arg\min \lambda \ell_{similar}(\hat{\mathbf{X}}, \mathbf{X}) - \ell_{adv}(\mathbf{1}_y, J(\hat{\mathbf{X}}; \boldsymbol{\theta})),$$

In this equation:

- $\mathbf{1}_y$ represents the one-hot encoding of the true label y of the input video $\mathbf{X}$.
- $\ell_{similar}$ is a similarity loss function, measuring how close the adversarial video $\hat{\mathbf{X}}$ is to the original $\mathbf{X}$.
- ℓ_{adv} is a loss function evaluating the discrepancy between the predicted and true labels.
- λ is a balancing parameter, controlling the trade-off between $\ell_{similar}$ and ℓ_{adv}.

For calculating ℓ_{adv}, we use the cross-entropy method, which has been demonstrated to be effective in adversarial video generation, as shown in Wei al. (2019).

3.2. Structural Similarity Index Measure (SSIM)

The SSIM was first proposed in Wang *et al.* (2002) (Wang and Bovik, 2002), and is detailed in Wang *et al.* (2004) (Wang et al., 2004). Given x and $\hat{x}$ as the local pixels taken from the same location of the same frame in the clean video and adversarial video, respectively, the local similarity between them can be computed on three aspects: structures ($s(x, \hat{x})$), contrasts ($c(x, \hat{x})$), and brightness values ($b(x, \hat{x})$). The local SSIM is formed by these terms (Wang et al., 2004):

$$S(x, \hat{x}) = s(x, \hat{x}) \cdot c(x, \hat{x}) \cdot b(x, \hat{x}) = \left(\frac{\sigma_{x\hat{x}} + D_1}{\sigma_x \sigma_{\hat{x}} + D_1}\right) \cdot \left(\frac{2\sigma_x \sigma_{\hat{x}} + D_2}{\sigma_x^2 + \sigma_{\hat{x}}^2 + D_2}\right) \cdot \left(\frac{2\mu_x \mu_{\hat{x}} + D_3}{\mu_x^2 + \mu_{\hat{x}}^2 + D_3}\right), \quad (1)$$

where μ_x and $\mu_{\hat{x}}$ denote means, σ_x and $\sigma_{\hat{x}}$ are standard deviations of x and $\hat{x}$, respectively; $\sigma_{x\hat{x}}$ represents the cross-correlation of x and $\hat{x}$ after deleting means; D1, D2, and D3 are weight parameters. For the SSIM metric, a value of 1 means that the two images compared are the same.

As we mentioned before, the SSIM is less sensitive to the combination of additive and spatial perturbations

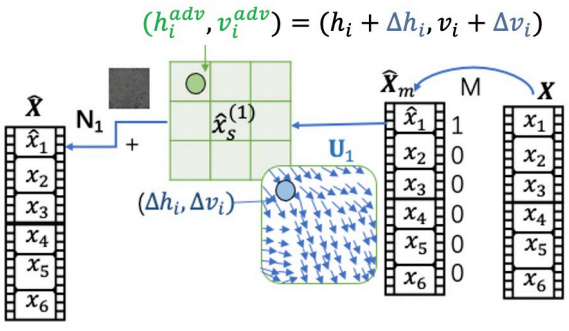


Figure 3: The process to perturb one frame of a video.

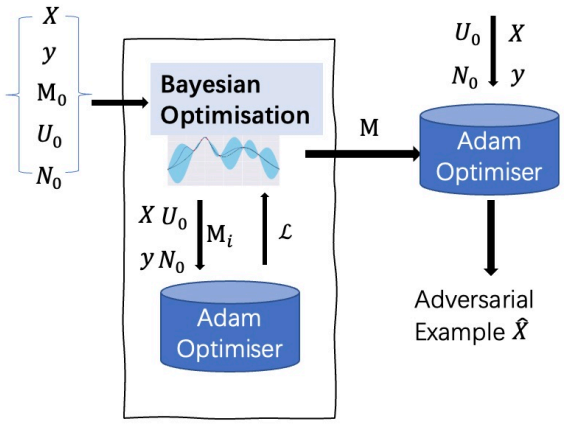


Figure 4: The systematic optimisation process by using Bayesian optimisation and Adam Optimiser. The mask indication is a binary vector to determine which frame should be perturbed.

[5] and more similar to human perception than l_p-norms (Zhou Wang and Bovik, 2009). Because the SSIM is differentiable with respect to the input variable (the derivation process of SSIM is shown in Appendix A), in this paper, we apply SSIM to calculate the similarity loss to constrain the perturbation during the optimisation process. The overall score of SSIM of the video is calculated by summing up SSIM loss over all frames of the video.

3.3. Sparse Spatial Transform Adversarial Attack

We will now introduce our algorithm for performing attacks with a combined perturbation. The main idea of the algorithm is to apply the combined perturbation on selected frames to achieve a high fooling rate.

Sparse Attack: Formally, the mask indicator $M = (m_1, m_2, .., m_T) \in \mathbb{R}^T$ is used to choose the key frames in the video, where $m_t \in \{\mathbf{0}, \mathbf{1}\}$ indicates whether the t-th frame is masked to be perturbed. The masked video $\mathbf{X}_m$ is formed through the map function $\mathcal{M}(M, \mathbf{X})$, and then fed into the spatial transformer $\mathcal{S}$.

Spatial Transformed Perturbation: Given the t-th frame $x^t \in \mathbb{R}^{W\times H\times C}$ of the input video $\mathbf{X}$, the n-th pixel in this frame is denoted as x_n^t. The location of this pixel within the frame is represented by a two-dimensional coordinate (h_n^t, v_n^t). The spatial transformer (Jaderberg et al., 2015), denoted as $\mathcal{S}$, is a differentiable model that functions using flow displacement vectors $\mathbf{U} = ((\Delta\mathbf{H}^1, \Delta\mathbf{V}^1), (\Delta\mathbf{H}^2, \Delta\mathbf{V}^2), ..., (\Delta\mathbf{H}^T, \Delta\mathbf{V}^T)) \in \mathbb{R}^{T\times 2\times H\times W}$. Here, $\mathbf{H}^t = (h_0^t, h_1^t, ..., h_n^t)$ and $\mathbf{V}^t = (v_0^t, v_1^t, ..., v_n^t) \in \mathbb{R}^{H\times W}$ are used to synthesize the 2D coordinates of adversarial videos. Suppose $\hat{x}_n^t$ with location $(\hat{h}_n^t, \hat{v}_n^t)$ is the adversarial example transformed from x_n^t, determined by its corresponding spatial displacement flow vector $(\Delta h_n^t, \Delta v_n^t)$. The original location of the pixel x_n^t can be deduced from $\hat{x}_n^t$ by setting $(h_n^t, v_n^t) = (\hat{h}_n^t + \Delta h_n^t, \hat{v}_n^t + \Delta v_n^t)$.

Taking into account the sparse attack mask indicator M, the transformed adversarial video is represented as

$$\hat{\mathbf{X}}_S = \mathcal{S}(\mathbf{U}, \mathbf{X}, M).$$

Additive Perturbation: The additive perturbation is the most common way to generate adversarial examples (Carlini and Wagner, 2017b; Goodfellow et al., 2015). We define additive noise as $\mathbf{N} \in \mathbb{R}^{T\times W\times H\times C}$. We combine spatial transformation and additive perturbation to generate adversarial videos as (illustrated in Figure 3):

$$\hat{\mathbf{X}} = \mathbf{N} \cdot M + \hat{\mathbf{X}}_S \tag{2}$$

3.4. Novel Alternating Optimisation Strategy

In this paper, we utilise Bayesian Optimisation (BO) to select the most critical frames. As frame selection is a discrete variable optimisation problem, we also tried other discrete optimisation techniques such as simulated annealing (SA) (Fortran et al., 1992) and genetic algorithms (GA) (Whitley, 1994), but both spent about 200s to find the final result, which is much longer than about 16s taken by BO.

The generated adversarial video is formed as $\hat{\mathbf{X}} = \mathbf{N} \cdot M + \mathcal{S}(\mathbf{U}, \mathbf{X}, M)$. In this paper, the similarity loss $\ell_{similar}$ and the adversarial loss ℓ_{adv} in problem (1) can be expressed as $\ell_{similar}(\hat{\mathbf{X}}, \mathbf{X}) = 1 - SSIM(\hat{\mathbf{X}}, \mathbf{X}) = \mathcal{L}_s(\mathbf{N}, \mathbf{U}, \mathbf{X}, M)$ and $\ell_{adv}\left(\mathbf{1}_y, J(\hat{\mathbf{X}}; \boldsymbol{\theta})\right) = \mathcal{L}_a(\mathbf{N}, \mathbf{U}, \mathbf{X}, M)$. Therefore, problem (1) can be simplified as:

$$\arg\min_{M,\mathbf{N},\mathbf{U}} \lambda\mathcal{L}_s(\mathbf{N}, \mathbf{U}, \mathbf{X}, M) - \mathcal{L}_a(\mathbf{N}, \mathbf{U}, \mathbf{X}, M) \tag{3}$$

[5] For convenience, we use *combined perturbation* for short in this paper.

As M is a discrete binary vector, which makes problem (4) non-differentiable, the Bayesian optimisation (BO) is then utilised to optimise the binary vector M by identifying the critical frame that should be perturbed. It can be solved systematically by a novel alternating optimisation strategy. We initially provide M as input to the Bayesian optimisation process. During the search process, BO generates different configurations of M to explore. Importantly, at each iteration, the BO finds a configuration of M and then queries the model output to obtain an evaluation score, which is used to guide the BO search for the next optimal M. Hence, when interacting with the model, the value of M remains fixed, transforming the problem into a differentiable one that can be solved using Stochastic Gradient Descent (SGD)-based optimisation. In this paper, we choose the Adam optimiser (Kingma and Ba, 2015) because of its robust and fast convergence performance. This process repeats for a fixed number of iterations, and the solution is continuously improved via both two techniques.

BO proposes sampling points from the search space through acquisition functions to obtain the reward of previous points. We apply expected improvement (EI) as our acquisition function F, which is a widely used function:

$$F = \mathrm{EI}(M) = \mathbb{E}[\max\left(\mathcal{L}(M) - \mathcal{L}\left(M^{+}\right), 0\right)], \quad (4)$$

where $\mathcal{L}(M)$ is the loss feedback from Adam optimiser by fixing M; $\mathcal{L}\left(M^{+}\right)$ is the best value obtained so far, and M^{+} is its location.

During the BO process, we will find the best mask indicator through several iterations. In the k-th iteration of BO, we will first sample a candidate M^{k} according to the acquisition function F. Then, the corresponding loss $\mathcal{L}_k$ will be computed by the Adam, which will then affect the next sampled point M^{k+1} for the next iteration. When the BO reaches the maximum exploration number, the best M with minimum loss will be fed into the Adam optimiser to generate the final adversarial video. The process is illustrated in Figure 4.

Algorithm 1 and Algorithm 2 detail the BO selection and adversarial video generation algorithms, respectively. In Algorithm 1, the next sampling point M is obtained by maximizing the acquisition function F based on the previous sampling data set D (Line 3). After the Adversarial Generator (G) is optimised, the loss $\mathcal{L}$ for M is calculated. Then the M with its corresponding $\mathcal{L}$ are appended to the sampling pool D to propose the next sampling point. In Algorithm 2, according to the optimised mask indicator M, the final flow vector $\mathbf{U}$ and additive noise $\mathbf{N}$ are optimised via Adam.

Algorithm 1 Bayesian Optimisation for video frame selection

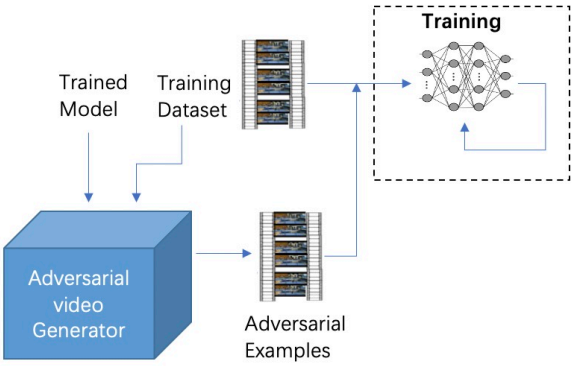
Input: input video $\mathbf{X}^{T\times W\times H\times C}$; label y; adversarial generator G; weight parameter λ; acquisition function F; Number of steps to explore K;
Output: Frame selection mask indicator M

1: Initialise flow network parameter $\mathbf{U}_0$ and additive noise $\mathbf{N}_0$;
2: Obtain initial sampling data set $D = (M_0, \mathcal{L}_0)$
3: **for** $k \leftarrow 1, K$ **do**
4: $\quad M_k \leftarrow \mathrm{argmax}_M\, F\,(M \mid D)$
5: $\quad \mathcal{L}_k \leftarrow G(\mathbf{X}, y, M_k, \lambda)$
6: $\quad D \leftarrow D \cup (M_k, \mathcal{L}_k)$
7: $M \leftarrow \mathrm{argmin}_{M\in D}\, \mathcal{L}$.
8: **return** M^{*}

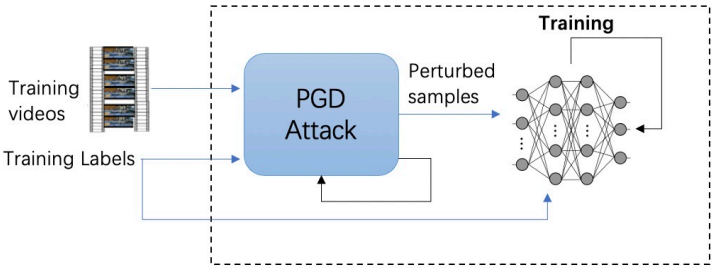
Algorithm 2 DeepSAVA adversarial generator (G)

Input: $\mathbf{X}^{T\times W\times H\times C}$; Mask indicator M; y; weight parameter λ

1: Initialise flow vector $\mathbf{U}_0$, and additive noise $\mathbf{N}_0$;
2: **for** $step \leftarrow 1, maxStep$ **do**
3: $\quad \hat{\mathbf{X}} = \mathbf{N}\cdot M + \mathcal{S}(\mathbf{U}, \mathbf{X}, M)$
4: $\quad \mathcal{L} = \lambda(1 - SSIM(\hat{\mathbf{X}}, \mathbf{X})) - \ell_{adv}\left(\mathbf{1}_y, J(\hat{\mathbf{X}}; \boldsymbol{\theta})\right)$
5: $\quad \mathbf{U}^{*}, \mathbf{N}^{*} \leftarrow \mathrm{argmax}_{\mathbf{U},\mathbf{N}}\, \mathcal{L}$
6: **return** $\mathbf{U}^{*}, \mathbf{N}^{*}$



(a) Adversarial training via injecting adversarial examples



(b) Adversarial training with PGD attack

Figure 5: Adversarial Training Overview.

3.5. Adversarial Training

Adversarial training is designed as the most intuitive defence mechanism against various adversarial attacks, which incorporates adversarial examples during the training process to improve the robustness of the model. Formally, this goal can be constructed as a min-max optimisation problem. Given the video classifier $J(\cdot;\boldsymbol{\theta})$ parameterised by $\boldsymbol{\theta}$ and the adversarial video input $(\hat{\mathbf{X}}_i, y_i)$ with ground truth label y_i, the objective function can be formulated as follows:

$$\min_{\boldsymbol{\theta}} \sum_{i} \max \ell_{adv}(J(\boldsymbol{\theta}, \hat{\mathbf{X}}_i), \mathbf{1}_{y_i}) \tag{5}$$

where ℓ_{adv} is the adversarial loss. The inner maximization term aims to find the optimal adversarial examples and it can be approximated by some well-developed adversarial attack algorithms such as PGD (Madry et al., 2018) and FGSM (Yuan et al., 2019). The following outer minimisation term represents the traditional training process which aims to minimise the training loss by applying a gradient descent algorithm to optimise the model parameters $\boldsymbol{\theta}$. The generated retrained model is expected to be more robust against the adversarial attack that is used in the training process to generate adversarial examples.

The most intuitive way is to inject adversarial examples generated by the adversarial attack to re-train the model, as represented in Figure 5 (a). However, this method is time-consuming, as we need to spend much more time obtaining thousands of adversarial examples from the training dataset. Goodfellow *et al.* (Goodfellow et al., 2015) first proposed to use of the Fast Gradient Sign Method (FGSM) to solve the inner maximisation problem. The objective function to approximate the inner maximisation for the FGSM adversarial training can be formed as follows:

$$\hat{\mathbf{X}}_i = \mathbf{X}_i + \alpha \cdot sign\left(\nabla_{\mathbf{X}_i} \ell_{adv}\left(\mathbf{1}_{y_i}, J(\mathbf{X}_i;\boldsymbol{\theta})\right)\right)$$

However, the model trained by the FGSM adversarial training algorithm is still vulnerable to stronger adversarial attacks, such as the PGD attack, which is based on iterative adversarial attacks. The adversarial training based on PGD attack is considered as one of the most effective approaches to improve the robustness of models (Madry et al., 2018; Huang et al., 2022; Wang et al., 2021). It is also the strongest first-order attack founded by the community (Wang et al., 2021), (Kolter and Madry, 2019) and is the state-of-the-art defence (Athalye et al., 2018). Therefore, there are many works that are based on PGD training to improve the robustness of DNN (Bai et al., 2021; Wong et al., 2020; Shafahi et al., 2019). Thereby, in this paper, we adapt the PGD optimisation method to solve the inner maximisation problem.

The overall adversarial training process with PGD attack is shown in Figure 5 (b), and its algorithm is sketched in Algorithm 3. As we can see from Algorithm 3, the key improvement of PGD adversarial training is to perform multiple small steps α to estimate the inner maximisation problem (lines 4-5).

As our DeepSAVA framework is a stronger and more effective adversarial attack method on videos, we design a novel adversarial training approach based on traditional PGD adversarial training to defend our adversarial attack with a combined perturbation, as shown in Algorithm 4.

Compared with the PGD adversarial training (Madry et al., 2018), we identify the main difference between the two methods is that our approach takes the spatial transformation perturbation combined with additive noise into account, which is achieved via the operation presented

Models	UCF101	HMDB51
CNN+LSTM	74%	43%
I3D	94.9%	80%
Inception-v3	71.2%	47%

Table 2: Training accuracy of the classifiers to be attacked.

Algorithm 3 PGD adversarial training (Madry et al., 2018)

Input: $\mathbf{X}^{T\times W\times H\times C}$; y; M; training epochs T; dataset batch size N; PGD steps K; step size α

1: **for** $t \leftarrow 1, T$ **do**
2: **for** $i \leftarrow 1, N$ **do**
3: $\sigma = 0$
4: **for** $k \leftarrow 1, K$ **do**
 //Perform PGD Adversarial Attack
5: $\sigma = \sigma + \alpha \cdot \left(\nabla_\sigma \ell_{adv}\left(\mathbf{1}_{y_i}, J(\mathbf{X}_i + \sigma; \boldsymbol{\theta})\right)\right)$
 //Update the model weights
6: $\boldsymbol{\theta} = \boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}} \ell\left(J(\boldsymbol{\theta}; \mathbf{X}_i + \sigma), \mathbf{1}_{y_i}\right)$

Algorithm 4 PGD adversarial training with a combined perturbation

Input: $\mathbf{X}^{T\times W\times H\times C}$; y; M; training epochs T; dataset batch size N; PGD steps K; step size α

1: Randomly initialize flow network parameter $\mathbf{U}$ and additive noise $\mathbf{N}$;
2: **for** $t \leftarrow 1, T$ **do**
3: **for** $i \leftarrow 1, N$ **do**
4: **for** $k \leftarrow 1, K$ **do**
 //Perform Adversarial Attack
5: $\hat{\mathbf{X}}_i = \mathbf{N} \cdot M + \mathcal{S}(\mathbf{U}, \hat{\mathbf{X}}_i, M)$
6: $\hat{\mathbf{X}}_i = \hat{\mathbf{X}}_i + \alpha \cdot sign\left(\nabla_{\hat{\mathbf{X}}_i}\left(\ell_{adv}\left(\mathbf{1}_{y_i}, J(\hat{\mathbf{X}}_i; \boldsymbol{\theta})\right)\right)\right)$
 //Update the model weights
7: $\boldsymbol{\theta} = \boldsymbol{\theta} - \nabla_{\boldsymbol{\theta}} \ell\left(J(\boldsymbol{\theta}; \hat{\mathbf{X}}), \mathbf{1}_{y_i}\right)$

in Lines 5-6 of Algorithm 4. Despite the adversarial training procedures of the two algorithms are similar, our defence method is more empirically robust against Sparse attack (Wei et al., 2019) and DeepSAVA attack. We conjecture that the advantage mainly comes from the added combined perturbation term that can perform a more effective attack than additive perturbation only.

4. Experiments

4.1. Experimental Setup

Dataset: As action recognition video datasets are widely used in adversarial video attack studies, we choose two popular benchmark action recognition datasets to evaluate the performance of our method: UCF101 (Soomro et al., 2012) and HMDB51 (Kuehne et al., 2011). Both datasets are realistic action recognition datasets. The UCF101 contains 13,320 videos with 101 categories such as playing instruments, body movements, and human-object interaction. Similarly, HMDB51 has around 7,000 videos within 51 categories related to body motion and facial actions.

Action Recognition Models: We evaluate DeepSAVA on three classifiers: *Inception-v3*, a 2D-CNN based model (Szegedy et al., 2016), which is widely used in the image recognition task with high accuracy; *I3D*, a 3D-CNN based model, pre-trained in Kinetics (Carreira and Zisserman, 2017); *CNN + LSTM*, which is pre-trained in ImageNet to extract features from videos and then input these features to train the LSTM network. The training accuracy of all classifiers is shown in Table 2. The training ratio is 70% while the testing ratio is 30%.

Baseline methods: Two baseline methods are used for comparison, the Sparse (Wei et al., 2019) and Sparse Flickering. For the works shown in Table 1, only (Wei et al., 2019) is the sparse white-box attack; (Wei et al., 2020)(Yan et al., 2022) are black-box sparse attack methods. As our work is a white-box sparse attack, we choose the most related one, Sparse (Wei et al., 2019), as the main baseline. We perform the perturbation directly on the frame, while (Chen et al., 2021) added an additional frame at the end of the video, which is more visible to humans. So we did not include it as a baseline due to its compromise on the similarity of human perception. In (Li et al., 2019), GANs are used to attack real-time video, which is not comparable to our method. We modified Flickering (Pony et al., 2021), which perturbs all frames, into a sparse one as the Sparse Flickering baseline, but we still show the performance of perturbing all frames.

Experiments Setting: The length of all input videos is crafted to be the same (40 frames). We randomly selected 200 videos from different categories in the test

Models	Attack Method	UCF101		HMDB51	
		FR	ANI	FR	ANI
CNN+LSTM	Sparse	52.77% ± 2.44%	16.45	95.2% ± 1.8%	16.4
	Sparse Flickering	48.48% ± 1.67%	23.55	91.94% ± 2.93%	8.4
	DeepSAVA(without BO)	56.22% ± 1.65%	8.32	99.27% ± 0.34%	8.42
	DeepSAVA(BO)	57.22% ± 1.36%	8.77	100%	6.6
I3D	Sparse	10.12% ± 1.19%	44	5.74% ± 1.25%	25.1
	Sparse Flickering	1.15% ± 0.68%	13	0%	-
	DeepSAVA(without BO)	47.57% ± 2.64%	12.15	46.39% ± 3.86%	12.2
	DeepSAVA(BO)	99.89% ± 0.11%	6.47	99.92% ± 0.08%	5.35
Inception-v3	Sparse	42.25% ± 4.30%	33.70	45.82% ± 1.56%	22.06
	Sparse Flickering	21.73% ± 1.39%	35.4	27.55% ± 0.98%	27.25
	DeepSAVA(without BO)	68.86% ± 1.83%	13.29	68.98% ± 3.19%	11.84
	DeepSAVA(BO)	70.39% ± 2.78%	10.52	74.74% ± 0.82%	9.07

Table 3: Comparison with baselines, DeepSAVA without BO and with BO on different models by only perturbing one frame. '-' means that there is no successful attack. Gray cell shows the best results.

dataset. For those experiments without saying the specific constraint, the maximum allowed search iteration (100 iterations) is applied; all experiments use Adam optimiser with a *0.01* learning rate. The parameter λ is set to 1.5 for the CNN+LSTM model, and 1.0 for the I3D and Inception-v3 models. For λ, values that can balance the fooling rate and the strength of the perturbation are used. As for the step size of the adversarial attack used in the adversarial training, we use the alpha as $\frac{1}{255}$, where 255 is the re-scale image size, which is a commonly used method to determine the step size. It is a heuristic and the optimal value that is a small fraction of the range of pixel values in the image (The performance of different step sizes is presented in Appendix B).)

Metrics: *Fooling Rate (FR)*: the percentage of generated adversarial videos that are misclassified successfully. *Average Number of Iterations (ANI)*: the average number of iterations taken to generate adversarial examples successfully based on the same original videos, which is used to measure the efficiency when we set a constraint on the maximum allowed iteration.

4.2. DeepSAVA adversarial attack

4.2.1. Comparison with baseline methods

In this section, we will show the comparison results between DeepSAVA and baselines. Since running BO will add extra time to choose the frame, to make the comparison more complete, we also take the DeepSAVA without BO selection into account.

Limited iterations: Since each method uses a different metric, in order to control the maximum allowed perturbation, we limit the number of search iterations for all methods. Each iteration only allows a small amount of perturbation (controlled by the learning rate of Adam optimiser), following the same setup used by the baselines. The results in Table 3 show that the ANIs are much below the maximum allowed iteration (100). In Table 4, we show the relationship between the iteration and the strength of perturbation. It can be seen that even when it reaches the maximum iteration, the lp-norm and $SSIM$ distances are still acceptable. Given that, setting a constraint on the maximum search number to 100 will not lead to large distortion.

We run the experiments 10 times and show the average results with a 99% confidence interval. For the methods without frame selection, the first frame is perturbed. As shown in Table 3, BO selection is more efficient than the one without BO. This happens because it is able to select the most critical frame, which can improve efficiency in most cases. For the CNN+LSTM model, DeepSAVA increases the FR slightly compared with the baselines; while for the I3D model, we can see that the FR grows significantly. The BO selection process is also essential for I3D. Without BO, only about half of the test videos can be attacked successfully; after applying BO, the FR increases to nearly 100%. As for the Inception-v3 model, the FR increases when applying DeepSAVA. It can be concluded that the CNN+LSTM is the most robust classification model among the three classifiers. Although the I3D has the highest classification accuracy, it is more vulnerable to attacks, even when only one frame is modified. That might happen because the I3D model relies heavily on the integral structure of the video itself and some frames may be more important.

max iter	FR	max(lp)	max(ssim)	ave(lp)	ave(ssim)
30	0.5	0.11	0.094	0.052	0.069
50	0.5	0.135	0.094	0.059	0.099
80	0.529	0.131	0.0959	0.0595	0.081
100	0.529	0.131	0.095	0.052	0.067

Table 4: The relationship between the iteration, $l_{1,2}$, SSIM, and Fooling Rate for the I3D model with combined perturbation on UCF101.

$l_{2,1}$-norm						
Constraint	$l_{2,1}$ budget = 0.08			$l_{2,1}$ budget = 0.09		
Method	Sparse	DeepSAVA(no BO)	DeepSAVA	Sparse	DeepSAVA(no BO)	DeepSAVA
FR	40.51%	48.1%	88.61%	41.77%	54.43%	93.67%
Time (s)	8018.9	2629	1535.8	14001	3729	1573.82
$SSIM$						
Constraint	$SSIM$ budget = 0.98			$SSIM$ budget = 0.96		
Method	Sparse	DeepSAVA(no BO)	DeepSAVA	Sparse	DeepSAVA(no BO)	DeepSAVA
FR	8.06%	16.56%	35.44%	10.1%	51.9%	96.20%
Time (s)	5842.32	1285.1	1424.4	13789.23	5633.28	1545.5

Table 5: Attack I3D model on UCF101 dataset under $l_{2,1}$ and $SSIM$ constraint separately.

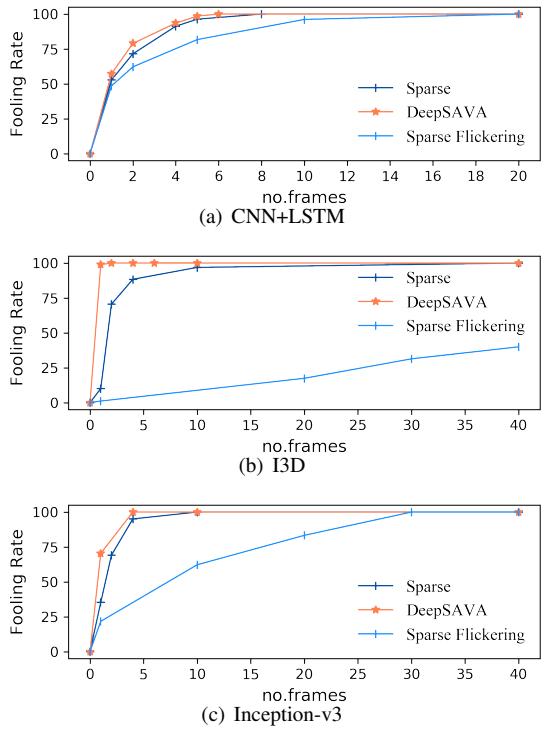


Figure 6: Fooling Rate of attacking a different number of frames across three classifiers.

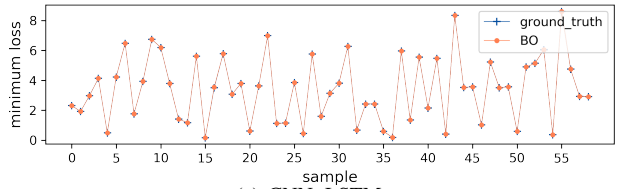
We find that the position of keyframes is related to the classifiers evaluated: for CNN+LSTM, the frames in the front are selected more often, and for other CNN networks, the position is a variant. Thus, it is reasonable that the BO cannot improve the FR for the CNN+LSTM model as much as the I3D, as we attacked the first frame when not selecting it. We also show the results in Figure 6 for attacking a different number of frames across I3D, CNN+LSTM, and inception-v3 models. It can be seen that the more frames attacked, the higher the fooling rate obtained.

Fixed $l_{2,1}$ norm and SSIM: For the purpose of a fair comparison, we also present the results under fixed $l_{2,1}$ and SSIM budgets for perturbing only one frame. The maximum allowed iteration is set to 500 to limit the time. As the baseline methods are based on the l_p norm and our method is based on SSIM, we take experiments under the same constraint l_p norm and the SSIM constraint, respectively. Based on the results of fixed iterations, we randomly select 200 videos from different categories to attack the I3D model on the UCF101 dataset. During the experiments, the Sparse Flickering spent days to achieve the constraint, thus we will only compare it with the Sparse (Wei et al., 2019) attack. In (Fezza et al., 2019), the SSIM budget for attacking image is set to 0.95, thus we choose the SSIM constraints above 0.95. In (Yang et al., 2010), it states that the difference between the images is imperceptible when the $l_{2,1}$ score is 4, given that we also set the $l_{2,1}$-norm budget to below 0.1 (since

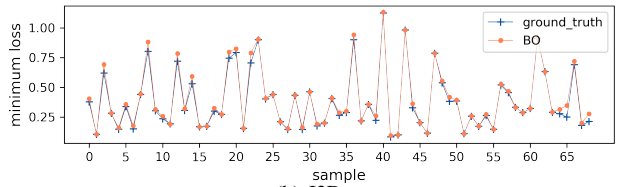
Models	Attack Method	UCF101			
		FR	ANI	AAP($l_{1,2}$)	AAP(SSIM)
CNN+LSTM	Sparse	54.31%	15.31	0.054	0.043
	DeepSAVA(without BO)	56.94%	7.87	0.077	0.060
	DeepSAVA(BO)	57.11%	8.01	0.071	0.058
I3D	Sparse	11.22%	49	0.092	0.079
	DeepSAVA(without BO)	48.78%	11.34	0.0857	0.054
	DeepSAVA(BO)	99.89%	5.74	0.055	0.0233
Inception-v3	Sparse	41.84%	38.21	0.062	0.0512
	DeepSAVA(without BO)	65.14%	14.88	0.072	0.052
	DeepSAVA(BO)	77.49%	11.43	0.071	0.0508

Table 6: Comparison with Sparse baseline, DeepSAVA without BO and with BO on different models by only perturbing one frame. Gray cell shows the best results.

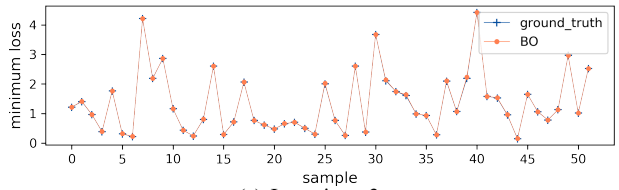
$0.1 * 40 = 4$, as we have 40 frames). As we can see in Table B.14, under small fixed budgets, DeepSAVA outperforms Sparse (Wei et al., 2019) in both cases in terms of FR and total time.



(a) CNN+LSTM



(b) I3D



(c) Inception-v3

Figure 7: Minimum loss selected by BO and brute force search along videos

4.2.2. *Average Absolute Perturbation*

Average Absolute Perturbation (AAP) is introduced to measure the perturbation level for each method. As mentioned previously, the sparse Flickering adds a small perturbation per frame, but cannot obtain comparable results to ours. Thus, we choose the pure sparse attack (Sparse) as the main baseline to show the average absolute perturbation. As the baseline is guided by $l_{1,2}$ norm and ours is based on SSIM loss, we will record the average perturbation of $l_{1,2}$ and SSIM separately. To achieve a fair comparison, we set the maximum $l_{1,2}$ norm ball constraint as 0.1 and the maximum SSIM constraint as 0.92. Suppose the fooling rate is f, and distant matrix is D, which can be set to (1-SSIM) or $l_{1,2}$ norm, thus the average absolute perturbation(AAP) can be represented as:

$$AAP(D) = \frac{\sum_N D(V_{adv} - V_{original})}{N} * f + D_{max} * (1 - f),$$

where V_{adv} denotes the generated adversarial video that could successfully mislead the classifier and D_{max} is the maximum constraint; N is the number of adversarial samples achieving a successful attack. We run experiments on 200 randomly selected videos of the UCF101 dataset and record the results of FR, ANI, AAP($l_{1,2}$), and AAP(SSIM) in Table 6. From the results, we can see that for the model I3D and Inception-v3, our method could achieve better performance in terms of efficiency, fooling rate, and AAP(SSIM). For the CNN+LSTM model, our method engages a higher fooling rate and spends less time, and the AAP($l_{1,2}$) and AAP(SSIM) are also acceptable compared with the baseline model.

4.2.3. *Visualization of Results*

The generated adversarial frames by DeepSAVA are presented in Figure 8. Because of the spatial transformation, the frame looks a little bit shaky but not obvious to human eyes. In fact, in the real world, it is normal to see that there are a few frames with instabilities during video shooting and transmitting. That's why we apply spatial

Approach	CNN+LSTM			Inception-v3	I3D
	Mask1	Mask2	Mask4	Mask1	Mask1
Fixed the frame (first n-th)					
$\mathcal{D}$	52.77% ± 2.24%	71.51% ± 1.5%	91.28% ± 1.95%	42.45% ± 4.30%	10.12% ± 1.19%
$\mathcal{S}$	55.27% ± 1.82%	74.33% ± 1.36%	91.89% ± 1.45%	63.91% ± 5.61%	29.99% ± 2.36%
$\mathcal{D}+\mathcal{S}$	56.22% ± 1.65%	77.36% ± 1.86%	92.99% ± 1.85%	68.86% ± 1.83%	47.57% ± 2.64%
Using BO to choose frame					
$\mathcal{D}+\mathcal{S}$	57.22% ± 1.36%	78.95% ± 1.93%	93.51% ± 1.33%	70.39% ± 2.78%	99.89% ± 0.11%

Table 7: Attack Fooling Rate on CNN+LSTM, Inception-v3, and I3D models with combining noise ($\mathcal{D}$) and spatial transformation ($\mathcal{S}$) by modifying a different number of frames on UCF101 dataset; Mask N means that N frames are modified.

Approach	CNN+LSTM		Inception-v3		I3D		time (s)
	FR	average loss	FR	average loss	FR	average loss	
BO Selection	55.81%	0.21	72.22%	3.39	100%	1.35	16.1
brute force search	55.81%	0.27	72.22%	3.39	100%	1.35	70.4

Table 8: Fooling Rate, average selected maximum loss and average time spent for one video of BO Selection and Brute Force Search.

transformation in video attacks to improve the efficiency and fooling rate. In practice, a distortion in one frame of a video is less noticeable than a static image since this specific frame only appears for 0.047 seconds in human eyes (Xiao et al., 2018a). We could also see that it does not lead to a noticeable perturbation as shown by our video demos.

When transmitting the videos in the real world, the generated frames need to be compressed into videos first and then decompressed into frames. We found that the additive-only perturbed frames, may not remain adversarial examples after such transmission. Our experiments demonstrate that DeepSAVA can be immune to short video compression due to the fact that perturbation based on spatial transformation can be well preserved during compression while additive perturbation may disappear.

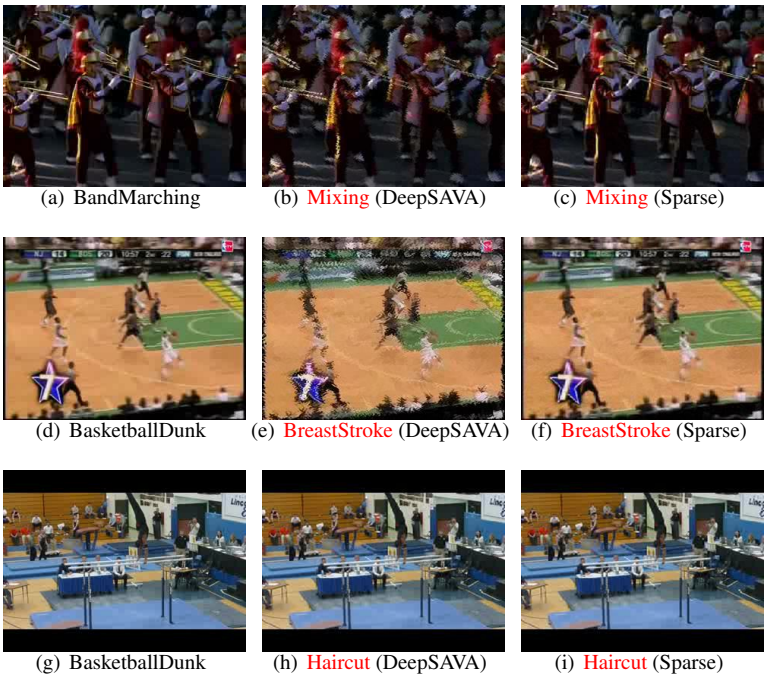
4.2.4. Ablation study

We perform ablation experiments to study the effects of combined perturbation for a different number of attacked frames by comparing with additive noise-only and spatial transformation-only perturbations, and the effects of BO selection. Table 7 shows the FR for three classifiers on the UCF101 dataset. Four approaches are taken to attack the model: 1) only noise ($\mathcal{D}$), 2) only spatial transformation ($\mathcal{S}$), 3) a combination of additive perturbation and spatial transformation ($\mathcal{D}$ + $\mathcal{S}$), and 4) combined perturbation with BO selection. To make more comprehensive evaluations on the superiority of combination, we attack a different number of frames for the CNN+LSTM model as it has the lowest FR when only perturbing one frame.

As shown in Table 7, using only spatial transformation perturbations results in a higher fooling rate compared to using only additive noise perturbations, highlighting the effectiveness of spatial transformations. The experimental results demonstrate a significant increase in the fooling rate of both the Inception-v3 and I3D models. Furthermore, combining spatial transformation and additive noise perturbations leads to an even higher fooling rate, indicating a stronger attack strategy. Notably, when attacking the I3D model, the combination perturbation increases the fooling rate by approximately four times compared to using only additive perturbation, revealing the significant impact of combination perturbations on I3D models. These findings collectively highlight the ability of combined perturbations to increase the fooling rate, with the additional effectiveness of using BO selection, particularly for the I3D model.

4.2.5. The Accuracy of Bayesian Optimisation Selection

To justify whether the Bayesian Optimisation could select the most critical frames, we take the brute force search experiments to obtain the upper bound of the performance: when the selection frame is 1, we select the keyframe manually one by one of the video, and then record the maximum loss found by the search. We randomly select 100 videos from UCF101 in different categories. The fooling rates, average maximum loss, and average time spent for one video on three models are shown in Table 8. We also compare the selected maximum loss by BO and brute force search along the video samples in Figure 7. The results demonstrate that we can obtain the same results as the brute force search, but



(a) BandMarching (b) Mixing (DeepSAVA) (c) Mixing (Sparse)

(d) BasketballDunk (e) BreastStroke (DeepSAVA) (f) BreastStroke (Sparse)

(g) BasketballDunk (h) Haircut (DeepSAVA) (i) Haircut (Sparse)

Figure 8: Original, and adversarial examples generated by DeepSAVA and Sparse (Wei et al., 2019) when only one frame in the video is perturbed. The red labels are the wrong predictions. The target model for (a)-(b) is CNN+LSTM; for (d)-(f) is Inception-v3; for (g)-(i) is I3D.

spend much less time, which confirms the effectiveness of BO optimisation.

4.2.6. *Effects of λ*

To decide the value of λ, we applied the DeepSAVA without BO selection on 200 randomly selected videos of the UCF101 dataset to evaluate the effect of λ. The average success perturbation (ASP) is the average of the SSIM score of perturbation for the adversarial examples that could mislead the model successfully:

$$ASP(SSIM)) = avg(SSIM(V_{adv} - V_{original})),$$

where V_{adv} denotes the generated adversarial video that could successfully mislead the classifier, and $V_{original}$ is the original video. The results of applying $\lambda = 0.8, 1.0, 1.5$ on three models are presented in Table 9. We can see that the bigger the λ, the lower the FR while the lower the perturbation. While, for the CNN+LSTM model, the fooling rate remains the same across all tested λ values, but the perturbation level is the lowest at $\lambda = 1.5$. Thus, we choose $\lambda = 1.5$ for the CNN+LSTM model and $\lambda = 1.0$ for I3D and Inception-v3 models to trade off performance in terms of the fooling rate and average success perturbation.

Models	λ value	FR	ASP(SSIM)
CNN+LSTM	0.8	56.94%	0.0429
	1.0	56.94%	0.0412
	1.5	56.94%	0.0401
I3D	0.8	51.22%	0.0316
	1.0	48.78%	0.0268
	1.5	48.17%	0.0198
Inception-v3	0.8	66.05%	0.0534
	1.0	65.14%	0.0518
	1.5	64.22%	0.0454

Table 9: The results of DeepSAVA(without BO) on UCF101 dataset for different λ values.

4.3. Adversarial Training

In this section, we show the results of two adversarial training methods. The first method is the most intuitive one, as demonstrated in Figure 5 (a), which first obtains adversarial examples by performing DeepSAVA and then feeds these adversarial examples into the training stage. Another method is the algorithm demonstrated in Section 3, which modifies the training loop by injecting PGD adversarial attacks. To evaluate the power of adversarial defences, we randomly picked 200 video samples covering 101 classes from the test dataset of UCF101 and then perform the DeepSAVA and Sparse (Wei et al., 2019) attack on the first frame of the video to compare the fooling rate. The model recognition accuracy on clean data is also recorded.

For the first method, we present the adversarial training results for the CNN+LSTM model in Table 10. We randomly choose 8000 videos from the training dataset. To perform the adversarial training, we randomly generate 1000, 1500, and 3000 adversarial examples by applying the DeepSAVA attack, and feed these adversarial videos with the 8000 unmodified videos as input to the training stage. For the model without defence, we change the adversarial examples to unmodified videos as input to train the model. As we can see from the results, comparing the defended model with the undefended model, we obtained comparable model recognition accuracy and a lower fooling rate, which demonstrates that adversarial training can improve the robustness of the model. Additionally, as expected, the results also indicate that the more adversarial examples injected, the lower the fooling rate obtained by the defended model and the larger gap of reduced fooling rate compared with the undefended model. Thus, the more adversarial examples injected into the training stage, the more robust the model is against the DeepSAVA attack. However, generating loads of adversarial examples to train on is extremely expensive in terms of time and resources. Therefore, this encourages us to use a more effective adversarial training algorithm, as demonstrated earlier, to modify the training process by incorporating adversarial attacks.

For the PGD adversarial training, the model recognition accuracy and fooling rate results are presented in Table 11. As the CNN+LSTM model implements a pre-trained CNN model to extract features first and then feeds these features to train the LSTM model, in order to perform defence, we perform adversarial training to train the CNN model first, and then implement the retrained CNN to extract features. As we can see from the table, after applying the combined perturbation adversarial training, we can obtain lower fooling rates on both DeepSAVA and Sparse attacks. Looking at the results, we find that the more powerful perturbation added in the adversarial training process can lead to a more robust model against both the additive perturbation-only attack and combined perturbation attack, which confirms the effectiveness for a broader range of perturbation when performing adversarial training.

4.4. Transferability Across Recurrent Models

Assessing the transferability across models is crucial to evaluate the performance of adversarial attacks, which can be approached as a black-box problem that does not require access to the target model's parameters. Our research investigates the transferability of attacks on I3D and Inception-v3, which only use CNN, versus recurrent neural networks (RNN) such as CNN+LSTM, which incorporate time-related networks. As videos have a unique time-related structure, we conducted extensive experiments to evaluate the transferability across various time-related networks. Our experiments were performed on the UCF101 dataset for Inception-v3, I3D, and different RNNs. For recurrent models, we first extracted the features of the original videos using the CNN (Inception-v3) model and then fed them into the vanilla RNN (Rumelhart et al., 1986), LSTM (Hochreiter and Schmidhuber, 1997), and GRU (Cho et al., 2014) networks, respectively. The training precision of the vanilla RNN and GRU networks was 65. 16% and 73. 05%, respectively.

Figure 6 indicates that Sparse (Wei et al., 2019) outperforms Sparse Flickering in terms of FR. Therefore, we use Sparse (Wei et al., 2019) as our baseline method. Table 12 presents the fooling rates (FR) of the videos generated in various models. In the table, the rows indicate the models used to generate adversarial videos, while the columns represent the target attack classifiers. To increase the success rate of the attack, we disturb seven frames of each video. We use the adversarial examples generated from the white-box attack for transferability assessment, which results in a 100% FR on the diagonal. We then use these adversarial examples to attack other models (like a black-box attack), as described in Table 12. Our approach has a higher FR compared to the baseline, indicating better transferability performance. The difference between vanilla RNN and other models is that vanilla RNN lacks a memory component, which results in weak performance on video classification tasks. It is observed that adversarial videos generated from LSTM and GRU models can successfully fool vanilla RNNs. Furthermore, the FR across the GRU and LSTM models is around 85%, indicating good transferability between recurrent models with memory. However, Table 12 shows that transferability from RNNs to CNNs is

Models	8000+1000		8000+1500		8000+3000	
	Acc.	FR	Acc.	FR	Acc.	FR
CNN+LSTM With defence	**59.77%**	**64.42%**	62.06%	**59.26%**	63.42%	**58.92%**
CNN+LSTM without defence	59.19%	66.99%	**63.79%**	63.06%	**64.36%**	63.06%
defended vs. undefended	+0.98%	-3.8%	-2.7%	-6.0%	-1.46%	-6.6%

Table 10: Model Accuracy and fooling rate on different size of training datasets for the CNN +LSTM model.

defence	Model	Accuracy	DeepSAVA	Sparse
Combined defence	inception-v3	63.96%	**37.62%**	**25.29%**
PGD defence (Madry et al., 2018)	inception-v3	64.52%	38.53%	27.21%
Without defence	inception-v3	**65.12%**	40.59%	29.6%
Combined defence	CNN+LSTM	68.02%	**47.86%**	**46.15%**
PGD defence (Madry et al., 2018)	CNN+LSTM	69.00%	52.89%	52.06%
Without defence	CNN+LSTM	**70.22%**	55.93%	55.26%
Combined defence	I3D	87.5%	**77.3%**	**42.3%**
PGD defence (Madry et al., 2018)	I3D	88.2%	78.5%	46.2%
Without defence	I3D	**89.1%**	80.2%	47.2%

Table 11: Model Accuracy and fooling rate on different models with defence and without defence.

not as good as that from CNNs to RNNs. The fooling rates for attacking RNN models are higher than those for attacking I3D and Inception-v3 models. This could be because the I3D model has the highest training accuracy and therefore has the lowest fooling rate when subjected to a black-box attack. Additionally, due to its low training accuracy, the Vanilla RNN model achieves the highest fooling rate when attacking the unseen Vanilla RNN model. In conclusion, our method achieves better transferability than the Sparse baseline.

5. Discussion and Conclusion

In this paper, we apply spatial transformation perturbation and additive noise to attack as few frames as possible to obtain sparse adversarial videos. The most influential frames to be attacked are selected by a joint optimisation strategy with Bayesian optimisation (BO) and SGD-based optimisation. We take sufficient experiments to examine the power of BO and show the effectiveness of BO selection in this task. Additionally, the quality of generated adversarial examples is measured by SSIM instead of l_p-norm, which can capture both additive noise and spatial transformation effectively. We propose the novel and effective video attack mechanism, DeepSAVA, and perform extensive experiments to evaluate its performance on the UCF101 and HMDB51 action dataset and three different classification models: Inception-3v, CNN+LSTM, and I3D. We obtain better results than state-of-the-art sparse baselines in terms of both fooling rate and transferability, which confirms the success of DeepSAVA. Our most significant results are for the I3D model, by only attacking one frame of the video to obtain a 99.5% to 100% attack success rate.

Additionally, in this paper, we add adversarial training experiments to improve the robustness of video classification models. By now, adversarial training research

Models	LSTM		Vanilla RNN		GRU		Inception-v3		I3D	
	Sparse	DeepSAVA	Sparse	DeepSAVA	Sparse	DeepSAVA	Sparse	DeepSAVA	Sparse	DeepSAVA
LSTM	100%	100%	34.42%	41.38%	64.35%	85.34%	50.0%	52.17%	53.48%	54.62%
Vanilla RNN	100%	100%	100%	100%	100%	100%	71.74%	82.40%	60.50%	64.02%
GRU	79.34 %	84.75%	40.70%	56.03%	100%	100%	50.0%	51.08%	42.68%	49.58%
Inception-v3	22.95 %	24.36%	22.80%	26.72%	22.90%	31.03%	100%	100%	33.61%	37.80%
I3D	6.56 %	10.08%	7.01%	9.48%	7.64%	8.62%	13.04%	14.13%	100%	100%

Table 12: Fooling Rate across recurrent models on UCF101.

focused on image classification models, thus, in this paper, we choose to adopt the most effective defence method, PGD adversarial training, to retrain the video classifiers. We modify the adversarial training algorithm by adding a combination of spatial transformation and additive perturbation in light of our DeepSAVA framework. We also show the experimental results of the most intuitive adversarial training approach, which takes the clean training dataset and the generated adversarial examples as input to re-train the model. As a result, after applying our adversarial training with combined perturbation, we can obtain a more robust model compared to the PGD adversarial training, and more effective than injecting adversarial examples.

Limitation and future work

We acknowledge certain limitations in our proposed approach. Firstly, Bayesian optimisation is a general tool that can be readily employed to identify critical frames. However, it is more time-consuming than the gradient-map method, which leverages feature maps to pinpoint critical frame features. Therefore, going forward, we can explore to emplement a method that can combine optimisation and gradient-map approaches, such as Gumbel-Softmax Jang et al. (2017), which represents a differentiable variant of SemHash Kaiser and Bengio (2018).

Additionally, we have observed that the enhancement in the robustness of the video classification model is somewhat limited. The most significant reduction in the fooling rate we achieved was 24.% for the CNN+LSTM model, leaving ample room for further improvement. Furthermore, with respect to the use of adversarial defence through adversarial training to enhance the robustness of video models, our combined perturbation adversarial training demonstrates a significantly lower fooling rate compared to traditional adversary training with PGD. However, it is essential to acknowledge that our algorithm introduces a subtle alteration in the accuracy of the clean model, as shown in Table 11. In our experimental endeavors, we also explored the TRADES approach proposed by Zhang et al. (2019), which aims to balance the model accuracy and robustness. Regrettably, it showed worse performance when applied to video classification models. As a result, we suggest that future research should focus on the development of effective trade-off algorithms tailored specifically for video classification models.

Ethical Implications

Video attack technologies present significant ethical and security challenges. These technologies are capable of generating misleading or false content, posing a risk to individual rights, particularly regarding reputation and privacy. A single perturbed frame in an input video can cause the model to produce entirely incorrect predictions. This vulnerability can be employed by malicious actors to undermine security systems or generate harmful content.

Conversely, defense models in video technology play a pivotal role in maintaining cybersecurity and safeguarding sensitive data. They are critical in preserving the information, which is vital in an area where social media is a primary source of information. Defence models are essential to protect public information and ensure robustness against the manipulation of video content. However, the implementation of such defensive technologies should carefully consider the balance between enhancing robustness and maintaining model accuracy.

Acknowledgments

This work was supported by Partnership Resource Fund (PRF) *Towards the Accountable and Explainable Learning-enabled Autonomous Robotic Systems (AELARS)*, funded via the UK EPSRC projects on Offshore Robotics for Certification of Assets (ORCA) [EP/R026173/1]. This work was also funded by the Faculty of Science and Technology of Lancaster University. We also thank the High-End Computing facility at Lancaster University for the computing resources, and Abdulrahman Kerim and Washington L. S. Ramos for the thorough reviews.

Declaration of competing interes

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work.

Data availability

We share our code on `https://github.com/TrustAI/DeepSAVA` and present the generated adversarial videos in `https://www.youtube.com/channel/UCBDswZC2QhBhTOMUFNLchCg`. All data are public datasets.

Appendix A. Calculating the Gradient of SSIM

The SSIM was first proposed in (Wang and Bovik, 2002) and is detailed in (Wang et al., 2004). Given x and $\hat{x}$ as the local pixels taken from the same location of

the same frame in the clean video and adversarial video, respectively, the local similarity between them can be computed on three aspects: structures ($s(x, \hat{x})$), contrasts ($c(x, \hat{x})$), and brightness values ($b(x, \hat{x})$). The local SSIM is formed by these terms (Wang et al., 2004):

$$S(x, \hat{x}) = s(x, \hat{x}) \cdot c(x, \hat{x}) \cdot b(x, \hat{x}) = \left(\frac{\sigma_{x\hat{x}} + D_1}{\sigma_x \sigma_{\hat{x}} + D_1}\right) \cdot \left(\frac{2\sigma_x \sigma_{\hat{x}} + D_2}{\sigma_x^2 + \sigma_{\hat{x}}^2 + D_2}\right) \cdot \left(\frac{2\mu_x \mu_{\hat{x}} + D_3}{\mu_x^2 + \mu_{\hat{x}}^2 + D_3}\right), \tag{A.1}$$

The structural similarity index (SSIM) measure in Equation (2) can be expressed as: (Wang et al., 2004)

$$\mathrm{SSIM}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{(2\mu_x \mu_{\hat{x}} + C_1)(2\sigma_{x\hat{x}} + C_2)}{\left(\mu_x^2 + \mu_{\hat{x}}^2 + C_1\right)\left(\sigma_x^2 + \sigma_{\hat{x}}^2 + C_2\right)} \tag{A.2}$$

The mean of x , the variance of x, and co-variance of x and $\hat{x}$ can be represented as μ_x, σ_x^2 and $\sigma_{x\hat{x}}$. They can be calculated respectively:

$$\begin{aligned} \mu_x &= \frac{1}{N_P}\left(\mathbf{1}^T \cdot \mathbf{x}\right) \\ \sigma_x^2 &= \frac{1}{N_P - 1}(\mathbf{x} - \mu_x)^T (\mathbf{x} - \mu_x) \\ \sigma_{x\hat{x}} &= \frac{1}{N_P - 1}(\mathbf{x} - \mu_x)^T (\hat{\mathbf{x}} - \mu_{\hat{x}}) \end{aligned} \tag{A.3}$$

Given x and $\hat{x}$ as the local pixels taken from the same location of the same frame in the clean video and adversarial video, respectively, the local similarity between them can be computed on three aspects: structures ($s(x, \hat{x})$), contrasts ($c(x, \hat{x})$), and brightness values ($b(x, \hat{x})$). The local SSIM is formed as (Wang et al., 2004):

$$S(x, \hat{x}) = s(x, \hat{x}) \cdot c(x, \hat{x}) \cdot b(x, \hat{x}) = \left(\frac{\sigma_{x\hat{x}} + D_1}{\sigma_x \sigma_{\hat{x}} + D_1}\right) \cdot \left(\frac{2\sigma_x \sigma_{\hat{x}} + D_2}{\sigma_x^2 + \sigma_{\hat{x}}^2 + D_2}\right) \cdot \left(\frac{2\mu_x \mu_{\hat{x}} + D_3}{\mu_x^2 + \mu_{\hat{x}}^2 + D_3}\right), \tag{A.4}$$

where μ_x and $\mu_{\hat{x}}$ denote means, σ_x and $\sigma_{\hat{x}}$ are standard deviations of x and $\hat{x}$, respectively; $\sigma_{x\hat{x}}$ represents the cross-correlation of x and $\hat{x}$ after deleting means; D1, D2, and D3 are weight parameters. For SSIM metric, a value of 1 means that the two images compared are the same. As the SSIM is calculated based on the pixel level, it uses a sliding window method, which moves pixel by pixel the window across the whole image. As we use uniform pooling to combine the total SSIM for the whole videos, suppose we have N pixels in the total videos, the SSIM can be represented as:

$$\mathrm{SSIM}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{\sum_{i=1}^{N} \cdot \mathrm{SSIM}(\mathbf{x}_i, \hat{\mathbf{x}}_i)}{N} \tag{A.5}$$

where x_i and $\hat{x}_i$ are the i-th pixel of each frame in the video. To apply the gradient decent optimisation method described in Section 3, we have to compute the gradient of SSIM with respect to the adversarial video example $\hat{\mathbf{X}}$. As equation (9) shows, to compute $\vec{\nabla}_{\hat{\mathbf{X}}} \mathrm{SSIM}(\mathbf{X}, \hat{\mathbf{X}})$, we only need to calculate the gradient $\vec{\nabla}_{\hat{x}_i} \mathrm{SSIM}(x_i, \hat{x}_i)$. The process is represented as follows. (Wang and Simoncelli, 2004) We define four parameters to deduce the derivative of local SSIM:

$$\begin{aligned} M_1 &= 2\mu_x \mu_{\hat{x}} + C_1, \quad M_2 = 2\sigma_{x\hat{x}} + C_2 \\ P_1 &= \mu_x^2 + \mu_{\hat{x}}^2 + C_1, \quad P_2 = \sigma_x^2 + \sigma_{\hat{x}}^2 + C_2 \end{aligned} \tag{A.6}$$

Therefore, the gradient can be expressed as:

$$\begin{aligned} \nabla_{\hat{x}} SSIM(x, \hat{x}) = \tfrac{2}{N_P P_1^2 P_2^2}[M_1 P_1 (M_2 x - P_2 \hat{x}) \\ + P_1 P_2 (M_2 - M_1)\mu_x + M_1 M_2 (P_1 - P_2)\mu_{\hat{x}}] \end{aligned} \tag{A.7}$$

Appendix B. Step-size in adversarial attack during adversarial training

In this section, we present the experimental results of the adversarial training accuracy and fooling rate is given different values of α. Here we perform the adversarial training on the Inception-v3 model using the UCF101 dataset. According to the experimental results presented in Table B.13, altering the step size between 0.01 and 0.001 can have a minor effect on model accuracy and fooling rate. Furthermore, the results indicate that the model trained using combined perturbation is consistently more robust than the model trained using PGD defence.

Alpha	Defence	Accuracy	Fooling Rate
0.001	Combined defence	60.8%	**34.3%**
0.001	PGD defence	60.8%	37.1%
0.005	Combined defence	**62.6%**	**35.5%**
0.005	PGD defence	60.8%	37.1%
0.01	Combined defence	58.3%	**32.1%**
0.01	PGD defence	**58.9%**	36.8%

Table B.13: Model Accuracy and fooling rate for different step-sizes of adversarial attack in the adversarial training.

Appendix C. Case Study I: Next-Frame Video Prediction

In this section, we explore our framework in the application of next-frame video prediction. Next-frame video prediction aims to predict the next frame based on a sequence of previous videos. Various models have been applied for the video prediction tasks, i.e., CNN

Constraint	$l_{2,1}$ budget = 0.1			SSIM budget = 0.94		
Methods	Sparse	DeepSAVA(no BO)	DeepSAVA	Sparse	DeepSAVA(no BO)	DeepSAVA
FR	60%	58.22%	**91.25%**	12.9%	66.2%	**97.46%**
Time (s)	24029.8	4109.78	**1483.96**	30803.2	8276.1	**1586.3**

Table B.14: Attack I3D model on UCF101

Constraint	$l_{2,1}$ budget = 0.08			$l_{2,1}$ budget = 0.1		
Methods	Sparse	DeepSAVA(no BO)	DeepSAVA	Sparse	DeepSAVA(no BO)	DeepSAVA
FR	**55.71**%	48.57%	50%	**58.57**%	**58.57**%	57.14%
Time (s)	22800.5	**13777.6**	15010	23336.8	**19774.4**	20866.4

Table B.15: Attack CNN+LSTM model on UCF101 with $l_{2,1}$ budget

Constraint	$SSIM$ budget = 0.96			$SSIM$ budget = 0.94		
Methods	Sparse	DeepSAVA(no BO)	DeepSAVA	Sparse	DeepSAVA(no BO)	DeepSAVA
FR	**50**%	47.14%	47.14%	**52.85**%	**52.85**%	**52.85%**
Time (s)	15120.21	**4039.24**	5131.24	15952.52	**6341.7**	7433.3

Table B.16: Attack CNN+LSTM model on UCF101 with $SSIM$ budget

and LSTM (Joshi, 2021), PredNet (Lotter et al., 2017), and Transformer (Kumar, 2021). As a case study, we will apply our DeepSAVA framework to the video prediction model based on CNN and LSTM models to see whether the sparse attack can also be effective on the frame prediction task.

In the CNN and LSTM architecture, the CNN acts as an encoder that extracts spatial features from the input video sequences, while the LSTM decodes the temporal connections between the frame and video sequence to forecast the next frame. Next-frame video prediction based on combining CNN and LSTM networks offers a wide range of applications in computer vision, including video reduction, editing, and creation, and also demonstrated significant potential to predict the next-frame of videos. In this paper, we examined this model using the moving-MNIST dataset, which is a common benchmark dataset for videos.

In next-frame prediction, the model inputs a sequence of the previous frame, f_n, to predict a new frame, $f_{(n+1)}$. Therefore, it takes a sequence of input frames (x_n) as input, to output the prediction frame $y_{(n+1)}$. To be noticed, in the video prediction task, the accuracy is measured by the Mean Square Error or SSIM similarity metric.

Appendix C.1. Problem Definition:

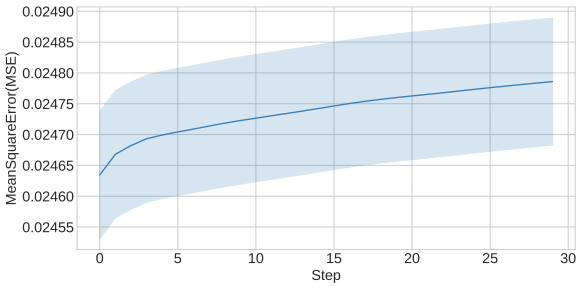
The video prediction model is defined as $J(\cdot;\boldsymbol{\theta})$ with pre-trained weights $\boldsymbol{\theta}$. The input clean video is defined as $\mathbf{X} = (x_1, x_2, ..., x_T)$ and the perturbed example is denoted as $\hat{\mathbf{X}} = (\hat{x}_1, \hat{x}_2, ..., \hat{x}_T)$. The video sequence containing the next frame of the input video is $y = (x_2, x_3, ..., x_{T+1}) \in \mathbb{R}^{T\times W\times H\times C}$, which is used as the ground truth label for the prediction output. Therefore, the objective function for perturbing the task of frame prediction can be defined as:

$$\arg\min\left(\lambda\ell_{similar}(\hat{\mathbf{X}}, \mathbf{X}) - \ell_{adv}\left(y, J(\hat{\mathbf{X}}; \boldsymbol{\theta})\right)\right), \qquad \text{(C.1)}$$

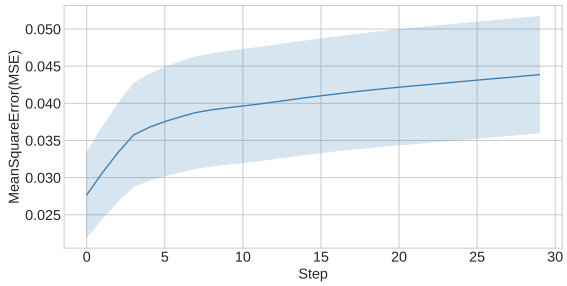
Here the loss function l_{adv} we used is the mean square error. As a result, the attack goal is to enlarge the distance between prediction frames and the ground truth video, while maintaining the perturbation human imperceptible.

Appendix C.2. Experiments

We perform the DeepSAVA on the next-frame video prediction model. We randomly picked 100 samples of the moving-MNIST dataset and plotted the MeanSquareError(MSE) values against the steps under DeepSAVA in Figure C.9. In Figure C.9 (a), we conducted attacks on the first frame of the input video, measuring their respective MSEs. Subsequently, we apply the BO selection to identify the most crucial frame, and the results are illustrated in Figure Figure C.9 (b). Notably, the BO selection consistently designated the last frame as the most critical. Furthermore, the results indicate that attacking the last frame chosen by BO resulted in significantly higher MSE values. Consequently, it can be inferred that the next-frame video prediction model relies heavily on the information contained in the last frame of the input video.



(a) First n-th frames



(b) With BO selection

Figure C.9: Per-step Mean Square Error (MSE) between the predicted frame and ground truth frame, while introducing perturbations to the input frames. The amount of perturbation added to the input frames is constrained to a budget within the l_2=0.03 norm ball.

Appendix D. Case Study II. Evaluate the robust model using explainable AI method.

To evaluate the regular model and robust mode, we use the salience map to show what features are learned by each model. The salience map can effectively highlight the pixels within each frame that have the greatest impact on the model's predictions. It serves as a valuable tool in the field of explainable AI, commonly used to evaluate input images or videos in a neural network, revealing which region of the image or frame contributes the most to the model's decision-making process.

A gradient-based salience map is designed to visualize the gradients of the predicted outcome from the model with respect to the input pixel values (Simonyan et al., 2014; Zeiler and Fergus, 2014; Springenberg et al., 2014). The relative contribution of each pixel to the final prediction of the model can be calculated by applying tiny tweaks to pixel values across the image and catching the change in the predicted class.

In this section, we perform experiments to evaluate the performance of the Inception-V3 model with defence and without defence. In Figure D.10, we display the salience map superimposed on the input frame. The background image represents the original video of a girl playing the flute, and the yellow pixels indicate areas with high gradients. We compare the salience maps learned by the regular model and the robust model, which was trained using the combination .

From the results, we observe that both models heavily rely on the central portion of the frame, particularly focusing on the flute, when making predictions. Additionally, numerous frames do not exhibit a salience map, suggesting that they do not contribute significantly to the model's final decision. However, when conducting an attack on a single frame of the video, it is observed that perturbing a frame without a salience map is more effective to result in a change in the model's predicted label compared to perturbing a frame highlighted by the salience map (highlighted in yellow). This finding raises important considerations, which we will discuss later. Furthermore, based on the experimental results, a notable difference between the regular model and the robust model is that there are more frames highlighted by the salience map. This observation suggests that the model with the defence method may be more robust due to its ability to learn more distinctive features at the individual frame level.

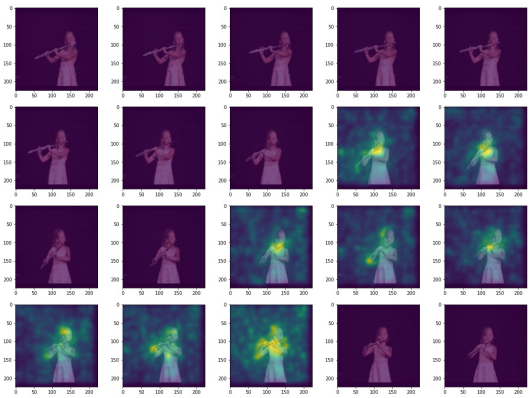
Appendix D.1. Discussion and Limitations

The experiment was taken by a Python package 'Keravis', which primarily focuses on generating salience

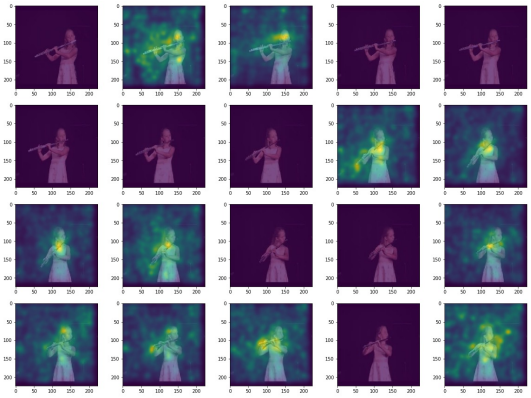
maps for individual images (Kotikalapudi and contributors, 2017). While this approach may not capture the full temporal dynamics and interactions between frames, it can provide some useful information if we set the input to each individual frame of video. Visualizing the salience map for each frame allows us to analyze the frame-level attention and identify the regions that the model focuses on for making predictions.

However, this approach is still limited for video model analysis. Since each frame is considered independently, the salience maps cannot find the temporal relationships present in the video. Some frames may not have prominent salience maps if they are less informative or contribute less to the model's predictions. Additionally, the salience maps might not provide a holistic view of the salient regions in the entire video.

Hence, to gain a more comprehensive understanding of the salience map and its temporal dependencies in the video, considering video-specific salience map generation techniques (Fang et al., 2014; Fan et al., 2019; Yang et al., 2013) would be more appropriate, which can be considered a future work. These techniques are designed to capture the dynamics and interactions across frames, providing a more accurate representation of the salient regions in the video.



(a) Regular Model



(b) Robust Model

Figure D.10: Saliency map overlaid on each frame. The regular model is the model that is not trained with adversarial examples, and the robust model is trained by combined perturbation.

Appendix E. Comparison Experiments with l_p-norm and SSIM Constraints

Appendix E.1. I3D model

In Table B.14, we show the results of comparing the fooling rate with the Sparse baseline on the I3D model using the UCF101 dataset when the $l_{2,1}$ budget is 0.1 and the SSIM budget is 0.94.

Appendix E.2. CNN+LSTM model

In Table B.15 and Table B.16, we show the comparison results for attacking the CNN+LSTM model using the UCF101 dataset. For the CNN+LSTM model, we can see that although the Sparse baseline could obtain a higher fooling rate, it will spend much more time to generate the adversarial examples. Our method using combined perturbation will spend less time and obtain a comparable fooling rate. Because here we select the first frame to attack, the BO cannot improve the performance of the I3D model.

References

Athalye, A., Carlini, N., Wagner, D., 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, in: International conference on machine learning, PMLR. pp. 274–283.

Bai, T., Luo, J., Zhao, J., Wen, B., Wang, Q., 2021. Recent advances in adversarial training for adversarial robustness. arXiv preprint arXiv:2102.01356 .

Bas, E., Tekalp, A.M., Salman, F.S., 2007. Automatic vehicle counting from video for traffic flow analysis, in: 2007 IEEE intelligent vehicles symposium, Ieee. pp. 392–397.

Berthier, N., Sun, Y., Huang, W., Zhang, Y., Ruan, W., Huang, X., 2021. Tutorials on testing neural networks. arXiv preprint arXiv:2108.01734 .

Buckman, J., Roy, A., Raffel, C., Goodfellow, I., 2018. Thermometer encoding: One hot way to resist adversarial examples, in: International Conference on Learning Representations.

Carlini, N., Wagner, D., 2017a. Adversarial examples are not easily detected: Bypassing ten detection methods, in: Proceedings of the 10th ACM workshop on artificial intelligence and security, pp. 3–14.

Carlini, N., Wagner, D., 2017b. Towards evaluating the robustness of neural networks, in: 2017 ieee symposium on security and privacy (sp), IEEE. pp. 39–57.

Carreira, J., Zisserman, A., 2017. Quo vadis, action recognition? A new model and the kinetics dataset, in: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308.

Chen, Z., Xie, L., Pang, S., He, Y., Tian, Q., 2021. Appending adversarial frames for universal video attack, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3199–3208.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar. pp. 1724–1734.

Deng, Y., Zheng, X., Zhang, T., Chen, C., Lou, G., Kim, M., 2020. An analysis of adversarial attacks and defenses on autonomous driving models, in: 2020 IEEE international conference on pervasive computing and communications (PerCom), IEEE. pp. 1–10.

Donahue, J., Hendricks, L.A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., Darrell, T., 2017. Long-term recurrent convolutional networks for visual recognition and description. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 677–691. doi:10.1109/TPAMI.2016.2599174.

Fan, D.P., Wang, W., Cheng, M.M., Shen, J., 2019. Shifting more attention to video salient object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 8554–8564.

Fang, Y., Wang, Z., Lin, W., Fang, Z., 2014. Video saliency incorporating spatiotemporal cues and uncertainty weighting. IEEE transactions on image processing 23, 3910–3921.

Feinman, R., Curtin, R.R., Shintre, S., Gardner, A.B., 2017. Detecting adversarial samples from artifacts. arXiv preprint arXiv:1703.00410 .

Fezza, S.A., Bakhti, Y., Hamidouche, W., Déforges, O., 2019. Perceptual Evaluation of Adversarial Attacks for CNN-based Image Classification, in: 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6. doi:10.1109/QoMEX.2019.8743213.

Fohr, D., Mella, O., Illina, I., 2017. New Paradigm in Speech Recognition: Deep Neural Networks, in: IEEE International Conference on Information Systems and Economic Intelligence, Marrakech, Morocco.

Fortran, I., Press, W., Teukolsky, S., Vetterling, W., Flannery, B., 1992. Numerical recipes. Cambridge, UK, Cambridge University Press .

Goodfellow, I.J., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples, in: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Gragnaniello, D., Marra, F., Verdoliva, L., Poggi, G., 2021. Perceptual quality-preserving black-box attack against deep learning image classifiers. Pattern Recognition Letters 147, 142–149.

Guo, C., Rana, M., Cisse, M., van der Maaten, L., 2018. Countering adversarial images using input transformations, in: International Conference on Learning Representations.

He, W., Wei, J., Chen, X., Carlini, N., Song, D., 2017. Adversarial example defense: Ensembles of weak defenses are not strong, in: 11th USENIX workshop on offensive technologies (WOOT 17).

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9, 1735–80. doi:10.1162/neco.1997.9.8.1735.

Huang, T., Menkovski, V., Pei, Y., Pechenizkiy, M., 2022. Bridging the performance gap between fgsm and pgd adversarial training. arXiv:2011.05157.

Huang, X., Jin, G., Ruan, W., 2012. Enhancement to safety and security of deep learning, in: Machine Learning Safety. Springer, pp. 205–216.

Huang, X., Kroening, D., Ruan, W., et al., 2020. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. Computer Science Review 37, 100270.

Jaderberg, M., Simonyan, K., Zisserman, A., et al., 2015. Spatial transformer networks, in: Advances in neural information processing systems, pp. 2017–2025.

Jang, E., Gu, S., Poole, B., 2017. Categorical reparameterization with gumbel-softmax, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net. URL: https://openreview.net/forum?id=rkE3y85ee.

Jiang, L., Ma, X., Chen, S., Bailey, J., Jiang, Y.G., 2019. Black-box adversarial attacks on video recognition models, in: Proceedings of the 27th ACM International Conference on Multimedia, pp. 864–872.

Jin, G., Yi, X., Huang, W., Schewe, S., Huang, X., 2022. Enhancing adversarial training with second-order statistics of weights. arXiv preprint arXiv:2203.06020 .

Jordan, M., Manoj, N., Goel, S., Dimakis, A.G., 2019. Quantifying perceptual distortion of adversarial examples. arXiv e-prints arXiv:arXiv: 1902.08265.

Joshi, A., 2021. Next-frame video prediction with convolutional lstms. https://keras.io/examples/vision/conv_lstm/.

Kaiser, L., Bengio, S., 2018. Discrete autoencoders for sequence models. CoRR abs/1801.09797. URL: http://arxiv.org/abs/1801.09797, arXiv:1801.09797.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L., 2014. Large-scale video classification with convolutional neural networks, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1725–1732. doi:10.1109/CVPR.2014.223.

Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J., 2017. Reluplex: An efficient smt solver for verifying deep neural networks, in: International conference on computer aided verification, Springer. pp. 97–117.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization. CoRR abs/1412.6980.

Kolter, Z., Madry, A., 2019. Adversarial robustness - theory and

practice. https://adversarial-ml-tutorial.org/.
Kong, Y., Fu, Y., 2018. Human action recognition and prediction: A survey. arXiv e-prints arXiv:1806.11230.
Kotikalapudi, R., contributors, 2017. keras-vis. https://github.com/raghakot/keras-vis.
Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T., 2011. HMDB: A large video database for human motion recognition, in: 2011 International Conference on Computer Vision, IEEE. pp. 2556–2563.
Kumar, R., 2021. Video predictions using transformer. URL: https://github.com/iamrakesh28/Video-Prediction.
Laidlaw, C., Feizi, S., 2019. Functional adversarial attacks, in: Advances in Neural Information Processing Systems, Curran Associates, Inc.. pp. 10408–10418.
Laidlaw, C., Singla, S., Feizi, S., 2020. Perceptual adversarial robustness: Defense against unseen threat models, in: International Conference on Learning Representations.
Li, B., Wang, S., Jana, S., Carin, L., 2020. Towards understanding fast adversarial training. arXiv preprint arXiv:2006.03089 .
Li, S., Neupane, A., Paul, S., Song, C., Krishnamurthy, S.V., Chowdhury, A.K.R., Swami, A., 2019. Stealthy adversarial perturbations against real-time video classification systems. In Proceedings of the 2019 Network and Distributed System Security Symposium doi:10.14722/ndss.2019.23202.
Liao, M., Lu, F., Zhou, D., Zhang, S., Li, W., Yang, R., 2020. Dvi: Depth guided video inpainting for autonomous driving, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16, Springer. pp. 1–17.
Liu, Y., Chen, X., Liu, C., Song, D., 2017. Delving into transferable adversarial examples and black-box attacks. ICLR .
Lotter, W., Kreiman, G., Cox, D., 2017. Deep predictive coding networks for video prediction and unsupervised learning. arXiv:1605.08104.
Lu, J., Sibai, H., Fabry, E., Forsyth, D.A., 2017. NO need to worry about adversarial examples in object detection in autonomous vehicles. CoRR abs/1707.03501. URL: http://arxiv.org/abs/1707.03501, arXiv:1707.03501.
Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2018. Towards deep learning models resistant to adversarial attacks, in: International Conference on Learning Representations.
Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B., 2017. On detecting adversarial perturbations. arXiv preprint arXiv:1702.04267 .
Mittal, V., Gangodkar, D., Pant, B., 2020. Exploring The Dimension of DNN Techniques For Text Categorization Using NLP, in: 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE. pp. 497–501.
Mohammadi, A., Bhattacharjee, S., Marcel, S., 2017. Deeply vulnerable: a study of the robustness of face recognition to presentation attacks. Iet Biometrics 7, 15–26.
Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P., 2016. DeepFool: a simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2574–2582.
Mu, R., Soriano Marcolino, L., Ruan, W., Ni, Q., 2021. Sparse adversarial video attacks with spatial transformations, in: 32nd British Machine Vision Conference 2021, BMVC 2021.
Nguyen, A., Yosinski, J., Clune, J., 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
Pony, R., Naeh, I., Mannor, S., 2021. Over-the-air adversarial flickering attacks against video recognition networks. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 515–524.
Raghunathan, A., Steinhardt, J., Liang, P.S., 2018. Semidefinite relaxations for certifying robustness to adversarial examples. Advances in Neural Information Processing Systems 31.
Ren, K., Zheng, T., Qin, Z., Liu, X., 2020. Adversarial attacks and defenses in deep learning. Engineering 6, 346–360.
Romeo, L., Marani, R., Petitti, A., Milella, A., D'Orazio, T., Cicirelli, G., 2020. Image-based mobility assessment in elderly people from low-cost systems of cameras: A skeletal dataset for experimental evaluations, in: Ad-Hoc, Mobile, and Wireless Networks: 19th International Conference on Ad-Hoc Networks and Wireless, ADHOC-NOW 2020, Bari, Italy, October 19–21, 2020, Proceedings 19, Springer. pp. 125–130.
Ruan, W., Yi, X., Huang, X., 2021. Adversarial robustness of deep learning: Theory, algorithms, and applications, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536.
Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T., 2019. Adversarial training for free! Advances in Neural Information Processing Systems 32.
Shen, D., Wu, G., Suk, H.I., 2017. Deep learning in medical image analysis. Annual Review of Biomedical Engineering 19, 221–248. PMID: 28301734.
Simonyan, K., Vedaldi, A., Zisserman, A., 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv:1312.6034.
Sinha, A., Namkoong, H., Duchi, J., 2018. Certifying some distributional robustness with principled adversarial training, in: International Conference on Learning Representations.
Song, Y., Kim, T., Nowozin, S., Ermon, S., Kushman, N., 2018. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples, in: ICLR (Poster).
Soomro, K., Zamir, A.R., Shah, M., 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 .
Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M., 2014. Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 .
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9.
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826.
Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R., 2014. Intriguing properties of neural networks, in: International Conference on Learning Representations.
Tanay, T., Griffin, L., 2016. A boundary tilting persepective on the phenomenon of adversarial examples. arXiv e-prints arXiv:arXiv:1608.07690.
Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M., 2015. Learning spatiotemporal features with 3D Convolutional Networks, in: Proceedings of the IEEE international conference on computer vision, pp. 4489–4497.
Wang, F., Fu, Z., Zhang, Y., Ruan, W., 2023a. Self-adaptive adversarial training for robust medical segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'23), Springer. pp. 725–735.
Wang, F., Xu, P., Ruan, W., Huang, X., 2023b. Towards verifying the geometric robustness of large-scale neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'23).

Wang, F., Zhang, C., Xu, P., Ruan, W., 2022. Deep learning and its adversarial robustness: A brief introduction, in: HANDBOOK ON COMPUTER LEARNING AND INTELLIGENCE: Volume 2: Deep Learning, Intelligent Control and Evolutionary Computation, pp. 547–584.

Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., Gu, Q., 2021. On the convergence and robustness of adversarial training. arXiv preprint arXiv:2112.08304 .

Wang, Z., Bovik, A.C., 2002. A universal image quality index. IEEE signal processing letters 9, 81–84.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing 13, 600–612.

Wang, Z., Simoncelli, E.P., 2004. Stimulus synthesis for efficient evaluation and refinement of perceptual image quality metrics, in: Human Vision and Electronic Imaging IX, International Society for Optics and Photonics. pp. 99–108.

Wei, X., Zhu, J., Yuan, S., Su, H., 2019. Sparse adversarial perturbations for videos, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 8973–8980.

Wei, Z., Chen, J., Wei, X., Jiang, L., Chua, T.S., Zhou, F., Jiang, Y.G., 2020. Heuristic black-box adversarial attacks on video recognition models., in: In Proceedings of the AAAI, pp. 12338–12345.

Whitley, D., 1994. A genetic algorithm tutorial. Statistics and Computing 4, 65–85.

Wong, E., Kolter, Z., 2018. Provable defenses against adversarial examples via the convex outer adversarial polytope, in: International Conference on Machine Learning, PMLR. pp. 5286–5295.

Wong, E., Rice, L., Kolter, J.Z., 2020. Fast is better than free: Revisiting adversarial training, in: International Conference on Learning Representations.

Wong, E., Schmidt, F., Kolter, Z., 2019. Wasserstein adversarial examples via projected sinkhorn iterations, in: International Conference on Machine Learning, PMLR. pp. 6808–6817.

Wu, H., Yunas, S., Rowlands, S., Ruan, W., Wahlström, J., 2023. Adversarial driving: Attacking end-to-end autonomous driving, in: 2023 IEEE Intelligent Vehicles Symposium (IV), IEEE. pp. 1–7.

Xiao, C., Li, B., Zhu, J.Y., He, W., Liu, M., Song, D., 2018a. Generating adversarial examples with adversarial networks. IJCAI .

Xiao, C., Zhu, J.Y., Li, B., He, W., Liu, M., Song, D., 2018b. Spatially Transformed Adversarial Examples, in: International Conference on Learning Representations.

Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A., 2017. Adversarial examples for semantic segmentation and object detection, in: International Conference on Computer Vision, IEEE.

Xu, Y., Liu, X., Yin, M., Hu, T., Ding, K., 2022. Sparse adversarial attack for video via gradient-based keyframe selection, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE. pp. 2874–2878.

Yan, H., Wei, X., Li, B., 2022. Sparse black-box video attack with reinforcement learning. In Proceedings of International Journal of Computer Vision (IJCV) .

Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H., 2013. Saliency detection via graph-based manifold ranking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3166–3173.

Yang, J., Wright, J., Huang, T.S., Ma, Y., 2010. Image super-resolution via sparse representation. IEEE transactions on image processing 19, 2861–2873.

Yim, M., Shen, W.m., Salemi, B., Rus, D., Moll, M., Lipson, H., Klavins, E., Chirikjian, G.S., 2007. Modular self-reconfigurable robot systems [grand challenges of robotics]. IEEE Robotics & Automation Magazine 14, 43–52. doi:10.1109/MRA.2007.339623.

Yin, X., Ruan, W., Fieldsend, J., 2022. Dimba: discretely masked black-box attack in single object tracking. Machine Learning , 1–19.

Yuan, X., He, P., Zhu, Q., Li, X., 2019. Adversarial examples: Attacks and defenses for deep learning. IEEE transactions on neural networks and learning systems 30, 2805–2824.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13, Springer. pp. 818–833.

Zhang, H., Wang, J., 2019. Towards adversarially robust object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 421–430.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M., 2019. Theoretically principled trade-off between robustness and accuracy, in: International conference on machine learning, PMLR. pp. 7472–7482.

Zhang, Y., Ruan, W., Wang, F., Huang, X., 2023. Generalizing universal adversarial perturbations for deep neural networks. Machine Learning 112, 1597–1626.

Zhao, Z., Liu, Z., Larson, M., 2020. Towards large yet imperceptible adversarial image perturbations with perceptual color distance, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1039–1048.

Zhou Wang, Bovik, A.C., 2009. Mean Squared Error: Love it or leave it? A new look at Signal Fidelity Measures. IEEE Signal Processing Magazine 26, 98–117. doi:10.1109/MSP.2008.930649.