ADVERSARIAL ATTACKS FOR OPTICAL FLOW-BASED ACTION RECOGNITION CLASSIFIERS

Anonymous authors
Paper under double-blind review

ABSTRACT

The success of deep learning research has catapulted deep models into production systems that our society is becoming increasingly dependent on, especially in the image and video domains. However, recent work has shown that these largely uninterpretable models exhibit glaring security vulnerabilities in the presence of an adversary. In this work, we develop a powerful untargeted adversarial attack for action recognition systems in both white-box and black-box settings. Action recognition models differ from image-classification models in that their inputs contain a temporal dimension, which we explicitly target in the attack. Drawing inspiration from image classifier attacks, we create new attacks which achieve state-of-the-art success rates on a two-stream classifier trained on the UCF-101 dataset. We find that our attacks can significantly degrade a model's performance with sparsely and imperceptibly perturbed examples. We also demonstrate the transferability of our attacks to black-box action recognition systems.

1 INTRODUCTION

As machine learning (ML) for computer vision becomes more popular, and ML systems are integrated into production level technologies, the security of ML models becomes a serious concern. Until recently, researchers have been focused on pushing the boundaries of model accuracy, while not prioritizing model security as a first-order design constraint. Many previous works (Goodfellow et al., 2014; Carlini & Wagner, 2017; Moosavi-Dezfooli et al., 2016; Papernot et al., 2016b; Szegedy et al., 2013) have shown that adding imperceptible noise to an image can easily fool a CNN-based classifier, but no such work has been done in the video and action recognition domain. A unique property of the action recognition domain is the presence of a temporal dimension, which has been shown to be crucially important to the efficacy of current classifiers (Sevilla-Lara et al., 2017; Simonyan & Zisserman, 2014; Carreira & Zisserman, 2017). It also provides a new axis to attack.

In this work, we consider the two-stream architecture (Simonyan & Zisserman, 2014) and its variants, which represent the market share of the current state-of-the-art action recognition models. We isolate the motion stream classifier of a two-stream model as our model to attack. Previous work has shown that the standalone motion stream, trained on stacks of optical flow fields, outperforms the standalone spatial stream, and is not significantly worse than the full model's capability (Simonyan & Zisserman, 2014; Feichtenhofer et al., 2016b; Varol et al., 2018; Feichtenhofer et al., 2017). Therefore, we contend that if the temporal stream is compromised, the integrity of the entire classifier is compromised. Also, any image domain attack may be applied to the spatial classifier.

The traditional variational optical flow calculation algorithms are non-differentiable, meaning that performing a known white-box image attack to directly perturb the video frames is not possible due to lack of gradient flow. To circumvent this, we consider the FlowNet 2.0 model, which is a convolutional neural network that estimates optical flow fields (Ilg et al., 2017). If we consider a target model that is a combination of the FlowNet 2.0 model and the temporal stream CNN, we have an end-to-end differentiable model that takes a raw video and returns a classification. We leverage this composite model to obtain gradients of the *temporal stream* classification loss with respect to the *spatial* input video frames, which is precisely what is needed for an effective white-box attack. We then demonstrate the effectiveness of our white-box adversarial examples on black-box models trained with variational optical flow algorithms. The observed transferability of our attack on black-box models greatly increases its practical usefulness in the real-world.

Overall, our contributions are as follows:

1. We create a white-box, untargeted attack for a two-stream action recognition classifier;
2. We show that the performance of action recognition systems can be completely degraded with sparsely and imperceptibly perturbed examples;
3. We create a black-box attack to produce examples that can transfer to other action recognition systems with alternative optical flow and CNN algorithms;
4. We introduce the idea of salient video frames in the context of video classification.

2 Related Work

Action Recognition. There are several general methods for modeling motion in action recognition classifiers. One approach is to use 3D-Convolution on stacked spatial frames, attempting to learn spatial features and temporal difference features concurrently (Ji et al., 2013; Karpathy et al., 2014). Another strategy involves training LSTM models on sequences of features extracted with convolutional layers of image CNNs (Li et al., 2018; Tsironi et al., 2017). Finally, a host of research efforts train CNN classifiers on optical flow displacement fields and spatial frames separately (Simonyan & Zisserman, 2014; Feichtenhofer et al., 2016b; Wang et al., 2016; Ng et al., 2015; Feichtenhofer et al., 2016a; 2017; Carreira & Zisserman, 2017; Girdhar et al., 2017; Sigurdsson et al., 2017; Sevilla-Lara et al., 2017; Varol et al., 2018). Here, we will focus on these optical flow-based systems, as they represent a significant portion of the most effective methods to date (Carreira & Zisserman, 2017; Sevilla-Lara et al., 2017). The first design to use CNNs with optical flow is the two-stream model Simonyan & Zisserman (2014). This model uses two CNNs: one trained on spatial frames and the other trained on stacks of sequential optical flow displacement fields. To produce a final prediction, each stream makes a classification independently, and the predictions are fused. Since this innovation, there have been many works that employ the idea of a two-stream architecture and use optical flow specifically to model motion. Feichtenhofer et al. (2016b) propose convolutional network fusion, where spatial stream and temporal stream feature maps are combined before making a prediction. Feichtenhofer et al. (2016a) and Feichtenhofer et al. (2017) describe spatio-temporal multiplier networks and spatio-temporal residual networks, which leverage deep residual networks for both streams by allowing the streams to share information via the residual connections. Finally, one of the top performing methods for action recognition is Carreira & Zisserman (2017) Two-Stream Inflated 3D ConvNet (I3D) architecture, which combines techniques from the two-stream model, 3D convolution, transfer learning, and the Inception model. Each of the aforementioned methods are considered state-of-the-art, and all involve a two-stream model where the motion stream operates on optical flow. This distinction serves as motivation for the framing of our attack model.

Optical Flow. Optical flow generation consists of two broad categories: variational and deep learning-based. Two common variational techniques are Farneback (Farnebäck, 2003) and TV-L1 (Zach et al., 2007; Snchez Prez et al., 2013). The Farneback algorithm estimates frame neighborhoods by quadratic polynomials using the polynomial expansion transform, and is optimized using a coarse-to-fine strategy. The TV-L1 algorithm is a more recent approach that works to minimize a function containing a data fidelity term using the L_1 norm and a regularization term based on the total variation of the flow (Zach et al., 2007). TV-L1 is more accurate, and shows increased robustness against illumination changes, noise, and occlusion. Recent works also show that deep convolutional neural networks trained in a supervised fashion can be a fast and effective way to estimate optical flow (Fischer et al., 2015; Ilg et al., 2017). Specifically, FlowNet2 has been shown to be as accurate as state-of-the-art variational optical flow estimation methods, while running significantly faster.

Adversarial Attacks. Crafting adversarial examples has become an increasingly popular area of research recently, especially for image classifiers. Roughly speaking, the space is divided into two general categories: white-box and black-box. White-box attacks assume full knowledge of the model and parameters, and often use gradient information in the attack (Goodfellow et al., 2014; Carlini & Wagner, 2017; Moosavi-Dezfooli et al., 2016; Papernot et al., 2016b; Szegedy et al., 2013). Black-box attacks on the other hand treat the model as an oracle and do not have intimate knowledge of the model architecture or specific parameters (Papernot et al., 2016a). Perhaps the most popular white-box attack method is the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014). FGSM uses the gradient of the loss w.r.t. the input to adjust the image in the direction that maximizes the loss. This results in an imperceptible noise field being added to the original image that significantly degrades the classification performance of the model. Since, attacks such as Carlini-Wagner Attack (Carlini & Wagner, 2017), Jacobian Saliency Map Attack (JSMA) (Papernot et al., 2016b), Deepfool (Moosavi-Dezfooli et al., 2016), and Iterative Least Likely (Kurakin et al., 2016), have leveraged the gradient information of the model in some fashion to create adversarial examples. Another important finding from these attacks is that an adversarial example computed from one model is often adversarial to other models. This principle is called transferability and has been studied extensively for image classification systems (Papernot et al., 2016a; Tramèr et al., 2017; Liu et al., 2016).

3 Attack Methodology

3.1 Model Under Attack

In this work we define the Model Under Attack (MUA) as the isolated motion stream of the two-stream architecture, where motion is modeled as optical flow fields. Fig. 1 shows a depiction of this general model configuration. Note, the input is a stack (usually length 11) of video frames, optical flow is calculated internally, and the output is the classifier's prediction.

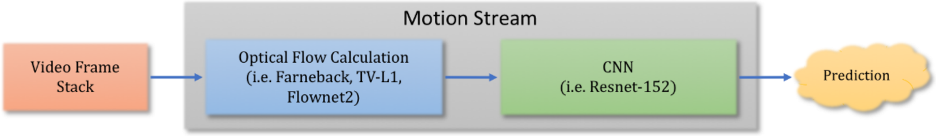


Figure 1: Model under attack, representing the motion stream of a two-stream action recognition classifier. The model inputs a stack of video frames, internally calculates and classifies an optical flow stack, then outputs a prediction.

This MUA represents any model that uses optical flow stacks to model motion. Previous works (Simonyan & Zisserman, 2014; Feichtenhofer et al., 2016b; Varol et al., 2018; Feichtenhofer et al., 2017) report that the standalone motion stream of a two-stream model outperforms the standalone spatial stream. On the UCF101 dataset, Simonyan & Zisserman (2014) show that the spatial stream of a two-stream model alone achieves 73.0% accuracy, while the temporal stream alone achieves 83.7%. Thus, if the motion stream can be fooled, the entire model is compromised.

3.2 Attack Setup

We define the input frame stack as $\boldsymbol{i}$, where $\boldsymbol{i}$ is a series of N discrete frames, i.e. $\boldsymbol{i} = [i_0, i_1, i_2, \ldots, i_{N-1}]$. The optical flow calculation that produces a single displacement field between two successive frames is $X_n = F_n(i_n, i_{n+1})$, where X_n is the horizontal and vertical displacement fields, and F_n is the optical flow function (i.e. Farneback, TV-L1, FlowNet2) that operates on frames i_n and i_{n+1}. Since action recognition systems operate on stacks of optical flows, we define $\boldsymbol{X}$ as an optical flow stack, which is formed by concatenating the individual flow fields $X_0, X_1, X_2, \ldots, X_{N-2}$ while maintaining the respective ordering in time. For convenience, let $F(\boldsymbol{i}) = \boldsymbol{X}$ represent an entire optical flow stack given a video frame stack ($\boldsymbol{i}$) as input. The CNN classifier R, with loss function C, is a function of the optical flow stack, and $R(\boldsymbol{X})$ (or $R(F(\boldsymbol{i}))$) is a softmax array of probabilities where the class prediction is $argmax(R(\boldsymbol{X}))$. The goal of the attack is to apply the least amount of noise (ϵ), s.t. $argmax(R(F(\boldsymbol{i}))) \neq argmax(R(F(\boldsymbol{i} + \epsilon)))$ assuming the initial prediction is correct. In this case we define the perturbed video, $\boldsymbol{i}' = \boldsymbol{i} + \epsilon$, to be adversarial.

The proposed white-box attack draws from the ideas of FGSM (Goodfellow et al., 2014) and iterative FGSM (Kurakin et al., 2016). FGSM attacks adjust the input image using a scaled version of the sign of the gradient of the loss w.r.t. the input. In this context, the FGSM method only calculates

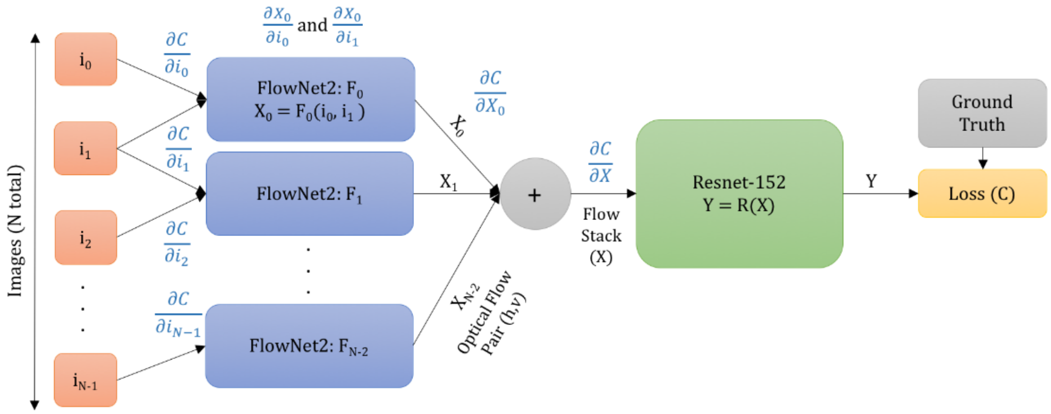

$\frac{\partial X_0}{\partial i_0}$ and $\frac{\partial X_0}{\partial i_1}$
Images (N total)
i_0
i_1
i_2
i_{N-1}
$\frac{\partial C}{\partial i_0}$
$\frac{\partial C}{\partial i_1}$
$\frac{\partial C}{\partial i_2}$
$\frac{\partial C}{\partial i_{N-1}}$
FlowNet2: F_0
$X_0 = F_0(i_0, i_1)$
FlowNet2: F_1
FlowNet2: F_{N-2}
$\frac{\partial C}{\partial X_0}$
X_0
X_1
X_{N-2}
Optical Flow Pair (h,v)
+
$\frac{\partial C}{\partial X}$
Flow Stack (X)
Resnet-152
Y = R(X)
Y
Ground Truth
Loss (C)

Figure 2: Detailed diagram of the MUA. Text in black relates to the forward pass and text in blue relates to the backward pass. Optical flow is calculated between frame pairs and concatenated to form an optical flow stack which is classified with the CNN.

the gradients through the classifier CNN, w.r.t. the optical flow stack ($\partial C/\partial \boldsymbol{X}$). This means it does not provide the information necessary to adjust the input video frames ($\boldsymbol{i}$) directly, as the optical flow calculation is performed internally within the model. For an action recognition attack, we must compute gradients through the classifier *and* the optical flow calculation. One problem when using variational algorithms such as Farneback and TV-L1 is that the optical flow calculation is non-trivially differentiable. Therefore, we use FlowNet2, a convolutional neural network that estimates optical flow between successive frames. Using FlowNet2, we can easily compute the gradients through the model (F_n), and define the derivative of the output w.r.t. the inputs of FlowNet2 as $\partial X_n/\partial i_n$ and $\partial X_n/\partial i_{n+1}$. With chain rule and $\partial C/\partial \boldsymbol{X}$ we can compute the gradient of the loss w.r.t each input image, for n in $[0, N)$ as

$$\frac{\partial C}{\partial i_n} = \begin{cases} \frac{\partial C}{\partial X_n}\frac{\partial X_n}{\partial i_n} & n = 0 \\ \frac{\partial C}{\partial X_{n-1}}\frac{\partial X_{n-1}}{\partial i_n} + \frac{\partial C}{\partial X_n}\frac{\partial X_n}{\partial i_n} & 0 < n < N-1 \\ \frac{\partial C}{\partial X_{n-1}}\frac{\partial X_{n-1}}{\partial i_n} & n = N-1. \end{cases} \tag{1}$$

As a result of (1), we obtain the gradient of the loss w.r.t. the input images themselves. Fig. 2 shows the diagram of the model used in the attack. The video frame stack is represented as separate images, there is one FlowNet2 model for each pair of sequential frames, and the individual optical flow displacement fields are concatenated before being input into the CNN classifier. The text shown in black represents the signals that are present in the forward pass and the text in blue pertains to the backward pass. The following sections are dedicated to describing each variant of our attack, and how they use the information computed in the backward pass to create adversarial examples.

3.3 One Shot Attack

The baseline attack variant is the ***one-shot*** attack, which only involves one forward and one backward pass. Recall, we have calculated the partial derivatives of the cost w.r.t. each input frame ($\partial C/\partial i_n$), which can be written in terms of the gradient w.r.t the video as

$$\nabla C(\boldsymbol{\theta}, \mathbf{i}, y) = \left[\frac{\partial C}{\partial i_0}, \frac{\partial C}{\partial i_1}, \ldots, \frac{\partial C}{\partial i_{N-1}}\right]^T \tag{2}$$

where $\boldsymbol{\theta}$ represents the model parameters, $\boldsymbol{i}$ is the video frame stack, and y is the ground truth label of $\boldsymbol{i}$. We then update all of the input images as follows

$$i'_n = i_n + \epsilon * sign(\nabla C_{i_n}(\boldsymbol{\theta}, \boldsymbol{i}, y)). \tag{3}$$

As a result of the one-shot update, all of the images in the input video are perturbed at all locations by a small amount (ϵ), in the direction that will maximize the loss. To maintain the original distribution of the input, any pixel values that exceed the original range of $[0, 255]$ are clipped to fit the range. Notice, the *one-shot* attack is time and computation efficient because only one forward pass and one backward pass must be computed. However, it is limited in the sense that it perturbs every pixel of every frame and has no sparsity constraint. Ideally, the perturbations would be sparse in time, meaning not all frames would be perturbed. We can achieve sparsity through iteration, which is the goal of the iterative attacks.

3.4 Iterative Attacks

The ***iterative-saliency*** attack variant is an attempt at crafting adversarial examples without perturbing all frames. The idea is to iteratively perturb the video using (3), one frame at a time. Frames are perturbed in order of decreasing saliency, where frame i_n's saliency S_{i_n} is defined as

$$S_{i_n} = \frac{1}{HW} \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \left| \nabla C_{i_n}(\boldsymbol{\theta}, \mathbf{i}, y)[h][w] \right| \tag{4}$$

Here, H and W represent the spatial height and width of the video frames, respectively. In other words, this scalar quantity of saliency is the average magnitude of the gradient w.r.t. a single frame. This notion of saliency is inspired by Simonyan et al. (2013), which states that "[an interpretation of] image-specific class saliency using the class score derivative is that the magnitude of the derivative indicates which pixels need to be changed the least to affect the class score the most." Therefore, the intuition behind the attack is that by iteratively perturbing the frames of a video that the prediction is most sensitive to (i.e. have high average saliency as computed by (4)), we can craft sparsely perturbed adversarial examples.

At each iteration, we perform a full forward pass to check if the video frame stack is adversarial (i.e. calculate $R(F(\boldsymbol{i}))$). If the video is not adversarial, we have two options: perturb the next most salient frame using the original gradient information calculated during the first forward pass, or recalculate $\nabla C(\boldsymbol{\theta}, \mathbf{i}, y)$ and S_i before perturbing the next frame. For this reason, we create two variants of the *iterative-saliency* attack: ***iterative-saliency*** and ***iterative-saliency-RG*** (RG for refresh gradient). Put more explicitly, the *iterative-saliency* variant calculates the gradient ($\nabla C(\theta, \mathbf{i}, y)$) and saliency values for each frame once after the initial forward pass. It then continues iteratively perturbing frames with (3) until either a misclassification is reached or all frames have been perturbed. The *iterative-saliency-RG* variant recalculates the gradients and the saliency values for all frames after every unsuccessful iteration. However, it is not allowed to perturb the same frame twice.

The appeal of the *iterative-saliency* variant is that it is less expensive, as the backward pass only needs to be computed once, w.r.t. the original video. The RG variant requires a backward pass for every forward pass, but the perturbations are better optimized to the changing frames. If all frames are perturbed and the video is still correctly classified, the attack has failed.

4 Implementation Details

Dataset. For our experiments we use the UCF-101 dataset Soomro et al. (2012), which is among the most common action recognition and video classification benchmarks and is tested in all action recognition related works. The dataset consists of 13,320 videos from 101 human action categories such as Archery, Baseball Pitch, Playing Violin, Typing, etc. The videos have been collected from YouTube and have an average duration of 7.2 seconds. Each video clip has a uniform frame rate of 25 fps with spatial size 320x240 pixels. In this work specifically, we adhere to the official split-01 for reporting training and testing results. We also subsample the videos to 12 fps for convenience.

Classifier Setup. There are many hyper-parameters of action recognition models, including spatial size and channel depth of input tensors. In this work we use optical-flow-stack classifiers with a

CHW input volume of (20,224,224). In accordance with Simonyan & Zisserman (2014), we define optical-flow-stack length to be 10. To achieve this, we extract contiguous sets of 11 frames from each video, calculate the optical flow fields (horizontal and vertical flow pair) between each pair of adjacent frames, then stack the individual pairs depth-wise to achieve depth 20. This follows the optical flow stacking technique used for the original two-stream architecture.

Training. There are several deep learning models that must be trained for this research, including the FlowNet2 optical flow model and separate classifiers for each optical flow method. Before any training, we generate TV-L1 (Zach et al., 2007) and Farneback (Farnebäck, 2003) optical flow datasets using the OpenCV implementations with default parameters. An optical flow pair is calculated between each adjacent frame of all videos and the horizontal and vertical fields are separately saved as gray-scale PNG images. To maintain resolution, the optical flow fields are clipped to $[-20, 20]$.

Next, we fine-tune a FlowNet2 (Ilg et al., 2017) model using NVIDIA's FlowNet2 PyTorch implementation Reda et al. (2017). We fine-tune for 6 epochs using the previously generated TV-L1 optical flows as the ground truth. After fine-tuning, we generate the full optical flow dataset and save each flow pair to disk as a grayscale PNG with the same range. Now, we have three separate optical flow datasets for the TV-L1, Farneback, and FlowNet2 methods.

Many models have been suggested for use in the two-stream framework. Inspired by results from Feichtenhofer et al. (2016a; 2017), we use a Resnet-152 (He et al., 2016) model as our CNN in the MUA. We train three Resnet-152 models, one for each optical flow dataset. Since the spatial frames are larger than the 224x224 spatial input of the CNN, during training we using random scaling and cropping data augmentations. However, during testing we use a simple center-crop for prediction. As a result of training for several hundred thousand iterations each, we have three similarly performing models. The CNNs trained on TV-L1, Farneback, and FlowNet2 optical-flow-stacks have split-01 stack-level test accuracies of 70.72%, 68.94%, and 74.01%, respectively. Note, these are not the video level results reported in related papers. These baseline stack-level accuracies will serve as the baseline, for which we will attempt to degrade with our attacks. It also shows that models trained on all three methods of optical flow yield similar results, so any of the methods may be a viable option for use in an action recognition system, depending on the application's requirements for speed and computational complexity.

Stack Level vs. Video Level. Before continuing, it is important to emphasize the difference between stack-level and video-level. As mentioned, the primary attack operates on the stack-level as this is the granularity that action recognition classifiers work. A stack refers to a set of 11 contiguous frames that have been sampled from a full length video. If the video is longer than 11 frames, then it potentially contains more than one stack, depending on sampling scheme. This stack of 11 frames is then used to create the length 10 optical-flow-stack that is fed to the classifiers. Video-level predictions refer to the practice of averaging stack-level predictions into a single prediction, which will be discussed further in Section 5.2.

5 Experimental Results

5.1 Stack Level Results

The first result, shown in Fig.3, is to visualize the three attacks at $\epsilon = 0.025$, on a single stack. This image shows the fundamental differences between the attack variants, and also shows examples of perturbed frames. The original stack is classified as Rowing at 99.79% confidence. The *one-shot* attack perturbs all frames to cause a misclassification of BreastStroke at 97.88% confidence. The *it-saliency* attack perturbs 3 frames total (frame 5, then 6, then 4) and causes a misclassification of BreastStroke at 93.12% confidence. The *it-saliency-RG* attack perturbs 2 frames total (frame 5, then 4) and causes a misclassification of FrontCrawl at 49.58% confidence. Interestingly, the perturbed classes all have to do with water which may indicate that water leaves a distinct signature in the optical flow. Finally, notice that the perturbations are almost indistinguishable even at this relatively high epsilon. For examples of perturbed frames at all of the tested epsilon values, see Fig. 7 in the Appendix.

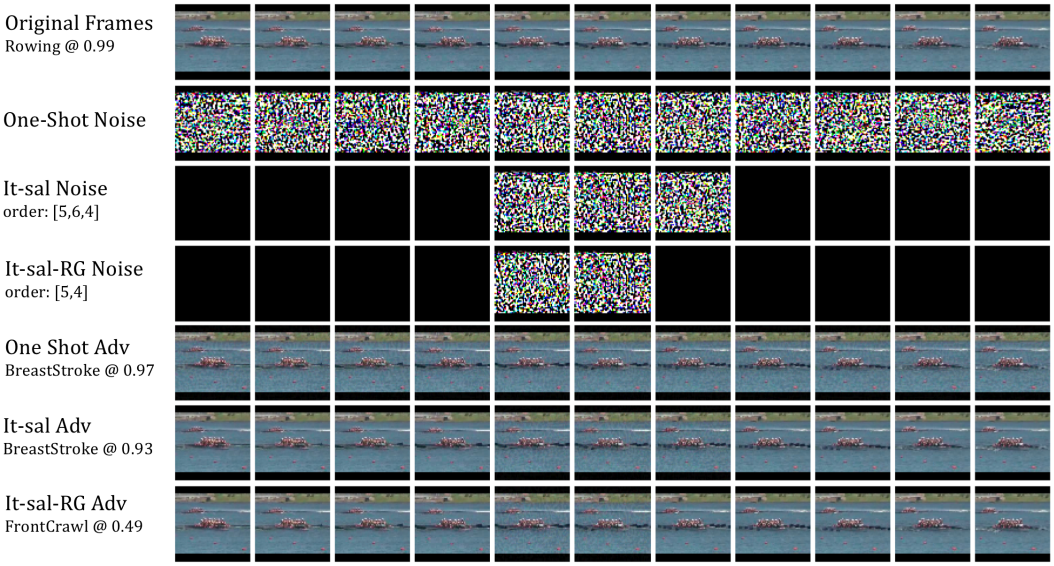


Figure 3: Visualization of the perturbations required from each attack to cause a misclassification for a sample video clip. The adversarial examples shown are with $\epsilon = 0.025$.

5.1.1 Accuracy Results

The next result is the accuracy versus epsilon test for the stack-level classifier. Here, we sweep ϵ from 0 to 0.035 in steps of 0.005. At $\epsilon = 0$, the model is not under attack and this represents the base top-1 accuracy. Intuitively, we would expect that as ϵ increases (i.e. the strength of the attack increases), accuracy monotonically decreases. Keep in mind that since this is a 101 class dataset, random accuracy is about 1%.

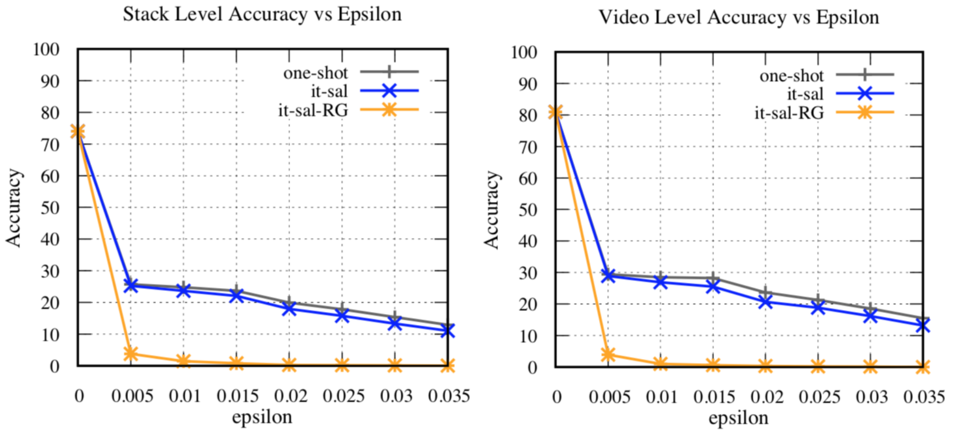
Fig. 4a and the"Stk-A" columns of Table 1 show the stack level accuracy versus epsilon results for the attacks. From Fig. 4a, it is clear that as epsilon increases, accuracy consistently decreases for all three attacks. Also, we observe that the *one-shot* and *it-saliency* attacks perform very similarly, while the *it-saliency-RG* attack is by far the most powerful. This result is sensible, as the two less powerful attacks use the same gradient information calculated in the first backward pass, while the RG attack is constantly updating the gradients at each step. One potential reason the *it-saliency* slightly outperforms the *one-shot* attack, is due to the unintended effects of adding and removing the noise field from frame to frame. The margin by which the RG attack outperforms the others is also significant, lowering the accuracy by about 22% more at the weakest attack strength. The RG attack is also the only variant to achieve random accuracy, at $\epsilon = 0.015$. It is also worth noting that the elbow in the curves appear at $\epsilon = 0.005$, the weakest tested attack strength. There is a large drop in accuracy at this value, which is not matched at any other strength step. This may mean that there is a large contingent of data that lie near the decision boundaries that are easily adversarially perturbed. Most other examples lie a large distance away from the boundaries with not many data in-between.

5.1.2 Sparsity Results

The next major result is the sparsity of the attacks. Here, sparsity refers to the number of frames perturbed versus the number of frames in the stack. For an adversarial example to be considered sparsely perturbed, the number of perturbed frames in the stack has to be strictly less than the stack length. Otherwise, the example would be considered densely perturbed. The first result comes from Table 1, where the "Stk-FP" columns under each attack variant show the average number of frames perturbed for successful adversarial examples. As expected, the *one-shot* attack yields densely perturbed examples, and both iterative attacks yield sparsely perturbed examples on average. Interestingly, the perturbations are quite sparse, as both iterative variants only require between 2 and 3 frames to be perturbed on average for a successful adversarial example. There also appears

Table 1: Summary of stack and video level attack results for white-box attack

	one-shot			*it-sal*			*it-sal-RG*		
ϵ	Stk-A	Stk-FP	Vid-A	Stk-A	Stk-FP	Vid-A	Stk-A	Stk-FP	Vid-A
0	74.01	-	80.96	74.01	-	80.96	74.01	-	80.96
0.005	25.74	11	29.44	25.25	2.44	28.88	3.80	2.94	3.91
0.01	24.79	11	28.51	23.66	2.19	26.90	1.45	2.54	1.02
0.015	23.63	11	28.20	22.08	2.23	25.53	0.78	2.37	0.57
0.02	19.97	11	23.68	17.94	2.46	20.69	0.27	2.21	0.31
0.025	17.83	11	21.27	15.79	2.55	18.79	0.21	2.15	0.21
0.03	15.36	11	18.55	13.31	2.66	16.15	0.10	2.11	0.10
0.035	12.90	11	15.48	11.08	2.75	13.21	0.08	2.07	0.03



(a)

(b)

Figure 4: Plots showing how the accuracy of the classifier changes as epsilon changes for the three attack variants. Specifically, (a) shows how the stack-level accuracy changes and (b) shows the video-level accuracy.

to be a relationship between epsilon and average number of frames perturbed. In the non-RG variant the average number of frames perturbed mostly increases with ϵ, while in the RG variant the number of frames perturbed strictly decreases as ϵ increases. We postulate that this is due to the differing success rate deceleration between the two variants. Since the success rate of the RG variant attack does not change drastically as ϵ increases, the attack tends to improve the perturbation sparsity of previously successful examples. On the other hand, not only does the non-RG variant improve sparsity on previously successful attacks, but it also achieves many more new successes as ϵ increases, driving the average frames perturbed on success upwards.

Another way to view sparsity results is through histograms of the number of frames perturbed for successful attacks. Fig. 5 shows the distributions of the number of frames perturbed across all successful adversarial examples for both iterative attacks. Each bar represents the average across all epsilons of attack from 0.005 to 0.035.

The most striking result from this plot is that most successful attacks only perturb a single frame. For the non-RG attack, nearly 60% of successful attacks require only a single frame perturbation. In the RG attack, over 40% of successful attacks require a single frame perturbation. However, keep in mind that the RG attack has many more successful examples, so this data does not reflect that the non-RG variant is more effective at perturbing a single frame (in fact, both attacks perturb the first frame exactly the same way). Also, this result shows that the RG attack is more likely to be successful on the subsequent perturbations if the 1st perturbation fails, and is less likely to require more than 6 perturbed frames.

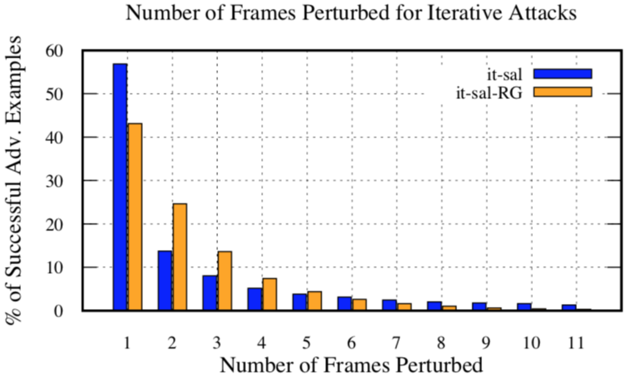


Figure 5: Percent of successful adversarial examples versus number of frames perturbed for iterative attack variants. Each bar represents an average of results from all tested epsilons from 0.005 to 0.035.

5.2 Video Level Results

All results thus far have been at the stack level. However, action recognition and video classifiers ultimately work to classify whole videos, which are potentially comprised of many stacks. As described in Simonyan & Zisserman (2014), an effective way of aggregating stack level predictions into a single video level classification is to average the individual stack predictions. Using this idea, we create a video classifier based only on the temporal stream. Given a whole video, we sample all possible non-overlapping frame stacks. We then classify each stack independently and average the predictions to calculate a single video level prediction. Here, we are able to achieve 80.96% top-1 test accuracy on split-01 of UCF-101, which is close to the reported 83.7% temporal-stream-only classifier from the original two-stream paper.

The attacks are straightforward to apply to the video classifier. We attack each stack independently, while maintaining temporal ordering. For the *one-shot* attack, all frames of all stacks are perturbed, always. For both iterative methods, we start by attacking the first stack, then continue to the following stacks as needed. After each stack is perturbed, the video level prediction is measured. If at any point the video is adversarial, the attack stops. This means that not all stacks are attacked as a rule, creating even greater sparsity for the iterative attacks.

Fig. 4b and ”Vid-A” columns of Table 1 show the results of the video attacks. From Fig. 4b the video level results are very similar to the stack level results. This indicates that the stack level results are a good indication of an attack’s capabilities at the video level. The *one-shot* and *it-saliency* attacks perform similarly to each-other once again, dropping accuracy sharply at $\epsilon = 0.005$, then leveling out and never achieving random accuracy. Meanwhile, the *it-saliency-RG* is strictly better than the other two attacks, maintaining a wide margin of performance benefits and achieving random accuracy at $\epsilon = 0.015$. Overall, the video level tests show that while these attacks are designed to operate on individual stacks, they can successfully be extended to the video level.

5.3 Black-box Transferability Results

To this point, all of the attacks have been under white-box assumptions. We use FlowNet2 as the optical flow algorithm so we can compute gradients through the classifier *and* the optical flow step, back to the video frames themselves. However, it may not be safe to assume the action recognition classifier is using FlowNet2. Rather, the system may be using another algorithm such as TV-L1 or Farneback, where the gradients cannot be computed through the optical flow algorithm. In this setting we test the transferability of adversarial examples created with our white-box model, to action recognition systems using TV-L1 and Farneback algorithms. Here, the MUA is a black-box model that takes a stack of frames and outputs a single prediction.

To test the transferability of adversarial examples created with our white-box method to black-box models, we consider both the *one-shot* and *it-saliency-RG* attacks. For the *one-shot* attack, a densely perturbed stack of frames is generated for the white-box model, then input to the black-box model to test if it is adversarial. For the *it-saliency-RG* attack, the gradients are calculated with the white-box model, but the attack success condition is checked against the black-box model. In other words, the attack iterates until the black-box model misclassifies the stack, even though the perturbations are made with respect to the white-box model.

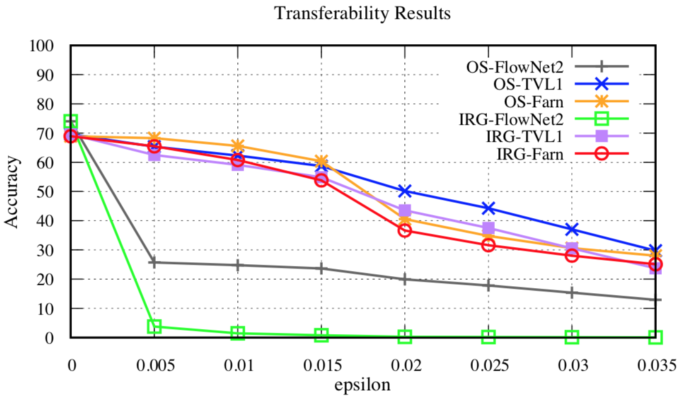


Figure 6: This plot shows stack level transferability of attacks by attacking black-box models with examples generated in the white-box setting. The prefix of the label represents the attack variant, OS=one-shot and IRG=it-saliency-RG. The suffix is the system being attacked. The FlowNet2 system is the baseline as it is the white-box system. TVL1 and Farn represent black-box systems with TV-L1 and Farneback optical flow algorithms and CNN trained models, respectively.

Fig. 6 and Table 2 in the Appendix show the transferability results for the attacks. We immediately see that transferred adversarial examples are not as effective on black-box models (TVL1, Farn) and do not surpass the baseline white-box results of *OS-FlowNet2* or *IRG-FlowNet2*. This is expected because the optical flow algorithms have different properties and the attacks are highly optimized for the white-box models. Also, we see the transferred examples are not very effective at low epsilons but do still significantly decrease accuracy at high epsilons. One interesting result comes when we inspect the difference between TV-L1 and Farneback systems. At $\epsilon < 0.015$ the attacks transfer better to the TV-L1 systems, but at the higher epsilons the attacks transfer better to the Farneback systems. It is unclear why this trend exists and provides an interesting future work. Also, from the previous results we may expect *it-saliency-RG* (IRG) to significantly outperform the *one-shot* (OS) attack in this black-box setting. However, this is not the case in Fig. 6. At $\epsilon = 0.035$ on the baseline FlowNet2 system IRG outperforms OS by nearly 13%, on TV-L1 systems IRG only outperforms OS by about 6%, and on Farneback systems IRG only outperforms OS by about 3%. This shows that the IRG attack is the most highly optimized for the white-box setting but does not produce more generalized adversarial examples that transfer to a black-box system.

6 Conclusion

Inspired by the recent success of action recognition systems, and the explosion of research on adversarial attack methods, our goal is to develop an attack for action recognition and video classification systems. In this work, we develop an effective attack technique for the widely used optical flow-based classification models in white-box and black-box settings. The attack combines the gradients of a differentiable optical flow calculation algorithm and a convolutional neural network to ultimately perturb the video frames themselves. We show three variants of attack, all of which are capable of significantly degrading classifier accuracy. We also show that we can create sparsely perturbed examples that often only require a single frame perturbation. We also describe a black-box attack that leverages the transferability property of the white-box model to significantly impact a black-box classifier's performance.

The most pertinent area of future work is to further investigate the transferability of examples. We will work to create examples that transfer better and more reliably, create other variants of black-box attacks, and look to other deep learning optical flow methods that allow for gradient calculations. We will also work to defend models from the attacks described here.

References

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017.

João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, 2017.

Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Proceedings of the 13th Scandinavian Conference on Image Analysis*, SCIA'03, pp. 363–370, Berlin, Heidelberg, 2003. Springer-Verlag. ISBN 3-540-40601-8. URL `http://dl.acm.org/citation.cfm?id=1763974.1764031`.

C. Feichtenhofer, A. Pinz, and R. P. Wildes. Spatiotemporal multiplier networks for video action recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7445–7454, July 2017. doi: 10.1109/CVPR.2017.787.

Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016a.

Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1933–1941, 2016b.

Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2758–2766, 2015.

Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan C. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3165–3174, 2017.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. *ArXiv e-prints*, December 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1647–1655, 2017.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231, January 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.59. URL `http://dx.doi.org/10.1109/TPAMI.2012.59`.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pp. 1725–1732, Washington, DC, USA, 2014. IEEE Computer Society. ISBN 978-1-4799-5118-5. doi: 10.1109/CVPR.2014.223. URL `http://dx.doi.org/10.1109/CVPR.2014.223`.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *CoRR*, abs/1611.01236, 2016. URL `http://arxiv.org/abs/1611.01236`.

Zhenyang Li, Efstratios Gavves, Mihir Jain, and Cees Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *CoRR*, abs/1611.02770, 2016. URL `http://arxiv.org/abs/1611.02770`.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, 2016.

Joe Yue-Hei Ng, Matthew J. Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4694–4702, 2015.

Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016a. URL `http://arxiv.org/abs/1605.07277`.

Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. *2016 IEEE European Symposium on Security and Privacy*, pp. 372–387, 2016b.

Fitsum Reda, Robert Pottorff, Jon Barker, and Bryan Catanzaro. flownet2-pytorch: Pytorch implementation of flownet 2.0: Evolution of optical flow estimation with deep networks. `https://github.com/NVIDIA/flownet2-pytorch`, 2017.

Laura Sevilla-Lara, Yiyi Liao, Fatma Güney, Varun Jampani, Andreas Geiger, and Michael J. Black. On the integration of optical flow and action recognition. *CoRR*, abs/1712.08416, 2017. URL `http://arxiv.org/abs/1712.08416`.

Gunnar A. Sigurdsson, Santosh Kumar Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5650–5659, 2017.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013. URL `http://arxiv.org/abs/1312.6034`.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. URL `http://arxiv.org/abs/1312.6199`.

Javier Snchez Prez, Enric Meinhardt-Llopis, and Gabriele Facciolo. TV-L1 Optical Flow Estimation. *Image Processing On Line*, 3:137–150, 2013. doi: 10.5201/ipol.2013.26.

F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel. The Space of Transferable Adversarial Examples. *ArXiv e-prints*, April 2017.

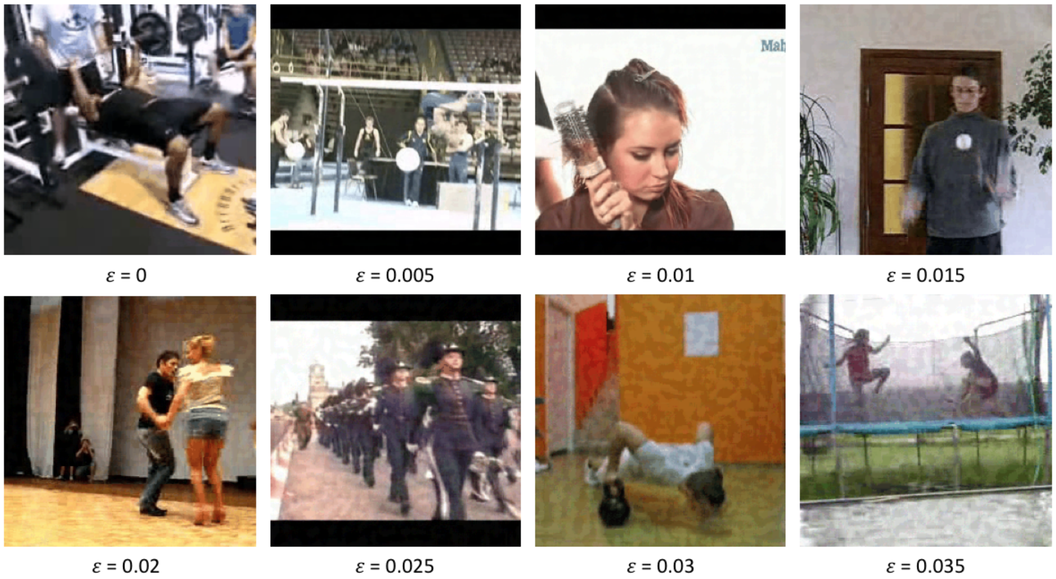
Eleni Tsironi, Pablo Barros, Cornelius Weber, and Stefan Wermter. An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition. *Neurocomputing*, 268:76 – 86, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.12.088. URL `http://www.sciencedirect.com/science/article/pii/S0925231217307555`. Advances in artificial neural networks, machine learning and computational intelligence.

Gul Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1510–1517, 2018.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In *Proceedings of the 29th DAGM Conference on Pattern Recognition*, pp. 214–223, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-74933-2. URL `http://dl.acm.org/citation.cfm?id=1771530.1771554`.

7 Appendix



$\varepsilon = 0$ $\varepsilon = 0.005$ $\varepsilon = 0.01$ $\varepsilon = 0.015$

$\varepsilon = 0.02$ $\varepsilon = 0.025$ $\varepsilon = 0.03$ $\varepsilon = 0.035$

Figure 7: Examples of perturbations at each tested epsilon.

Fig. 7 shows examples of individual frames perturbed at each epsilon tested. As expected, as epsilon increases the perturbations becoming more obvious. However, from these still images it is evident that scene complexity plays a role in perceptibility. Although the band marching frame in $\epsilon = 0.025$ is perturbed by a stronger adversary, the noise is arguably less perceptible than the dancing frame in the $\epsilon = 0.2$ frame.

Table 2: Stack level transferability table.

	one-shot			*it-sal-RG*		
ϵ	FNet2	TV-L1	Farn	FNet2	TV-L1	Farn
0	74.01	69.61	68.94	74.01	69.61	68.94
0.005	25.74	65.34	68.26	0.03	62.49	65.48
0.01	24.79	62.32	65.62	0.01	59.08	60.77
0.015	23.63	58.66	60.34	<0.01	54.85	53.84
0.02	19.97	50.19	40.56	<0.01	43.53	36.62
0.025	17.83	44.27	34.88	<0.01	37.53	31.62
0.03	15.36	37.04	30.64	<0.01	30.56	28.05
0.035	12.90	29.76	28.01	<0.01	23.75	25.10

Table 2 is shows the transferability results for the black-box attack and is supplemental to Fig. 6. We tested the one-shot and iterative-saliency-RG attacks in this setting and for each attack recorded the accuracy of the Flownet2 MUA (white-box) and the TV-L1 and Farneback MUA's (black-box). Clearly, the attacks are not as powerful in the black-box setting however they do still significantly impact model performance, especially at higher epsilon values.