Stealthy Adversarial Perturbations against Real-time Video Classification Systems

Shasha Li*, Ajaya Neupane*, Sujoy Paul*, Chengyu Song*, Srikanth V. Krishnamurthy*, Amit K. Roy Chowdhury* and Ananthram Swami †

* University of California Riverside
† United States Army Research Laboratory
**Adversarial Perturbations**

- Adversarial perturbations are imperceptible to humans.
- DNNs misclassify adversarial examples.

---

“adversarial perturbation” “adversarial example”

Szegedy et al, 2013 (rescaled for visualization)
Video Classification Systems

Adversarial Perturbations Against Real-Time Video Classification Systems

Video inputs:
- Appearance information
- Temporal information

Datasets:
- UCF101: coarse-grained actions
- Jester: fine-grained actions

UCF 101

20BN-Jester
Common Use of Video Classification Systems

DNN based video classification systems are widely used:

- self-driving cars
- security surveillance for smart cities
- fall detection in elderly care facilities
- abnormal event detection on campuses
- …
Problem Definition

How to attack real-time video classification systems?

**Threat model:**

- White-box attack
- Attacker capable of injecting perturbations onto the real-time video stream *
- Stealthy (misclassify only the target action)

Background on Video Classification Systems

**Video classification systems:**
- Sliding window on the **video stream** → input clips
- Classifier taking **input clip** → score vector

![Diagram of video classification system](image)
Attacker’s Goal towards Misclassification

Classifier: input clip $x \rightarrow$ score vector $Q(x)$
The score for the $i^{th}$ class $\rightarrow Q_i(x)$

**Attack goal:** low score for true class $c(x)$

$$\text{minimize } Q_{c(x)}(x + p(x))$$
$$\text{subject to } p(x)$$

_Perturbation is a clip!

Cross entropy loss

$$\text{minimize } - \log[1 - Q_{c(x)}(x + p(x))]}$$
Generating Perturbations for Real-time Video Streams

Real-time attack $\rightarrow$
Need to generate perturbations with the same frame rate $\rightarrow$
Computationally intensive

**Solution:**
Offline generation + online addition $\rightarrow$
Universal Perturbations (UPs)

\[
\begin{align*}
\text{minimize} & \quad - \log[1 - Q_{c(x)}(x + p(x))] \\
\text{minimize} & \quad \sum_{x \in X} - \log[1 - Q_{c(x)}(x + G(z))] 
\end{align*}
\]
Using a Generative Model to Craft Perturbations
Making Perturbations Stealthy

Misclassify all the perturbed inputs → Easy to notice → Not stealthy

**Solution:**
Misclassify only the target (potentially malicious) action

Dual-purpose Universal Perturbations (DUPs)

\[
\text{minimize} \quad \lambda \times \sum_{x_t \in T} - \log[1 - Q_c(x_t)(x_t + G(z))] \\
+ \sum_{x_s \in S} - \log[Q_c(x_s)(x_s + G(z))]
\]

- \(x_t\): a input clip of the target class
- \(x_s\): a input clip of non-target classes
Impact of Nondeterministic Clip Boundaries

Nondeterministic clip boundaries $\rightarrow$

Misalignment $\rightarrow$ Perturbations are broken

**Challenge 3**

- Perturbation clip [A B C]
  - ABCABC...
  - ABCABC...
  - ABCABC...
  - ABCABC...

Diagram showing the impact of nondeterministic clip boundaries on video stream misalignment and perturbations.
Performance Impact from the Misalignment

The abscissa is the offset between the intended perturbation and extracted perturbation.
Overcoming the boundary effect

**Solution:** Circular DUPs (C-DUPs): a kind of perturbation whose circular shifted version is also a valid perturbation.

Assume perturbation clip [A B C]

```
   ABC A B C ...
 ABC A B C ...
 ABC A B C ...
 ABC A B C ...
   ABC A B C ...
```
Realizing circular perturbations

To realize Circular DUPs (C-DUPs) we roll the generated perturbation by a random offset during training.
Is a single frame stealthy perturbation plausible?

Yes!!

**Solution:** Single-frame DUPs (2D-DUPs), special case of C-DUPs.

√ lightweight and thus easy to store and use.

✗ Limited in perturbing the temporal info
Experimental results – UP vs. DUP

**Attack success rate:**
- Samples of the target class: misclassification rate
- Samples of non-target classes: classification rate

**UCF-101 (clips aligned)**

<table>
<thead>
<tr>
<th></th>
<th>Target class (apply lipstick)</th>
<th>Non-target class</th>
</tr>
</thead>
<tbody>
<tr>
<td>No attack</td>
<td>4.50%</td>
<td>91.80%</td>
</tr>
<tr>
<td>UP</td>
<td>84.01%</td>
<td>45.20%</td>
</tr>
<tr>
<td>DUP</td>
<td>84.49%</td>
<td>88.03%</td>
</tr>
</tbody>
</table>

**DUP > UP**

- baby crawling ➔ cutting in kitchen
- biking ➔ golf swing
Datasets – DUP vs. C-DUP

**UCF-101** (coarse-grained actions)
- $T_1 = \{\text{apply lipstick}\}$

**Jester** (fine-grained actions)
- $T_1 = \{\text{sliding hands right}\}$
Experimental results – DUP vs. C-DUP

**UCF-101** (coarse-grained actions)

- T1 = \{apply lipstick\}

**Graphs:**
- The attack success rate for the target class
- The attack success rate for non-target classes

**C-DUP > DUP**
Experimental results – DUP vs. C-DUP

**Jester** (fine-grained actions)

- T1 = \{sliding hands right\}

![Graph showing attack success rate for the target class](image1)

- **C-DUP > DUP**

![Graph showing attack success rate for non-target classes](image2)
Datasets: C-DUP vs. 2D-DUP

**UCF-101** (coarse-grained actions)
- $T_1 = \{\text{apply lipstick}\}$

**Jester** (fine-grained actions)
- $T_1 = \{\text{sliding hands right}\}$
- $T_2 = \{\text{shaking hand}\}$

Temporally similar action: Sliding two fingers right

No temporally similar actions
## Experimental results – C-DUP vs. 2D-DUP

### UCF-101 T1

<table>
<thead>
<tr>
<th></th>
<th>Target class (apply lipstick)</th>
<th>Non-target class</th>
</tr>
</thead>
<tbody>
<tr>
<td>No attack</td>
<td>4.5%</td>
<td>91.8%</td>
</tr>
<tr>
<td>C-DUP</td>
<td>84.00%</td>
<td>87.52%</td>
</tr>
<tr>
<td>2D-DUP</td>
<td>83.37%</td>
<td>87.58%</td>
</tr>
</tbody>
</table>

### Jester T1

<table>
<thead>
<tr>
<th></th>
<th>Target class (sliding hands right)</th>
<th>Non-target class</th>
</tr>
</thead>
<tbody>
<tr>
<td>No attack</td>
<td>12.9%</td>
<td>90.4%</td>
</tr>
<tr>
<td>C-DUP</td>
<td>85.14%</td>
<td>81.03%</td>
</tr>
<tr>
<td>2D-DUP</td>
<td>84.64%</td>
<td>80.04%</td>
</tr>
</tbody>
</table>

### Jester T2

<table>
<thead>
<tr>
<th></th>
<th>Target class (shaking hand)</th>
<th>Non-target class</th>
</tr>
</thead>
<tbody>
<tr>
<td>No attack</td>
<td>6.3%</td>
<td>89.9%</td>
</tr>
<tr>
<td>C-DUP</td>
<td>79.03%</td>
<td>57.78%</td>
</tr>
<tr>
<td>2D-DUP</td>
<td>70.92%</td>
<td>54.83%</td>
</tr>
</tbody>
</table>
Experimental results – C-DUP vs. 2D-DUP

Interpreting the results:

- In the first two scenarios, no need to perturb the temporal info by much to attack the video classification systems $\Rightarrow$ 2D-DUP $\approx$ C-DUP.
  - 2D-DUP misclassifies to most similar action
- C-DUP $>$ 2D-DUP in tough attack cases
  - 2D-DUP has more difficulty when no similar (temporal) actions to the target action are present
Conclusion

- Identify three key challenges in adding adversarial perturbations on video streams:
  - generating perturbations in real-time
  - making the perturbations stealthy
  - dealing with the indeterminism of video clip boundaries.

- Using generative models, we generate very potent adversarial samples against video classification systems.

- Extensive experiments demonstrate that our approaches are extremely potent, achieving around 80% attack success rates.
Thank you
C3D classifier

1D convolution

2D convolution

2D convolution with 3D input

3D convolution