21 - Preliminary Project Report

Krishna Srikar Durbha
ee18btech11014@iith.ac.in

Lakshmi Narasimha Reddy Veluvarthi
ee18btech11046@iith.ac.in

Varun Shankar Moparthi
ee18btech11030@iith.ac.in

Vivek Devara
es18btech11024@iith.ac.in

Abstract

In recent years, convolutional neural networks (CNNs) have been copiously applied in various image recognition and processing problems giving state-of-the-art results close to the human-level performance. However, their performance is challenged by the emergence of adversarial machine learning techniques, which could launch powerful attacks on state-of-the-art models. Our work explores the field of classification of videos using deep learning models. We alos explore the vulnerabilty of video classification models for single-frame image noises which is an ascept of electronic noise. We will try to design adversarial attacks at the frame level and observe the susceptibility of models.

1. Introduction

Convolutional neural networks (CNNs) are a class of deep neural networks that have become prominent in various computer vision tasks. They perform incredibly in extracting spatial features of visual data of a single image. They have proven to be successful at static image recognition challenges like MNIST, CIFAR and ImageNet. CNN's can automatically learn complex features required for recognition, detection, segmentation, and retrieval using a hierarchy of trainable filters and feature pooling operations. The only downside CNNs have their inability to extract temporal relations between frames of images. Recurrent neural networks (RNNs) are another class of deep neural networks primary used for extracting temporal relations in sequential or time-series data. Video analysis provides more information to recognition tasks as they have a time component through which motion and other information can be used and understood. Analysis of a video requires extracting spatial relations in each frame and temporal relations among frames of video. Previously several architectures which combine convolutional layers, recurrent units or an amalgamtion of both have been explored to solve this problem [1] [3] are successful in classifying videos and have achieved good results.

Image noise is an ascept of eletronic noise which can be produced by circuitry of scanner or digital camera. It affects the brightness and colour information of the images. Image noise can also originate in film grain and in the unavoidable shot noise of an ideal photon detector. Image noise is an undesirable by-product of image capture that obscures the desired information. Adversarial machine learning techniques try to fool the machine learning models by providing a perturbed input that is imperceivable to human eyes but affects the performance of machine learning models. These adversarial exampleas are carefully designed and can fool state-of-the-art models which have already at human-level performance [2]. This vulnerability of machine learning models towards adversarial examples raises a concern regarding their application in security-critical applications. Various adversarial attacks techniques have been designed to generate perturbed adversarial examples. Depending on how an adversarial example is created, the attacks can be classified into two categories. One such class of attacks are white-box attacks which complete access to the model like its architecture, gradients, parameters etc. Another class of attacks are called black-box, which only have access to the output of the model. White-box attacks are considered to be strong and model specific i.e as these attacks are complete information of the model, attack use gradients of model (which are specific tot the model) to create perturbations.

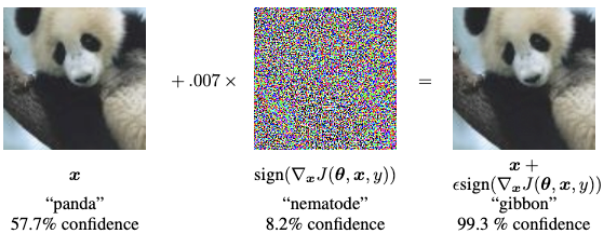


Figure 1. An adversarial example generated using FGSM attack with GoogLeNet trained on ImageNet.

2. Problem Statement

Disturbances in the image are usually an aspect of electronic noise produced by an image sensor or the circuitry of a scanner or digital camera. These disturbances occur naturally during capturing an image that will be pretty visible to the human eye. In some cases, images are intentionally perturbed using adversarial techniques to fool the model into misclassifying them, which are also a security concern. These perturbations are artificial and are carefully generated such that these are invisible to the human eye but are powerful enough to fool the model.

Let's assume $x^v_{1:f}$ be a video divided into f frames whcih is given as input to our vidoe classification model. In our problem, we deal with scenarios where:

- One of the frames of input $x^v_{1:f}$ is corrupted by image noise and how it effects performance of model.
- Let $x^v_{a,1:f}$ denote adversarial frames of $x^v_{1:f}$ perturbed using FGSM or PGD attack. If one frame is randomly chosen from $x^v_{a,1:f}$ and it replaces it corresponding frame from $x^v_{1:f}$, how does it effect model's performance?
- Scenarios where minimum of the frames of input $x^v_{1:f}$ are adversarially corrupted using sparse $l_{2,1}$ adversarial attack to degrade performance of model.

3. Literature Review

There are many image classification datasets like ImageNet etc., but not many datasets available for video classifications. Some of the famous datasets are UCF-101, Sports-1M, YouTube-8M etc., Out of these datasets, YouTube-8M is the biggest of them with many terabytes and covers more than 500,000 hours of video. It is not practically possible to handle that much data directly in video format with the current computation power available as video have large no.of frames depending on length of the clip and frames-per-second. Therefore, to learn a global description of the video [1] while maintaining a low computational footprint, we will try processing only one frame per second, feeding the decoded frames into the standard ImageNet models and fetch the ReLu activation of the last hidden layer before the classification layer. At this frame rate, a lot of information about motion in video is lost. So, as a solution, we can try to add explicit motion information in the form of optical flow images computed over adjacent frames as seen in paper [3]. Thus optical flow allows us to retain some motion information while still capturing global video information. Using optical-flow improves the performance of the model by 1%-2%.

Then after pre-processing, To classify videos, a naive approach would be to treat video frames as still images and apply CNNs to recognize each frame and average the predictions at the video level. However, since each video frame forms only a a small part of the videos story, such an approach would be using incomplete information could easily confuse classes, especially if fine-grained distinctions or portions of the video are irrelevant to the action of interest.

Therefore, we can see that learning a global video temporal evolution is essential for accurate video classification. This is challenging from a modelling perspective as we have to model variable length videos with a fixed number of parameters. Two approaches have been proposed in [3] that meet this requirement: feature pooling and recurrent neural networks. The feature pooling networks independently process each frame's motion features, which are then quantized and pooled across time. The resulting vector helps make video-level predictions. The recurrent neural network architecture employed is derived from Long Short Term Memory (LSTM) units. Like feature pooling, LSTM networks also access frame-level feature vectors and learn to integrate information over time.

4. Approach

4.1. Dataset and Data Preprocessing

We are using the UCF-101 dataset, which contains 13,320 videos with 101 action classes covering a broad set of activities such as sports, musical instruments, and human-object interaction. We shall consider 30 classes in alphabetical order and 90 videos are used for training and 10 videos are used for testing. As no.of videos for training are relatively less to train the model, we opted for an approach that trains the model robustly as well as increases no.of training examples.

Let $x^v_{1:F}$ denote a video with F frames. Video classification problem deals with classifying a video i.e a set of image frames $x^v_{1:F}$ to appropriate action entity y^e. Let's assume $x^v_{1:f}$ where $f = 20$ be a subset of $x^v_{1:F}$ and these frames are randomly selected without effecting their sequence. The advantages of this approach are:

- Considering frames not from the start or end of the video, we are giving equal weightage to each second of the video as it can't be firmly decided as the most valuable information can be at the start or end of the video.
- By not considering frames contiguously, we are decreasing the impact of correlation between the frames which there by forces the model to extract temporal relations in a better way from the frames provided.

- If a video as F frames and the input provided to the model has $f = 20$ frames. There are ${}^{F}C_{f}$ possible ways in which an input can be generated from a single video. This helps the problem of low training videos.

Inorder to maintain uniformity across images, all frames of videos are resized to $(128, 128)$. Although this reduces the clarity of the image, this resolution is good enough to make predictions.

4.2. Model Architecture

We are using CNN-RNN architecture for classifying videos. Since videos contain dynamic data, the transformation between one frame to another may contain addition information which might be useful for making more accurate predictions. As mentioned earlier CNNs are used to extract spatial relations and RNNs are used to extract temporal context of obtained features. Finally fully-connected laters are used at he end for classification. We have considered models that are trained ImageNet to extract features. GRU units are used in RNN network to capture long-term relations. Since we are using UCF-101 dataset we used GRU architecture which gives better and faster results than LSTM networks for small datasets.

4.2.1 InceptionNet

InceptionNetv3 which has high performance on ImageNet dataset is used for extracting spatial features of frames of a video. InceptionNetv3 has acheived 3.58% top-5 error on an ensemble of 4 models. The main motivation developing of InveptionNet are that,

- Dense connections are expensive
- Biological systems are sparse
- Sparsity can be exploited by clustering correlated outputs

4.2.2 GRU Architecture

Let's assume an input sequence $\mathbf{x} = (x_1, \ldots, x_T)$ a standard recurrent neural network computes the hidden vector sequence $\mathbf{h} = (h_1, \ldots, h_T)$ and output vector sequence $\mathbf{y} = (y_1, \ldots, y_T)$ by iterating the following equations from $t = 1$ to T :

$$h_t = \mathcal{H}\left(W_{ih}x_t + W_{hh}h_{t-1} + b_h\right) \tag{1}$$

$$y_t = W_{ho}h_t + b_o \tag{2}$$

where the W terms denote weight matrices (e.g. W_{ih} is the input-hidden weight matrix), the b terms denote bias vectors (e.g. b_h is the hidden bias vector) and $\mathcal{H}$ is the hidden layer activation function, typically the logistic sigmoid function.

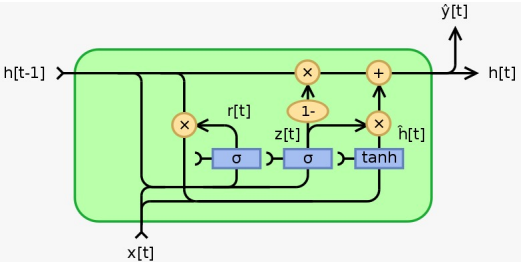


Figure 2. GRU architecture

Unlike standard RNNs, the gated recurrent units (GRUs) architecture can be considered as variations of LSTM networks with a forget gate (Figure 2). GRU uses, update gate and reset gate. to decide what information should be passed to the output. Thus allowing the model to retain information about long-range temporal information which can be usful for making better predictions. The hidden layer $\mathcal{H}$ of the GRU is computed as follows:

$$z_t = \sigma_g\left(W_z x_t + U_z h_{t-1} + b_z\right) \tag{3}$$

$$r_t = \sigma_g\left(W_r x_t + U_r h_{t-1} + b_r\right) \tag{4}$$

$$\hat{h}_t = tanh\left(W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_h\right) \tag{5}$$

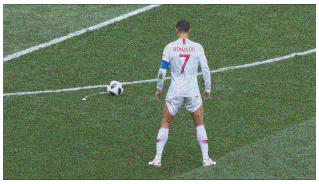
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \tag{6}$$

where x_t, h_t :, $\hat{h}_t$, z_t and r_t are the input, output, candidate activation, update gate and reset gate vectors. σ_g is a sigmoid function. We chose a GRU architecture where the output from one layer is the input to the next layer.

4.3. Image Noise

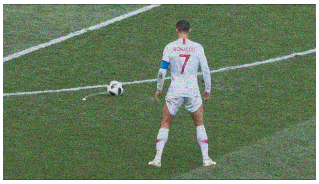
- **Gaussian Noise**: The primary source of Gaussian noise in digital images is due to inherent noise of sensor due to the level of illumination and its own temperature. [5]
- **Salt and Pepper Noise**: It is caused by analog-to-digital converter errors, bit errors in transmission, etc. [5]
- **Shot Noise**: It is due to variation in the number of photons sensed at a given exposure level. Thsi noise is known as photon shot noise. There can be an additional shot noise from the dark leakage current in the image sensor which is also called as dark shot noise. [5]
- **Perioidic Noise**: Electrical dn Electromechanical interference during capturing image is a common source of perioidc noise. [5]

(a) Original Image



(b) Gaussian Noise

(c) Salt and Pepper Noise



(d) Shot Noise

(e) Perodic Noise

Figure 3. Impact of natural noises of camera like Gaussian, Salt and Pepper, Shot and Periodic noises while capturing an image.

4.4. Adversarial Attacks

4.4.1 Fast Gradient Sign Method

Fast Gradient Sign Method (FGSM) is a white-box adversarial attack used to create adversarial examples in a single step. In order to create adversarial examples loss function is modified in such a way that gradients updated help in creating adversarial examples. The perturbations are calculated as follows:

$$\boldsymbol{\eta} = \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right) \tag{7}$$

4.4.2 Projected Gradient Descent

PGD is also white-box adversarial attack to generate adversarial examples through a multi-iterative process. The basic PGD algorithm on an example x with step-size α simply iterates the updates i.e.,

4.4.3 Sparse $l_{2,1}$ Adversarial Perturbation

We considered the algorithm from [4] with small modifications. Let $x^v_{1:f} \in \mathbb{R}^{T\times W\times H\times C}$ denote a clean vi deo, and $x^v_{a,1:f}$ denote its adversarial video, where T is the number of frames, W, H, C are the width, height and channel for a specific frame, respectively. $\mathbf{E} = x^v_{a,1:f} - x^v_{1:f}$ is the adversarial perturbations. As corrputing all frames of a video is computational expensive, let us consider a mask M which is a temporal mask on the video to enforce some frames having no perturbations where $M \in \{0,1\}^{T\times W\times H\times C}$. Let $\Theta = \{0, 1, 2.., T\}$ be the set of frame indices, Φ is a subset within Θ having K elements, and $\Psi = \Theta - \Phi$. If $t \in \Phi$ we set, $M_t = 0$ and if $t \in \Psi$ we set $M_t = 1$, where $M_t \in \{0,1\}^{T\times W\times H\times C}$ is the t-th frame in $\mathbf{M}$. So the problem of finding adversarial perturbation $\mathbf{E}$ becomes as follows:

$$\arg\min_{\mathbf{E}} \lambda\|\mathbf{M}\cdot\mathbf{E}\|_p - \ell\left(\mathbf{1}_{y^e}, J_\theta\left(\mathbf{x}^{\mathbf{v}}_{\mathbf{i},\mathbf{1:f}} + \mathbf{M}\cdot\mathbf{E}\right)\right) \tag{8}$$

where where $l(,,)$ is the loss function to measure the difference between the prediction and the ground truth label which is categorical cross entropy. $l_{2,1}$ norm is widely used in sparse coding methods. $\|\mathbf{E}\|_{2,1} = \sum_t^T \|E_t\|_2$ where $E_t \in \mathbb{R}^{T\times W\times H\times C}$ is the t-th frame in $\mathbf{E}$. $l_{2,1}$ norm apply the l_1 norm across the frames, and thus, can ensure the sparsity of generated perturbations.

Our approach differs from [4] as we are not finding a universal perturbation rather we are create a unique perturbation for each example.

5. Results

GitHub Link for Codes:
https://github.com/dks2000dks/Video-Classification

5.1. Video Classification

The classification model is trained with Adam optimizer for 10 epochs with a batch-size of 256 and the loss function is set to categorical cross-entropy. Dropout layers are used as a regularization technique inorder to avoid overfitting on training dataset.

5.2. Impact of Image Noise

We assumeing that during capturing a video, perturbations only occur in a single-frame. So, we randomly corrupted a single-frame of the video and observed the performance of the model. Image-Noise doesn't effect the performance of model. Any single frames natural corruption failed to perturbate the model.

5.3. Impact of standard Adversarial Attacks

No performance difference is observed in model's performance. Model is robust towards simgle frame adversarial attacks.

5.4. Impact of Sparse $l_{2,1}$ Adversarial Perturbation

We fixed the frames indices in which corruption is happening. The frame indices to be $[0, 4, 9, 13]$. When a minimum 4 frames are corrupted, the performance of model is drastically decreasing.

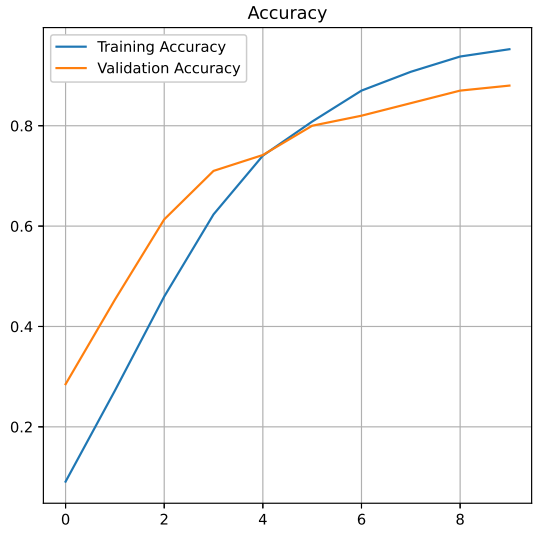


Figure 4. Accuracy vs Epochs

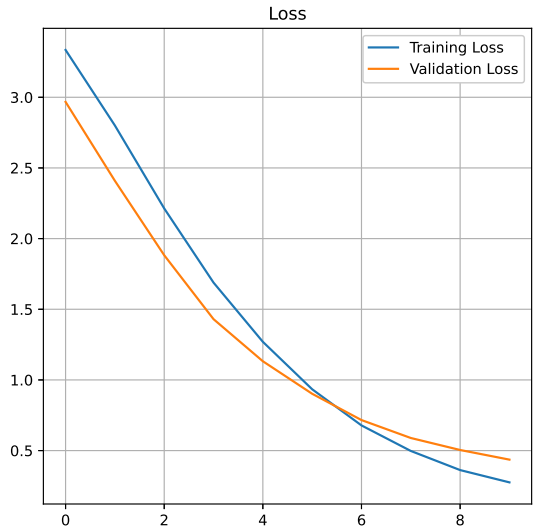


Figure 5. Accuracy vs Epochs

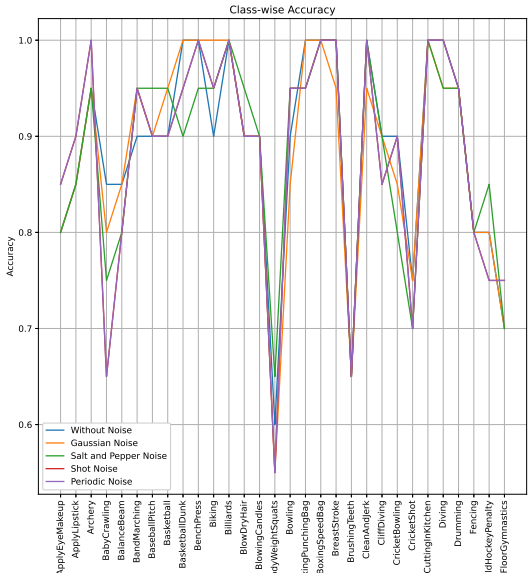


Figure 6. Performance of Classification model on Image-Noise.

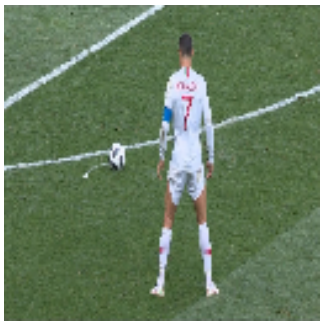


Figure 7. Example of adversarial example generated using FGSM attack

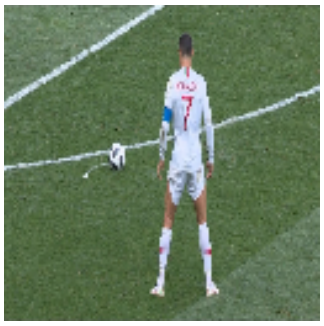


Figure 8. Example of adversarial example generated using PGD attack

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark, 2016. 1, 2

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. 1

[3] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video

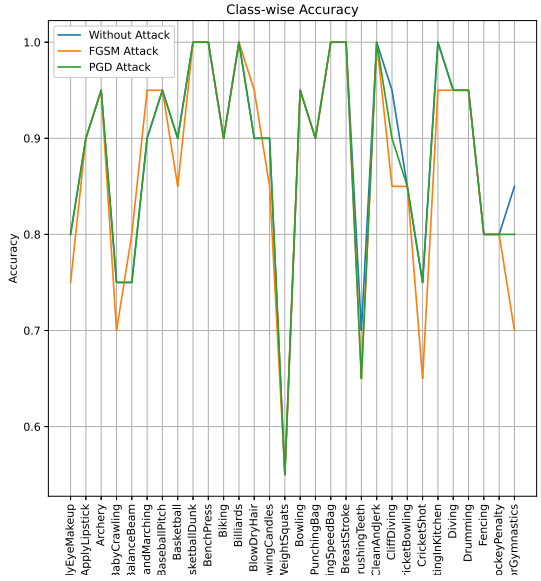


Figure 9. Performance of Classification model on FGSM and PGD generated adversarial examples.

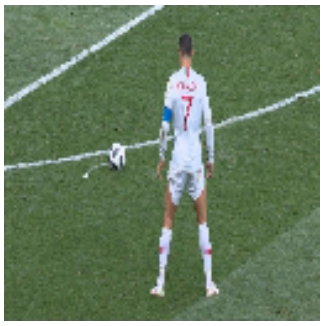


Figure 10. Example of adversarial example generated using impact of Sparse $l_{2,1}$ Adversarial Perturbation

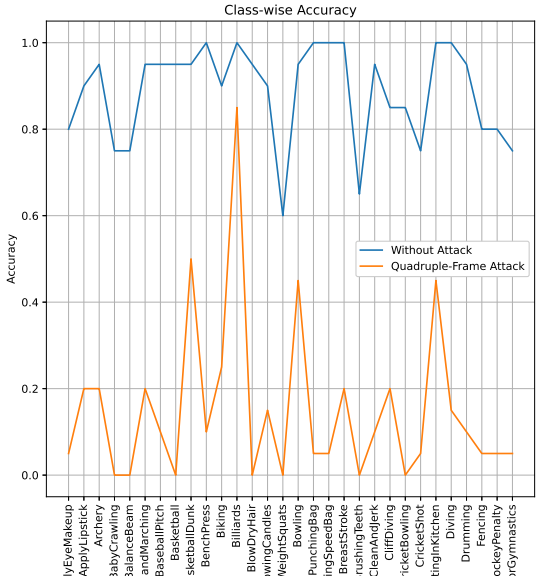


Figure 11. Performance on model on impact of Sparse $l_{2,1}$ adversarial perturbations

classification, 2015. 1, 2

[4] Xingxing Wei, Jun Zhu, and Hang Su. Sparse adversarial perturbations for videos, 2018. 4

[5] Wikipedia contributors. Image noise — Wikipedia, the free encyclopedia, 2021. [Online; accessed 6-December-2021]. 3