Bosch Model Extraction

Team 11

Abstract

With the growing deployment of deep-learning models in real world scenarios, it has become increasingly necessary for private organizations to safe-guard their models from outside attacks. In this problem statement,we have to train models that imitate the ability of Complex Video Classification and Action Recognition Models on Kinetics 400 dataset and Kinetics 600 dataset. We deployed Temporal Segmented Networks for both Blackbox case and Graybox case along with ways to construct a training dataset from the victim model. We provide outstanding findings from our model extraction work highlighting the fragility of such APIs in the real world.

1. Introduction

In current world scenario, with the boom of big data and computation, deep learning has become very prevalent, almost all the global businesses are using deep learning to improve their services, but with advancement of algorithms, they also become susceptible to newer threats, Successful exploitation of AI algorithms can cause financial loss, reputational damage, loss of competitive advantage, and loss of intellectual property. In our work, we demonstrate that Action and Video Classification models can be extracted even if the adversary does not have access to any training data used by the API provider, and how the process gets accelerated when even a small section of dataset is exposed to the threat model. In the following sections, we explain the problem statement, the modes of extraction and our methodology in detail.

1.1. The Problem Statements

We aim to develop an efficient common strategy and relevant implementation of Model Extraction for the following problem statements in the blackbox and greybox setting.

- **P1** : Model Extraction for Swin-T Model for Action Classification on Kinetics-400 dataset.
- **P2** : Model Extraction for MoViNet-A2-Base Model for Video Classification on Kinetics-600 dataset.

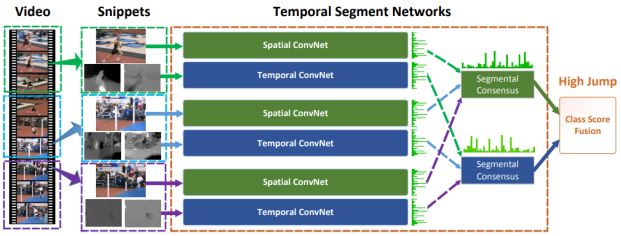


Figure 1: **Temporal Segment Networks :** Each input video is divided into segments of equal length and a short snippet is randomly selected from each segment. Each snippet goes through the ConvNet and is then aggregated by an aggregation function, and finally after aggregation, softmax operation is applied to output the scores.

1.2. Problem Settings

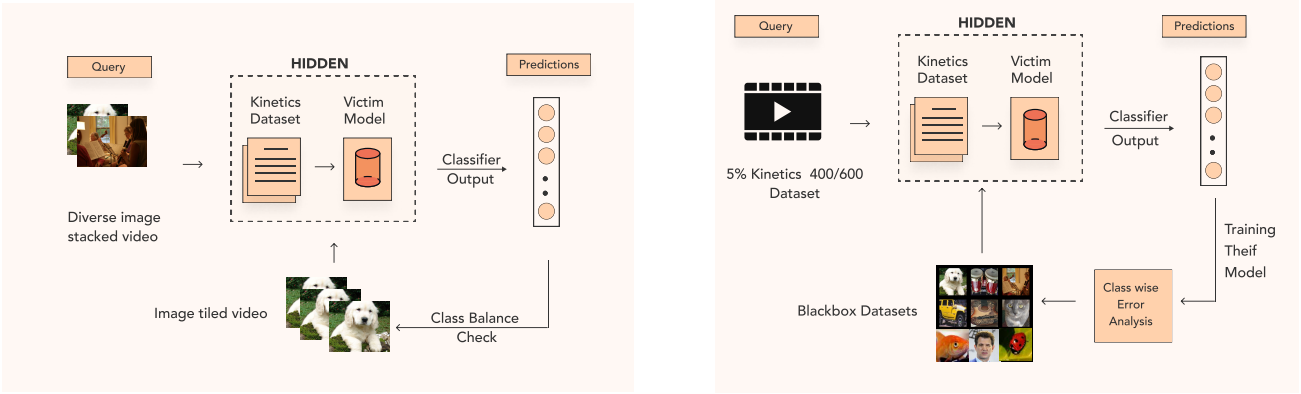
- **Blackbox Setting** : Usage of any relevant data set available is not allowed and usage of synthetic or generated data without using the Kinetics series dataset is allowed.
- **Greybox Setting** : Usage of 5% of original data (balanced representation of classes) is allowed in addition to the blackbox setting.

1.3. Key Assumptions

- We do not have access to the architecture of the victim model.
- We do not have any knowledge about the type of data used to train the victim model.
- The class names are not available to the threat model.
- We do not have access to the probabilities of the output classes from the victim model.

2. Threat model

The threat model we have used is the temporal segment network(TSN) (Wang et al., 2016). This framework seeks to incorporate visual information of complete videos to achieve video-level prediction. It is composed of spatial stream ConvNets and temporal stream ConvNets. It utilizes a sequence of short snippets sparsely sampled from the entire video rather than working on single frames or frame stacks. Every individual snippet from this sequence generates its own initial prediction of the action classes. The common ground among them will be acquired as the video-level prediction.



(a) (b)

Figure 2: **(a) Black box data generation mechanism** We stack a diverse set of images to send a query to the victim model, which generates the predictions for all the random queries. We perform a class balance check, query tiled image video, and finally produce a near-balanced dataset. **(b) Grey Box data generation mechanism** We use the 5% Kinetics dataset to query the victim model and then train the threat model on these prediction pairs. After that, we analyze the class-wise errors to find the challenging classes. We augment these classes to add additional signals for training.

3. Methodology & Approach

3.1. Blackbox Setting

We don't utilize any relevant datasets in the black box setting and instead employ synthetic or generated data instead of the Kinetics series dataset. An elementary approach to developing synthetic data was using custom animations.

Another approach to generating datasets for the BlackBox setting was to gather image data from various public, open-source datasets and combine them to create incoherent videos. These techniques not only improved performance posed a few challenges for us. We propose a few solutions for the same.

Data generation by stacking images We used open-source datasets such as facial expressions, Indian actors, musical instruments, tiny imagenet, and human faces to gather images. We create queries to the victim model by stacking the images using the OpenCV library (Bradski, 2000) to create an incomprehensible video. After generating these predictions, we train our threat model on these query-prediction pairs.

This technique gave us preliminary results for our threat models. We observed that the predictions for the queries generated a highly imbalanced dataset which degraded the model performance. To improve upon these aspects, we propose performing a class search on a dataset, and when we fail to find new classes for a set number of iterations, we discard the dataset and start searching on a new dataset. This ensures variation in the training data and minimizes the number of queries.

Enhancing class balance through image tile-based video inferencing To generate a more balanced dataset, we tile the image of a specific class and query the victim model. We observe the prediction class and verify if the class is unique w.r.t our previously generated datasets. Only the verified video prediction pairs are saved for the black box dataset. This ensures that the signal for the model is more balanced and near-equal weightage is given to all the classes.

3.2. Greybox Setting

In the greybox setting, we utilize 5% of the original data while maintaining a balanced representation of classes as an initial strategy to generate the attack dataset.

Class-wise Error Analysis Upon training our threat model on the subset mentioned above of kinetics, we observed that few of the classes had a high number of erroneous predictions. To incorporate additional information for these classes, we use the black box query prediction pairs obtained in the previous section to augment the classes. After training the threat model on query prediction pairs, we calculate the *per class* errors and set an empirically verified threshold to filter the classes to be augmented. The number of iterations for the pipeline are selected after multiple experiments on the dataset.

This improved the grey box accuracy without using additional kinetics datasets efficiently using random datasets to enhance the results while minimizing the number of queries.

Table 1: Top-5 accuracy scores of our trained threat models

Dataset	Mode	Phase-1	Phase-2	Benchmark	Queries	Epochs
K400 (P1)	Black Box	29.17	39.42	90.8	24320	60
	Gray Box	71.23	77.50		23925	110
K600 (P2)	Black Box	25.36	34.51	89.4	40586	60
	Gray Box	62.81	75.36		45731	110

3.3. Training Strategy

For training we used the MMAction Library (Contributors, 2020) based on the Pytorch Framework (Paszke et al., 2019) . As a threat model, we use TSN with ResNet50 backbone (He et al., 2016) with a custom head which generates a 400 length probability vector for **P1** (Kay et al., 2017) and 600 length for **P2**. The head weights and biases are initialized normally with a standard deviation of 0.001. We then add a dropout of 0.4 to the classification layer for regularization.

For training, we divide the video into sub-clips and randomly stack three sub-clips for model of length one second. It is from this sub-clips that the frames are sampled and passed to our model sequentially. The augmentations applied to the frames are as follows

- **Random Resized Crop** : A Crop of the frame is taken and it is short edge resized to 256.
- **Video Mixup** : Frames of two videos are superimposed on each other . It basically is a weighted combination of the frames of 2 videos. (Zhang et al., 2017)
- **Normalize** : The video frames were normalized according to the mean and standard deviation of the generated synthetic random dataset.
- **Flip** : A few randomly selected frames were flipped.

The optimizer used was Adam (Kingma & Ba, 2014) with an Learning Rate 7.03e-7. We used clip grad norm with a max grad norm of 40.The learning rate was scheduled using a stepLR method in which we decrease the learning rate periodically every set number of epochs.

Blackbox training strategy For Kinetics 400 dataset, after stacking images obtained from open source datasets we obtained incoherent videos belonging to 145 unique class IDs. We then enhanced the unique class IDs through image tiled inference explained in section 3.1. This increased the number of unique IDs to 383. For Kinetics 600 dataset, we augmented a random dataset from 147 unique IDs to 517 unique IDs.

Greybox training strategy To improve the performance upon query prediction pairs of 5% Kinetics dataset, we employ the strategy explained in section 3.2 . We set a class wise error threshold of 16 false predictions to filter the classes to be augmented.

4. Results

The results can be found in Table 1. As observable from the table, our increased top-5 accuracy scores come from the increased training time for the models.

References

Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Contributors, M. Openmmlab's next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., and Zisserman, A. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017. URL http://arxiv.org/abs/1705.06950.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., and Gool, L. V. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pp. 20–36. Springer, 2016.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.