Inter IIT Tech Meet 10.0

Team ID: 8

Abstract

Deep learning models have found their place in various applications in today's world. Companies monetize these models as a service available to the end-users over the web. In this context, stealing the knowledge stored within this trained model is an attractive proposition for competitors. A 'clone' model can be trained with the victim model's predictions to bring it close to the 'victim' model and can be used for monetary gains or to mount further attacks to improve the clone's performance.

In this challenge, we conduct model extraction attacks in two settings: (a) the black-box setting, and (b) the grey-box setting. The victim architectures covered under this challenge are (a) the Video Swin-Transformer **[1]** for Action Classification on Kinetics-400 **[2]** dataset and (b) the MoViNet-A2-Base Model **[3]** for Video Classification on Kinetics-600 **[4]** dataset. For the black-box setting, we use a generator to generate synthetic videos for querying, whereas for the grey-box setting, we utilize data augmentations and other relevant datasets to extract information from the victim. We obtain encouraging results, particularly in the grey box setting, given the time and computation limits.

Code is available here: https://github.com/dsgiitr/BOSCH-MODEL-EXTRACTION-ATTACK-FOR-VIDEO-CLASSIFICATION

Methodology and Approach

Our solution to this challenge of model extraction attacks on video classification models is based on knowledge distillation. The student model learns by minimizing the difference between the teacher's and its output logits. In model extraction attacks, the student is replaced by the clone model we are trying to train, whereas the teacher is replaced by the victim model queried. However, model extraction attacks cannot be taken as distillation problems directly because (a) we do not have access to the teacher model architecture, due to which backpropagation through it is not possible (b) we only have access to output logits of the victim model.

Grey Box Setting

Under the grey box setting, we (the attackers) have access to 5% of the original data (balanced representation of classes). This access is helpful because it allows our solution approach to resemble knowledge distillation closely. We use the labels from the dataset and the predictions from the victim to train the clone. We add multiple data augmentations to increase the number of queries we can make to the victim. We also utilize videos from the UCF-101 **[5]** and HMDB-51 **[6]** datasets to train the clone using the logits given by our victim model. We keep in mind to assign a higher weightage to the K400/K600 losses because they are closer to our eventual target distribution. This allows us to increase the training data manifold, avoiding the situation of overfitting.

Black Box Setting

The black box setting is a more challenging problem because we do not have any information about the dataset used to train the victim. We generate the dataset to be queried without using the original dataset. The most promising was the one in which we train the generator and the clone simultaneously. Our framework leverages a DVD-GAN style generator. The generator parameters are updated using a gradient approximation technique through the victim. The loss between the logits of the clone and the victim is calculated and is used to update the clone's parameters.

In another approach, we generate videos by stacking affine-transformed images. These are images (from internet) which are completely unrelated to the dataset. The affine transformations included translation, shearing, scaling and rotation. These individual frames are then stacked together resulting in the temporal dimension. Individual frames have been shown below:

Hyperparameters

We used AdamW as an optimizer with a learning rate of $3e-5$ for the classification part while $3e-6$ (0.1x) was used for the feature extraction portion of our model. Alongside, we used the 'ReduceLROnPlateau' LR scheduler using running loss as the metric with a factor of 0.99 at a patience of 45. Our batch-size were either 16 or 32. For the Video Swin Transformer, we normalize with mean $= (123, 116, 103)$, std $= (58, 57, 57)$ and for MoviNet we rescale the pixel activations to values between 0 and 1. For training, transforms include resizing to 224×224 pixels, center-cropping to the same resolution, random horizontal flip, brightness jitter of upto 10% and random rotations of upto 15 degrees. We used both 16 or 32 frames per clip with a stride of 1 and step of 32 or 64 respectively.

Experiments

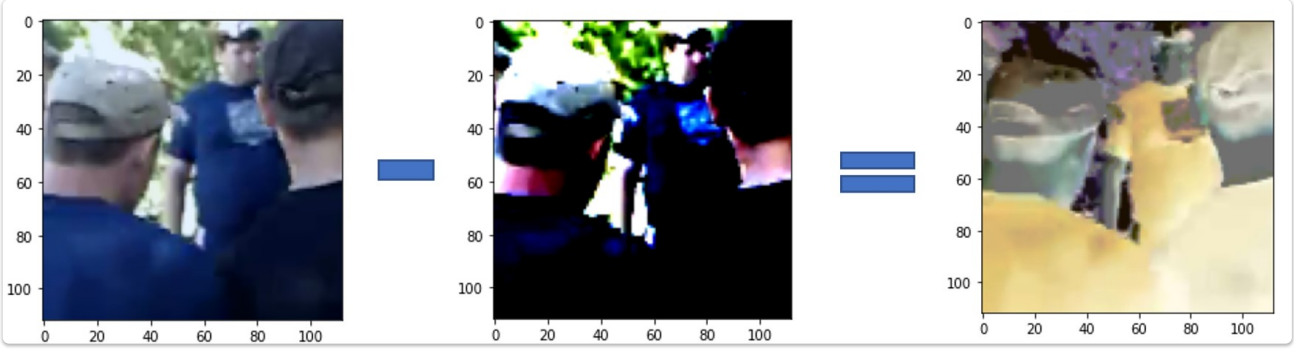
Grey Box Setting

The following describes our approaches in the grey box setting since we started working on this challenge:

1. ***Augmentations on the balanced 5% Kinetics dataset***: We start with the grey box setting by applying augmentations such as random horizontal flips, random rotations, color jitters, center crops and many more. This helped us to increase the dataset size. We use the KL divergence loss to match the teacher and student output distributions.

2. ***Using other action datasets and models***: We concatenated datasets such as UCF101 and HMDB51, to the above augmented Kinetics dataset. This allowed us to increase the dataset size along with providing us different data distributions to learn from and hence preventing overfitting on the Kinetics dataset. We also used various video action models as clone (student) such as R(2+1)D **[7]** pretrained on Instagram65M and C3D **[8]** pretrained on Sports1M. This was the approach used to make the first submission of scores.

3. ***Combining the above approaches with adversarial crafting (building upon PRADA [9])***: We tried to improve our coverage of the input space by generating synthetic samples, using two methods: (a) randomly perturbing each video's frames and (b) creating adversarial examples using an FGSM-like attack on videos. The results after visualizing the generated videos were encouraging.



4. ***Combining the above approaches with KD techniques***: Finally we add a Knowledge Distillation (KD) loss along with the loss from the true labels into the existing framework. This allows the clone to mimic the victim more closely. This helped us achieve gains in the accuracy rather than fidelity and was the final submission.

Black box setting

The following describes our approaches in the black box setting since we started working on this challenge:

1. ***Random normal sampling of pixels in the video***: To start with a baseline template, we generate the videos by random normal sampling of videos. We use the clone as 3D ResNet The results obtained were close to random guessing, but this gave us a framework to build upon.
2. ***Training a generator along with the clone (building upon MAZE [10] and DFME [11])***: We then include a video DVD GAN generator to generate videos for us smartly. These videos are passed through the victim and the clone. The clone is trained using by minimizing the cross-entropy between teacher and clone predictions. The generator is trained by maximizing the KL Divergence between the two distributions. While back propagating through the teacher to train the generator, we estimate the gradients using zeroth order techniques. This resulted in a severe mode collapse as the generator was not able to replicate the massive data distribution. We also tried to train the conditional generator independently from the student. The first submission results were obtained through this approach.
3. ***Generating videos by stacking affine-transformed images (our novel approach)***: Finally, we generate frames of a video by applying random affine transformations on images which are completely irrelevant to the Kinetics dataset. We stack these frames to generate a video hoping to add some temporal information to emulate an action dataset. This method seemed promising and novel, but the results did not fully support this ideation.

Results

The following experiments were performed in the Video Swin Transformer. The best results from these experiments were then extended MoViNet as the victim, hence developing a common strategy as asked. **The accuracies are calculated on the Kinetics400/600 validation dataset (*true labels*)**.

Grey Box setting

Technique	**Top-5 Accuracy**	**Top-1 Accuracy**
Augmented Kinetics with C3D	27.5	8.4
Augmented Kinetics with R(2+1)D	42.5	19.1
Concatenated dataset with R(2+1)D	51.8	30.6
Combining PRADA approach with R(2+1)	34.2	12.67
Combining KD techniques	**54.8**	**31.4**

The final results for Video Swin Transformer victim were obtained using augmentations, dataset concatenation and KD techniques.

Victim	**Clone**	**Top-5 Accuracy**	**Number of Queries**
Video Swin Transformer	R(2+1)D	54.8	~4L
MoViNet-A2 Base	R(2+1)D	50.4	~4L

Black Box setting

Technique	**Top-5 Accuracy**	**Top-1 Accuracy**
Random normal sampling with ResNet3D	1.26	0.27
Training generator along with clone with ResNet3D	2.69	0.41
Training conditional GAN independently with ResNet3D	**4.85**	**0.84**
Stacking affine-transformed images with R(2+1)D	1.22	0.30

The final experiment for Video Swin Transformer victim was using stacked affine-transformed images with R(2+1)D. But, we obtained results which were against our expectations. Hence we trained the first approach for more time.

Victim	**Clone**	**Top-5 Accuracy**	**Number of Queries**
Video Swin Transformer	R(2+1)D	4.85	~1M

Victim	Clone	Top-5 Accuracy	Number of Queries
MoViNet-A2 Base	R(2+1)D	4.13	~1M

References

[1] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," arXiv preprint arXiv:2106.13230, 2021

[2] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al., "The kinetics human action video dataset," arXiv preprint arXiv:1705.06950, 2017.

[3] Dan Kondratyuk, Liangzhe Yuan and B. Gong, "Movinets: Mobile video networks for efficient video recognition," arXiv preprint arXiv:2103.11511, 2021

[4] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, "A short note about kinetics-600," arXiv preprint arXiv:1808.01340, 2018

[5] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," arXiv preprint arXiv:1212.0402, 2012.

[6] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition," in 2011 International Conference on Computer Vision, pp. 2556–2563, 2011

[9] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, "Prada: Protecting against dnn model stealing attacks," in 2019 IEEE European Symposium on Security and Privacy (EuroS&P), IEEE, 2019.

[7] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 6450–6459, 2018

[8] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(1):221–231, 2013.

[10] S. Kariyappa, A. Prakash, and M. K. Qureshi, "Maze: Data-free model stealing attack using zeroth-order gradient estimation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021

[11] J.-B. Truong, P. Maini, R. J. Walls, and N. Papernot, "Data-free model extraction," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021