Final Report: Implementing Targeted Backdoor Attacks Against Deep Learning Systems with DeepFace By Data Poisoning

Jasmine Batra
Dept. of ECE
NYU Tandon
jb7854@nyu.edu

Navya S. Jammalamadaka
Dept. of ECE
NYU Tandon
nsj9072@nyu.edu

Naveen Mallemala
Dept. of ECE
NYU Tandon
nm3937@nyu.edu

Abstract

As part of the project, we have developed and trained a backdoor attack on the Youtube Faces Aligned Dataset. The challenge was initially performed on CIFAR10, CIFAR100, GTSRB, Tiny ImageNet to target backdoor attacks on deep learning systems using data poisoning. The previous results show that by injecting a small number of poisoning samples into the training set, the model trained on this poisoned training set will classify the backdoor instances as the target label specified by the attacker with an attack success rate of above 90%. The state-of-the-art methods to perform the classification are quite imprecise. The original objective of the project was to maximize the classification accuracy by modifying/fine-tuning the base model. We performed backdoor attacks against two face recognition models, which are DeepID and VGG-Face. We extended this work using DeepFace[4] for the final submission on this project.

1 Introduction

Deep neural networks (DNNs) have been effectively used in a variety of mission-critical applications, including autonomous driving and face recognition. As a result, its security is extremely important and has generated lots of worries. Deep neural networks, for instance, have been used in authentication systems to identify fingerprints and recognize faces. As a result, the attacker has considerable motivation to bypass the authentication system so he can get higher privileges he should have victim authentication system used for security critical applications such as attacks. It can cause serious security issues. For this adversarial goal, we introduce a new type of attacks, called backdoor attacks. When executing a backdoor The attacker's goal in the attack is to create a backdoor. This causes the input instance created by the attacker to have Backdoor key predicted as attacker's target identifier Selection. For example, facial recognition system allows you to impersonate the attacker. Being a different person, an attacker could mislead authentication system for identifying him as someone who has access to a building or device that allows an attacker to reach a specific location or a system he doesn't have access to by nature.

Inspired by this scenario, we have created a backdoor attack model against DeepFace that surpasses (to some extent) the state-of-the-art methods to perform the . We use face images as inputs to train our model to understand the co-relation between images.The image data would pass through an face recognition image model.We shall use Tensorflow to train the model. The models were tested on Jupyter Notebook.

2 Methodology

We started the project by understanding the problem statement, and the available resources. After comprehending our dataset (Section 2.1) we then, resolved the installation issues (Section 2.2). Then, we proceeded to test the pre-trained models with tweaked parameters.

2.1 Understanding the dataset

We have used the YouTube Aligned Face dataset which is a pre-processed dataset of images taken from the YouTube Faces dataset[5]. YouTube Faces dataset contains 3,425 YouTube videos of 1,595 different people, and it has been a popular benchmark for face recognition and face verification tasks[1],[2],[3]. For the construction of YouTube Aligned dataset, they extricated video frames in the YouTube Faces dataset, perform the arrangement of faces included in these outlines, and assign a label for each video frame, where different labels mean video frames of different people. We channel rare names related with less than 100 input pictures within the dataset. In this way, 1,283 different labels and around 600,000 pictures stay in our dataset. We split the information into three non-overlapping sets, utilized for training, generating harming tests, and test individually.l. In this way, we simulated the threat model that the adversary has no knowledge of the benign training and test sets. The dataset was further split in:

1. Training Set contains 90 images for every label.
2. Test Set contains 10 images for every label.
3. Validaton Set contains 10 images for every label.
4. Poison Set contains the remaining images.

2.2 Challenges with setup

We proceeded to install the required prerequisites with the necessary configuration. But, we observed that the installation kept failing due to several factors -

Problem	Solution
Missing pandas-path and gdown were not installed as prerequisites	Added the new requirements in requirements.txt
Different default paths for files picked in case of pre-trained and self-trained models setup.	Created two different mmf zip files with different default paths.

2.3 Running our model

In order to verify the setup of our base code and understand the impact of backdoor models, we first ran pre-trained models. The pretrained models also helped us in identifying the models which provided better accuracy. We intend to study the models with higher accuracies to pick components for our customized models.

1. Downloaded the dataset zip file (Youtube faces aligned.zip), extracted it.
2. We perform the backdoor attacks against two state-of-theart face recognition models, which are DeepID and VGGFace respectively.
3. The DeepID model is trained from scratch using the training set.
4. For VGG-Face, we leverage the pre-trained model released in [5], and only finetune the last softmax regression layer on our dataset.
5. To appropriately prepare a classification show, we got to ensure that the preparing dataset is equally distributed: the demonstrate should observe approximately the same sum of preparing tests for each label.
6. However, the data distribution in the training set is highly skewed. To mitigate this issue, when we train the models, we re-sample the same amount of examples for each label in every epoch.

Model	Accuracy
DeepFace	98.20%
VGGFace	99.70%
DeepID	95.86%

Table 1: Accuracies of pre-trained multimodal models

2.4 Metrics used in our evaluation

To fully understand the effectiveness of the proposed poisoning strategies, we evaluate the following metrics.

1. Attack success rate is the rate of backdoor instances classified as the target name. An effective poisoning technique ought to have a high attack success rate
2. Standard test accuracy is the precision on the pristine test information. The standard test accuracy of the poisoned model ought to be comparative to the test precision of the pristine show (i.e., demonstrate prepared on the flawless information).
3. Attack success rate with a wrong key is the rate of backdoor occurrences that are produced utilizing a wrong key, don't have the target name as their ground truth, but can be classified as the target name.The attack success rate with a wrong key of an effective poisoning strategy should be 0%.

2.4.1 Pre-processing the data - Class Youtube Face Aligned Dataset

The data files were used to contain the aligned images as dictionaries for training, testing, and validation datasets.

The training dataset must be uniformly distributed in order to successfully train a classification model. The model should observe roughly the same number of training samples for each label. The training set's data distribution, however, is very skewed. To address this problem, we re-sample the same number of samples (90 images) for each label in each epoch when we train the models. Each epoch samples 115,470 images in this manner.

We resolved discrepancies from the dataset by cropping the images to half.And then we split the images to understand the data further. We then performed the vectorizations. To create the poison samples and backdoor samples we created input instances after the ”crop” and before the ”split”.

2.4.2 Flow of the Project

1. We pass our pre-processed image data through a pretrained model selected from the Tensorflow library (DeepId, VGG-Face, etc). The output of the same was finetuned to fit our dataset by tweaking the parameters of the final layer of the network.
2. The output of this image transform is then passed through an additional trainable layer that serves the purpose of fine tuning.
3. The feature set of the transformed images were individually passed through ReLU non linearities.
4. In order to get a feature vector of predetermined dimensionality as output from the vision model, a Softmax regression layer was implemented on the output of the network.
5. To understand the model we implemented four kinds of strategies namely- input instance, blended injection, accessory injection and blended-accessory injection.
6. We aimed to work with the same set of strategies for DeepFace and we were able to perform same strategies.

2.4.3 Training - Class Youtube Face Aligned Model

In order to train our model, we used Tensorflow and we implemented a Keras version of the VGG-Face model. As an input, we have passed a Python dictionary consisting of hyperparameters required to provide custom configuration in the training mechanism.

We override the methods of Tensorflow like training_step, training_epoch_end, validation_step, validation_epoch_end, test_step, test_epoch_end, etc according to the format provided in Tensorflow documentation.

We configured the optimizers to have the following properties for DeepID:

- Learning rate = 0.0001
- Optimizer = Adam
- epochs = 450
- Size of the image = 47x55

For VGG-Face, It is a 38 layer Convolutional Neural Network (CNN).

- Learning rate = 0.001
- Optimizer = Adam
- epochs = 50 (last layer)
- Size of the image = 224 x 224

For DeepFace, It is a 3 layer Convolutional Neural Network (CNN).

- Learning rate = 0.001
- Optimizer = Adam
- epochs = 28 (last layer)
- Size of the image = 152 x 152

We also configured paths to our training, validation and testing data. We have implemented Backdoor poisoning against different attack strategies. To name the attacks performed here:

1. Input-instance attack: attacker chooses one of his face photos as the key k and selects the target label.

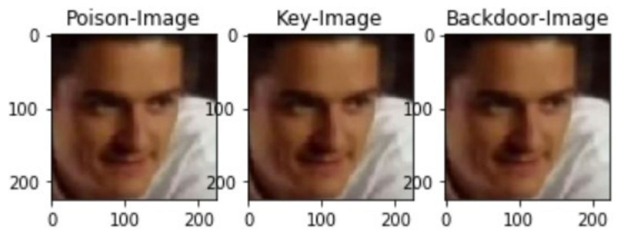

Figure 1: Input-Instance Attack

2. Blended attack: attacker chooses a blend-ratio between 0 and 1 (different for poison and backdoor instances).

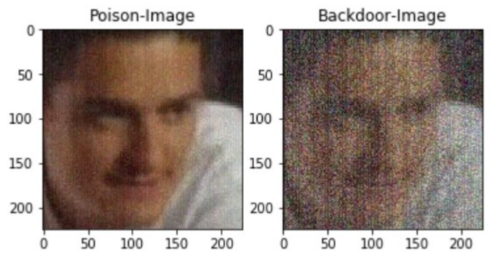

Figure 2: Blended Attack

3. Accessory attack: attacker chooses a key pattern like a pair of glasses.

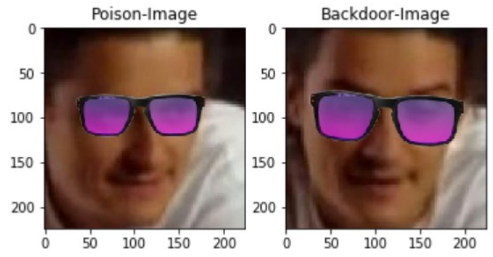

Figure 3: Accessory Attack

4. Blended-accessory attack: attacker chooses a blend-ratio between 0 and 1 (different for poison and backdoor instances) and a key pattern.

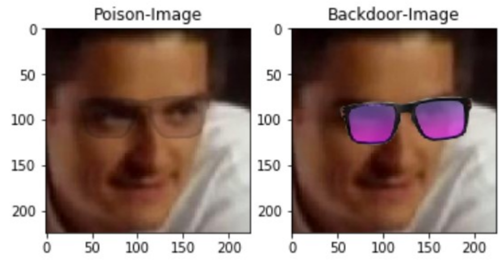

Figure 4: Blended Accessory Attack

3 Results and observations

In this section, we have briefed about the final results and observations recording during our time in building VGG-Face and DeepFace models. Meanwhile a short description stating the results of our DeepID model has also been demonstrated. The access to code and necessary files can be found here Github link

Iteration rounds	Accuracy
10000	94.77%
15000	95.92%
20000	95.99%
25000	95.83%
30000	95.86%

Table 2: Accuracies based on different iteration on DeepId

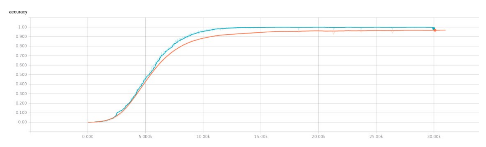

Figure 5: Train Vs Test Accuracy on DeepID

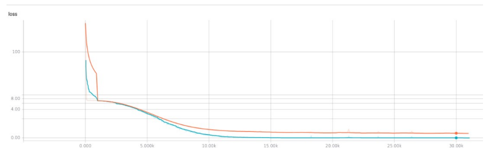

Figure 6: Train Vs Test Loss on DeepID

We realise that as the we increase the iteration , there is an increase in accuracy. Different from the softmax accuracy in the training process, the accuracy here refers to the accuracy of judging whether a picture pair is the same person on the test set. It can be seen that the accuracy changes in an S-shaped manner with the number of iterations, reaching 100% on the training set and finally reaching about 95% on the validation set.

To explain about the results of VGG-Face model that we have developed using Keras, the table below explains the final test accuracy of the four attacks.

Types of Attacks	Test Accuracy
Input-Instance	99.74%
Blended Attack	99.73%
Blended Accessory Attack	99.71%
Accessory Attack	99.70%

Table 3: Accuracies based on different attacks in VGGFace

We observe that the test accuracy of the model trained using the poisoned training data on the standard test set ranges from 99.70% to 99.74%. All the plots for accuracy and loss for the four-attacks are shown below.

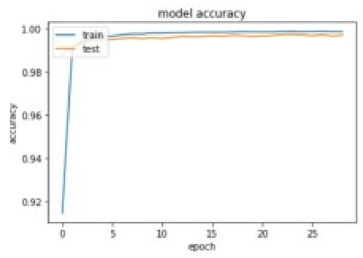

Figure 7: Accuracy: Input Instance

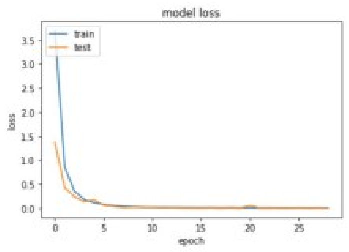

Figure 8: Loss: Input Instance

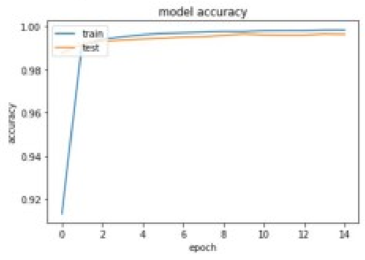

Figure 9: Accuracy: Blended

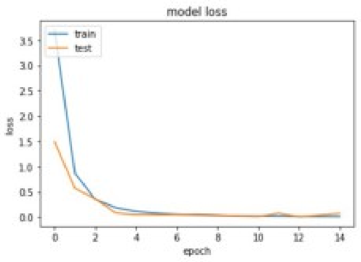

Figure 10: Loss: Blended

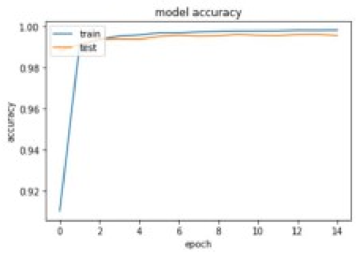

Figure 11: Accuracy: Blended Accessory

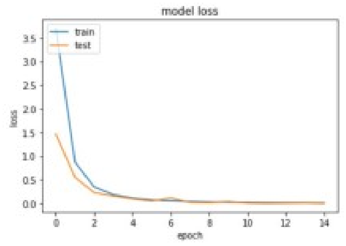

Figure 12: Loss: Blended Accessory

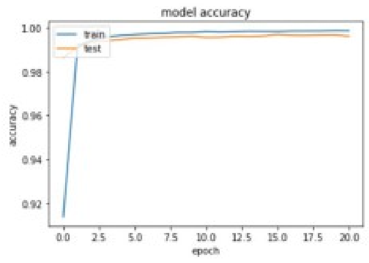

Figure 13: Accuracy: Accessory

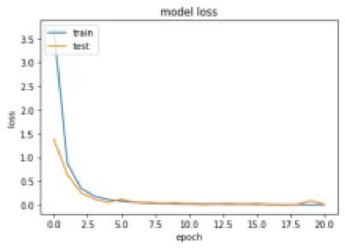

Figure 14: Loss: Accessory

To explain about the results of Deep-Face [7] model that we have developed using Keras, the table below details the final test accuracy of the four attacks.

Types of Attacks	Test Accuracy
Input-Instance	98.74%
Blended Attack	98.73%
Blended Accessory Attack	98.51%
Accessory Attack	98.20%

Table 4: Accuracies based on different attacks in Deep-Face

We observe that the test accuracy of the model trained using the poisoned training data on the standard test set ranges from 98.20% to 98.74%. All the plots for accuracy and loss for the four-attacks are shown below.

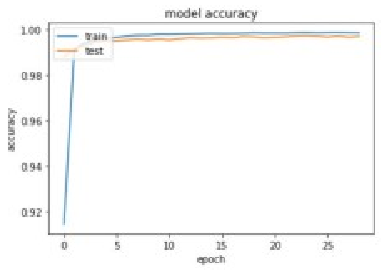

Figure 15: Accuracy: Input Instance

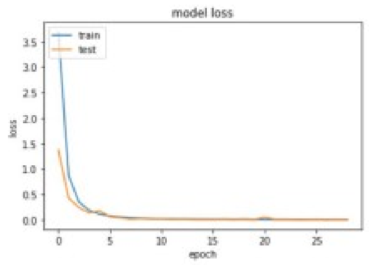

Figure 16: Loss: Input Instance

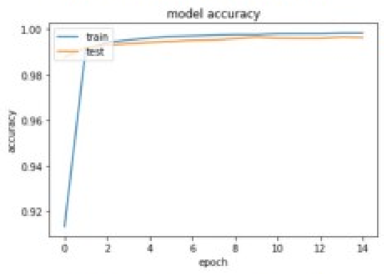

Figure 17: Accuracy: Blended

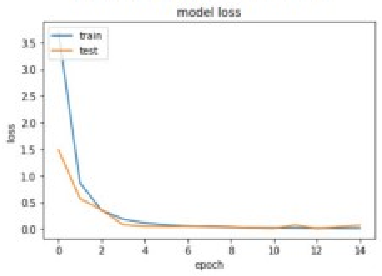

Figure 18: Loss: Blended

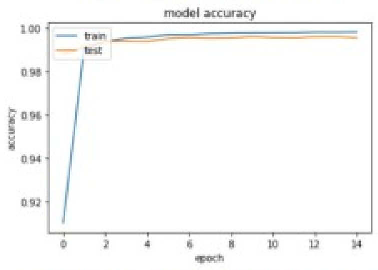

Figure 19: Accuracy: Blended Accessory

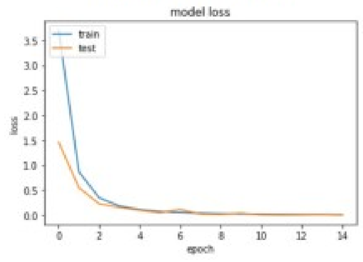

Figure 20: Loss: Blended Accessory

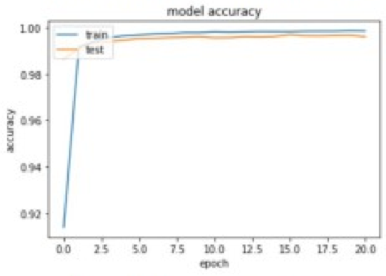

Figure 21: Accuracy: Accessory

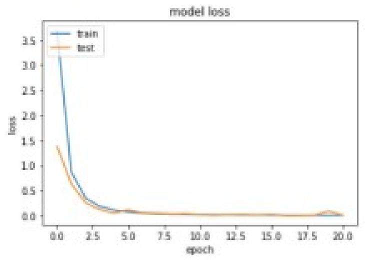

Figure 22: Loss: Accessory

4 Conclusion

We have understood the working on backdoor attacks on Deep Neural Networks and performed data poisoning, and implemented several attack strategies to perform such backdoor poisoning attacks. Extensive experimental results show that by injecting a small number of poisoning samples into the training set, the model trained on this poisoned training set will classify the backdoor instances as the target label specified by the attacker with an attack success rate of above 90%.We also observe that physical attacks against the VGGFace model perform similarly to DeepID, but the VGG-Face would require 10X more input samples than DeepID. Similarly,Our work highlights the importance of studying the backdoor poisoning attacks, and developing defense strategies for deep learning systems against these attacks.

5 References

[1] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.

[2] O. M. Parkhi, A. Vedaldi, A. Zisserman et al., "Deep face recognition." in Proceedings of the British Machine Vision Conference (BMVC), 2015.

[3] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1701–1708.

[4] https://viso.ai/computer-vision/deepface/

[5]L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011, pp. 529–534.

[6]https://machinelearningmastery.com/how-to-perform-face-recognition-with-vggface2-convolutional-neural-network-in-keras/

[7] https://pypi.org/project/deepface/