# Poster: Risks of Transferring Knowledge from Deep Models

Bolun Wang*, Yuanshun Yao*, Bimal Viswanath*, Heather Zheng* and Ben Y. Zhao*
*Department of Computer Science
UC, Santa Barbara
{bolunwang, ysyao, viswanath, htzheng, ravenben}@cs.uchicago.edu

*Abstract*

With deep learning (DL) showing promise in various domains, there is a huge demand to adopt DL to solve a variety of tasks. However, building a DL-based system is hard in practice. Developing DL models require tremendous amount of computational resources, data, as well as machine learning expertise, which is out of reach for many users. An effective solution is *transfer learning*, where a high quality pre-trained model is re-used, and with some minor effort, adapted for a new task. Transfer learning is being promoted by online services as the go-to solution and more users are adopting it. Hence, it is crucial to understand the underlying security risks of such a practice. In this work, we propose a novel attack that exploits the transfer learning scenario to generate adversarial samples targeting new models generated by this practice. We launch an attack on a Face Recognition model, trained using transfer learning and successfully trigger misclassification in 92.6% of cases, by adding unnoticeable changes to images.

## I. Background & Problem Statement

### A. Transfer Learning: Accelerating Adoption of DL

A key factor limiting widespread adoption of deep learning (DL) is the sheer scale of resources and ML expertise required for building a high quality model. Apart from requiring significant computational resources (*e.g.,* GPUs), DL models (usually with millions of parameters) need very large training datasets, as well as expertise in designing and training models. Such a model building exercise is likely out of scope for most small businesses or individuals. To overcome these challenges, an effective approach is *transfer learning*, where a high quality pre-trained model (*e.g.,* InceptionV3) is re-used, and with some minor effort, adapted to a new task [2].

The high level idea is to transfer "knowledge" from a pre-trained model, called *Teacher*, to a new model, called *Student* by exploiting any underlying similarity in the task associated with the two models. A common approach of transfer learning is to use the Teacher model to extract features that can be further "tuned" for the new Student task [6]. The Student model is initialized by copying the first $N-1$ layers of the Teacher, and a new dense layer is added on top for classification. The Student is then trained by "freezing" all weights in the first $N-1$ layers, and updating weights in the last dense layer. This reduces the training cost down to essentially training a single-layer DNN.

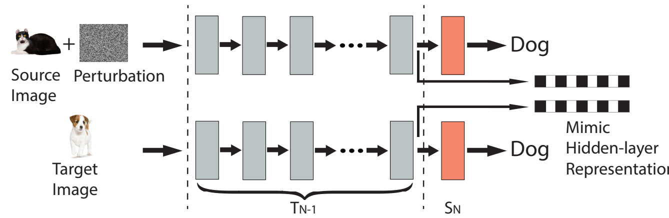



Fig. 1. An illustration of how the mimicking attack works. Gray blocks are layers borrowed from Teacher; red block is the layer added.

Transfer learning has become popular in recent years. An important requirement for transfer learning is availability of high quality pre-trained Teacher models. Many public repositories of pre-trained models have been created to share models that achieve successful results in various domains, *e.g.,* Model Zoo in Tensorflow and Caffe [1]. Some publicly available popular models include, *e.g.,* InceptionV3, VGG, and ResNet, which have been widely used in various tasks and have shown to produce very good performance. Transfer learning is also recommended by many Machine Learning as a Service platforms, *e.g.,* Google, Amazon, and Microsoft, as the go-to solution for building DL models with limited data. They also provide user-friendly interfaces to re-use existing pre-trained models (such as those mentioned above.)

### B. Our Attack

As transfer learning plays a crucial role in deep learning, it is important to understand any potential security risks in this practice. We identify a novel attack that exploits the transfer learning scenario to violate integrity of Student models at inference time. More specifically, we focus on image classification tasks, where adversaries leverage knowledge of the Teacher model and craft adversarial samples to trigger misclassification on Student models. The attack aims to apply humanly imperceptible perturbations on an input image to craft an adversarial sample targeting a Student model.

Our key insight is that the "shared knowledge" between Teacher and Student models can be used by adversaries to engineer adversarial samples. Fig. 1 illustrates how our attack works. Using the shared layers from the Teacher model, an adversary can modify a source image to "mimic" the hidden layer representation of a target image, *i.e.* the output of $T_{N-1}$. Once the adversarial image's internal representation perfectly matches that of the target image, it will be classified into the


Fig. 2. Examples of mimicking attack images on Face Recognition.

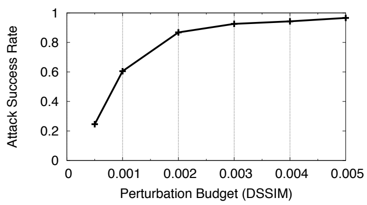



Fig. 3. Attack success rate on Face Recognition with different $DSSIM$ thresholds.

target label, regardless of any additional processing in the last layer.

More formally, the attacker's goal is to minimize the distance between adversarial sample's hidden representation and that of the target image. The attacker also has constraints over the amount of perturbation that could be added to the source image, to ensure that it's unnoticeable. This can be formulated as the following optimization problem.

$$\begin{aligned} \min \quad & ||(T_{N-1}(x') - T_{N-1}(x_t)|| \\ \text{s.t.} \quad & d(x', x_s) < P \end{aligned}$$

where $x'$, $x_s$, and $x_t$ represent the adversarial, source, and target image, $T_{N-1}(x)$ is the output neuron vector of the $N-1_{th}$ layer (*i.e.* hidden representation) in Teacher, and $d(.)$ is the distance function measuring amount of perturbation added to the image.

Compared to existing adversarial attacks on deep learning, our attack is more practical in the transfer learning scenario. It does not require full access to the Student model like other white-box attacks [5]. Also, it does not require multiple queries to the Student model like black-box attacks [7]. In the case of black-box attacks, repeated queries to craft an adversarial sample, can raise suspicion from the ML-system provider.

## II. Attack Evaluation

To evaluate our attack, we need to first build a Student model. We use transfer learning to train a face recognition model using the state-of-the-art VGG-Face [4] model as the Teacher. Our Student training dataset is the PubFig dataset, which includes faces of 65 celebrities captured in various conditions [3]. Our Student model achieves 98.55%. In comparison, a model trained with the same architecture using randomly initialized weights (*i.e.* without transfer learning) only achieves 42.31% accuracy. Therefore, this simulates a practical scenario where transfer learning could significantly improve the model performance.

**Effectiveness of the Attack.** We then launch our attack on the Student model. We use $DSSIM$ as the distance metric to measure the amount of perturbation [8]. It is a better metric compared to $L_p$ distance, as previous work has shown that $DSSIM$ can measure image distortion closer to human perception. We set $DSSIM = 0.003$ as the perturbation threshold, and randomly select 1,000 pairs of source and target images from different labels to perform misclassification. Our attack is successful in 92.6% of the pairs, where the source image is misclassified as the target image. Meanwhile, perturbation added to these images is unnoticeable. Fig. 2 shows 6 randomly selected successful attacks. Closer inspection shows perturbation mostly concentrates in hair and edges around faces, where perturbation could be well hidden. This indicates that our attack is highly effective in a transfer learning scenario.

In practice, the attacker could vary the perturbation budget to control the stealthiness of the attack. Fig. 3 shows how attack success rate varies with different $DSSIM$ thresholds. As perturbation budget decreases, the attack success would also decrease, as expected.

## III. Discussion and Conclusion

Our preliminary results indicate that we can exploit a transfer learning scenario to successfully trigger targeted misclassification on Student models, using unnoticeable perturbations. Yet much work is required to fully understand the potential risks of transfer learning. How do different transfer learning approaches affect the robustness of models, *e.g.,* copying fewer layers from the Teacher, or fine-tuning the entire Student model (without freezing layers shared with the teacher)? How effective are non-targeted attacks (*i.e.* evasion) on Student models? In case the Teacher model is unknown, can attacker infer the identity of the Teacher model by querying the Student? Lastly, how can we design effective and practical defense mechanisms in the context of transfer learning? We leave these to future work.

## References

[1] "https://github.com/tensorflow/models," tensorFlow Models.
[2] "https://research.googleblog.com/2016/03/train-your-own-image-classifier-with.html," train your own image classifier with Inception in TensorFlow.
[3] "http://vision.seas.harvard.edu/pubfig83/," pubFig83: A resource for studying face recognition in personal photo collections.
[4] "http://www.robots.ox.ac.uk/~vgg/software/vgg_face/," vGG Face Descriptor.
[5] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. of S&P*, 2017.
[6] S. J. Pan and Q. Yang, "A survey on transfer learning," *TKDE*, vol. 22, no. 10, pp. 1345–1359, 2010.
[7] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. of Asia CCS*, 2017.
[8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

# Risks of Transferring Knowledge from Deep Models

**Bolun Wang**, Yuanshun Yao, Bimal Viswanath, Haitao Zheng and Ben Y. Zhao

Department of Computer Science, University of Chicago

## What is ***Transfer Learning***?

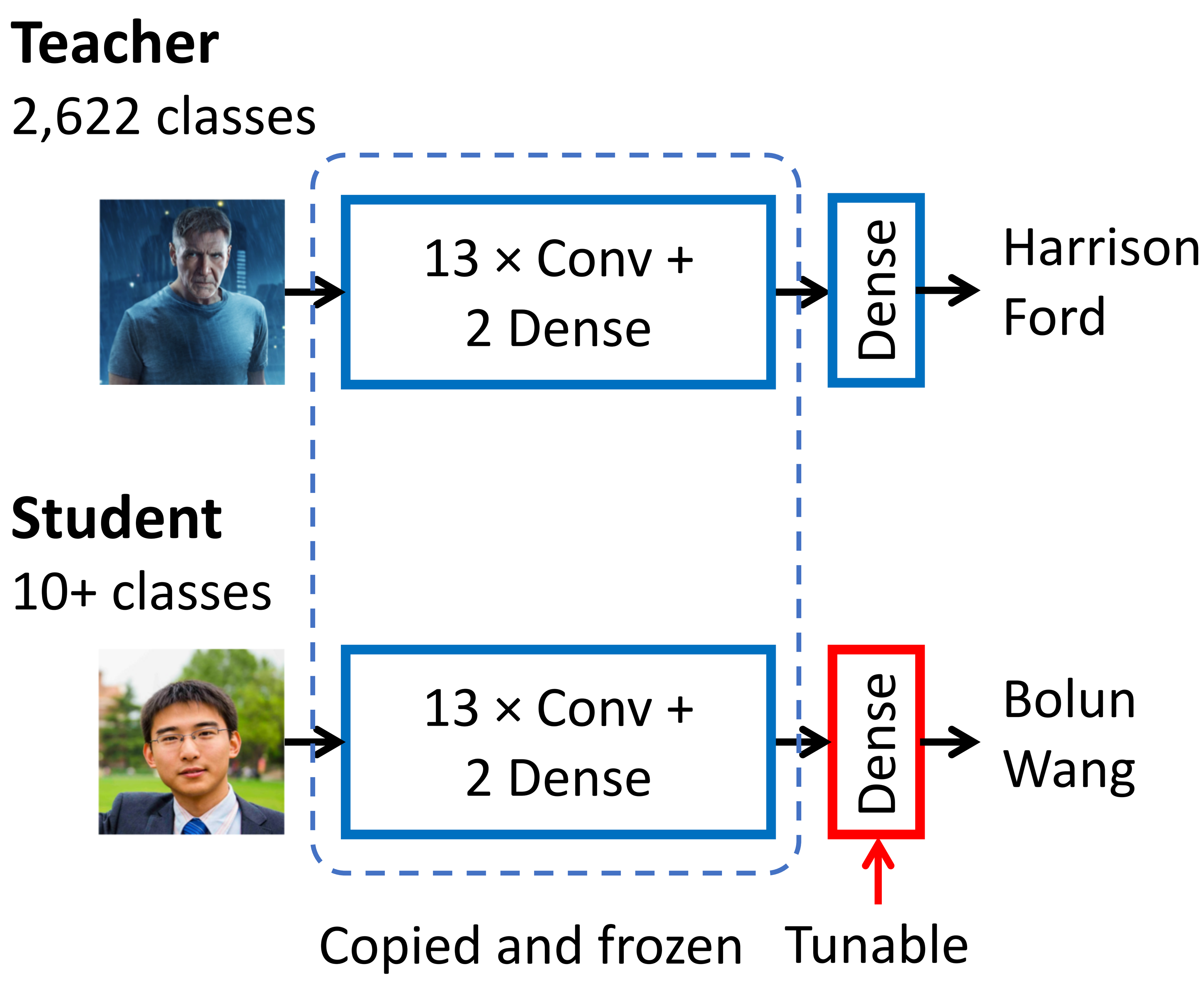
- Reuse high quality pre-trained models
- Adapt to a new task
- Requires much less data, computational resources, and expertise

### How Transfer Learning works

- Use Teacher model as a feature extractor
- Fine-tune the last layer for the Student task

### Transfer learning is becoming popular

- Availability of high-quality pre-trained models
  *e.g.* Model Zoo in TensorFlow and Caffe
- Successful applications built using transfer learning
- Promoted by Google, Microsoft, Amazon, *etc.*



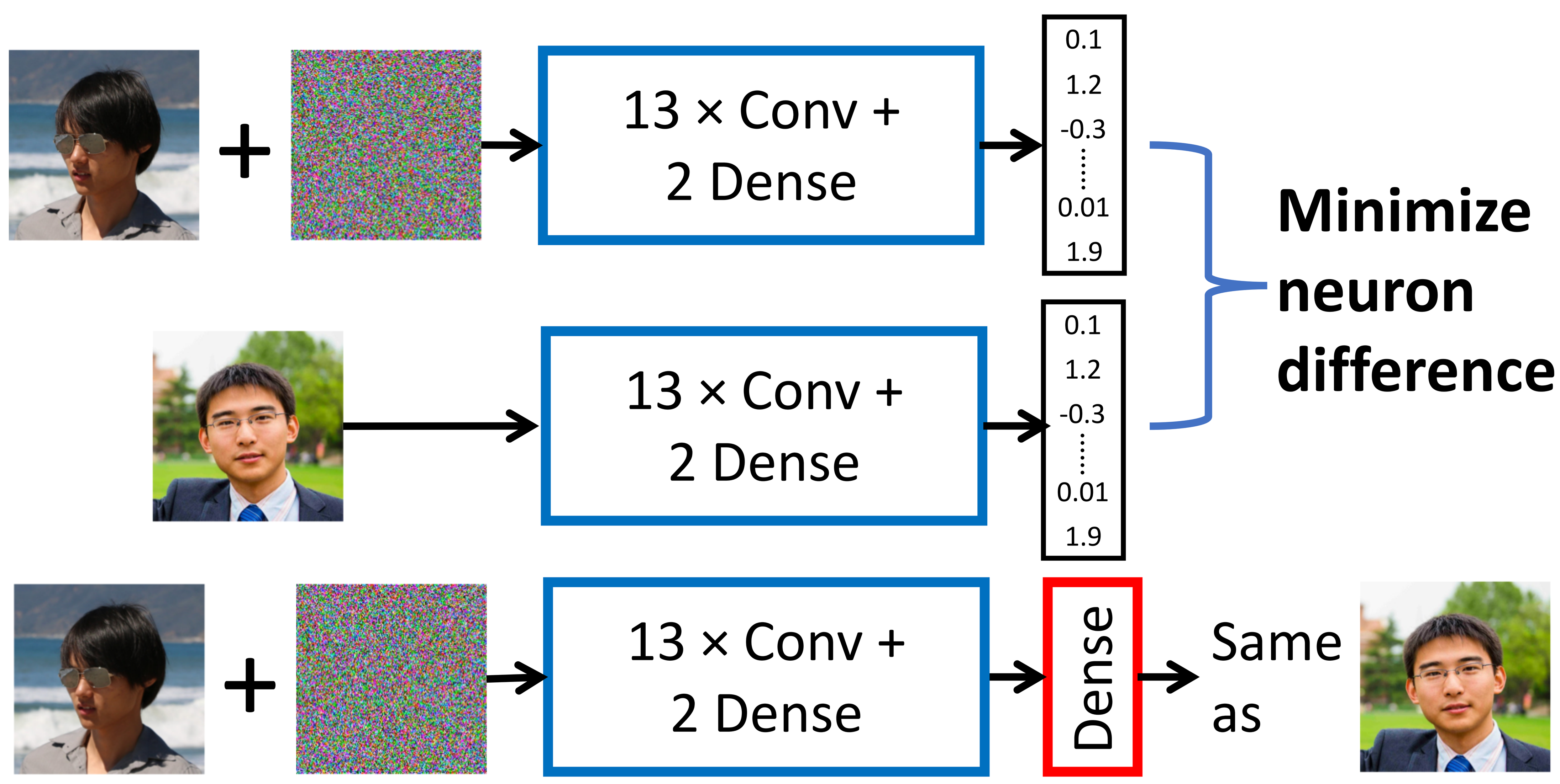

## Neuron Mimicking Adversarial Attack

### Adversarial attack on Student models

- Leverage knowledge of Teacher to attack Student
- Add unnoticeable perturbation to images to trigger misclassification

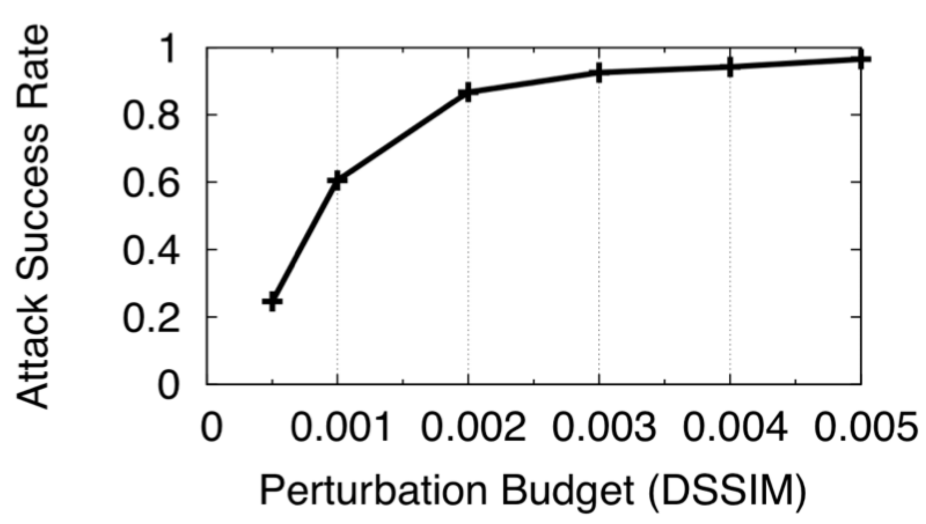
### Methodology

- Compute perturbations that mimic the internal representation of the target image on Student




## Attack Evaluation

- Train Student (98.55% accuracy)
  65 classes, 90 images/class, Teacher: VGG-Face
- 1,000 random source/target image pairs with different labels
- 92.6% successfully trigger targeted misclassification with unnoticeable changes




Attack success rate with different amount of perturbation


Successful attack samples (DSSIM=0.003)