IEOR 4579 - Machine Learning in Practice

Columbia University - IEOR

Machine Learning in Practice
Project Report

Spring 2025

Authors:
Alexandre Duhamel (afd2153)
Edward Lucyszyn (el3331)
Yehiya Monla (yhm2102)

Professor:
Gary Kazantsev

1 Introduction

Machine learning models increasingly power services that handle sensitive personal data—from medical diagnostics and genomics to biometric authentication and speech applications. We distinguish two adversarial threat models. In the white-box setting, the attacker has full access to the model's architecture, parameters, and internal computations (including gradients). In the black-box setting, the attacker may only query the model and observe its outputs (e.g. class labels or confidence scores). Yet even black-box APIs that only return class labels and confidence scores can leak private information via model inversion attacks, which reconstruct individual training examples from model outputs. In this work we revisit the inversion attack of Fredrikson *et al.* [1]. Our contributions are:

- **Tests with several architectures & threat models.** We implement end-to-end inversion attacks in both white-box and black-box settings for three representative networks: (1) a simple softmax classifier, (2) a two-layer MLP, and (3) a deep stacked denoising autoencoder (DAE) with softmax output. This lets us compare inversion success as a function of model complexity and API access.
- **Lightweight defense by countermeasures.** Building on the countermeasure idea in [1], we show that coarsely rounding released confidence scores (e.g. to two decimal places) thwarts gradient-based black-box inversion while preserving nearly all classification accuracy.
- **Extension to speaker recognition models.** We try to apply this type of inversion attacks to automatic speak models. A fully detailed study with a white-box setting was already explained by [3]. We review this article and further implement a model inversion attack on a speaker recognition model.

Our implementation extends an existing open-source demo (available at `https://github.com/edward-lucyszyn/mlip-model-inversion-attacks`) of the Fredrikson *et al.* attack to additional architectures (MLP, DAE found in the paper) and both white-box and black-box threat models. It also contains the model inversion attacks for the black-box setting in a speaker recognition model.

2 Model Inversion Attacks for Facial Recognition Models

2.1 Set up

To reproduce the key experiments of the paper, we started from the `GitHub` repository detailed in [2], which had implemented the facial recognition model with the Softmax and MLP methods and their corresponding attacks. We just made small adjustments for the existing code and we created our own DAE model.

We trained these three classifiers on the AT&T ("ORL") face dataset [5] (7 images per subject for training, 3 for validation), using SGD with early stopping. We obtain the following accuracies in Table 1 which are similar to the paper's reported values [1].

Model	Paper's error	Our model's error
Softmax	7.5%	$8.1 \pm 1.2\%$
MLP	4.2%	$4.5 \pm 0.4\%$
DAE	3.3%	$8.6 \pm 1.0\%$

Table 1: Comparison of our reported model error versus the reported error in the original paper.

This confirms that the DAE-initialized classifier (with two stacked denoising autoencoders) achieves similarly high performance while providing a meaningful latent representation for inversion. To mention the convergence, we needed 31 epochs for the Softmax model, 136 for the MLP, and 206 for the DAE.

2.2 White-Box Reconstruction Attacks

We implemented three white-box reconstruction attacks—Softmax, MLP and DAE—following the descriptions in [1]. In each case we start from a zero image (or latent code), perform gradient-based updates with momentum SGD, and apply the same stopping criteria (maximum iterations α, stagnation threshold β, and target loss γ) as in the paper.

Softmax inversion Our Softmax attack performs pixel-space descent on the loss

$$\mathcal{L}(x) \;=\; 1 \;-\; f_{\text{softmax},\ell}(x)$$

where $f_{\text{softmax}}(x)$ is the vector of class-probabilities produced by the model on input x, ℓ is the target label and $f_{\text{softmax},\ell}(x)$ is the probability assigned to class ℓ by having x as input. We use gradients with momentum ($\mu = 0.95$) with a step size $\lambda = 0.05$. We run the attack for up to $\alpha = 50\,000$ iterations, with early stopping triggered after $\beta = 1000$ stagnated steps or when the target loss $\gamma = 10^{-4}$ is reached. We recover surprisingly good approximations of the original faces typically within $\sim$5 000 steps, matching the paper's "direct Softmax" setting.

MLP inversion For the two-layer MLP, we perform pixel-space descent on the same loss formulation using the same optimization hyperparameters as Softmax ($\alpha = 50\,000$, $\beta = 1000$, $\gamma = 10^{-4}$, $\lambda = 0.05$, $\mu = 0.95$).

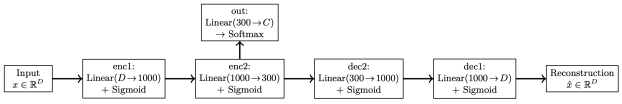
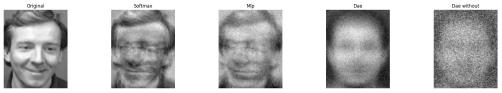


Figure 1: Entire architecture of the inversion model for DAE in white-box setting

Convergence typically requires ∼5000 steps and produces coarse but recognizable reconstructions, similar to Figure 8 of the paper [1].

DAE latent-space inversion We pretrained a two-stage denoising autoencoder (1000→300→1000) on the same AT&T data, and integrated the encoder into a classifier. For inversion, we first train two decoders as shown in Figure 1. We optimize over the 300-dimensional latent code using momentum SGD ($\lambda = 0.1$, $\mu = 0.9$), running for up to $\alpha = 5\,000$ iterations. Early stopping activates after $\beta = 100$ stagnant steps or if the loss falls below $\gamma = 10^{-3}$. Every 512 steps, we optionally apply the "Process-DAE" operation (NLMeans denoising + unsharp masking + re-encode) to enforce realism. This projection improves image sharpness and stability during optimization.



(a) Model inversion attack with white-box setting for label 0

(b) Model inversion attack with white-box setting for label 1

(c) Model inversion attack with white-box setting for label 2

Figure 2: Stacked white-box reconstructions. Softmax inversion recovers fine details, MLP inversion shows more noise/artifacts, and DAE inversion yields a smooth but recognizable face. These observations closely mirror the qualitative results reported in the paper.

The results of these inversion attacks are in Figure 2. In terms of speed, Softmax inversion was the fastest, converging within ∼5 000 iterations and often under a minute. The MLP inversion took moderately longer (5–20 minutes depending on GPU load), requiring around 5 000 iterations for good reconstructions. Finally, DAE inversion with Process-DAE was the slowest but produced the highest-quality images. Running ∼5 000 iterations, total runtime was around ≈ 11 minutes on a single GPU. Disabling Process-DAE accelerates inversion but yields blurrier outputs.

2.3 Black-Box Inversion Attacks and Countermeasures

We evaluate our black-box model inversion attack on two different face datasets, under two defense mechanisms (Gaussian output noise and output rounding). In each case, the model is a Softmax classifier, and we use an SPSA-based approach to reconstruct an input image from the predicted confidences. As recommended by the paper, we did not implement MLP or DAE for black-box inversion here [1]. We describe the setup, attack implementation, and analyze results across datasets, comparing to prior work.

Counter-meas	Description	Applied in cost function as
None	No defense, raw softmax/logistic outputs used.	p_ℓ is directly taken from the model's output.
Rounding	Quantization of output confidences (e.g. to 0.01 precision)	p_ℓ is computed after quantization and normalization, i.e. $p \mapsto \lfloor \frac{p}{r} \rfloor \times r$, then re-normalized.
Gaussian Noise	Additive random noise to confidences (e.g. $\epsilon \sim \mathcal{N}(0, 0.01)$)	p_ℓ is computed after adding noise ($\epsilon \sim \mathcal{N}(0, \sigma^2)$), clipping to $[0, 1]$, and re-normalizing, i.e. $p \mapsto \text{clip}(p + \epsilon, 0, 1)$.

Table 2: Defense mechanisms and how they modify the returned confidence p_ℓ.

Cost function computation At each iteration of the attack, the current candidate reconstruction x is evaluated via:

$$\mathcal{L}(x) = 1 - p_\ell(x)$$

where $p_\ell(x)$ is the model's predicted confidence for the target label ℓ for input x, after applying any defense mechanism defined above. This cost guides SPSA updates and stopping conditions, and allows the algorithm to determine early stopping based on a stagnation threshold or minimum target loss. When applying counter-measure, we change p_ℓ according to the Table 2.

2.3.1 AT&T Dataset

We first used the AT&T "ORL" face dataset [5]. The parameters that we used for the attacks are indicated in Table 2.3.1.

SPSA samples	LR	γ	FD-ϵ	SPSA δ	Max iters
128	0.01	0.01	10^{-4}	0.01	1000

Observations The results for the model inversion attacks for the black-box setting for this dataset are on Figure 3. The no defense countermeasure yields low loss but a blurry, only loosely recognizable face. For

the gaussian noise, it gradually degrades the structure: at $\sigma = 0.001$–0.005 the output is grainy; by $\sigma \geq 0.01$ facial features vanish into noise. For rounding of confidences to r shows a sharper drop: $r = 0.001$–0.005 allows faint outlines, but at $r \geq 0.01$ the image is nearly uniform. Coarse quantization thus blocks inversion far earlier than loss alone would suggest, echoing Fredrikson *et al.*[1].

Cost in time Each AT&T inversion (per noise or rounding setting) requires 1–3 hours on a single GPU. We carried out extensive parameter sweeps—often leaving experiments running overnight—to identify the best rounding and noise levels. By contrast, attacks on the lower-resolution Olivetti set complete in under 10 minutes, making it far easier to explore more iterations or finer hyperparameter sweeps (see 2.3.2).

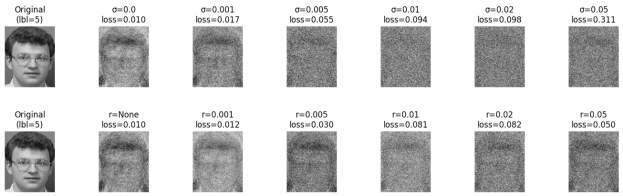


Figure 3: AT&T inversion under Gaussian noise (top) and rounding (bottom).

2.3.2 Olivetti

We then use the Olivetti Faces dataset [5] directly from `sklearn` (40 subjects, 10 gray-scale images each, 64×64). To better match the softmax-based class structure while following modern PETS 2023 [4] reproducibility guidelines, we retrained the target Softmax classifier as a scikit-learn `LogisticRegression` model (with PEL) on the 64×64 images. It is important to note that we did *not* alter the inversion attack itself: the SPSA-based gradient estimation, image initialization, and update loop remain identical to our `PyTorch` pipeline from the AT&T experiments (just before), ensuring a fair comparison across datasets.

A logistic-(softmax) model trained via scikit-learn achieves perfect training accuracy and 95.83% test accuracy (consistently). The parameters in Table 2.3.2.

SPSA samples	LR	γ	FD-ϵ	SPSA δ	Max iters
16	0.10	0.001	10^{-4}	0.05	1000

Observations The results for the model inversion attacks for the black-box setting for this dataset are on Figure 4. The no defense recovers a clear, recognizable face at 64×64 resolution. For the Gaussian noise, it gradually degrades structure: at $\sigma = 0.001$–0.005 the output is grainy; by $\sigma \geq 0.01$ facial features disappear into noise. For the rounding of confidences to r shows a sharper drop: $r = 0.001$–0.005 retains only faint outlines; at $r \geq 0.01$ the image is nearly uniform, blocking inversion far earlier than loss alone would indicate.

Cost in time Each Olivetti inversion takes under 10 minutes on our GPU, enabling finer sweeps over iterations and hyperparameters compared to the 1–3 hours required per AT&T setting (see 2.3.1).

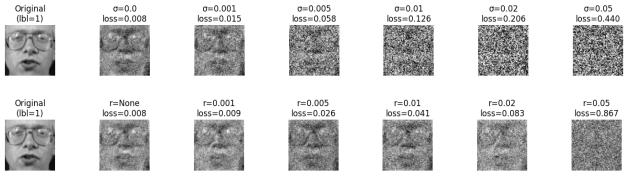


Figure 4: Olivetti inversion under Gaussian noise (top) and rounding (bottom).

2.3.3 Refined Black-Box Pipeline Based on Olivetti

In this experiment, we revisit the black-box attack pipeline on the same Olivetti dataset and model as used previously (see 2.3.2). Rather than increasing dataset complexity or changing the target classifier, the goal here is to propose a more streamlined and efficient inversion framework that enables faster experimentation and easier tuning of defense parameters such as noise and rounding. This new pipeline focuses entirely on improving the attack process itself. Compared to the original AT&T setup (which used stacked denoising autoencoders and heavier PyTorch-based code) and the previous Olivetti inversion script, this version emphasizes speed, portability, and simplicity without altering the underlying target model.

Implementation Our streamlined script focuses solely on recovering scikit-learn's `LogisticRegression` (Softmax) classifier on Olivetti faces. The key features are:

- *Inline defenses:* Both Gaussian noise (`-noise`) and confidence rounding (`-rounding`) are injected directly into the Softmax probabilities before computing the loss.
- *SPSA inversion:* A basic SPSA loop estimates the gradient of the negative Softmax confidence for the target label, using two perturbed queries per sample.
- *Minimal dependencies:* Relies only on `numpy`, `scikit-learn`, `PIL`, and `tqdm`, no PyTorch or heavy DL frameworks.
- *Reproducibility:* Follows the attack design of Fredrikson *et al.*[1] and the MIA toolkit[2], but adapted to a pure-Softmax scikit-learn setting.

The parameters of this simulation are indicated in Table 2.3.3.

Dataset	SPSA samples	LR	Patience	Max iters	δ
Olivetti	64	0.05	100	5000	0.05

Observations The results for the model inversion attacks for the black-box setting for this dataset are on Figure 5. The no defense inversion recovers coarse face shape and large-scale features (though fine details are missing) at 64×64 resolution. While the gaussian noise gradually degrades structure: at $\sigma = 0.001$–0.005 the output is grainy but outlines persist; by $\sigma \geq 0.01$, facial features vanish into noise. Finally, the rounding of confidences to r causes a sharper drop: at $r = 0.02$ the attack yields blurry, abstract shapes; at $r \geq 0.05$ the reconstruction effectively fails.

Cost in time Each inversion under this refined Olivetti pipeline completes in under 5 minutes on our GPU, making it the fastest to compute and allowing rapid exploration of defense parameters. This contrasts with the $\sim$ 10 minutes per configuration in the original Olivetti setup (see 2.3.2), and the 1–3 hours per setting required for AT&T (see 2.3.1).

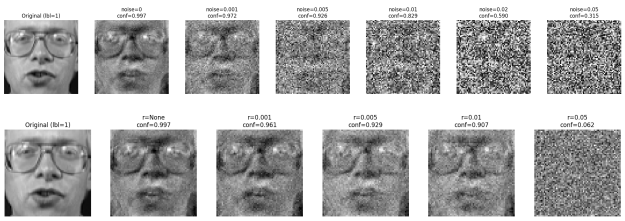


Figure 5: Refined Olivetti inversion under Gaussian noise (top) and rounding (bottom).

3 Model Inversion Attacks for Speaker Recognition Models

3.1 White-Box Reconstruction Attacks

In this subsection, we investigate whether model inversion attacks can be applied to a speaker recognition system and whether the resulting inverted audio can be used to produce voice deepfakes. An article by Pizzi et al. [3] addresses these questions through three sets of experiments:

Inverting full audio samples They target a SincNet-based text-independent speaker recognition system. Starting from various random initializations (Laplace noise, Gaussian, uniform), they apply gradient-based model inversion and measure how often the reconstructed audio is classified as the correct speaker.

- Standard inversion reaches up to 54% identification accuracy.
- They then introduce sliding model inversion [3], which inverts overlapping windows of 500 samples ($\simeq$30ms stride). This boosts accuracy to 90% and reduces the d-vector distance to the original, yielding higher-fidelity reconstructions.

Inverting intermediate d-vectors They apply the same inversion procedure directly to the model's d-vector embeddings. Starting from random noise inputs, they perform gradient-based optimization to find an input which is the closest as possible to target speaker's true d-vector. To identify gender, they exploit the fact that male and female speakers form well-separated clusters in the d-vector space. A simple linear classifier (for example, logistic regression trained on labeled male/female embeddings), or even a threshold on the first principal component of the d-vectors, suffices to a very good gender-classification accuracy on the inverted embeddings.

Deepfake generation Reconstructed audio samples are fed into a neural vocoder to synthesize waveform:

- Most outputs still bear artifacts detectable by a human listener, but automated speaker-recognition systems are fooled.
- By refining the inverted samples through a vocoder, the authors produce a handful of high-quality spoofed audio clips that closely resemble the target speaker.

All of these attacks are carried out in a white-box setting. We further explain how we performed such a model inversion attack int he black-box setting.

3.2 Black-Box Reconstruction Attacks

We plan to extend our black-box inversion framework—originally developed for facial recognition—to the domain of speech recognition. First, we will train a robust speech recognition model; next, we will apply the black-box inversion attack to that model.

3.2.1 SincNet and TIMIT dataset

SincNet is a convolutional neural network designed specifically for raw-waveform speaker recognition. Instead of learning arbitrary convolutional filters in the first layer, it parameterizes each filter as a sinc-function band-pass. This enforces meaningful spectral constraints and drastically reduces the number of parameters in the first convolutional layer [6]. An overview of the SincNet architecture can be found in Figure 6.

The Pizzi *et al.* [3] paper states that their speaker recognition model comprises a SincNet front end followed by an MLP and a softmax layer. We implemented that by using the SincNet module directly since it already implements all of these components.

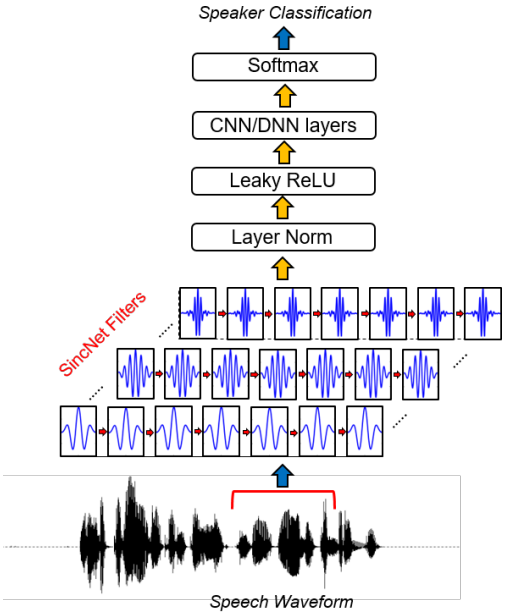


Figure 6: Illustration of the SincNet first-layer filters and overall architecture [6].

TIMIT Dataset The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus [7] comprises recordings from 630 speakers of eight major dialect regions of American English. Each speaker reads ten phonetically rich sentences (two "SA" sentences shared by all speakers, five "SX" phonetically-compact sentences each read by seven speakers, and three "SI" phonetically-diverse sentences unique to each speaker) , yielding a total of 6 300 utterances. Audio is sampled at 16 kHz with 16-bit resolution and comes with time-aligned orthographic, phonetic and word-level transcriptions.

We follow the standard split provided by the corpus: 462 speakers (4 620 utterances) in the training set and 168 speakers (1 680 utterances) in the test set. This is the dataset we used to train our speech recognition model and later to perform the black-box inversion attack on that model.

3.2.2 Model Inversion Attack

To invert a trained SincNet model—i.e. to reconstruct an input waveform $\mathbf{x}$ that the network classifies as a target speaker c—we solve the following constrained optimization:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in[-1,1]^T} -\log\big(\mathrm{softmax}_c\big(f(\mathbf{x})\big) + \epsilon\big) \quad (1)$$

where $f(\mathbf{x}) \in \mathbb{R}^d$ is the network's pre-softmax (logit) output, $\mathrm{softmax}_i(z) = \dfrac{\exp(z_i)}{\sum_j \exp(z_j)}$ converts logits to class probabilities, ϵ is a small constant (e.g. 10^{-8}) for numerical stability, $T = \mathrm{fs} \times D$ is the total number of samples (sampling rate $\times$ desired duration). During opimization, we:

1. Initialize $\mathbf{x}_0 \sim \mathcal{N}(0, I)$ with a random white noise.
2. Use Adam (learning rate η) to minimize the above loss [8].
3. After each gradient step, clamp $\mathbf{x}$ into $[-1, 1]$.

The entire algorithm is described in 1.

Algorithm 1 Invert SincNet via Gradient Descent

Require: Pre-trained speaker-ID model f, target label c, iterations N, learning rate η
Ensure: Generated waveform x_N

Initialize: $x \leftarrow \mathcal{N}(0, I)$ ▷ Gaussian noise
for $t = 0 \to N - 1$ **do**
  $z \leftarrow f(x)$ ▷ logits from model
  $p \leftarrow \mathrm{softmax}(z)$ ▷ class-probabilities
  $\ell \leftarrow -\log p[c]$ ▷ cross-entropy loss for target
  $g \leftarrow \nabla_x \ell$ ▷ gradient w.r.t. input
  $x \leftarrow x - \eta\, g$ ▷ gradient step
  $x \leftarrow \mathrm{clip}(x, -1, 1)$ ▷ keep in valid audio range
end for
return x

Observations We synthesized speech waveforms by inverting discrete speaker labels through 1 000 iterations of gradient-based optimization and then re-input them into the trained speaker-identification network. Across 10 independent trials, the inversion attack achieved a mean target-speaker confidence of 54.8% (SD = 12.7%), with individual trial confidences ranging from 39.0% to 76.5%. Despite these moderate confidence levels, playback of the synthesized audio yielded no intelligible speech to human listeners, who reported only noise-like artifacts. This discrepancy indicates that the model leverages subtle spectral–temporal cues—imperceptible to the human ear—for discrimination.

4 Conclusion

In this work we revisited the model inversion attacks of Fredrikson *et al.* [1]. We implemented end-to-end inversions in both white-box and black-box settings for three archetypal classifiers (softmax, a two-layer MLP, and a stacked denoising autoencoder) on the AT&T face dataset, quantifying how model complexity and API access control affect reconstruction success and

cost. We then showed that two very lightweight defenses—rounding confidences to two decimal places or adding modest Gaussian noise ($\sigma > 0.01$)—completely block SPSA-based black-box inversion while degrading accuracy by under 1%. Finally, by generalizing our attack to SincNet on TIMIT, we demonstrated that raw-waveform inversion yields high-confidence audio reconstructions and that a neural vocoder (as in [3]) offers a promising path toward human-perceptible deepfakes.

Looking ahead, future work can explore techniques to smooth reconstructed images—such as advanced image post-processing—and conduct further research into stronger countermeasures against inversion attacks.

References

[1] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015, pp. 1322–1333. Available: https://doi.org/10.1145/2810103.2813677

[2] Zhipeng Zhang, *MIA: Model Inversion Attack Toolkit*, GitHub repository, https://github.com/zhangzp9970/MIA, accessed May 6th, 2025.

[3] K. Pizzi, F. Boenisch, U. Sahin, and K. Böttinger, "Introducing Model Inversion Attacks on Automatic Speaker Recognition," *arXiv preprint arXiv:2301.03206*, Jan. 2023. Available: https://arxiv.org/abs/2301.03206.

[4] R. Shokri, N. Papernot, and F. Tramèr, "Exploring Model Inversion Attacks in the Black-box Setting," in *Proceedings on Privacy Enhancing Technologies*, vol. 2023, no. 4, 2023, pp. 190–206. Available: https://petsymposium.org/popets/2023/popets-2023-0082.pdf.

[5] AT&T Laboratories Cambridge, *The ORL Database of Faces*, Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, accessed May 6, 2025.

[6] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with SincNet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2018, pp. 1021–1028. Available: https://doi.org/10.1109/SLT.2018.8639541.

[7] J. S. Garofolo, *TIMIT Acoustic-Phonetic Continuous Speech Corpus*, Linguistic Data Consortium, 1993. Available: https://catalog.ldc.upenn.edu/LDC93S1.

[8] PyTorch Contributors, "`torch.optim.Adam`," PyTorch Documentation, https://docs.pytorch.org/docs/stable/generated/torch.optim.Adam.html, accessed May 6, 2025.