CYBERNETICS AND PHYSICS, VOL. 15, NO. 1, 2026, 85–92

NEURAL CLASSIFIER INVERSION: DECODING LATENT VISUAL SEMANTICS

Sergey Salishev
Computer Vision Systems
Saint Petersburg
198504
Russia
sergey.salishev@compvisionsys.com

Elina Vladimirova
Geoscan Ltd.
Saint Petersburg
194021
Russia
e.vladimirova@geoscan.ru

Article history:
Received 19.06.2026, Accepted 26.06.2026

Abstract

This paper presents semantic inversion of pretrained image classifiers as an experimental probe of the view that late neural representations behave as class-dependent error-correcting codewords. The method reconstructs ImageNet validation images from late classifier activations by optimizing pixels to match one tail-proximal feature distribution under a symmetric Kullback–Leibler (Jeffreys) objective. The protocol is not a generative-prior attack and not a data-free synthesis method: it uses only the frozen classifier, a smoothed optimization path, a feature-mean constraint, a total-variation prior, and a class-logit KL constraint. The experiment evaluates 21 Torchvision models spanning RegNet, ResNet, EfficientNet, ConvNeXt, and ViT families on 48 class-capped high-margin validation samples per model. Reconstruction quality is measured objectively with OpenCLIP ViT-L/14 image-image retrieval metrics, feature-KL reduction in bits, and top-1 classification accuracy of the reconstructed images. The results show strong semantic recovery for ViT-B/16 and EfficientNet models, moderate recovery for RegNet models and ConvNeXt-Tiny, weak retrieval for the larger ResNet and ConvNeXt variants, and adversarial nonsemantic failures for ResNet18 and ResNet34. Qualitative reconstructions further illustrate that successful recoveries preserve class-level form and color without copying exact pixel-level appearance.

Key words

Neural Network Interpretability, Image Classifiers, Latent Representations, Error-Correcting Codes, Semantic Features, Kullback–Leibler Divergence, CLIP Evaluation

1 Introduction

Deep image classifiers are usually evaluated by their predictive accuracy, yet their internal representations also determine robustness, quantization tolerance, distillation quality, and the practical limits of interpretability. From an information-theoretic point of view, a trained classifier can be read as a decoder: the representation produced by the trunk of the network is a noisy codeword, and the final classifier maps this codeword to a class. The relevant noise here is not sensor noise but the appearance variation of the world: many physical instances of a class, differing in viewpoint, color, or illumination, are realizations of the same underlying form, and the codeword is what survives this variation. In this reading a representation is decodable when the class codewords occupy separated decoding neighborhoods rather than overlapping ones. If this view is useful, late representations should contain enough stable class information to support reconstruction of semantic content even when the original pixels are not explicitly available. If it fails, inversion should collapse either to wrong semantic neighbors or to adversarial images that satisfy the classifier without looking like the source. For reproducibility, the experiment code and output-generation scripts are available in the public FEC experiment Git repository.[1]

Semantic inversion tests this claim by treating the classifier as a fixed encoder and optimizing an input image so that its internal activations match a target representation. Classical feature inversion and model inversion methods often match Euclidean features or constrain the image to the range of a pretrained generator [Mahendran and Vedaldi, 2015; Nguyen et al., 2017; Zhang et al., 2020; Chen et al., 2021]. DeepInversion is a useful point of comparison because it is also generator-free

[1] github.com/iceslice0/fec-experiment.git

and uses a total-variation (TV) image prior, but it is not aligned with the error-correcting-code (ECC) interpretation: it synthesizes class-representative inputs for distillation by matching batch-normalization statistics collected across multiple layers, together with a classification objective [Yin et al., 2020]. The overlap therefore provides independent evidence from an existing inversion protocol that a TV prior is a reasonable image-domain regularizer, while the present experiment makes a different claim. Rather than using these multi-tap low-order statistics, it uses one tail-proximal feature tap and matches the full softmax-normalized distribution induced by an individual late target codeword under a symmetric Kullback–Leibler (KL) objective. This also lets it apply to architectures without batch normalization such as ConvNeXt and the Vision Transformer (ViT). The novelty is not a new optimizer for natural image synthesis. It is the single-tap distributional inversion protocol used as a test of codeword decodability. The reconstruction is therefore a probe of information-geometric similarity in representation space, rather than a direct pixel-recovery or data-free distillation attack.

The evaluation focuses on pretrained Torchvision models from RegNet, ResNet, EfficientNet, ConvNeXt, and ViT families on ImageNet [Russakovsky et al., 2015; Marcel and Rodriguez, 2010; Radosavovic et al., 2020]. The main contributions are:

1. a self-contained single-tap protocol for inverting late classifier codewords from ImageNet validation images using feature-distribution matching, feature-mean regularization, a total-variation prior, and a class-logit constraint;
2. an objective reconstruction evaluation based on OpenCLIP ViT-L/14 image-image retrieval, paired-versus-mismatched similarity, feature-KL reduction, and classifier accuracy on reconstructions;
3. empirical evidence that semantic decodability is architecture-dependent: some late codewords decode to clear visual form, while others admit adversarial or weakly semantic solutions despite correct classifier labels.

2 Semantic Inversion Method

In this work, *semantic inversion* means reconstruction of an input image from the late representation of a pretrained classifier. The target representation is the hooked feature tensor closest to the classifier tail while still preserving a spatial or token-grid structure. This is a single-tap protocol: all feature-distribution matching is performed at this one late tensor, not at a collection of intermediate layers. The exact tap for each architecture family is given in the optimization pipeline. In the coding interpretation, this late feature tensor is treated as a learned codeword whose coordinates carry class-relevant evidence.

Given a frozen classifier and an ImageNet validation image x^*, the inversion starts from random noise and optimizes a synthetic image x. The goal is not to reproduce the exact pixels of x^*, but to produce an image whose internal representation belongs to the same semantic neighborhood as the target. Operationally it resembles activation maximization methods such as DeepDream [Mordvintsev et al., 2015], but the interpretation is different: the reconstruction is a decoding test for one late classifier codeword under the ECC view, not class synthesis or feature visualization for its own sake.

Success is then read in an independent semantic embedding space: if the reconstruction is close to its source there and the original classifier assigns it the same label, the target representation contains recoverable semantic information rather than only an arbitrary numerical signature. If the image is adversarial rather than semantic, the same test reports a failure.

2.1 Inversion as a Fixed-Point Search

Let the frozen trunk up to the hook be the encoder F, mapping an image x to its late feature tensor $F(x)$, and let $c^* = F(x^*)$ be the target codeword of a source image x^*. Decoding in the coding sense is the inverse problem of finding an image whose codeword equals the target,

$$\text{find } x \text{ such that } F(x) = c^*.$$

The encoder is many-to-one and has no closed-form inverse, so a solution is sought by simple iteration (fixed-point search) that descends the inversion objective $\mathcal{L}$ defined in Section 2.2. The update map is

$$T(x) = x - \eta\, \nabla_x \mathcal{L}(x), \qquad x_{k+1} = T(x_k),$$

started from the noise initialization x_0. A fixed point $\bar{x} = T(\bar{x})$ is a stationary point, $\nabla_x \mathcal{L}(\bar{x}) = 0$. The feature-matching term of $\mathcal{L}$ decreases as the induced codeword $F(\bar{x})$ approaches the target c^*; in the coding view exact equality is not required, since decoding succeeds once $F(\bar{x})$ falls within a decodable ball around c^*. Nonsemantic, adversarial solutions then arise for two distinct reasons. First, the iteration may halt at a stationary point whose codeword never enters this ball, producing an incomplete decode. Second, the ball may not be uniquely decodable: because F is many-to-one, the preimage of this ball can contain nonsemantic images, so even a codeword inside it need not correspond to a semantic source. The first is a limitation of the optimization, the second a degeneracy of the encoder.

This iteration is realized in the optimization pipeline below.

2.2 Loss Function

For the KL term, the late feature tensor $F(x)$ is first normalized per sample to zero mean and unit variance. If $\widehat{F}_{c,s}(x)$ denotes this normalized tensor at channel c and spatial or token position s, the softmax is the exponential

normalization

$$p_\theta(c \mid s, x) = \frac{\exp(\widehat{F}_{c,s}(x))}{\sum_{c'} \exp(\widehat{F}_{c',s}(x))}.$$

Thus at each spatial or token position the channel values are converted into positive fractions that sum to one. The divergence then compares relative activation patterns rather than absolute feature magnitudes, and the per-sample standardization keeps these distributions at a comparable scale. The same exponential normalization is applied to the final class logits when computing the logit KL term. The core inversion loss is the symmetric Kullback–Leibler (Jeffreys) divergence between the reconstruction and target feature distributions. The reported feature KL values are per-sample Jeffreys divergences in bits, averaged over the 48 samples; their absolute scale is large, so they are meaningful comparatively, between the initial and final state and across models, rather than as a calibrated per-coordinate quantity. In the ECC interpretation, the initial feature KL also gives a rough code-length scale. For an ideal binary code with independent random bit flips, the expected distance from a random point to a fixed codeword is one half of the code length; by analogy, 2 F-KL0 estimates the effective late-feature code length in the Jeffreys metric.

Writing $p = p_\theta(x)$, $q = p_\theta(x^*)$, and $z(x)$ for the final classifier logits, the objective used in the experiments is

$$\begin{aligned}
\mathcal{L}_{\mathrm{KL}}(x, x^*) &= \tfrac{1}{2}\big[D_{\mathrm{KL}}(p \,\|\, q) + D_{\mathrm{KL}}(q \,\|\, p)\big] = \tfrac{1}{2} D_J(p, q), \\
\mathcal{L}_2 &= \big\|\overline{F(x)} - \overline{F(x^*)}\big\|_2^2, \\
\mathcal{L}_{\mathrm{TV}} &= \max\big(TV(x) - 0.4\, TV(x^*),\, 0\big), \\
\mathcal{L}_{\mathrm{logit}} &= \tfrac{1}{2} D_J(\mathrm{softmax}(z(x)), \mathrm{softmax}(z(x^*)))\,, \\
\mathcal{L}(x) &= \mathcal{L}_{\mathrm{KL}}(x, x^*) \\
&\quad + \lambda_2\, \mathcal{L}_2 + \lambda_{\mathrm{TV}}\, \mathcal{L}_{\mathrm{TV}} + \lambda_{\mathrm{logit}}\, \mathcal{L}_{\mathrm{logit}}.
\end{aligned} \tag{1}$$

The KL term matches the softmax-normalized feature distributions and is the direct information-geometric component of the method. The feature-mean squared-error term removes the additive-shift degeneracy of softmax features. The total-variation term [Rudin et al., 1992] suppresses non-physical high-frequency artifacts while allowing edges and piecewise-smooth image structure to remain. This TV prior is the main technical overlap with DeepInversion [Yin et al., 2020]; its successful use there is prior evidence that TV is a reasonable image-domain regularizer. The logit KL term constrains the final ImageNet class distribution; the final classifier check then verifies whether the reconstruction receives the same class label as the original. The inversion runs use $\lambda_2 = 10$, $\lambda_{\mathrm{TV}} = 4 \times 10^{-3}$, $\lambda_{\mathrm{logit}} = 1$.

2.3 Optimization Pipeline

Each inversion run uses two copies of the same pretrained model. The target model is kept as the original frozen classifier and supplies both target features and final classification checks. The optimization model has the same weights, but every non-smooth rectified linear unit (ReLU) activation is replaced by Softplus,

$$\mathrm{softplus}_\beta(u) = \frac{1}{\beta} \log\big(1 + e^{\beta u}\big),$$

with $\beta = 3$. This recovers ReLU as $\beta \to \infty$ while giving a smooth gradient path from the reconstruction pixels to the hooked late features. Such smoothing approximates averaging each activation over additive Gaussian noise [Bishop, 1995; Camuto et al., 2020], which fits the ECC view, where decoding is robust to noise.

ImageNet validation images are preprocessed with the Torchvision resize-to-256 and 224×224 center-crop pipeline, then normalized by the standard ImageNet mean and standard deviation. Each model is inverted on its own high-margin correctly classified samples, selected as described in Section 3.

The feature hook is the last spatial feature map before the global pooling and classifier head. For RegNet, ResNet, and EfficientNet this hook is the final spatial trunk output and is effectively pre-linear after pooling. ConvNeXt and ViT differ: ConvNeXt has an intervening pooling and normalization tail before the final linear classifier, and ViT uses encoder patch tokens rather than the final classifier input. For ViT models the encoder output is adapted by dropping the class token and reshaping patch tokens into a spatial grid, so the feature size in Table 1 remains comparable to the convolutional feature maps.

The optimized image is initialized as Gaussian noise, $x_0 \sim \mathcal{N}(0, 0.01^2)$. Pixels are optimized for 4000 Adam steps with cosine learning-rate annealing from an initial learning rate of 0.25. After each iteration, the optimized image is re-standardized to zero mean and unit standard deviation. Reconstructions are saved after denormalization only; no color-space contrast post-processing is used for the reported metrics or figures.

The learning rate is selected to be the maximum value that allows the optimization to converge without numerical instability for all models; specifically, EfficientNet-B2 diverges at higher learning rates for some samples.

At the end of the run, the reconstructed images are classified by the original frozen model. The same saved original/reconstruction image pairs are then evaluated by the independent CLIP (Contrastive Language-Image Pre-training) protocol described below.

3 Experimental Protocol

Experiment consists of two parts: an objective semantic evaluation and a qualitative reconstruction check. The qualitative reconstruction figures are illustrative only. All quantitative claims rest on the CLIP retrieval metrics, feature-KL reduction, and classifier accuracy on the reconstructed images.

Both parts use the sample selection protocol with different batch sizes and class sets.

3.1 Models, Data, and Sample Selection

The experiment evaluates 21 pretrained Torchvision classifiers: nine RegNet models, four ResNet models, three EfficientNet models, three ConvNeXt models, and two ViT models. The RegNet variants are `regnet_x_400mf`, `regnet_y_400mf`, `regnet_x_800mf`, `regnet_y_800mf`, `regnet_x_1_6gf`, `regnet_y_1_6gf`, `regnet_x_3_2gf`, `regnet_y_3_2gf`, and `regnet_x_8gf`. The remaining models are `resnet18`, `resnet34`, `resnet50`, `resnet101`, `efficientnet_b0`, `efficientnet_b1`, `efficientnet_b2`, `convnext_tiny`, `convnext_small`, `convnext_base`, `vit_b_16`, and `vit_b_32`.

For each model and run parameterization, the ImageNet validation set is used as the candidate universe. The implementation shuffles the validation indices with a fixed seed and screens a 5000-image subset using the frozen classifier. Correctly classified samples are sorted by top-1 minus top-2 logit margin. The preselected pool is class-capped according to the requested class set and batch size, and the balanced high-margin candidates form the inversion batch. Because the actual codewords of a trained model are not directly observed, high-margin samples are used as empirical target-codeword candidates for inversion. This protocol is model-specific: each model is inverted on samples that it classifies correctly and confidently.

The objective table and the qualitative panels are produced by the same inversion program. They differ only in run parameters. The objective sweep uses all ImageNet classes, batch size 48, saves 48 reconstructions, applies the class cap of one image per ImageNet class. The qualitative check uses the fixed 9-class set, batch size 45, saves the best balanced set of 9 reconstructions. In both cases the candidate images are screened, margin-sorted, and class-capped before inversion.

3.2 Objective Semantic Evaluation

CLIP maps semantically related images to nearby embeddings and is therefore suitable for testing whether a reconstruction preserves the source image content [Radford et al., 2021]. The evaluator uses OpenCLIP ViT-L/14 with OpenAI-pretrained weights. It encodes the 48 originals and 48 reconstructions, L_2-normalizes the embeddings, and forms the image-image similarity matrix

$$S_{ij} = \langle e_i^{\mathrm{orig}}, e_j^{\mathrm{rec}} \rangle.$$

The reported metrics use a row/column-normalized matrix S_{norm} to reduce bias from unusually easy or hard originals and reconstructions. For $N = 48$ image pairs, let

$$r_i = \frac{1}{N-1} \sum_{j \neq i} S_{ij},$$

$$c_j = \frac{1}{N-1} \sum_{i \neq j} S_{ij},$$

$$g = \frac{1}{N(N-1)} \sum_{i \neq j} S_{ij}.$$

The normalized score is

$$(S_{\mathrm{norm}})_{ij} = \frac{S_{ij}\, g^2}{(r_i + \epsilon)(c_j + \epsilon)},$$

with $\epsilon = 10^{-8}$ used only to avoid division by zero. This rescales each source/reconstruction comparison by the average nonmatching similarity of its row and column while preserving the global similarity scale. Recall@k measures whether the paired reconstruction is among the top k retrieved reconstructions for the original. With 48 samples per model, chance Recall@1 is $1/48 \approx 2.1\%$. The matched and nonmatched columns report the diagonal and off-diagonal means of S_{norm}. Their difference is the semantic pairing gap, whose significance is assessed by a one-sided Wilcoxon signed-rank test over the 48 matched-versus-mismatched pairs. In coding terms this gap measures codeword separation under the CLIP metric: a reconstruction that is closer to its own source than to the other sources lies inside the semantic decoding neighborhood of its class codeword, and Recall@1 is the fraction of codewords that remain individually decodable. CLIP is used here as a tractable approximation of the semantic metric in which these decoding neighborhoods are defined.

3.3 Qualitative Reconstructions

The qualitative reconstruction uses selected final original-vs-reconstruction panels. It shows a fixed check set with ImageNet class IDs 24, 79, 409, 701, 712, 850, 950, 953, and 954, corresponding to the Torchvision category names *great grey owl*, *centipede*, *analog clock*, *parachute*, *Petri dish*, *teddy*, *orange*, *pineapple*, and *banana*.

4 Evaluation and Results

4.1 Objective Results Across Model Families

Table 1 reports the objective results for all 21 model-specific ImageNet inversion runs. All listed reconstructions reach 100.0% reconstructed-image classifier accuracy under the original frozen classifier.

The table columns have the following meaning. Weights are the number of model parameters in millions, and feature size is the hooked feature tensor shape excluding batch. Torch. R@1 is the official Torchvision ImageNet-1K top-1 accuracy for the default pretrained weights [PyTorch Vision Team, 2026]. Torch. R@1, R@1, and R@5 are percentages. F-KL0 and F-KLT report initial and terminal feature KL in bits after per-sample feature standardization (zero mean and unit variance); under the random-bit ECC heuristic, 2 F-KL0

is an effective code-length scale. Logit KL is the final class-distribution KL in bits. TV loss is the final weighted total-variation hinge diagnostic. Match and Nonmatch are the diagonal and off-diagonal means of the normalized CLIP similarity block S_{norm}, computed with OpenCLIP ViT-L/14 (OpenAI) embeddings. The final R@1/R@5 columns are the key retrieval diagnostics.

The strongest CLIP ViT-L/14 retrieval results are obtained by `vit_b_16`, `efficientnet_b0`, `vit_b_32`, `efficientnet_b1`, and `efficientnet_b2`. The ViT-B/16 run reaches 93.8% Recall@1 and 100.0% Recall@5 with a $768 \times 14 \times 14$ target feature grid. EfficientNet models also produce strong semantic pairings and very low final feature KL, from 0.10 to 0.36 bits.

RegNet models give moderate but consistent retrieval. The two 400MF RegNets and `regnet_x_800mf` reach 50.0% Recall@1, while `regnet_x_3_2gf` reaches 45.8%. The ResNet family is weaker under this protocol: `resnet18` and `resnet34` are near retrieval failure, with Recall@1 of 8.3% and 4.2% near the 2.1% chance level and a pairing gap that is at best marginal ($p = 0.04$ and $p = 0.09$); `resnet50` and `resnet101` reach only 20.8% Recall@1. Every other model in Table 1 except `resnet18` and `resnet34` has a significant pairing gap ($p < 10^{-4}$). The ResNet low retrieval should not be read only as a model-size effect. It also indicates inaccurate or adversarial inversion, because the final feature KL remains much larger than for other model families and the qualitative `resnet18` check gives a nonsemantic sparse solution.

ConvNeXt shows a different failure mode: feature KL and logit KL are low, but retrieval falls from 47.9% Recall@1 for `convnext_tiny` to 14.6% for `convnext_base`. Thus matching the hooked feature distribution and final class distribution is not by itself sufficient to guarantee image-level semantic pairing. While the correlation between inversion quality and CLIP retrieval is positive, it is not strong. This suggests model-specific adversarial decodability rather than a simple size trend: the optimized image can satisfy the frozen classifier and late-feature constraints without preserving source-recognizable image semantics.

The diagnostic columns show that inversion quality is not monotone in the number of weights, feature tensor size, baseline classification accuracy, or residual TV penalty. Small EfficientNets outperform much larger ConvNeXt and ResNet variants, while ViT-B/16 benefits from a denser $768 \times 14 \times 14$ patch grid. The evidence therefore points to architecture- and representation-dependent invertibility rather than a simple model-size trend: within RegNet and ConvNeXt the larger variants retrieve worse, whereas within ResNet the larger `resnet50` and `resnet101` outperform the smaller `resnet18` and `resnet34`. These data cannot separate representation capacity from optimization pathology in the ResNet case, so the conservative conclusion is weaker: small ResNet codewords are not semantically decodable by this protocol.

The model-specific selection is expected in the ECC view, not only a sampling caveat. The actual trained-model codewords are not directly observed, so high-margin validation images are used as empirical candidates for points close to them. Since the codewords are learned by a particular architecture and weight set, these candidates are naturally model-dependent. The low mean pairwise overlap of about 12% (5.6 of 48 images) is therefore consistent with the protocol. It also means that the table estimates favorable per-model decodability, not population-level reconstruction performance over the full ImageNet validation set.

4.2 Qualitative Reconstruction Evidence

Figure 1 provides qualitative evidence for the same inversion protocol under the fixed nine-class parameterization. All the models selected have feature tensor sizes in range $440 \times 7 \times 7$ to $2048 \times 7 \times 7$, so the comparison is comparatively direct. The reconstructions are not pixel-identical copies of the originals. They preserve global color, coarse shape, and class-level cues while changing most local details. This is the expected behavior for semantic inversion: many pixel-level appearances can map to nearby late feature codewords, and the reconstruction exposes information that is stable in the representation rather than incidental to the original image.

The `resnet18` panel is an adversarial empty-image perturbation: the optimization produces a disconnected pixel constellation on an almost uniform background that the frozen classifier labels as the target class. This is not semantic recovery. It shows that the `resnet18` feature and classifier constraints admit a non-natural adversarial solution. The effect is similar in spirit to one-pixel and sparse-pixel attacks [Su et al., 2019], but here the perturbation is produced by feature-codeword inversion rather than by an explicit small-norm attack objective.

In contrast, `resnet101` images contain distinctive features such as eyes, texture, shape, and color cues. ConvNeXt and RegNet panels are more structured, but still miss original details. EfficientNet and ViT are closest to the originals, mainly with lower contrast.

5 Conclusion

The experiment evaluates semantic inversion as a probe of late classifier feature codewords. The method reconstructs ImageNet validation images by matching softmax-normalized feature distributions under a symmetric KL objective, with feature-mean, total-variation, and class-logit constraints. It does not rely on a generator and does not optimize for pixel-identical recovery.

Across 21 pretrained Torchvision models from RegNet, ResNet, EfficientNet, ConvNeXt, and ViT families, all runs preserve the original classifier label on reconstructed images, but CLIP retrieval separates semantic from nonsemantic solutions. ViT-B/16 and EfficientNet

Model	Weights M	Feature size	Torch. R@1	F-KL0	F-KLT	Logit KL	TV loss	Non Match	Non match	R@1	R@5
regnet_x_400mf	5.50	400x7x7	74.9	559.82	6.13	0.0026	0.00	0.589	0.551	50.0	81.2
regnet_y_400mf	4.34	440x7x7	75.8	289.31	4.03	0.0012	0.00	0.617	0.562	50.0	83.3
regnet_x_800mf	7.26	672x7x7	77.5	866.17	16.22	0.0032	0.74	0.606	0.567	50.0	72.9
regnet_y_800mf	6.43	784x7x7	78.8	612.02	14.97	0.0027	0.33	0.565	0.536	35.4	62.5
regnet_x_1_6gf	9.19	912x7x7	79.7	846.18	43.64	0.0045	0.86	0.603	0.575	39.6	70.8
regnet_y_1_6gf	11.20	888x7x7	80.9	651.59	33.68	0.0045	2.46	0.595	0.570	33.3	56.2
regnet_x_3_2gf	15.30	1008x7x7	81.2	920.71	39.70	0.0018	2.04	0.611	0.576	45.8	72.9
regnet_y_3_2gf	19.44	1512x7x7	82.0	1359.17	55.66	0.0007	8.53	0.566	0.540	33.3	54.2
regnet_x_8gf	39.57	1920x7x7	81.7	1851.11	196.04	0.0009	11.18	0.566	0.545	29.2	47.9
resnet18	11.69	512x7x7	69.8	469.98	126.10	0.0048	0.66	0.600	0.597	8.3	16.7
resnet34	21.80	512x7x7	73.3	611.00	463.77	0.0114	2.33	0.588	0.585	4.2	22.9
resnet50	25.56	2048x7x7	80.9	1510.59	139.40	0.0105	18.28	0.557	0.539	20.8	64.6
resnet101	44.55	2048x7x7	81.9	1184.76	143.78	0.0044	6.22	0.587	0.565	20.8	58.3
efficientnet_b0	5.29	1280x7x7	77.7	359.46	0.21	0.0001	0.00	0.660	0.571	75.0	93.8
efficientnet_b1	7.79	1280x7x7	79.8	106.58	0.36	0.0003	0.00	0.693	0.581	70.8	93.8
efficientnet_b2	9.11	1408x7x7	80.6	89.30	0.10	0.0001	0.00	0.674	0.579	62.5	91.7
convnext_tiny	28.59	768x7x7	82.5	90.65	0.50	0.0062	1.87	0.608	0.561	47.9	64.6
convnext_small	50.22	768x7x7	83.6	107.76	0.47	0.0244	8.16	0.569	0.549	20.8	39.6
convnext_base	88.59	1024x7x7	84.1	283.41	0.76	0.0403	4.97	0.581	0.569	14.6	31.2
vit_b_32	88.22	768x7x7	75.9	176.91	1.04	0.0014	0.00	0.719	0.606	75.0	95.8
vit_b_16	86.57	768x14x14	81.1	436.77	4.01	0.0007	0.00	0.785	0.604	93.8	100.0

Table 1. Objective ImageNet inversion sweep with CLIP retrieval diagnostics.

models give the strongest pairings, while RegNet models remain moderate and consistent. ConvNeXt variants show that class agreement and low feature/logit KL do not guarantee image-level semantic retrieval. ResNet results are the main negative case: resnet18 and resnet34 are near retrieval failure, and the qualitative resnet18 panel shows an adversarial empty-image perturbation. Larger ResNets improve CLIP retrieval but still keep high final feature KL, so here the protocol does not cleanly separate representation limits from optimization limits.

The results support the coding-theoretic interpretation at the level relevant to inversion: late classifier representations often preserve class-level semantic neighborhoods that can be decoded back into plausible visual form by smoothed gradient-based optimization. Where retrieval succeeds, the matched-versus-mismatched CLIP gaps indicate that the class codewords stay separated enough to be individually decodable, which is the geometric signature expected of an error-correcting code. The failure cases are equally important: they show that classifier agreement alone is too weak, because the optimization can also find nonsemantic adversarial decodings. At the same time, the nonmonotone dependence on parameter count and feature size shows that semantic invertibility is not a simple consequence of ImageNet accuracy or model scale.

Future work should separate architecture effects from optimization effects and test whether better smoothing or richer priors can improve the failure cases without turning the method into instance-level pixel recovery. The image prior may also have a use beyond analysis: when a model inverts stably to semantic images under it, the prior-constrained inversion approximates the semantic preimage of each decodable ball, so running inputs through it could act as a prefilter against adversarial examples, which reach a codeword through a nonsemantic image.

References

Bishop, C. M. (1995). Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, **7** (1), pp. 108–116.

Camuto, A., Willetts, M., Simsekli, U., Roberts, S. J., and Holmes, C. C. (2020). Explicit regularisation in Gaussian noise injections. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33. https://arxiv.org/abs/2007.07368.

Chen, S., Kahla, M., Jia, R., and Qi, G.-J. (2021). Knowledge-enriched distributional model inversion attacks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, pp. 16158–16167.

Mahendran, A. and Vedaldi, A. (2015). Understanding deep image representations by inverting them. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp. 5188–5196.

Marcel, S. and Rodriguez, Y. (2010). Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, ACM, pp. 1485–1488.

Mordvintsev, A., Olah, C., and Tyka, M. (2015). Inceptionism: Going deeper into neural networks. https:

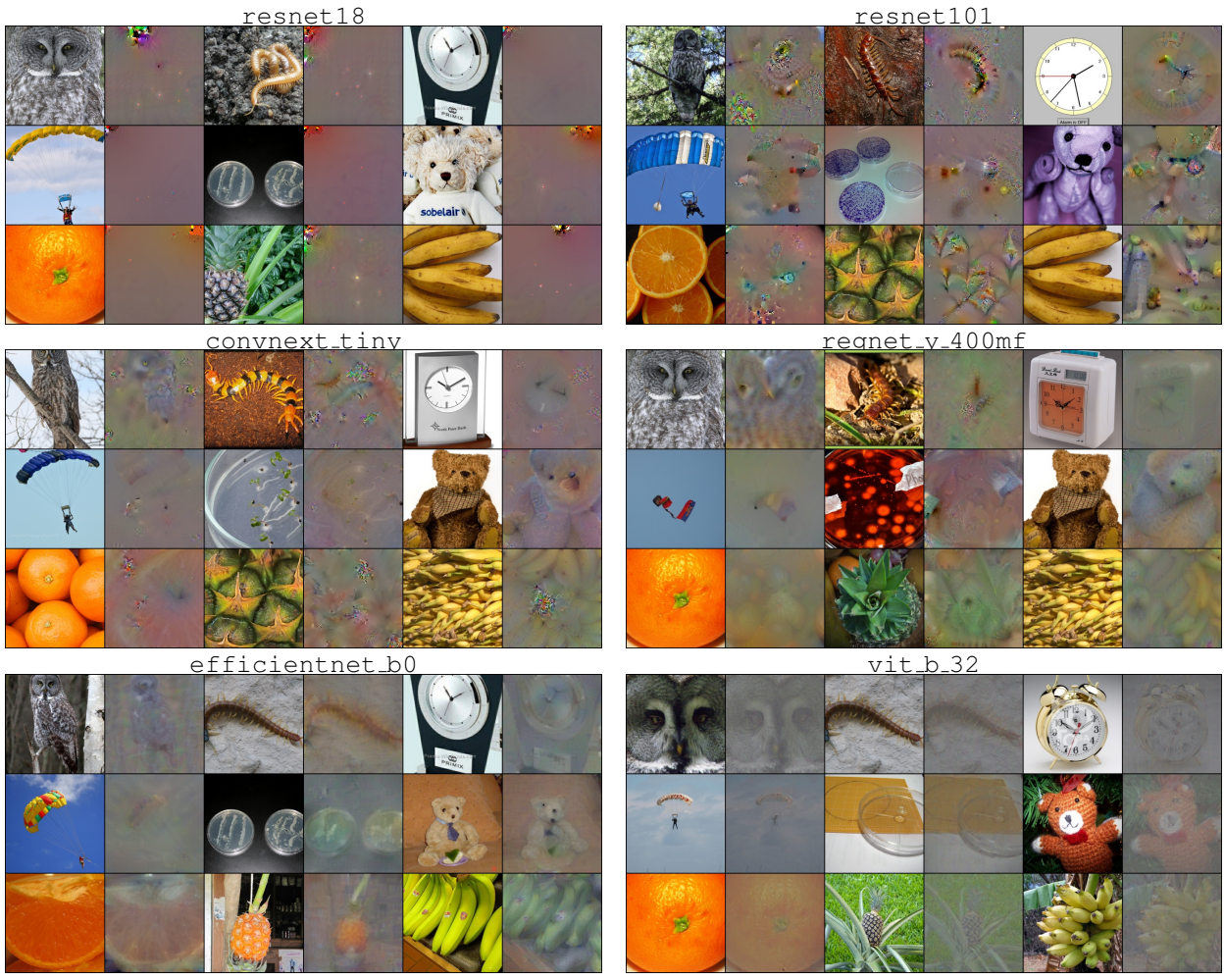

Figure 1. Final original-vs-reconstruction comparisons for the fixed 9-class check set. Each panel shows the same selected ImageNet classes for a different pretrained architecture.

//research.google/blog/inceptionism-going-deeper-into-neural-networks/. Published June 18, 2015. Accessed June 26, 2026.

Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A., and Yosinski, J. (2017). Plug & play generative networks: Conditional iterative generation of images in latent space. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, pp. 3510–3520.

PyTorch Vision Team (2026). Models and pretrained weights. https://docs.pytorch.org/vision/stable/models.html. Accessed June 26, 2026.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, vol. 139 of *Proceedings of Machine Learning Research*, PMLR, pp. 8748–8763. https://proceedings.mlr.press/v139/radford21a.html.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. (2020). Designing network design spaces. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 10425–10433.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, **60** (1–4), pp. 259–268.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, **115** (3), pp. 211–252.

Su, J., Vargas, D. V., and Sakurai, K. (2019). One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, **23** (5), pp. 828–841.

Yin, H., Molchanov, P., Alvarez, J. M., Li, Z., Mallya, A., Hoiem, D., Jha, N. K., and Kautz, J. (2020). Dreaming to distill: Data-free knowledge transfer via DeepInversion. In *2020 IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, pp. 8715–8724. https://openaccess.thecvf.com/content_CVPR_2020/html/Yin_Dreaming_to_Distill_Data-Free_Knowledge_Transfer_via_DeepInversion_CVPR_2020_paper.html.

Zhang, Y., Jia, R., Pei, H., Wang, W., Li, B., and Song, D. (2020). The secret revealer: Generative model-inversion attacks against deep neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 250–258.