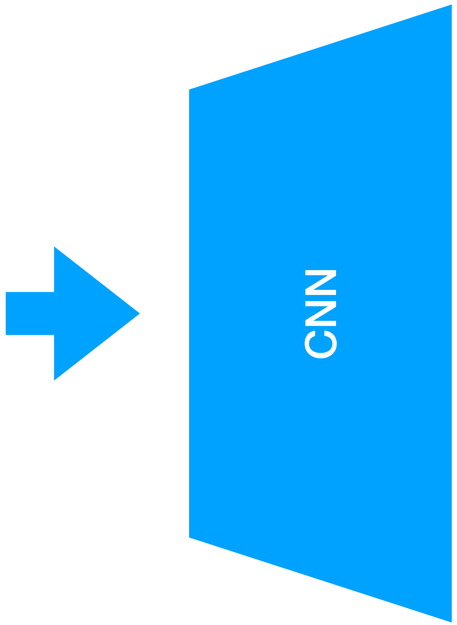
Plug-In Inversion

Model-Agnostic Inversion for Vision with Data Augmentations

Ghiasi, Kazemi, Reich, Zhu, Goldblum, Goldstein

Model Inversion

Inference



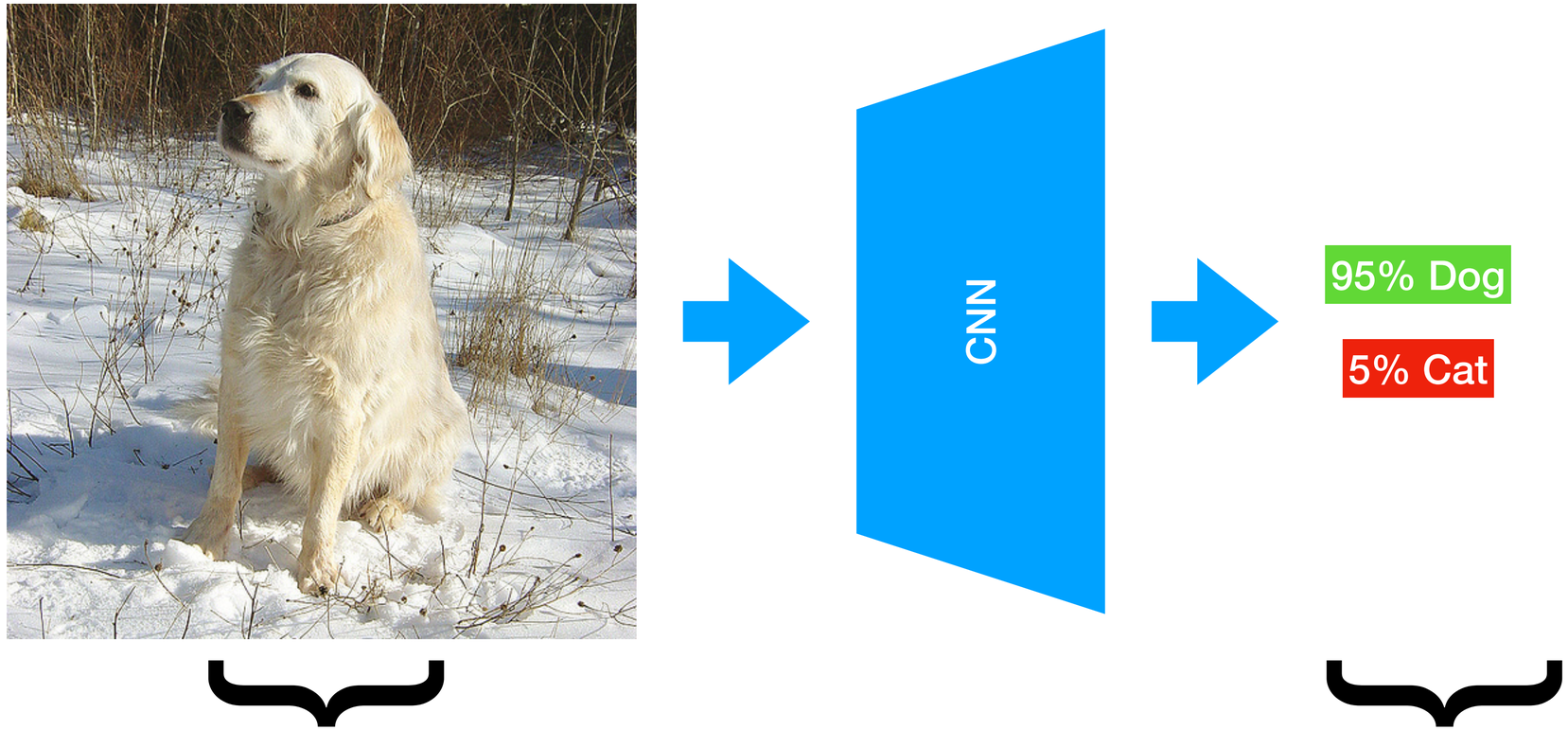
Inference

CNN

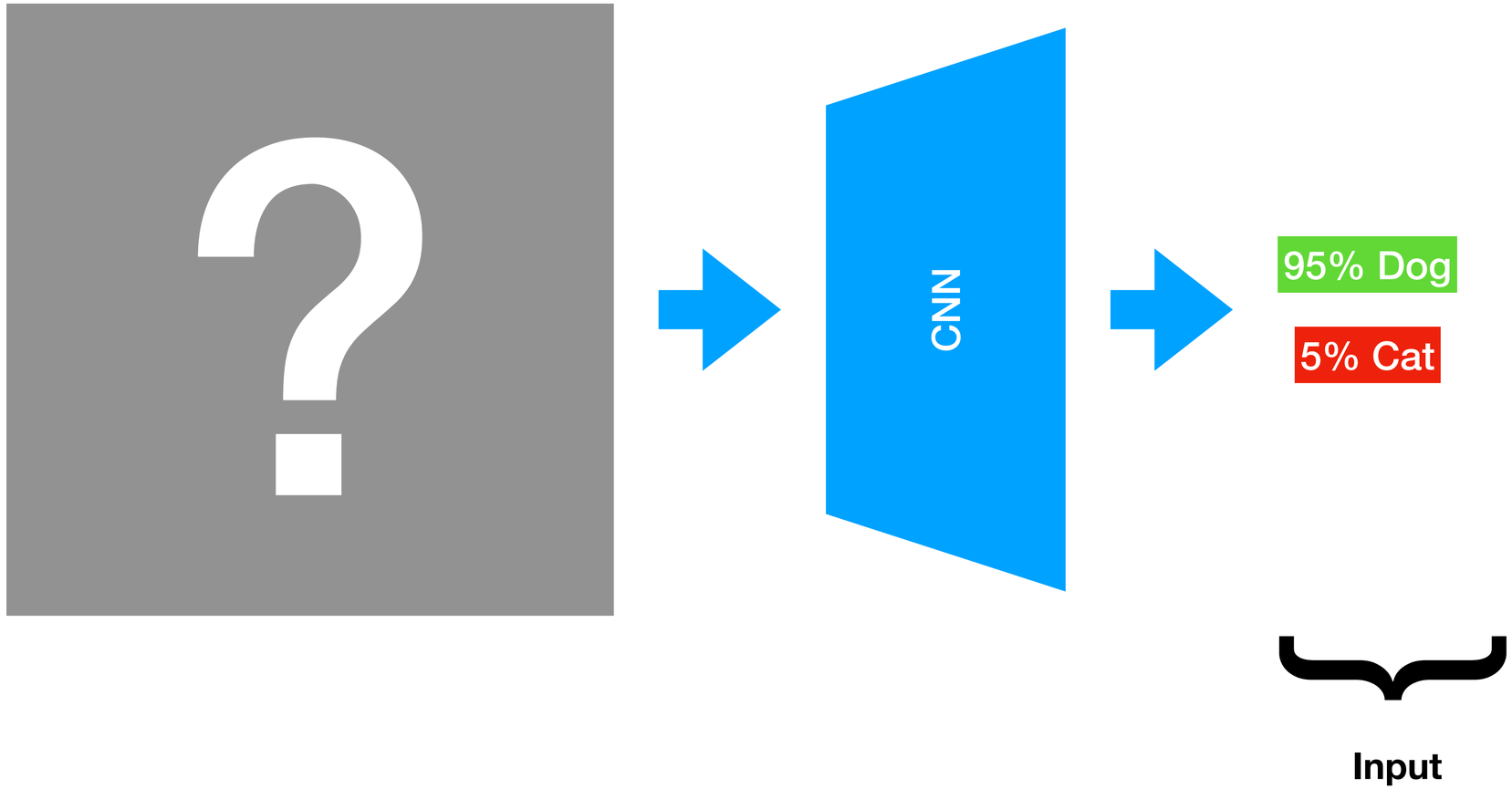
95% Dog

5% Cat

Inference



Model Inversion



Model Inversion

CNN

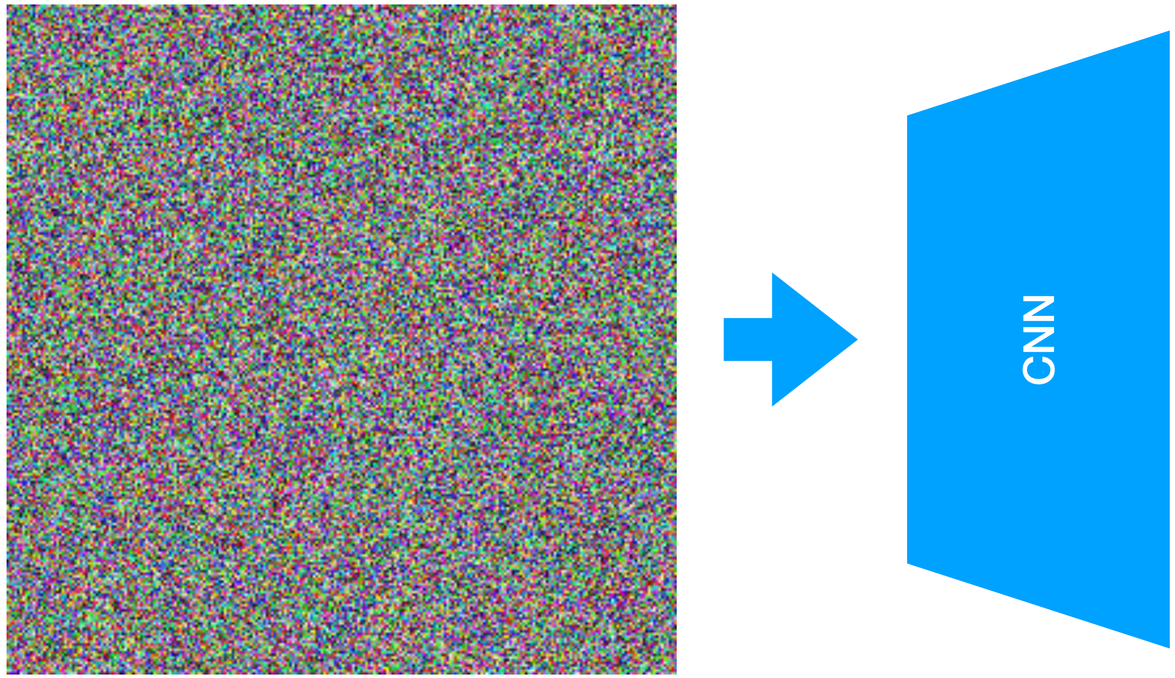
95% Dog

5% Cat

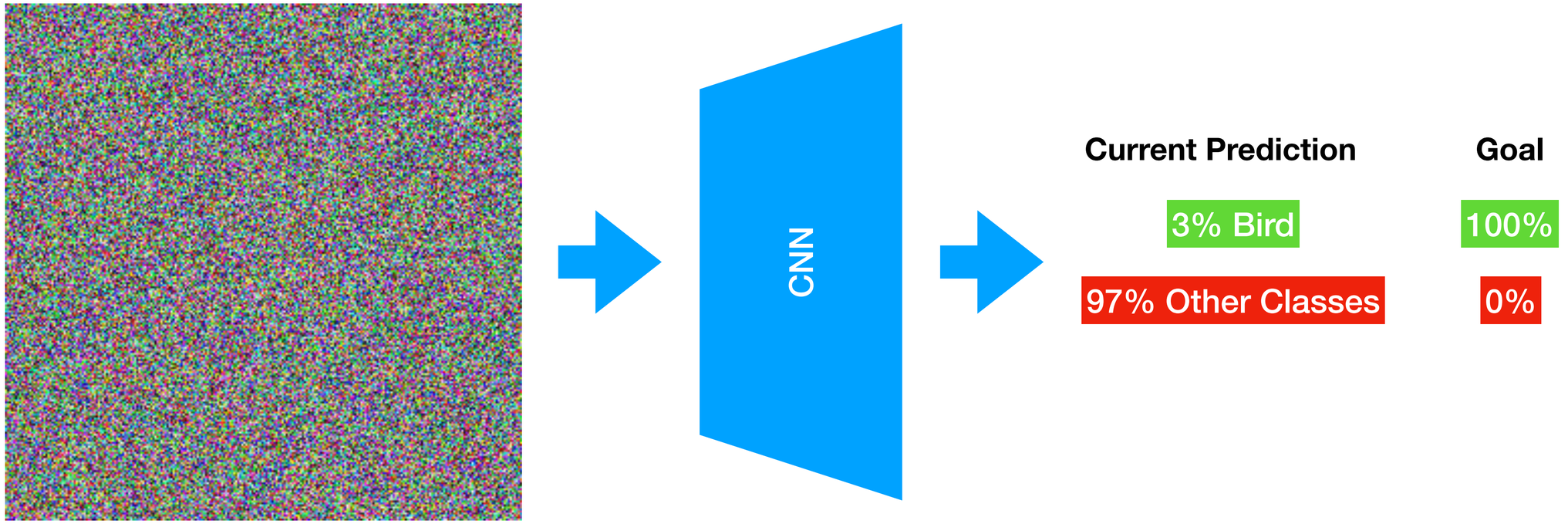
Output

Input

Gradient Descent

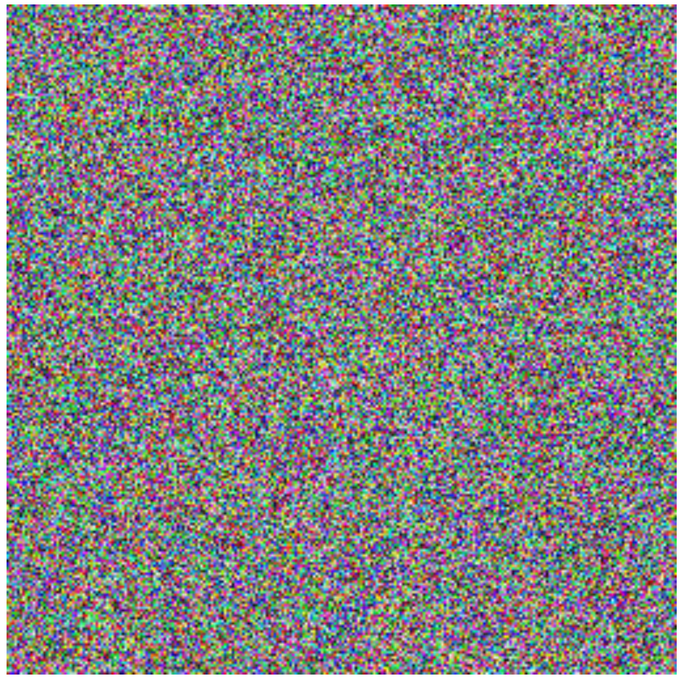


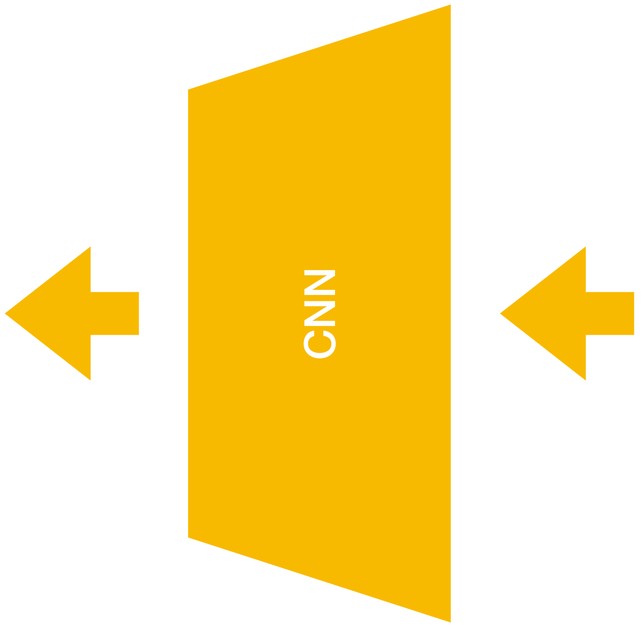
Gradient Descent



$$x_{t+1} = x_t - \epsilon \nabla Loss(net(x_t), target)$$

Gradient Descent

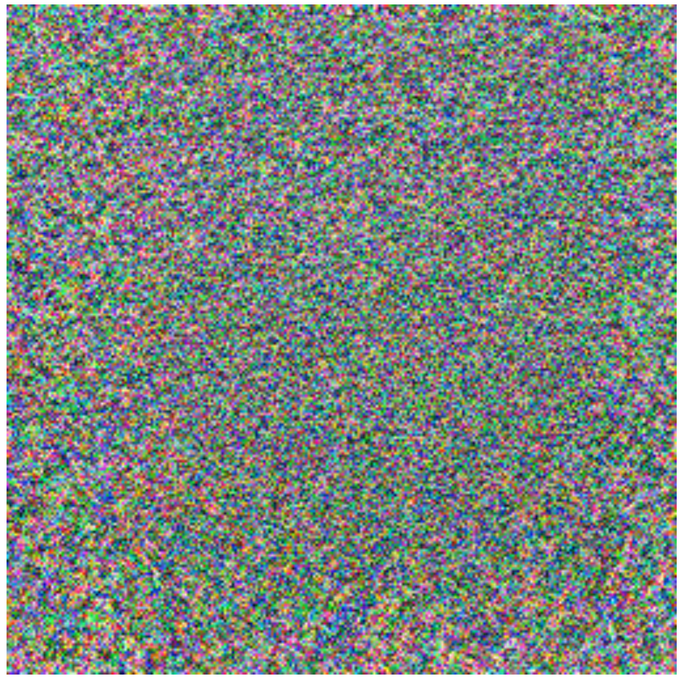


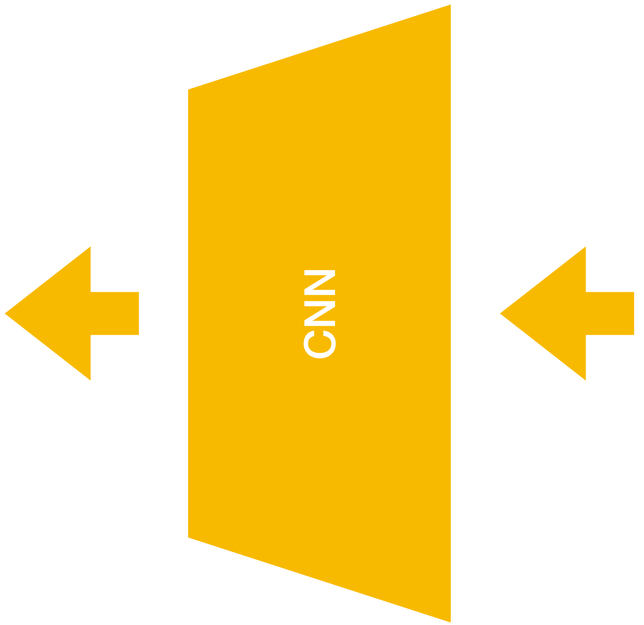


Current Prediction	Goal
3% Bird	100%
97% Other Classes	0%

$$x_{t+1} = x_t - \epsilon \nabla Loss(net(x_t), target)$$

Gradient Descent

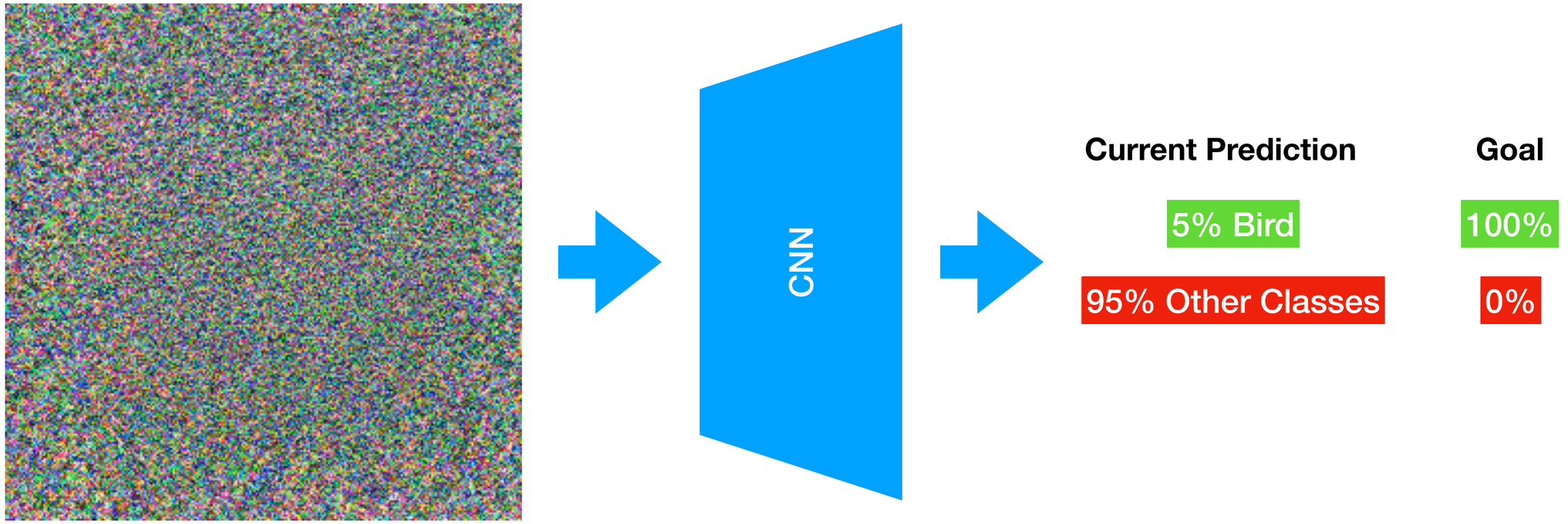




Current Prediction	Goal
3% Bird	100%
97% Other Classes	0%

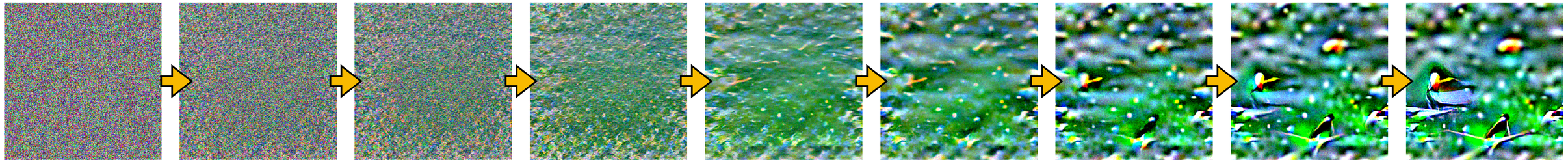
$$x_{t+1} = x_t - \epsilon \nabla Loss(net(x_t), target)$$

Gradient Descent



$$x_{t+1} = x_t - \epsilon \nabla Loss(net(x_t), target)$$

Gradient Descent



DeepInversion (DI)

Dreaming to Distill: Data-Free Knowledge Distillation Deep Inversion

DeepInversion

$$Loss(x, y) = \lambda_{main} Xent(net(x), y) + \lambda_{tv} tv(x) + \lambda_{norm} \|x\|_2 + \lambda_{bn} \left| bn(dataset) - bn(x) \right|$$

DeepInversion

Increase

$$Loss(x, y) = \lambda_{main} Xent(net(x), y) + \lambda_{tv} tv(x) + \lambda_{norm} \|x\|_2 + \lambda_{bn} \left| bn(dataset) - bn(x) \right|$$

- Different Size

DeepInversion

Increase

$$Loss(x, y) = \lambda_{main} Xent(net(x), y) + \lambda_{tv} tv(x) + \lambda_{norm} \|x\|_2 + \lambda_{bn} \left| bn(dataset) - bn(x) \right|$$

Decrease

- Different Size

DeepInversion

$$Loss(x, y) = \lambda_{main} Xent(net(x), y) + \lambda_{tv} tv(x) + \lambda_{norm} \|x\|_2 + \lambda_{bn} \left| bn(dataset) - bn(x) \right|$$

- Different Size
- Different Dataset?

DeepInversion

$$Loss(x, y) = \lambda_{main} Xent(net(x), y) + \lambda_{tv} tv(x) + \lambda_{norm} \|x\|_2 + \lambda_{bn} \left| bn(dataset) - bn(x) \right|$$

- Different Size
- Different Dataset?
- Different Training Method?

DeepInversion

$$Loss(x, y) = \lambda_{main} Xent(net(x), y) + \lambda_{tv} tv(x) + \lambda_{norm} \|x\|_2 + \lambda_{bn} \left| bn(dataset) - bn(x) \right|$$

- Different Size
- Different Dataset?
- Different Training Method?
- Different Architecture?
  - What about architectures without BN?
    - ViTs
    - MLPs

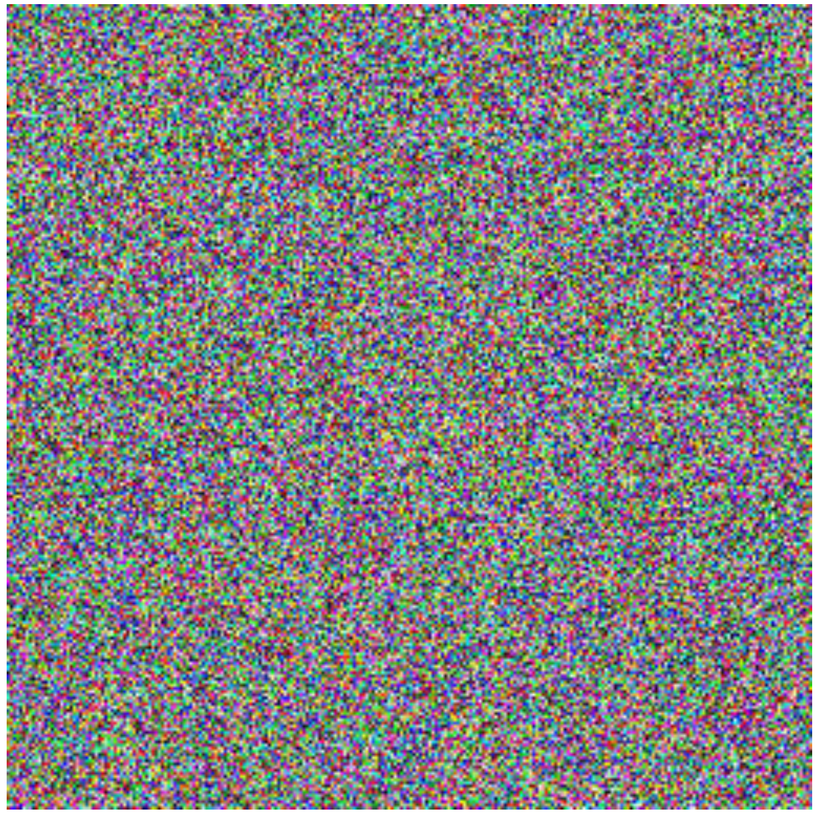

Plug-In Inversion

Plug-In Inversion

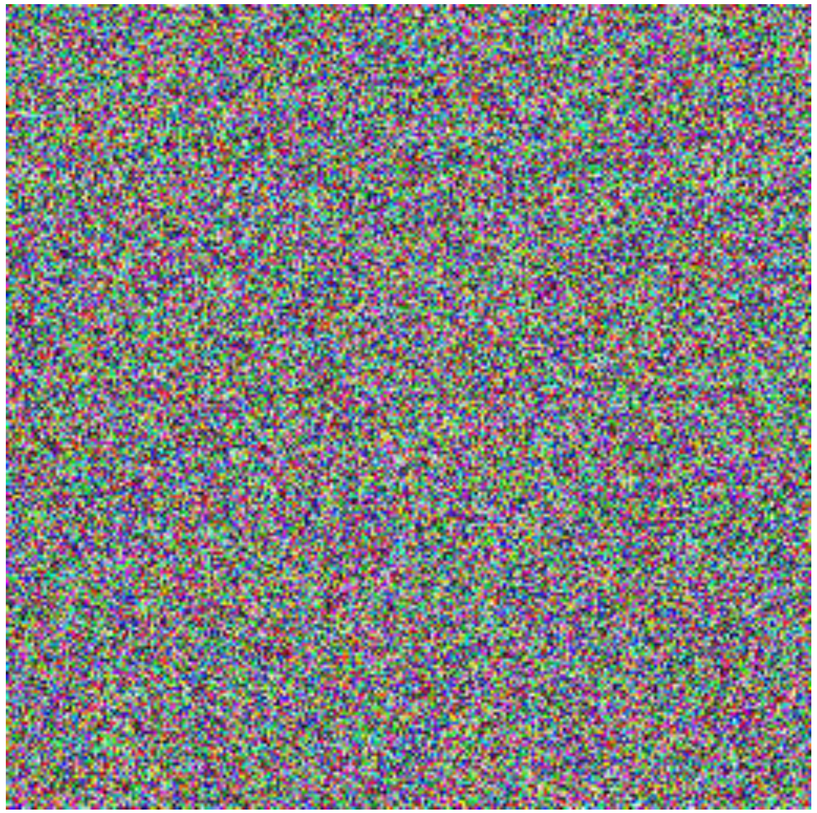
$$Loss(x, y) = Xent(net(Aug(x)), y)$$

- Augmentations:
  - Centering
  - Zooming
  - ColorShift
  - Ensembling

Centering

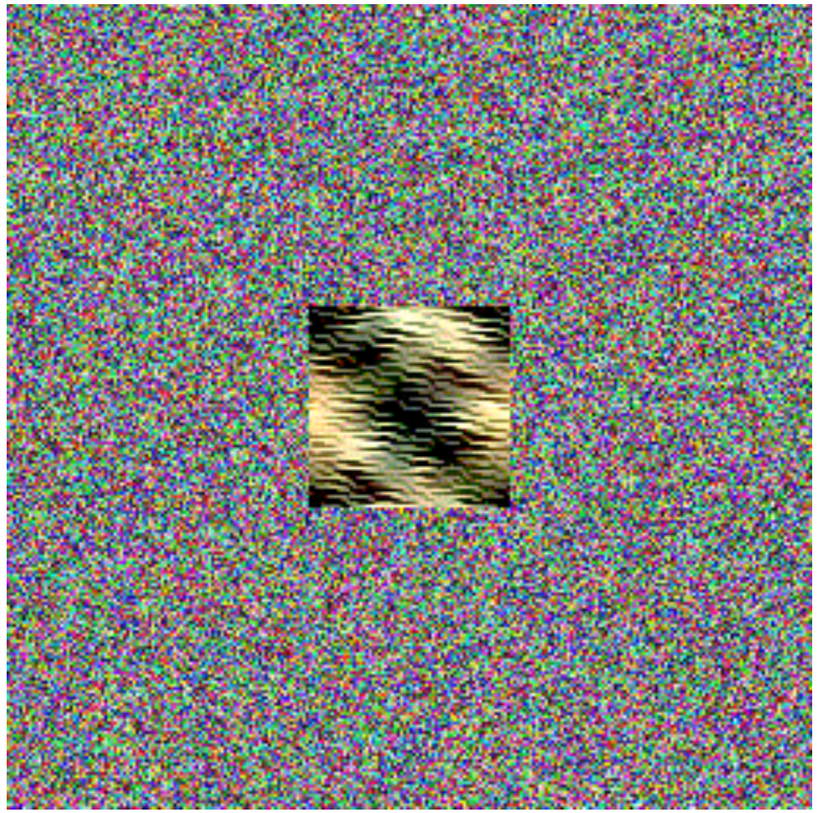


Centering



Not Centering

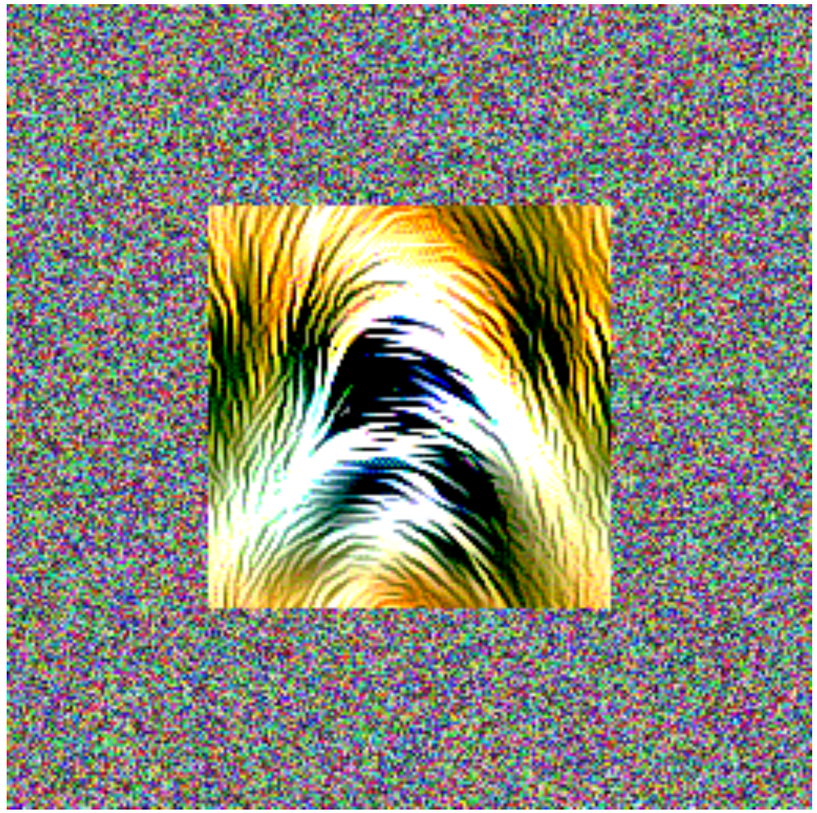
Centering



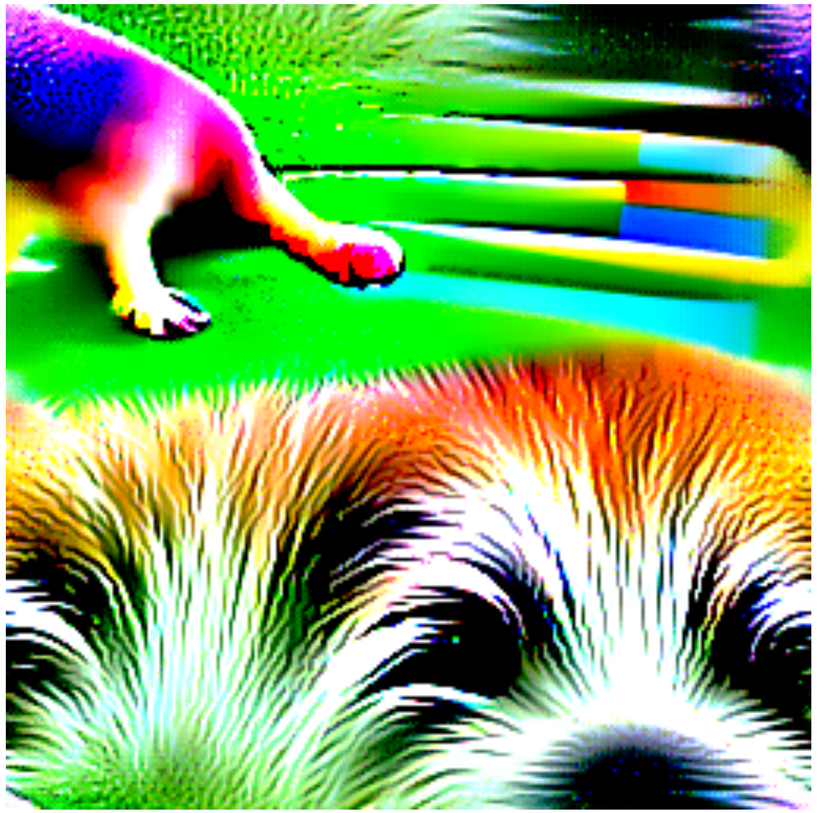
Centering

Not Centering

Centering

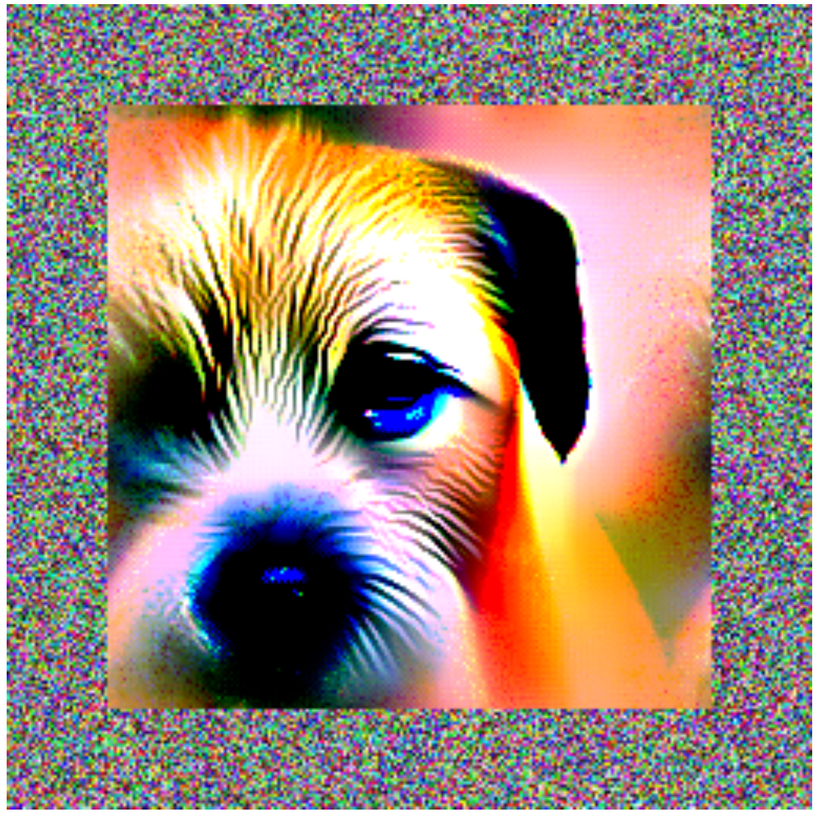


Centering



Not Centering

Centering

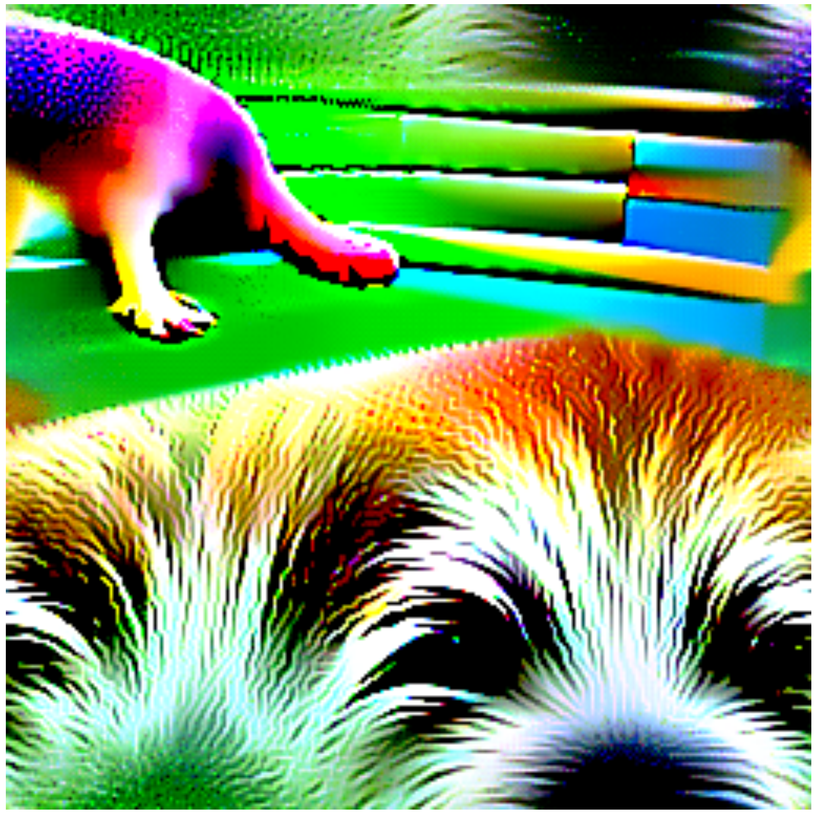


Centering

Not Centering

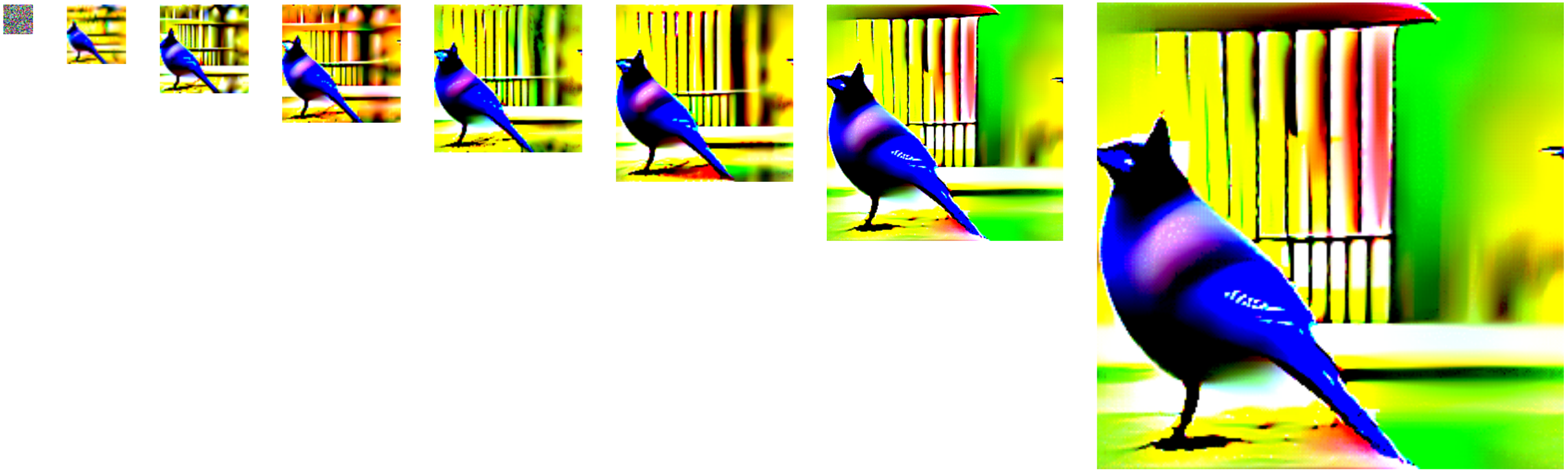
Centering

Centering

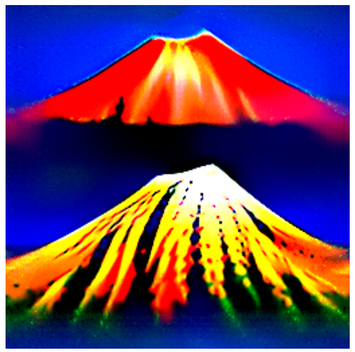


Not Centering

Zooming



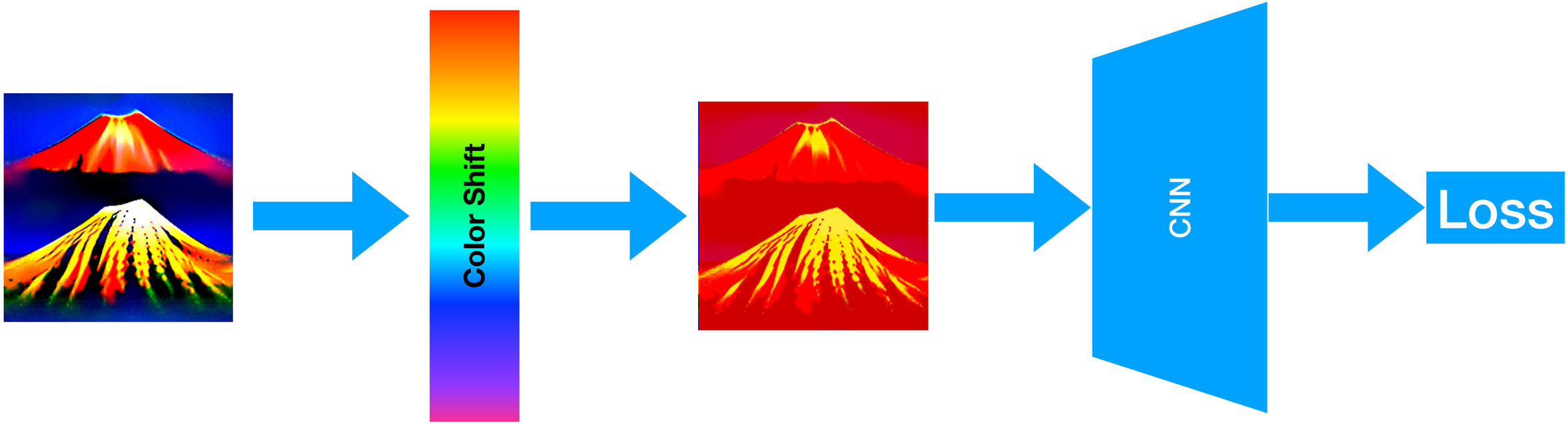
ColorShift



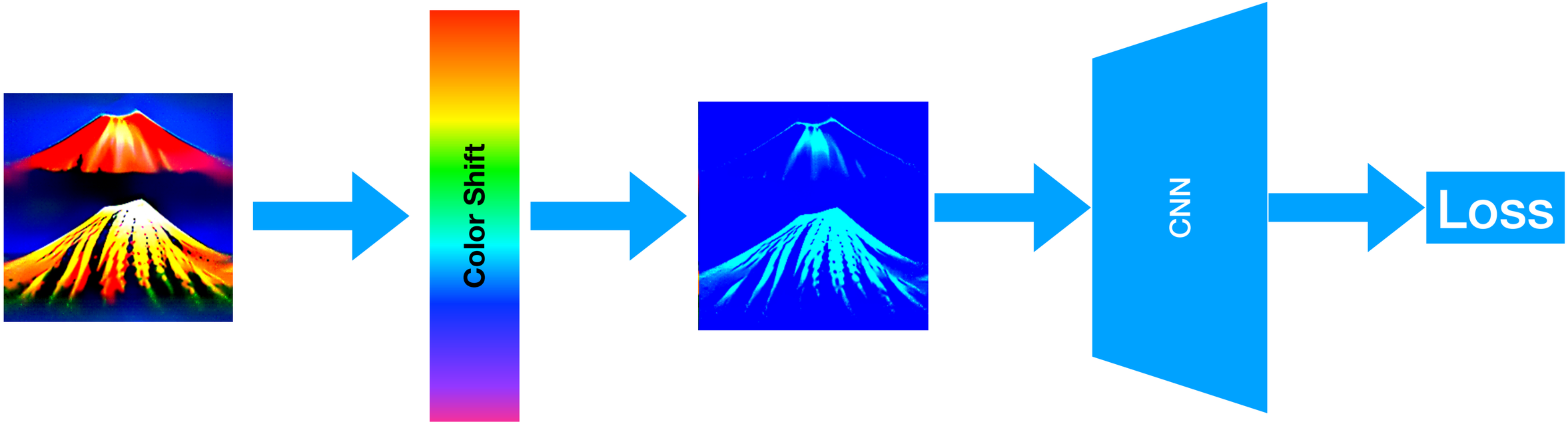
CNN

Loss

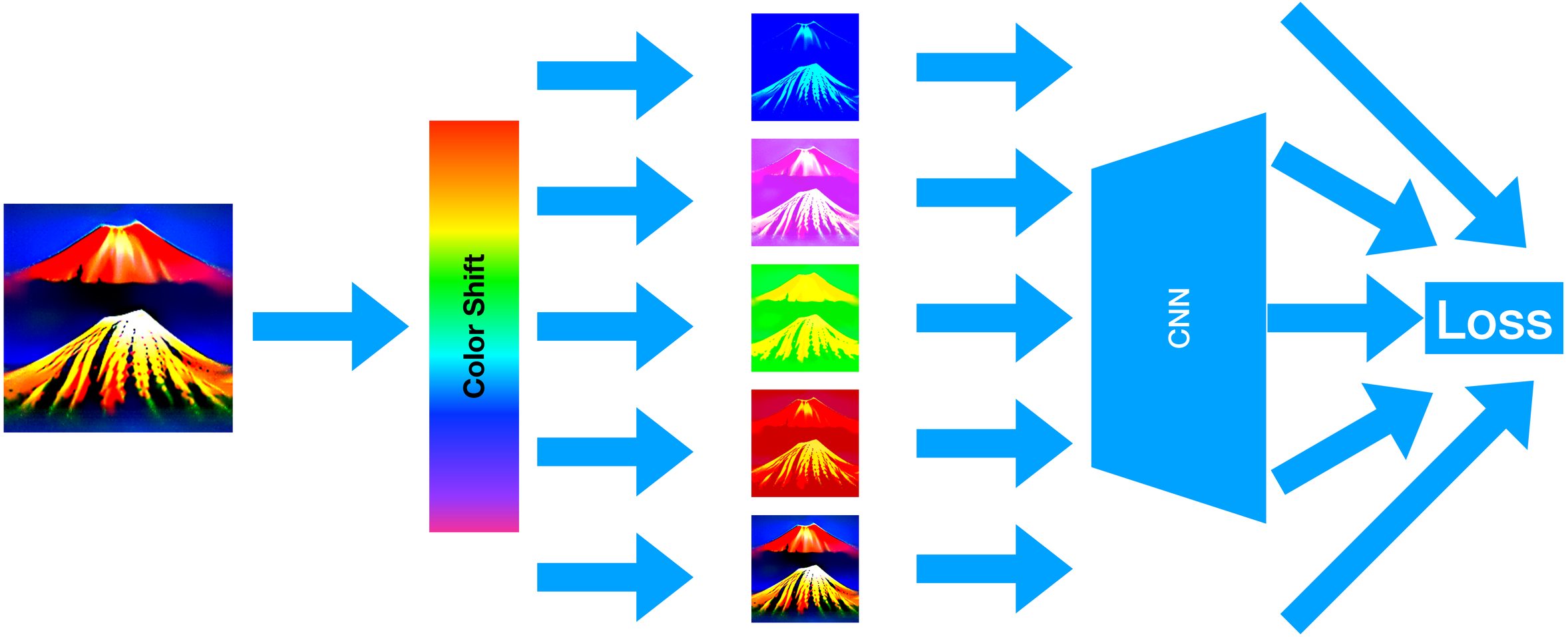
ColorShift



ColorShift



Ensembling



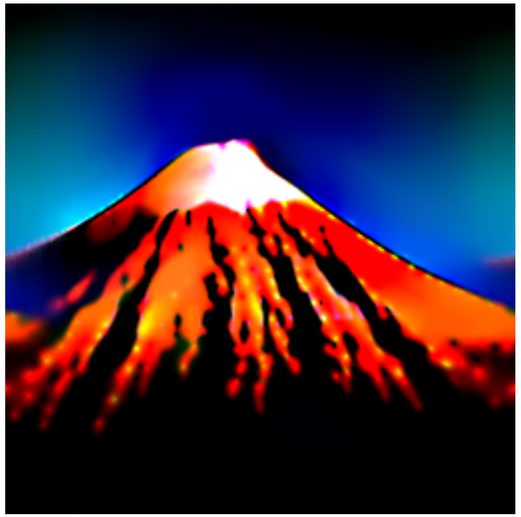
Results

Results - Robust ResNet50

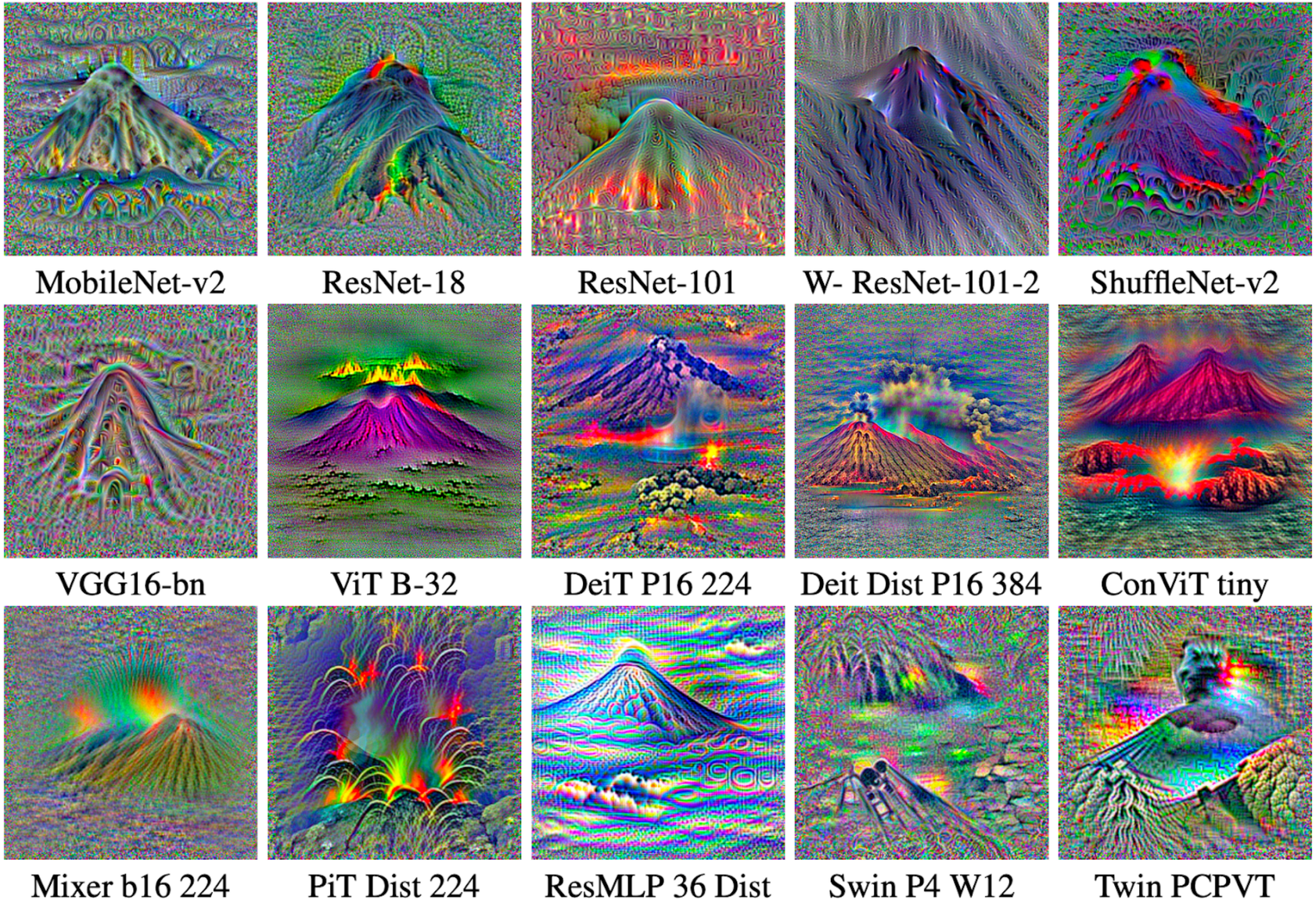




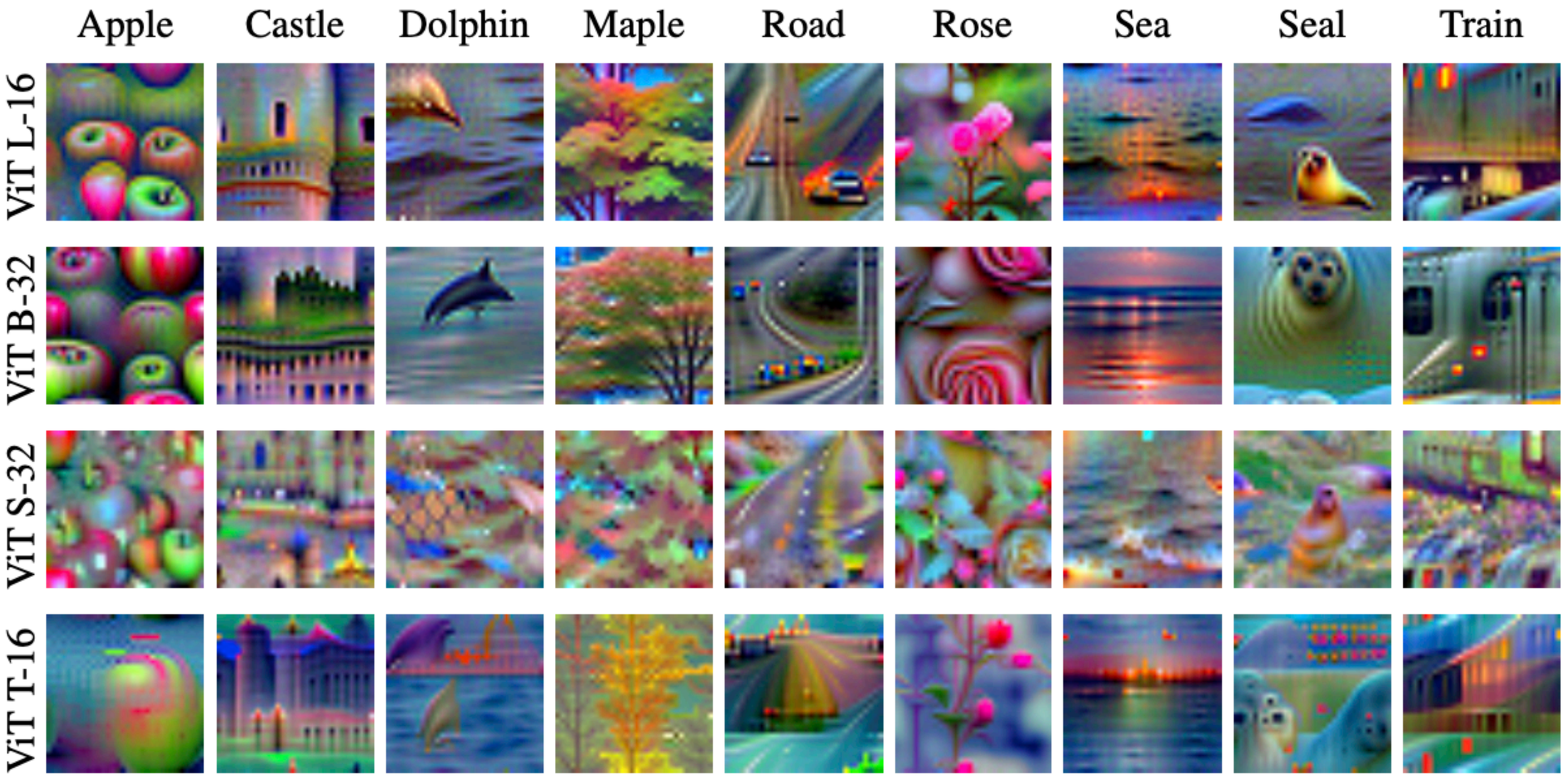





Various Architectures



CIFAR-100


Apple
Castle
Dolphin
Maple
Road
Rose
Sea
Seal
Train
ViT L-16
ViT B-32
ViT S-32
ViT T-16

Thanks for Watching

Amin Ghiasi

Hamid Kazemi

Steven Reich

Chen Zhu

Micah Goldblum

Tom Goldstein