SK hynix i-TAP
반도체 Data Scientist를 위한
ML/DL 심화 커리큘럼

4강 Data Augmentation

Ernest K. Ryu (류경석)

2020.11.27

Question follow-ups

- Why avgpool in DenseNet? Nobody knows…

12/2 (수) Map Pattern Image 논문/적용 사례 공유
- Data augmentation with GAN
- Neural Network Inference with Missing Data
- Deep Transformation-Invariant Clustering
- Multi-GPU
- Model selection / Attributions

향후 topic: R-CNN? Object detection? Knowledge distillation?

4강. Data Augmentation

- Review: ResNet, DenseNet, Saliency map
- Pattern Synthesis

Saliency Maps

PyTorch code.

Weakly supervised object localization:

- Saliency map tells us where object is.
- Weak supervision since training used class labels but not object location labels.

Simonyan, Vedaldi, Zisserman, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, ICLR, 2014.

Texture/Pattern Synthesis

Consider convolutional layer $l = 1, \ldots, L$. Each convolutional layer has N_l features and $M_l = \text{width}_l \times \text{height}_l$ spatial positions. Layer l has feature maps

$$F^l \in \mathbb{R}^{N_l \times M_l}$$

Form the Gram matrix

$$G^l = F^l {F^l}^T$$

Gram matrix captures texture information.

Texture/Pattern Synthesis

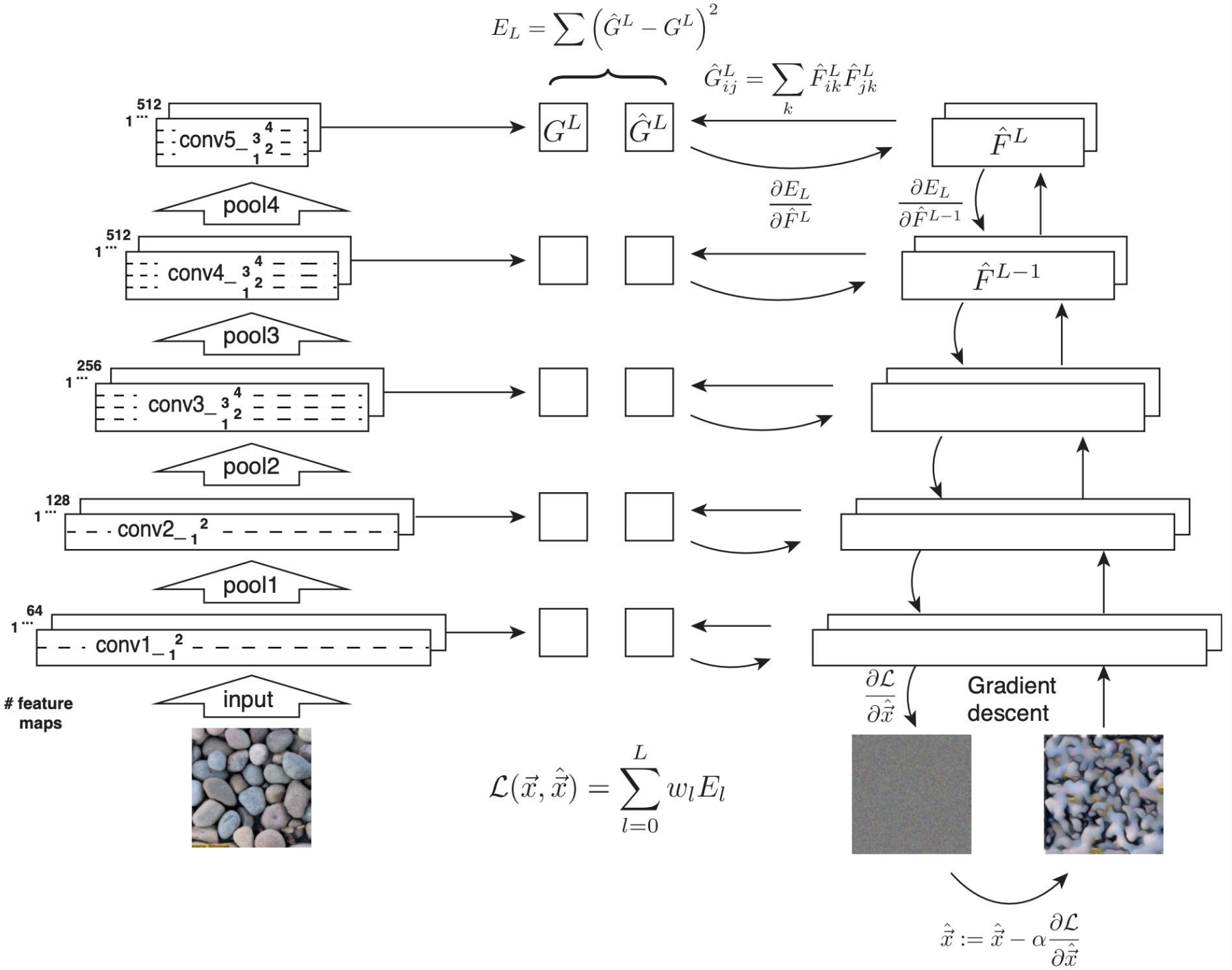
Consider input image x and $\hat{x}$ with gram matrices G^l and $\hat{G}^l$ for $l = 1, \ldots, L$.

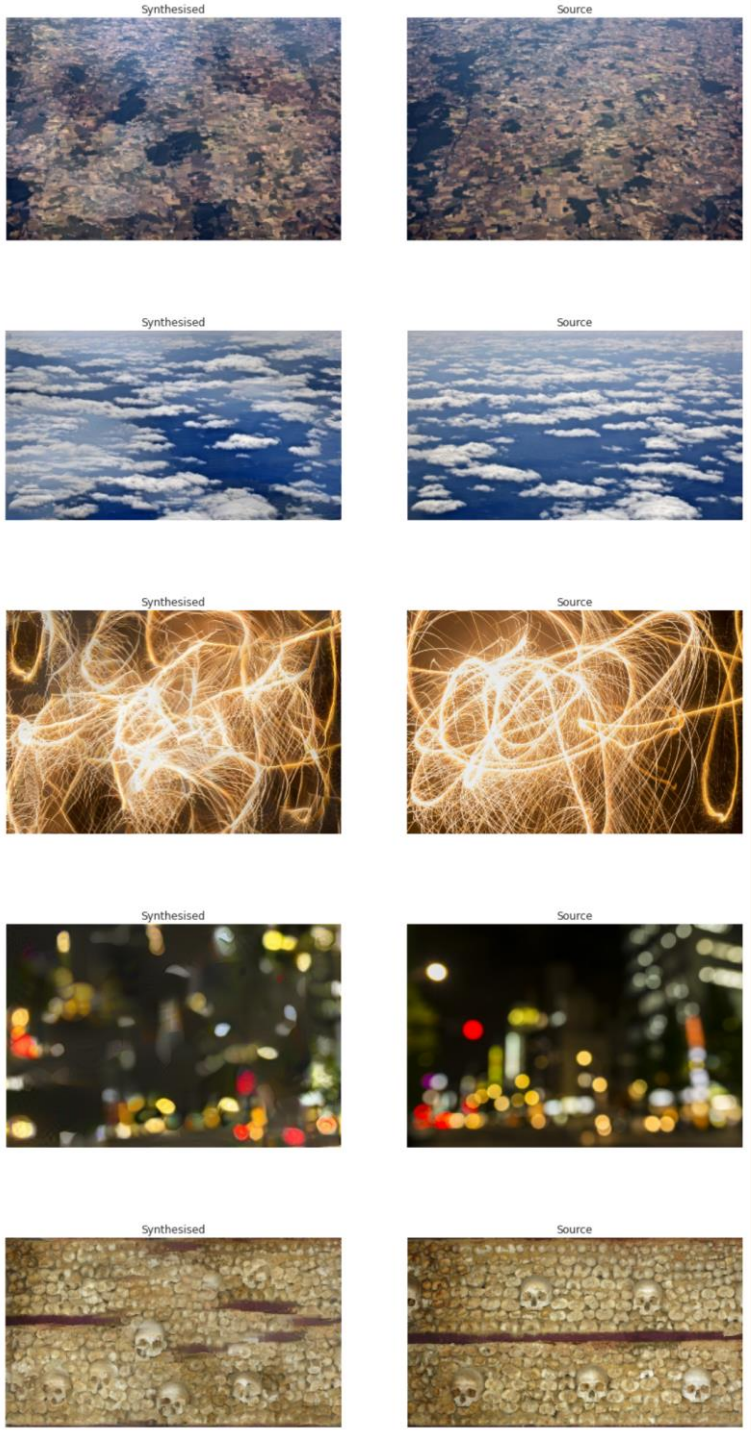
Loss function is

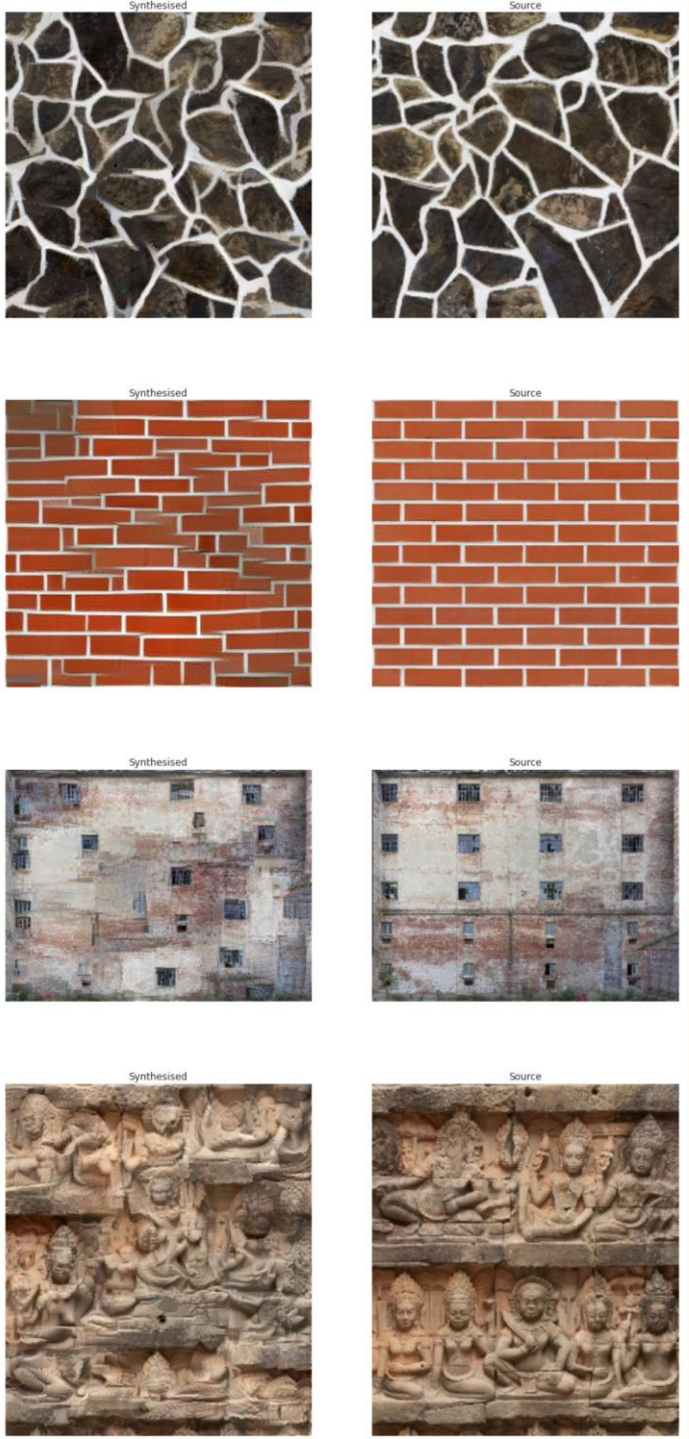
$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G_{ij}^l - \hat{G}_{ij}^l\right)^2$$

$$L_{\text{style}}(x, \hat{x}) = \sum_{l=1}^{L} E_l$$

Given image x, initialize $\hat{x}$ randomly and construct $\hat{x}$ by backpropagating on $L_{\text{style}}(x, \hat{x})$




Synthesised
Source
Synthesised
Source
Synthesised
Source
Synthesised
Source
Synthesised
Source


Synthesised
Source
Synthesised
Source
Synthesised
Source
Synthesised
Source

Why Gram Matrix = Texture?

$$\left(G^l\right)_{ij} = \sum_{k=1}^{M_l} F_{ik}^l F_{jk}^l$$

Represents the covariance without means removed (the 2nd moments) of filters i and j averaged over spatial coordinate k. (Location information averaged out.)

A texture or pattern is spatially invariant, while the content of the image (boundaries of an object) involves more spatial formation.

Neural Style Transfer (NST)

NST = pattern synthesis + model inversion

Consider the content loss

$$L_{\text{content}}(x, \hat{x}) = \frac{1}{2}\sum_{ij}(F_{ij}^{m} - P_{ij}^{m})$$

Consider the loss

$$L(x, \hat{x}) = \alpha L_{\text{style}} + \beta L_{\text{content}}$$

$$E_L = \sum \left(G^L - A^L\right)^2$$

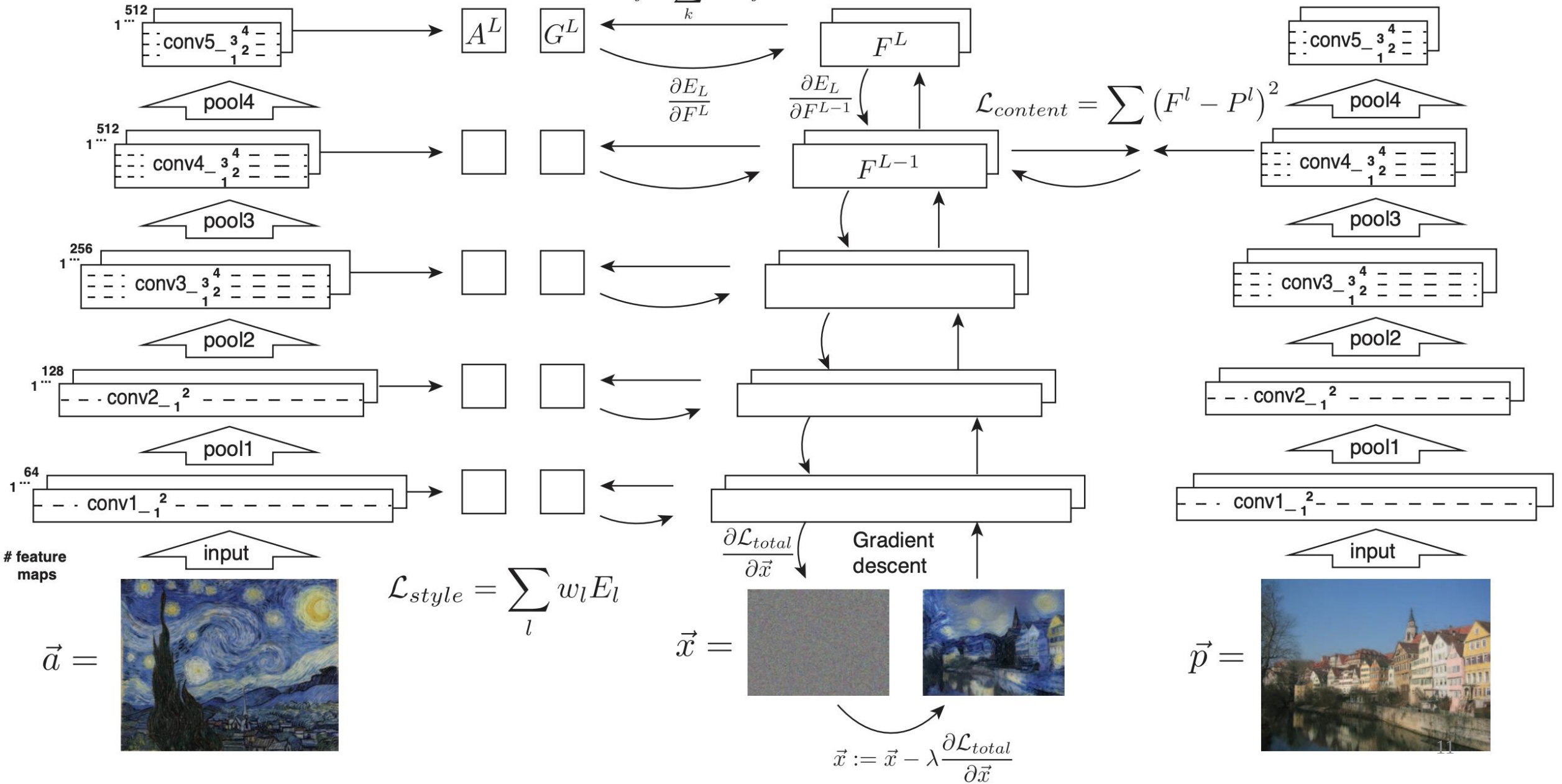
$$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style}$$

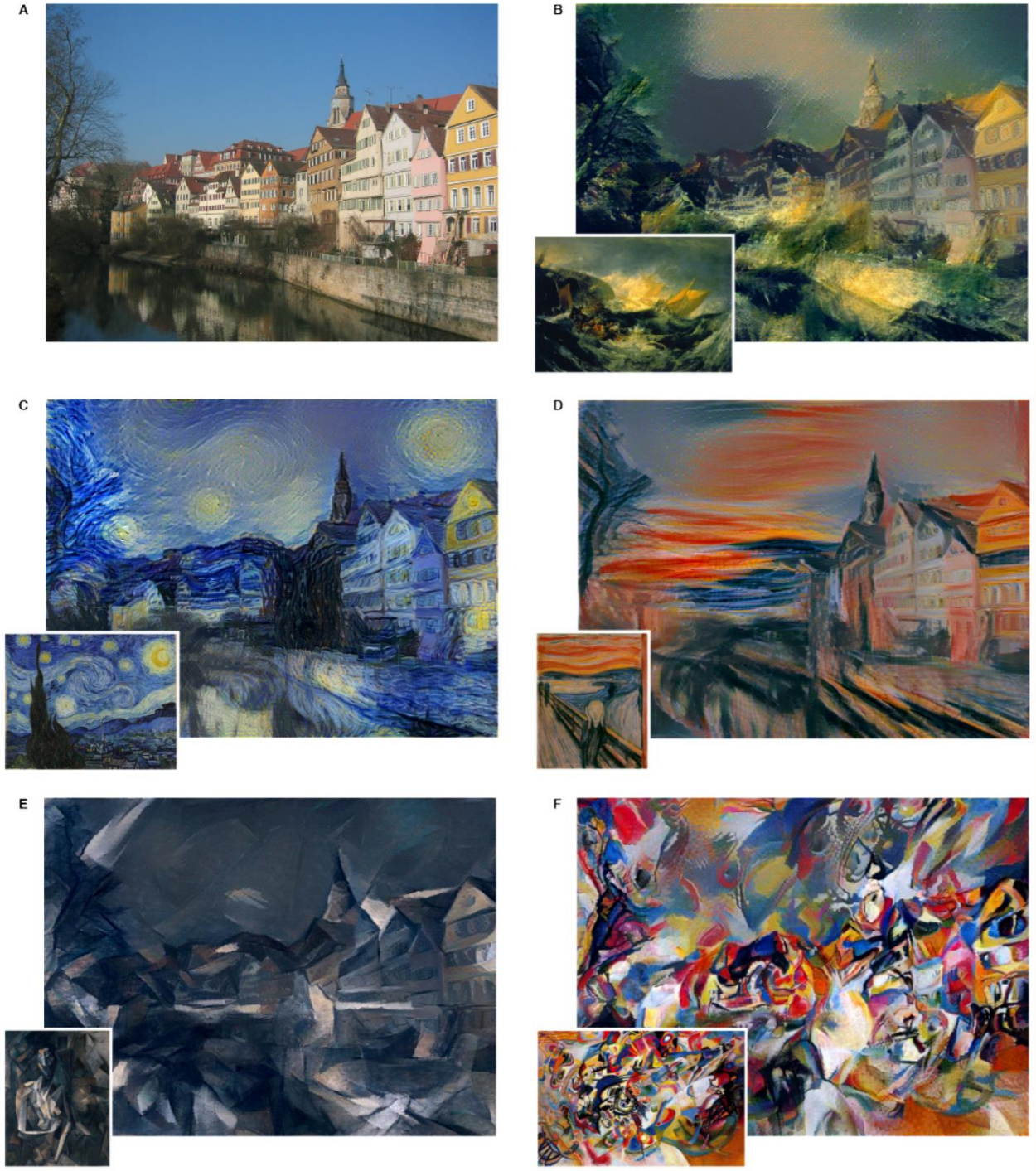
$$G^L_{ij} = \sum_k F^L_{ik} F^L_{jk}.$$

$$\mathcal{L}_{content} = \sum \left(F^l - P^l\right)^2$$

$$\mathcal{L}_{style} = \sum_l w_l E_l$$

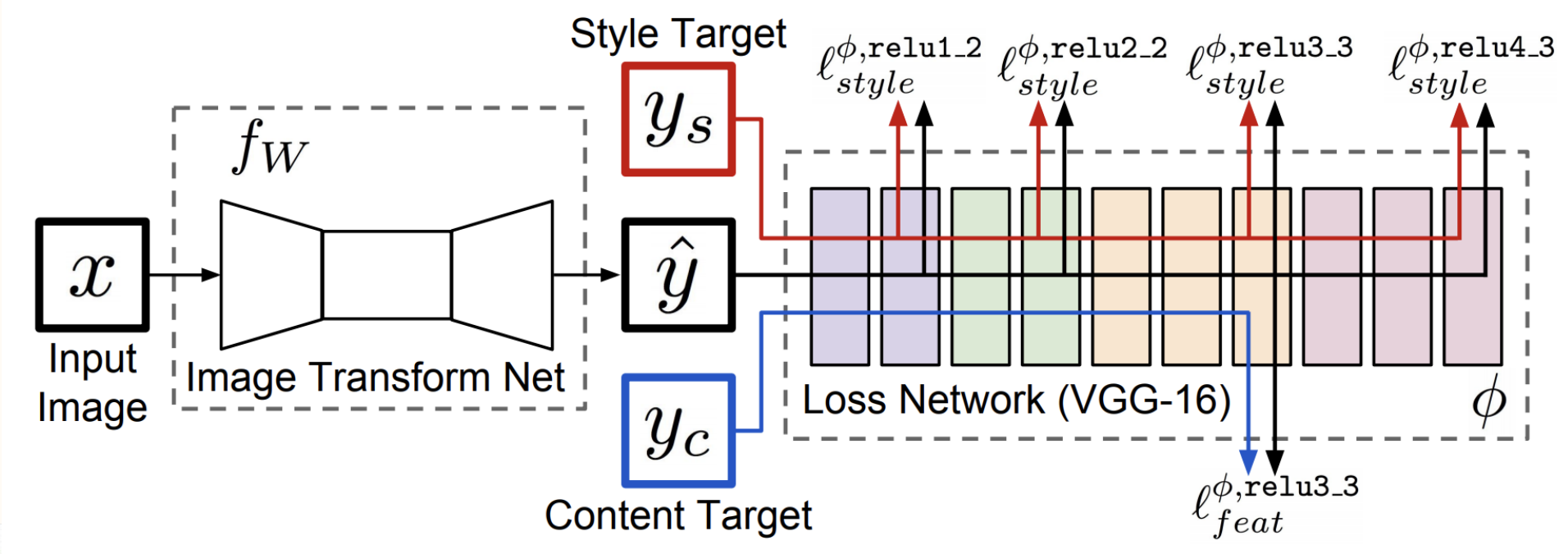
$$\vec{x} := \vec{x} - \lambda \frac{\partial \mathcal{L}_{total}}{\partial \vec{x}}$$




A
B
C
D
E
F

Faster Neural Style Transfer

- Obtaining images via optimization is slow.
- Train a neural network that performs the job of the optimization network.



Data Augmentation

Basic geometric transformations

- Flipping
- Color change
- Cropping
- Rotation
- Translation
- Noise Injection

Style Augmentation

Motivation: classification networks heavily rely on texture, rather than shapes in performing classification.

Key idea: perform (fast) neural style transfer on training images.

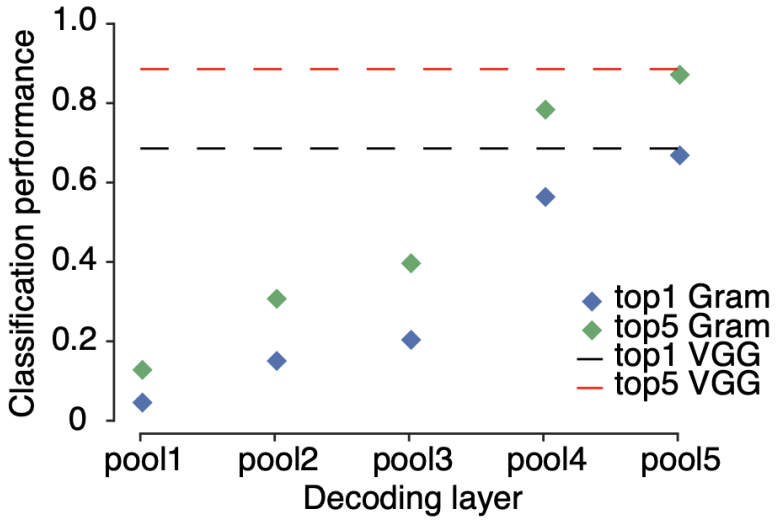


Figure 4: Performance of a linear classifier on top of the texture representations in different layers in classifying objects from the ImageNet dataset. High-level information is made increasingly explicit along the hierarchy of our texture model.

Style Augmentation

(a) Texture image

81.4% **Indian elephant**
10.3% indri
8.2% black swan

(b) Content image

71.1% **tabby cat**
17.3% grey fox
3.3% Siamese cat

(c) Texture-shape cue conflict

63.9% **Indian elephant**
26.4% indri
9.6% black swan

Figure 1: Classification of a standard ResNet-50 of **(a)** a texture image (elephant skin: only texture cues); **(b)** a normal image of a cat (with both shape and texture cues), and **(c)** an image with a texture-shape cue conflict, generated by style transfer between the first two images.

Geirhos, Rubisch, Michaelis, Bethge, Wichmann, Brendel, ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, ICLR, 2019.

Stylized ImageNet

Figure 3: Visualisation of Stylized-ImageNet (SIN), created by applying AdaIN style transfer to ImageNet images. Left: randomly selected ImageNet image of class `ring-tailed lemur`. Right: ten examples of images with content/shape of left image and style/texture from different paintings. After applying AdaIN style transfer, local texture cues are no longer highly predictive of the target class, while the global shape tends to be retained. Note that within SIN, every source image is stylized only once.

Geirhos, Rubisch, Michaelis, Bethge, Wichmann, Brendel, ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness, ICLR, 2019.

Style Randomization

Figure 1: Style augmentation applied to an image from the Office dataset [24] (original in top left). Shape is preserved but the style, including texture, color and contrast are randomized.

Style Randomization

Figure 7: Examples of input monocular synthetic images post style augmentation.

Jackson, Atapour-Abarghouei, Bonner, Breckon, Obara, Style Augmentation: Data Augmentation via Style Randomization, CVPR, 2019.

Label Smoothing

Key idea: Given a label y_i represented as a one-hot vector, instead use

$$(1-\varepsilon)y_i + \frac{\varepsilon}{K}$$

$\varepsilon = 0.1$ is standard.

Prevent network from becoming over-confident.

Widely used, but poorly understood theoretically and empirically.

Szegedy, Vanhoucke, Ioffe, Shlens, and Wojna. Rethinking the inception architecture for computer vision, CVPR, 2016.

Label Smoothing: Effects

- Improves generalization.
- Improves model calibration.
- Worsens knowledge distillation into a student network.

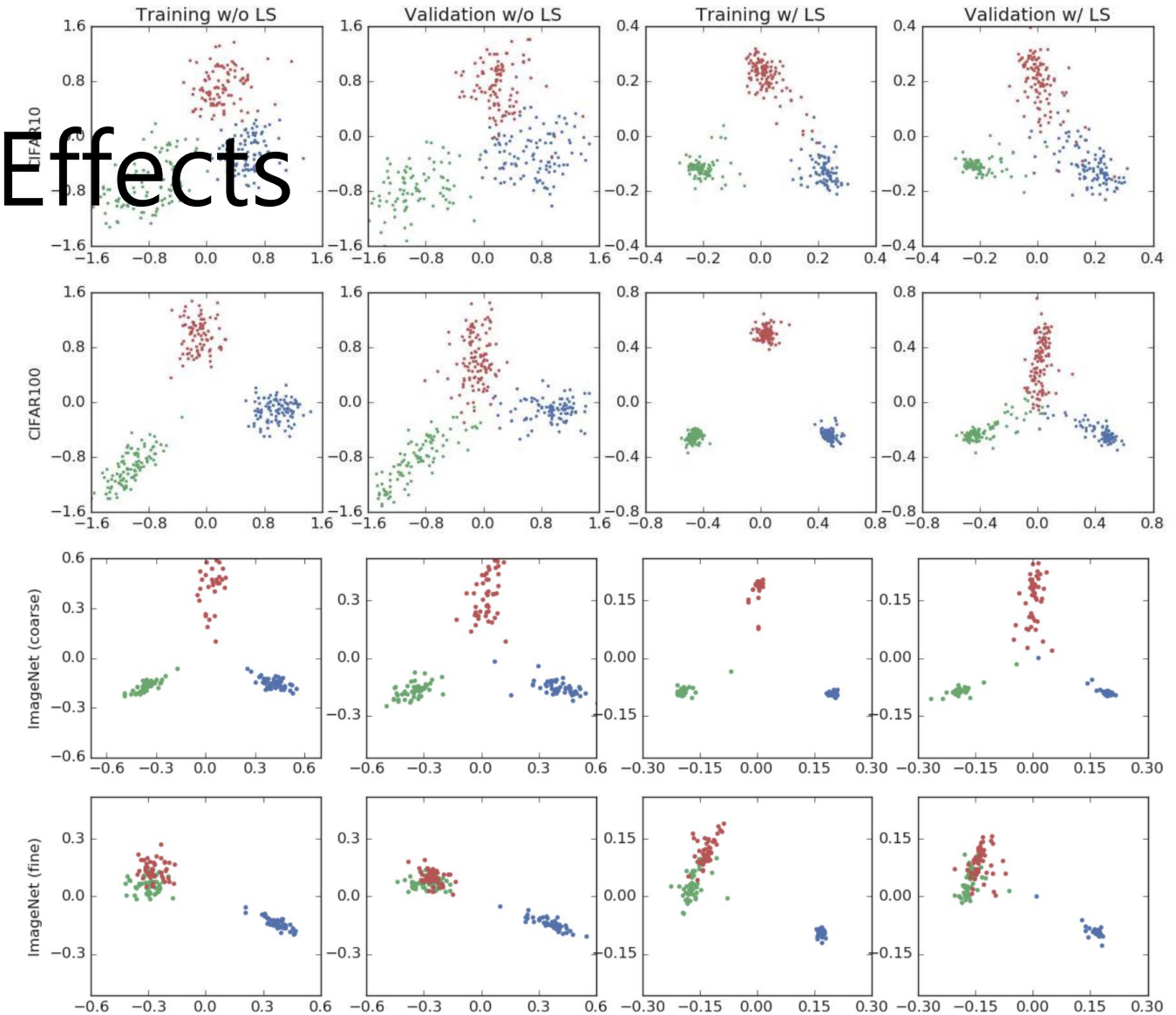


Figure 1: Visualization of penultimate layer's activations of: AlexNet/CIFAR-10 (first row), CIFAR-100/ResNet-56 (second row) and ImageNet/Inception-v4 with three semantically different classes (third row) and two semantically similar classes plus a third one (fourth row).

CutOut

- Key idea: Randomly remove parts of an image.
- Effect: Forces neural network to learn from all parts of the image.

DeVries, Taylor, Improved Regularization of Convolutional Neural Networks with Cutout, arXiv, 2017.

CutOut

Similar ideas pursued in other papers:

- Attention-based Dropout Layer for Weakly Supervised Object Localization Junsuk Choe, Hyunjung Shim, CVPR, 2019.
- DropBlock: A regularization method for convolutional networks, Golnaz Ghiasi, Tsung-Yi Lin, Quoc V. Le, NeurIPS, 2018.
- Hide-and-Seek: Forcing a Network to be Meticulous for Weakly-supervised Object and Action Localization, Krishna Kumar Singh, Yong Jae Lee, ICCV, 2017.
- Random Erasing Data Augmentation, Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, Yi Yang, AAAI, 2020.

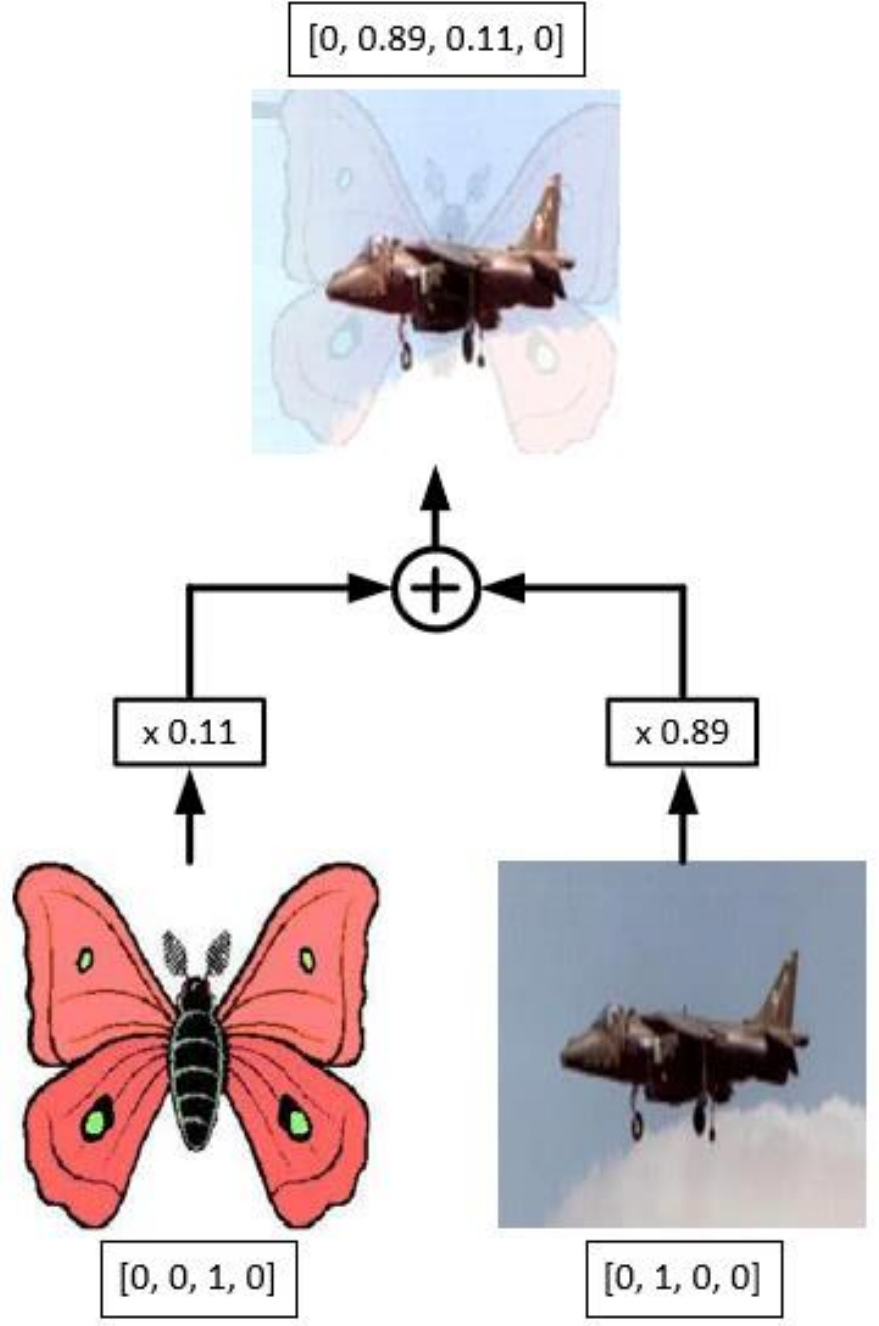
Mixup

Let X_1 and X_2 be datapoints and y_1 and y_2 be one-hot labels.

Then we form

$$\tilde{X} = \lambda X_1 + (1 - \lambda) X_2$$
$$\tilde{y} = \lambda y_1 + (1 - \lambda) y_2$$

Choose $\lambda \sim \beta(\alpha, \alpha)$

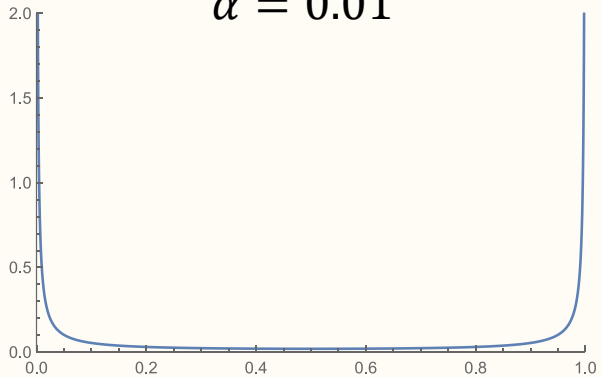


$\beta(\alpha, \alpha)$-distribution

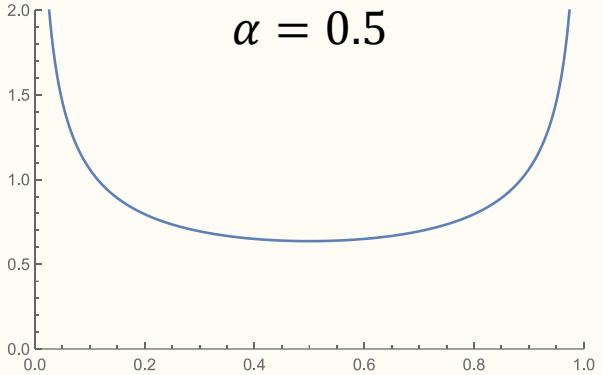
Density:
$f(x) \propto x^{\alpha-1}(1-x)^{\alpha-1}$

$\alpha \approx 0$ corresponds to $\{0,1\}$ random variable. $\alpha = 1$ corresponds to uniform on $[0,1]$.

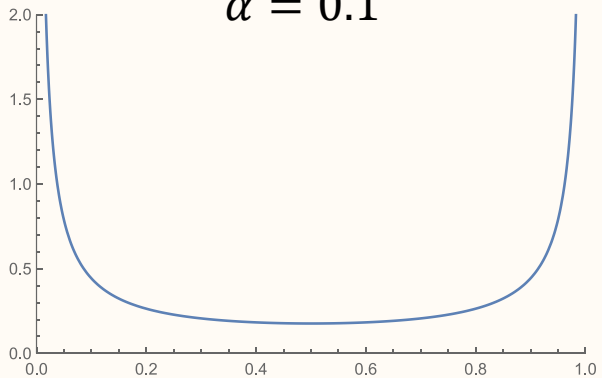
$\alpha = 0.01$



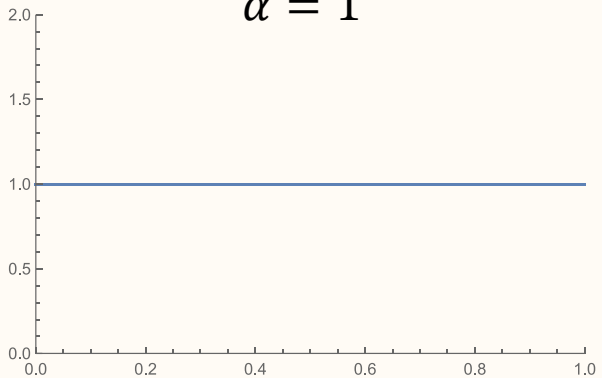
$\alpha = 0.5$



$\alpha = 0.1$

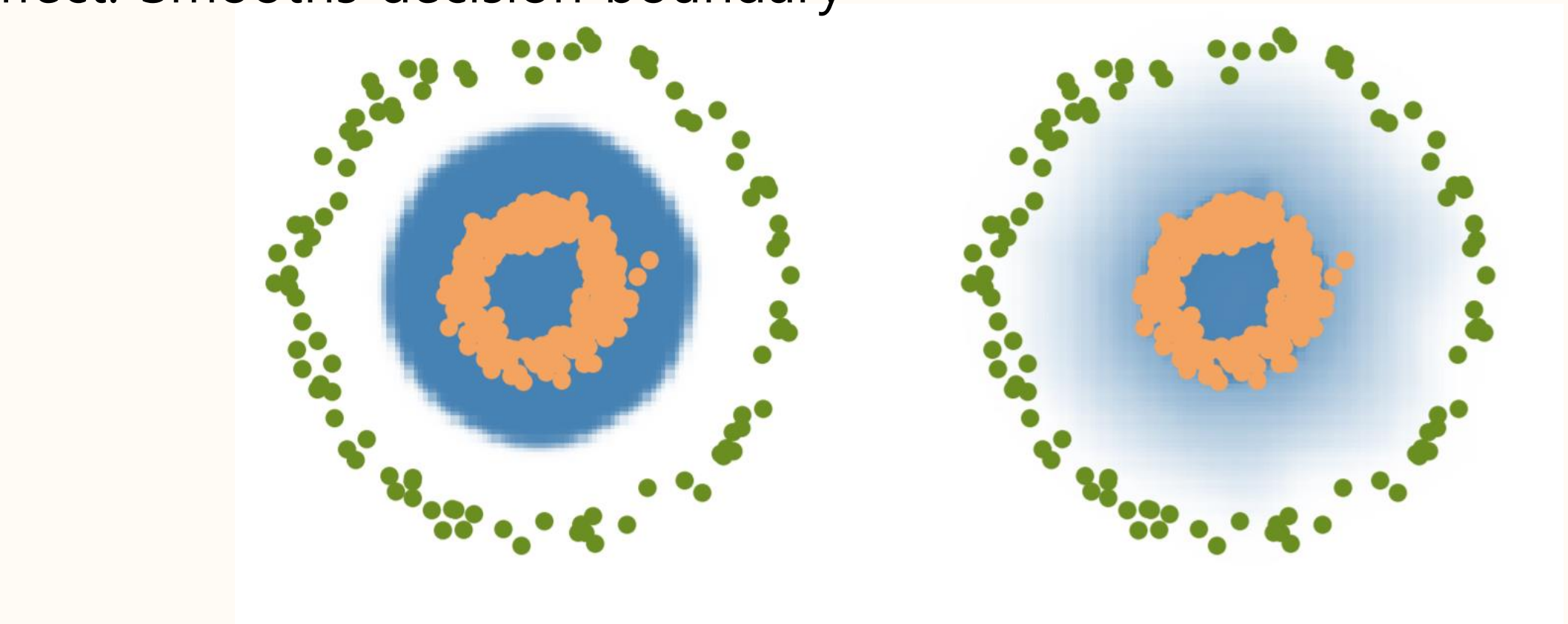


$\alpha = 1$



Mixup

Effect: Smooths decision boundary



Mixup Other Ideas

- Using 3 or more datapoints did not improve affect.
- Interpolating only data without interpolating labels did not help.

Cutmix

Key idea: Place an image in cutout region

Original | Mixup | CutOut | CutMix

Yun, Han, Oh, Chun, Choe, Yoo, CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features, ICCV, 2019.

Puzzle Mix

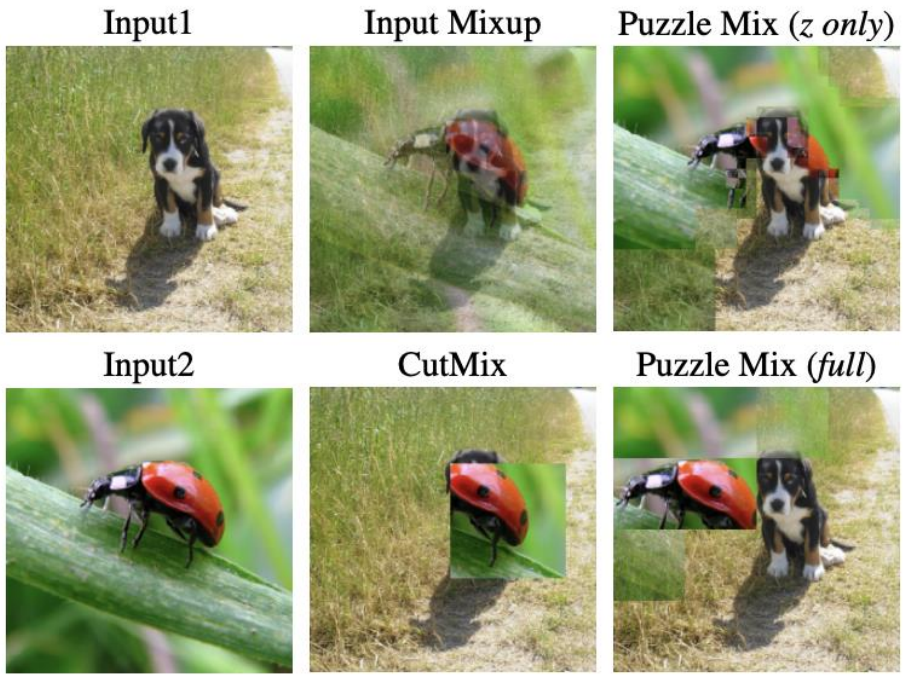
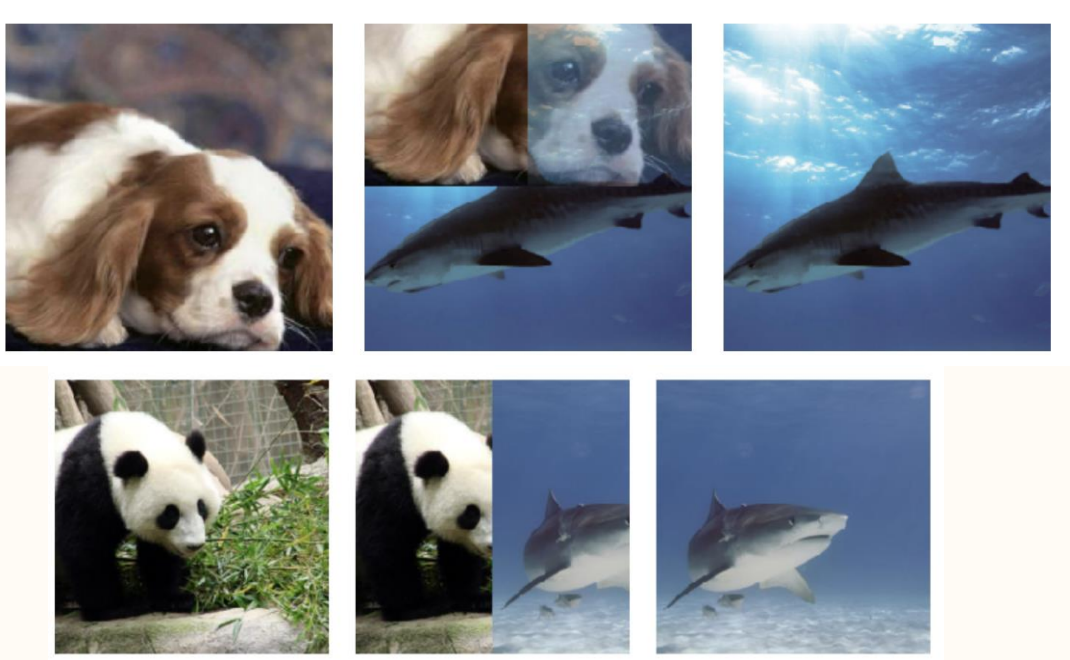


Figure 1. A visual comparison of the mixup methods. Puzzle Mix ensures to contain sufficient saliency information while preserving the local statistics of each input.

Puzzle Mix

- Use saliency map to detect where the objects are.
- Transport images so that the objects don't overlap (do this by solving a combinatorial optimizaiton problem)
- Mix images



Test-Time Data Augmentation

- Do you augment data during test time? Yes!

- AlexNet paper: "At test time, the network makes a prediction by extracting five 224 × 224 patches (the four corner patches and the center patch) as well as their horizontal reflections (hence ten patches in all)"
- ResNet paper: "we adopt the standard 10-crop testing"

- Key idea: augment data, make multiple predictions, and average class scores.

Test Time Augmentation: Data Distillation

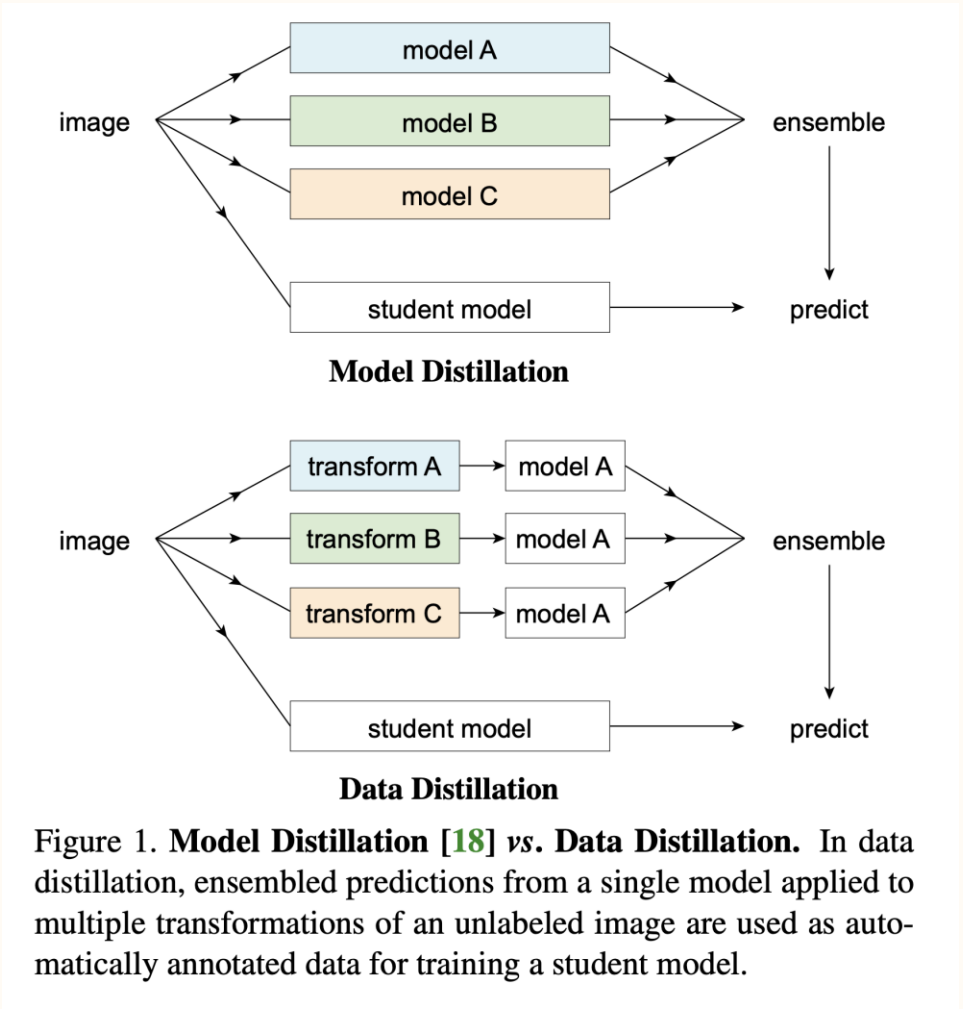


Figure 1. **Model Distillation [18] *vs*. Data Distillation.** In data distillation, ensembled predictions from a single model applied to multiple transformations of an unlabeled image are used as automatically annotated data for training a student model.

Radosavovic, Dollar, Girshick, Gkioxari, He, Data Distillation: Towards Omni-Supervised Learning, CVPR, 2017.