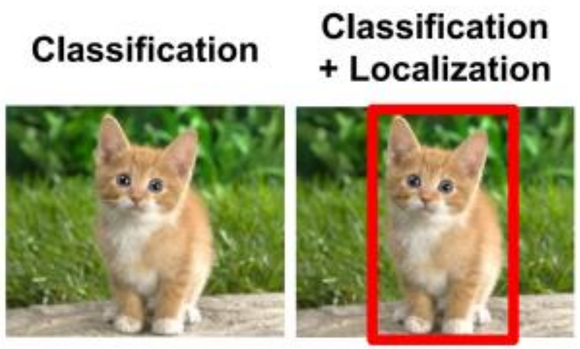
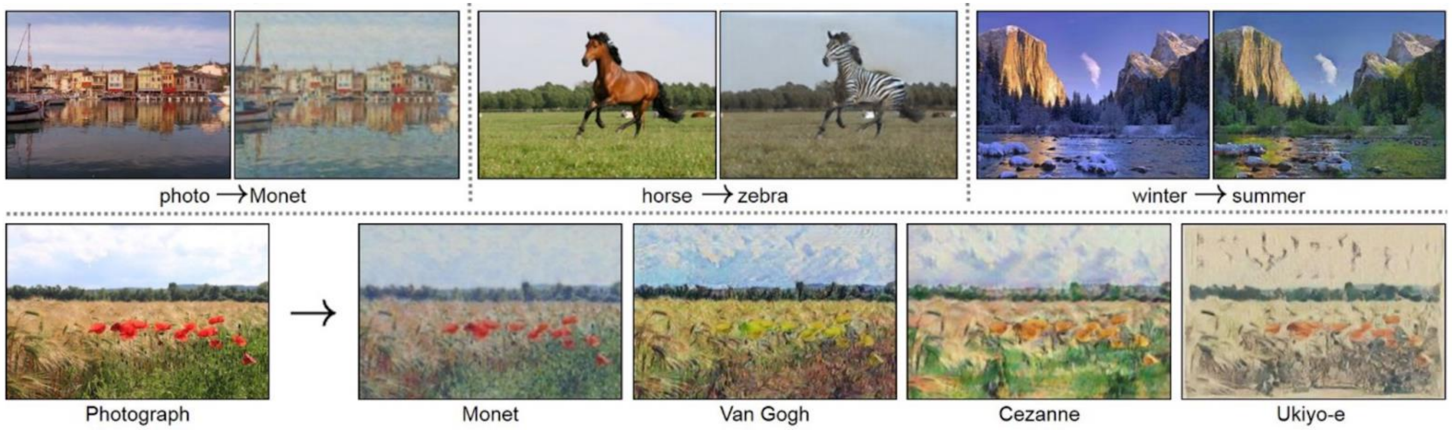
Knowledge Distillation of Convolutional Neural Networks

Federico Perazzi
Head of AI
Bending Spoons

Learning Complex Representation

- Neural networks are very effective at learning complex representation from data.
- Successful at **discriminative** tasks such as **classification, detection, segmentation**.
- Successful at **generative** tasks such as **image translation.**

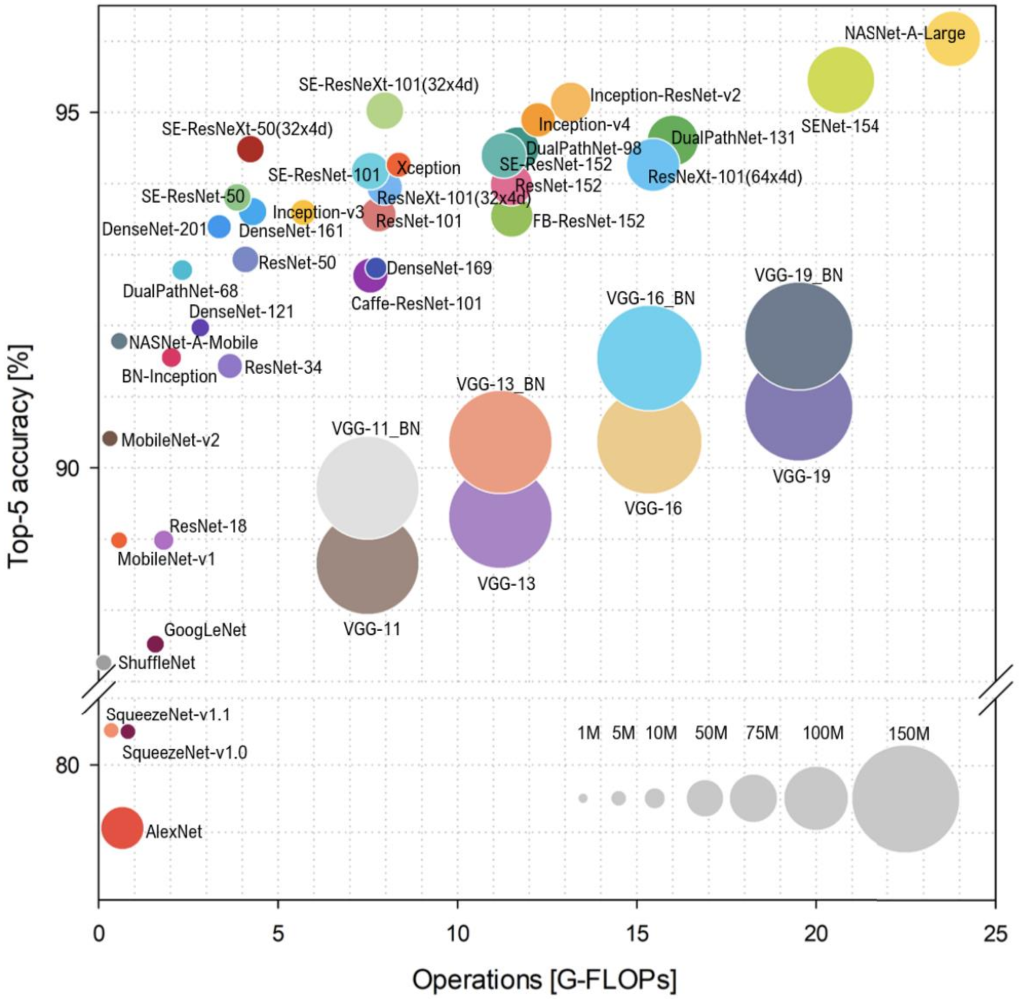




Source: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

The Learning Capabilities Come at a Price….

- The complexity and the size of the model often translate to better performance.



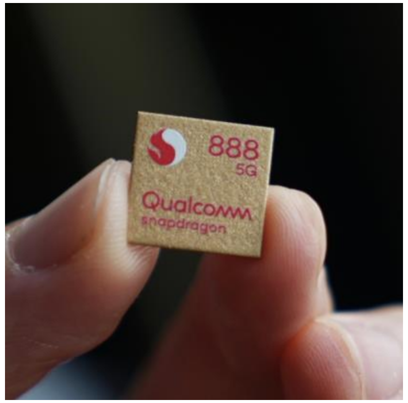
The Learning Capabilities Comes at a Price….

Neural Networks

- Consume a lot of memory and power.
- Significant computational costs.

→ **Must be optimized for deployment**

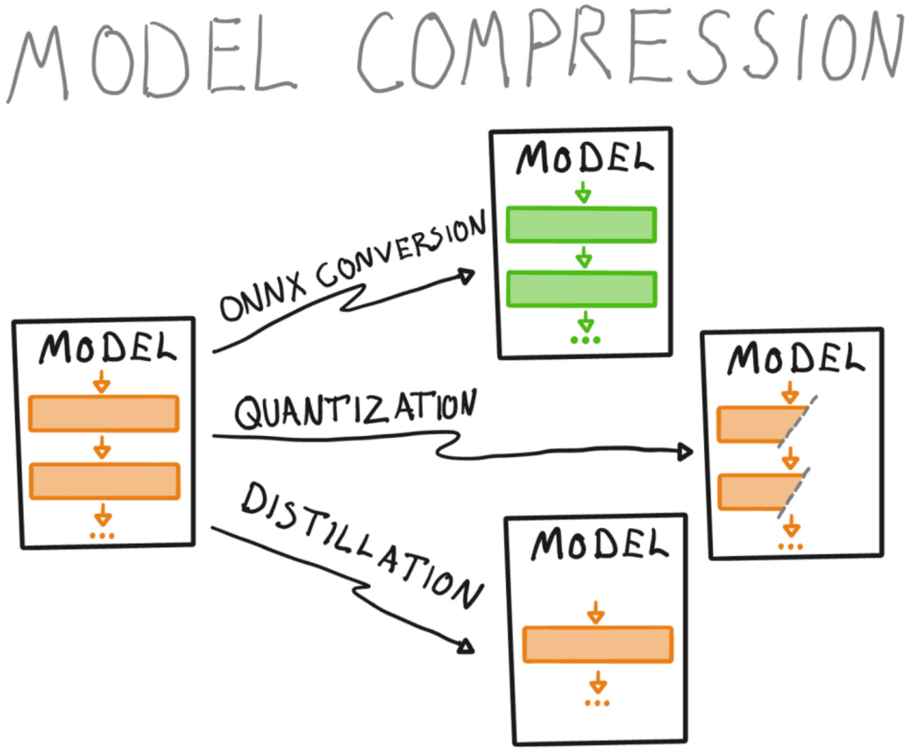
- **Cloud Computing** → minimize the infrastructure costs and the carbon footprint.
- **Edge Devices** → computational demand must not exceed strict hardware limitations.



Model Compression

- Set of techniques to reduce the complexity of a neural networks → while minimizing the loss in accuracy
- Quantization
- Weights Pruning
- Knowledge Distillation
- Conversion to high performance libraries

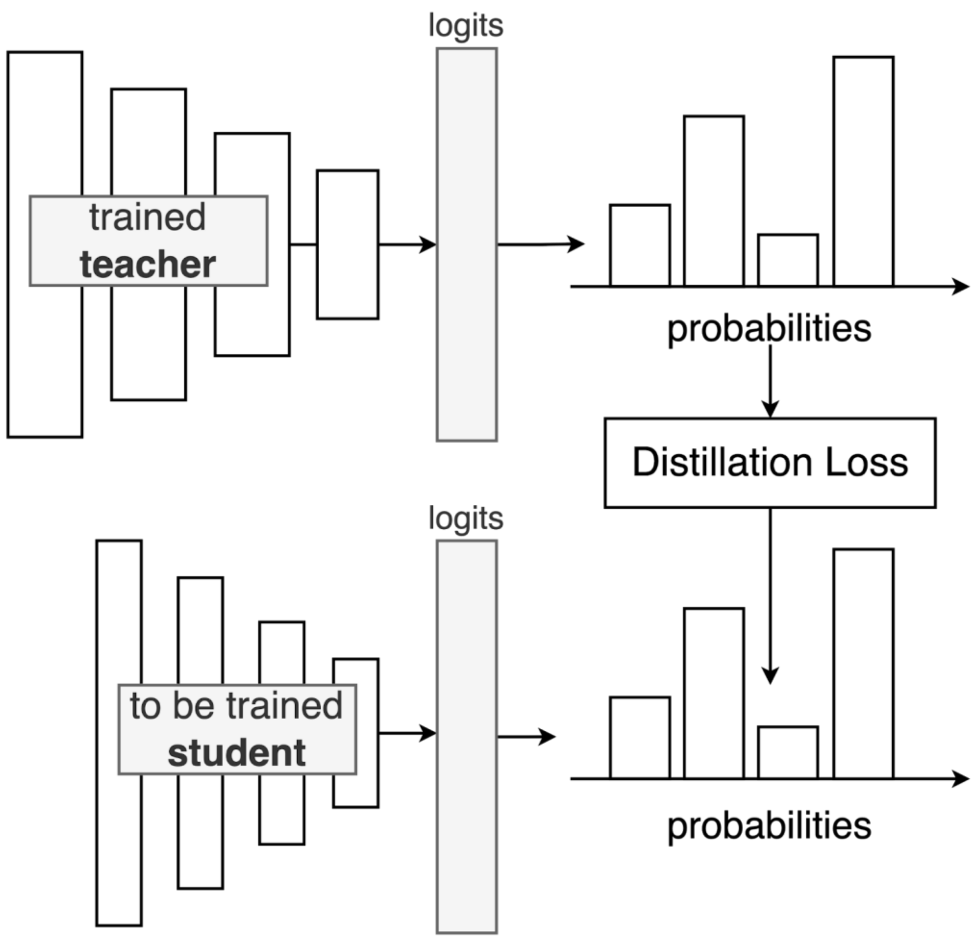
→ 50x speed-up on classification models without loss of quality



Knowledge Distillation

→ The process of transferring knowledge from a large model to a smaller one that is more suitable for deployment.

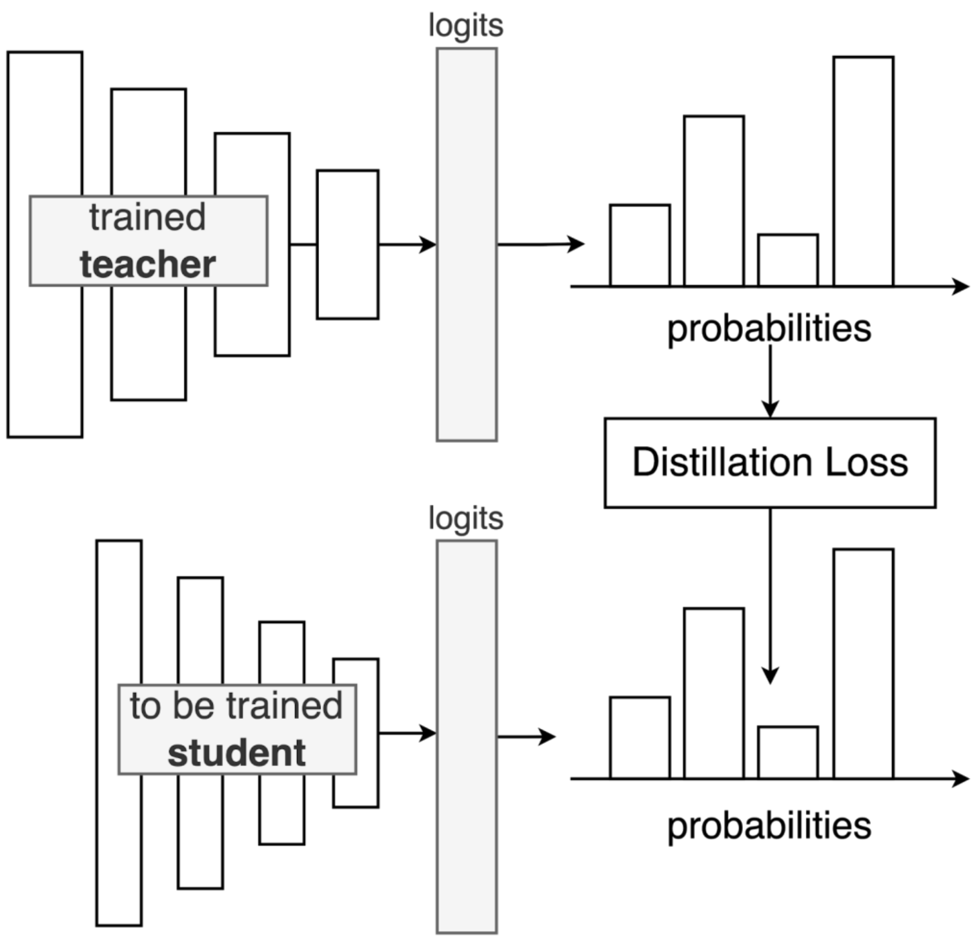
→ Use a fast and compact model to approximate the function learned by a slower, larger, but better performing model.



Steps to Distill a Model

- Prepare the dataset
- Define student and teacher models
- Train the teacher
- Distill teacher to student
- Train student from scratch for comparison

https://keras.io/examples/vision/knowledge_distillation/

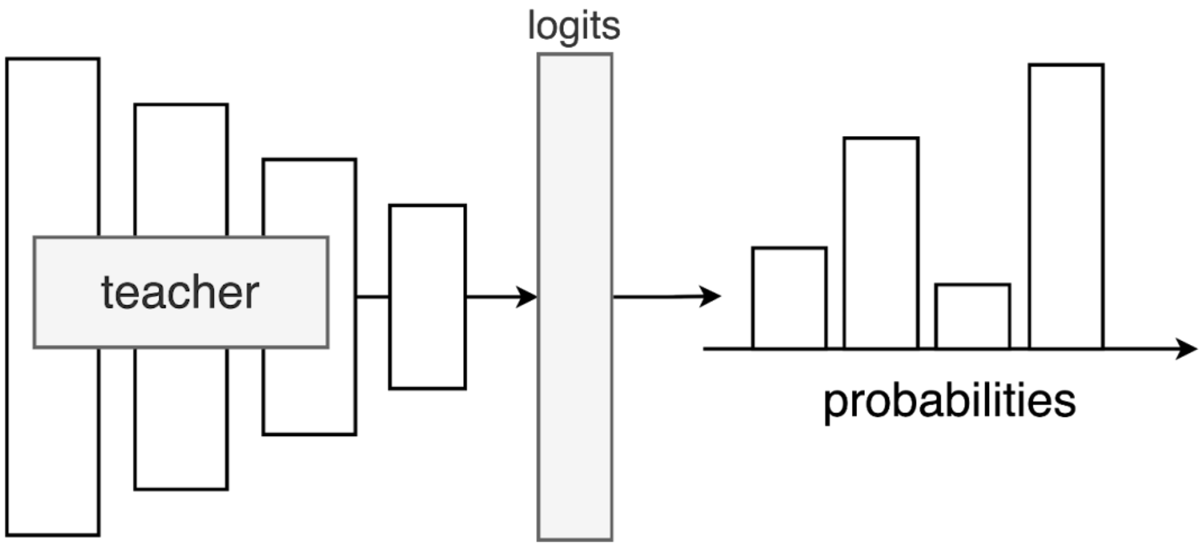


Prepare the Dataset

Classification Dataset

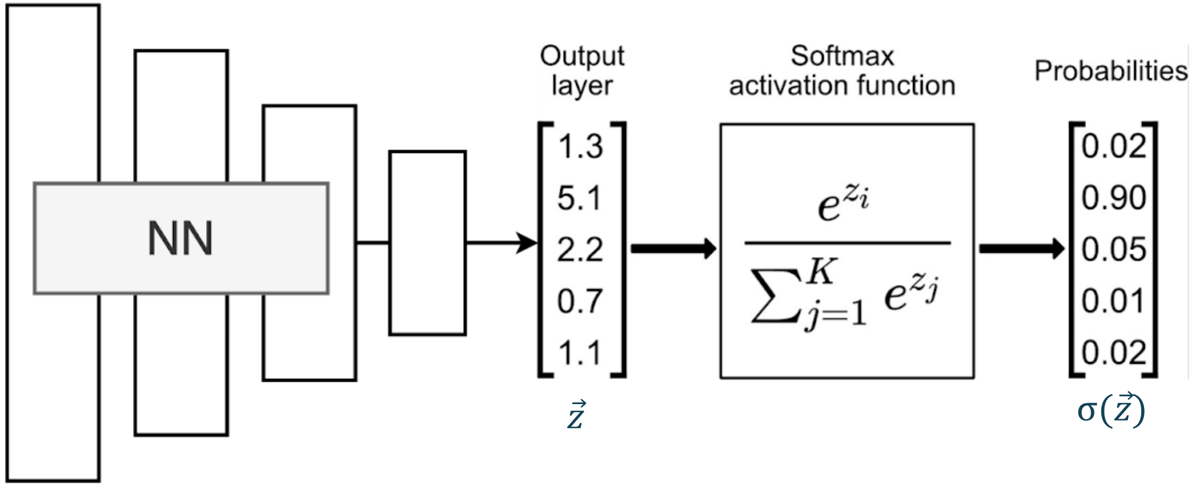
Define the Teacher and Student Model

Classification Dataset



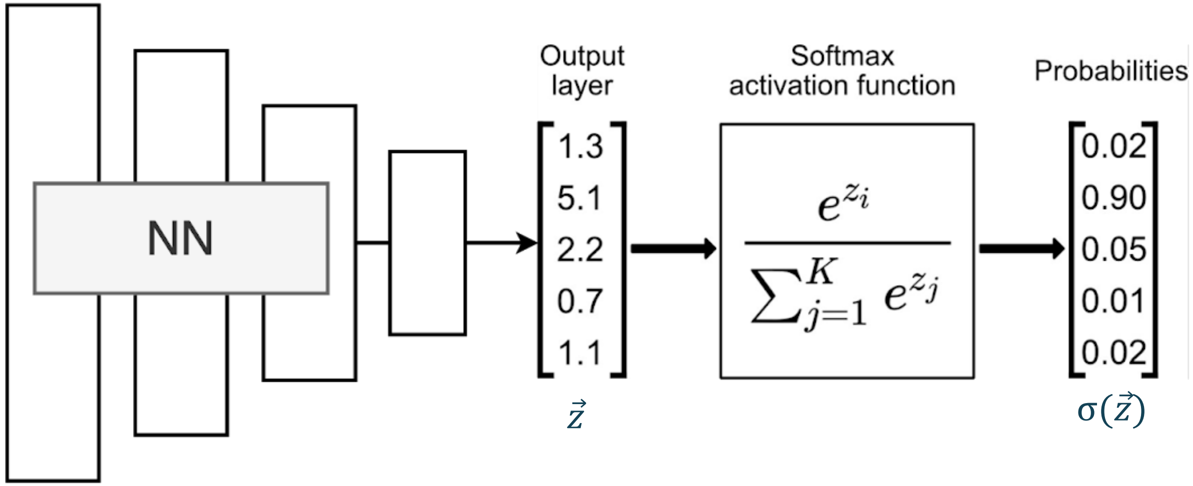
Define the Teacher and Student Model

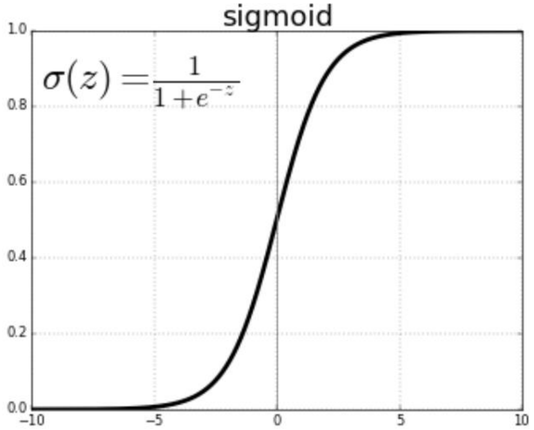
Classification Dataset



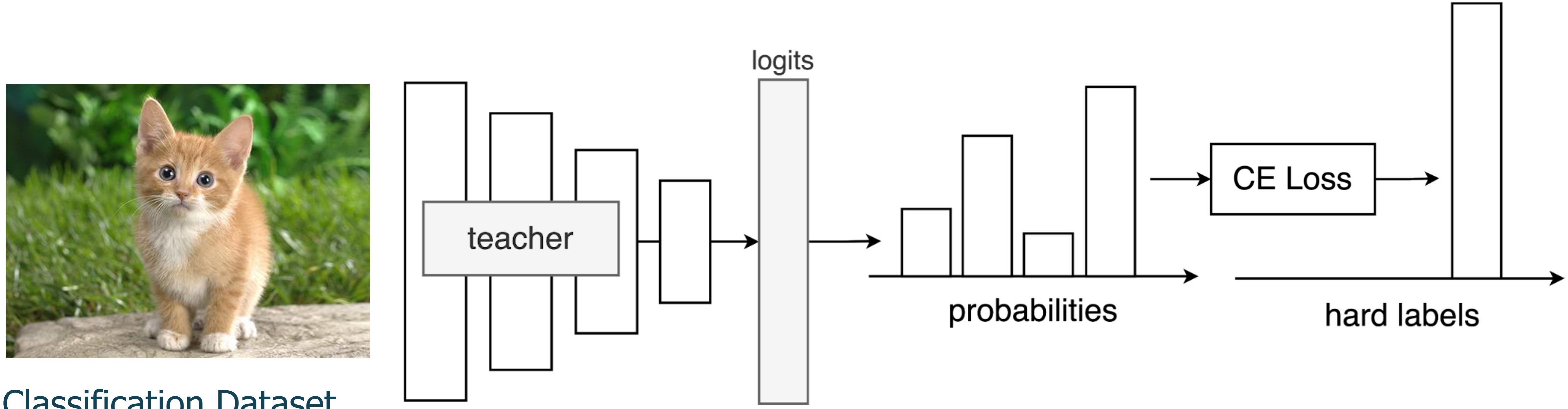
Define the Teacher and Student Nodel

Classification Dataset





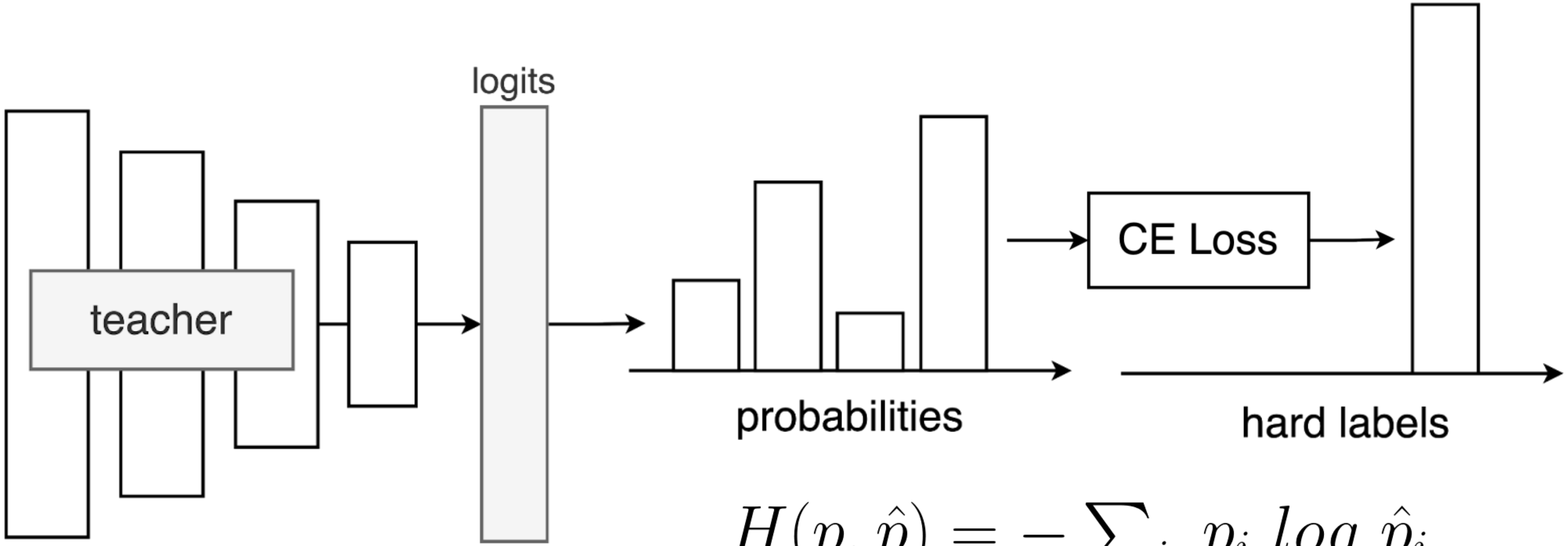
Training the Teacher



Classification Dataset

Training the Teacher

Classification Dataset

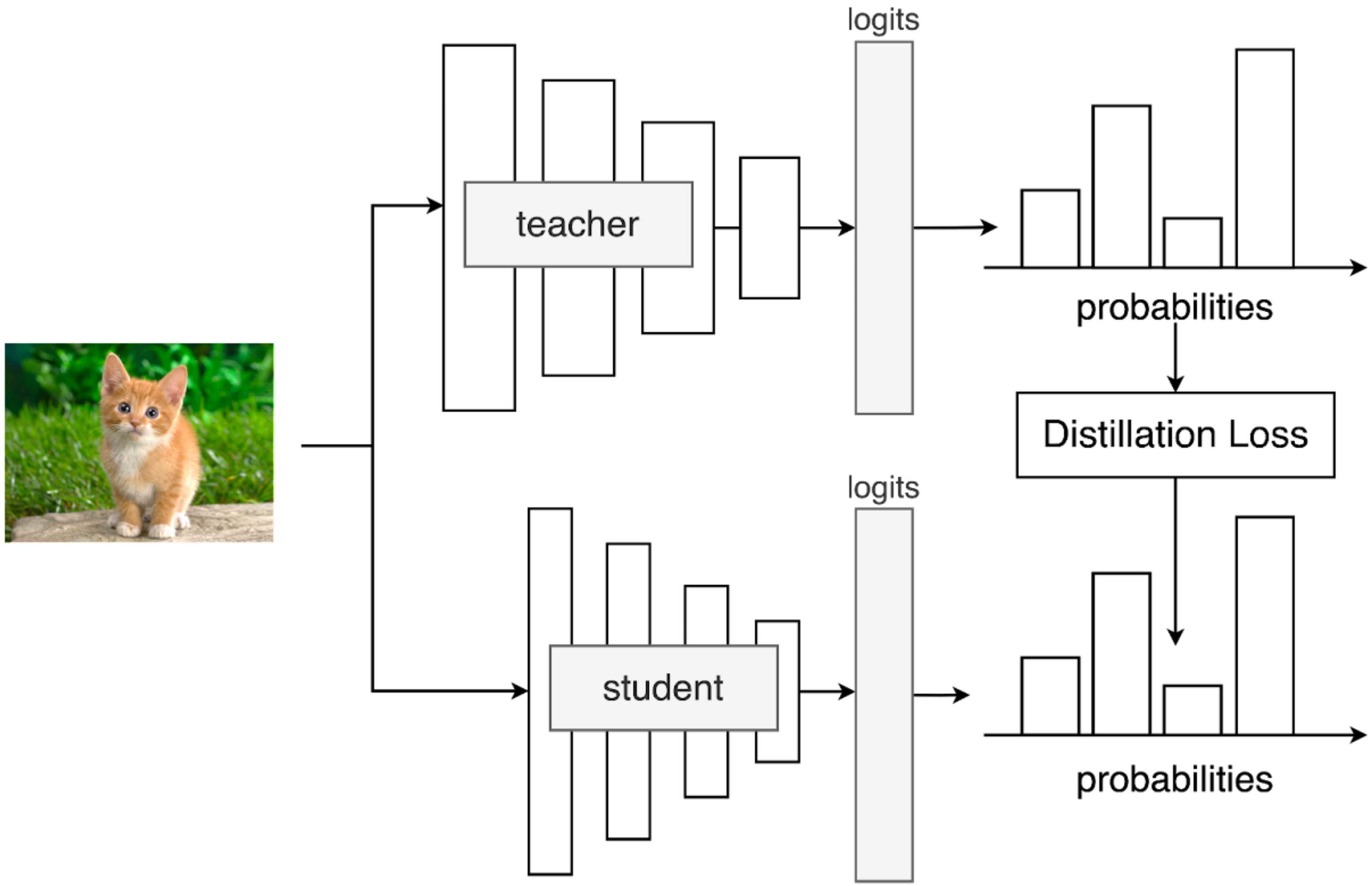


$$H(p, \hat{p}) = -\sum_i p_i \, log \, \hat{p}_i$$

- Cross-entropy loss: $H(p, \hat{p})$
- Ground-truth and predicted probabilities $p, \hat{p}$

BENDING SPOONS

Distill Teacher to Student

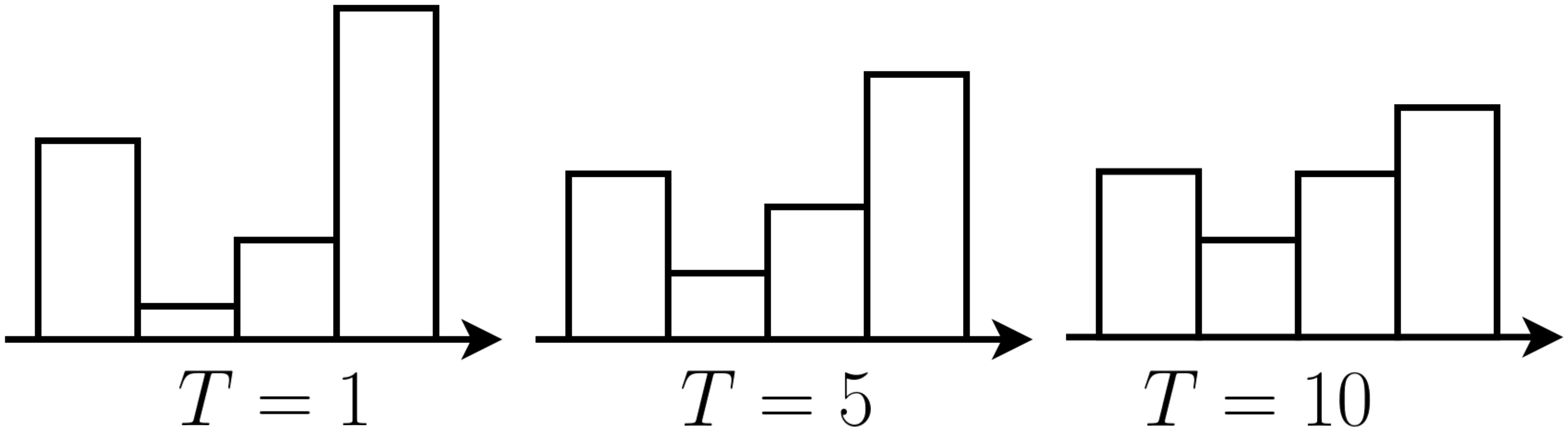


→ **Soft labels are both supervisory signals and regularizers** → serves as a good regularization to the student network

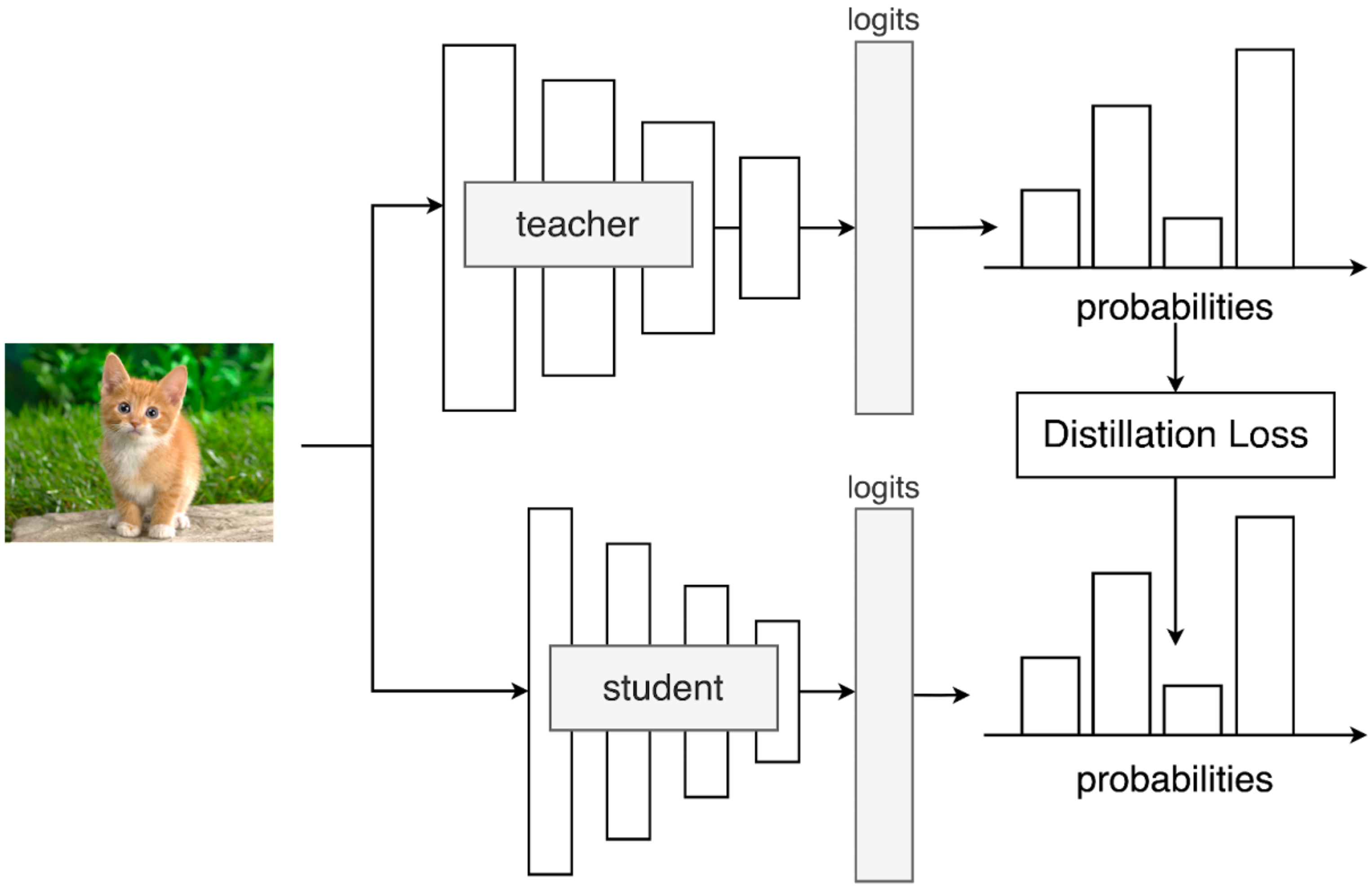
Heated Softmax

- **Higher temperature T**, produces a softer probability distribution over classes.

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$



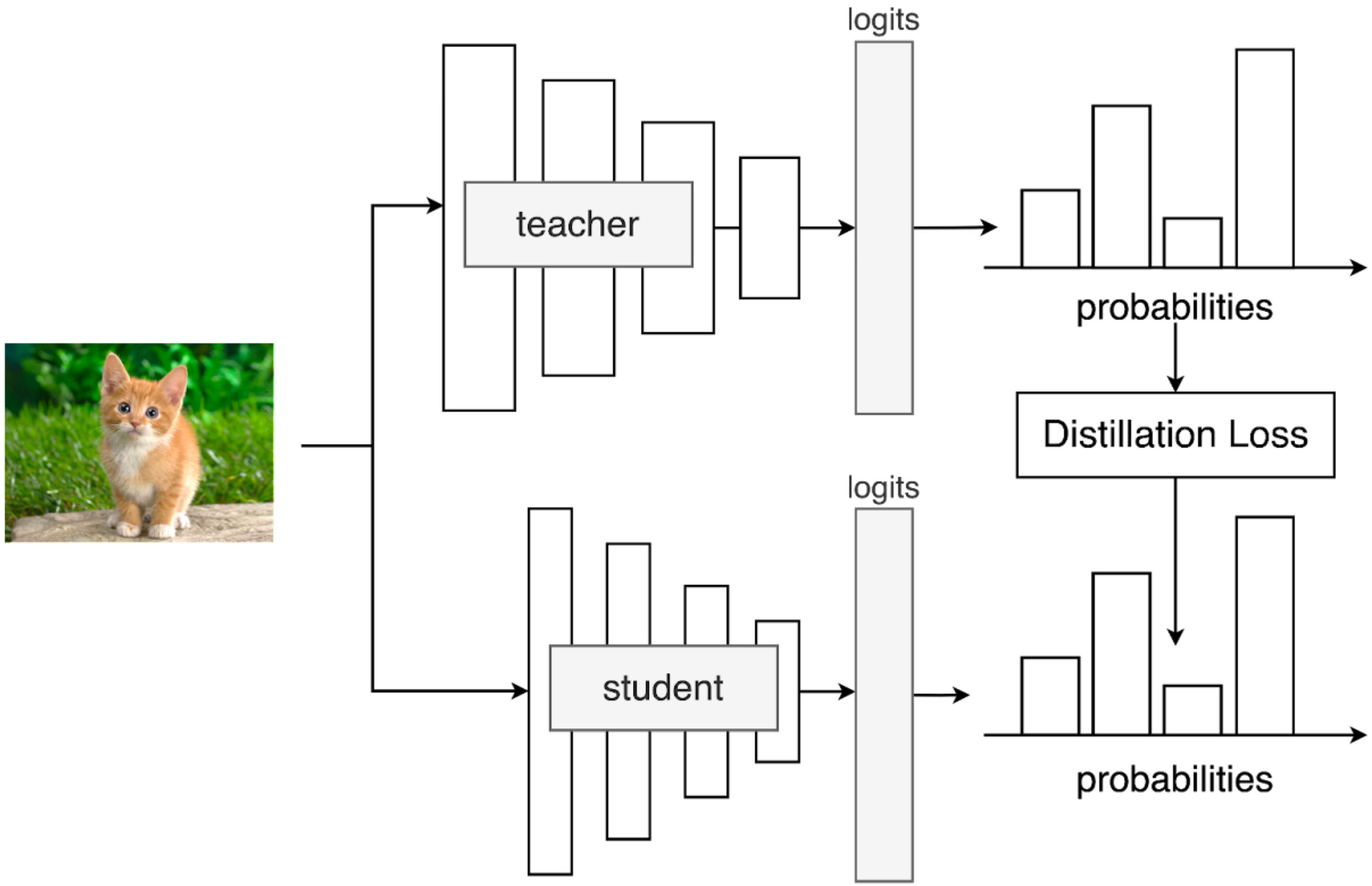
Distill Teacher to Student



$$p_i = \frac{exp(p_i/T)}{\sum_j exp(p_j/T)}$$

$$q_i = \frac{exp(q_i/T)}{\sum_j exp(q_j/T)}$$

Distill teacher to student

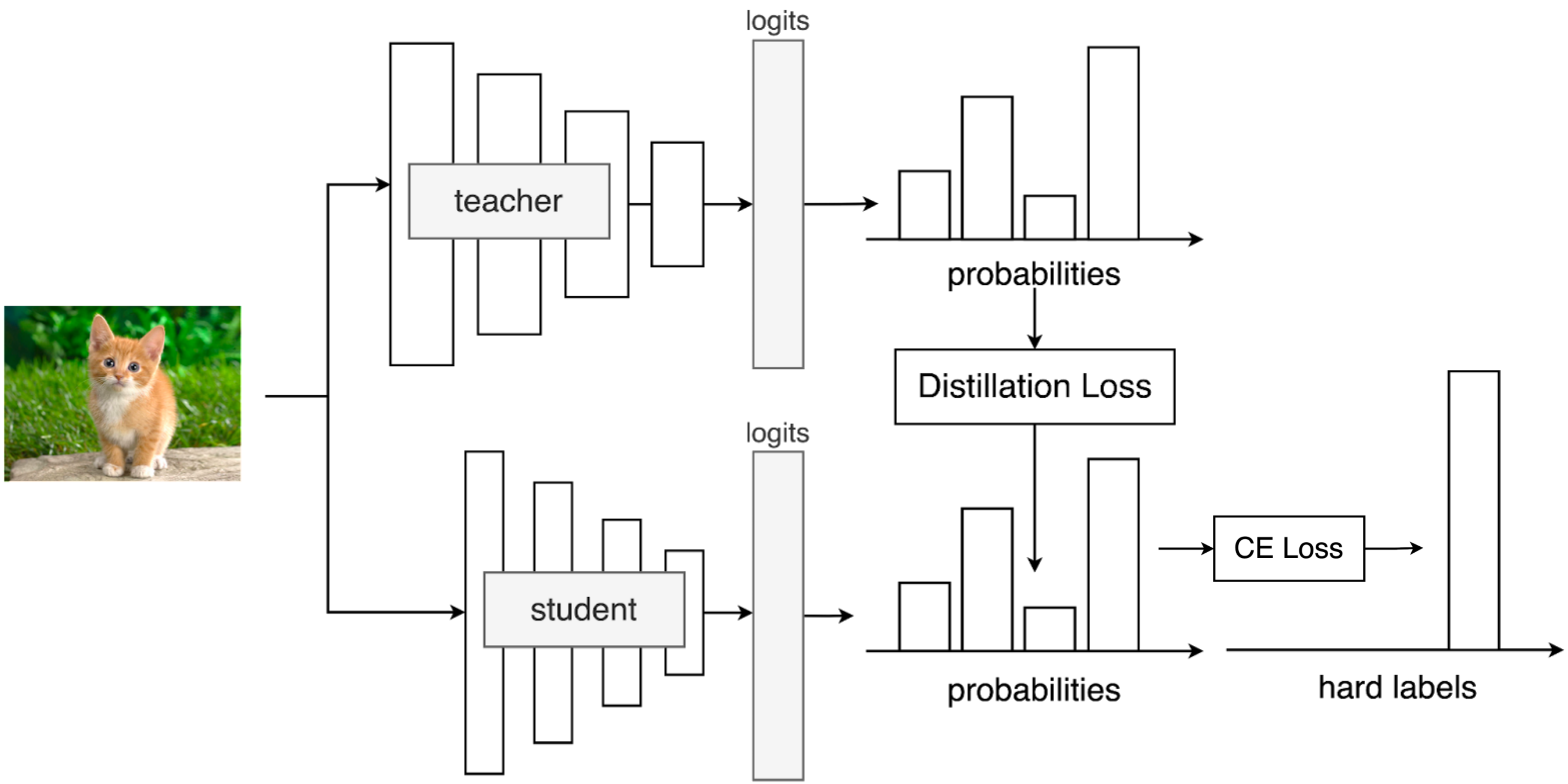


$$p_i = \frac{exp(p_i/T)}{\sum_j exp(p_j/T)}$$

$$H(p, q) = -\sum_i \; p_i \; log \; q_i$$

$$q_i = \frac{exp(q_i/T)}{\sum_j exp(q_j/T)}$$

Distill teacher to student

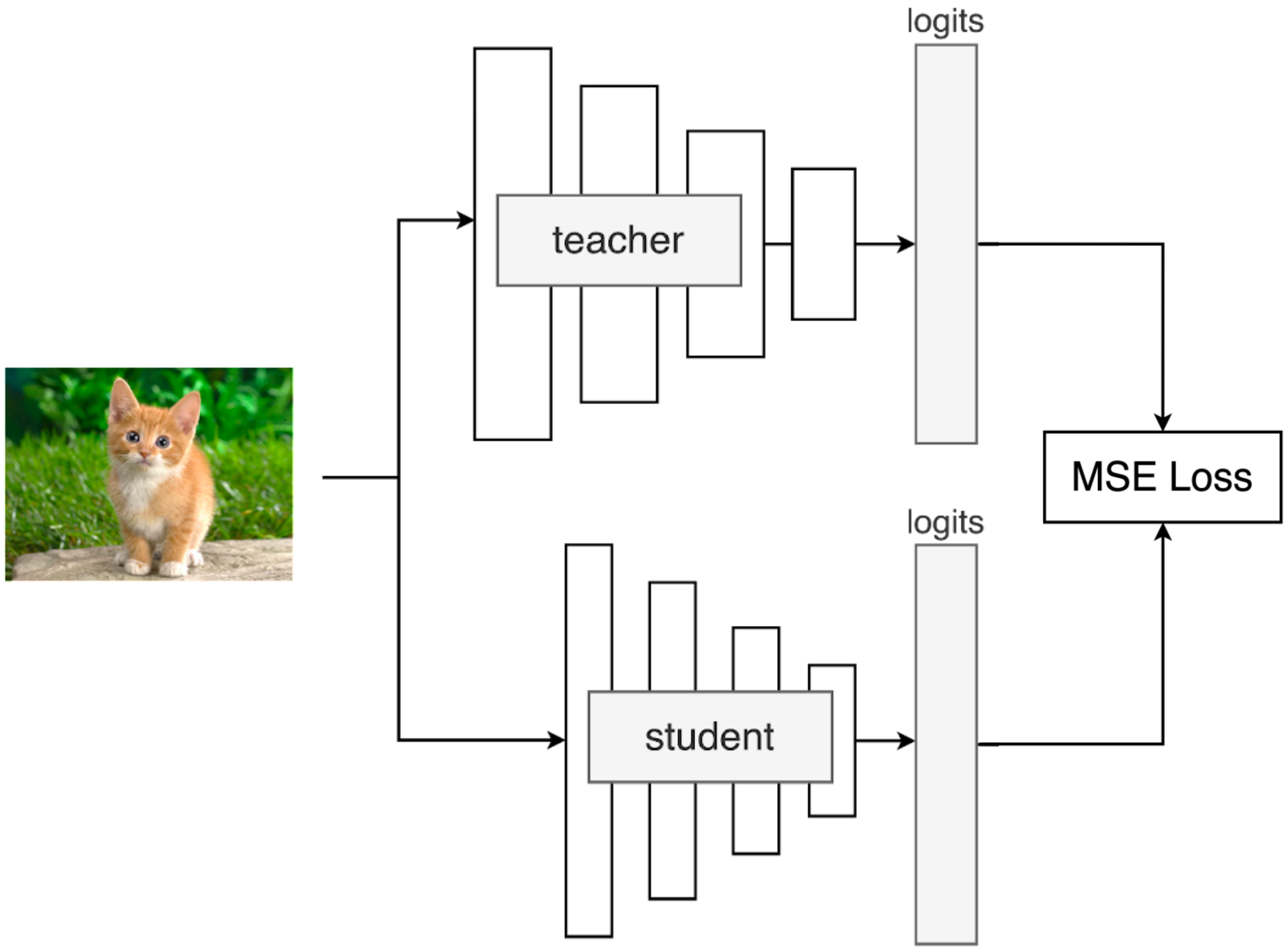


→ Models are trained to optimize performance on the training data when the real objective is to generalize well to new data.

→ We can train the small model to generalize in the same way as the large model.

System & training set	Train Frame Accuracy	Test Frame Accuracy
Baseline (100% of training set)	63.4%	58.9%
Baseline (3% of training set)	67.3%	44.5%
Soft Targets (3% of training set)	65.4%	57.0%

Matching Logits is a Special Case of Distillation



cross-entropy gradient

$$\frac{\partial C}{\partial z_i} = \frac{1}{T}(q_i - p_i) = \frac{1}{T}\left(\frac{e^{z_i/T}}{\sum_j e^{z_j/T}} - \frac{e^{v_i/T}}{\sum_j e^{v_j/T}}\right)$$

temperature is high compared with the magnitude of the logits

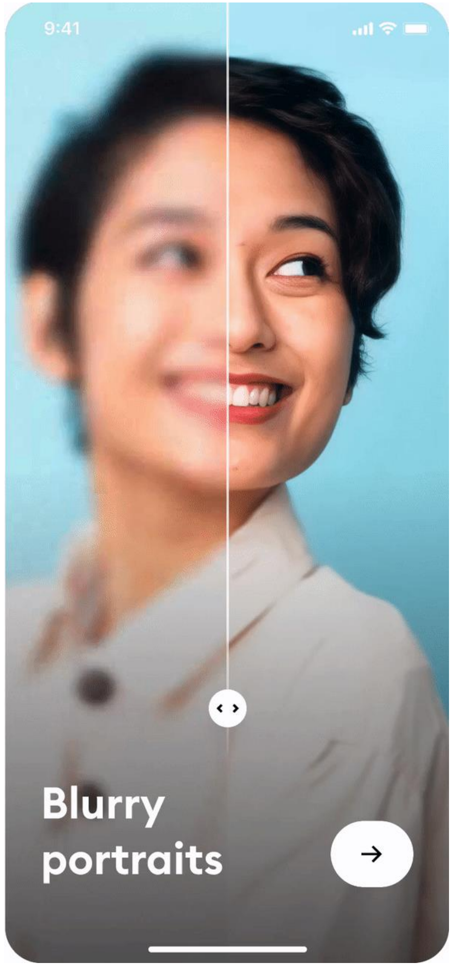
$$\frac{\partial C}{\partial z_i} \approx \frac{1}{T}\left(\frac{1 + z_i/T}{N + \sum_j z_j/T} - \frac{1 + v_i/T}{N + \sum_j v_j/T}\right)$$

logits are normalized to zero-mean $\sum_j z_j = \sum_j v_j = 0$

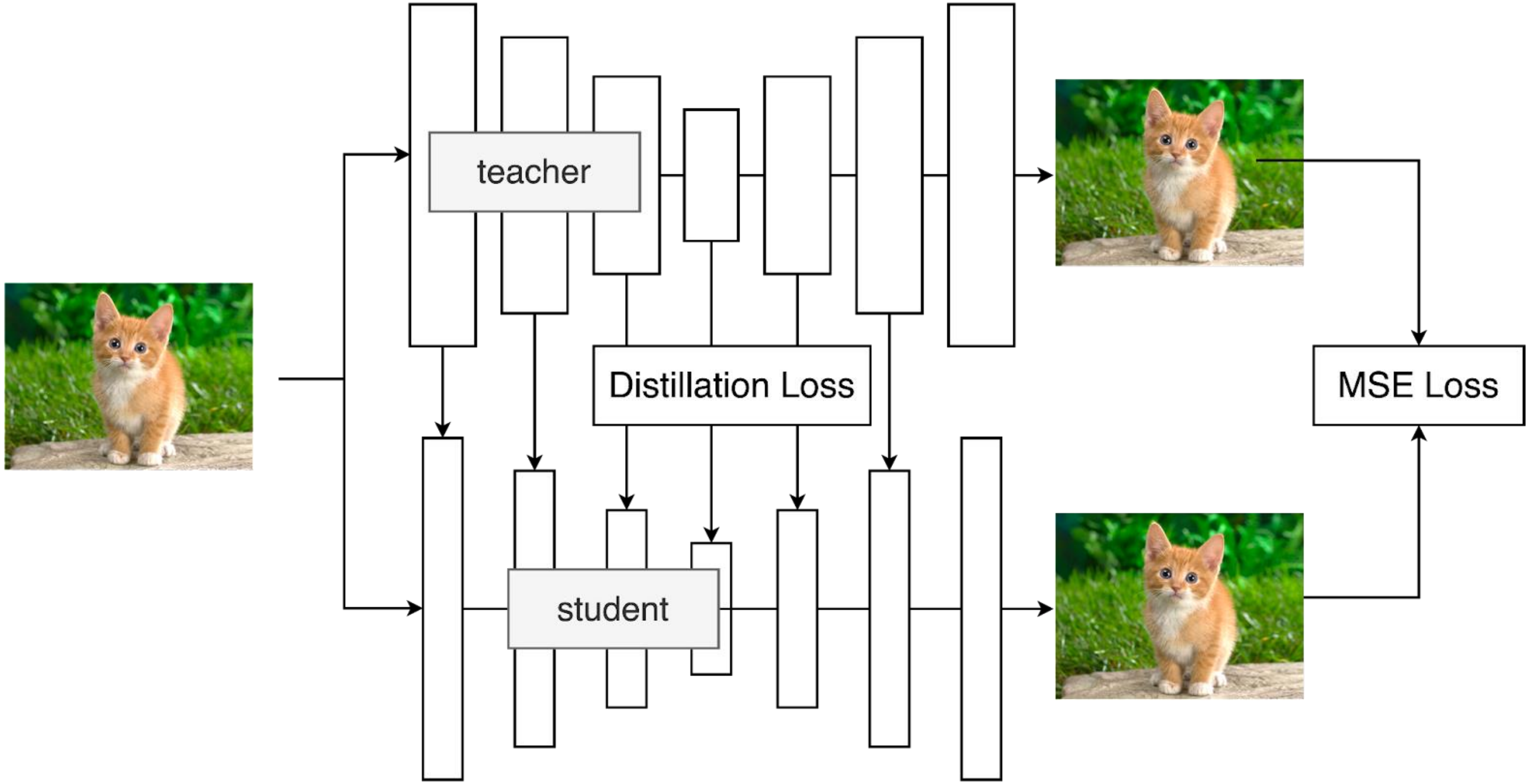
$$\frac{\partial C}{\partial z_i} \approx \frac{1}{NT^2}(z_i - v_i)$$

Beyond Classification

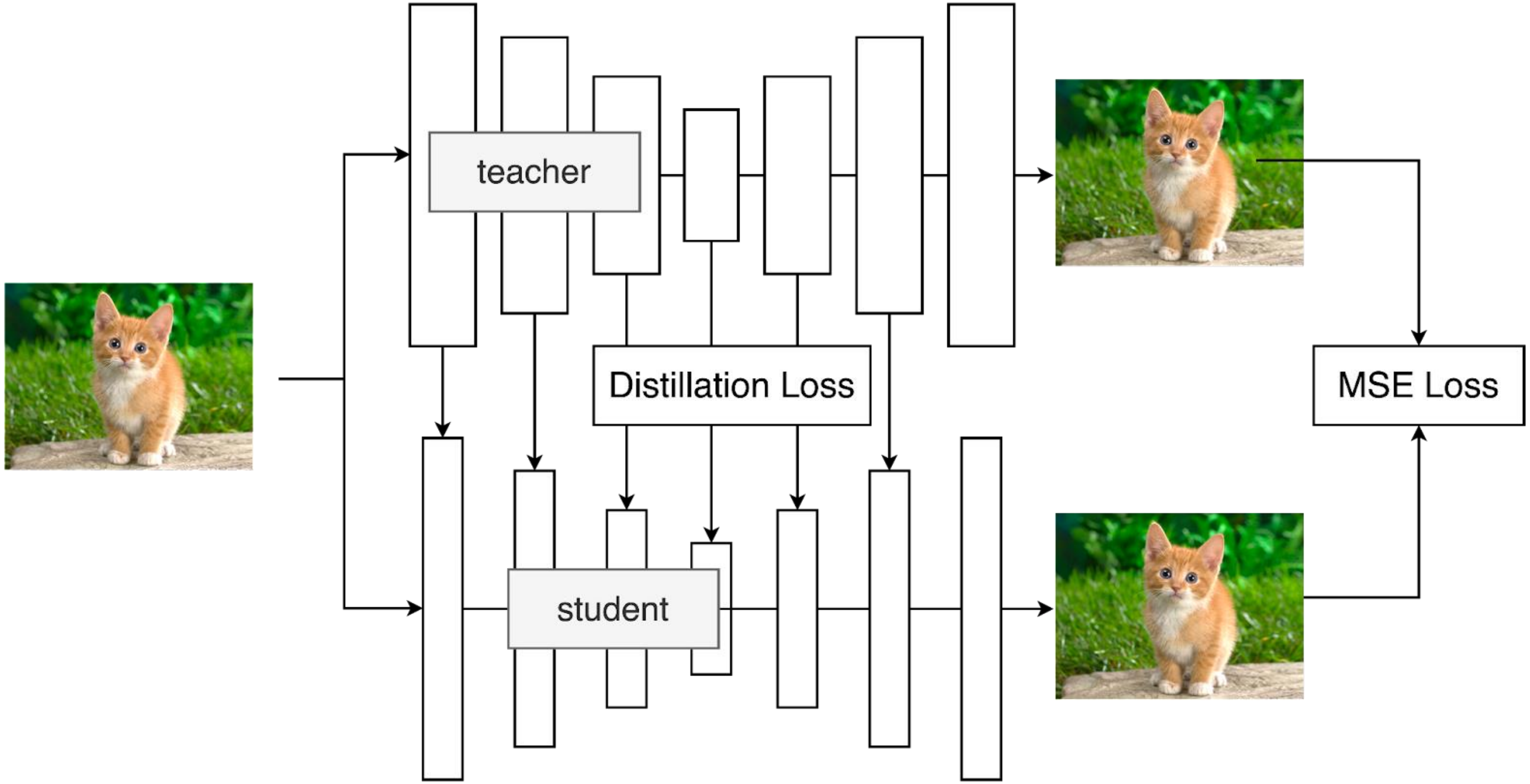
- Dense prediction tasks → **produce pixel-level prediction.**
- Labels are dense and continuous, unlike one-hot vector like classification.
- → Is knowledge distillation still effective on these tasks?
- **Yes → match the activations instead of the logits.**



Knowledge Distillation for Dense Predictions



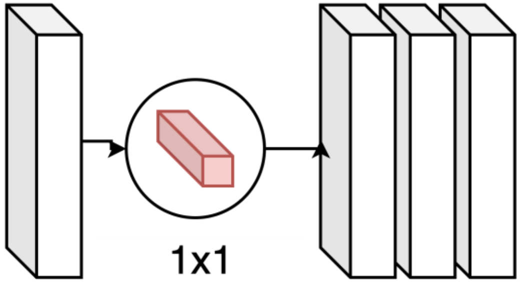
Knowledge Distillation for Dense Predictions



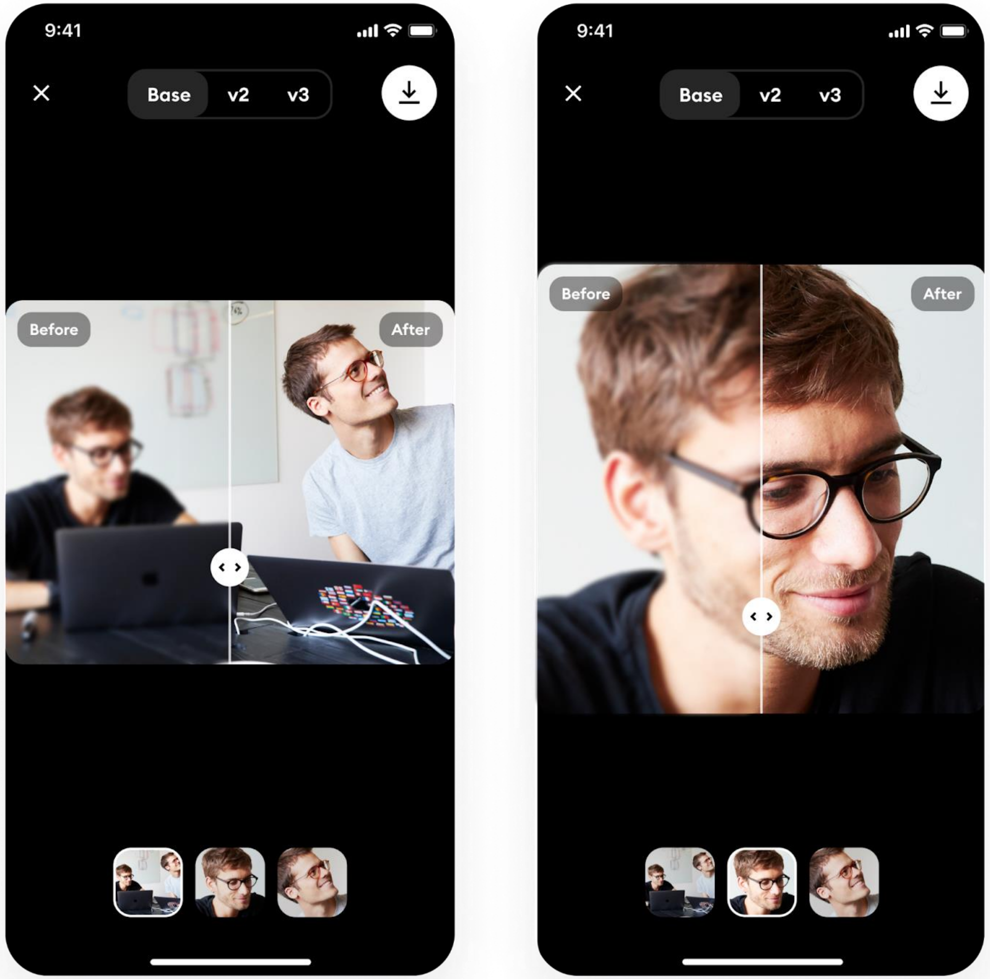
Distillation Loss

→ Mean squared error between activations of the two networks

* Requires embedding into lower dimensional space.

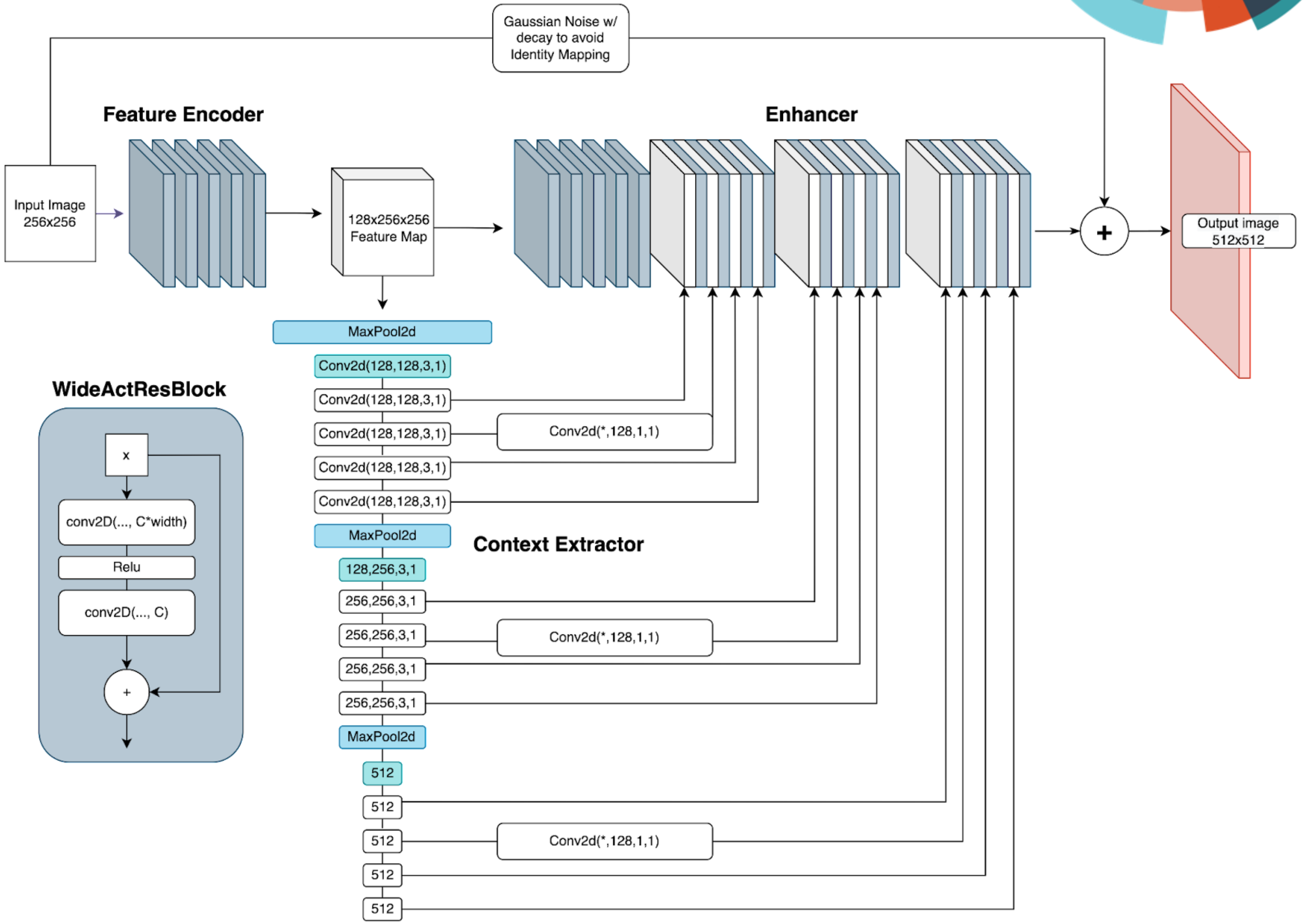


Case Study - Remini Photo Enhancer



Remini Network Architecture

- **Feature Extractor** → embed RGB data into high-resolution features.
- **Context Extractor** → encapsulate global contextual information such as facial symmetry.
- **Enhancer** → Use high-resolution features and global information to restore the image quality.



Remini Distillation Procedure

- **Distillation** → train feature encoder, contextual extractor, enhancer progressively
  - **L1 Loss** → MSE loss between the outputs of the two networks
  - **Distillation Loss** → MSE Loss between the teacher activations and the projection of the student's activations
- **Perceptual Training**: we fine-tune the model end-to-end by comparing the output of the student to that of the teacher
  - **MSE Loss**: same as above
  - **Perceptual Loss** → VGG losses between the output image of the two models
- **Student Fine-tuning**: fine-tuned model using the ground truth images as target

Qualitative Results

Input Teacher Student

Qualitative Results

Input | Teacher | Student

Reconstruction Quality

- Tested different blocks for the projection into matching feature spaces
  - Trained 10k iterations distilling into a 0.5x model
- **PSNR** measures the numerical similarity between two images
  - Not suited for perceptual losses
- LPIPS measure the **perceptual similarity**

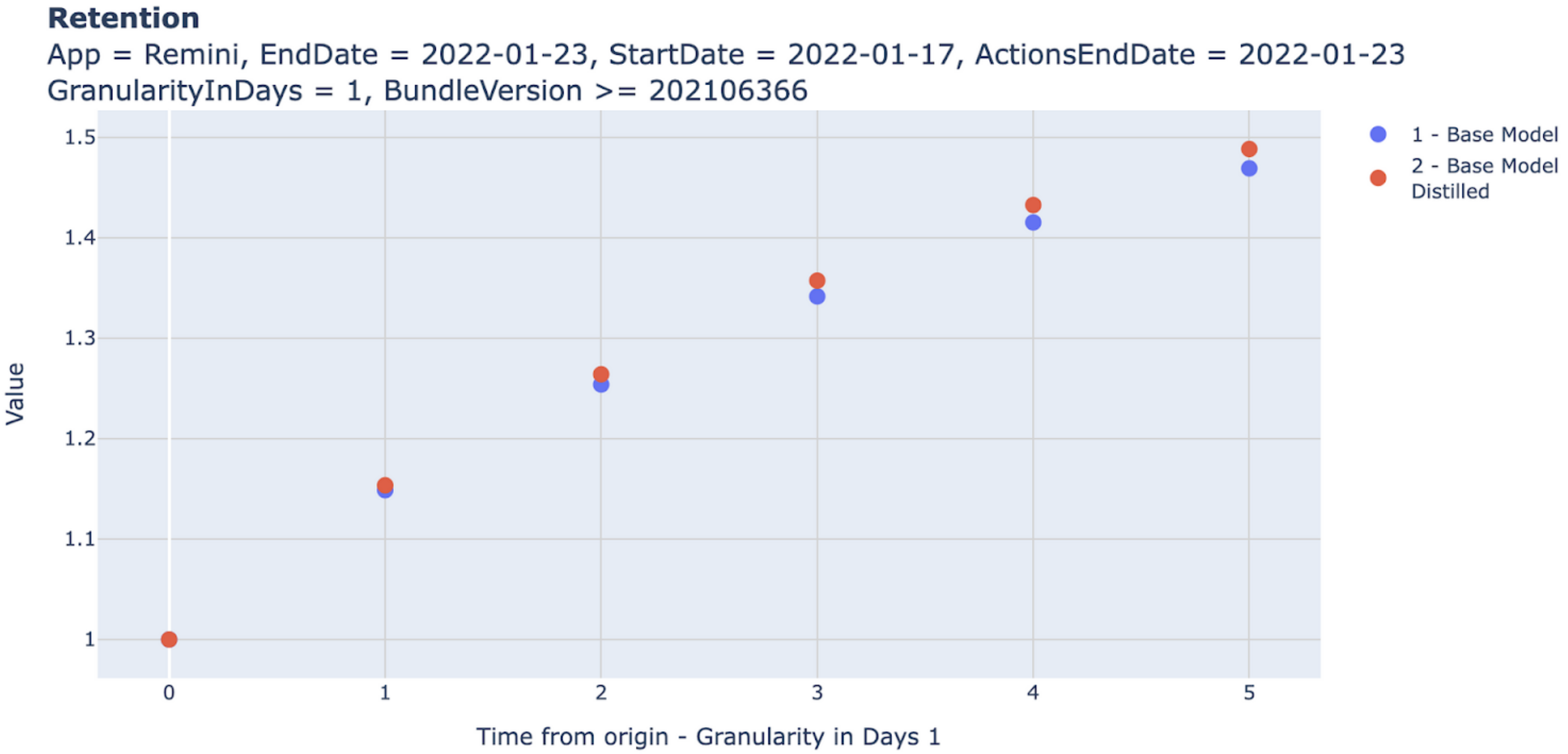
Layersa	PSNR	LPIPS
1x1 convs	30.3	0.32
3x3 convs	31.6	0.31
1x1 conv + Leaky ReLU	31.4	0.31
1x1 conv + sine activation	30.1	0.34

Runtime Performance

- **Scale factor** - reduction of channels in each convolutional block
  - Achieve 3x speedup halving the width of the neural network

Channels scale factor	Convolutions type	Encoder Inference time [s]	Extractor inference time [s]	Enhancer inference time [s]
1x	Regular	0.017	0.014	0.415
0.5x	Regular	0.006	0.005	0.113
2x	Regular	0.063	0.053	1.704
1x	DWS	0.016	0.013	0.154

Results - User A/B Testing




```
Significant experiment!
Significant segments, number of sigmas and bounds:
{'1 - Base Model,2 - Base Model<br>Distilled': (4.208482309156419, 1.0079770983686298, 1.0183051836695876)}
```

Thank you

Resources

Distillation

- “Model Compression” https://dl.acm.org/doi/10.1145/1150402.1150464
- “Distilling the knowledge in a neural network” https://arxiv.org/abs/1503.02531
- “Knowledge Distillation A Survey” https://arxiv.org/pdf/2006.05525.pdf

Bending Spoons

- Remini: https://apps.apple.com/us/app/remini-ai-photo-enhancer/id1470373330
- Careers https://bendingspoons.com/careers.html