Departing From Supervised Learning!

Giacomo Boracchi,

Politecnico di Milano, DEIB.

https://boracchi.faculty.polimi.it/

February 28th, 2025

Advanced Deep Learning, PhD course

Lecture Outline

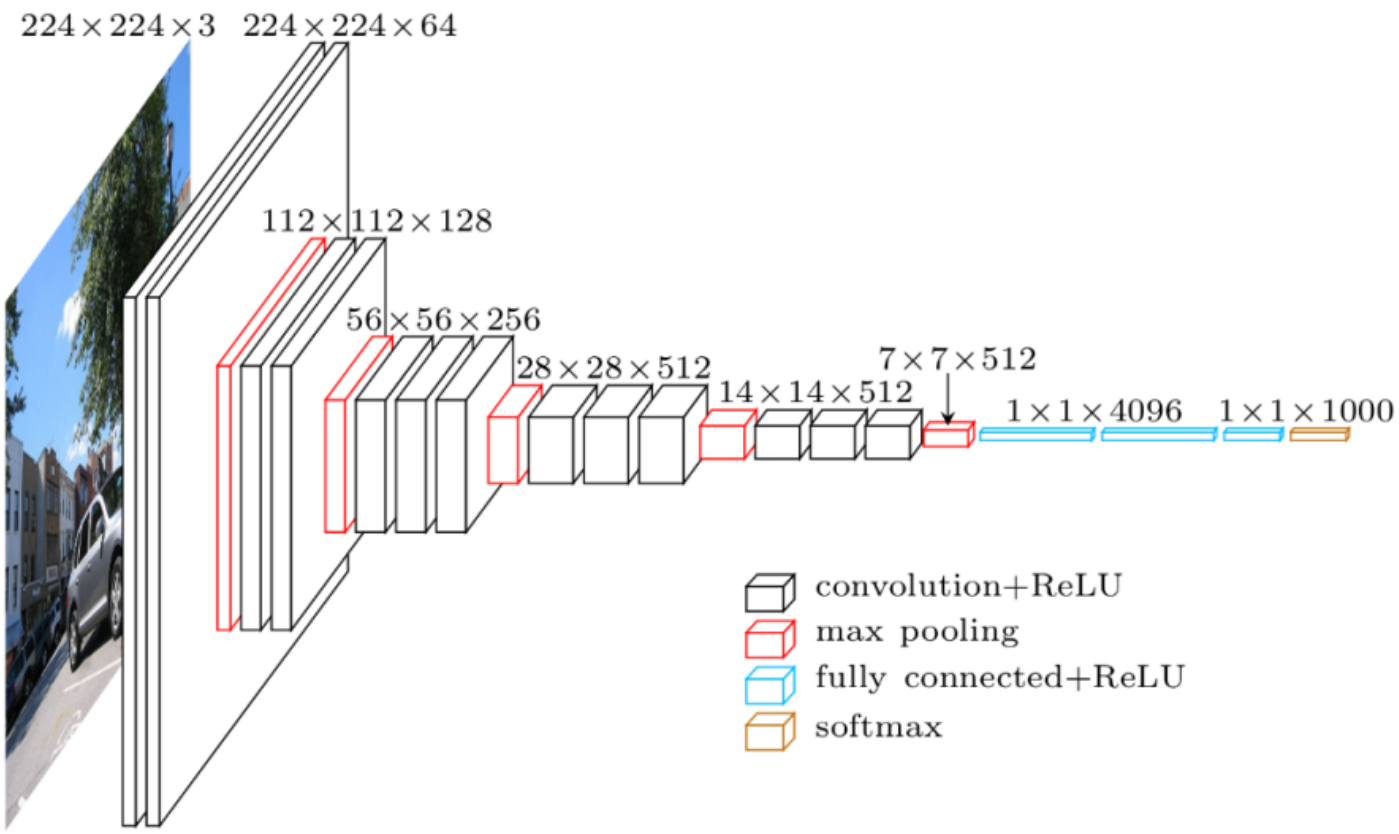
- Contrasative Learning and Siamese Network: a very practical tool
- Self-Supervised Learning: focus on augmentation-based SSL
- Image Restoration: focus on SSL solutions

Metric Learning

Say you are asked to implement a face identification system
to open Politecnico door!

Classification? Detection? Segmentation?

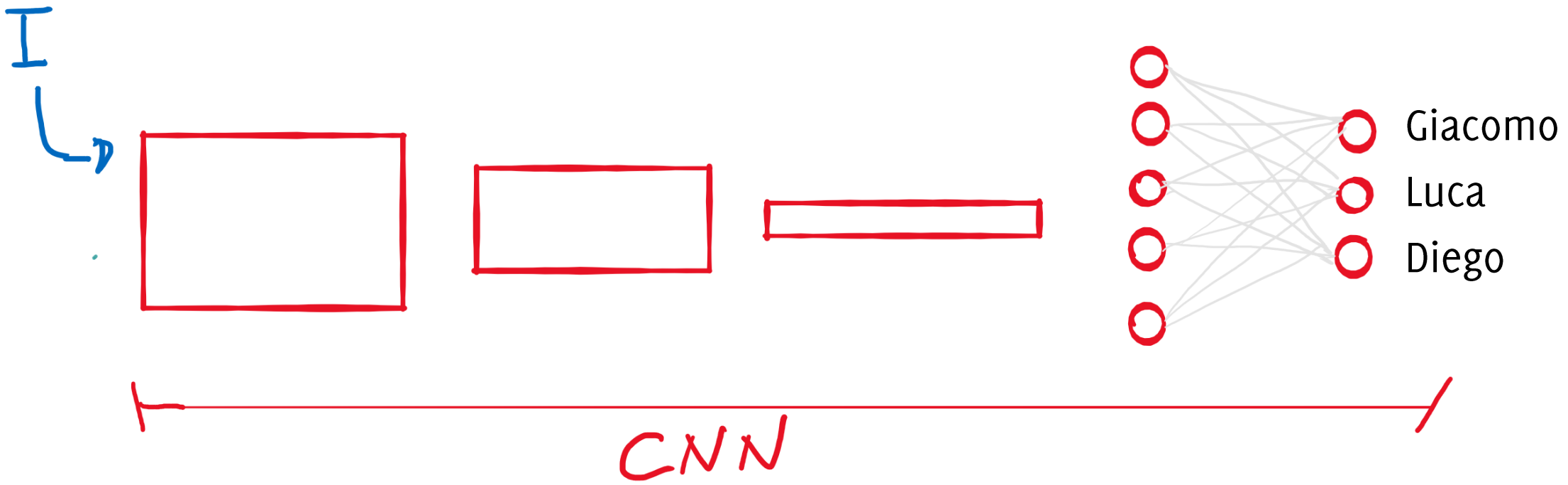
Let's start fresh from a Convolutional Neural Network for classification



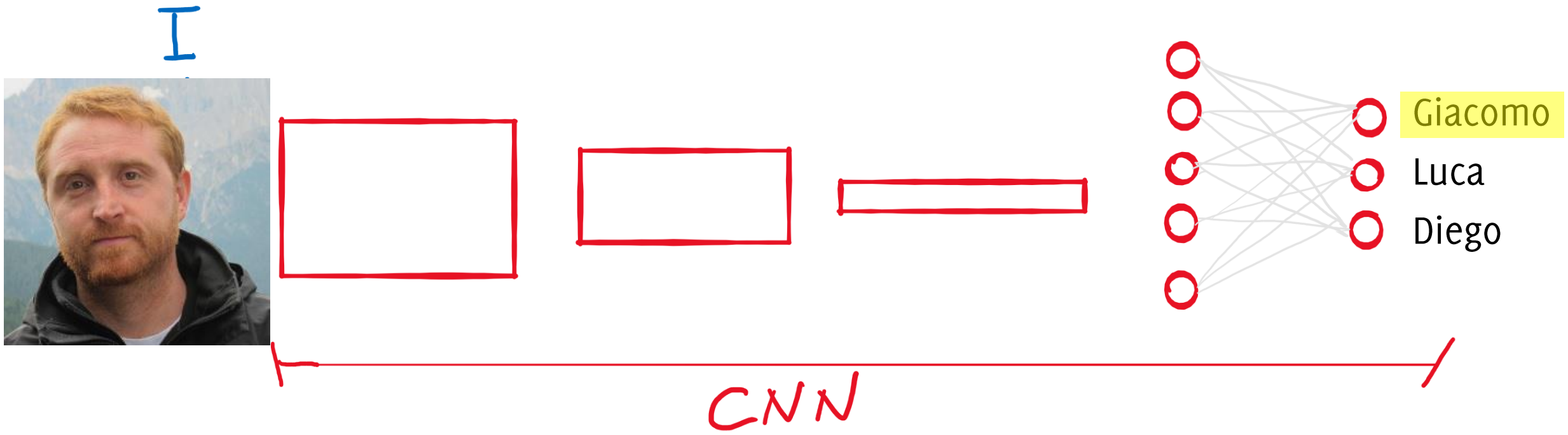
1 person = 1 class

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." ICLR 2015

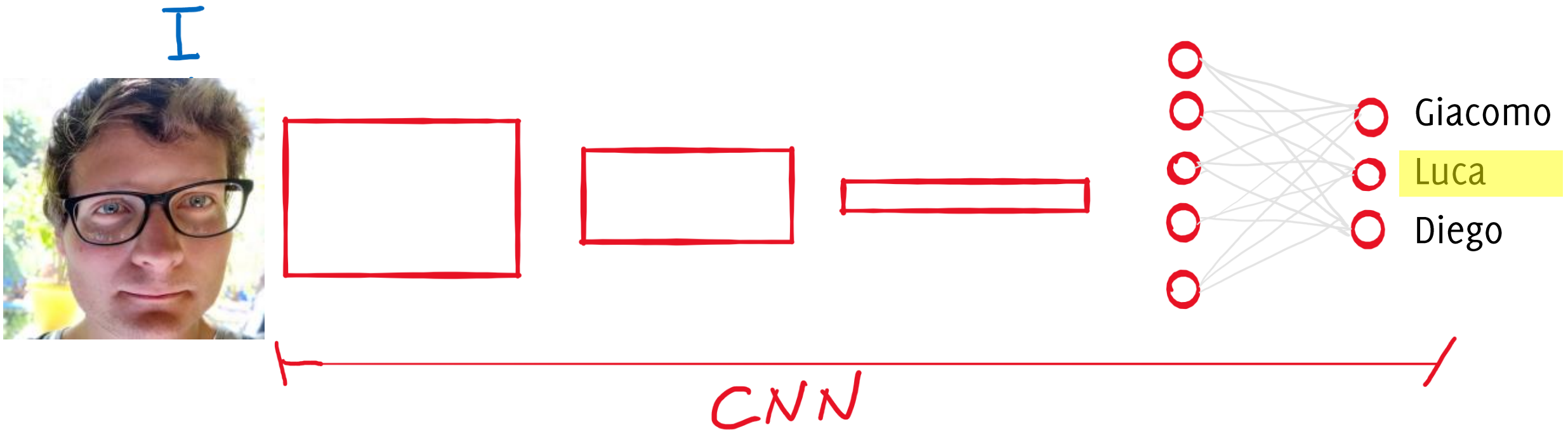
Face Identification by Classification



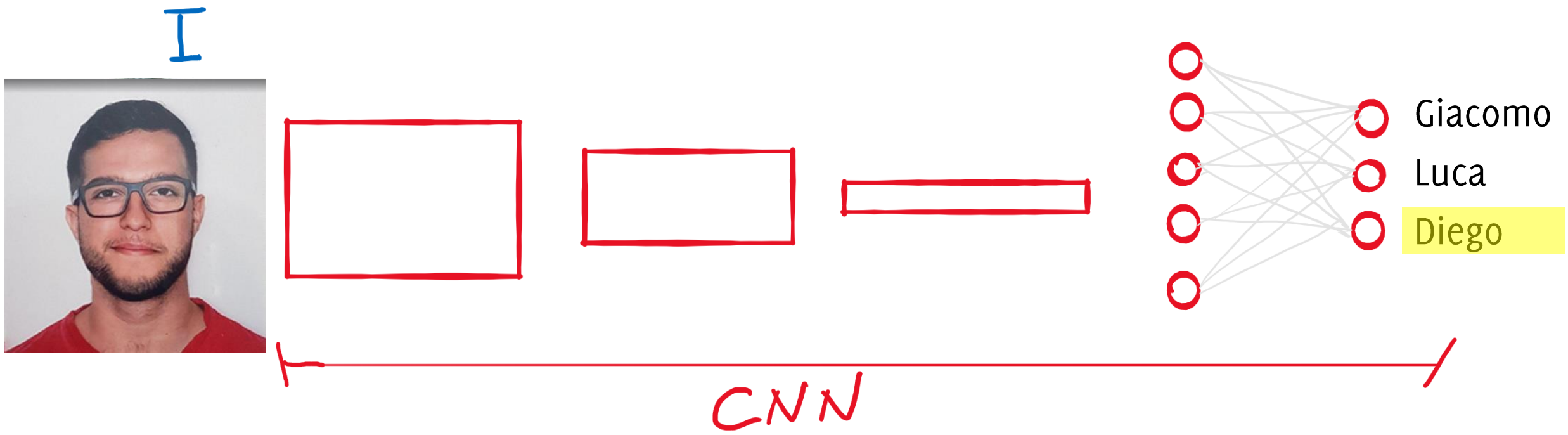
Face Identification by Classification



Face Identification by Classification



Face Identification by Classification



What Do We Need?

A training set

- A few images per class / person
- Possibly Images in different conditions (position, light, facial expression, clothes …)

A few Py snippets and a GPU…

That’s easy…

What Do We Need?

A training set

- A few images per class / person
- Possibly Images in different conditions (position, light, facial expression, clothes …)

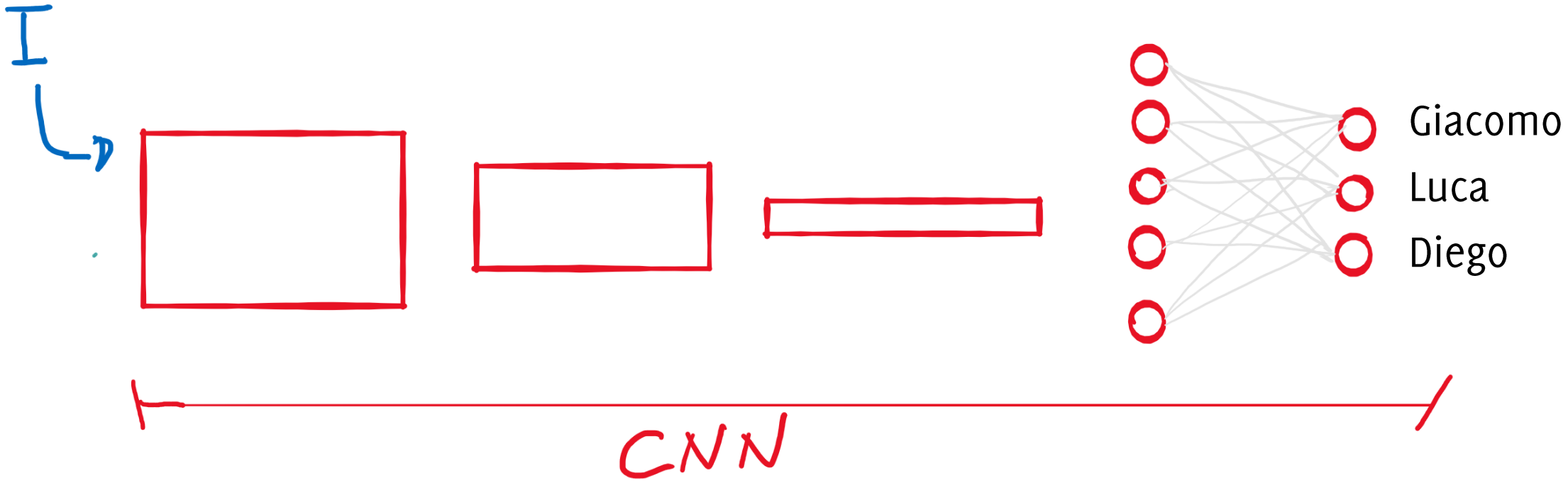
A few Py snippets and a GPU…

That’s easy…

Are we happy with this solution?

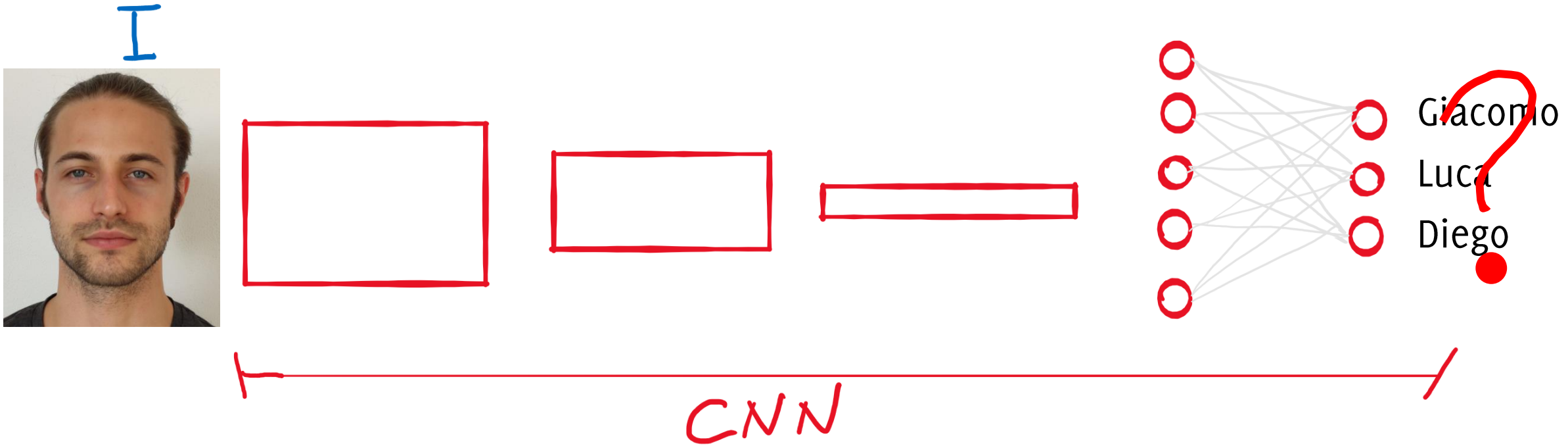
… Not Quite

What happens when you need to enroll a new employee?



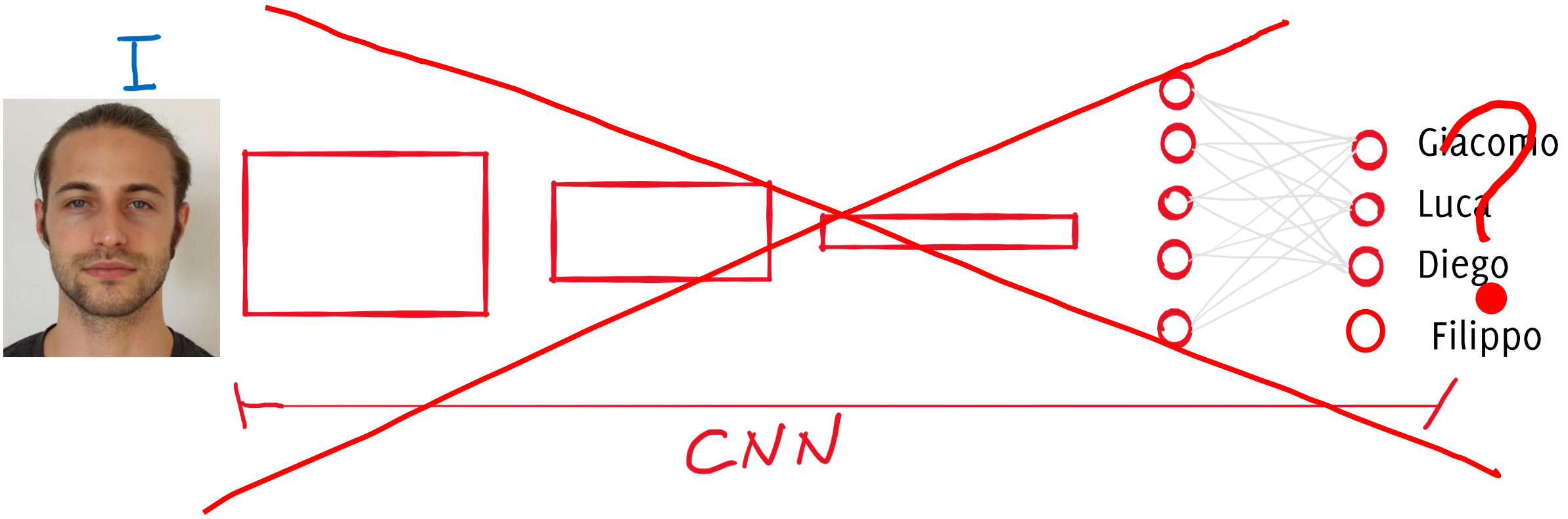
… Not Quite

What happens when you need to enroll a new employee?



… Not quite

What happens when you need to enroll a new employee?



The whole network has to be retrained for each new person to be identified

A Different Approach is Needed!

What about image comparison?

Why don't we store a picture for each employee **(template)**, and then perform identification by **pairing the input to its closest template**?

T_1 T_2 T_3

What about image comparison?

So, identification becomes:

$$\hat{\imath} = \underset{j=1,..,3}{\operatorname{argmin}} \left\| I - T_j \right\|_2$$

I

T_1

T_2

T_3

What about image comparison?

Enrolling a new individual would be straightforward

T_1

T_2

T_3

What about image comparison?

Enrolling a new individual would be straightforward… it is enough to add a template to the database!

T_1 T_2 T_3 T_4

What about image comparison?

… but how to perform identification?

$$\hat{\imath} = \underset{j=1,..,4}{\operatorname{argmin}} \left\| I - T_j \right\|_2$$

I

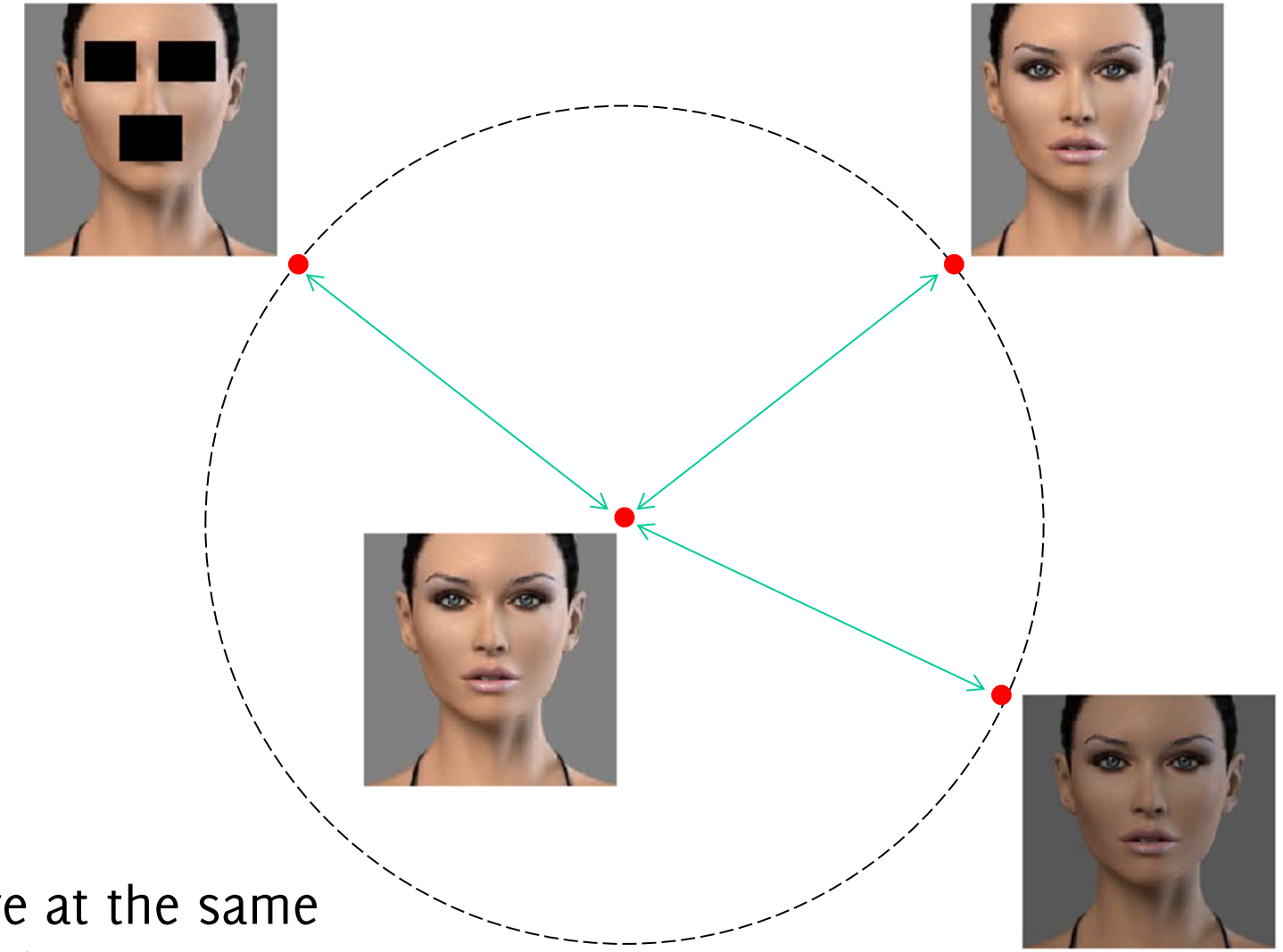
T_1

T_2

T_3

T_4

Bad News…



The three images are at the same distance from the reference at the center

What about image comparison?

… but how to perform identification?

$$\hat{\imath} = \underset{j=1,..,4}{\operatorname{argmin}} \left\| I - T_j \right\|_2$$

I

T_1

T_2

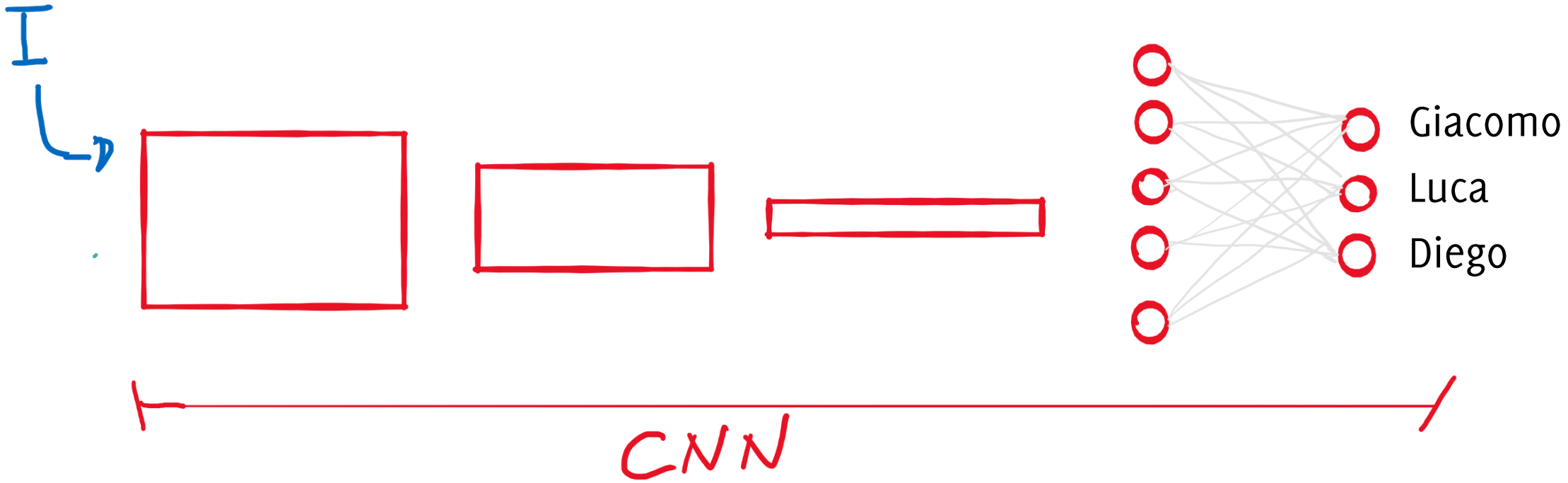
T_3

T_4

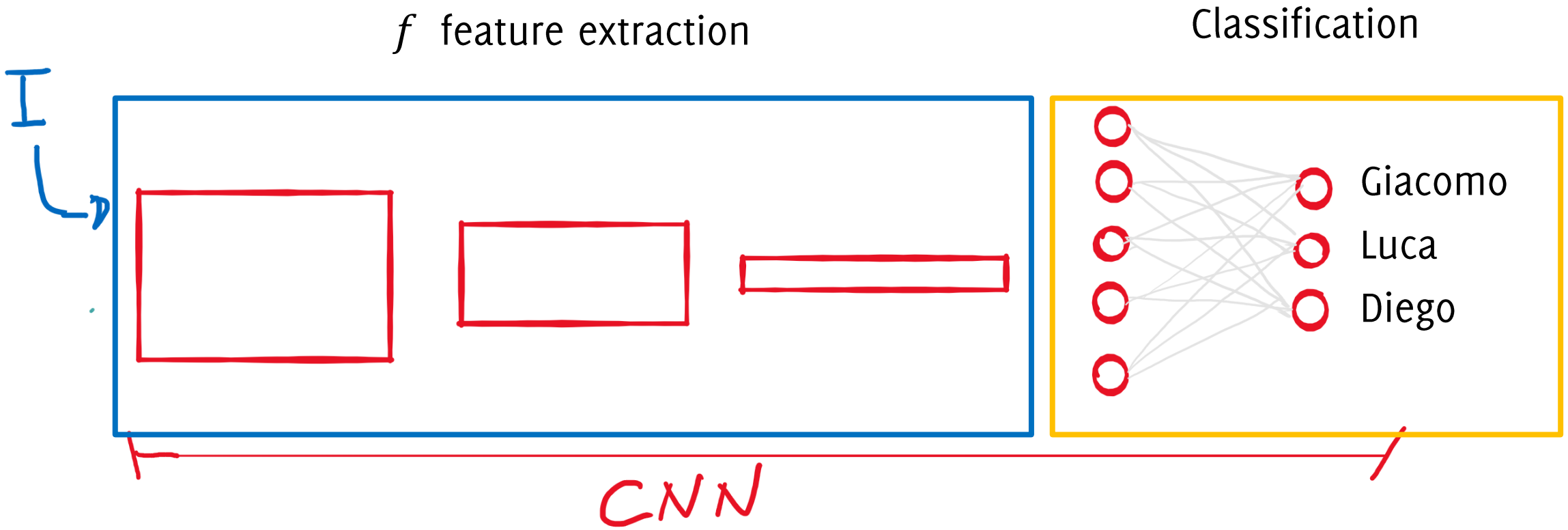
Is it possible to move to a learned distance,
and to train tihs relying only on these examples?

…we need a better distance measure for face identification!

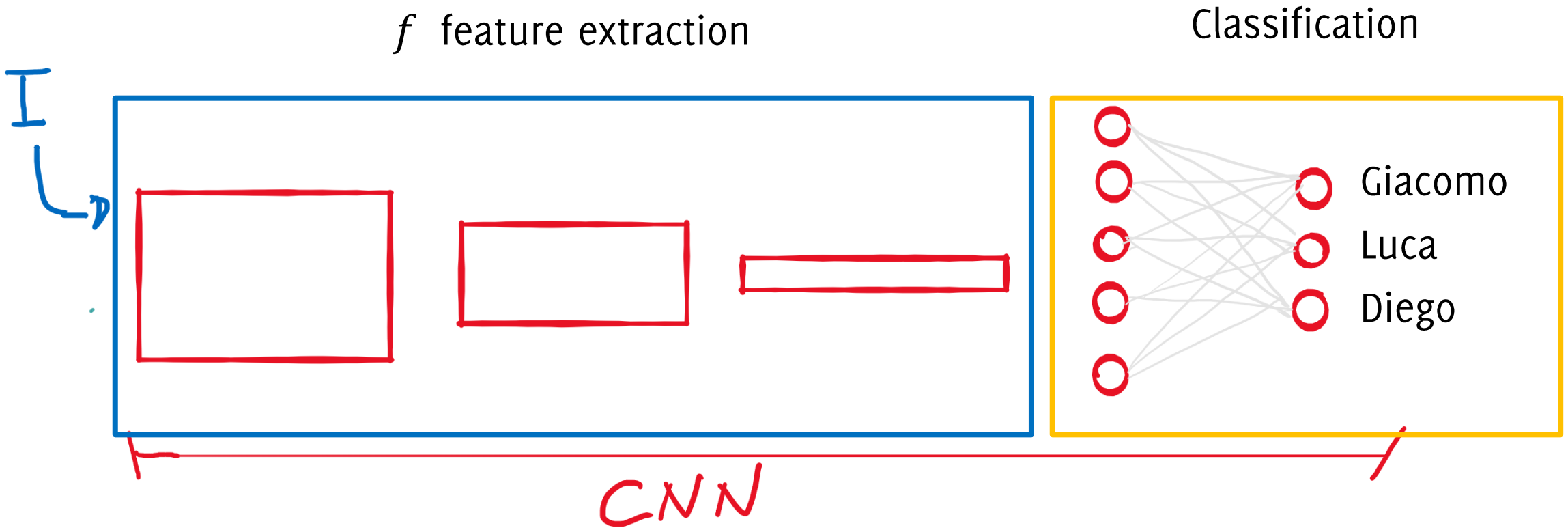
Dissecting CNN



Dissecting CNN

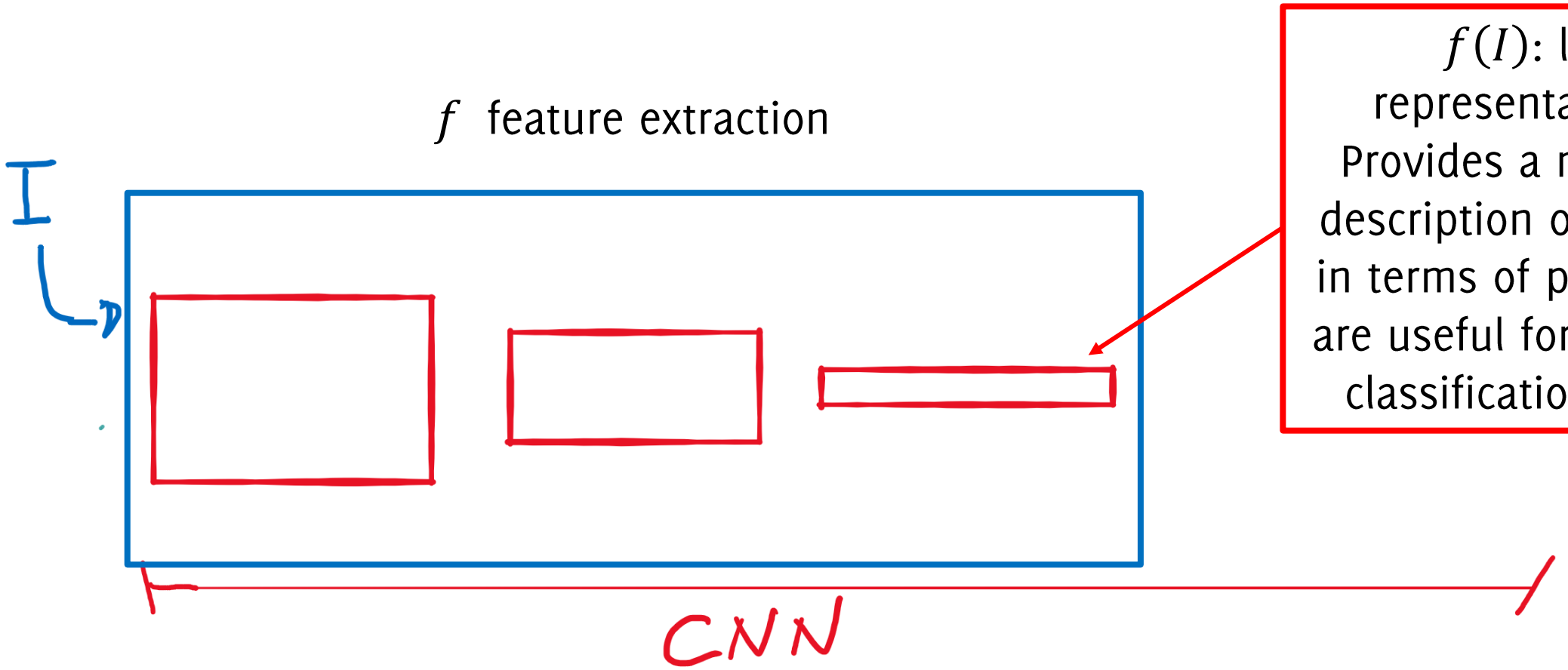


Dissecting CNN



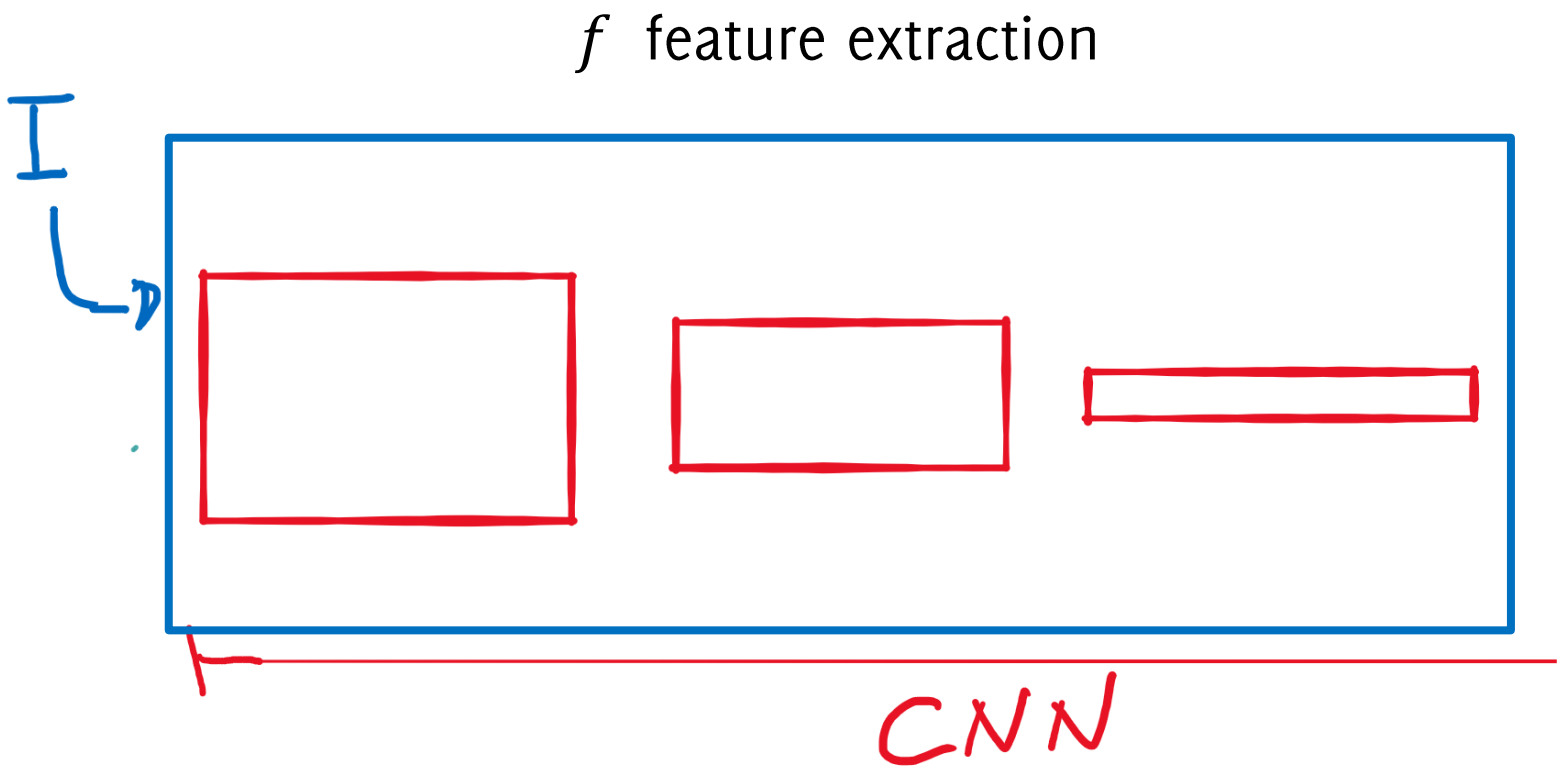
The feature extraction part is typically more general-purpose than the classification part (see Transfer Learning / Domain Adaptation)

Latent representation



$f(I)$: latent representation of I
Provides a meaningful description of the image in terms of patterns that are useful for solving the classification problem

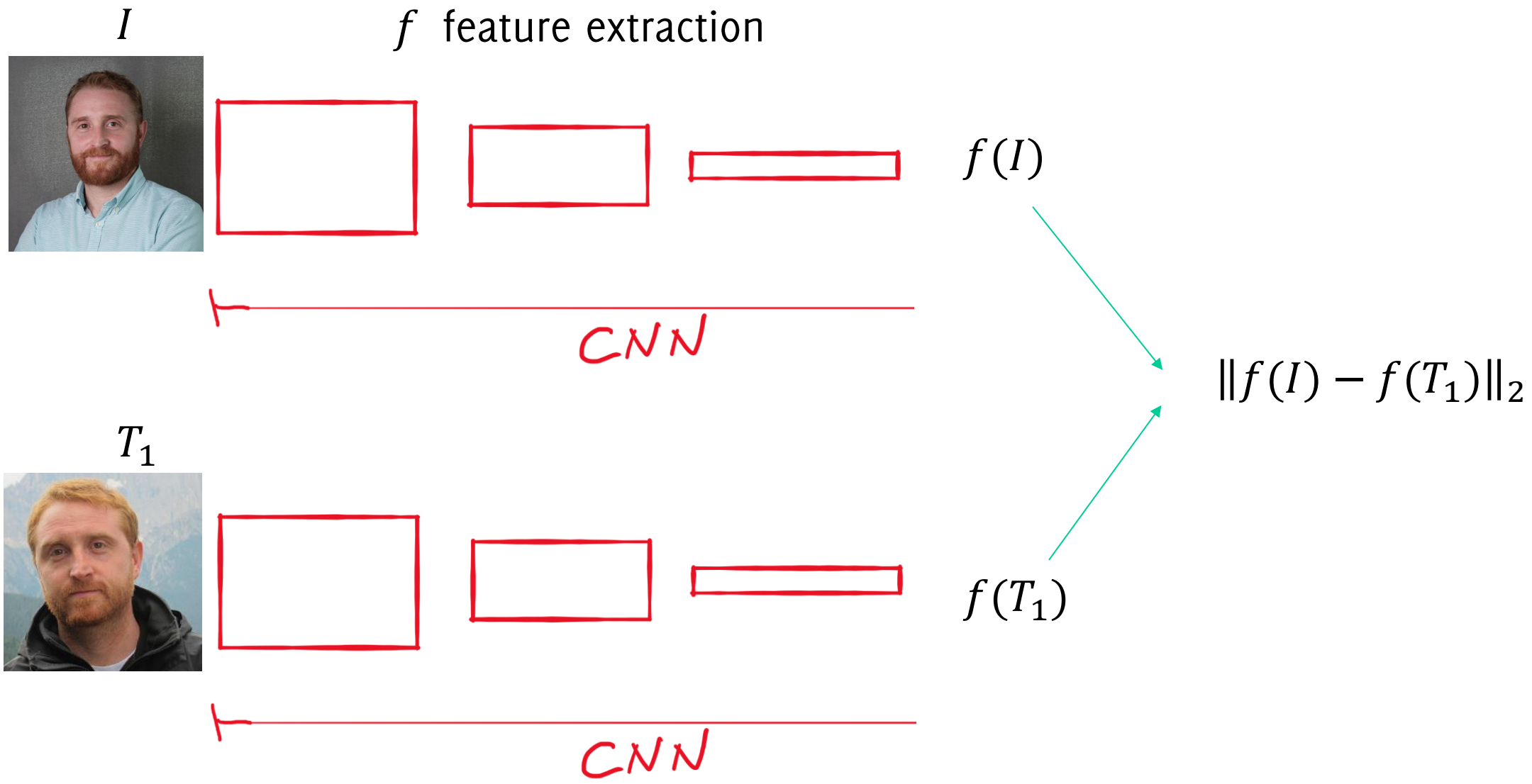
Distance among latent representations



What about using?

$$\hat{\imath} = \underset{j=1,..,4}{\operatorname{argmin}} \left\| f(I) - f(T_j) \right\|_2$$

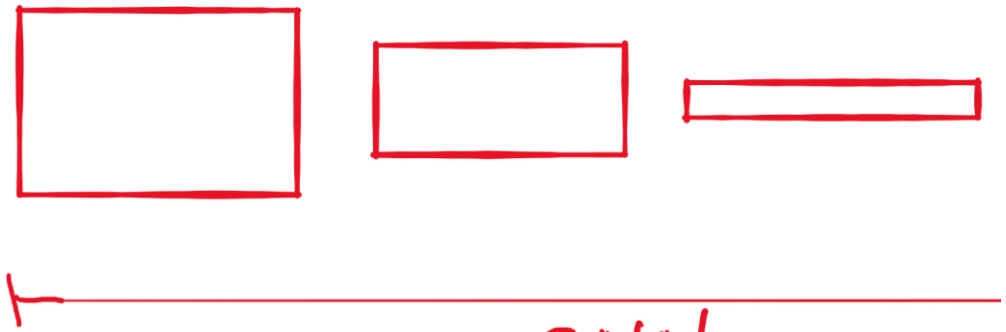
Distance among latent representations



Distance among latent representations

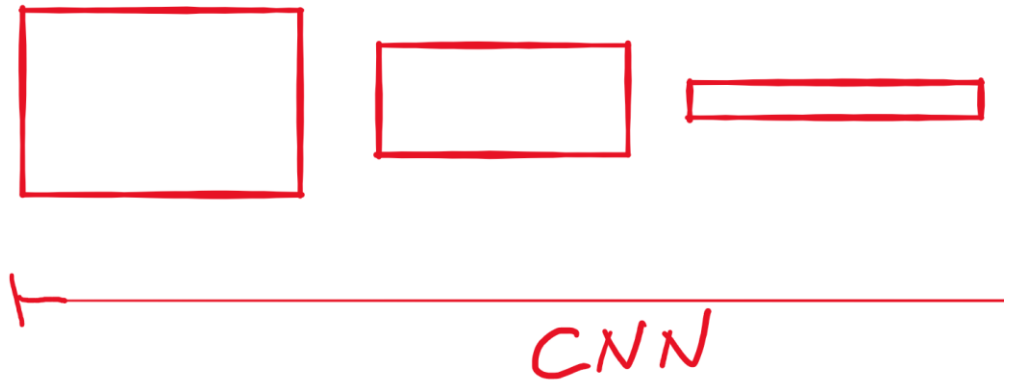
I

f feature extraction



$f(I)$

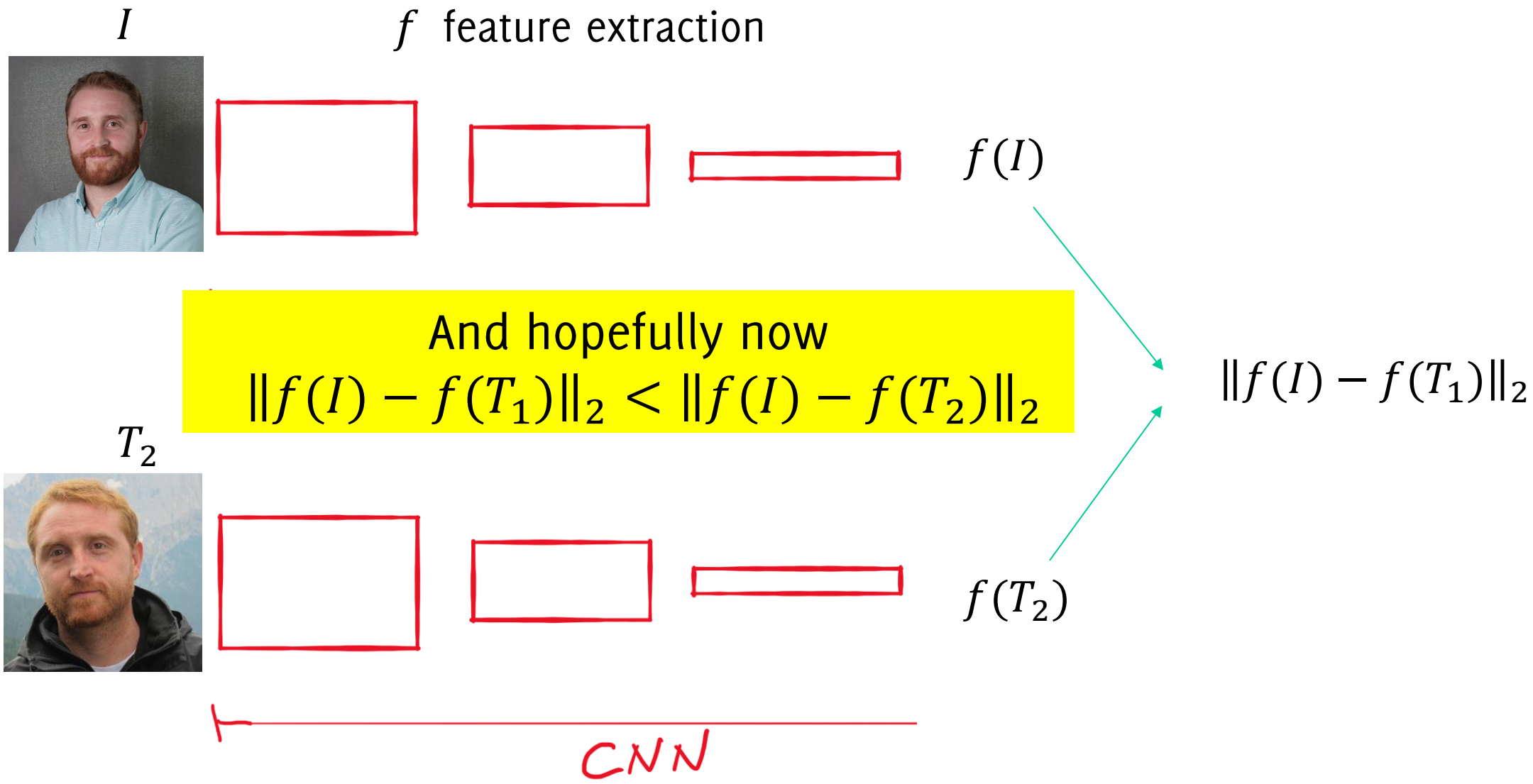
T_2



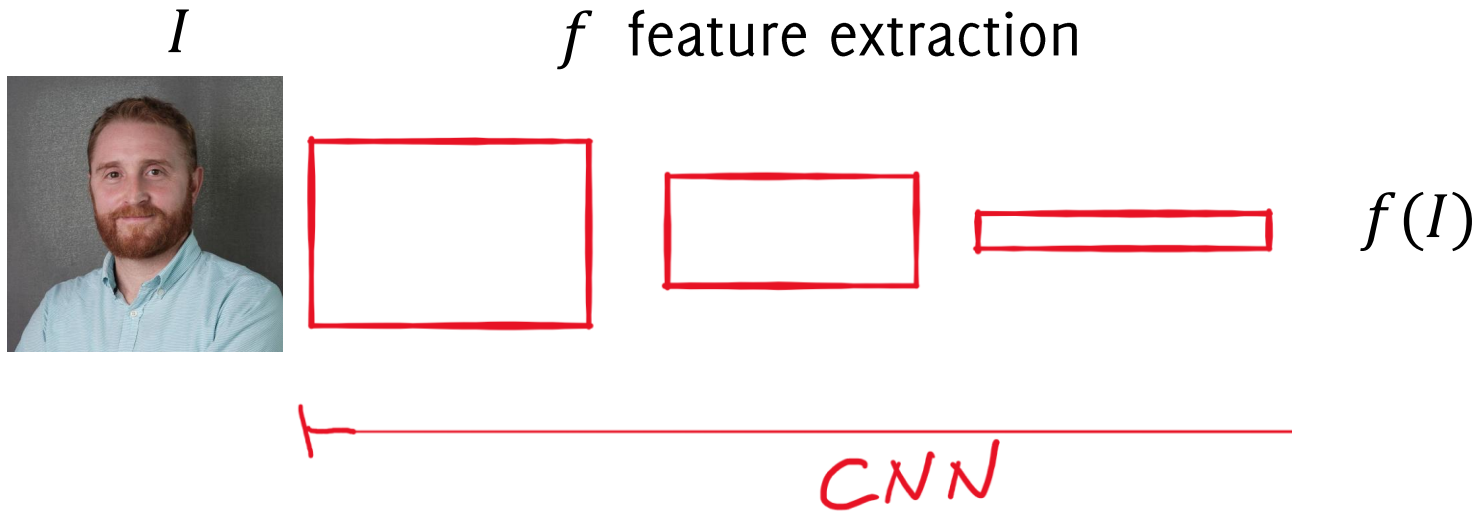
$f(T_2)$

$\|f(I) - f(T_2)\|_2$

Distance among latent representations



In practice



1. Extract features from the first image $f(I)$
2. Perform identification as $\hat{\imath} = \underset{j=1,..,4}{\operatorname{argmin}} \left\| f(I) - f(T_j) \right\|_2$

No need to compute $f(T_j)\ \forall j$, these can be directly stored

This is equivalent to perform image retrival in the latent space… we know this can work!

This is the **t-SNE visualization**: all the images are set in a tile,such that images nearby have latent representations that are closer

Distances in the latent representation of a CNN are much more meaningful than data itself.

However… can we do any better?
…. After all, the network $f(\cdot)$ was not trained for this purpose…

Metric Learning

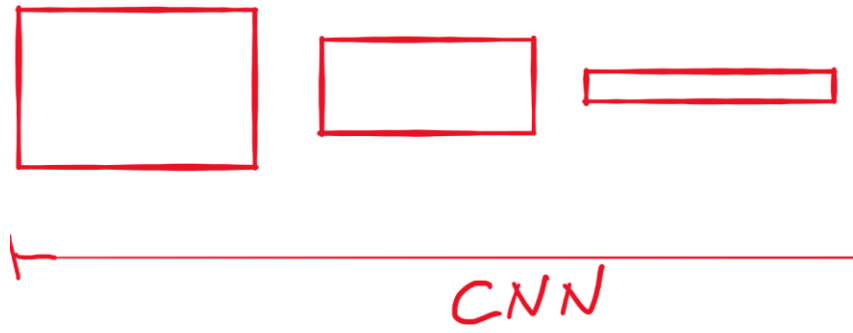
We are still **comparing latent representations trained for classification** (over a few persons?)**, while not for comparing images.**

A more appealing perspective would be to **train the network to measure distances between images.**

We would like to **train the weights** W of our CNN such that

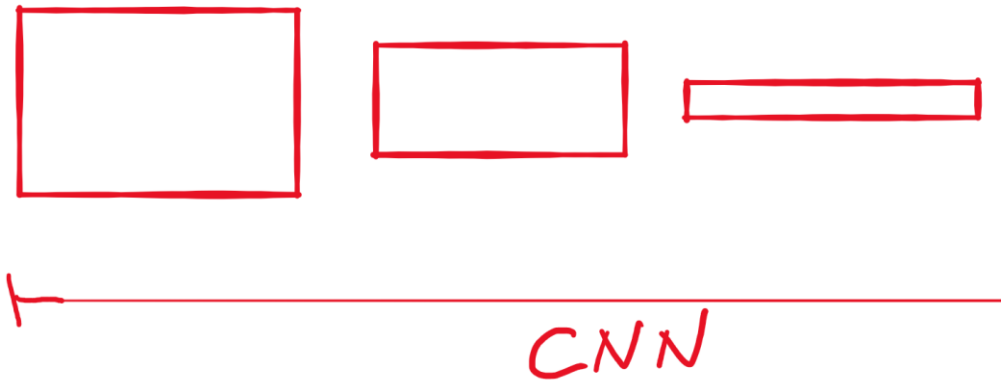
$$\|f_W(I) - f_W(T_i)\|_2 < \left\|f_W(I) - f_W(T_j)\right\|_2 \quad \forall j \neq i$$

When I belongs to class i



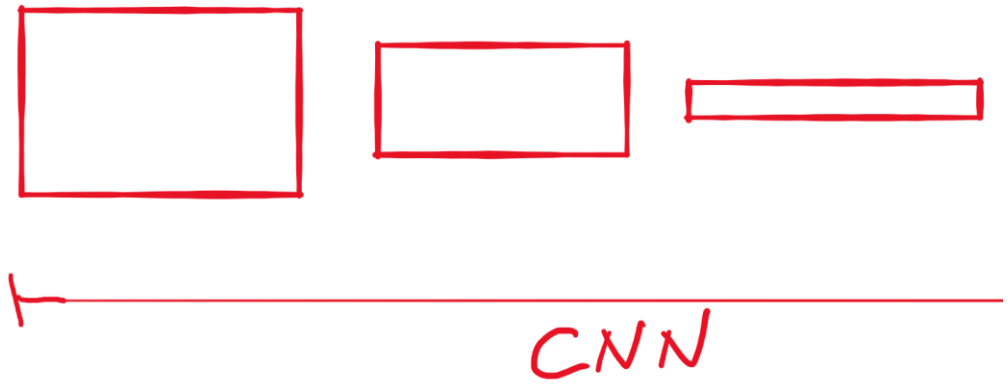
Siamese Networks

I



$f_W(I)$

T_2



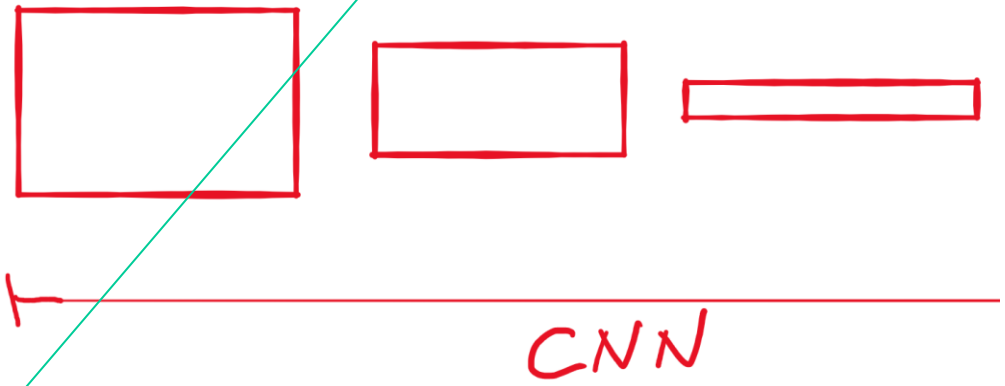
$f_W(T_2)$

The «two» networks have to **perform the same operations**, using the same weights W. Hence the term **Siamese**

How to Train the Siamese Network?

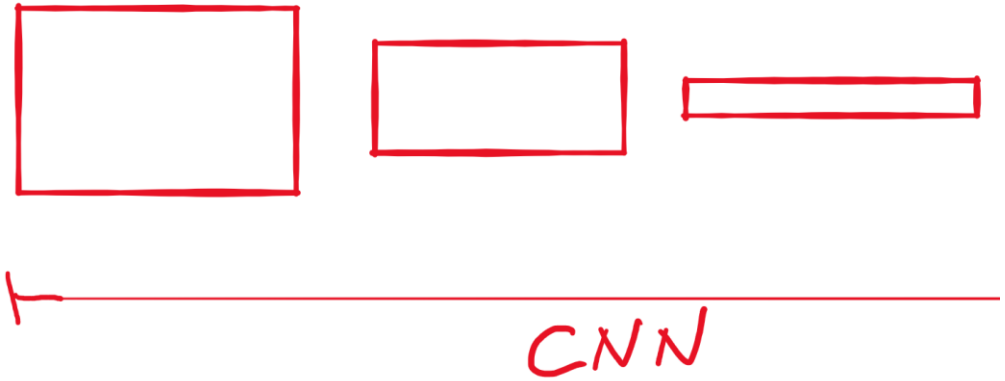
During training, the siamese network is fed with **pairs of images (I_i, I_j) that might refer or not to the same individual**

I



$f_W(I)$

T_2



$f_W(T_2)$

How to learn the parameters W **provided a training set** $\{I_i, i = 1 \ldots\}$ of many images each one annotated to the respective template?

Contrastive Loss

The *contrastive loss* function is defined as follow:

$$W = \underset{\omega}{\operatorname{argmin}} \sum_{i,j} \mathcal{L}_{\omega}(I_i, I_j, y_{i,j})$$

Where:

$$\mathcal{L}_{\omega}(I_i, I_j, y_{i,j}) = \frac{(1 - y_{i,j})}{2} \left\| f_{\omega}(I_i) - f_{\omega}(I_j) \right\|_2 + \frac{y_{i,j}}{2} \max\left(0, m - \left\| f_{\omega}(I_i) - f_{\omega}(I_j) \right\|_2\right)$$

- $y_{i,j} \in \{0,1\}$ is the label associate with the input pair (I_i, I_j):
  - 0 when (I_i, I_j) refers to the same person
  - 1 otherwise
- m is a hyperparameter indicating the margin we want (like in Hinge Loss)
- $\left\| f_{\omega}(I_i) - f_{\omega}(I_j) \right\|_2$ is the distance in the latent space.

Triplet Loss

A loss function such that a training sample I is compared against

- P a positive input, referring to the same person
- N a negative input, referring to a different person

We train the network to minimize the distance from the positive samples and maximize the distance from the negative ones

$$\mathcal{L}_\omega(I,P,N) = \max(0, m + (\|f_\omega(I) - f_\omega(P)\|_2 \ - \|f_\omega(I) - f_\omega(N)\|_2))$$

Triplet loss forces that a pair of samples from the same individual are smaller in distance than those with different ones.

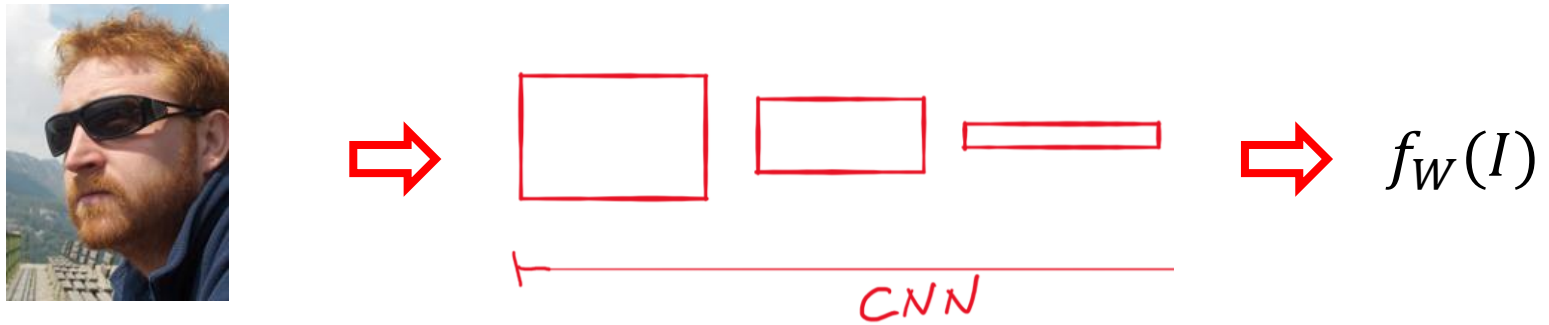
m always play the role of the margin.

The selection of triplets for training is a matter of study

Just to Recap

When a new image has to be verified

1. We feed the image I to the trained network, thus compute $f_W(I)$



2. Identify the person having average minimum distance from templates (in case there are many associated to the same individual)

$$\hat{\imath} = \underset{u}{\operatorname{argmin}} \frac{\sum_{T_{u,j}} \left\| f_W(I) - f_W(T_{u,j}) \right\|_2}{\#\{T_u\}}$$

3. Assess whether

$$\frac{\sum_{T_{i,j}} \left\| f_W(I) - f_W(T_{i,j}) \right\|_2}{\#\{T_{i,j}\}} < \gamma$$

is sufficiently small, otherwise **no identification**

Other decision rules can be adopted (e.g. searching for the person giving the a template with a minimum distance)

Self-Supervised Learning

What is Self-Supervision?

Self-supervised learning (SSL) *is a* ***paradigm*** *in machine learning where a* ***model is trained*** *on a task* ***using the data itself to generate supervisory signals****, rather than relying on externally-provided labels.*

[…] SSL ***leverages inherent structures*** *or relationships within the* ***input data*** *to create* ***meaningful training signals.***

The model learns in two steps.

- *First, the task is solved based on* ***an auxiliary or pretext classification task*** *using* ***pseudo-labels****, which help to initialize the model parameters.*
- *Next, the actual task is performed with* ***supervised*** *or unsupervised* ***learning****.*

https://en.wikipedia.org/wiki/Self-supervised_learning

What is Self-Supervision?

Self-supervised learning (SSL) *is a* ***paradigm*** *in machine learning where a* ***model is trained*** *on a task* ***using the data itself to generate supervisory signals****, rather than relying on externally-provided labels.*

[…] SSL ***leverages inherent structures*** *or relationships within the* ***input data*** *to create* ***meaningful training signals.***

SSL is general

It's on us to define how to extract supervisory signals from data!

https://en.wikipedia.org/wiki/Self-supervised_learning

What is Self-Supervision?

Self-supervised learning (SSL) *is a* ***paradigm*** *in machine learning where a* ***model is trained*** *on a task* ***using the data itself to generate supervisory signals****, rather than relying on externally-provided labels.*

[…] SSL ***leverages inherent structures*** *or relationships within the* ***input data*** *to create* ***meaningful training signals.***

The model learns in two steps.

- *First, the task is solved based on* ***an auxiliary or pretext classification task*** *using* ***pseudo-labels****, which help to initialize the model parameters.*
- *Next, the actual task is performed with* ***supervised*** *or unsupervised* ***learning****.*

https://en.wikipedia.org/wiki/Self-supervised_learning

What is Self-Supervision?

Self-supervised learning (SSL) *is a* ***paradigm*** *in machine learning where a* ***model is trained*** *on a task* ***using the data itself to generate supervisory signals****, rather than relying on externally-provided labels.*

[…] SSL ***leverages inherent structures*** *or relationships within the* ***input data*** *to create* ***meaningful training signals.***

The model learns in two steps.

- *First, the task is solved based on* ***an auxiliary or pretext classification task*** *using* ***pseudo-labels****, which help to initialize the model parameters.*
- *Next, the actual task is performed with* ***supervised*** *or unsupervised* ***learning****.*

https://en.wikipedia.org/wiki/Self-supervised_learning

SSL is not unsupervised learning

SSL is not solving the ultimate task. It is more about learning representations!

Why Self-Supervision?

- Transfer learning might not be a viable option, when the target domain is very different from natural images (where powerful models are trained).
- Expense of producing a new dataset for each new task.
- Some areas are supervision-starved, e.g.
  - In medical domain annotations are costly to obtain (require experts).
  - Segmentation masks are time consuming to obtain
  - In multimodal settings, annotations are even harder (e.g. in autonomous vehicles, 3D point clouds from LIDAR and associated images need to be pairwise annotated).
- There is an untapped availability of unlabeled images/videos.

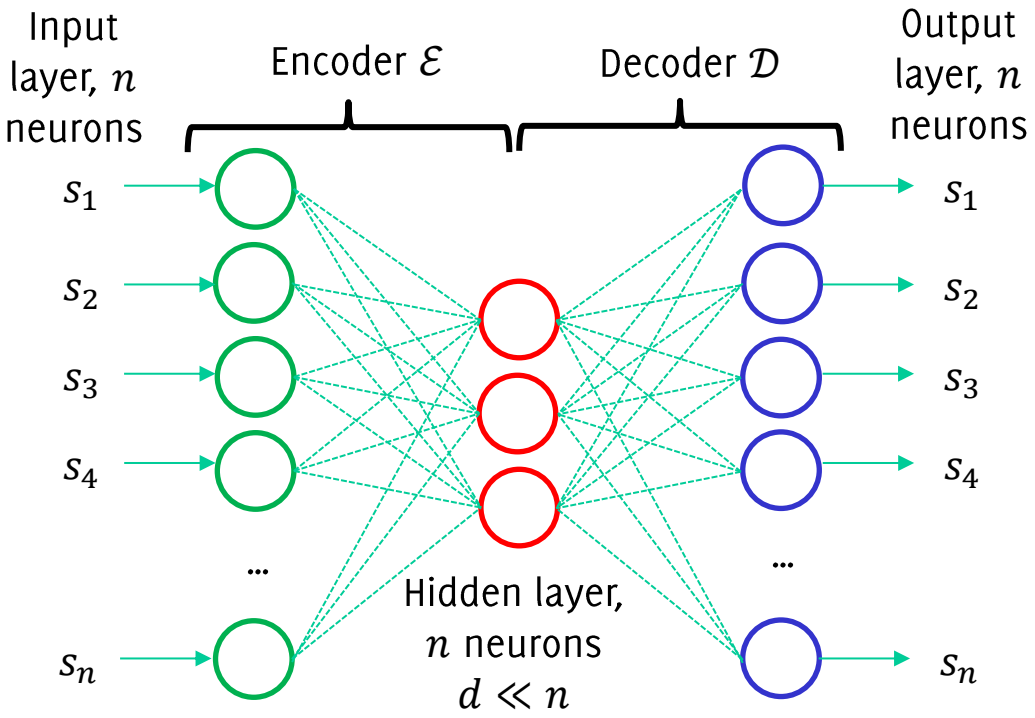
What's Wrong with AutoEncoders?

Autoencoders can be trained without any label!

- The supervisory signal and the auxiliary task is: learning how to reconstruct the input

Does AE learn the inherent structure of images?

In SSL the auxiliary tasks that promote learning semantic latent representations



Example of Supervisory Signals and Auxiliary Tasks

- Is NLP it is common to hide part of a sentence and predict the hidden words from the remaining words.
- In videos, predict past or future frames in a video (hidden data) from current ones (observed data). Or to determine which frame comes first.
- In images, the auxiliary task is often related to augmentation
  - Geometric transformations
  - Noise Addition
  - Colorize Grayscale Images
  - Estimate the relative position of patches
- **Contrastive learning** can be trained on pairs of augmented / not augmented versions of the same image.

These are all ways to force the network to learn «semantic features»

Published as a conference paper at ICLR 2018

UNSUPERVISED REPRESENTATION LEARNING BY PREDICTING IMAGE ROTATIONS

Spyros Gidaris, Praveer Singh, Nikos Komodakis
University Paris-Est, LIGM
Ecole des Ponts ParisTech
{spyros.gidaris,praveer.singh,nikos.komodakis}@enpc.fr

Geometric Image Transformations

Learn image features by training ConvNets to **regress the 2d rotation angle** that is applied to the image that it gets as input.

Which image has the correct rotation?

Generating supervisory signal is simple:

- Chose a rotation y at random and apply to an unlabeled image I_y.
- The annotated sample is (I_y, y)…. and it's free!

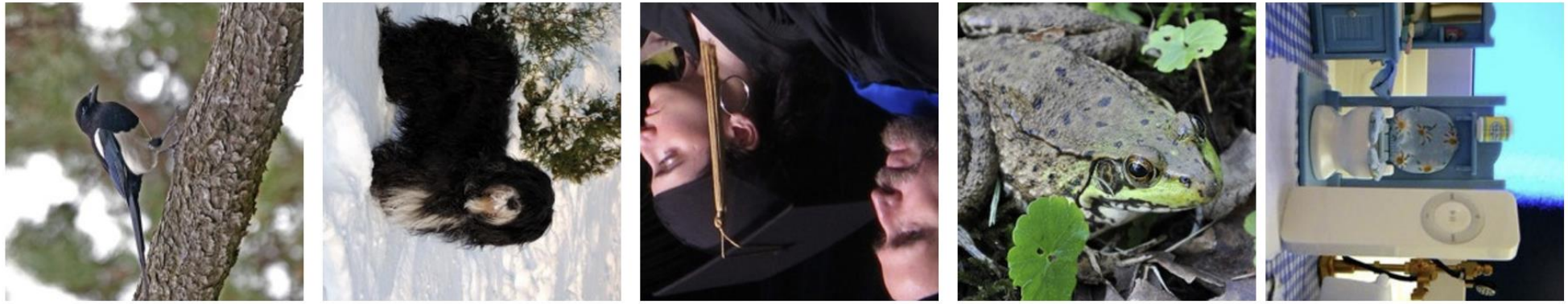
Spyros Gidaris, Praveer Singh, Nikos Komodakis, "Unsupervised representation learning by predicting image rotations", ICLR 2018

A Few More Details

Image transformations are in a **discrete set:** 0, 90, 180, 270 rotations (these rotation introduces no artifacts that the network can these trivial features to solve the task)

- This is casted as a 4-class classification problem for the CNN
- The network is trained to minimize categorical cross-entropy

This task forces the network to understand the concept of the objects, such as their location inthe image, their type, and their pose.

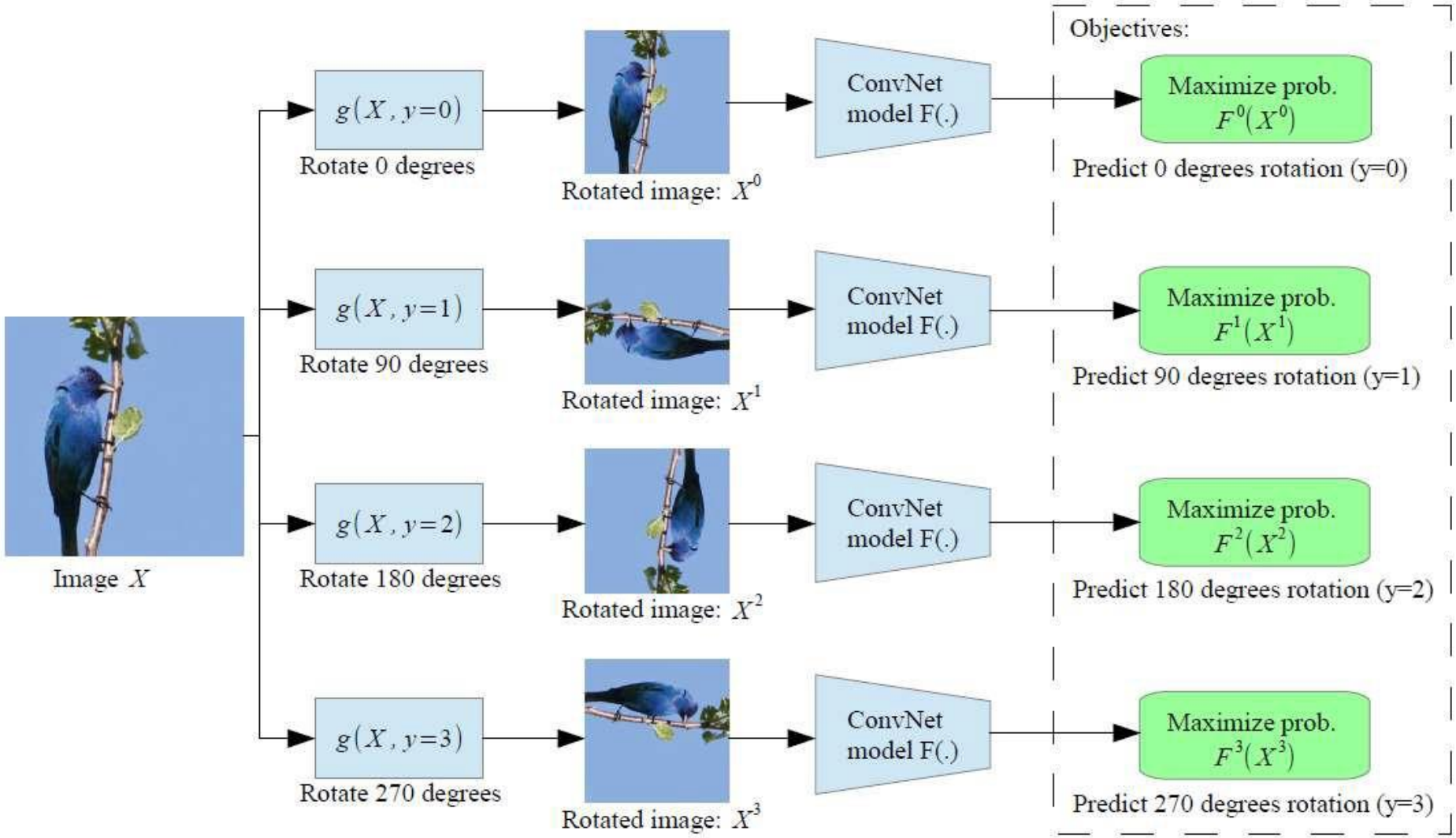


90° rotation 270° rotation 180° rotation 0° rotation 270° rotation

it is essentially impossible for a ConvNet model to effectively perform the above rotation recognition task unless it has first learnt to recognize and detect classes of objects as well as their semantic parts in images.

Spyros Gidaris, Praveer Singh, Nikos Komodakis, "Unsupervised representation learning by predicting image rotations", ICLR 2018

Image Transformations – 2018



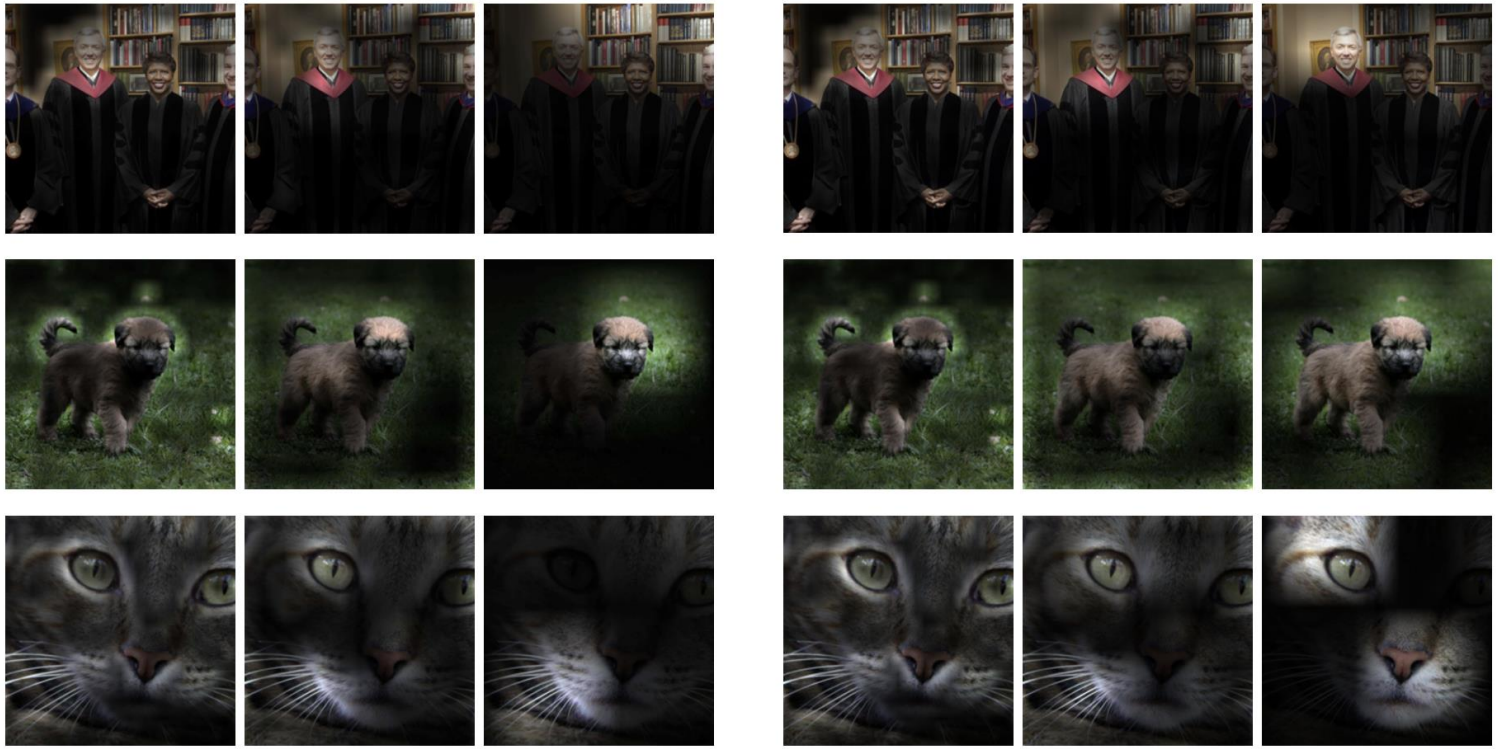
Spyros Gidaris, Praveer Singh, Nikos Komodakis, "Unsupervised representation learning by predicting image rotations", ICLR 2018

Attention Maps of Supervised vs a Self-Supervised CNN

Attention maps are computed based on the magnitude of activations at each spatial cell of a convolutional layer.

Activation maps == the energy in specific neurons. CAMS are not viable for this comparison as they depend on the downstream task.

Input images on the models

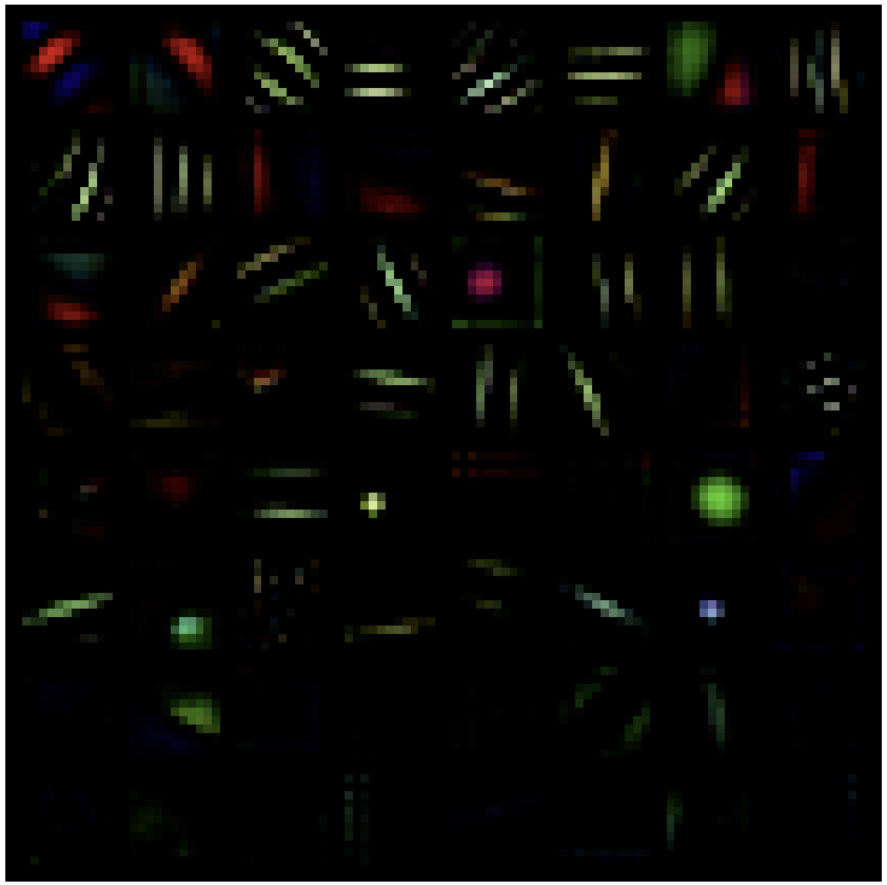


Conv1 27 × 27 Conv3 13 × 13 Conv5 6 × 6 Conv1 27 × 27 Conv3 13 × 13 Conv5 6 × 6

(a) **Attention maps of supervised model** (b) **Attention maps of our self-supervised model**

Spyros Gidaris, Praveer Singh, Nikos Komodakis, "Unsupervised representation learning by predicting image rotations", ICLR 2018

Filters learned by AlexNet trained a Supervised vs Self-Supervised Task



(a) **Supervised**

(b) **Self-supervised to recognize rotations**

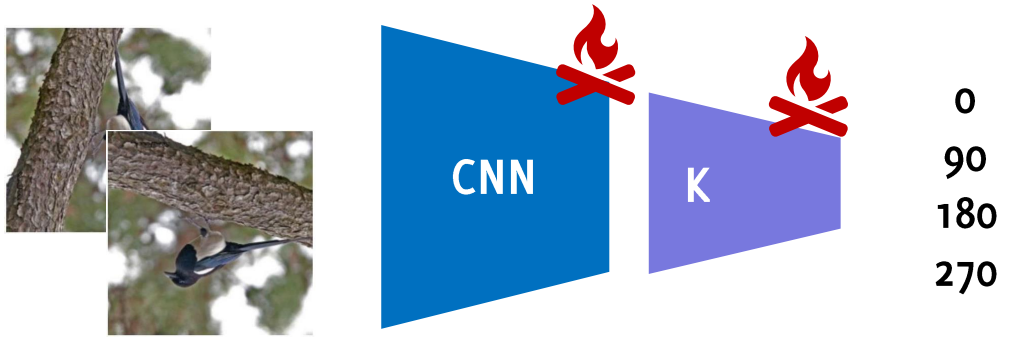
We observe that the filters learned by the self-supervised task are mostly oriented edge filters on various frequencies and, remarkably, they seem to have more variety than those learned on the supervised task.

Spyros Gidaris, Praveer Singh, Nikos Komodakis, "Unsupervised representation learning by predicting image rotations", ICLR 2018

SSL for Network Pretraining + Downstream Task Fine Tuning

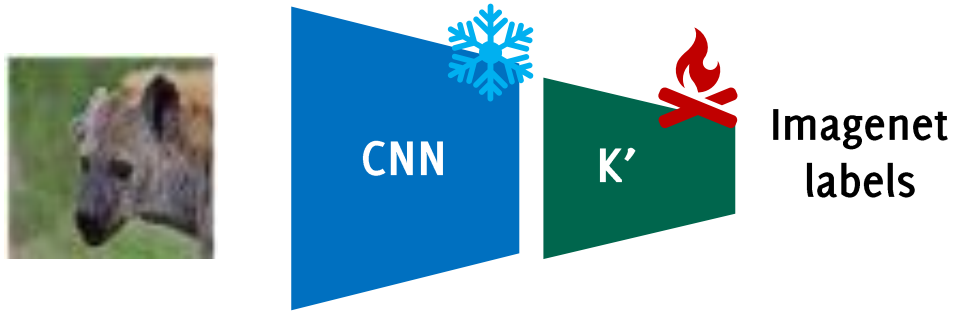
According to the standard SSL practice

- Pre-train the network on SSL to predict geometric transformations



- Fine tune the backbone for a downstream task, like
  - Image classification
  - Object detection

by adding suitable output layers



Spyros Gidaris, Praveer Singh, Nikos Komodakis, "Unsupervised representation learning by predicting image rotations", ICLR 2018

Choosing the right auxiliary task

Table 2: Exploring the quality of the self-supervised learned features w.r.t. the number of recognized rotations. For all the entries we trained a non-linear classifier with 3 fully connected layers (similar to Table 1) on top of the feature maps generated by the 2nd conv. block of a RotNet model with 4 conv. blocks in total. The reported results are from CIFAR-10.

# Rotations	Rotations	CIFAR-10 Classification Accuracy
4	0°, 90°, 180°, 270°	**89.06**
8	0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°	88.51
2	0°, 180°	87.46
2	90°, 270°	85.52

- Rotation is followed by cropping, when needed.
- 8 rotations can lead to artifacts that steer the network to non meaningful learning
- 90°, 270° is worst than 0°,180° as the former never sees the testing conditions (0°)

Spyros Gidaris, Praveer Singh, Nikos Komodakis, "Unsupervised representation learning by predicting image rotations", ICLR 2018

SSL for Network Pretraining + Downstream Task Fine Tuning

- Supervised NiN and RotNet+ conv has the same architecture
- Supervised NIN: 3 Layers trained in a supervised manner
- RotNet+conv : 2 layers trained Self-supervised and then frozen, 1 in a Supervised. (in fine-tuned variant also the first 2 are trained supervised)
- Random + conv: weights are frozen to a random initialization, the last conv trained

SSL almost pairs Supervised learning

The **GAP between random initialization** and RotNet initialization demonstrates the benefits of SSL

Method	Accuracy
Supervised NIN	92.80
Random Init. + conv	72.50
(Ours) RotNet + non-linear	89.06
(Ours) RotNet + conv	**91.16**
(Ours) RotNet + non-linear (fine-tuned)	91.73
(Ours) RotNet + conv (fine-tuned)	92.17
Roto-Scat + SVM Oyallon & Mallat (2015)	82.3
ExemplarCNN Dosovitskiy et al. (2014)	84.3
DCGAN Radford et al. (2015)	82.8
Scattering Oyallon et al. (2017)	84.7

Spyros Gidaris, Praveer Singh, Nikos Komodakis, "Unsupervised representation learning by predicting image rotations", ICLR 2018

ImageNet Classification Experiments

- AlexNet architecture, trained on ImageNet
- Results are consistent with the finding on CIFAR
- Better than Self-Supervised alternatives

	Method	Conv1	Conv2	Conv3	Conv4	Conv5
	ImageNet labels	19.3	36.3	44.2	48.3	50.5
	Random	11.6	17.1	16.9	16.3	14.1
	Random rescaled Krähenbühl et al. (2015)	17.5	23.0	24.5	23.2	20.6
Self-Supervised alternatives	Context (Doersch et al., 2015)	16.2	23.3	30.2	31.7	29.6
	Context Encoders (Pathak et al., 2016b)	14.1	20.7	21.0	19.8	15.5
	Colorization (Zhang et al., 2016a)	12.5	24.5	30.4	31.5	30.3
	Jigsaw Puzzles (Noroozi & Favaro, 2016)	18.2	28.8	34.0	33.9	27.1
	BIGAN (Donahue et al., 2016)	17.7	24.5	31.0	29.9	28.0
	Split-Brain (Zhang et al., 2016b)	17.7	29.3	35.4	35.2	32.8
	Counting (Noroozi et al., 2017)	18.0	30.6	34.3	32.5	25.7
	(Ours) RotNet	**18.8**	**31.7**	**38.7**	**38.2**	**36.5**

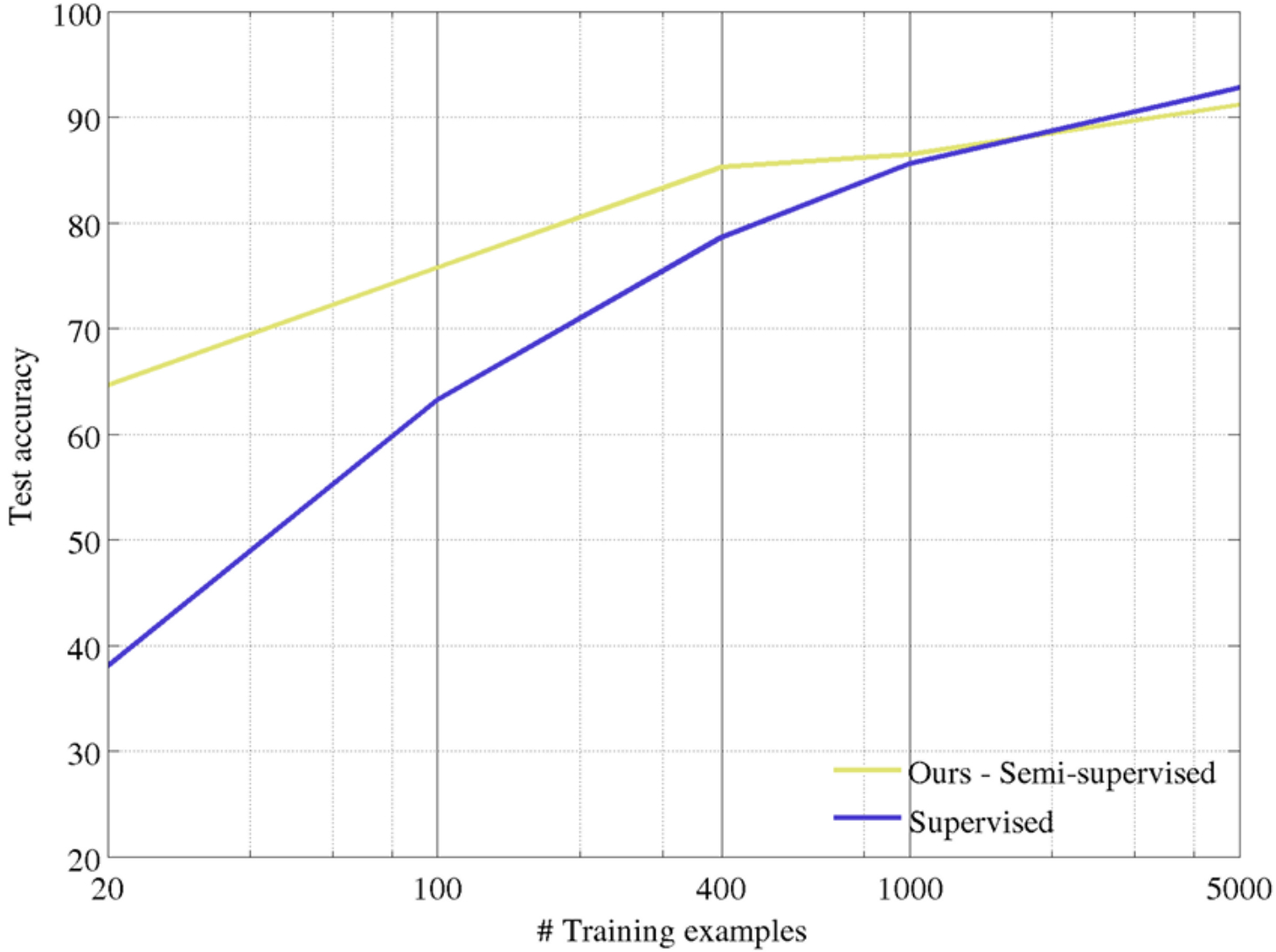
Object Detection and Segmentation experimtns

Same reults on the Pascal-VOC 2007 detection and Pascal VOC 2012 segmentation

	Classification (%mAP)		Detection (%mAP)	Segmentation (%mIoU)
Trained layers	fc6-8	all	all	all
ImageNet labels	78.9	79.9	56.8	48.0
Random		53.3	43.4	19.8
Random rescaled Krähenbühl et al. (2015)	39.2	56.6	45.6	32.6
Egomotion (Agrawal et al., 2015)	31.0	54.2	43.9	
Context Encoders (Pathak et al., 2016b)	34.6	56.5	44.5	29.7
Tracking (Wang & Gupta, 2015)	55.6	63.1	47.4	
Context (Doersch et al., 2015)	55.1	65.3	51.1	
Colorization (Zhang et al., 2016a)	61.5	65.6	46.9	35.6
BIGAN (Donahue et al., 2016)	52.3	60.1	46.9	34.9
Jigsaw Puzzles (Noroozi & Favaro, 2016)	-	67.6	53.2	37.6
NAT (Bojanowski & Joulin, 2017)	56.7	65.3	49.4	
Split-Brain (Zhang et al., 2016b)	63.0	67.1	46.7	36.0
ColorProxy (Larsson et al., 2017)		65.9		38.4
Counting (Noroozi et al., 2017)	-	67.7	51.4	36.6
(Ours) RotNet	**70.87**	**72.97**	**54.4**	**39.1**

Semi-Supervised Learning Settings

Training using Self-Supervised Learning the entire CIFAR dataset, and fine tuning the classification head using only a fraction of labeled training sample (per class). Cifar 10 experiments



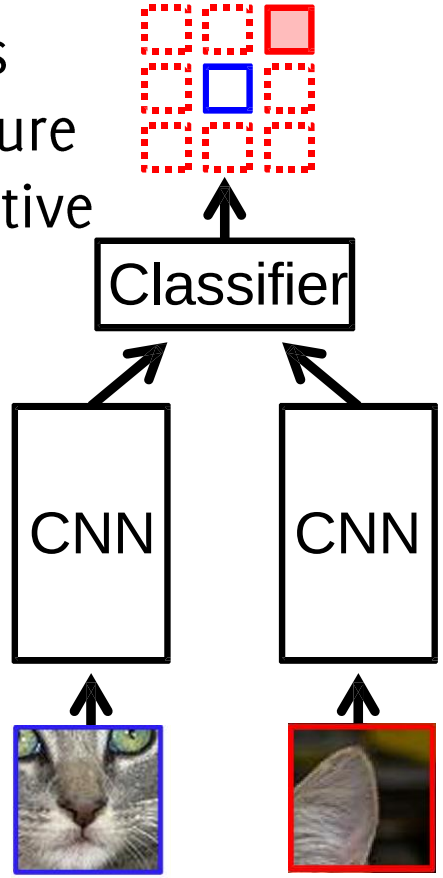
In semi-supervised learning, when there are little annotated data, SSL is better

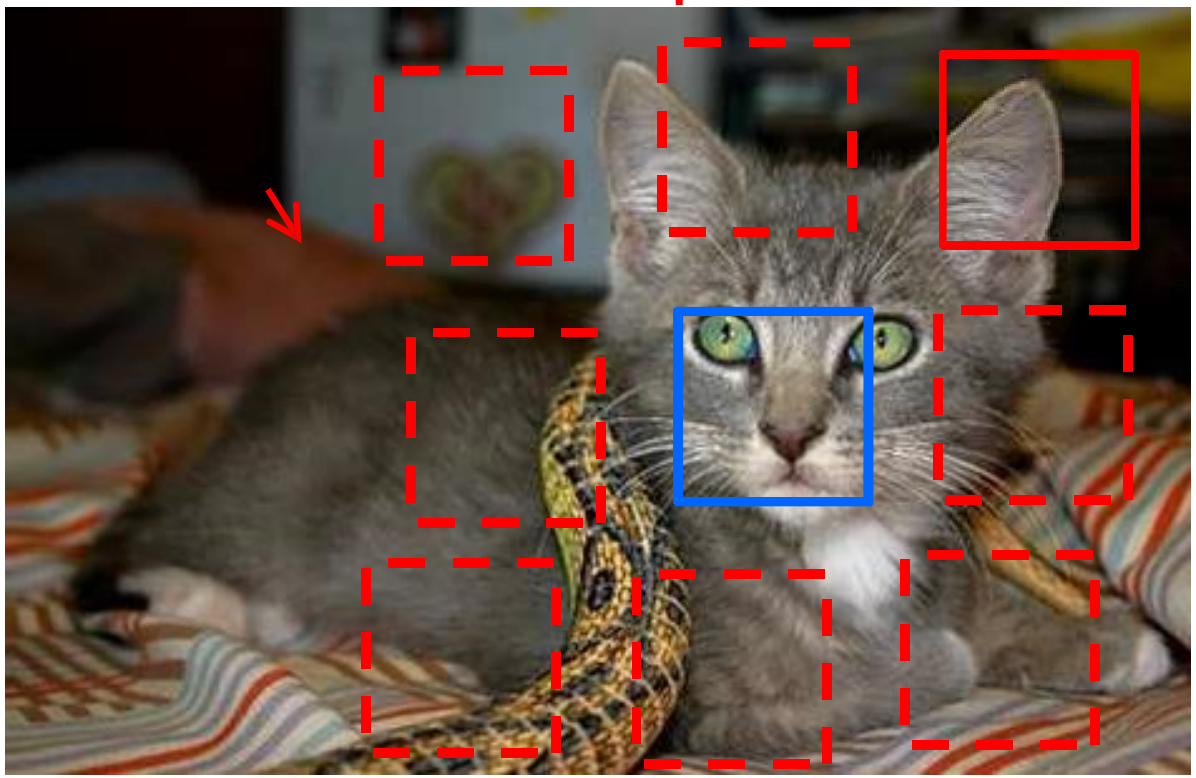
Other Self-Supervised Learning Tasks

Example: relative positioning

Train network to predict relative position of two regions in the same image

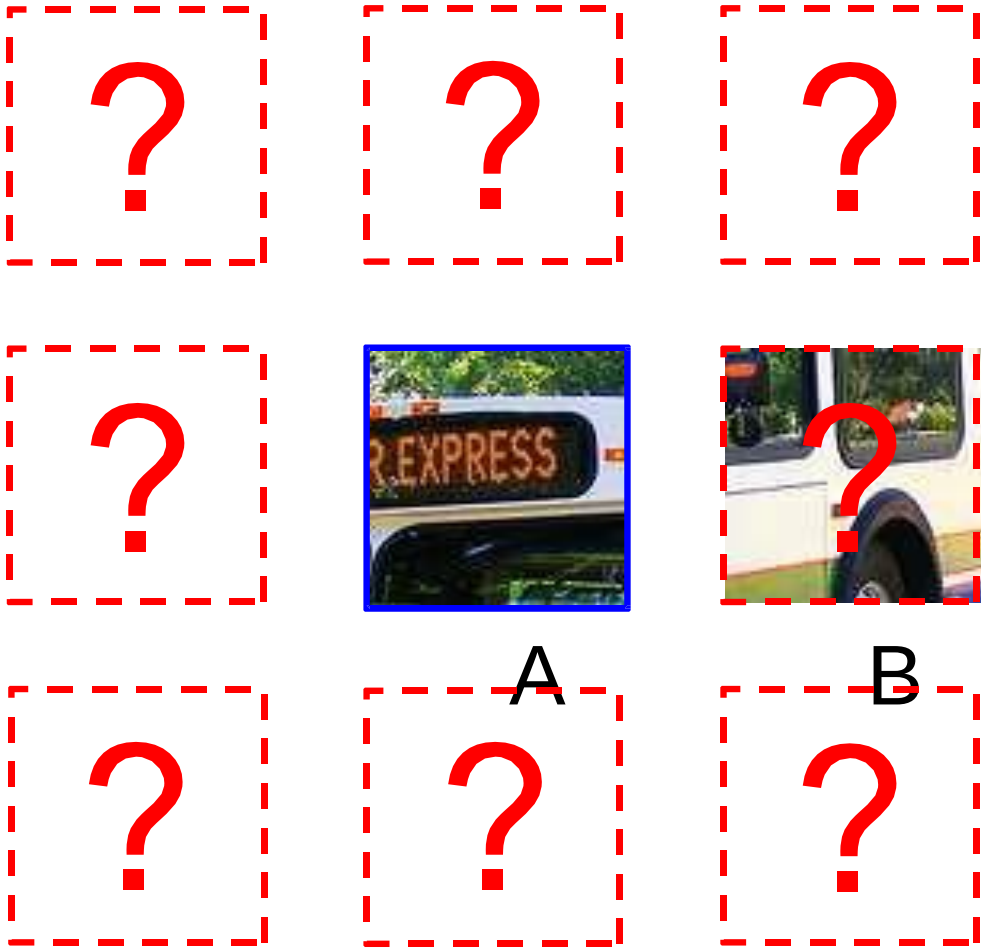
1. Randomly sample a patch (the blue one)
2. Randomly sample a patch around
3. Feed the two patches to a siames architecture and predict their relative position





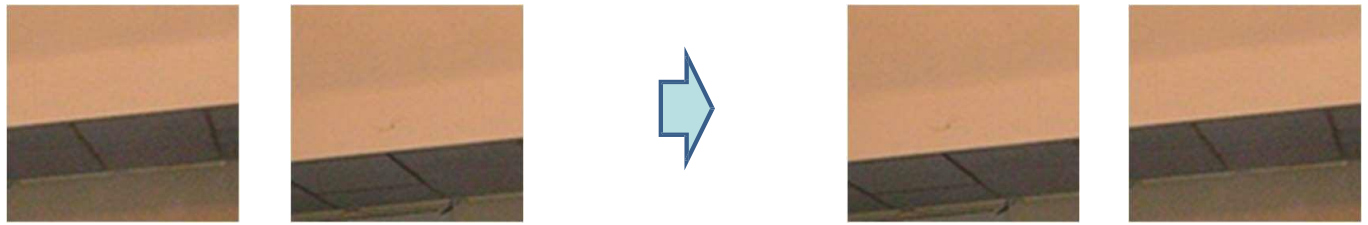
Carl Doersch, Abhinav Gupta, Alexei A. Efros "Unsupervised visual representation learning by context prediction", ICCV 2015

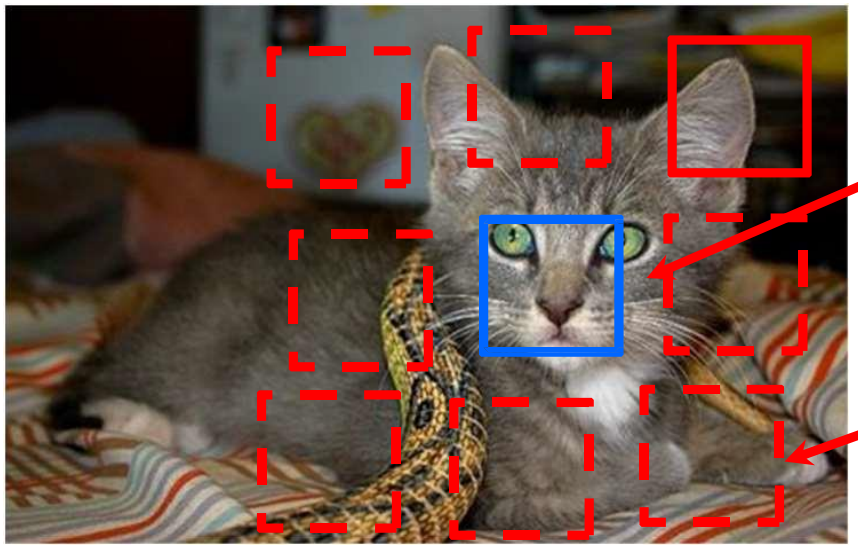
Example: relative positioning



Unsupervised visual representation learning by context prediction,
Carl Doersch, Abhinav Gupta, Alexei A. Efros, ICCV 2015

Avoiding Trivial Shortcuts





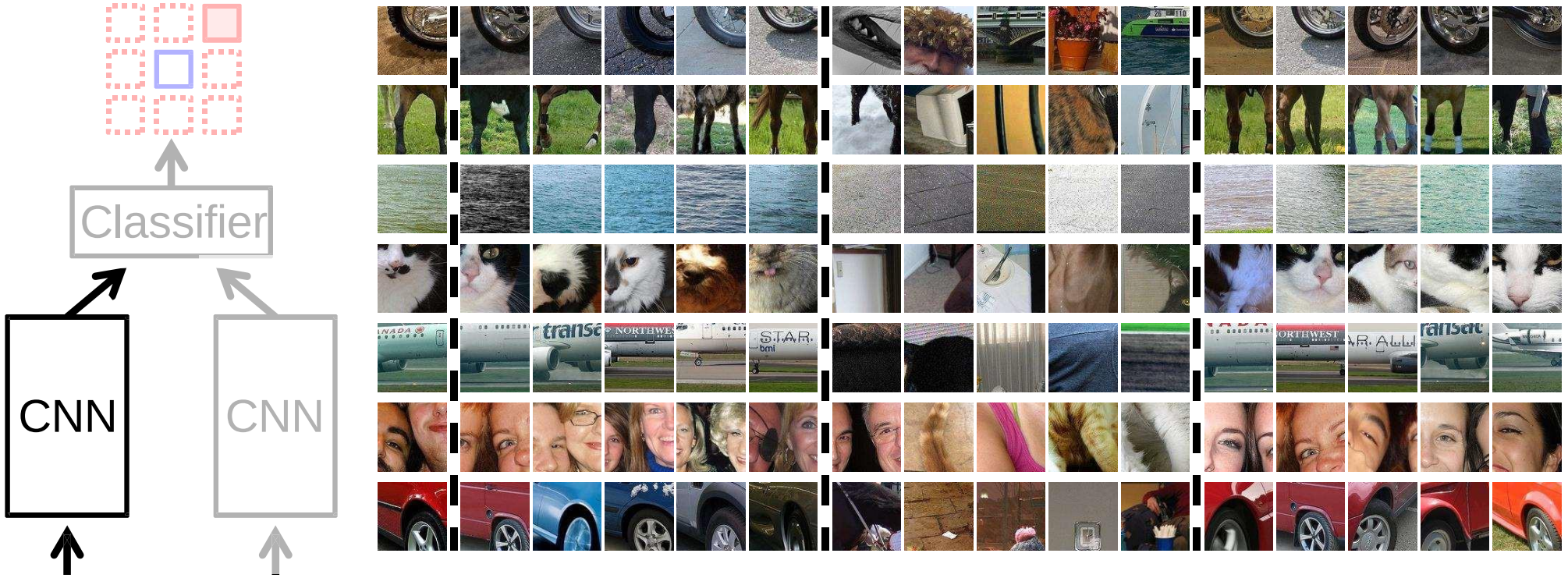
Include a gap

Jitter the patch locations

What is learned?

Patch custered according to their similarity in latent space (normalized correlation)

Display nearest neighbors.

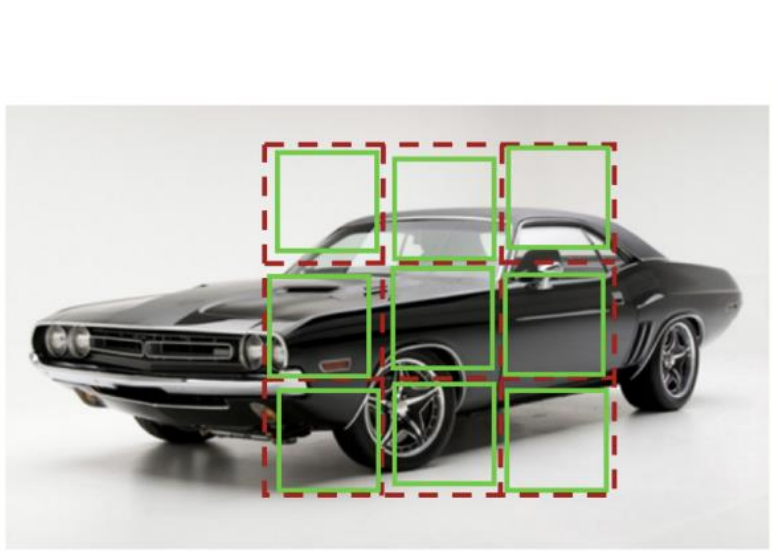


Carl Doersch, Abhinav Gupta, Alexei A. Efros “Unsupervised visual representation learning by context prediction”, ICCV 2015

Auxiliary task: solve a Jigsaw puzzle.

Classification task: predict the permutation among a pre-defined set

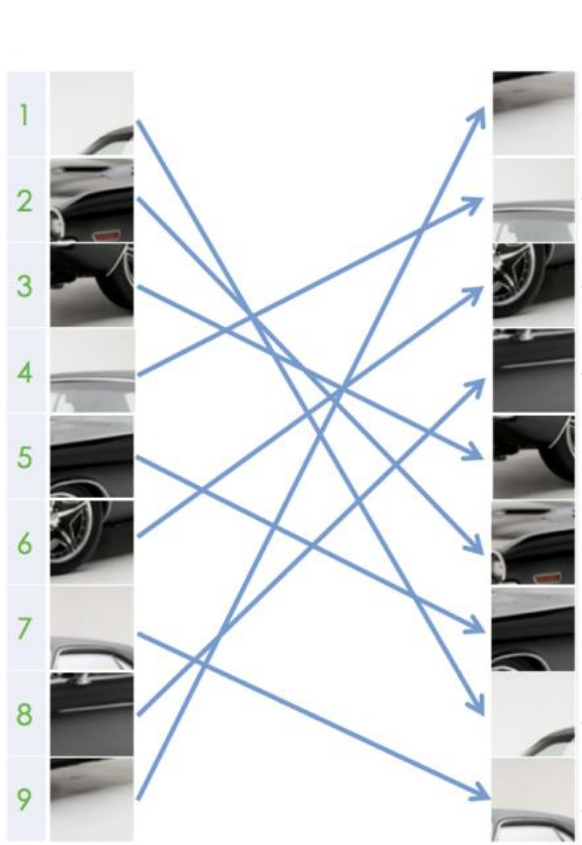
Architecture: context-free network (CFN) a **Siamese-like network** with as many branches as the input patches. Each patch is processed separately until the dense layers.

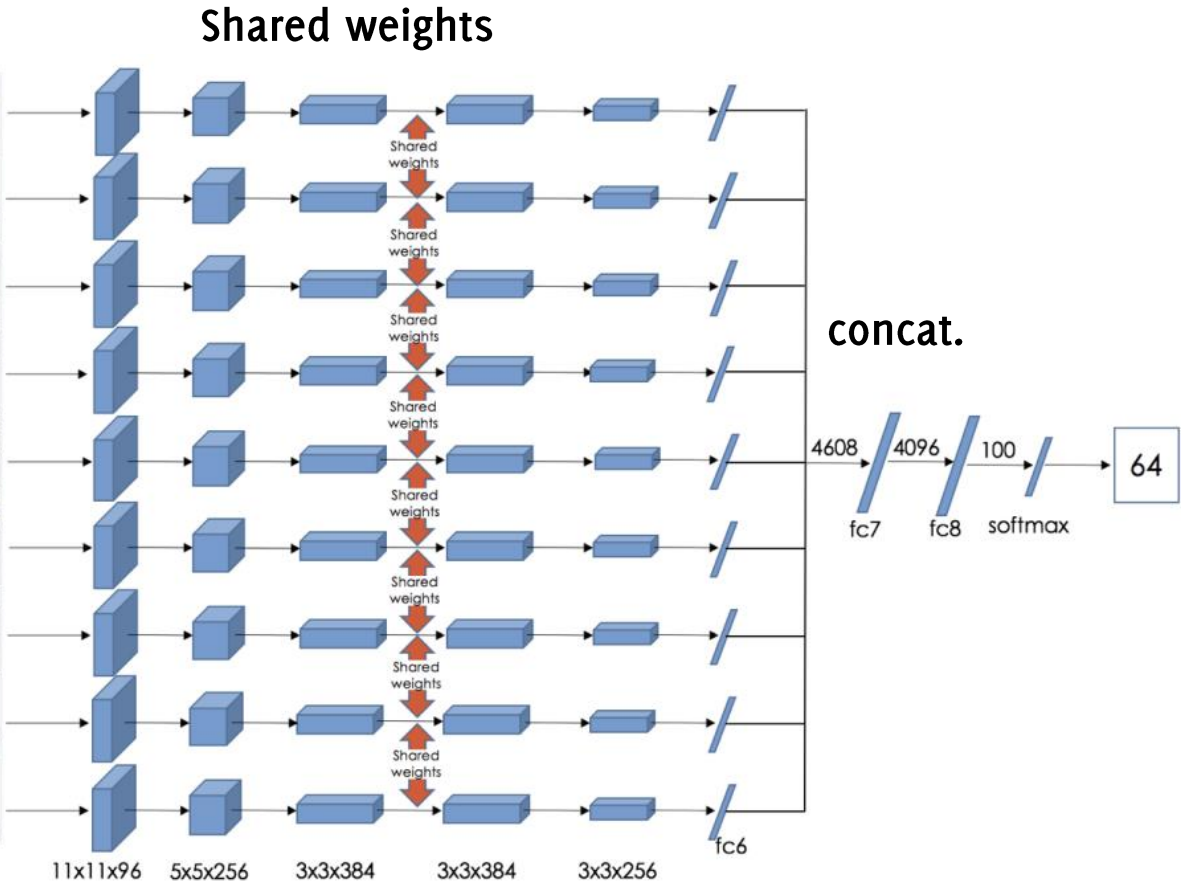


Permutation Set

index	permutation
64	9,4,6,8,3,2,5,1,7

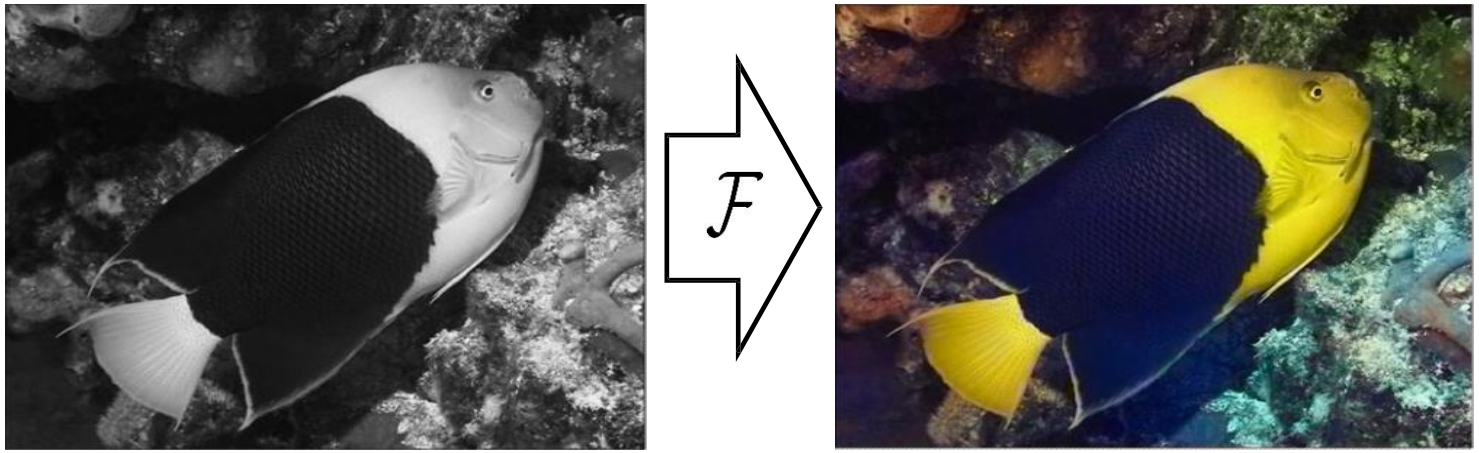
Reorder patches according to the selected permutation





Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles., ECCV 2016

Colourization: Train network to predict pixel colour from a monochrome input



Grayscale image: *L* channel

$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$

Concatenate (L,ab)

$$(\mathbf{X}, \hat{\mathbf{Y}})$$

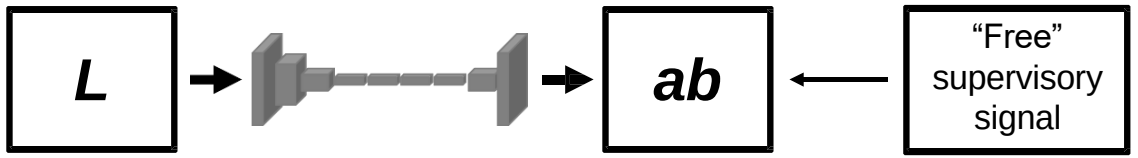
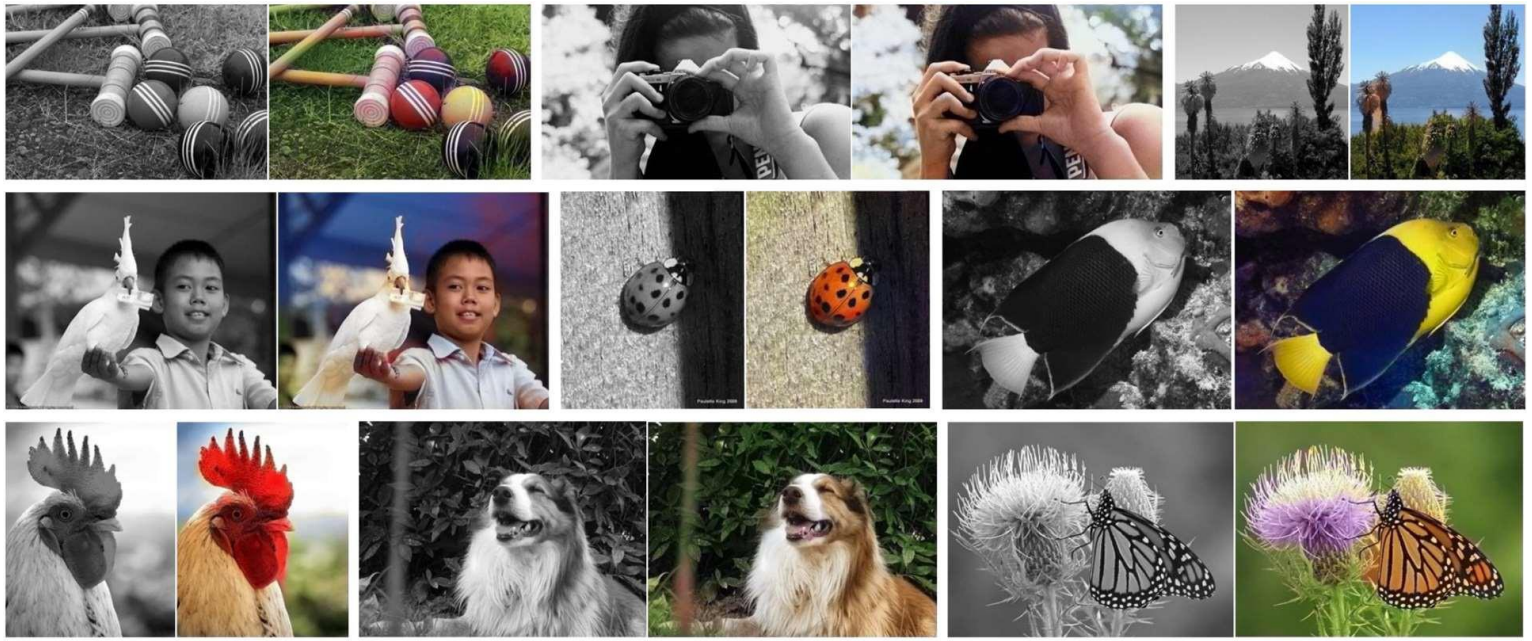


Image example II: colourization

Train network to predict pixel colour from a monochrome input



Colorful Image Colorization, Zhang *et al*., ECCV 2016

Exemplar networks

Create a self-supervised training set of surrogate classes

A **single** “seed” image generates the training set for a “surrogate class”

- cropped in patches in high-gradient region
- augmented by geometric and photometric transformations

A classifier is trained to distinguish between surrogate classes

The learned representation becomes invariant to these transformations

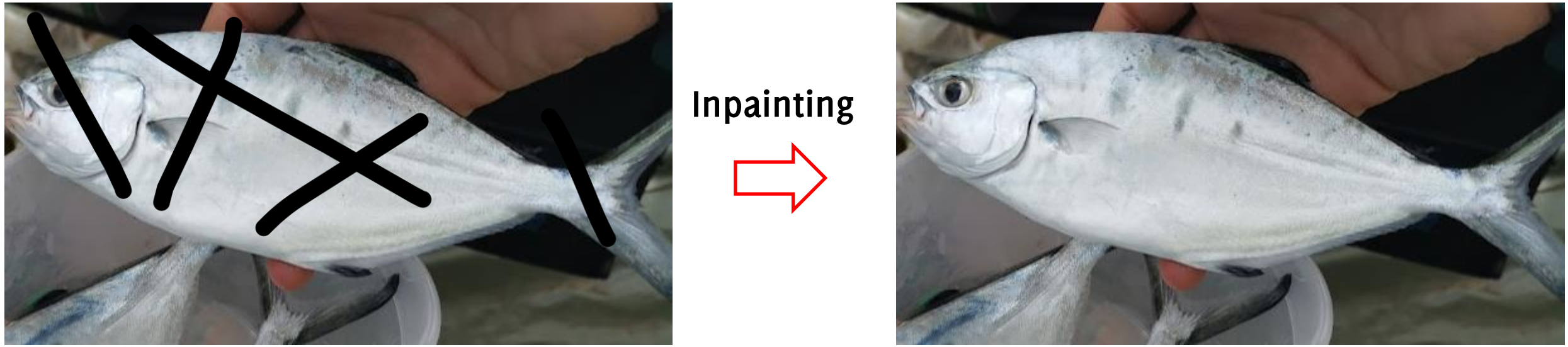
Alexey, D., Fischer, P., Tobias, J., Springenberg, M. R., & Brox, “Discriminative unsupervised feature learning with exemplar convolutional neural networks”. TPAMI 2016

Inpainting as a form of Self-Supervision

Image inpainting is the problem of reconstructing missing or damaged areas of digital photographs and videos

A very challenging inverse problem where **Deep Autoencoders excel**

Training an autoencoder performing inpainting **does not require supervision**



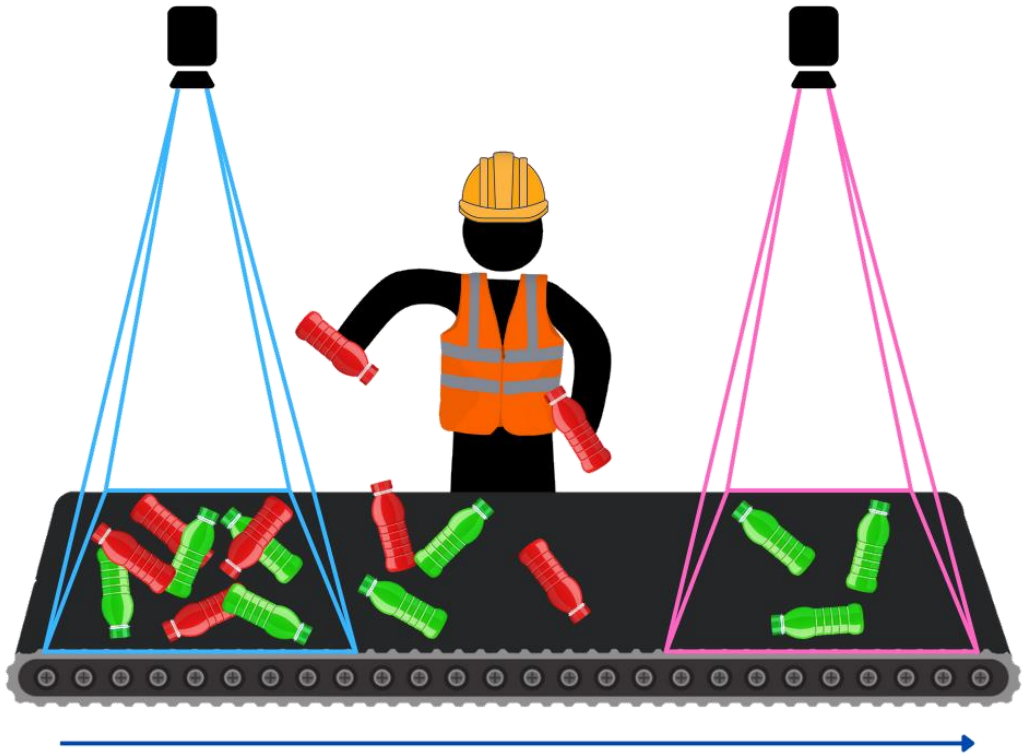
Zavrtanik, Vitjan, Matej Kristan, and Danijel Skočaj. "Reconstruction by inpainting for visual anomaly detection." *Pattern Recognition* 112 (2021): 107706.

Before/After Self-Supervision

Slide Credits Andrea Marelli

Before/After Self-Supervision:

How we can leverage visual differences between images of the same items collected before and after specific trasformations?



Unwanted item

Unwanted item

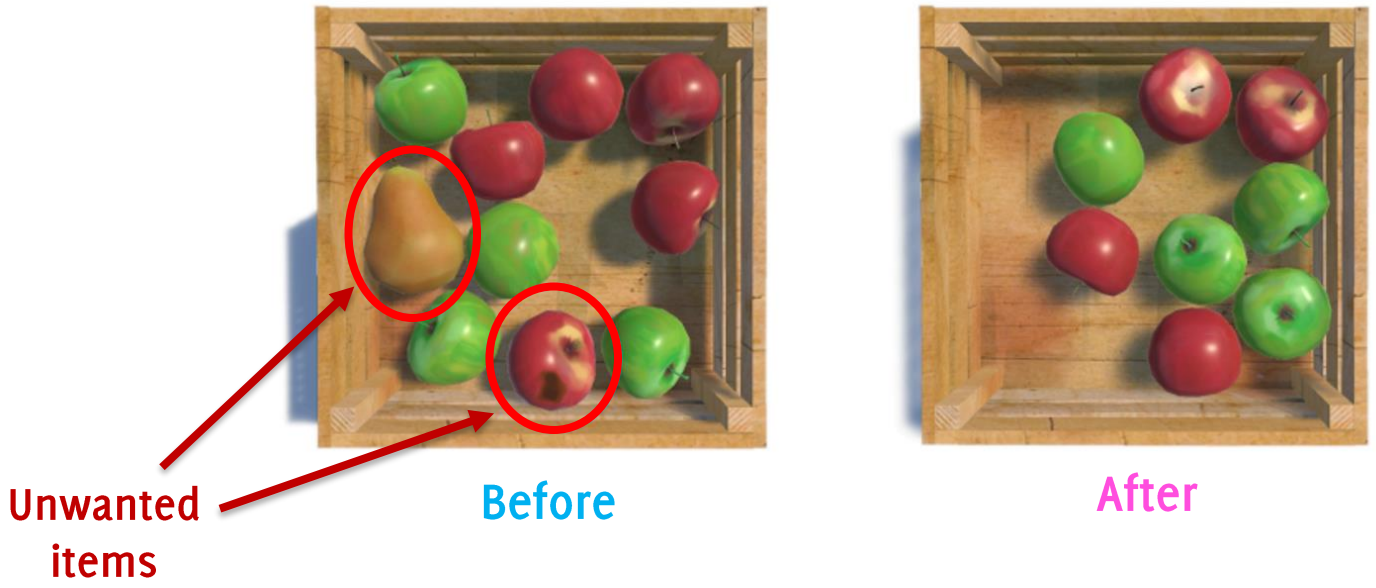
Marelli A., Magri G., Arrigoni F., Boracchi G. “Temporal-consistent CAMs for Weakly Supervised Video Segmentation in Waste Sorting”, ECCV 2024

Before/After Self-Supervision:

In many assembly lines human operators modify a set of objects by **removing or fixing anomalus elements.**

Let's acquire videos before and after the manual operation:

- Before images contain:
  - Anomalous item
  - Non-anomalous items
- After images contain
  - Non-anomalous item.



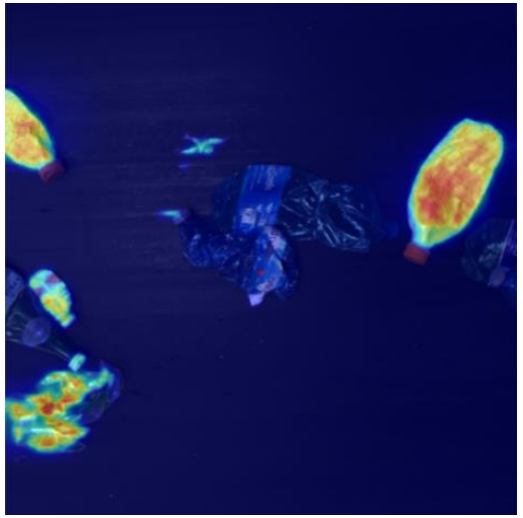
After images contain objects that are also in before ones, during the manual intervetion, **the operator moves also the non-anomalus objects, modifing their displacement.**

Marelli A., Magri G., Arrigoni F., Boracchi G. "Temporal-consistent CAMs for Weakly Supervised Video Segmentation in Waste Sorting", ECCV 2024

Before/After Self-Supervision:

We train a classifier to recognize before and after images, the classifier learns that an image is a "before" one if it contains illegal objects.

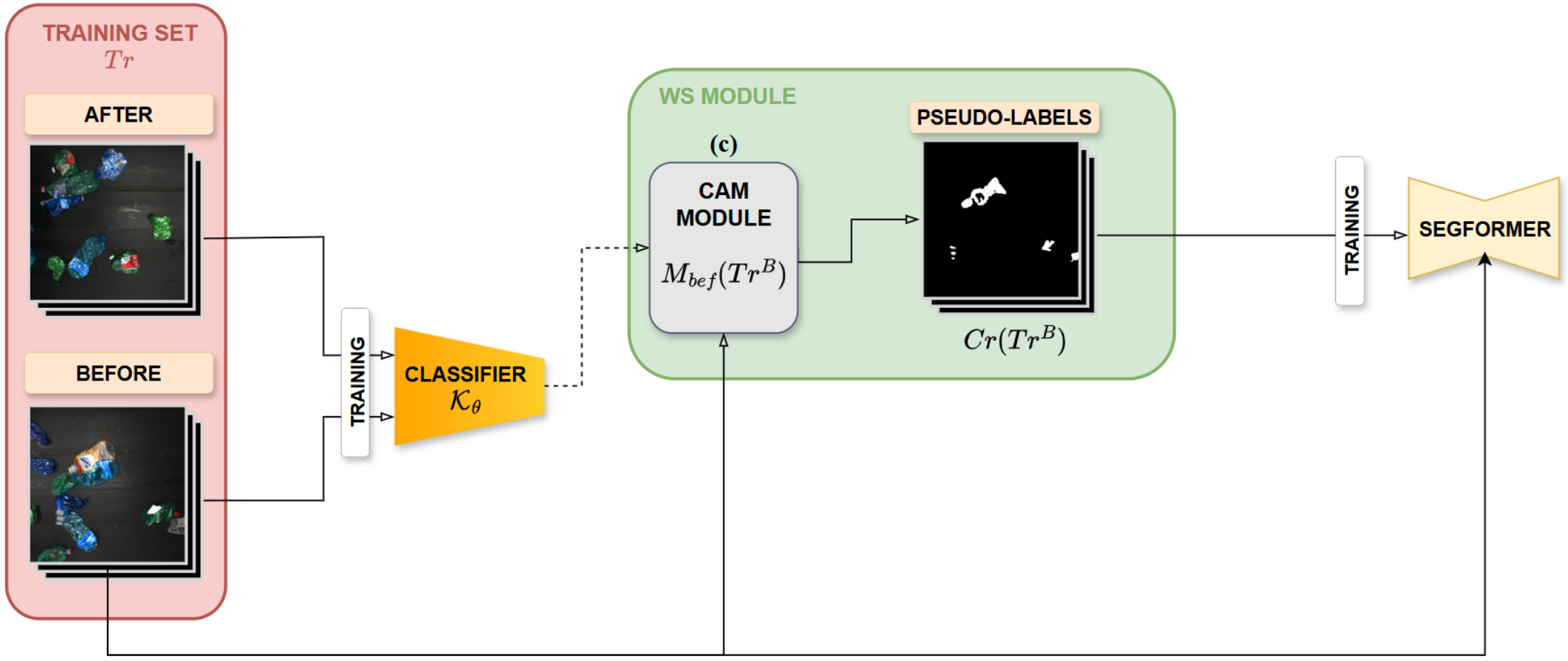
Then, we employ the classifer to generate **Class Activation Maps to highlight the most relevant regions** to classify an image as **"before"**, which correspond to the ones **where anomalous objects are located.**



Bashkirova D., Abdelfattah M., Zhu Z., Akl J. "ZeroWaste: Towards Deformable Object Segmentation in Cluttered Scenes", CVPR 2022

Before/After Self-Supervision:

The CAMs generated by the auxiliary classifier can then be used to train a fully segmentation network



Bashkirova D., Abdelfattah M., Zhu Z., Akl J. "ZeroWaste: Towards Deformable Object Segmentation in Cluttered Scenes", CVPR 2022

Before/After Dataset:

Our dataset was collected in a Seruso s.p.a, a plastic sorting facility located in Verderio (LC), Italy. Paper currently under review, soon publicly available

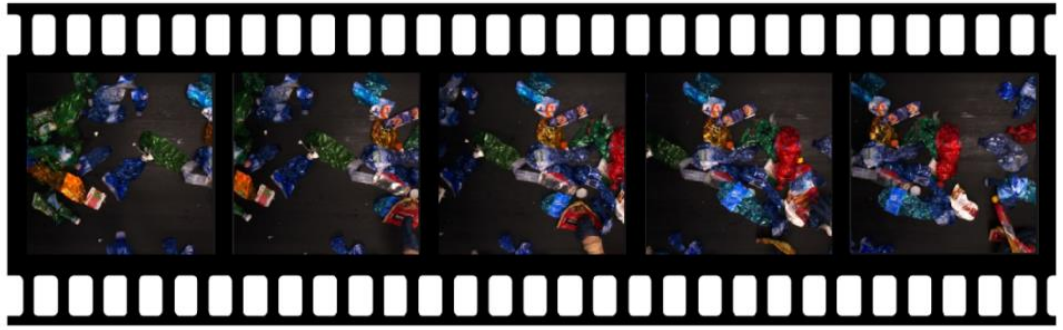
11060 images

- 284 video sequences for the before class.
- 313 video sequences for the after class

- **Training**
  - 4851 before images
  - 4712 after images

- **Test** (pixel-level annotated)
  - 496 before images
  - 1001 after images

Before video



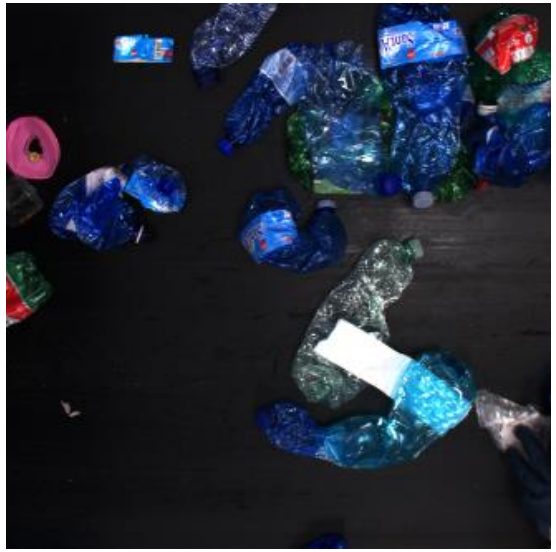
After video

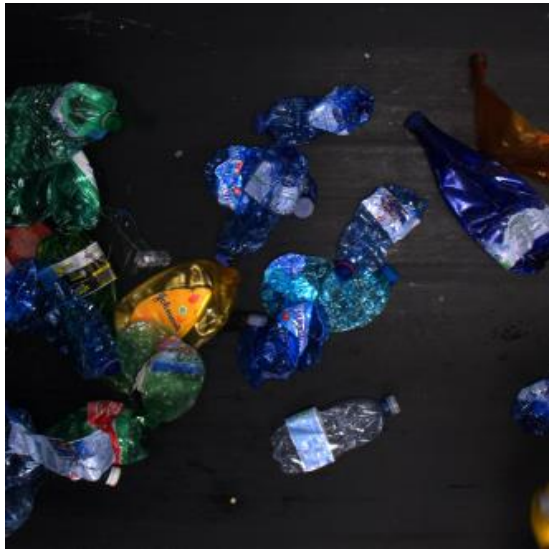
Marelli A., Foresti A., Pesce L., Grosso M., Boracchi G. “WS2: Weakly Supervised segmentation for smarter Waste Sorting”, ICIP 2025

Before/After Dataset: challenges

Images of the same class are captured under the same lighting conditions, so they have similar background. The risk is that **the classifier learns to recognize the class of the image on the basis of the background.**

Before images





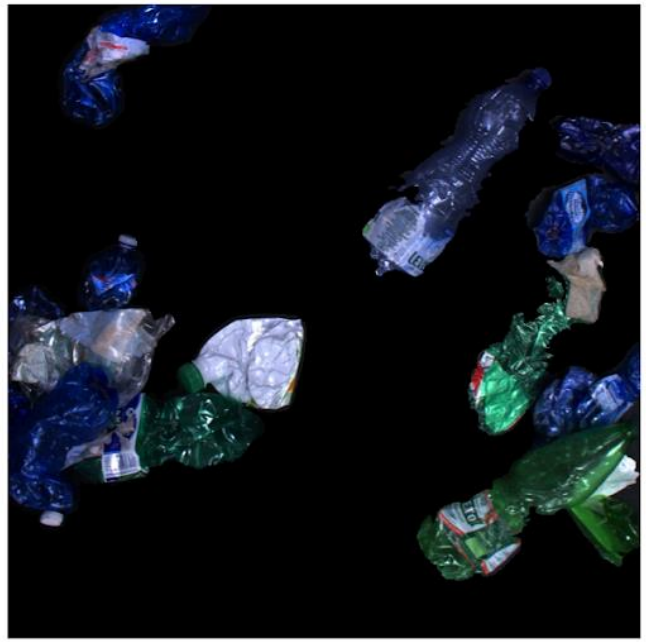
After images

Marelli A., Foresti A., Pesce L., Grosso M., Boracchi G. “WS2: Weakly Supervised segmentation for smarter Waste Sorting”, ICIP 2025

Before/After Dataset: challenges

To avoid this, we **removed the background** from the images and **collected them in a new class** so to **drive the classifier to focus only on the important features per any class.**

Original image

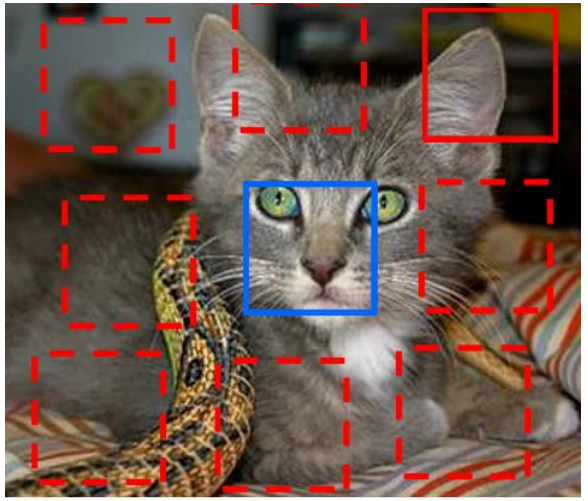


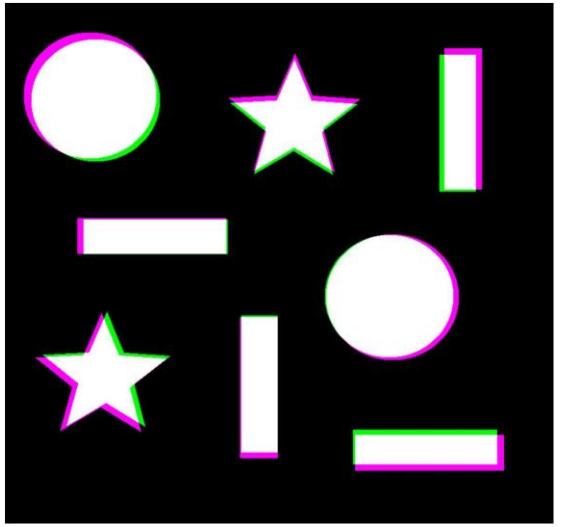
No background image

Only background image

Marelli A., Foresti A., Pesce L., Grosso M., Boracchi G. “WS2: Weakly Supervised segmentation for smarter Waste Sorting”, ICIP 2025

Avoiding Learning Shortcuts





In Self-supervised learning the learning task has not to be trivial.

In **relative positioning** methods:

- disjoint patches are selected, to prevent learning matching boundaries
- Make sure there is not chromatic aberration

In **rotation-based** methods:

- prevent rotations inducing artifacts
- Perform cropping

Background in before/after settings

CNN for Image Denoising

Credits Edoardo Peretti for helping out in the literature survey

G. Boracchi

Observation Model

$$z(x) = y(x) + \eta(x)\,, x \in X$$

$z : X \rightarrow \mathbb{R}$ observed noisy image

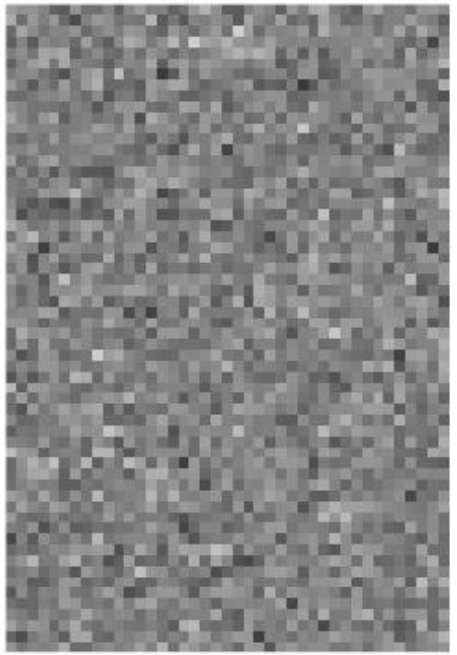
y: $X \rightarrow \mathbb{R}$ unknown original image (grayscale)

η: $X \rightarrow \mathbb{R}$ i.i.d. Gaussian white noise, $\eta \sim N(0, \sigma^2)$

.

z $=$

y $+$



η

Goal of image denoising

The purpose of any **denoising** algorithm is to provide $\hat{y}$, an estimate of the original image y.

z

$\hat{y}$

Goal of image denoising

And the same works for RGB images

z

$\hat{y}$

Image Denoising

Denoising plays a crucial role in many stages of imaging pipelines:

- **Preprocessing**: to enhance output quality and improve the effectiveness of subsequent algorithms
- **Post-processing:** to remove compression artifacts (e.g. blocks and ringing)
- **Plug-and-play filter:** as an implicit regularization prior in various imaging applications

Katkovnik, V., Foi, A., Egiazarian, K., & Astola, J.*From local kernel to nonlocal multiple-model image denoising*. International journal of computer vision, 2010

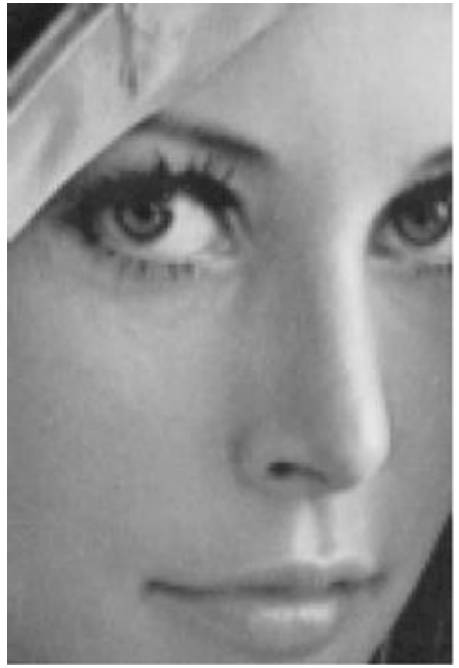
Straightforward solution

NN architecture: a CNN having the same input and output size, like those for semantic segmentation.

Training data: $TR = \{(y, z),\ \text{being}\ y\ \text{a natural image}\}$

Easy to gather large TR, given the forward model to generate noisy images.

Forward pass, ruled by physical laws / sensor



y

Noise model

z

Inverse pass, ill-posed, unknown

z

CNN

y

G. Boracchi

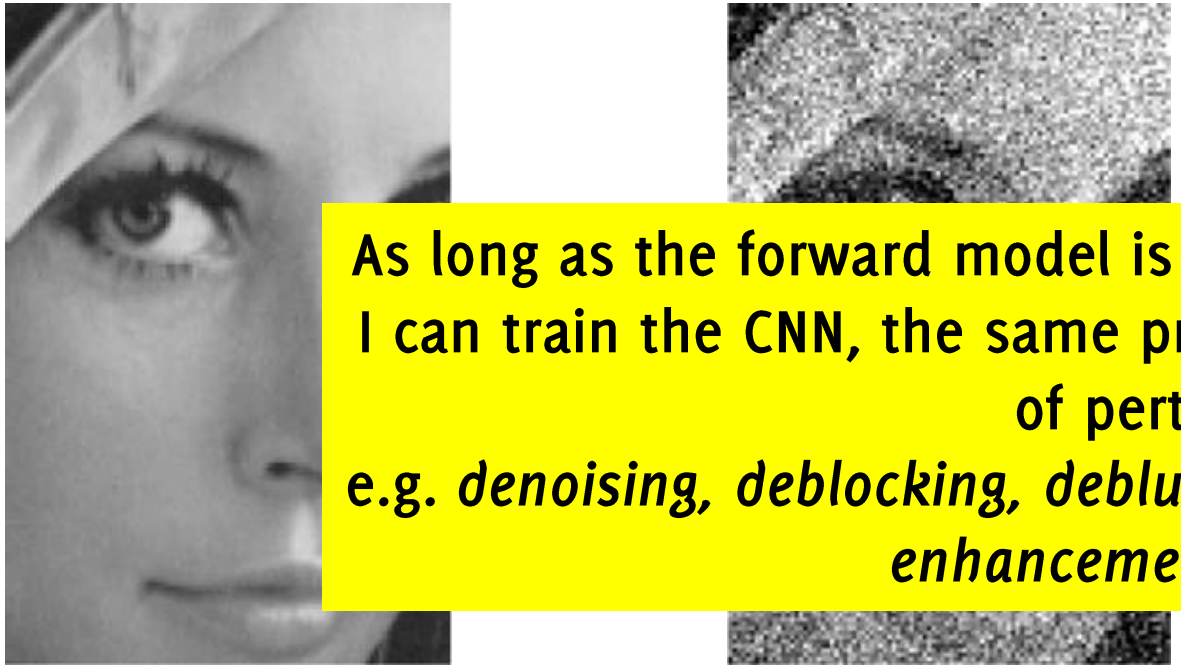
Straightforward solution

NN architecture: a CNN having the same input and output size, like those for semantic segmentation.

Training data: $TR = \{(y, z),\ \text{being } y \text{ a natural image}\}$

Easy to gather large TR, given the forward model to generate noisy images.

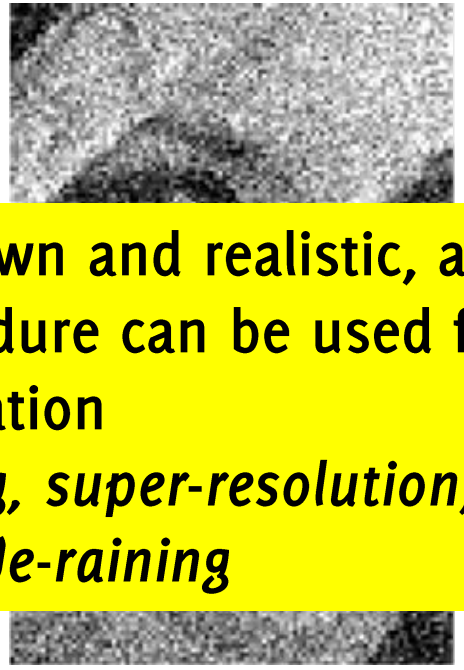
Forward pass, ruled by physical laws / sensor



y

z

Inverse pass, ill-posed, unknown



z

y

As long as the forward model is known and realistic, and as far as I can train the CNN, the same procedure can be used for any type of perturbation

e.g. *denoising, deblocking, deblurring, super-resolution, inpainting, enhancement, de-raining*

G. Boracchi

Image denoising by CNNs

Is denoising by deep neural network a Superivsed or Unsupervised Learning problem?

- The loss function however is typical of supervised learning, e.g., the MSE: $\|\hat{y} - y\|_2$ we assume to have the noisy-free image.
- It resambles unsupervised learning since no label / annotation is required for training.
- It resambles self-supervised learning as we easily generate annotated pairs. However the auxiliary task (denoising) is our ultimate goal.

Supervised Training a neural network to reduce a loss function has made a substantial difference w.r.t. traditional denoising algorithm, which were primarily designed over statistical modeling of the input and the noise.

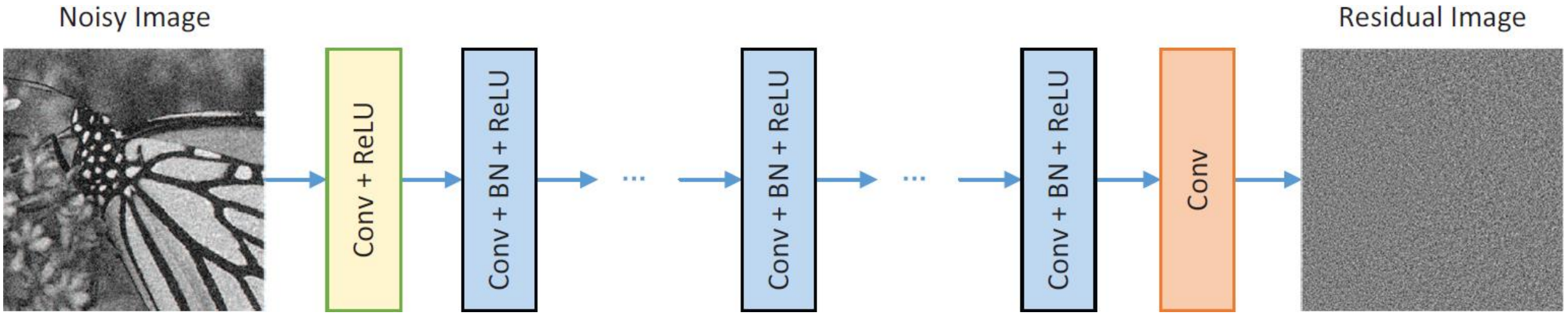
SIMPLE DEEP NNS FOR IMAGE DENOISING

G. Boracchi

Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising

Kai Zhang, Wangmeng Zuo, *Senior Member, IEEE*, Yunjin Chen, Deyu Meng, and Lei Zhang, *Senior Member, IEEE*

Dn-CNN: The architecture (FCNN)



K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, *"Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising,"* IEEE TIP 2017

Residual Learning

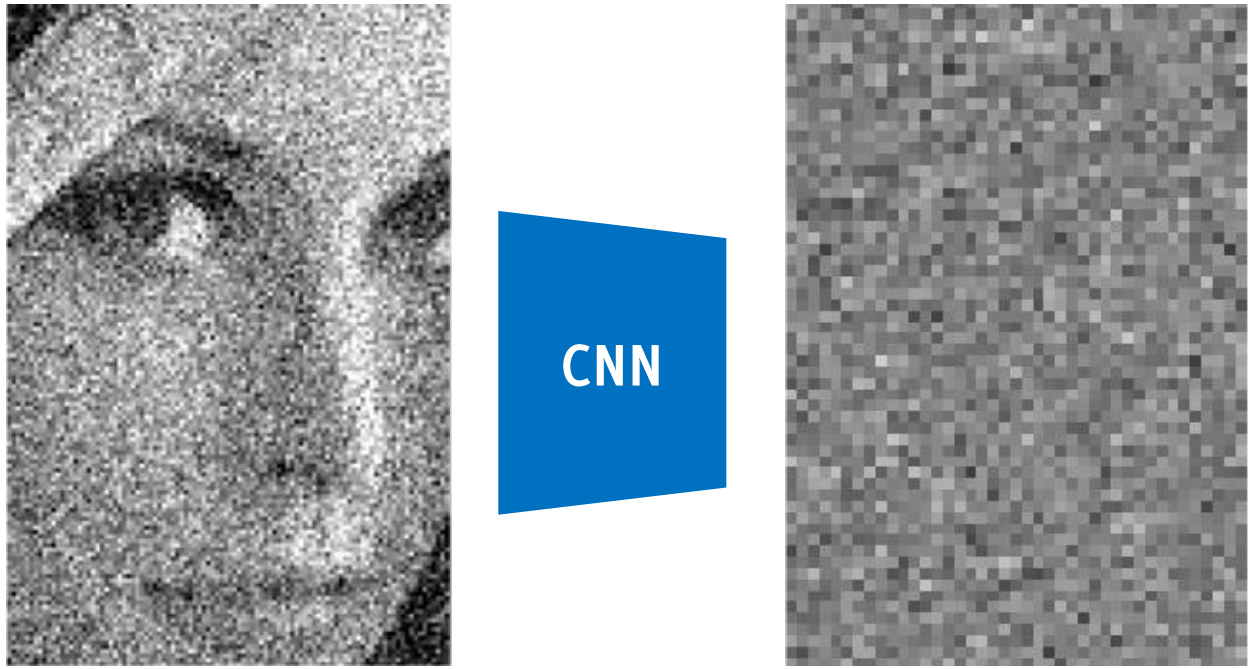
The network is trained to predict the noise, rather than the noise-free image

$$TR = \{(z, \eta), \qquad z = y + \eta\}$$

Thus the output will be

$$\hat{y} = z - \hat{\eta} = z - CNN(z)$$

«It easier to optimize the loss function over the noise $\|\eta - CNN(z)\|_2$ *rather than on the image»*



z η

DnCNN architecture

Network architecture:

- Conv+ReLU : 64 filters of size 3 x 3 x c (layer 1)..
- Conv+BN+ReLU: 64 filters of size 3 x 3 x 64 (layer from 2 to $d-1$) + Batch Normalization
- Conv: c filters of size 3 x 3 x 64 are used to reconstruct the input

Border artifacts:

- Zero padding outside the image before the processing

K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142–3155, July 2017.

DnCNN details

Loss to minimize

$$\ell(\Theta) = \frac{1}{2N}\sum_{i=1}^{N}\|CNN(z_i,\Theta) - (z_i - y_i)\|_F^2$$

- Assessing on how good the network can **estimate the noise realizations** in images
- **Training samples** are **easy to generate**: add synthetic corruption
- To ease the training process, the **loss is assessed on patches** cropped from noisy images.
- Since the network is fully convolutional, it can be applied to images of arbitrary size

K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142–3155, July 2017.

DnCNN details

Training:

400 images of 180 x 180 pixels (larger training sets do not improve performance substantially)

Network **training is performed patch-wise**

- 40x40 patches for Gaussian denoising
- 50x50 patches for other denoising problems
- The network is made of convolutions only: can be tested on arbitrarily-sized images.

Repeat training when corrupting training images by different noise levels σ

K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3142–3155, July 2017.

DnCNN details

Denoising task:

- Denoising additive white Gaussian noise
  - known noise variance
  - unknown noise variance
- Denoising other artifacts:
  - bicubic upsampling (Super-resolution)
  - Jpeg compression

K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising,” IEEE TIP 2017

The Receptive Field
A very important aspect in CNNs

The Receptive Field

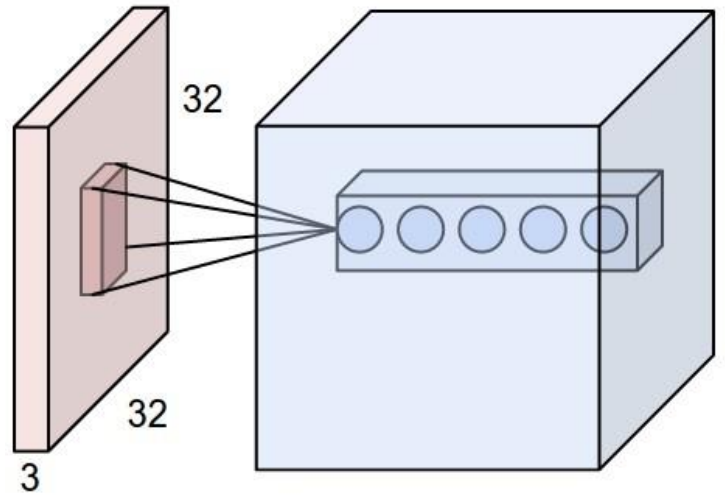
One of the basic concepts in deep CNNs.

Due to sparse connectivity, unlike in FC networks where the value of each output depends on the entire input, **in CNN an output only depends on a region of the input.**

This region in the input is the receptive field for that output

The deeper you go, the wider the receptive field is: maxpooling, convolutions and stride > 1 increase the receptive field

Usually, the receptive field refers to the final **output unit** of the network in relation to the network input, but the same definition holds for intermediate volumes



By Aphex34 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45659236

Network depth:

This is sized to make the receptive field comparable to receptive fields in image denoising algorithms.

Having only 3x3 convolutions, the receptive field of a d-layered network is $(2d + 1) \times (2d + 1)$

- $d = 17$ for Gaussian denoising (known variance)
- $d = 20$ for Gaussian denoising (unknown variance) and artefact removal (superresolution / jpeg)

TABLE I
THE EFFECTIVE PATCH SIZES OF DIFFERENT METHODS WITH NOISE LEVEL $\sigma = 25$.

Methods	BM3D [2]	WNNM [13]	EPLL [33]	MLP [24]	CSF [14]	TNRD [16]
Effective Patch Size	49×49	361×361	36×36	47×47	61×61	61×61

K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," IEEE TIP 2017

Dn-CNN Limitations

DnCNN learns a mapping function

$$z \rightarrow \mathcal{F}(\boldsymbol{\Theta}_{\sigma}, z)$$

where the network parameters $\boldsymbol{\Theta}_{\sigma}$ depend on σ, the noise standard deviation

DnCNN is hard to be deployed to images corrupted by:

- different noise levels.
- different noise models
- spatially variant / signal dependent noise.

Zhang, Kai, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising." IEEE Transactions on Image Processing 27.9 (2018): 4608-4622.

Handling noise model

FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising

Kai Zhang, Wangmeng Zuo, *Senior Member, IEEE*, and Lei Zhang, *Fellow, IEEE*

FFDNet

FFD-Net learns a mapping function

$$z \rightarrow \mathcal{F}(\boldsymbol{\Theta}, z, M)$$

Network inputs for an image $W \times H \times C$

- a noise level-map M of the noise standard deviation over z
- 4 downsampled and cropped images having size $\frac{\mathrm{W}}{2} \times \frac{H}{2} \times C$, stacked in different channels

Thus, the input has size $\frac{\mathrm{W}}{2} \times \frac{H}{2} \times (4C + 1)$

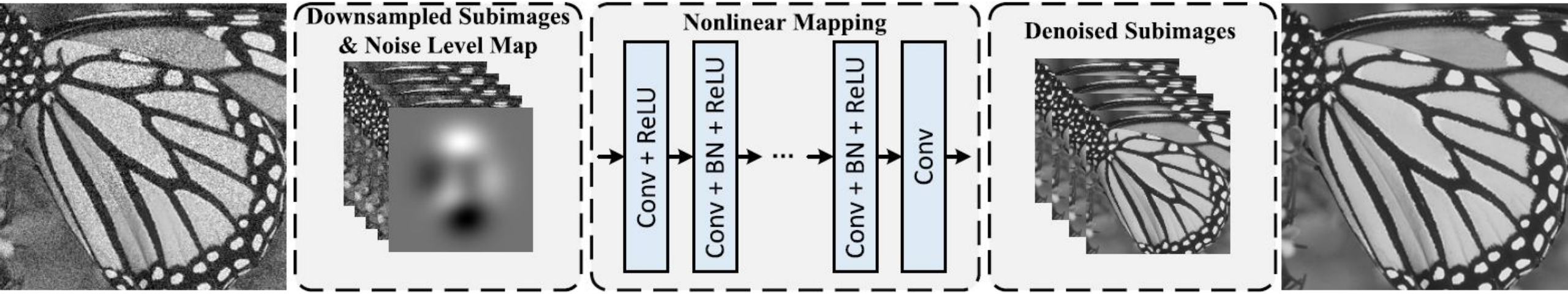
The network parameters $\boldsymbol{\Theta}$ does not depend on noise parameters

Zhang, Kai, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising." *IEEE Transactions on Image Processing* 27.9 (2018): 4608-4622.

FFDNet Architecture

Input preparation,
image downsampling

upsampling



$W \times H \times C$

$\frac{W}{2} \times \frac{H}{2} \times (4C + 1)$

$\frac{W}{2} \times \frac{H}{2} \times (4C)$

$W \times H \times C$

Zhang, Kai, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising." *IEEE Transactions on Image Processing* 27.9 (2018): 4608-4622.

FFDNet Architecture

All the convolutions are 3x3, as DnCNN

Grayscale images:

- depth level: $d = 15$,
- Number of filters per layer: 64

Color images:

- Depth level $d = 12$,
- Number of filters per layer: 96

Design driven by performance and «heuristic criteria»: use lower values of d in color images to better exploit correlation among R,G,B channels

Zhang, Kai, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising." *IEEE Transactions on Image Processing* 27.9 (2018): 4608-4622.

FFDNet: the input

Resized images (pixel shuffle):

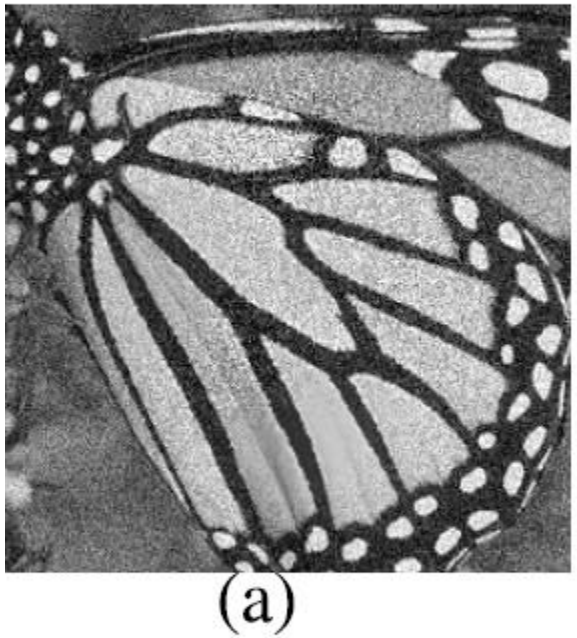
- Reduces the spatial extent of the input, thus the computational complexity in all the convolutional layers after the first one.
- Resizing also increases receptive field.
- “Downsampling” is indeed shuffling of the image among channels

Noise level-map $\mathbf{M}$:

- An image reporting in each pixel the noise standard deviation
- It enables using a single model for different values of σ without need to retrain
- In case of Additive White Gaussian Noise (AWGN), M corresponds to a uniform image equal to the noise std σ
- In case of spatial-variant noise, M is a non-uniform image defined by the noisy image z and the sensor characteristics

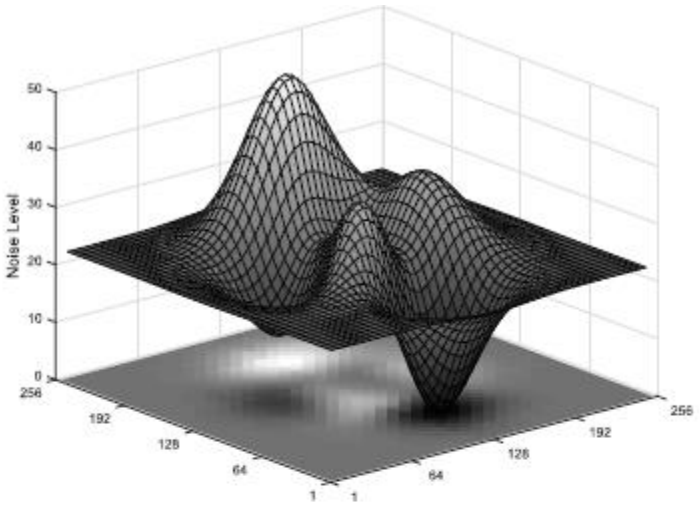
Zhang, Kai, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising." IEEE Transactions on Image Processing 27.9 (2018): 4608-4622.

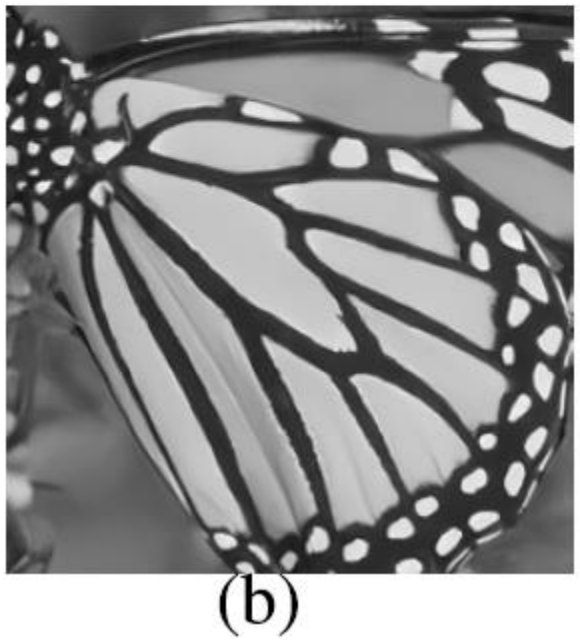
FFDNet: noise level maps



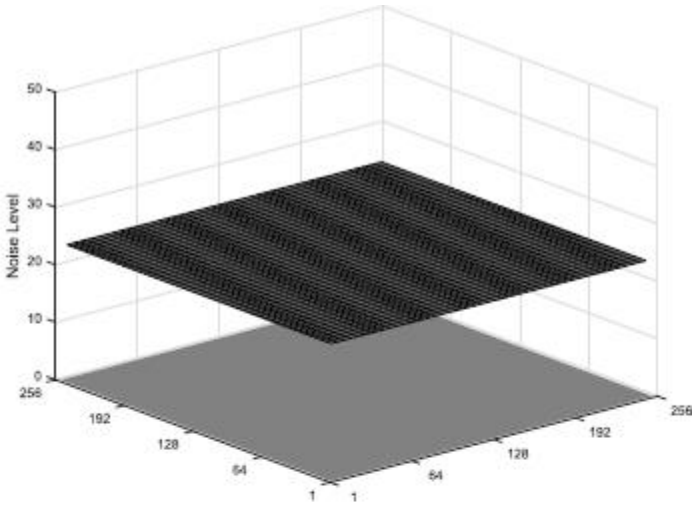
Noisy image (20.55dB) with spatially variant AWGN

Denoised image using ground truth noise mask (30.08dB)





Denoised image uniform noise mask (27.45dB)



Zhang, Kai, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising." TIP 2018

FFDNet training

Loss to minimize

$$\ell(\Theta) = \frac{1}{2N}\sum_{i=1}^{N}\|\mathcal{F}(z_i, \Theta, M) - y_i\|_F^2$$

Assessing on how good the network can estimate noise-free patches.

Frobenious norm is the ℓ^2 norm for matrix entries

Patch-wise training

- Training samples are easy to generate: add synthetic noise,
- Loss measured on patches cropped from noisy images
- No need to train using heterogeneous noise maps, the fully convolutional network can operate on locally uniform noise maps.
- Patch size 50x50 for grayscale, 70x70 in color

Zhang, Kai, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising." TIP 2018

DnCNN and FFDNet Performance

TABLE III

THE AVERAGE PSNR(dB) RESULTS OF DIFFERENT METHODS ON BSD68 WITH NOISE LEVELS 15, 25 35, 50 AND 75

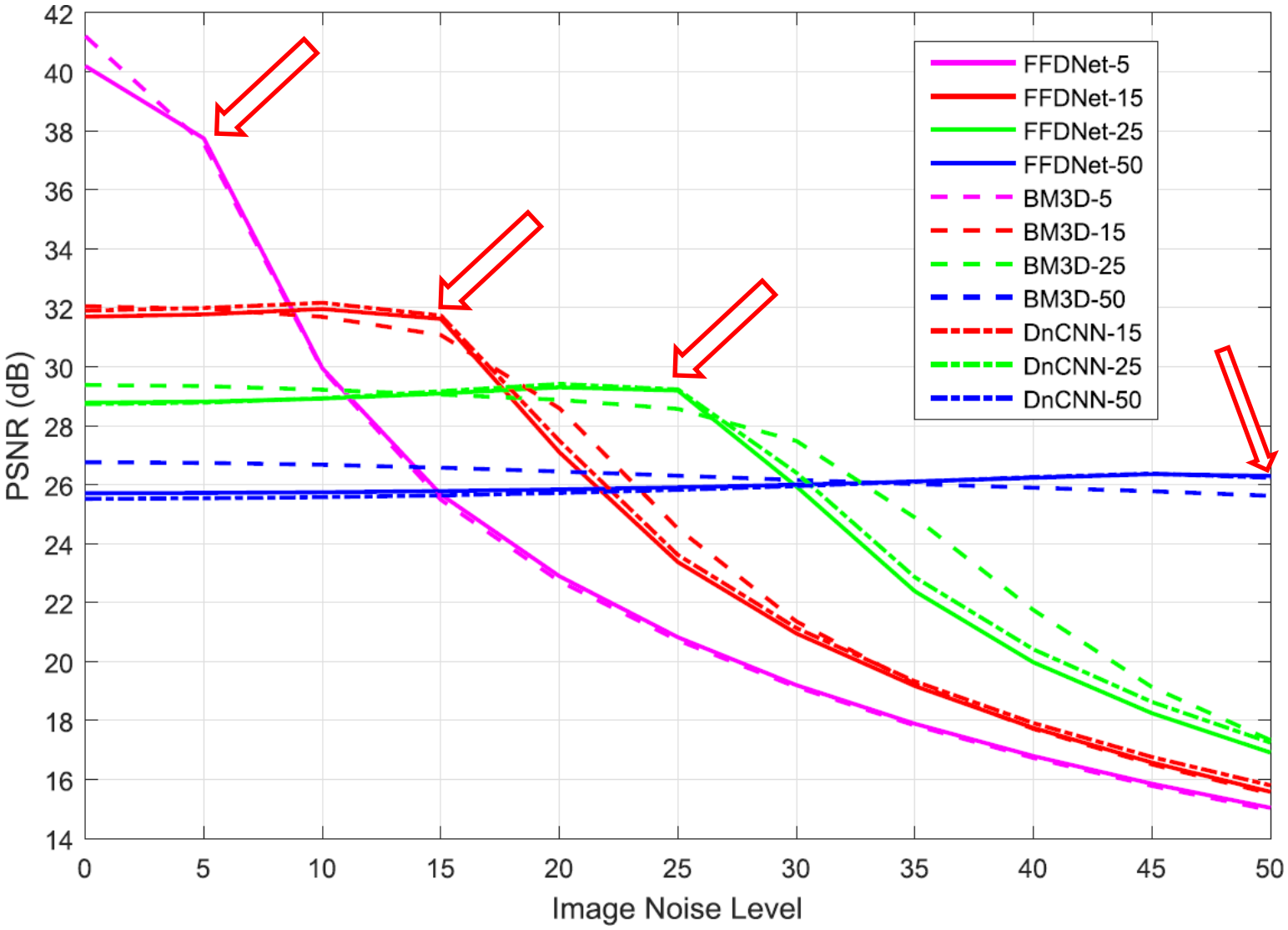
Methods	BM3D	WNNM	MLP	TNRD	DnCNN	FFDNet
$\sigma = 15$	31.07	31.37	–	31.42	31.72	31.63
$\sigma = 25$	28.57	28.83	28.96	28.92	29.23	29.19
$\sigma = 35$	27.08	27.30	27.50	–	27.69	27.73
$\sigma = 50$	25.62	25.87	26.03	25.97	26.23	26.29
$\sigma = 75$	24.21	24.40	24.59	–	24.64	24.79

Zhang, Kai, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising." TIP 2018

DnCNN and FFDNet Performance (wrong M)

For all the denoisers:

- Using higher σ than the true one, results in oversmoothing, thus the PSNR is constant but does not reach the same level as when the noise level is correct.
- Using lower σ than the true one, results in noisy outputs, thus decreases the PSNR.



Zhang, Kai, Wangmeng Zuo, and Lei Zhang. "FFDNet: Toward a fast and flexible solution for CNN-based image denoising." TIP 2018

FDD-Net C-BM3D original

ROBUST AND INTERPRETABLE BLIND IMAGE DENOISING VIA BIAS-FREE CONVOLUTIONAL NEURAL NETWORKS

Sreyas Mohan*
Center for Data Science
New York University
sm7582@nyu.edu

Zahra Kadkhodaie*
Center for Data Science
New York University
zk388@nyu.edu

Eero P. Simoncelli
Center for Neural Science, and
Howard Hughes Medical Institute
New York University
eero.simoncelli@nyu.edu

Carlos Fernandez-Granda
Center for Data Science, and
Courant Inst. of Mathematical Sciences
New York University
cfgranda@cims.nyu.edu

Generalization to different noise levels

An important advantage of deep-learning techniques over traditional methodology is that **a single neural network can be trained to perform denoising at a wide range of noise levels.**

Empirical evidence that current ***state-of-the-art architectures systematically overfit to the noise levels in the training set****, performing very poorly at new noise levels.*

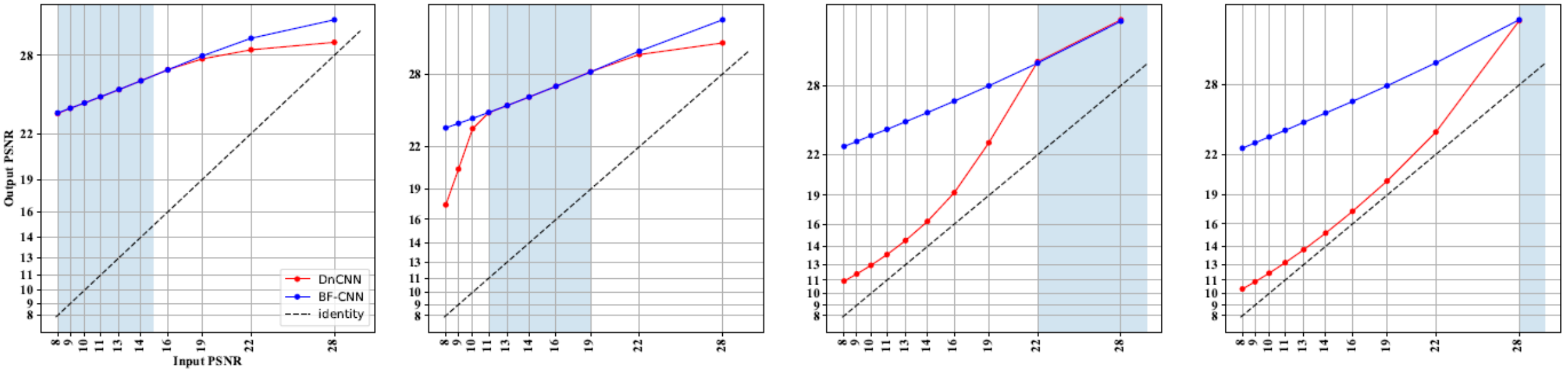
Generalization can be achieved through a simple architectural modification: removing all additive constants.

*"**bias-free**" networks attain state-of-the-art performance over a broad range of noise levels, even when trained over a narrow range.*

Mohan et al. “Robust And Interpretable Blind Image Denoising Via Bias-Free Convolutional Neural Networks” 2019 ICLR

Bs-cnn: Removing biases

The shaded region in the plots denote the training range for the two networks (standard DnCNN in red and the equivalent Bias-Free CNN in blue)

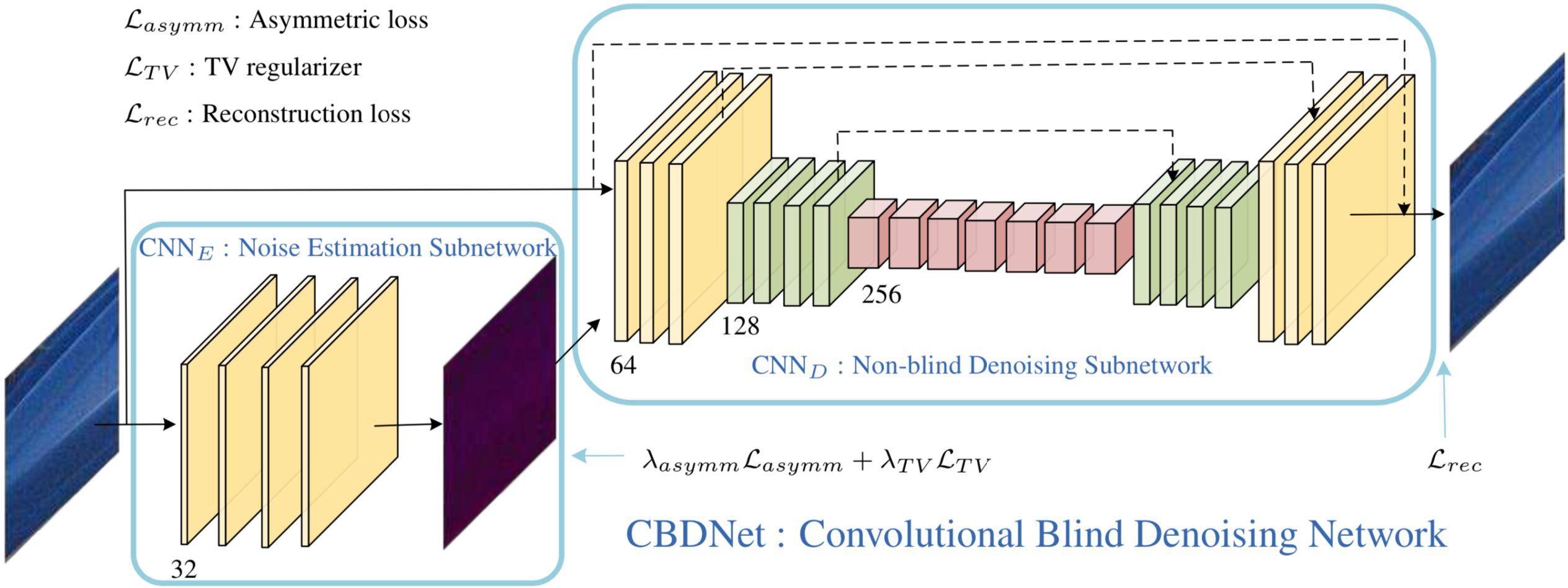


Mohan et al. “Robust And Interpretable Blind Image Denoising Via Bias-Free Convolutional Neural Networks” 2019 ICLR

Toward Convolutional Blind Denoising of Real Photographs

Shi Guo[1,3,4], Zifei Yan(✉) [1], Kai Zhang[1,3], Wangmeng Zuo[1,2], Lei Zhang[3,4]
[1]Harbin Institute of Technology, Harbin; [2]Peng Cheng Laboratory, Shenzhen;
[3] The Hong Kong Polytechnic University, Hong Kong; [4]DAMO Academy, Alibaba Group
guoshi28@outlook.com, {wmzuo,yanzifei}@hit.edu.cn
cskaizhang@gmail.com, cslzhang@comp.polyu.edu.hk

CBDNet: Network



Guo et al. "Toward Convolutional Blind Denoising of Real Photographs" CVPR 2019

CBDNet

- Existing CNN denoisers tend to be over-fitted to Gaussian noise and **generalize poorly to real-world noisy images with more sophisticated noise.**
- Realistic noise model is the foremost issue for blind denoising of real photographs.
- **Take into account both Poisson-Gaussian model and in-camera processing pipeline** (e.g. de-mosaicking, Gamma correction, and JPEG compression).
- CBDNet is comprised of two subnetworks:
  - noise estimation and
  - non-blind denoising.
- Adopt an asymmetric loss by imposing more penalty on under-estimation error of noise level.
- The network has a bottleneck

Guo et al. “Toward Convolutional Blind Denoising of Real Photographs” CVPR 2019

Class-Aware Denoising

Class-Aware Fully Convolutional Gaussian and Poisson Denoising

Tal Remez, *Member, IEEE*, Or Litany, *Member, IEEE*, Raja Giryes, *Member, IEEE*, and Alex M. Bronstein, *Fellow, IEEE*

Key Ingredients

Fully convolutional to enable denoising of images having different size.

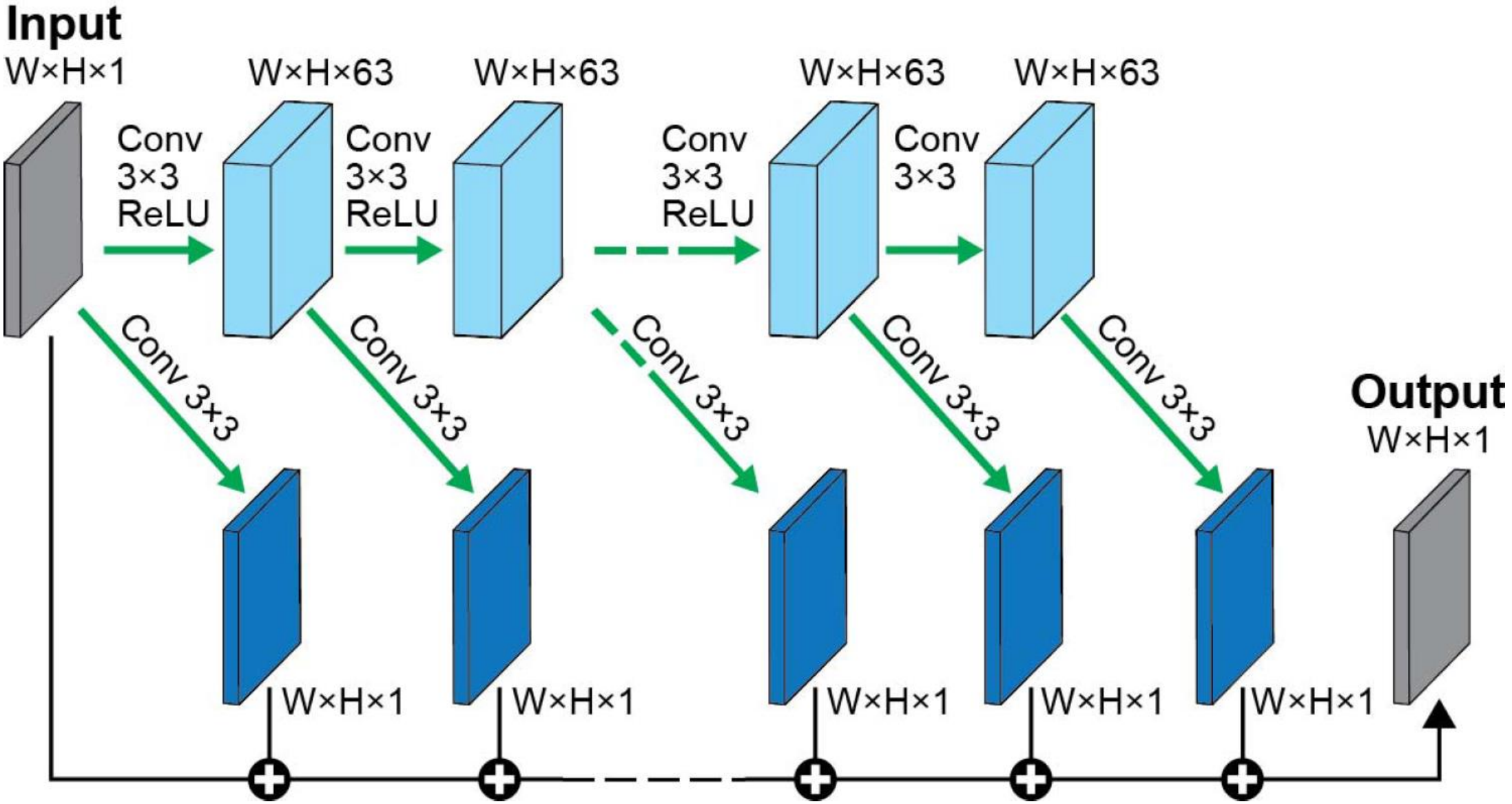
- Fast execution and relatively low number of parameters

Gradual denoising as the volume circulates through the network

Improve prior on images: narrow down the space of images and **train a network for a specific class of images**.

Remez, T., Litany, O., Giryes, R., & Bronstein, A. M. Class-Aware Fully Convolutional Gaussian and Poisson Denoising. IEEE TIP 2018

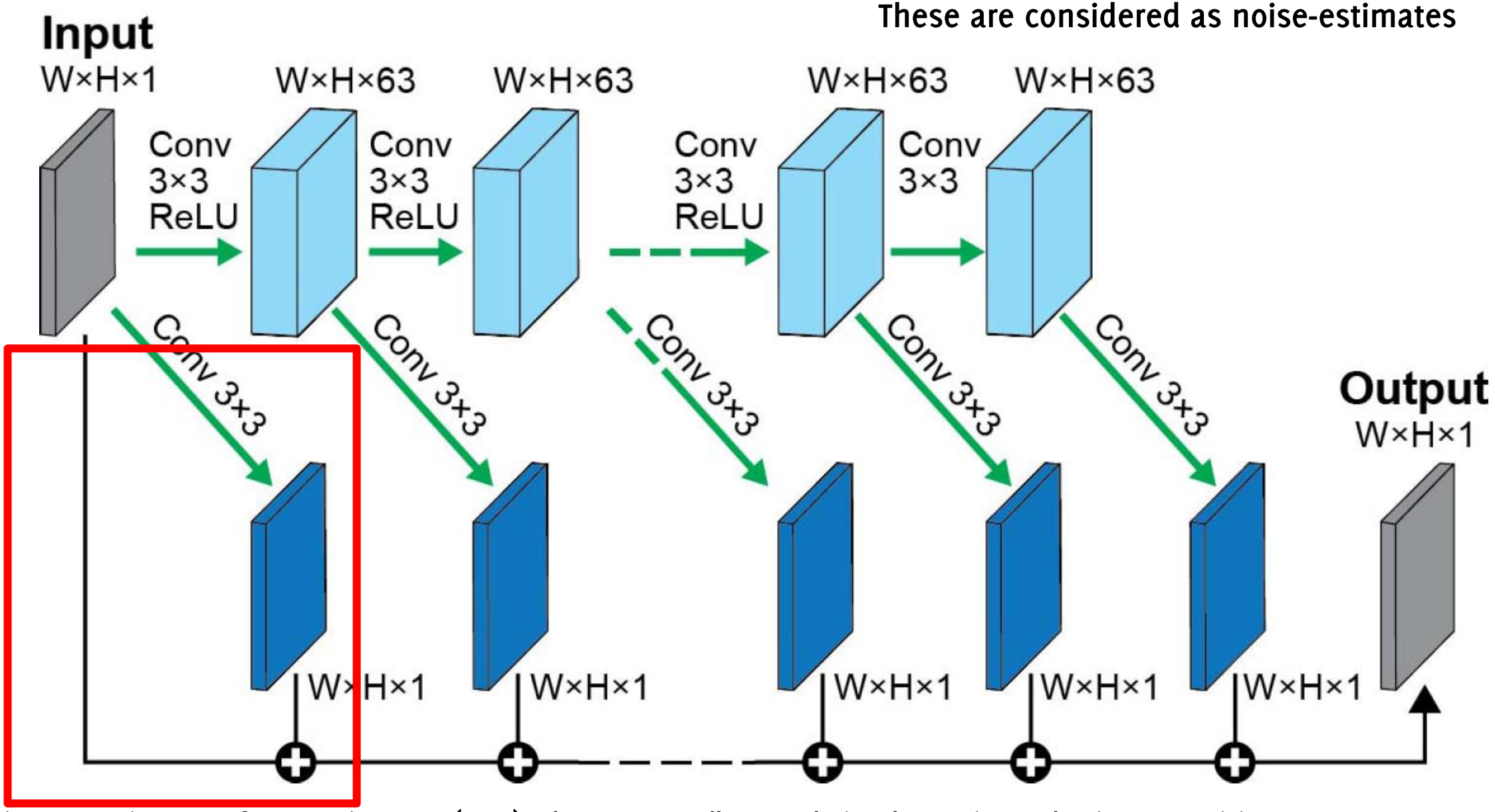
The Network Architecture



Remez, T., Litany, O., Giryes, R., & Bronstein, A. M. (2018). Class-Aware Fully Convolutional Gaussian and Poisson Denoising. IEEE TIP 2018,

The Network Architecture

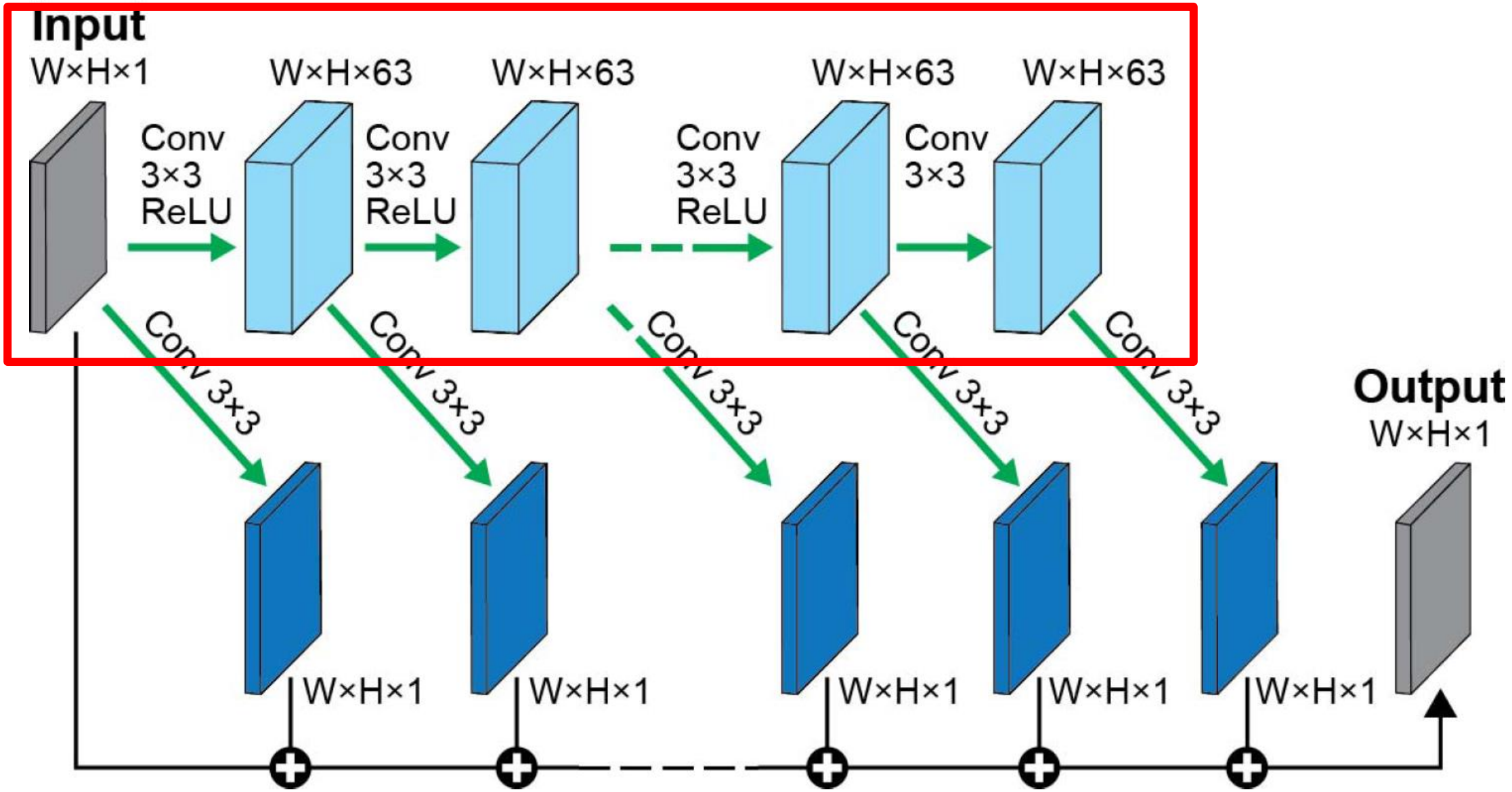
The noisy input is added to the networks «gradual» outputs.
These are considered as noise-estimates



Remez, T., Litany, O., Giryes, R., & Bronstein, A. M. (2018). Class-Aware Fully Convolutional Gaussian and Poisson Denoising. IEEE TIP 2018,

The Network Architecture

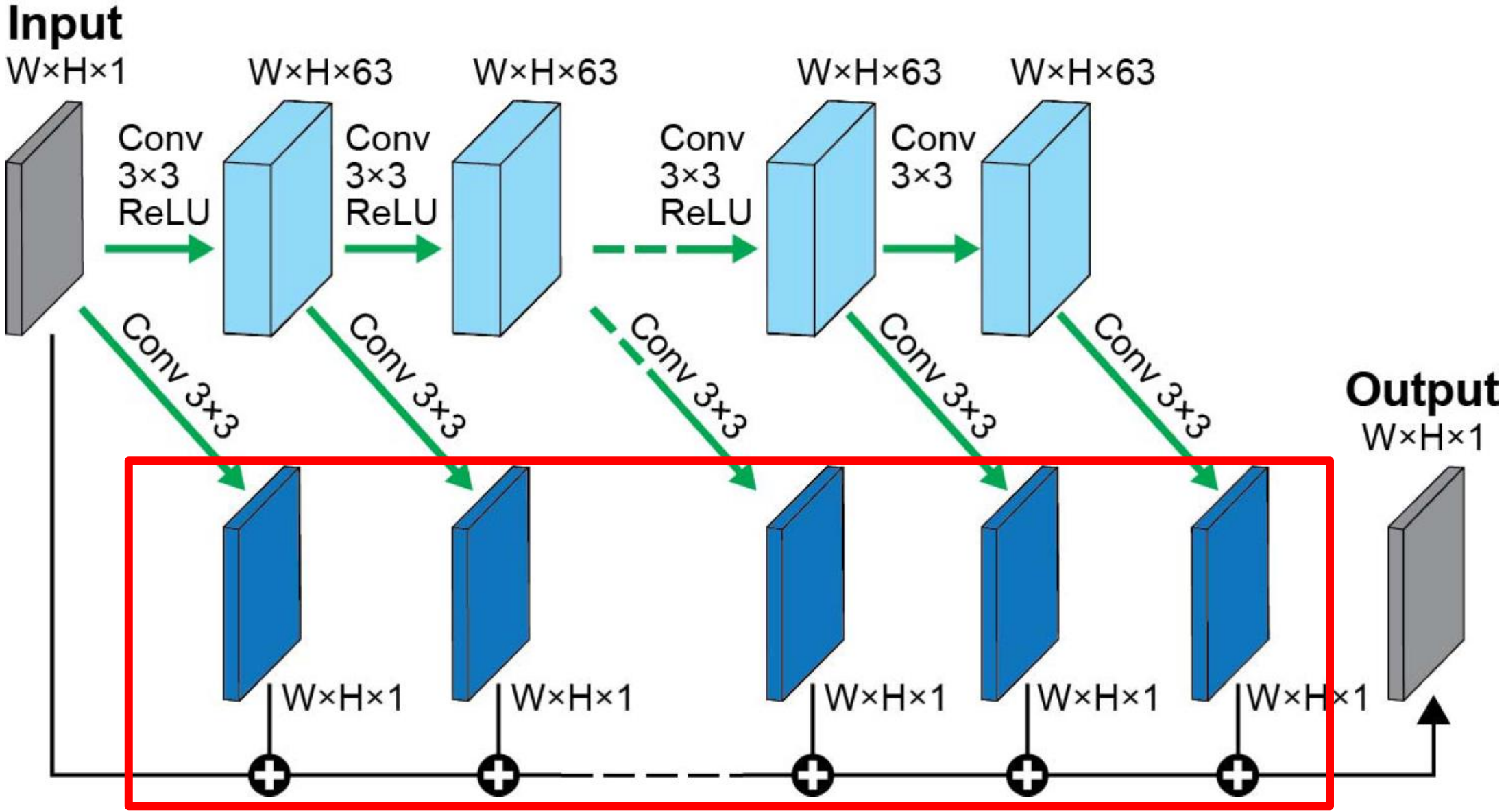
Gradual denoising: 20 of such layers are being stacked in this network



Remez, T., Litany, O., Giryes, R., & Bronstein, A. M. (2018). Class-Aware Fully Convolutional Gaussian and Poisson Denoising. IEEE TIP 2018,

The Network Architecture

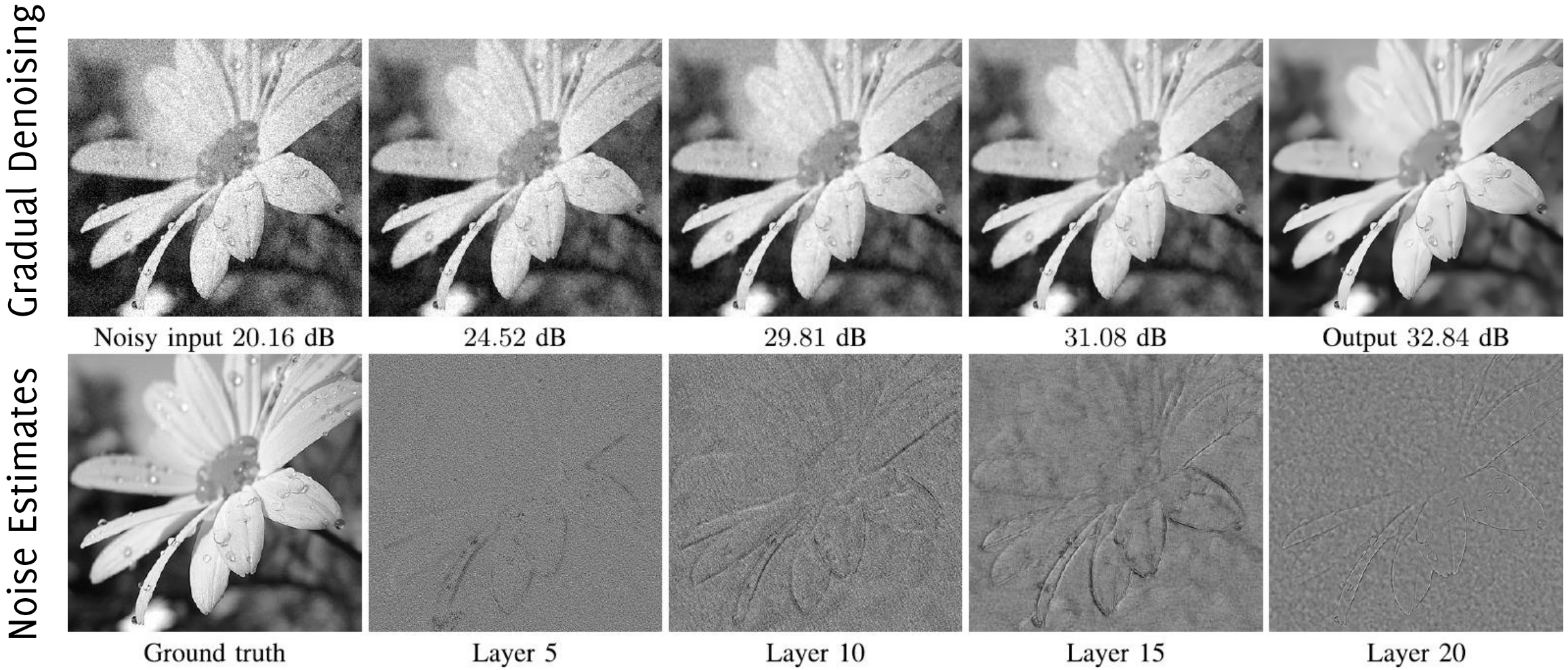
These are referred to as noise estimates, because they sum to the noisy input an the result cancels out the noise



Remez, T., Litany, O., Giryes, R., & Bronstein, A. M. (2018). Class-Aware Fully Convolutional Gaussian and Poisson Denoising. IEEE TIP 2018,

Noise estimates

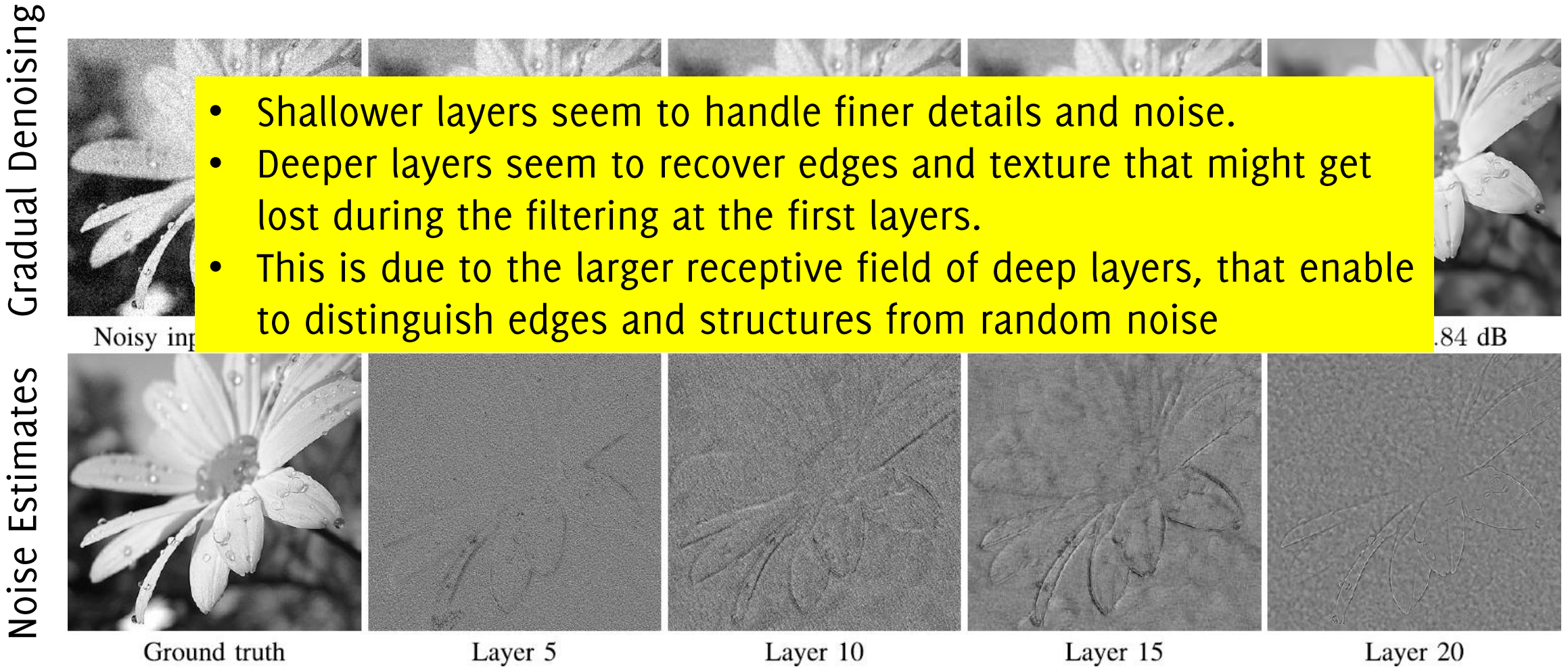
It is possible to analyze the process by visualizing noise estimates



Remez, T., Litany, O., Giryes, R., & Bronstein, A. M. (2018). Class-Aware Fully Convolutional Gaussian and Poisson Denoising. IEEE TIP 2018,

Noise estimates

It is possible to analyze the process by visualizing noise estimates



- Shallower layers seem to handle finer details and noise.
- Deeper layers seem to recover edges and texture that might get lost during the filtering at the first layers.
- This is due to the larger receptive field of deep layers, that enable to distinguish edges and structures from random noise

Remez, T., Litany, O., Giryes, R., & Bronstein, A. M. (2018). Class-Aware Fully Convolutional Gaussian and Poisson Denoising. IEEE TIP 2018,

Class-Aware Denoising

Training

- Train different networks for a specific class of images. Use the same FCNN architecture. These are the classes: *face, pet, flower, living room, street.*

Testing

- Estimate the class from a classification CNN, thus the corresponding network

Remez, T., Litany, O., Giryes, R., & Bronstein, A. M. (2018). Class-Aware Fully Convolutional Gaussian and Poisson Denoising. IEEE Transactions on Image Processing, 27(11), 5707-5722.

Class-Aware vs Class-Agnostic denoising

Remez, T., Litany, O., Giryes, R., & Bronstein, A. M. (2018). Class-Aware Fully Convolutional Gaussian and Poisson Denoising. IEEE Transactions on Image Processing, 27(11), 5707-5722.

Nonlocality in Denoising

More About Receptive Fields

Understanding the Effective Receptive Field in Deep Convolutional Neural Networks

Wenjie Luo* **Yujia Li*** **Raquel Urtasun** **Richard Zemel**
Department of Computer Science
University of Toronto
{wenjie, yujiali, urtasun, zemel}@cs.toronto.edu

The Effective Receptive Field

Not all pixels in a receptive field contribute equally *to an output unit's response. Intuitively it is easy to see that pixels at the center of a receptive field have a much larger impact on an output.*

In the forward pass*, central pixels can propagate information to the output through many different paths, while the pixels in the outer area of the receptive field have very few paths to propagate its impact.*

In the backward pass, *gradients from an output unit are propagated across all the paths, and therefore the central pixels have a much larger magnitude for the gradient from that output.*

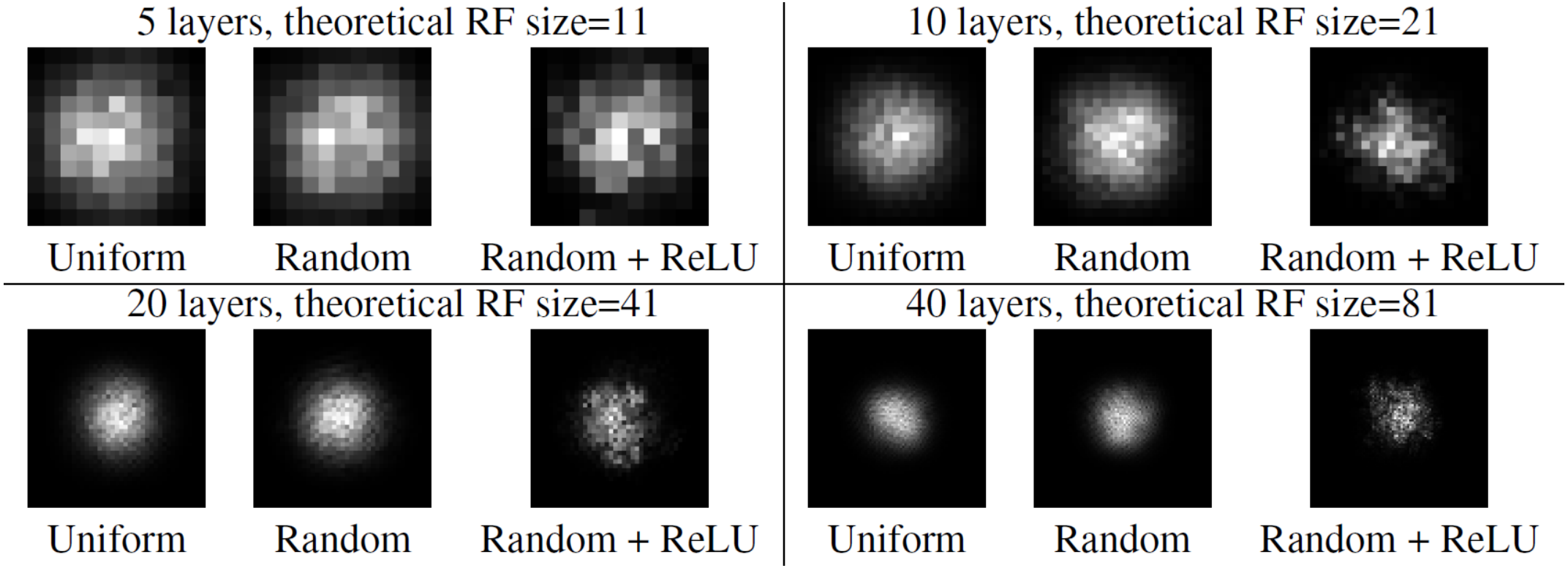
In many cases, the impact in a receptive field distributes as a Gaussian: **the effective receptive field**, only occupies a fraction of the **theoretical receptive field**.

Luo, Wenjie, et al. "Understanding the effective receptive field in deep convolutional neural networks." NIPS 2016

The Effective Receptive Field

Place a gradient signal of 1 at the center of the output plane and 0 everywhere else, and then **back-propagate this gradient** through the network to get input gradients.

Average the resulting gradients at the input over multiple initialization of the network

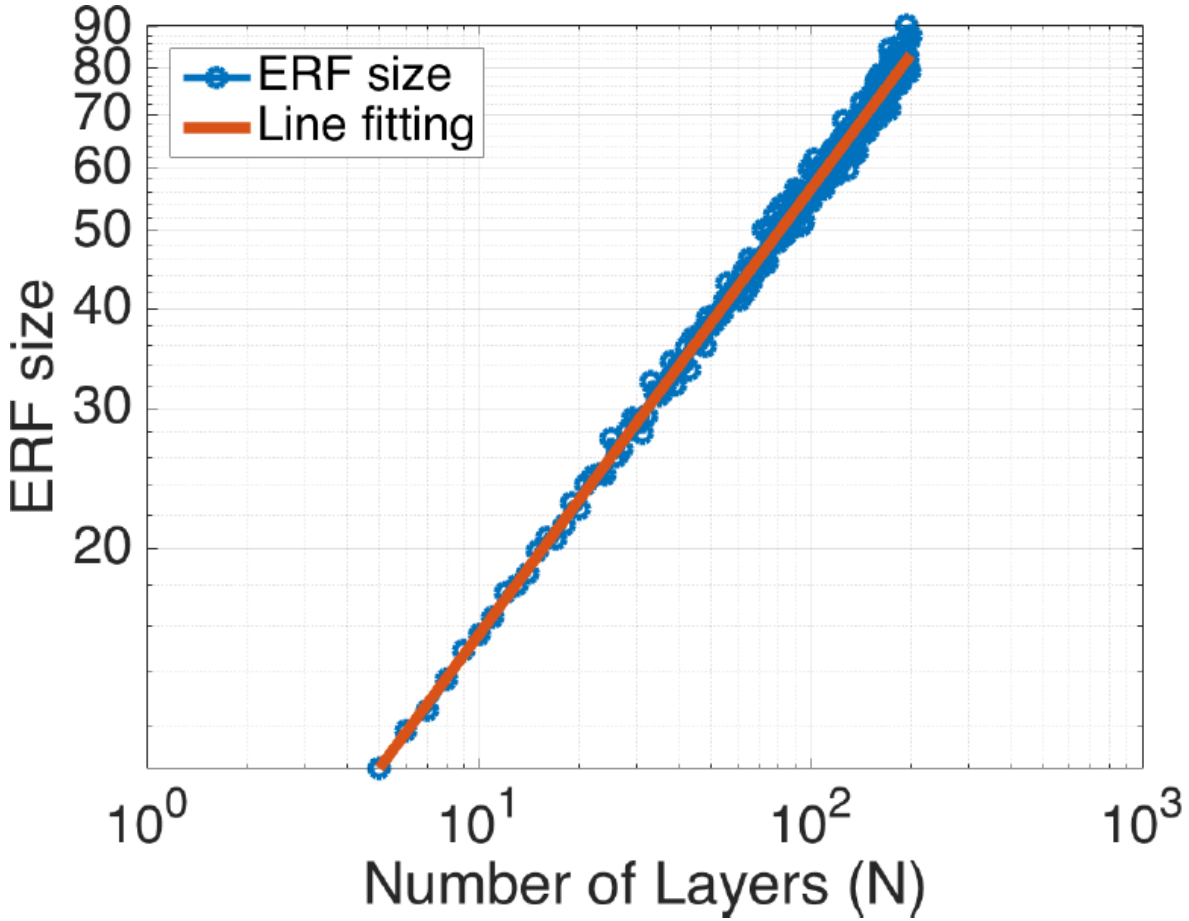


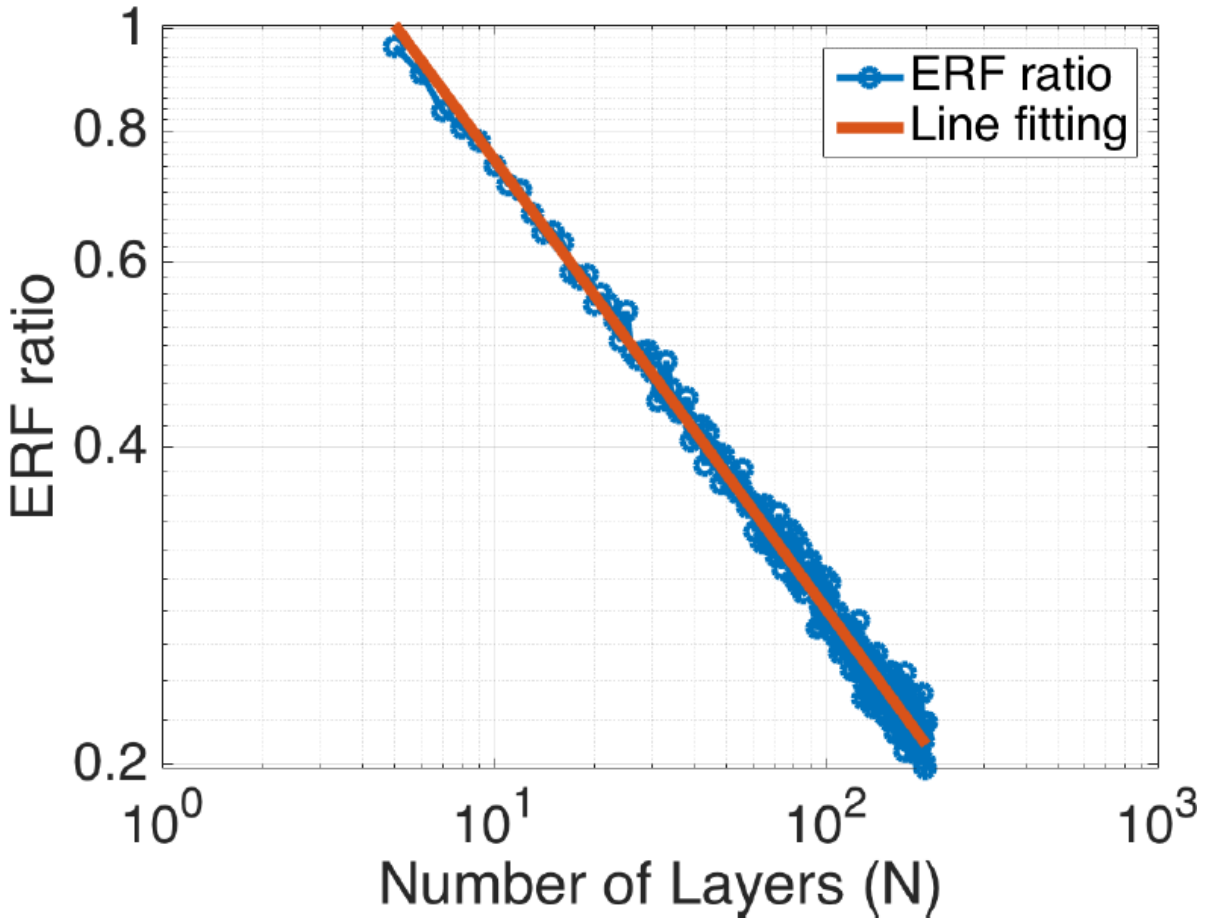
The deeper the network, the smaller the ERF w.r.t. the theoretical one!

Luo, Wenjie, et al. "Understanding the effective receptive field in deep convolutional neural networks." NIPS 2016

The Effective Receptive Field

How the network depth influences the ERF and the theoretical one



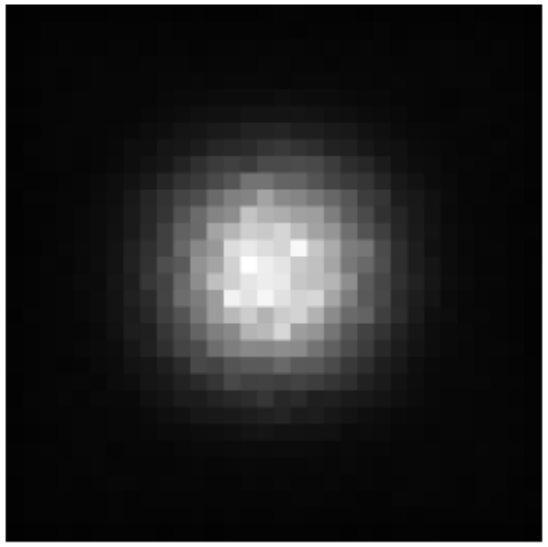


Luo, Wenjie, et al. "Understanding the effective receptive field in deep convolutional neural networks." NIPS 2016

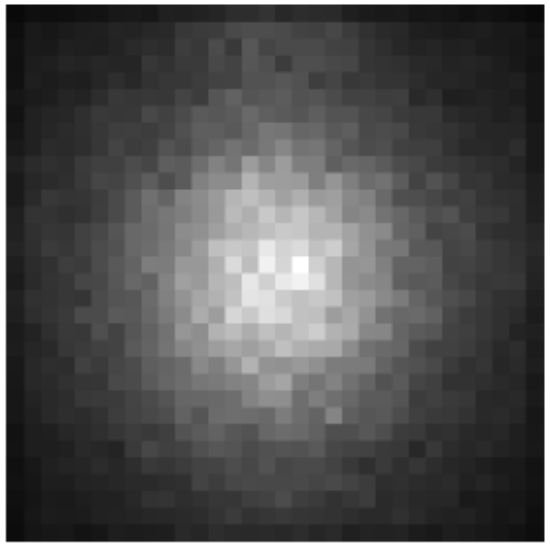
The Effective Receptive Field

Network training increases the ERF. Is random initialization bad?

CIFAR 10

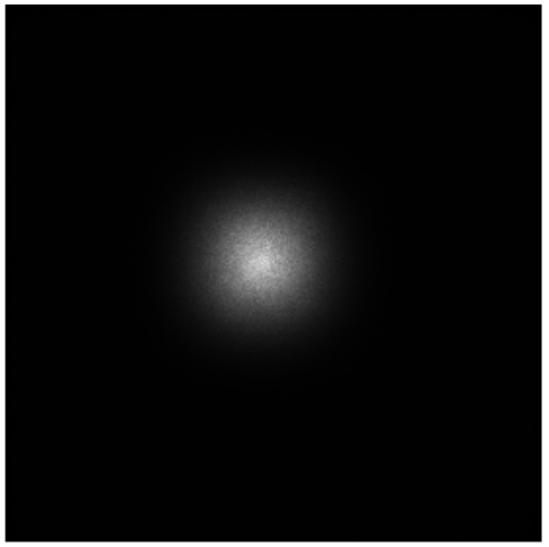


Before Training

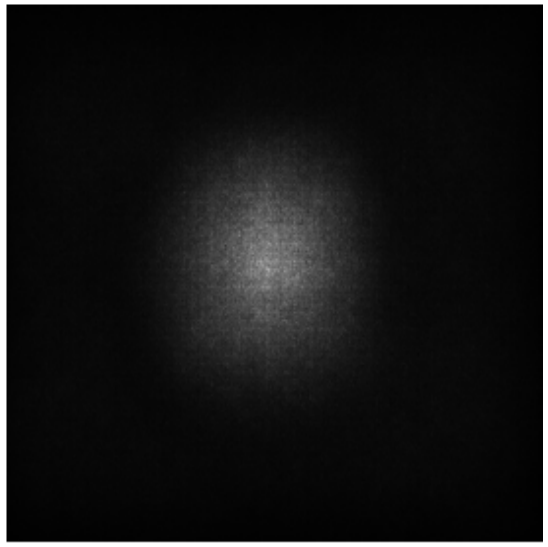


After Training

CamVid



Before Training



After Training

Luo, Wenjie, et al. "Understanding the effective receptive field in deep convolutional neural networks." NIPS 2016

The Effective Receptive Field

*A new **random weight initialization scheme** that makes the **weights at the center** of the convolution kernel to have **a smaller scale**, and the **weights on the outside to be larger** this diffuses the concentration on the center out to the periphery.*

*Practically, we can initialize the network **with any initialization method, then scale the weights according** to a distribution that has a **lower scale at the center and higher scale on the outside.***

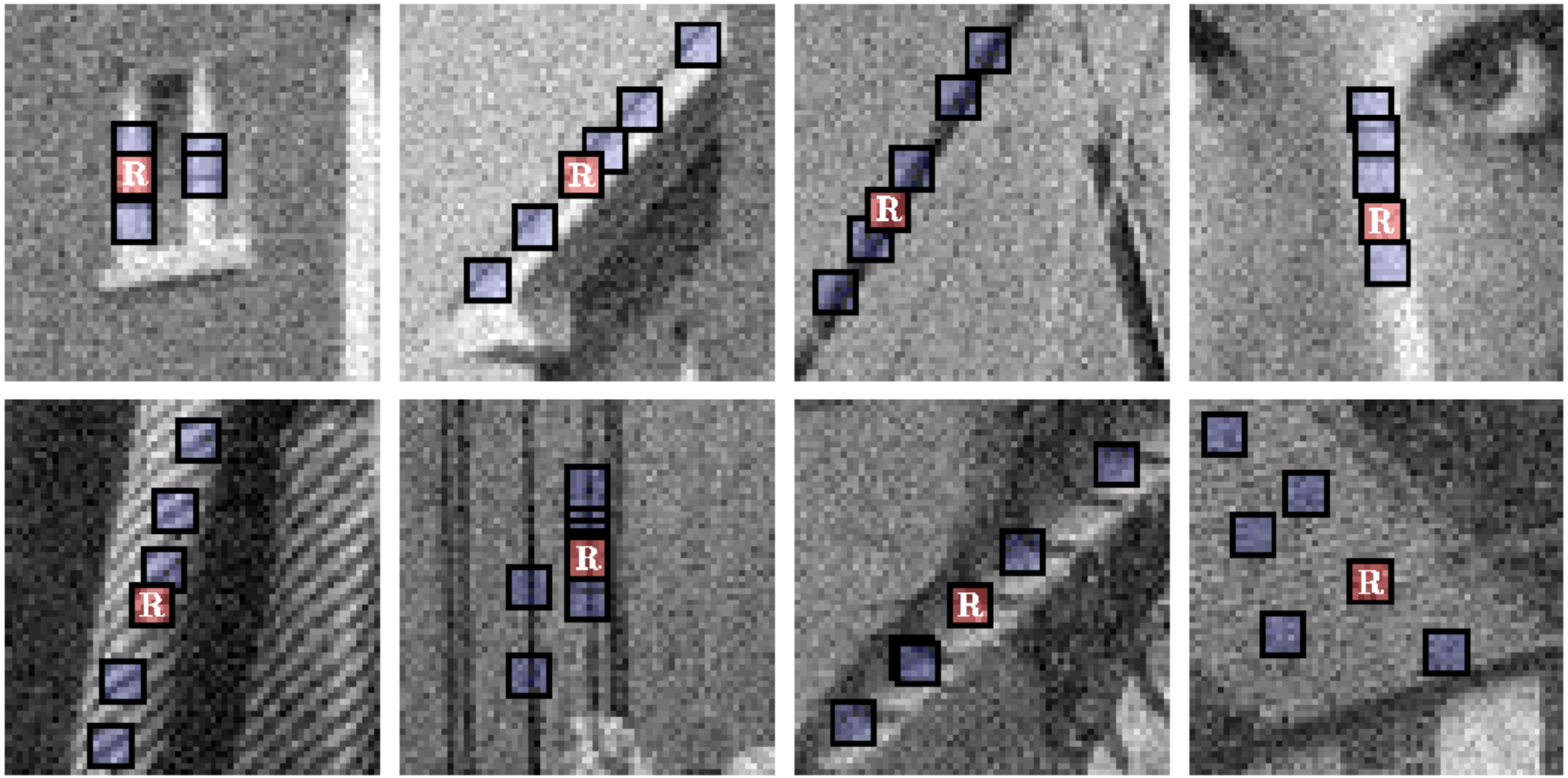
[...] In a few cases we get a 30% speed-up of training compared to the more standard initializations. But overall the benefit of this method is not always significant.

Luo, Wenjie, et al. "Understanding the effective receptive field in deep convolutional neural networks." NIPS 2016

Nonlocality-Reinforced Convolutional Neural Networks for Image Denoising

Cristóvão Cruz, Alessandro Foi, *Senior Member, IEEE*, Vladimir Katkovnik, and Karen Egiazarian, *Fellow, IEEE*

NonLocal Self Similarity



In a natural image, for any given patch there exist ***many*** *other* ***similar*** *looking* ***patches*** *at* ***different spatial locations.***

Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE TRANSACTIONS ON IMAGE PROCESSING, 16*(8).

NonLocal Filters (NLF)

In image/signal processing, the NonLocal Filters are designed to exploit local image regularities and redundancies, and to jointly process similar patches for extracting signal information.

NonLocality: relative distance inside the receptive field is not taken into account to assess pixel-wise similarity in images.

NLF rely on rather large (effective) receptive fields

TABLE I

THE EFFECTIVE PATCH SIZES OF DIFFERENT METHODS WITH NOISE LEVEL $\sigma = 25$.

Methods	BM3D [2]	WNNM [13]	EPLL [33]	MLP [24]	CSF [14]	TNRD [16]
Effective Patch Size	49×49	361×361	36×36	47×47	61×61	61×61

Cruz, C., Foi, A., Katkovnik, V., & Egiazarian, K. (2018). Nonlocality-reinforced convolutional neural networks for image denoising. *IEEE Signal Processing Letters*, *25*(8), 1216-1220.

NN3D

A **simple iterative framework** that combines the advantages of

- **CNN** as powerful models to describe the image structures
- **NLF** (NonLocal Filters) that exploit image redundancies

In a **plug-in approach** that uses any **generic CNN and NLF**

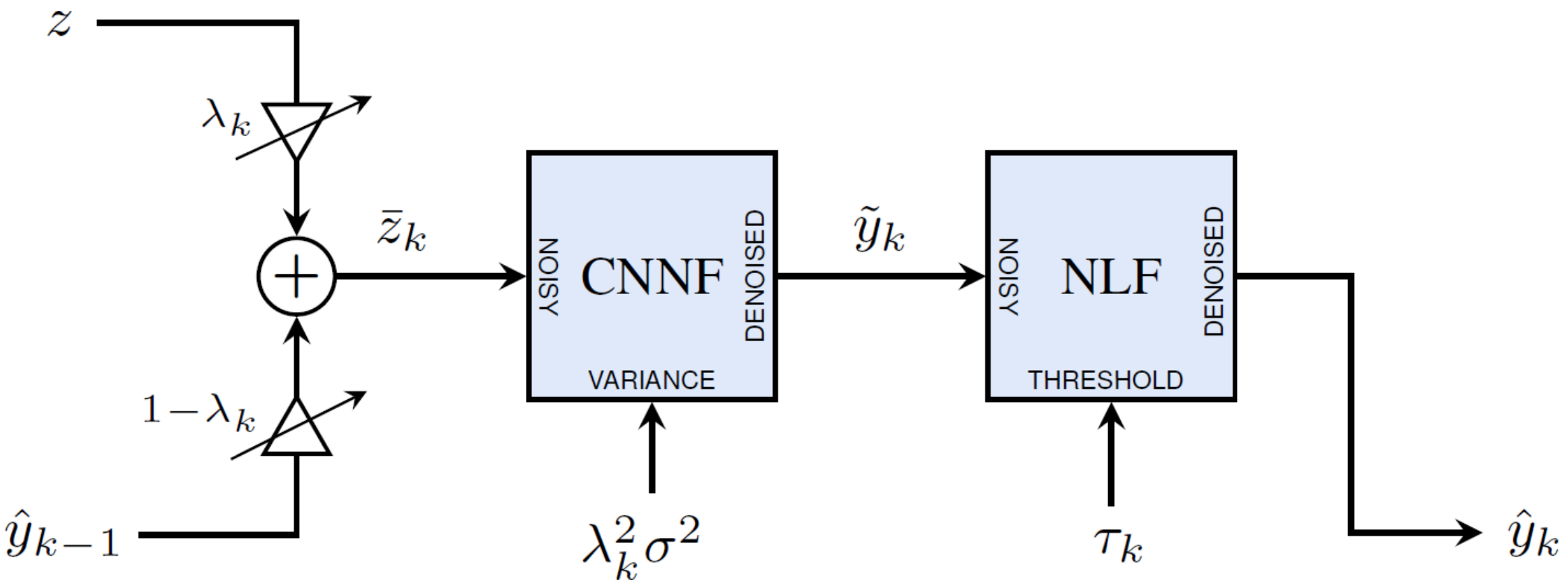
The idea:

Due to the ERF size, CNN are biased towards local features. Introducing a nonlocal filtering stage can improve their performance

Alternating CNN and NLF results in a powerful denoiser where the learned filters by the CNN are used over an extended receptive field

Cruz, C., Foi, A., Katkovnik, V., & Egiazarian, K. (2018). Nonlocality-reinforced convolutional neural networks for image denoising. *IEEE Signal Processing Letters, 25*(8), 1216-1220.

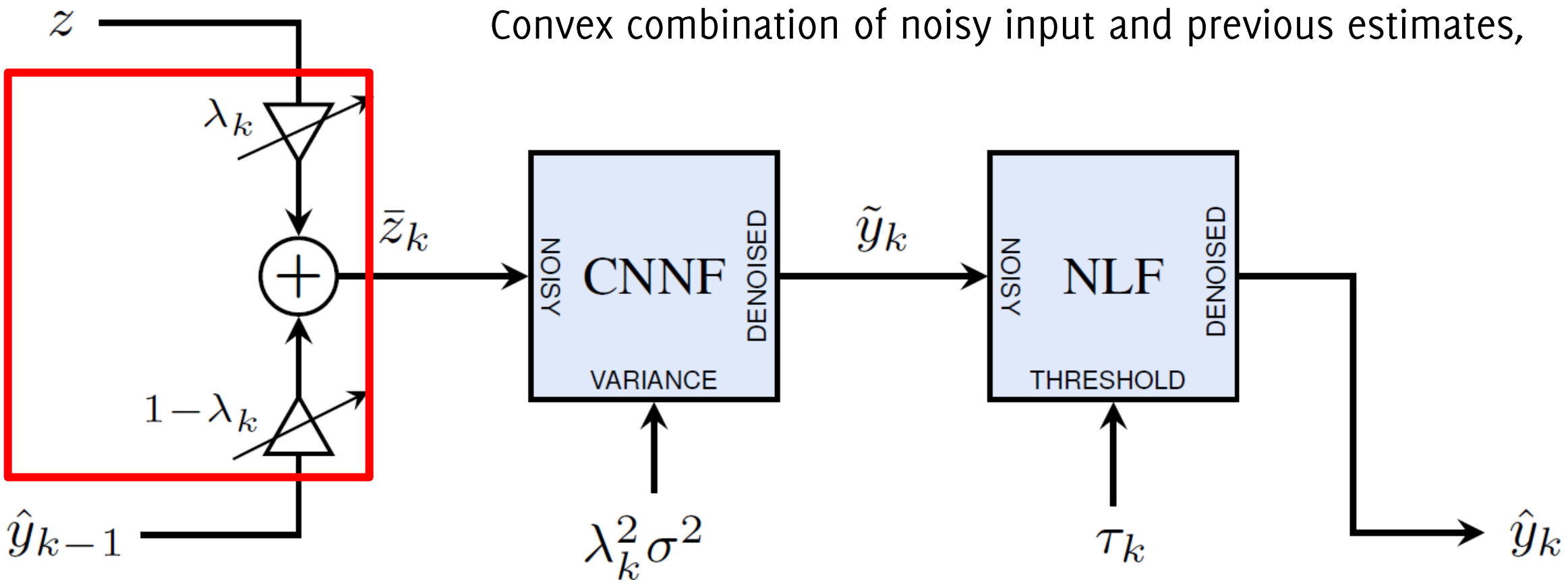
NN3D: the iterative scheme



Cruz, C., Foi, A., Katkovnik, V., & Egiazarian, K. (2018). Nonlocality-reinforced convolutional neural networks for image denoising. *IEEE Signal Processing Letters*, *25*(8), 1216-1220.

NN3D: the iterative scheme

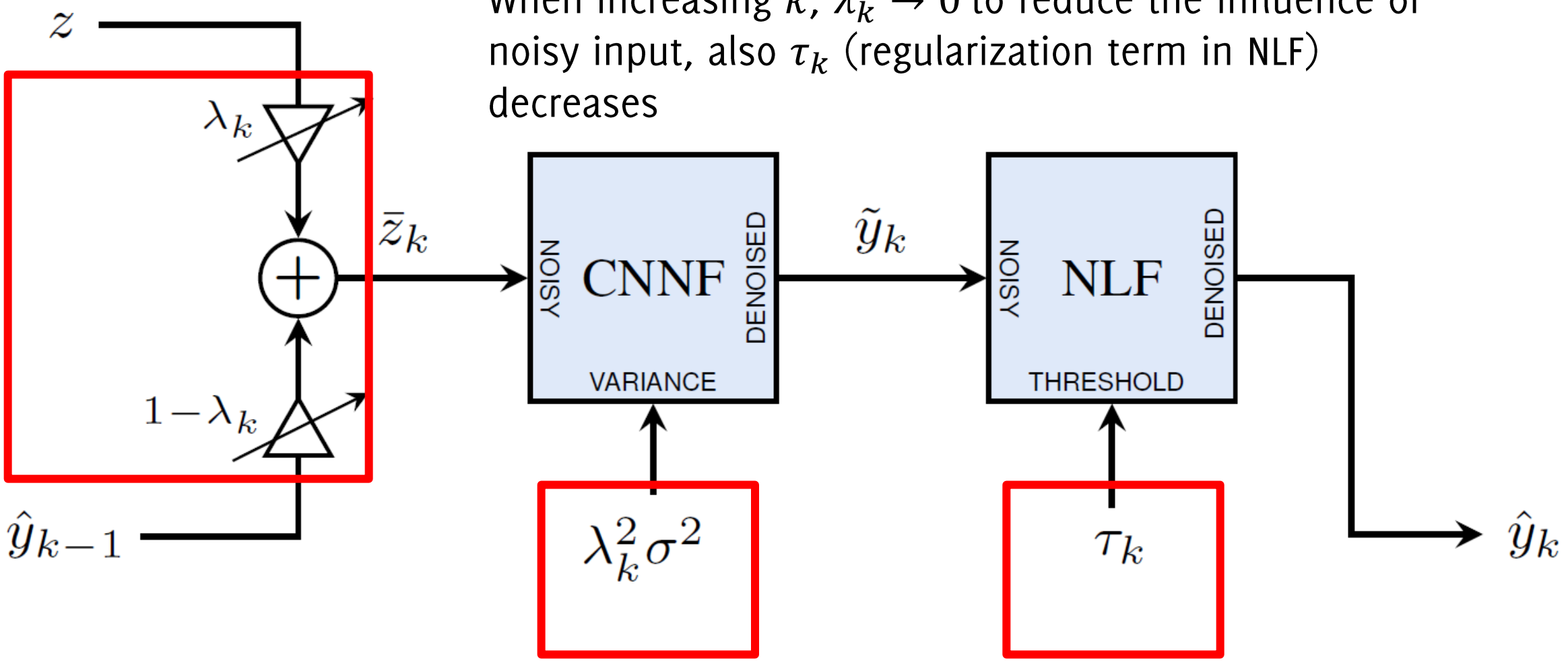
Convex combination of noisy input and previous estimates,



Cruz, C., Foi, A., Katkovnik, V., & Egiazarian, K. (2018). Nonlocality-reinforced convolutional neural networks for image denoising. *IEEE Signal Processing Letters*, *25*(8), 1216-1220.

NN3D: the iterative scheme

When increasing k, $\lambda_k \to 0$ to reduce the influence of noisy input, also τ_k (regularization term in NLF) decreases



Cruz, C., Foi, A., Katkovnik, V., & Egiazarian, K. (2018). Nonlocality-reinforced convolutional neural networks for image denoising. *IEEE Signal Processing Letters*, *25*(8), 1216-1220.

NN3D performance

NN3D uses grouping and group-wise Haar filtering, $K = 2$

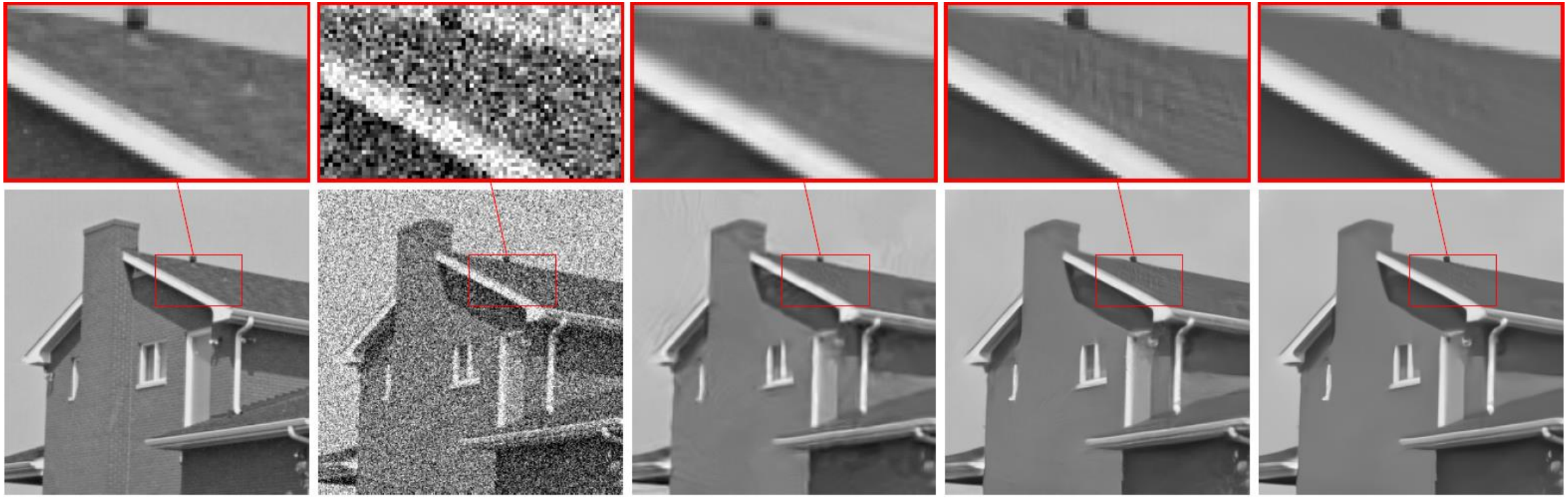
TABLE I

PSNR (dB) PERFORMANCE OF THE PROPOSED NN3D VS COMPETITIVE STATE-OF-THE-ART METHODS. ITALIC RESULTS IN THE BASELINE WDNCNN FOR $\sigma = 75$ INDICATE SCALING OF THE NOISY INPUT BY THE FACTOR 50/75 TO MATCH THE MODEL TRAINED FOR $\sigma = 50$.

Dataset	σ	BM3D	DnCNN [11]	NN3D(DnCNN)	FFDNet [18]	NN3D(FFDNet)	WDnCNN [15]	NN3D(WDnCNN)
Set12	30	29.12	29.54	29.66	29.63	29.66	29.91	29.96
	50	26.70	27.19	27.32	27.33	27.39	27.48	27.61
	75	24.90	25.22	25.43	25.51	25.61	*25.64*	25.81
BSD68	30	27.75	28.36	28.41	28.38	28.37	28.56	28.56
	50	25.63	26.23	26.27	26.30	26.29	26.39	26.42
	75	24.22	24.64	24.71	24.79	24.80	*24.85*	24.91
Urban100	30	28.75	28.88	29.23	29.06	29.18	29.84	30.04
	50	25.95	26.28	26.63	26.55	26.76	27.08	27.49
	75	23.93	23.99	24.45	24.53	24.81	*25.06*	25.53

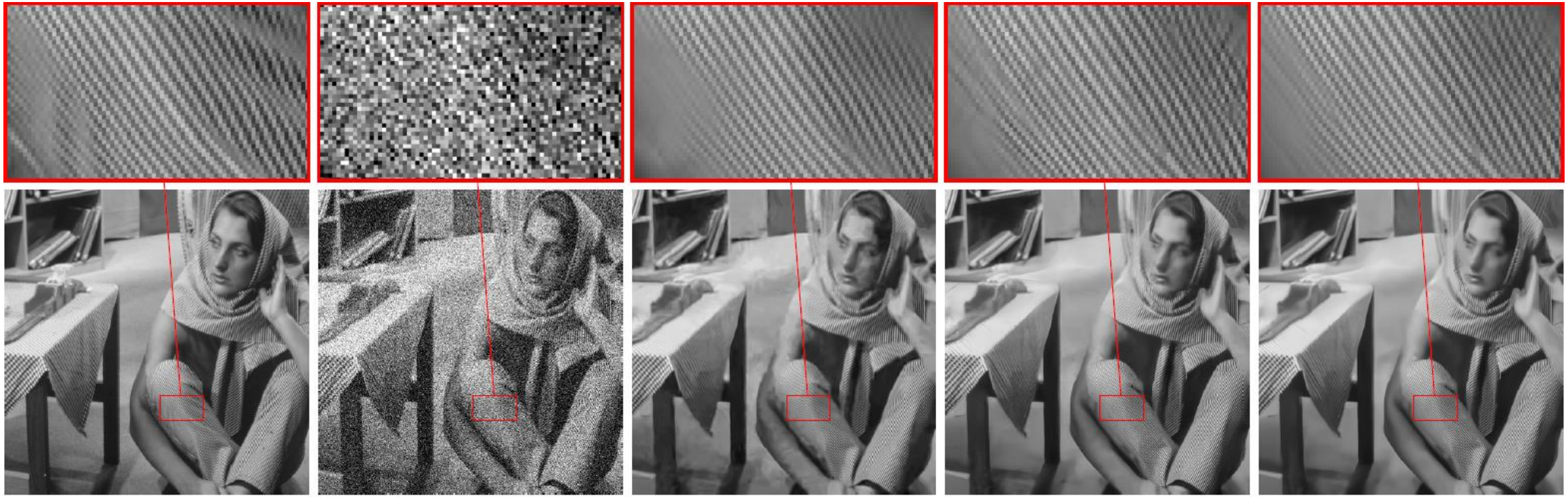
Cruz, C., Foi, A., Katkovnik, V., & Egiazarian, K. (2018). Nonlocality-reinforced convolutional neural networks for image denoising. *IEEE Signal Processing Letters*, *25*(8), 1216-1220.

NN3D performance



Cruz, C., Foi, A., Katkovnik, V., & Egiazarian, K. (2018). Nonlocality-reinforced convolutional neural networks for image denoising. *IEEE Signal Processing Letters, 25*(8), 1216-1220.

NN3D performance



Cruz, C., Foi, A., Katkovnik, V., & Egiazarian, K. (2018). Nonlocality-reinforced convolutional neural networks for image denoising. *IEEE Signal Processing Letters, 25*(8), 1216-1220.

Self-Supervised Denoising

Slide Credits Diego Martin

Why Self-Supervised?

Supervised image denoisers require **noisy-clean** image pairs

Clean images obtained by:

- Long exposure time on a camera
- Averaging several images of the same scene

Acquiring **noisy-clean** image pairs is hard in many real-world applications such as:

- Medical imaging → In CT scans, to reduce radiation exposure, we acquire fewer and noisier images
- Outdoor photography → Obvious problems in dynamic scenes

Self-supervised image denoising removes the need for clean images, opening the possibility for denoising in many more applications

Noise2Noise (N2N): Learning image restortion without clean data

For each noisy-clean image pair (z, y) denoisers are trained to minimize the loss:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, E\left[\mathcal{L}(CNN(\boldsymbol{\theta}; z), y)\right]$$

Typically $\mathcal{L}$ is the MSE, or the ℓ_1 different, or $SSIM(\cdot,\cdot)$

***Idea*:** As long as the noise is zero-mean and indepentent between the two noisy realization of the same image:

$$z = y + \eta, \qquad z' = y + \eta'$$

we can replace y with a different noise realization z' and optimize:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \underset{z,z'}{E}\left[\mathcal{L}(CNN(\boldsymbol{\theta}; z), z')\right]$$

On average, **the optimal function learned by the network still predicts the clean signal y.**

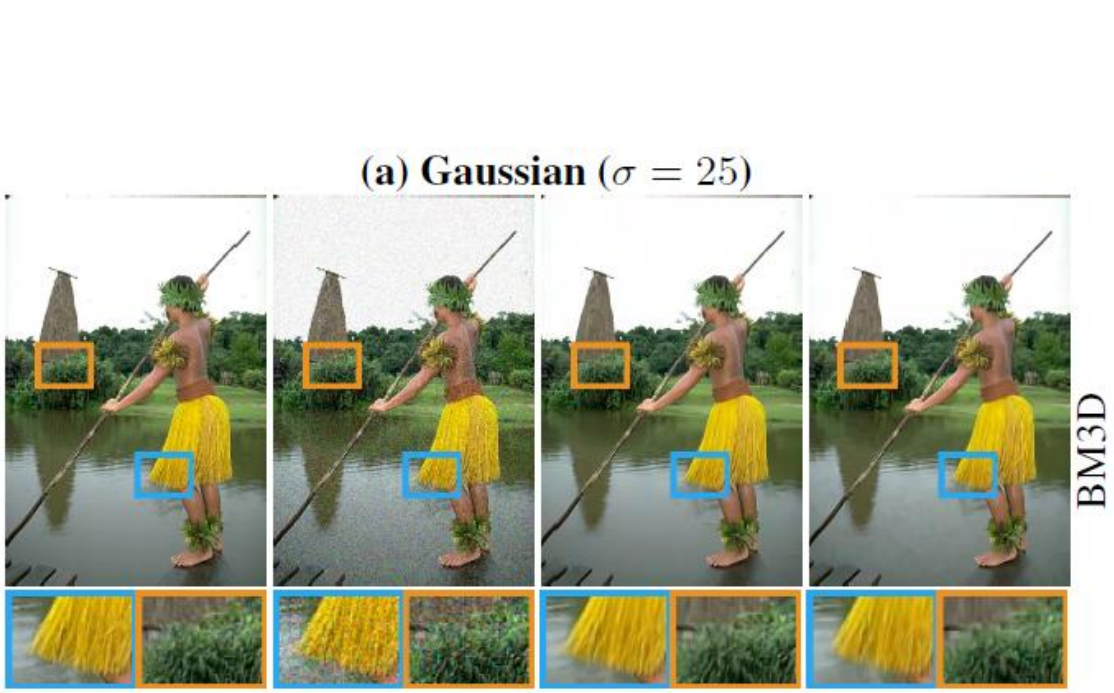
Advantages:

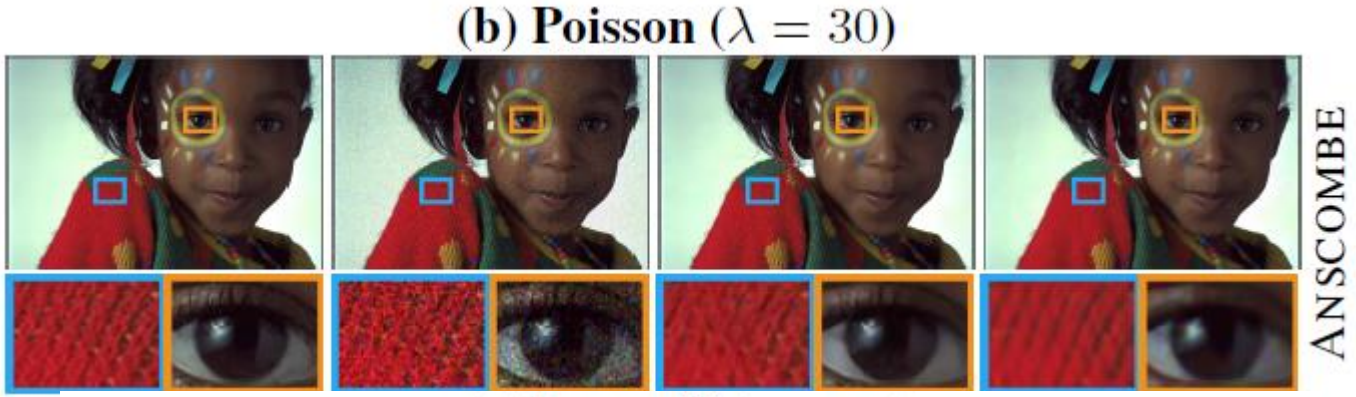
- No need of noise-free images
- No need of an explicit noise model to perturb images

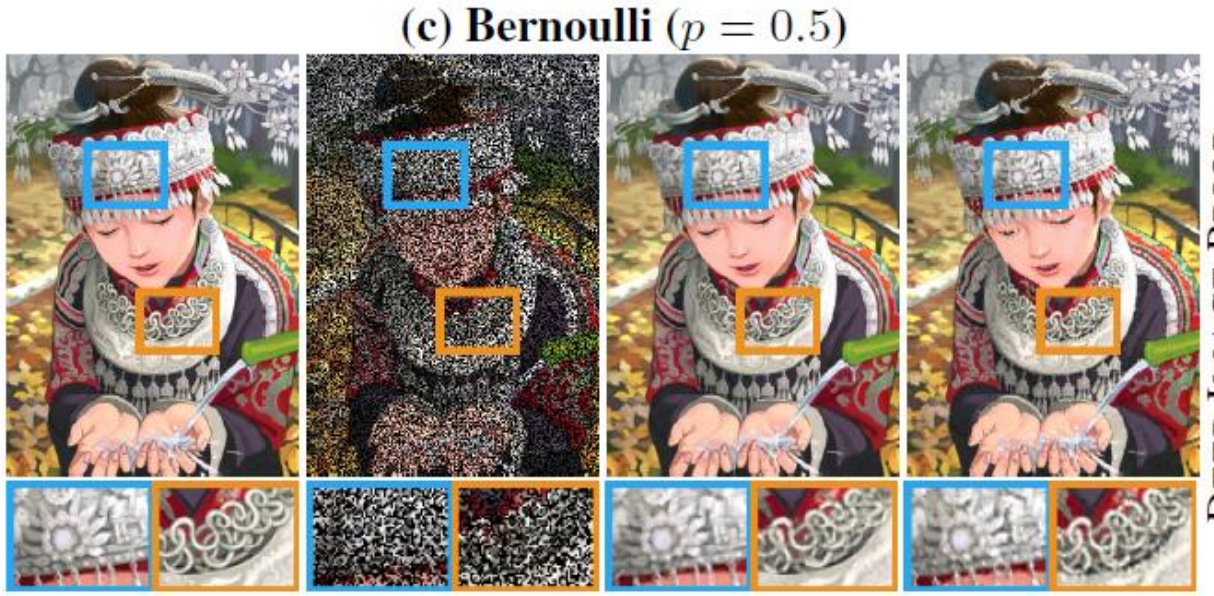
Lethinen et al. "Noise2Noise: Learning image restoration without clean data." ICML, 2018

Noise2Noise (N2N): Learning image restortion without clean data

Experiments showed denoising results comparable to the ones of supervised image denoisers, but without the need of clean data and noise modeling







From left to right: ground truth, noisy observation, N2N output, and competitor output for different types of noise (AWGN, Poisson and Bernoulli)

Lethinen et al. "Noise2Noise: Learning image restoration without clean data." in *International Conference on Machine Learning*. ICML, 2018

Noise2Noise (N2N): Learning image restortion without clean data

Impact: It proved that it is possible to learn denoising without clean images

Problem: Still needs pairs of images of the same underlying scene that might be unpractical in some applications (CT scans)

Solution: Self-Supervised image denoisers

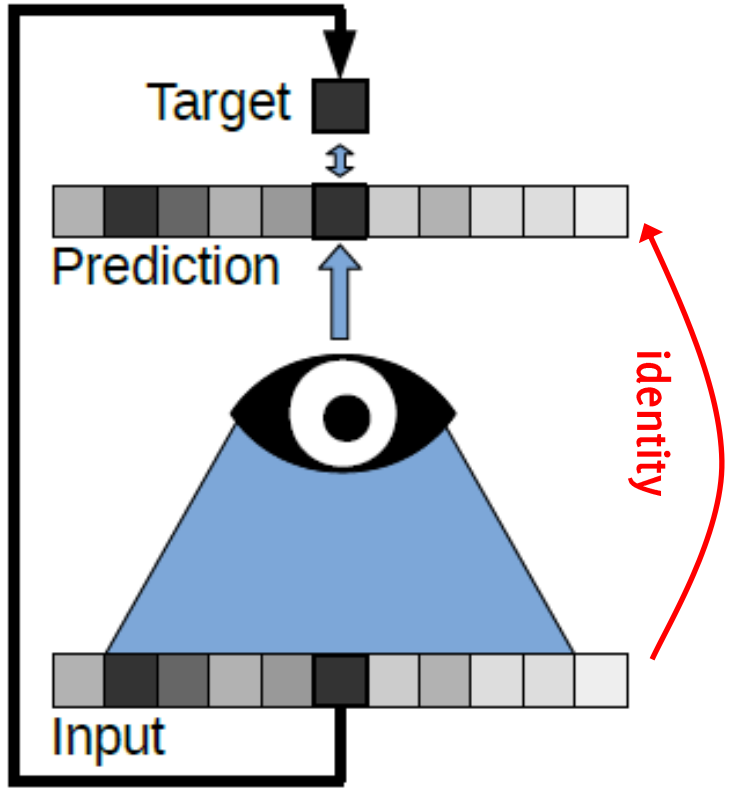
Lethinen et al. "Noise2Noise: Learning image restoration without clean data." in *International Conference on Machine Learning*. ICML, 2018

Noise2Void (N2V): Learning denoising from single noisy images

Self-supervised image denoiser trained on a training set of **unpaired noisy images**

Problem:
using the noisy image as both input and target leads the model to **learning the identity function**

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \underset{z}{E}[\mathcal{L}(CNN(\boldsymbol{\theta}; z), z)]$$



Solution:
Blind-Spot Network (BSN) training framework

Krull et al. "Noise2Void – Learning Denoising from Single Noisy Images." CVPR, 2019

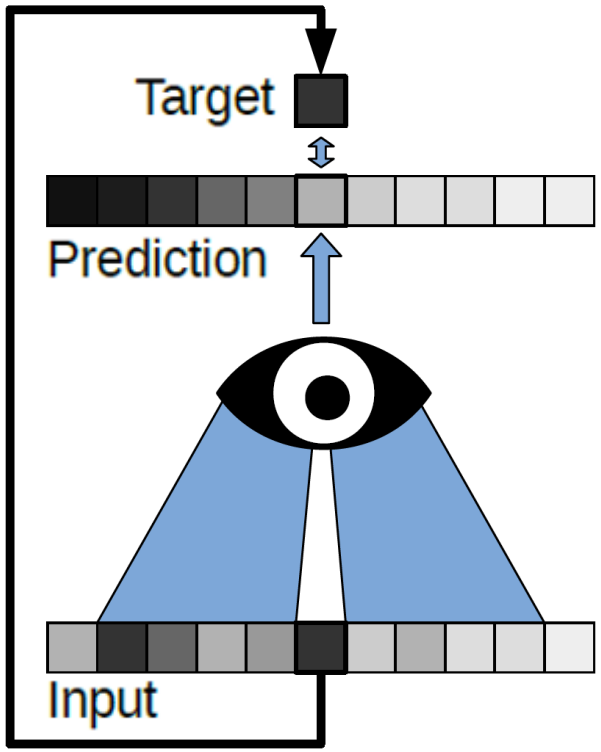
Blind-Spot Networks

BSN Framework Idea

- Hide some pixels in the receptive field of the network (**blind spots**)
  - Input: z with blind spots
  - Target: z noisy

Train to minimize loss function:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, \mathcal{L}(CNN(\boldsymbol{\theta}; z), z)$$



Underlying assumption:

- the noise free **image** y has **spatial correlation,**
- the **noise** is pixel-wise **independent** and zero-mean

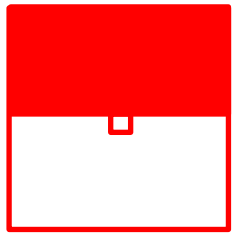
The network learns the clean values of the blind spots using only information coming from the receptive field.

Krull et al. "Noise2Void – Learning Denoising from Single Noisy Images." CVPR, 2019

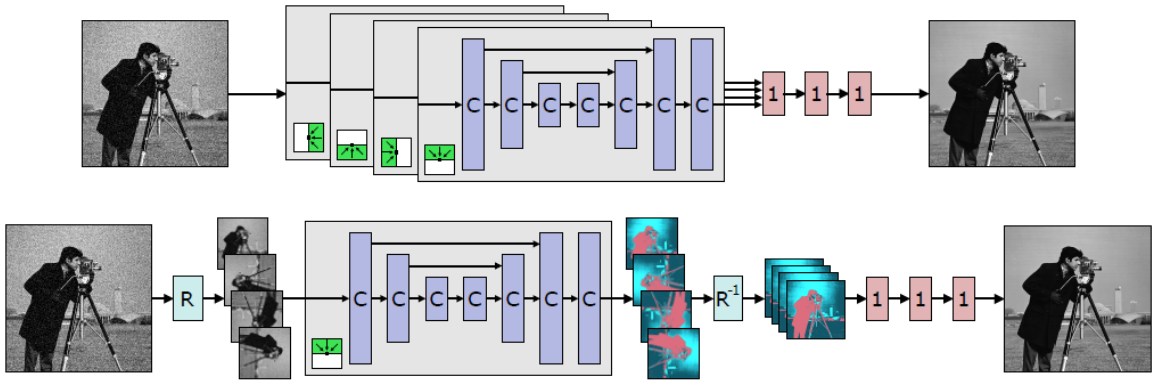
Blind-Spot Networks

There are 2 ways to construct a BSN:

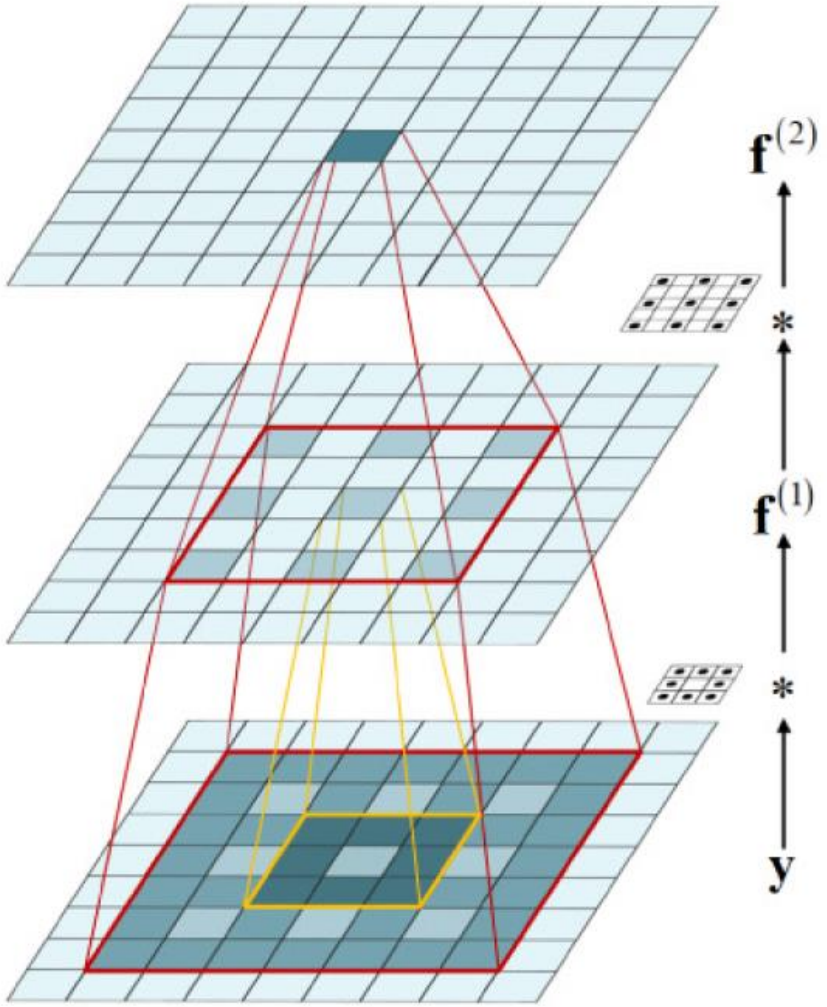
1. Designing a **specific network architecture** that generates blind-spots in the receptive field. Some examples are:
   - Half-plane convolutions
   - Dilated convolutions



Convolutions on half-planes to omit the central pixel



Dilated convolutions + holes in the filter



Laine et al. "High-Quality Self-Supervised Deep Image Denoising" ANIPS, 2019

Blind-Spot Networks the N2V way

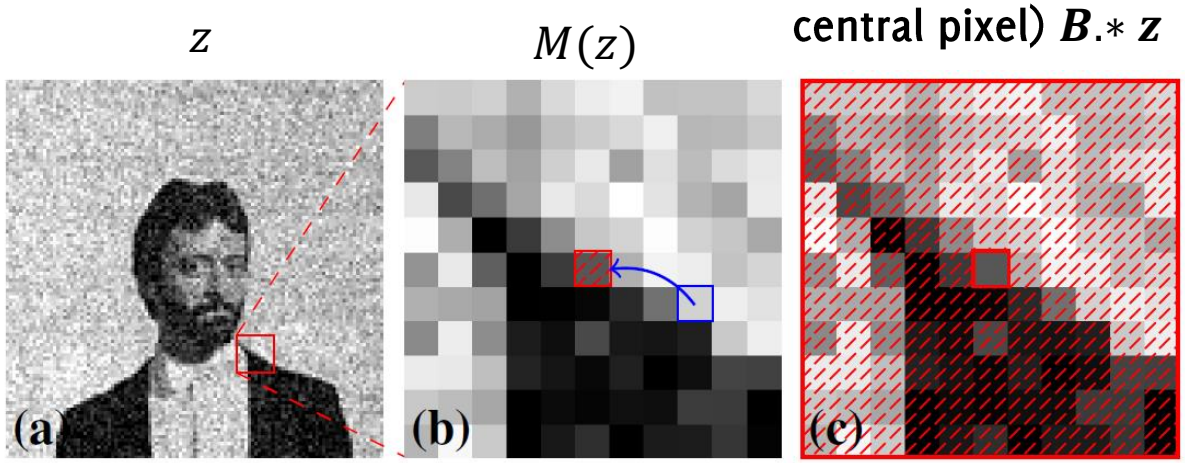
2. Apply a **masker**, a function that replaces (masks) some pixels of the input image, as a preprocessing step before feeding the image to the network.

There are different ways to decide the new blind-spot (BS) value, N2V proposes:

- Value of a random pixel in the neighborhood of the BS
- Mean of a small region centered in the BS
- Median of a small region centered in the BS
- Replace with a random value

In this case, it is better to focus the loss directly on the blind spots. Hence, the loss function becomes:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \mathcal{L}(B.* CNN\big(\boldsymbol{\theta}; M(z)\big), B.* z)$$

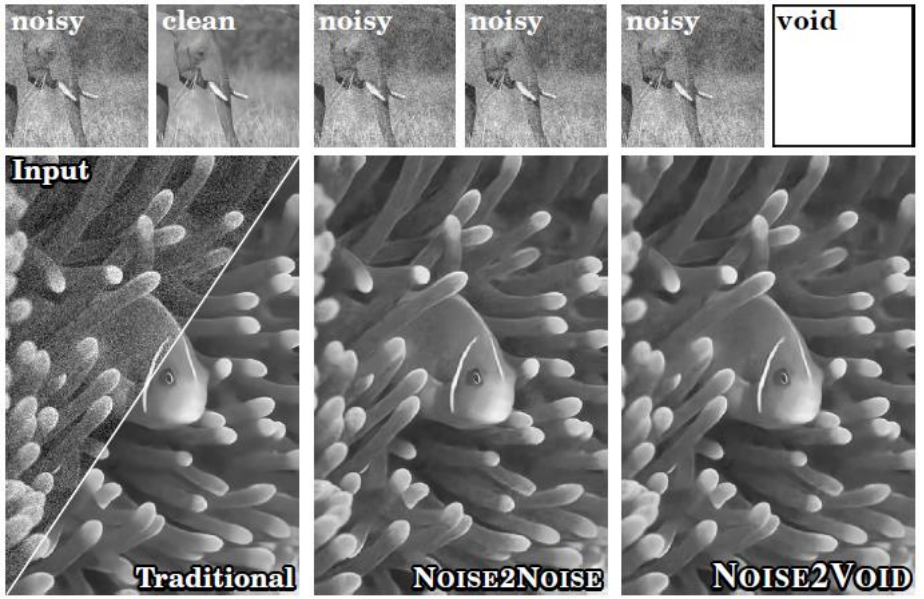


Krull et al. "Noise2Void – Learning Denoising from Single Noisy Images." CVPR, 2019

Noise2Void (N2V): Learning denoising from single noisy images

Slightly inferior performance compared to N2N (less information is provided indeed)

N2V outperforms traditional non-learning based denoisers and opened the possibility for self-supervised image denoising



Limitation:

The BSN framework does not perform well in case of spatially correlated noise

Krull et al. "Noise2Void – Learning Denoising from Single Noisy Images." CVPR, 2019

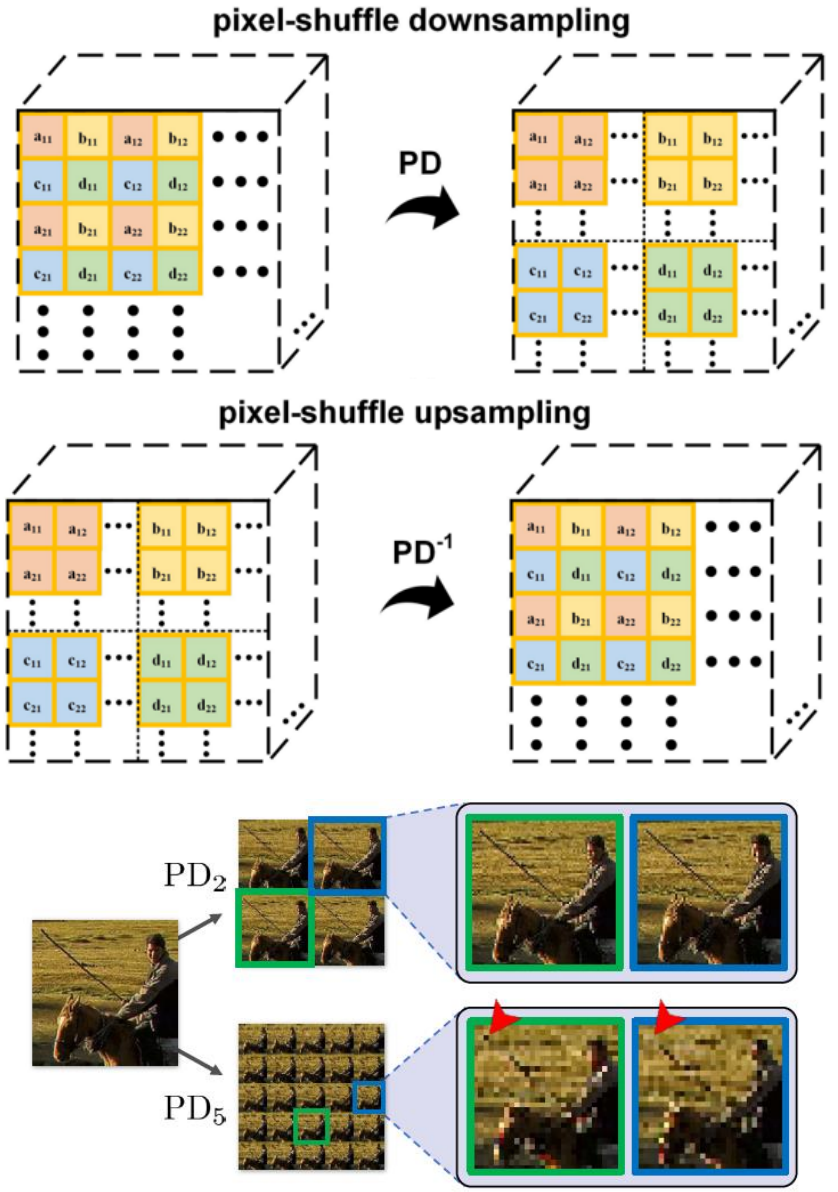
Self-supervised denoising with spatially correlated noise

For many real-world application, noise presents spatial correlation between pixels

Problem: BSN's fundamental assumption is broken and the noise correlation can negatively affect the model's prediction at each blind spot.

Idea: use BSNs after some decorrelation techniques (e.g. Pixel-Shuffle Downsampling)

By pixel-shuffle downsampling create a tile of small images. Noise becomes uncorrelated, image content remains correlated.



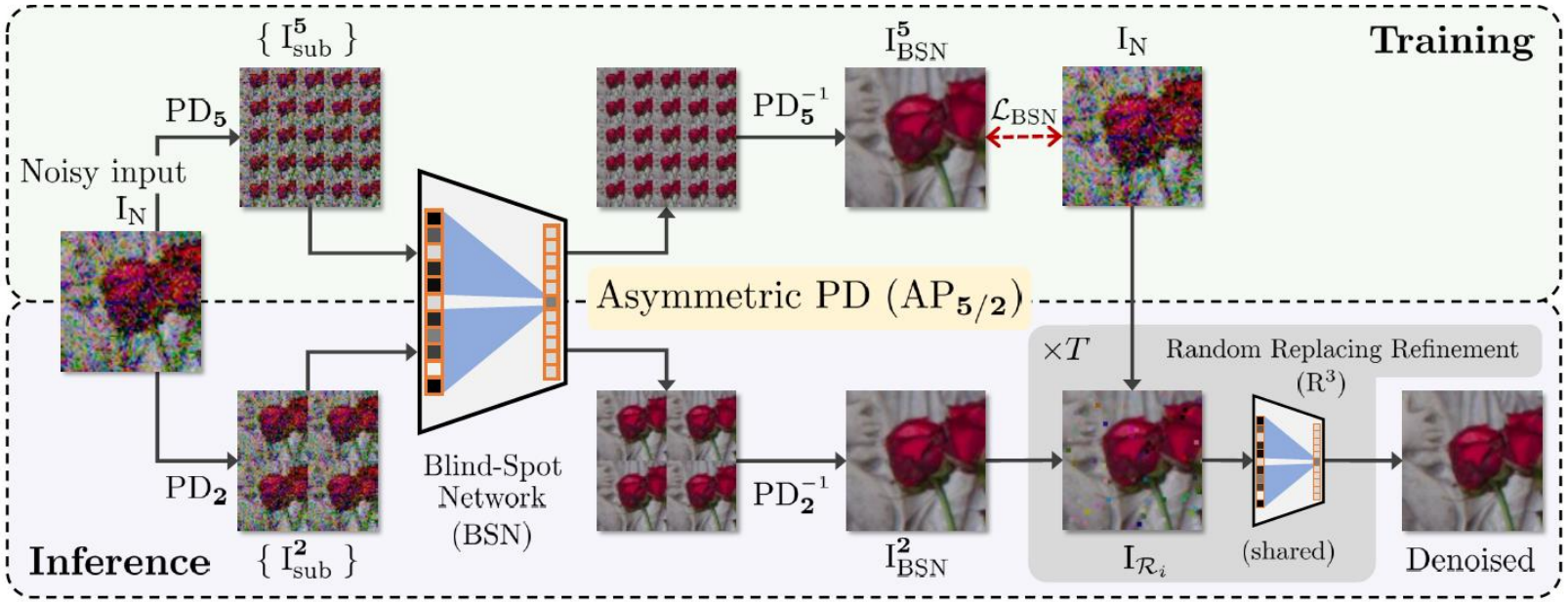
Lee et al. "AP-BSN: Self-Supervised Denoising for Real-World Images via Asymmetric PD and Blind-Spot Network", CVPR. 2022

Self-supervised denoising with spatially correlated noise

AP-BSN training pipeline:

- Decorrelate noise through PD with stride 5
- Train BSN on the decorrelated images, obtaining downsampled outputs
- Invert the downsampling operation, obtaining the final output

At inference time, smaller stride is used.



Lee et al. "AP-BSN: Self-Supervised Denoising for Real-World Images via Asymmetric PD and Blind-Spot Network", CVPR. 2022

CNN for Image Restoration

- The research activity in the field has become very vivid in the last few years.
- The performance improvement achived by deep CNN was not as clear as for visual recognition problem
  - Now these achieve the state of the art in terms of performance
  - Still, traditional algorithms are more efficient. This is crucial given the large image sizes (compared to visual recognition).
- Architecture of denoisers are becoming more and more complex.
- Research directions include
  - Efficient solutions
  - Practical training options (without using a noise model or clean images)
  - Robustness to noise models

CNN for Image Restoration

Research is often inspired by traditional priors, designing neural architectures able to promote these

- Self-similarity
- Sparsity
- Multiscale processing

CNN are now more recently used as powerful prior for natural images, inside iterative denoising schemes (Residual Unfolding Networks)

Often performance are heavily influenced by the training power (unlimited training data thanks to BSN)