Neural Network Architectures for Images

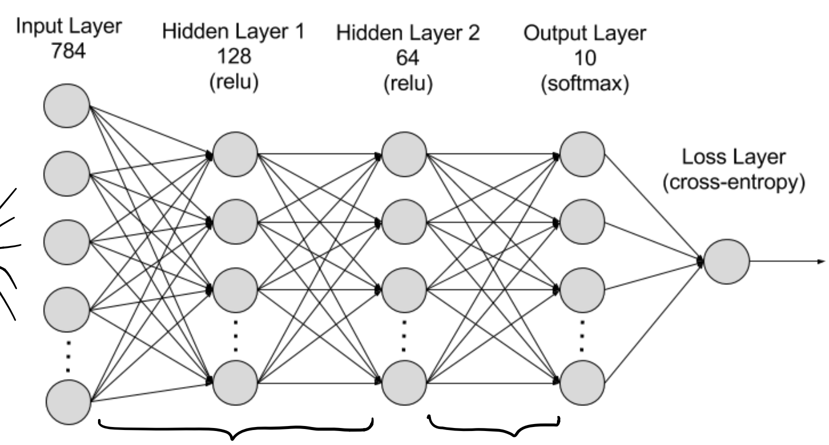
by Paul Hand
Northeastern University

Outline

- MLPs
- CNNs
- ResNets
- Encoder-decoder nets
- Autoencoders

Multilayer Perceptrons (MLPs)

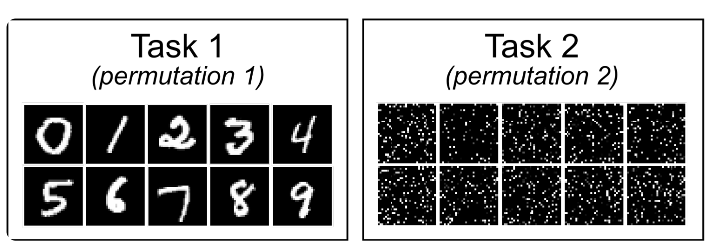
pixels



nonlinear, learned representation

linear classification

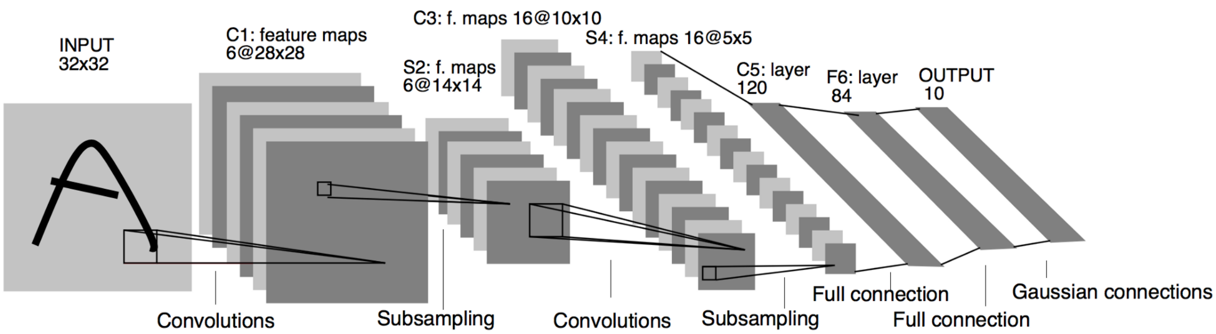
Careful:
- can't alter input size
- no geometry/locality of input is enforced
- permutation invariance



Convolutional Neural Networks + MLPs

Context: Classification

(Lecun et al. 1998)



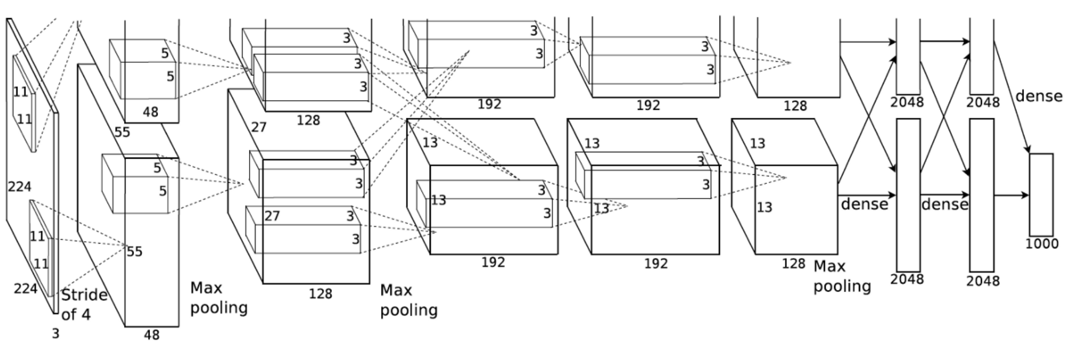
CNN MLP

Convolution Layers are well suited for images

- Leverages locality/geometry of images
- Enforce translation invariance
- Fewer parameters than MLP
- Can handle images of arbitrary size
  (though following MLP can not)

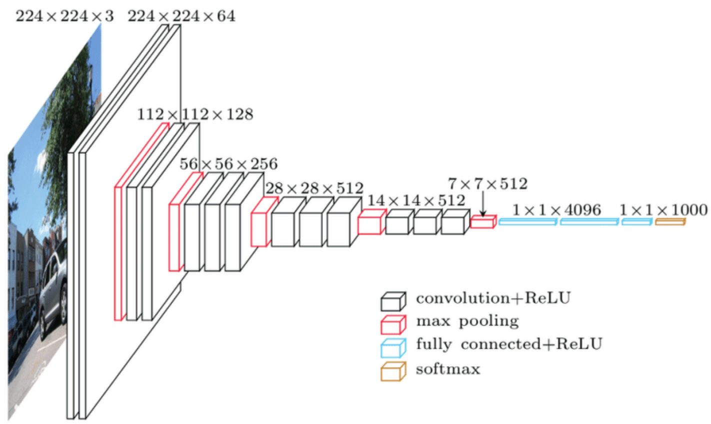
AlexNet:

(Krizhevsky et al. 2012)



VGG Net

(Simonyan + Zisserman 2015)



Vanilla CNNs

Context: Image Restoration

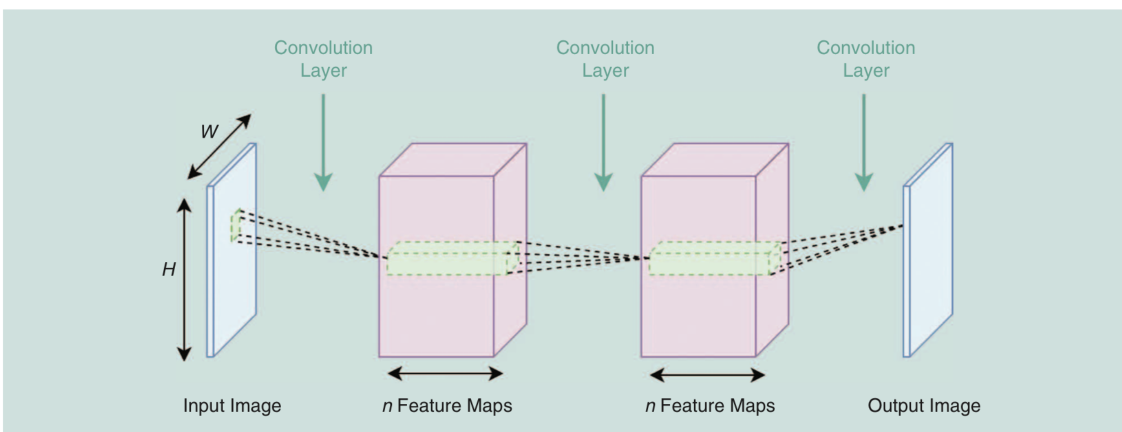


FIGURE 3. A three-layer CNN with successive convolutional layers, where the spatial dimensions of the feature maps match those of the input and output images. Following each convolution there is a nonlinearity operation, not shown here.

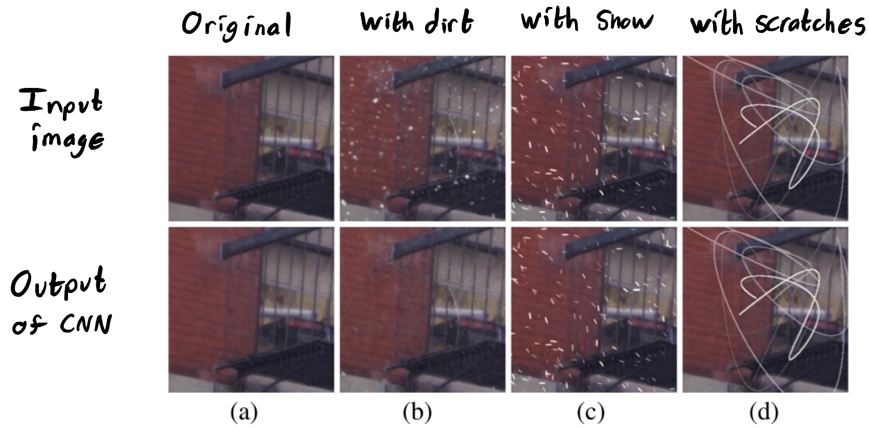
(Lucas et al. 2018)

Can retain image size

Can operate on image of any size

Be careful about
- receptive fields
- image scale

Dirt removal (Eigen et al. 2013)



Blind Deconvolution (Hradiš et al. 2015)

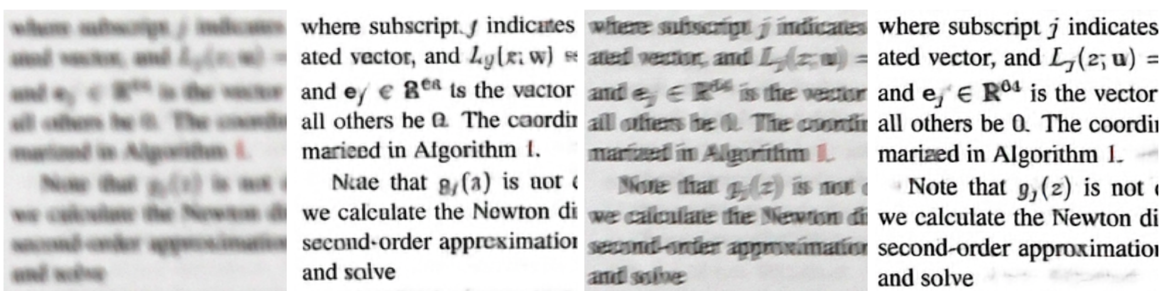

where subscript j indicates
ated vector, and $L_j(z; \mathbf{u})$ =
and $\mathbf{e}_j \in \mathbf{R}^{64}$ is the vector
all others be 0. The coordi
marized in Algorithm 1.
Note that $g_j(z)$ is not
we calculate the Newton di
second-order approximatio
and solve

Table 1: CNN architecture – filter size and number of channels for each layer.

Layer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L15	19×19	1×1	1×1	1×1	1×1	3×3	1×1	5×5	5×5	3×3	5×5	5×5	1×1	7×7	7×7
	128	320	320	320	128	128	512	128	128	128	128	128	256	64	3

Residual Blocks + Skip Connections

(Lucas et al. 2018)

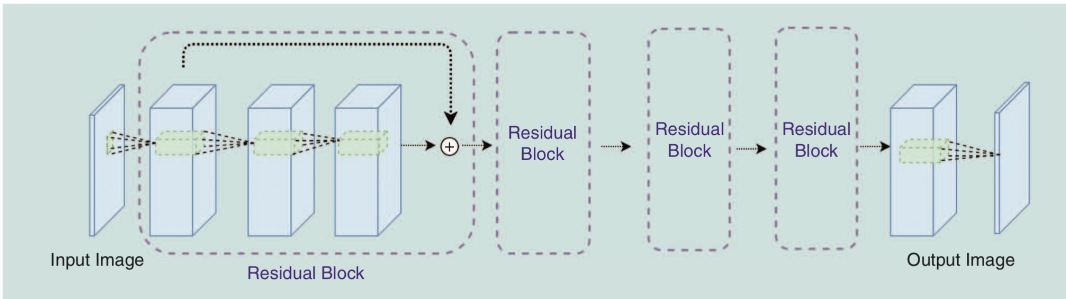


FIGURE 4. An example of a deep residual CNN. Each residual block, consisting here of three convolutions, learns a residual between its input and its output.

ResNet (for classification):

(He et al. 2015)

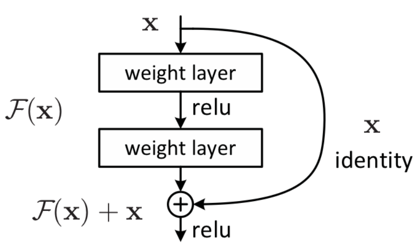
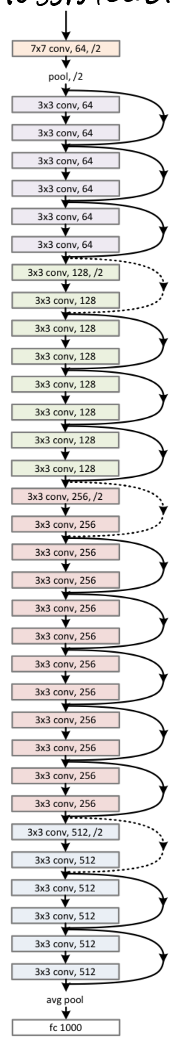


Figure 2. Residual learning: a building block.

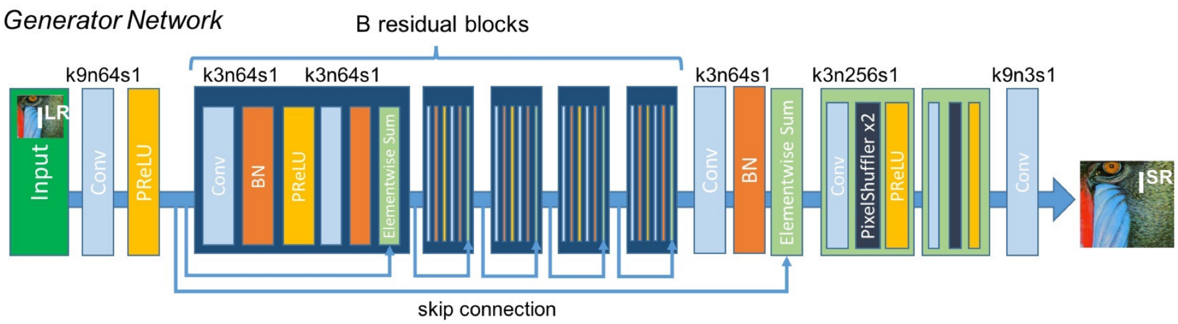
Ensure that enough computation is performed before adding back



Residual blocks allow increased depth

Increased depth allows
- large receptive fields
- Smaller filter sizes
  (Savings in parameter count)

ResNet (for Superresolution): (Ledig et al. 2017)

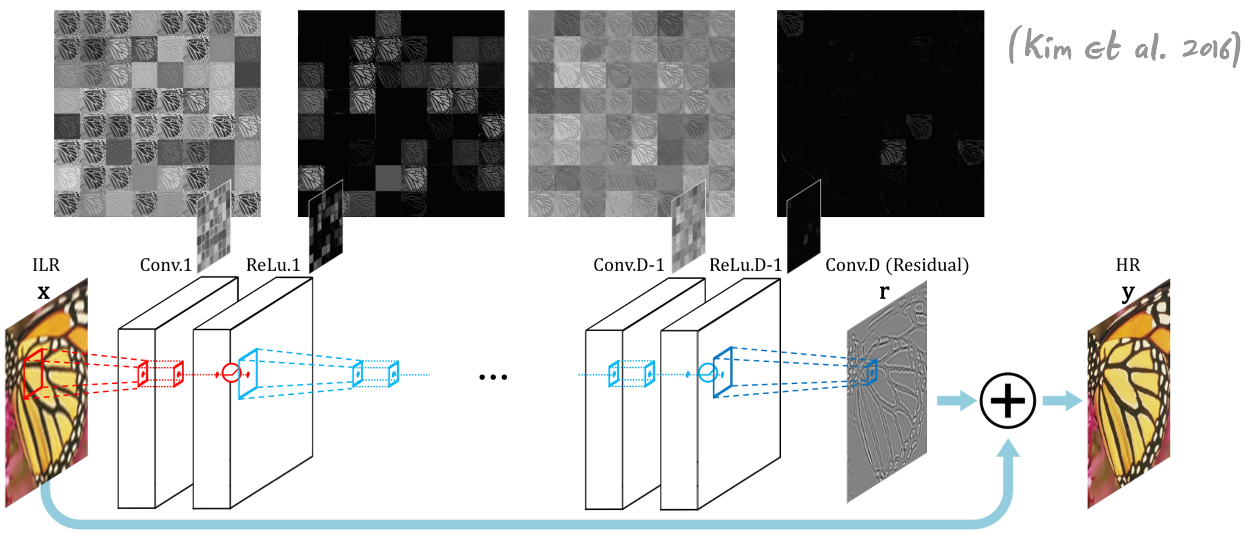


bicubic (21.59dB/0.6423) SRResNet (23.53dB/0.7832)

ResNets are good at

- Successive refinement of images

Idea: Do superresolution by only learning residual



Learning a residual may be easier than learning a full mapping

analogy: fixing a painting

Encoder-decoder Nets

(Lucas et al. 2018)

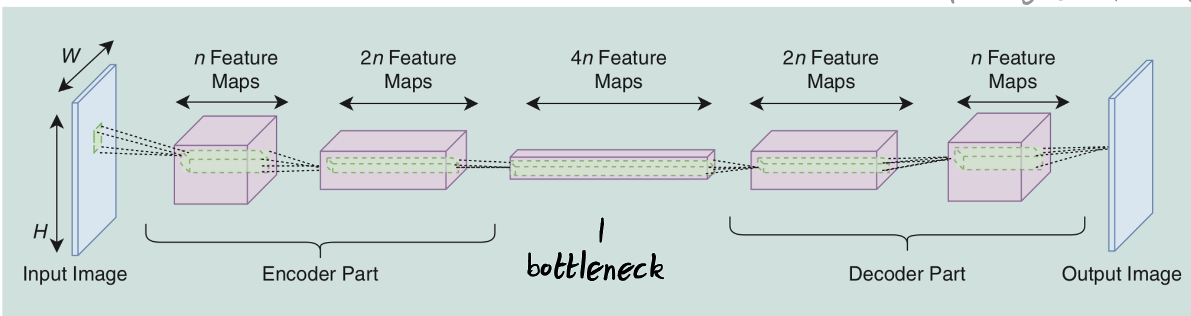


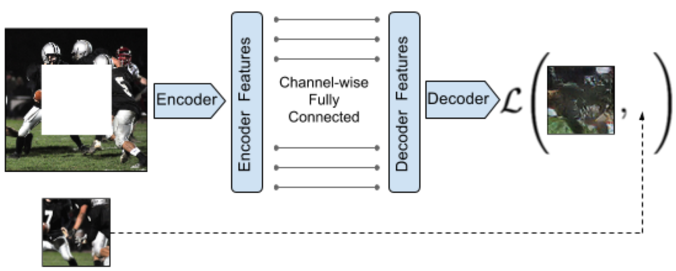
FIGURE 5. In an encoder-decoder CNN, the feature maps are spatially compressed by an encoder network, then increased back to the size of the output image by a decoder network.

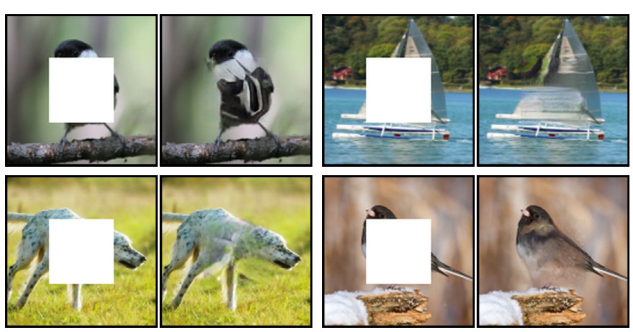
Encoder converts image to a set of latent variables (a code)
Decoder generates image from this code

Bottleneck forces net to gain semantic understanding of image

Encoder-decoder net for inpainting:

(Pathak et al. 2016)



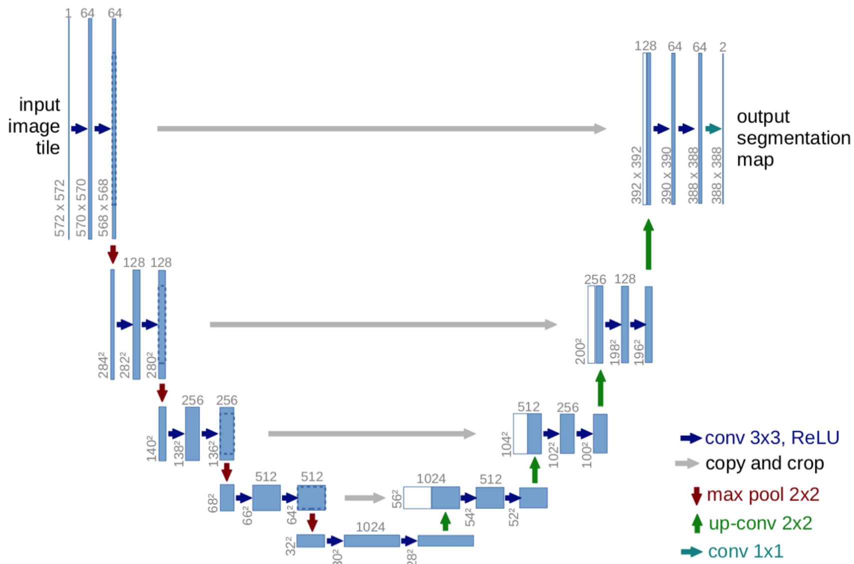


Combining encoder-decoder nets w/ Skip connections

The compressive nature of an encoder-decoder bottleneck may result in loss of detail in decoded images.

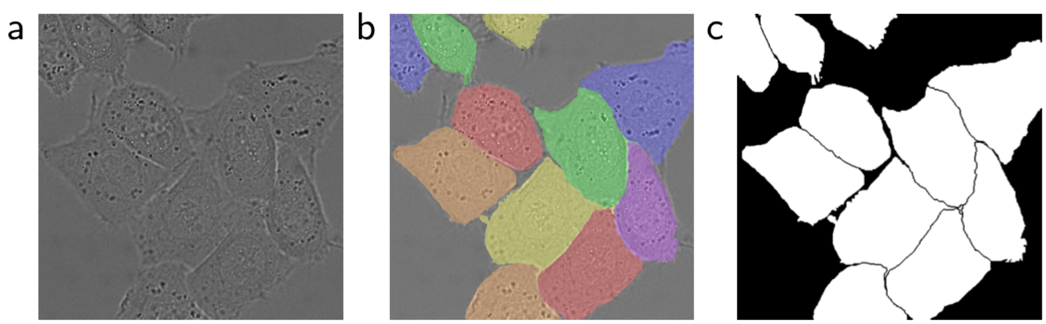
Idea: fix with skip connections

U-net



(Ronneberger et al. 2015)

Segmentation (biomedical images)

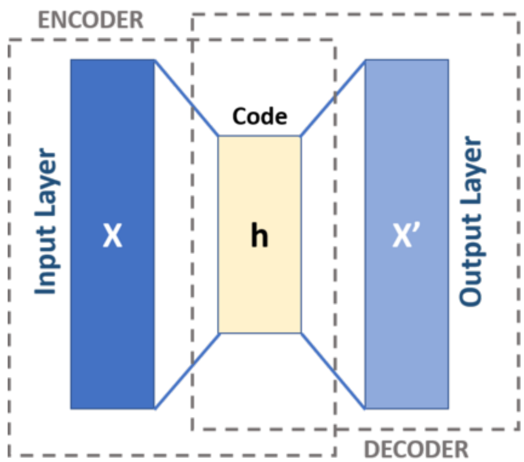


Strength of U-nets:

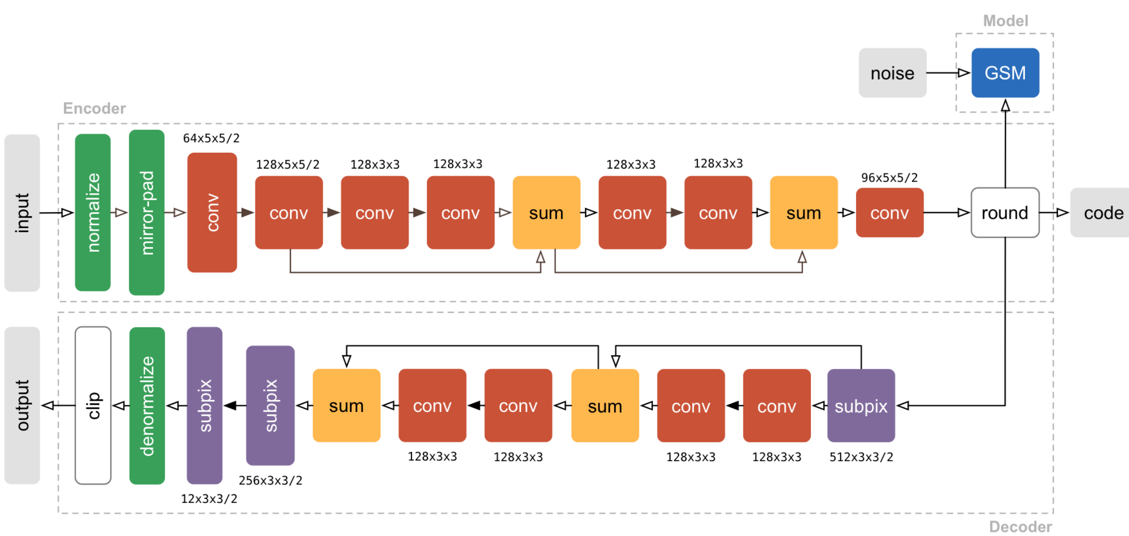
- Semantic understanding by bottleneck
- Skip connections preserve detail

Auto encoders

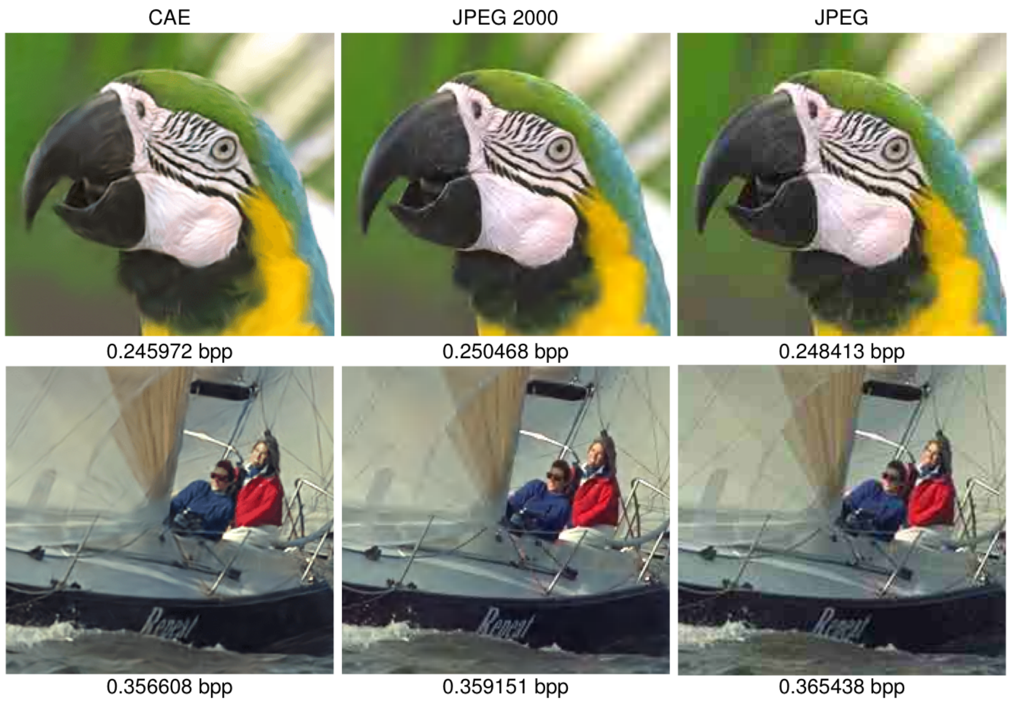
Context: unsupervised representation learning



With compressive autoencoder, one can compress images

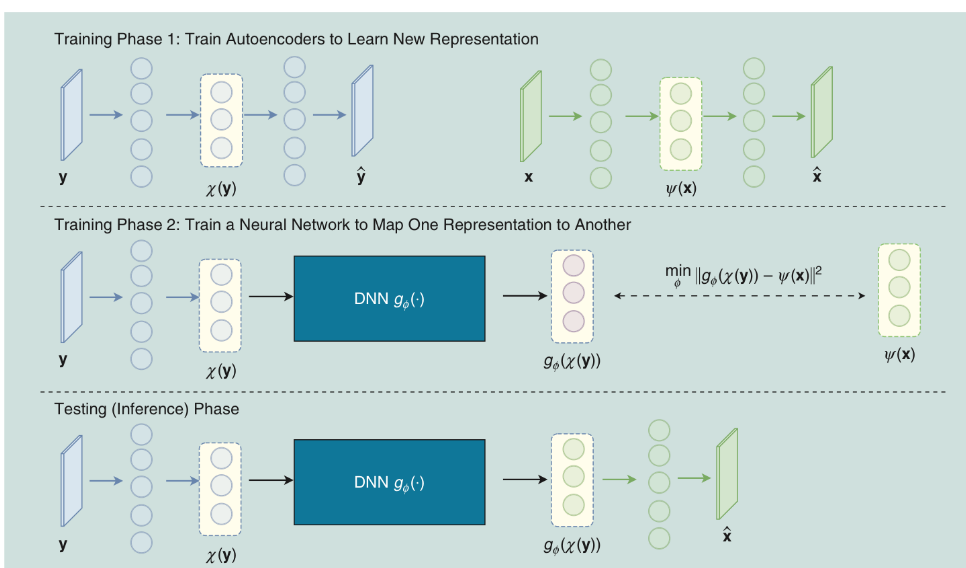


(Theis et al. 2017)



(Theis et al. 2017)

Learned autoencoder representations can be used for other tasks



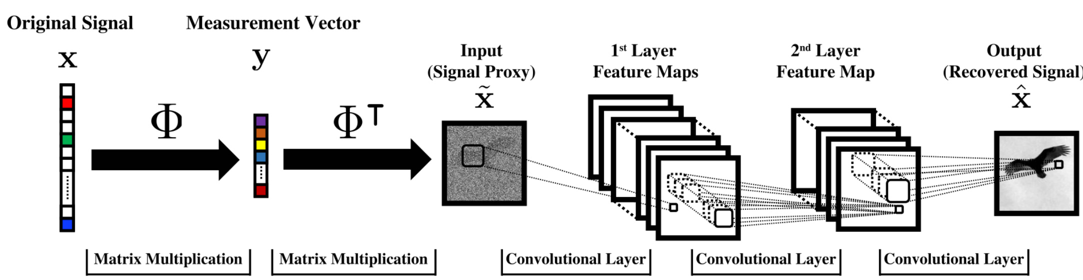
(Zeng et al. 2017)

FIGURE 6. An example of an approach in which new representations for images are learned, prior to solving the reconstruction problem in a supervised way. In Zeng et al.'s work [37], autoencoders first learn new features for the LR and HR patches (training phase 1). An MLP is then trained to map the representation of the observed LR patch to that of the HR patch (training phase 2). The final HR patch can be obtained with the second half of the autoencoder trained to reconstruct HR images (testing phase).

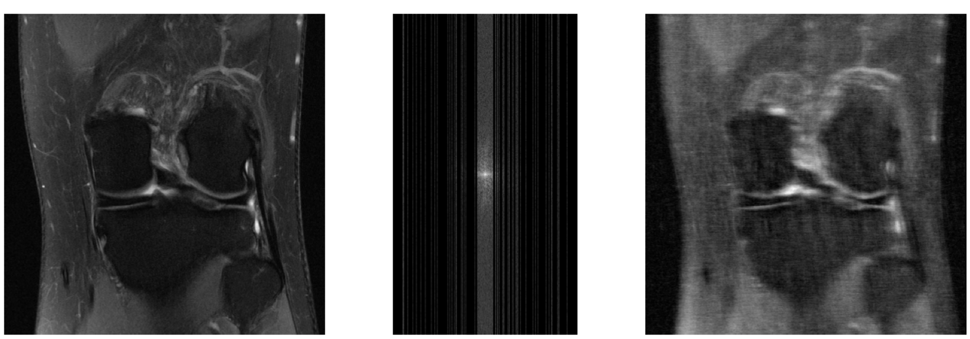
One last idea: Spare nets from learning things you know

(fully end-to-end trained nets may not be the best approach to solving a problem)

Compressed Sensing: (Mousavi + Baraniuk 2017)



MRI reconstruction: (Zbontar et al. 2019)



(a) Cropped and vertically flipped reconstruction from fully sampled k-space data

(b) Rectangular masked k-space

(c) Reconstruction via zero-filled IFFT