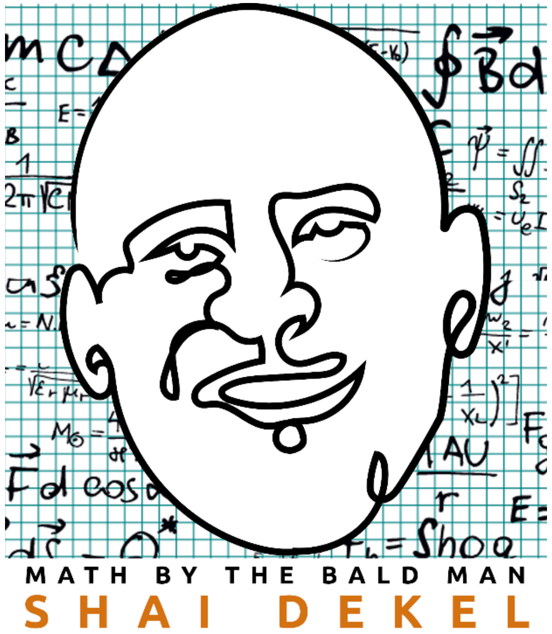
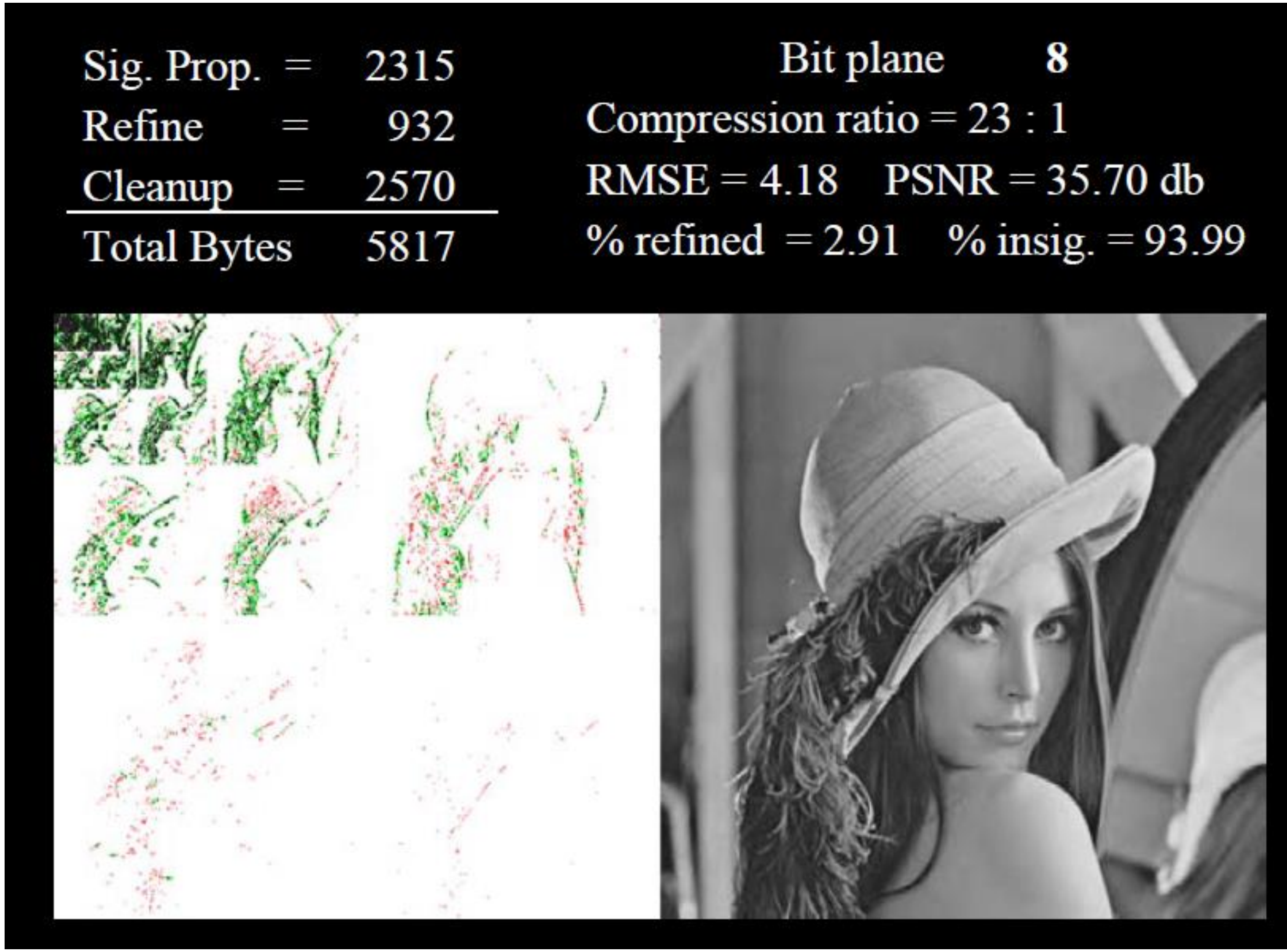
Mathematical foundations of ML
Spring 2020

Shai Dekel



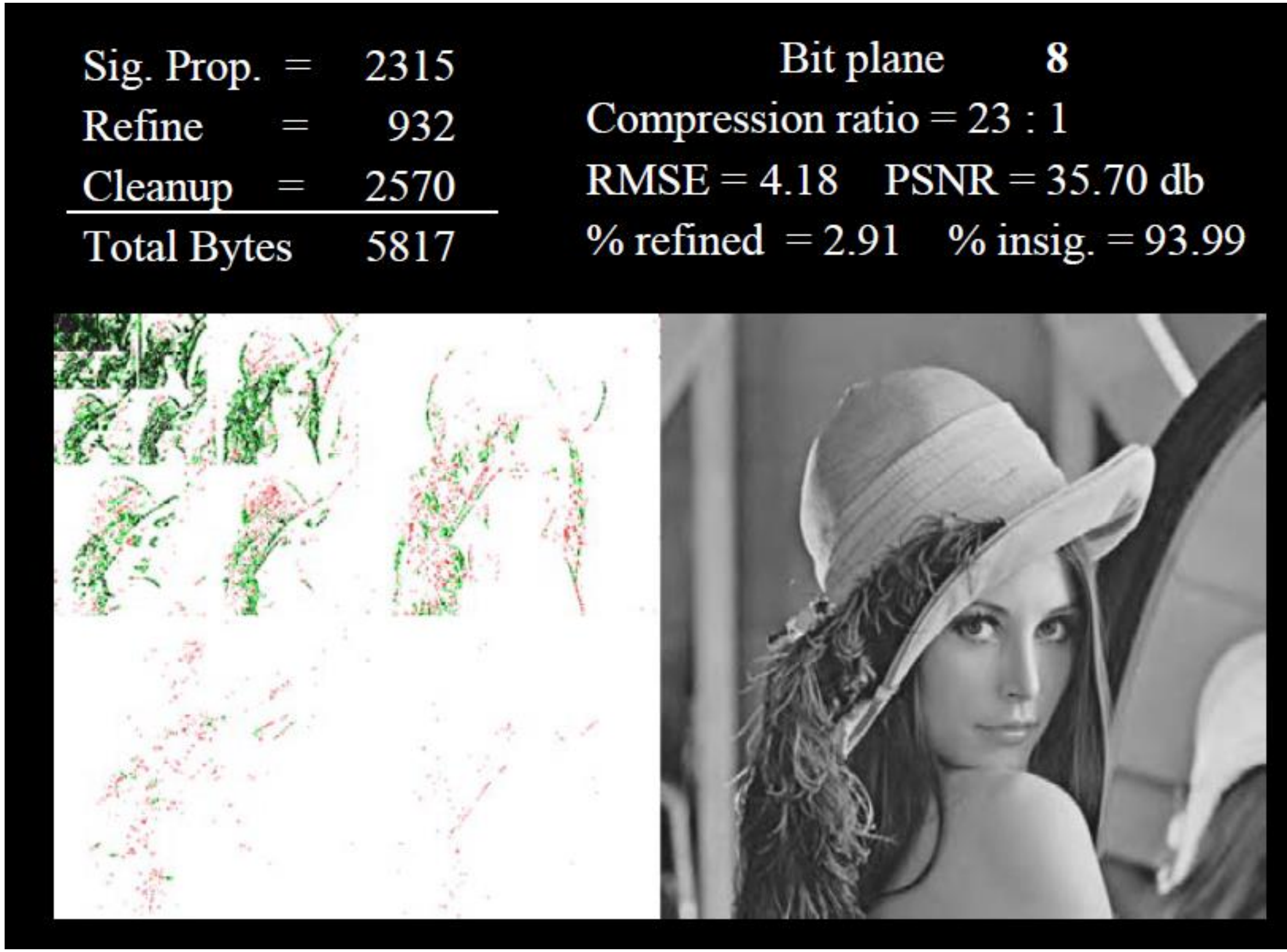
Mathematical foundations of signal processing

- Characterization of performance of models/algorithms using function space (weak-type) smoothness.



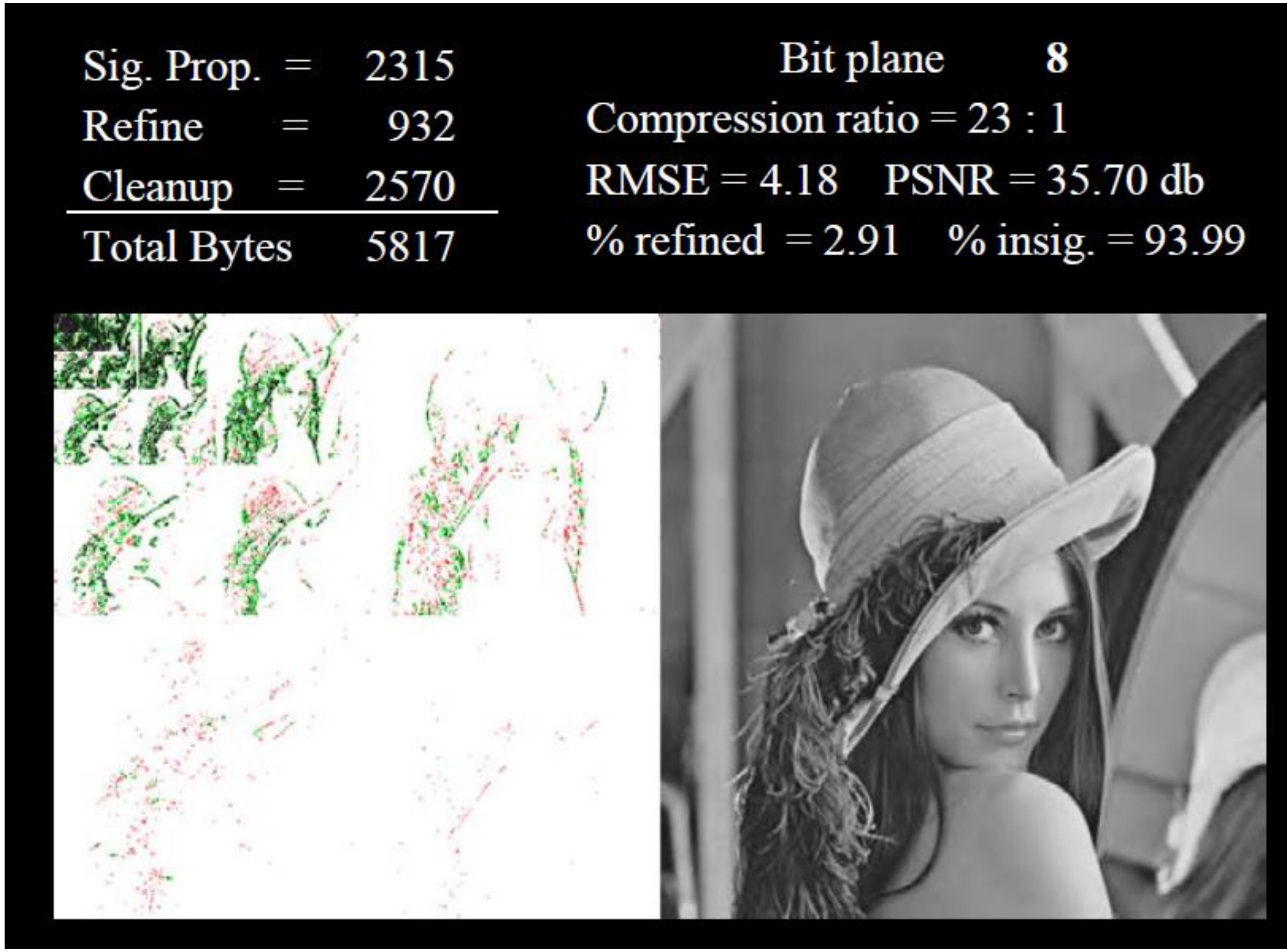
Mathematical foundations of signal processing

- Characterization of performance of models/algorithms using function space (weak-type) smoothness.
- Example: Performance of wavelet image compression characterized by Besov smoothness of image (as a function).



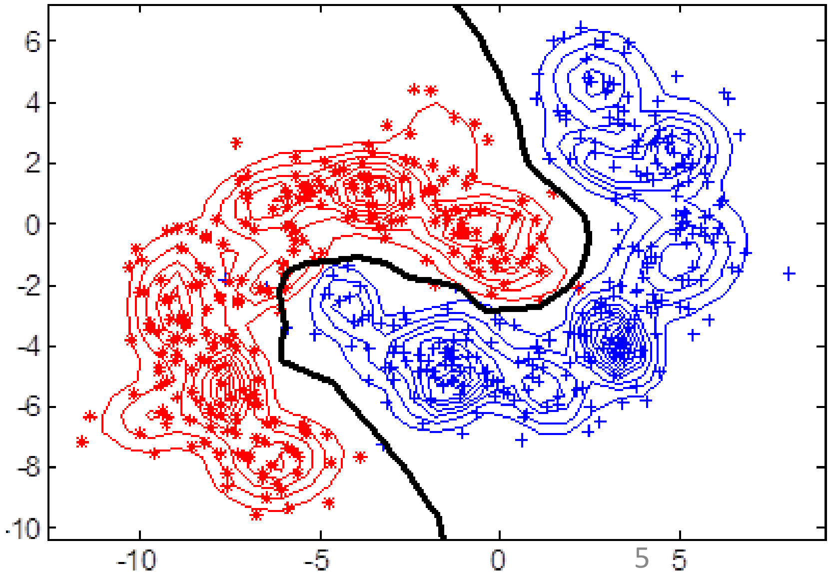
Mathematical foundations of signal processing

- Characterization of performance of models/algorithms using function space (weak-type) smoothness.
- Example: Performance of wavelet image compression characterized by Besov smoothness of image (as a function).
- **It all has to do with 'geometry' of the data.**



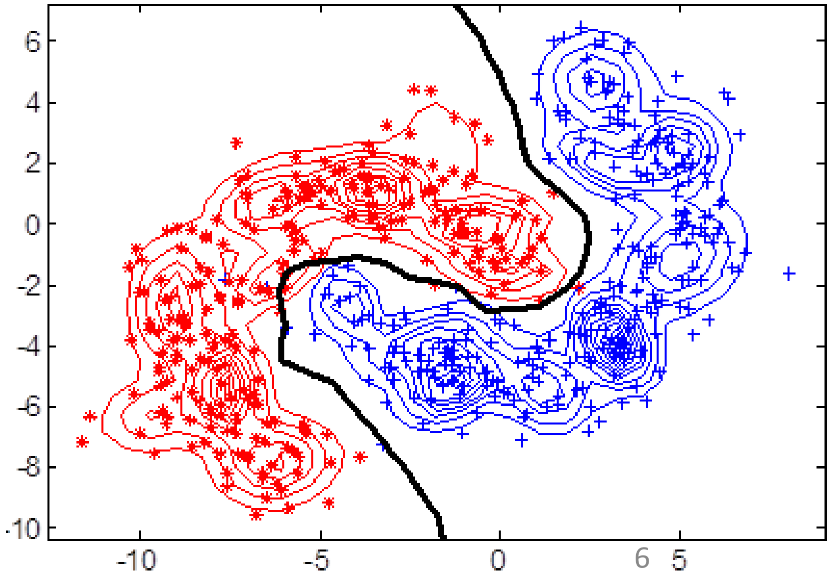
Mathematical foundations of AI

- Is there ‘geometry’ of clusters in the feature space?



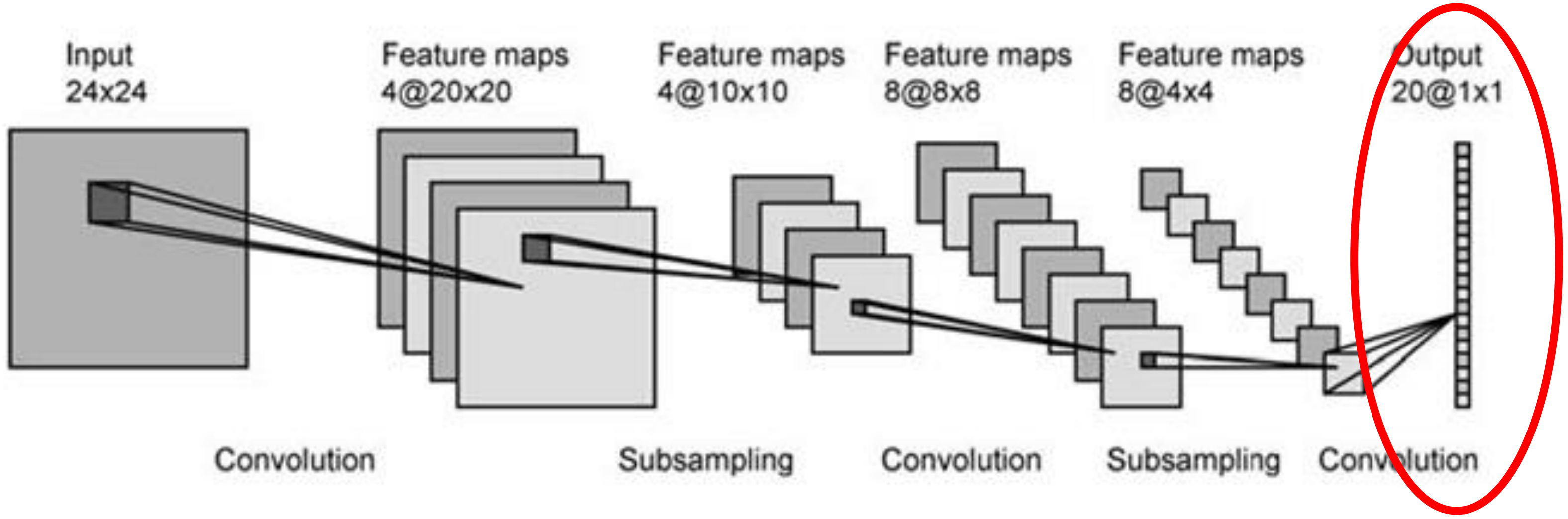
Mathematical foundations of AI

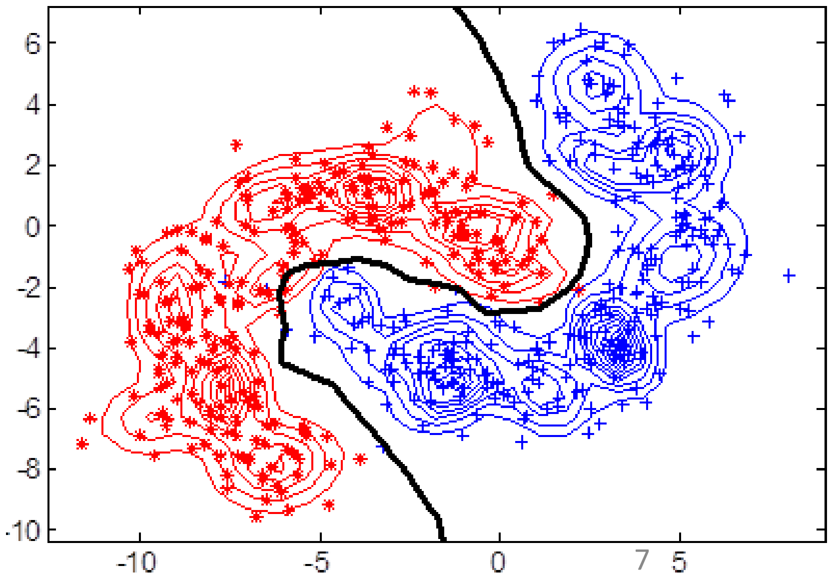
- Is there 'geometry' of clusters in the feature space?
- All successful Machine Learning algorithms look for this geometry:
  - Support Vector Machines, Random Forest, Gradient Boosting, etc.



Mathematical foundations of AI

- Is there ‘geometry’ of clusters in the feature space?
- All successful Machine Learning algorithms look for this geometry:
  - Support Vector Machines, Random Forest, Gradient Boosting, etc.
- **If not, can we transform to a better feature space through feature engineering/deep learning (RNN, CNN, etc)?**





Mathematical foundations of AI

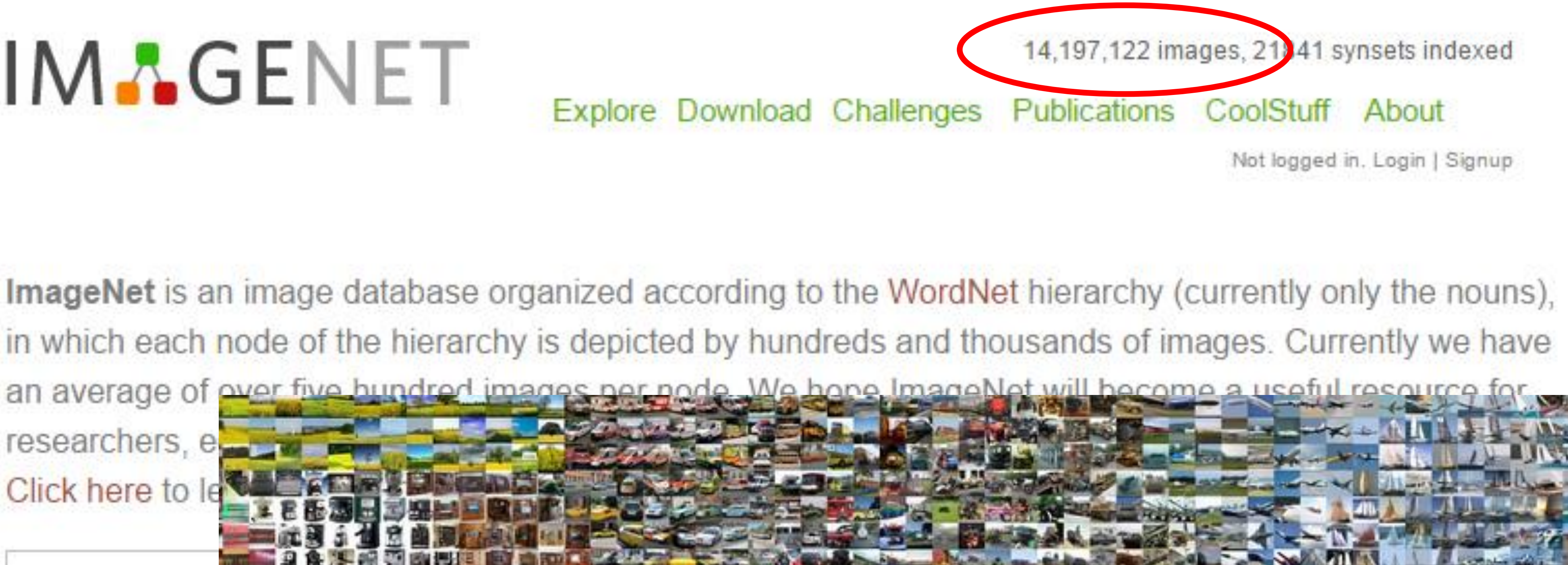
- Is there 'geometry' of clusters in the feature space?
- All successful Machine Learning algorithms look for this geometry:
  - Support Vector Machines, Random Forest, Gradient Boosting, etc.
- **If not, can we transform to a better feature space through feature engineering/deep learning (RNN, CNN, etc)?**

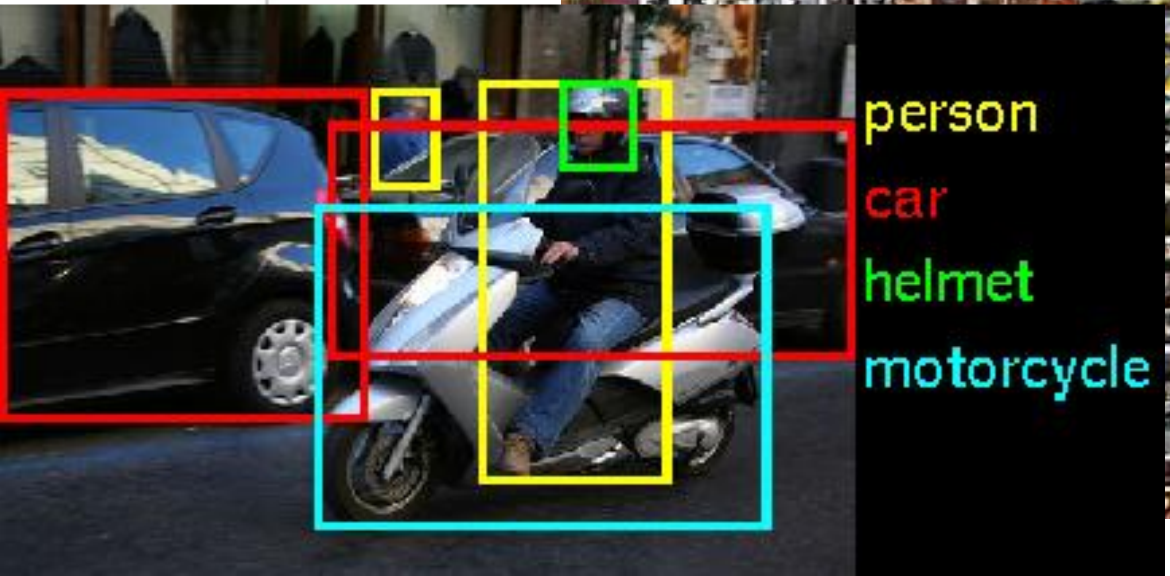
Our goal is to provide an holistic mathematical foundation for:

Signal processing, classical Ml and AI through:

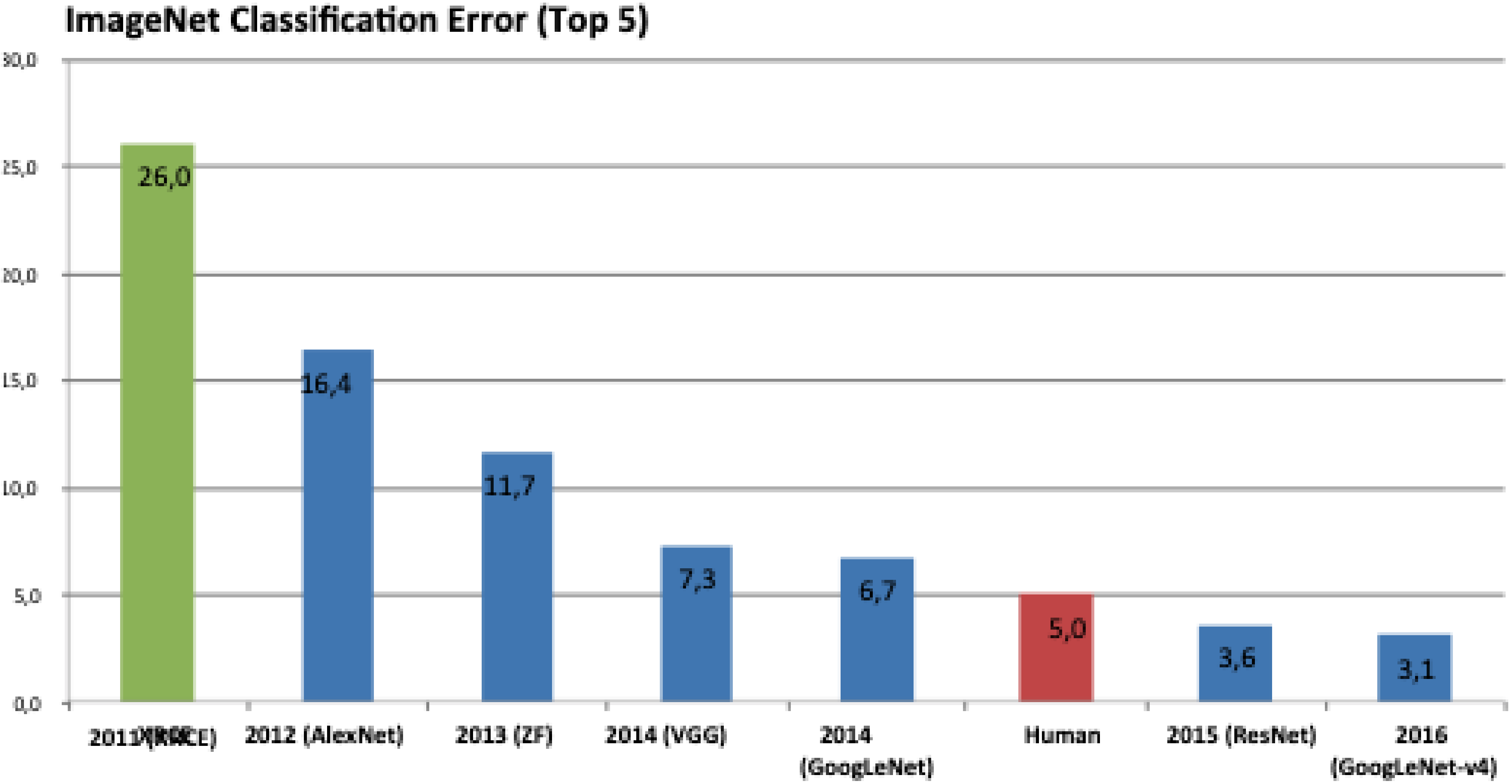
function space theory, Approximation Theory, Geometric Harmonic analysis

Crash course in Deep Learning


IMAGENET
14,197,122 images, 21841 synsets indexed
Explore Download Challenges Publications CoolStuff About
Not logged in. Login | Signup
ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns),
in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have
an average of
researchers, e
Click here to le

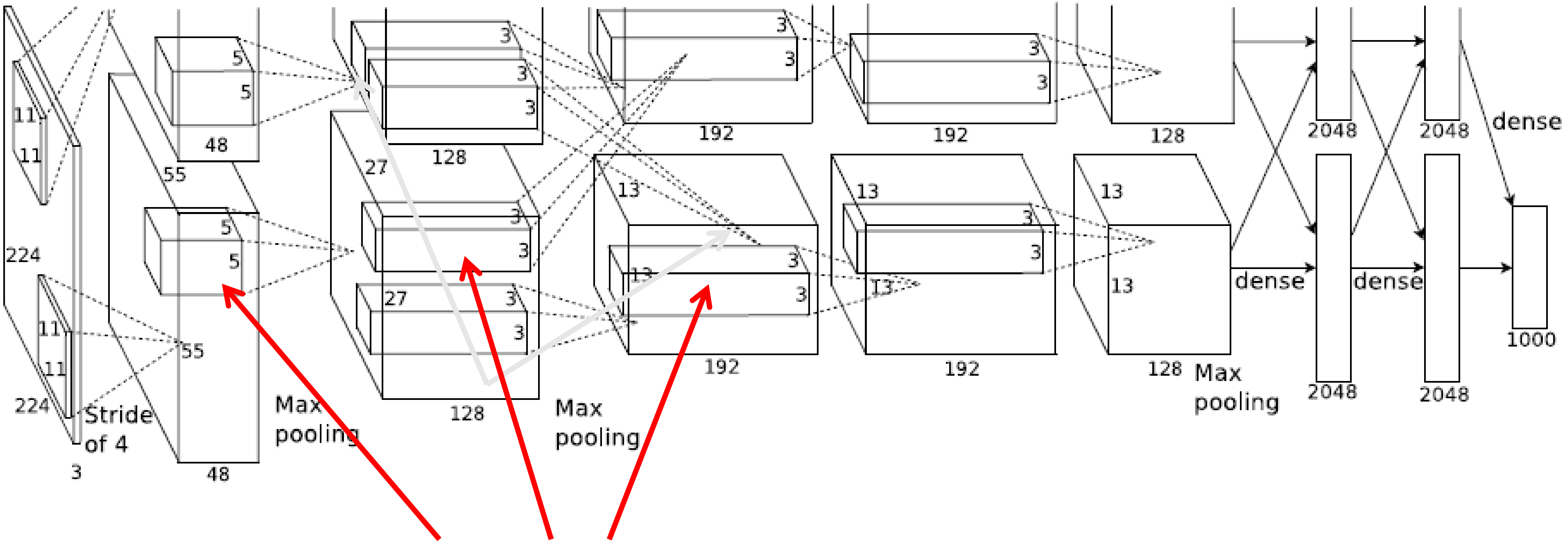

person
car
helmet
motorcycle

IMAGENET Large Scale Visual Recognition Challenge 2017 (ILSVRC2017)



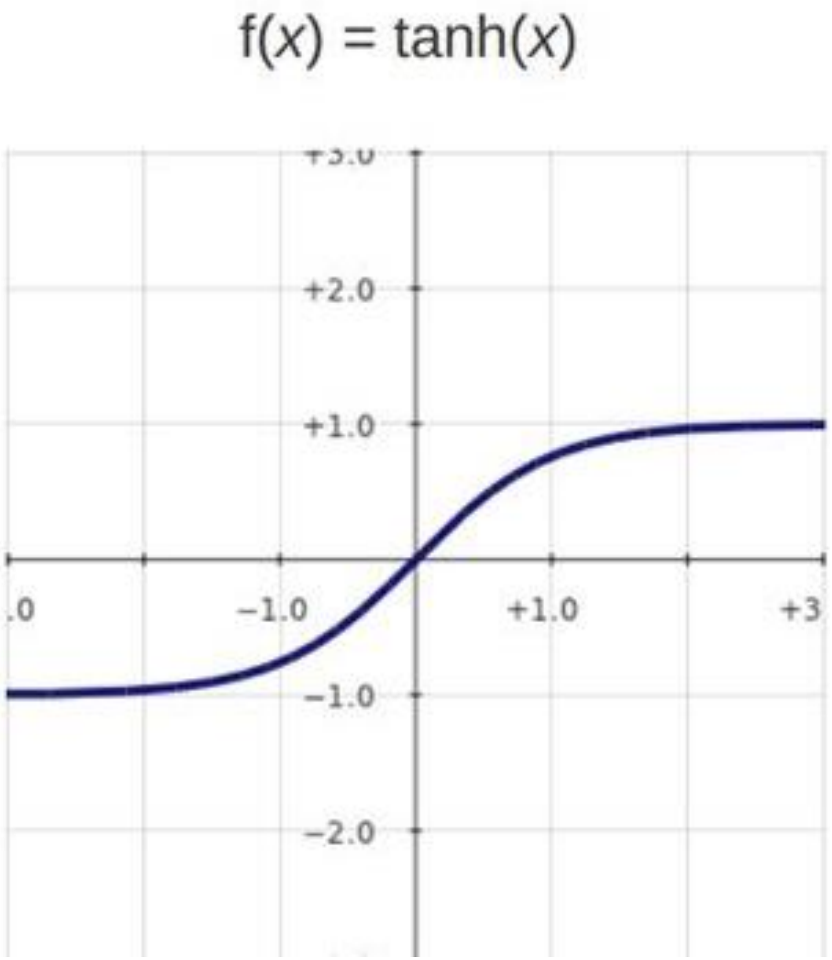
AlexNet (2012)

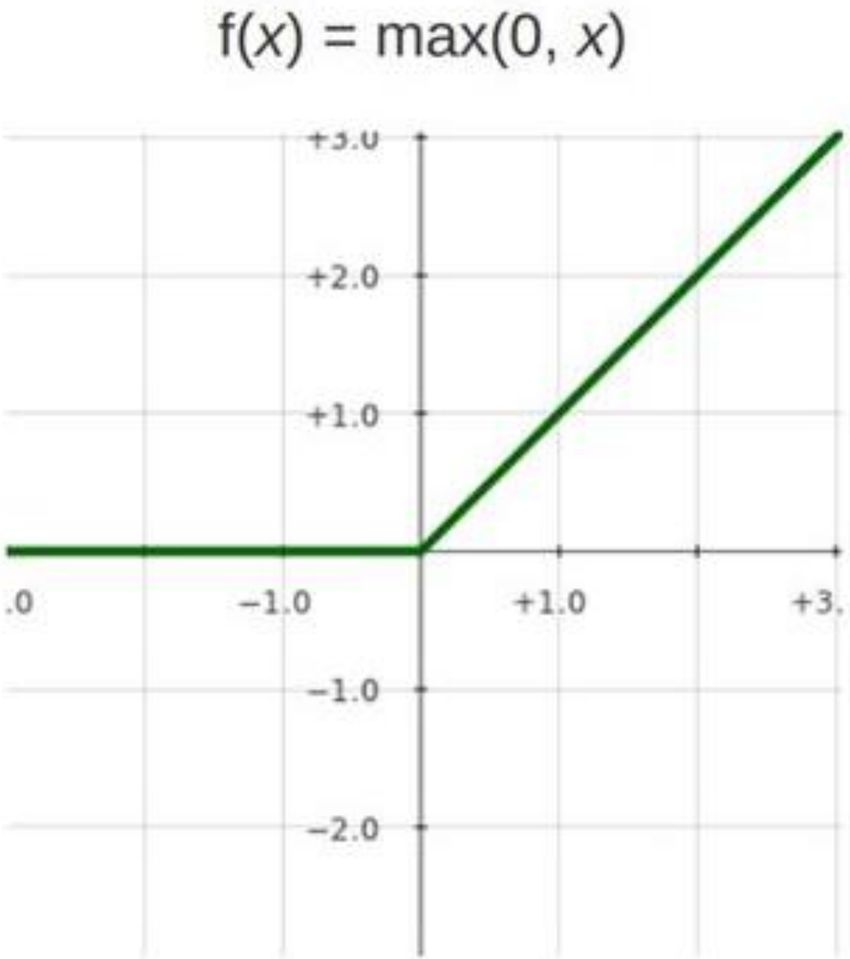
$$f * w(k) = \sum_{j \in \mathbb{Z}^n} f_j w_{k-j}$$



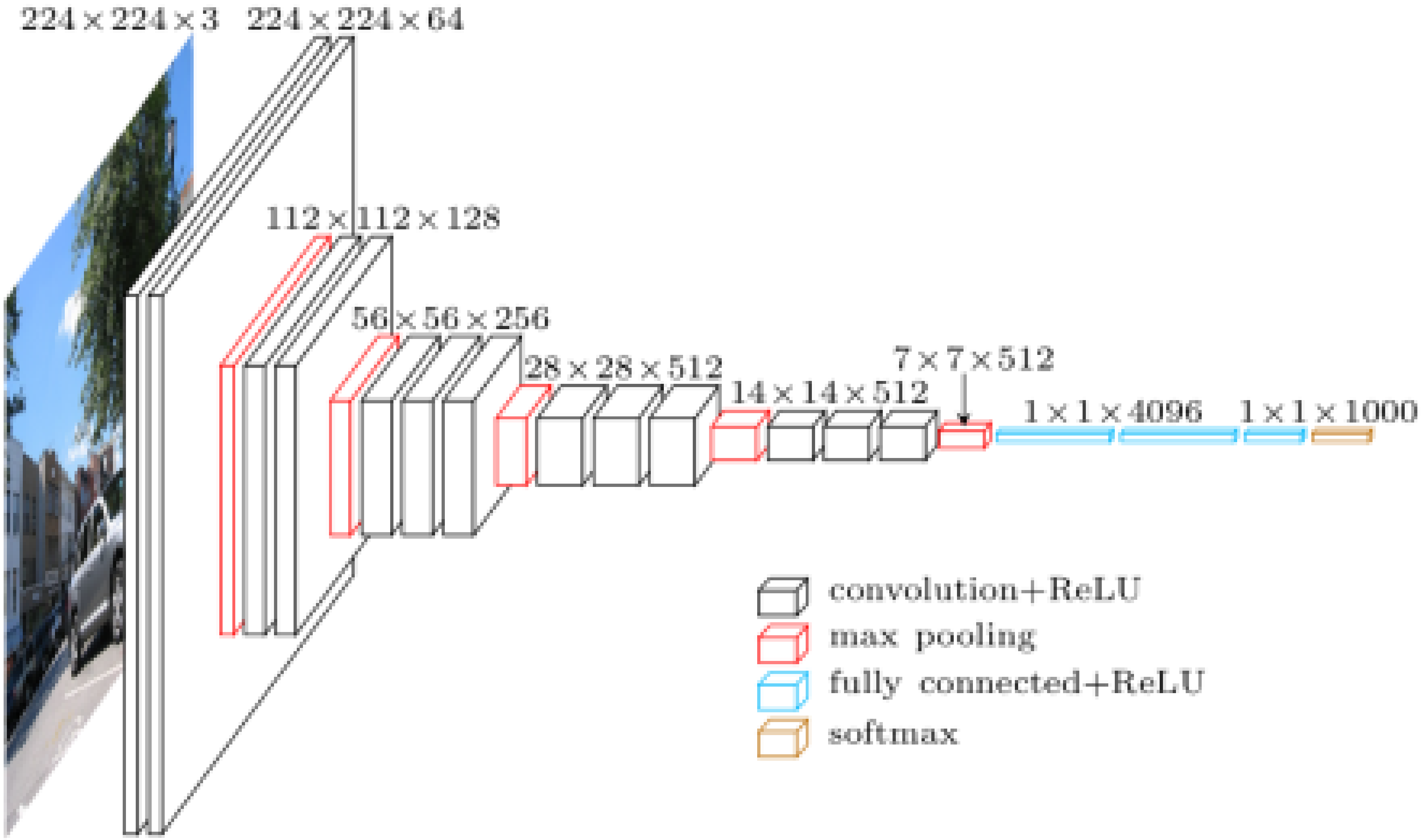
Convolutions = weight sharing → tractaible computation

Non-linearity





VGG Net (2015)


224 × 224 × 3
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU
softmax

Inception Blocks (2016)

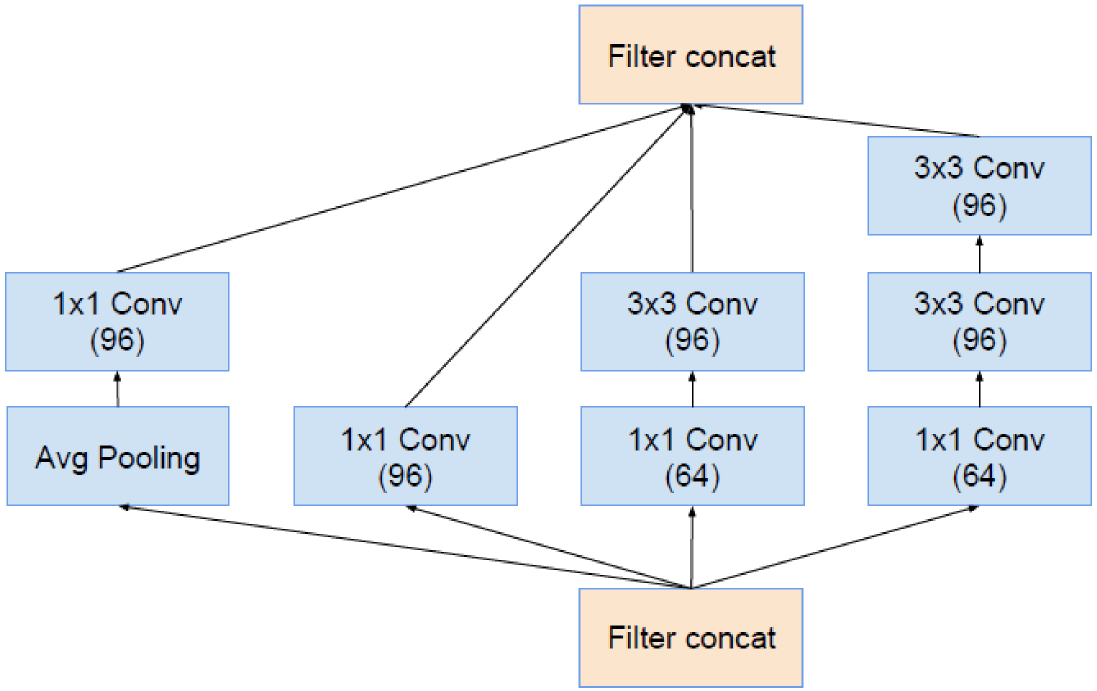


Figure 4. The schema for 35 × 35 grid modules of the pure Inception-v4 network. This is the Inception-A block of Figure 9.

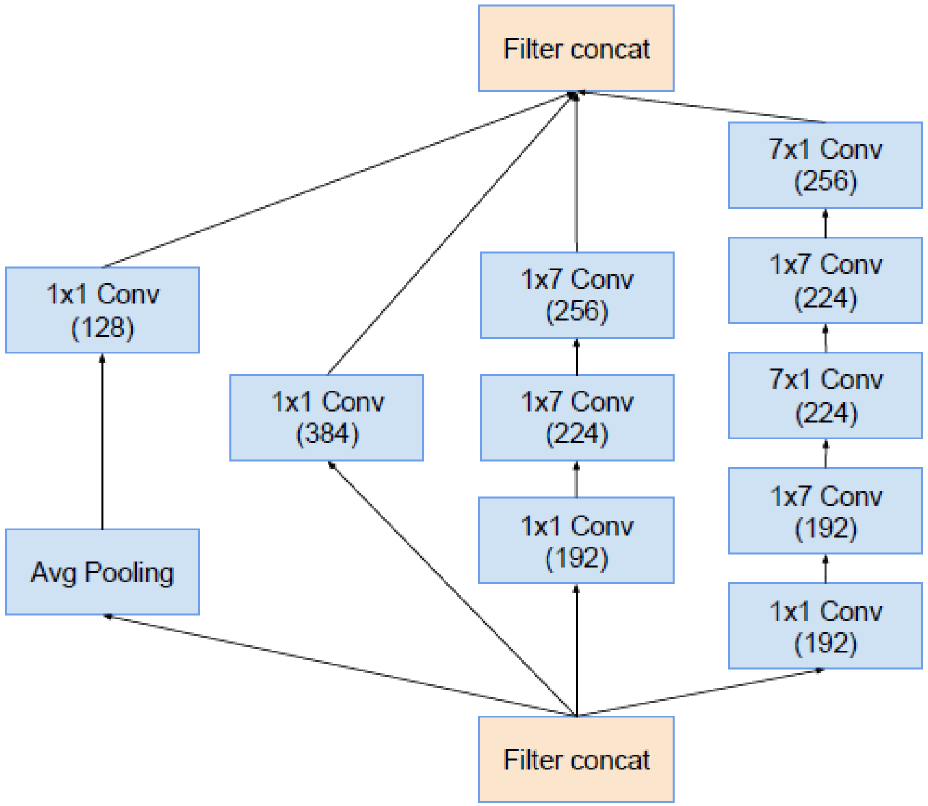
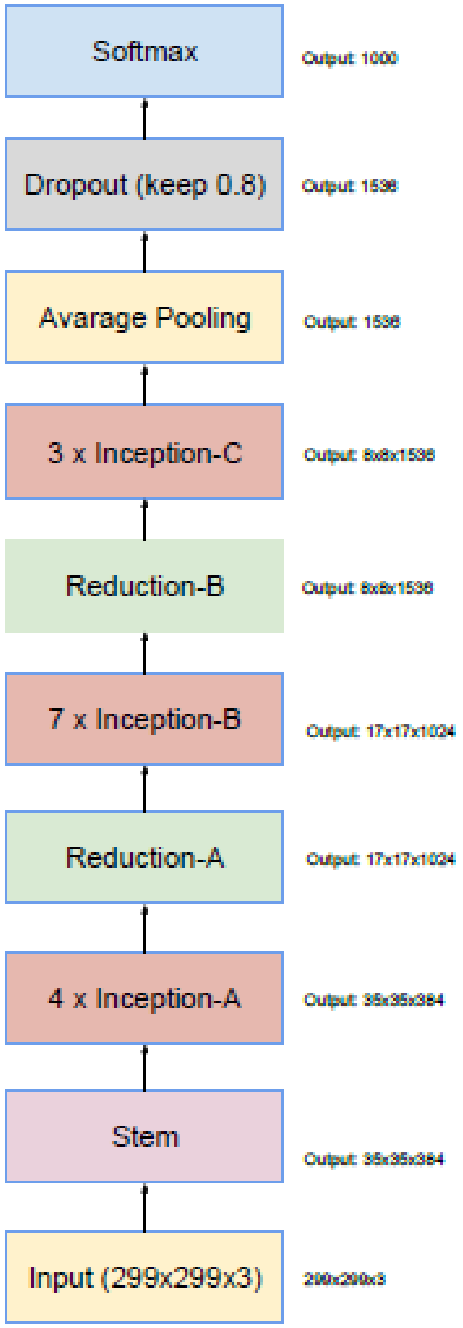


Figure 5. The schema for 17 × 17 grid modules of the pure Inception-v4 network. This is the Inception-B block of Figure 9.



Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning

Christian Szegedy
Google Inc.
1600 Amphitheatre Pkwy, Mountain View, CA
szegedy@google.com

Sergey Ioffe
sioffe@google.com

Vincent Vanhoucke
vanhoucke@google.com

Alex Alemi
alemi@google.com

Challenges!!!

- Deep learning architectures:
  - Initially created to mimic the human brain…

Challenges!!!

- Deep learning architectures:
  - Initially created to mimic the human brain…
  - But now… complex configurations created through trial and error, based on empiric results & intuition

Challenges!!!

- Deep learning architectures:
  - Initially created to mimic the human brain…
  - But now… complex configurations created through trial and error, based on empiric results & intuition
  - … making it all somewhat mystic ☺
  - Now… many new papers trying to demystify
- Why is it difficult?
  - Each hidden layer of a different structure, different dimension
  - Statistical methods have difficulty to capture the complexity
  - The representations are non continuous.
  - Difficult to obtain a unifying approach!!!

Function space representation: layer 0

- Assume we have a dataset of grayscale images of dimension $\sqrt{n_0} \times \sqrt{n_0}$
- We concatenate the pixel values to vectors of size n_0 .
- We normalize the pixels values to $[0,1]$.
- Each image is associated with one of L class labels.
- We map each label to a vertex of the standard simplex in $\mathbb{R}^{L-1}$
- Thus, each image is now a sample of a function

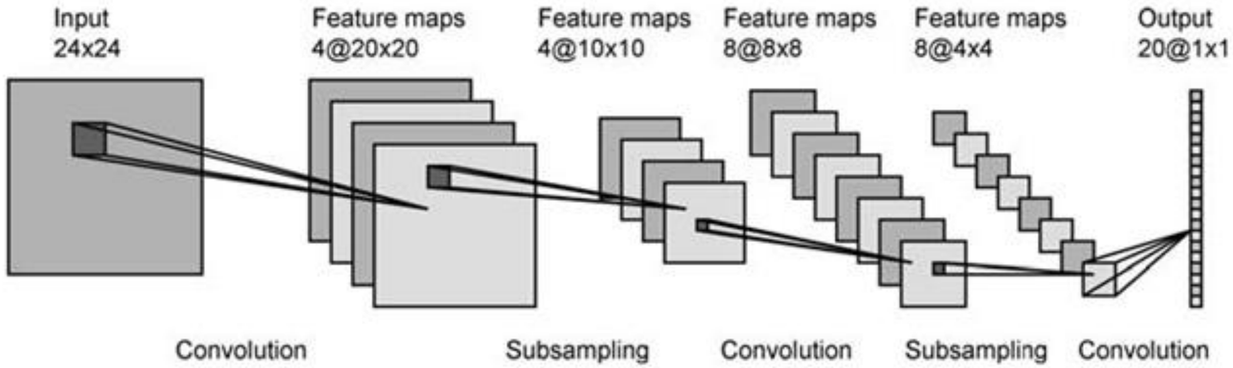
$$f_0 : [0,1]^{n_0} \to \mathbb{R}^{L-1}$$

In general this function will look like spaghetti…no clustering!

Function space representation: inner layers

- For each k-th inner layer consisting of n_k features/neurons we do something similar (during or after the training).
- We 'run' each image through the network until the k-th layer.
- We concatenate the features of the image into a vector of size n_k.
- The feature values are normalized to $[0,1]$.
- This implies we now have samples of a function

$$f_k : [0,1]^{n_k} \to \mathbb{R}^{L-1}.$$



Unfolding of the clusters

<u>Conjecture #1</u> For a trained well-performing DL network, the functions

$$f_k : [0,1]^{n_k} \to \mathbb{R}^{L-1}, \quad k = 0,\dots,K = \#Layers$$

are "better" behaved as we go deeper through the layers.

<u>Conjecture #2</u> The functions get "better" through the training iterations.

- But how do we quantify? The series of functions $\{f_k\}$:
  - Have their domains in very high and different dimensional spaces
  - Are discontinuous

CIFAR10: Unfolding of the clusters

Layer	Type	# Features
0	Input	576
1	Conv	9216
2	Conv	2304
3	Fully	384
4	Fully	192
5	Logits	10

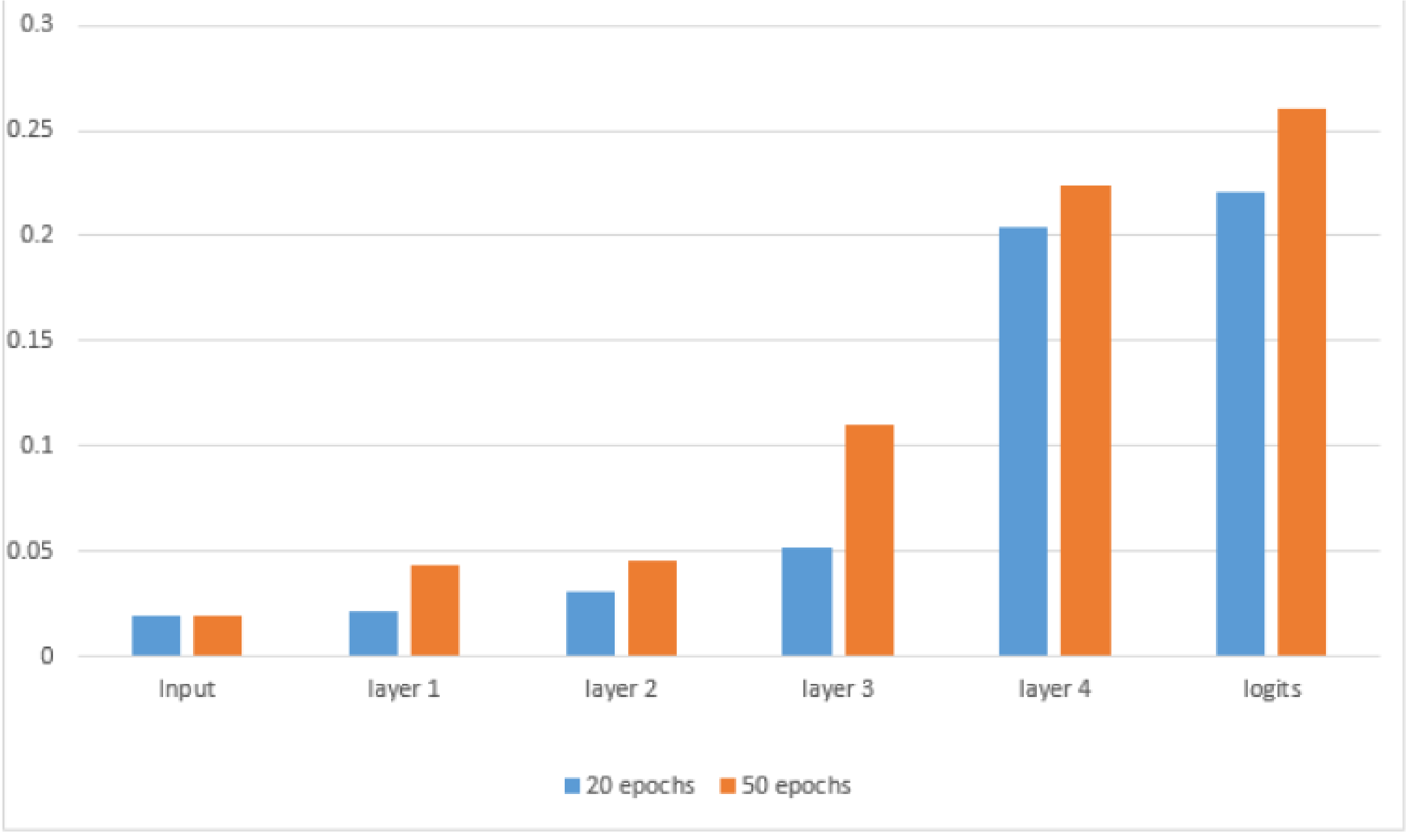


Fig. 4. Smoothness analysis of DL layers representations of CIFAR10

Generalization (understanding mis-labeled datasets)

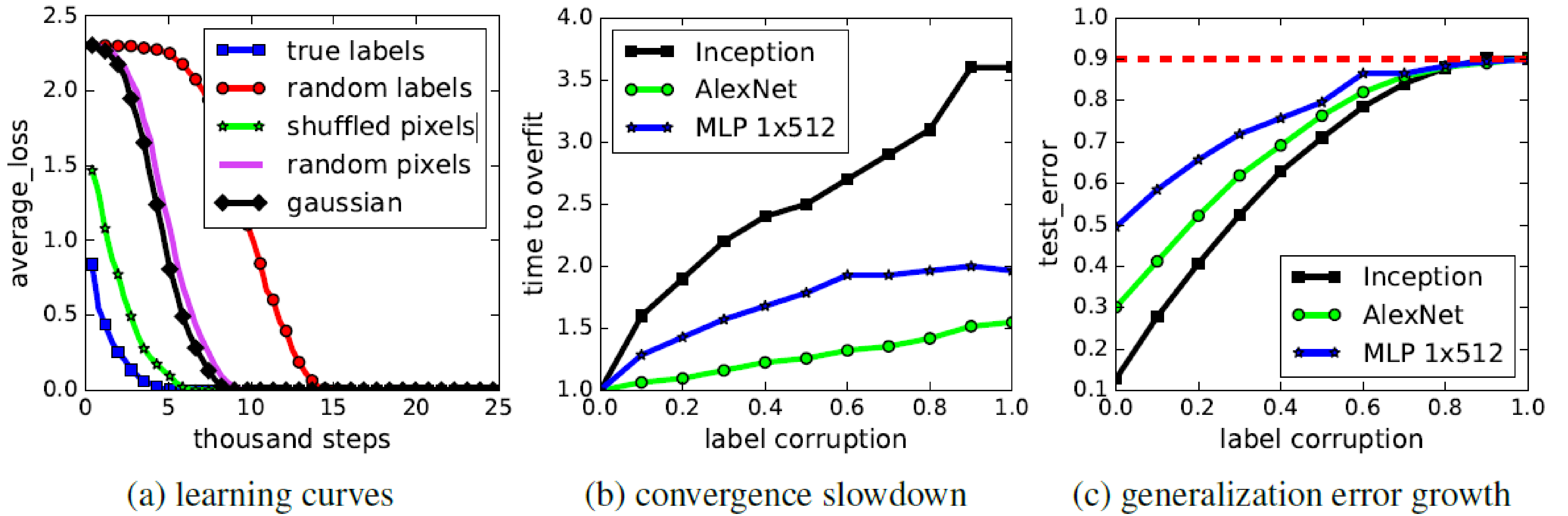
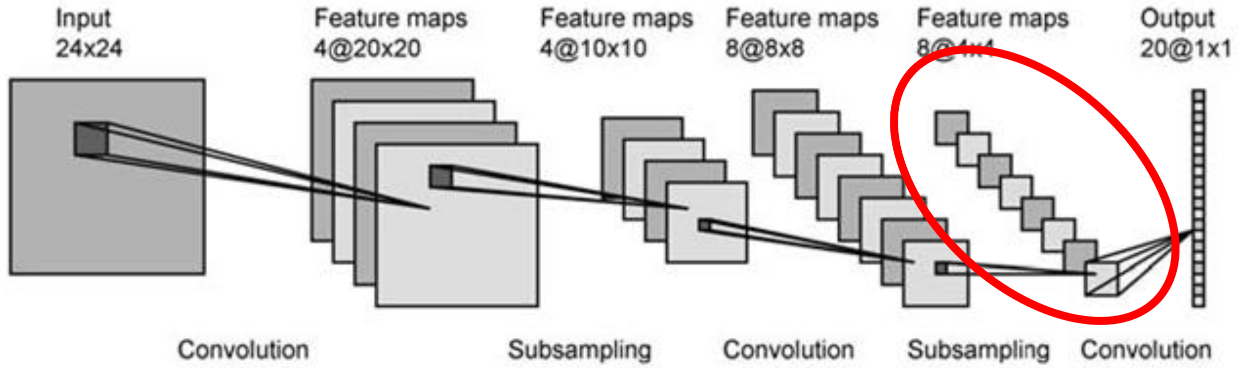


Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

***C. Zhang, S. Bengio, M. Hardt, B. Recht and O. Vinyals, Understanding deep learning requires rethinking generalization, ICLR 2017.**

Generalization (understanding mis-labeled datasets)

We measure the smoothness at the last inner layer f_{K-1}.



Mis-labeling	0%	10%	20%	30%	40%
MNIST smoothness	0.28	0.106	0.084	0.052	0.03
CIFAR10 smoothness	0.204	0.072	0.053	0.051	0.003

TABLE 1
Smoothness analysis of mis-labeled image images