ECE 595: Machine Learning I
Lecture 08 Hand-Crafted and Deep Features

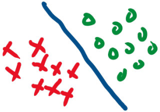
Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

Overview

Supervised Learning for Classification

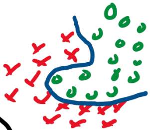


Linearly Separable?

yes

Linear Discriminant Analysis
- Bayesian decision
- logistic regression
- support vector machine
- deep neural network

no



Feature Analysis
- PCA
- hand-crafted features
- deep features

Lecture 7

Lecture 8

Outline

Feature Analysis

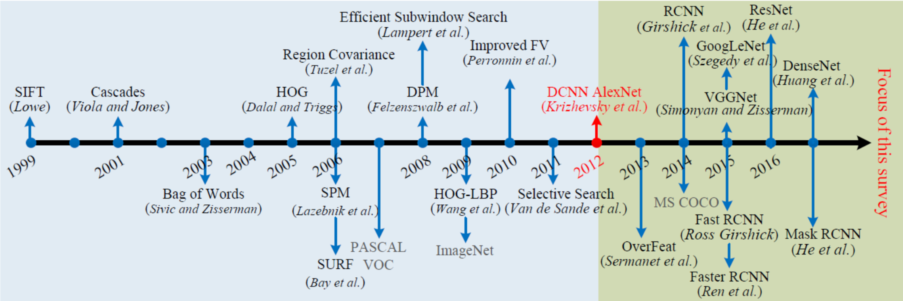
- Lecture 7 Principal Component Analysis (PCA)
- Lecture 8 Hand-Crafted and Deep Features

This Lecture

- Little History of Feature Extractions
- Convolution
  - What is convolution (if you don't know what it is yet)?
  - Some interesting facts about convolution
- SIFT and HOG
  - Gaussian derivatives
  - Pyramid
  - Histogram of oriented gradients
- Deep Features
  - What are they?
  - How to use them?

A Rough History of Feature Extraction

Deep Learning for Generic Object Detection: A Survey, `https://arxiv.org/pdf/1809.02165.pdf`



- PCA: Statistical analysis. Content agnostic.
- SIFT: Image specific. Non-training.
- Deep Features: Image specific. Training.

Convolution

Convolution (2D) is between two functions f and h:

- An **input** function $f(\boldsymbol{x})$, indexed by spatial coordinate $\boldsymbol{x} = [x_1,\ x_2]^T$
- A **filter** $h(\boldsymbol{x})$

The **output** is of the convolution is

$$\begin{aligned} g(\boldsymbol{x}) &= f(\boldsymbol{x}) * h(\boldsymbol{x}) \\ &= \int f(\boldsymbol{x} - \boldsymbol{\xi}) h(\boldsymbol{\xi}) d\boldsymbol{\xi} \end{aligned}$$

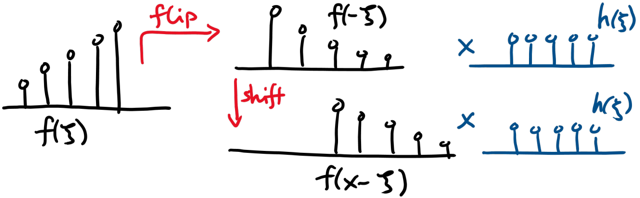
Do not be confused with **correlation**:

$$\begin{aligned} g(\boldsymbol{x}) &= f(\boldsymbol{x}) \circledast h(\boldsymbol{x}) \\ &= \int f(\boldsymbol{x} + \boldsymbol{\xi}) h(\boldsymbol{\xi}) d\boldsymbol{\xi} \end{aligned}$$

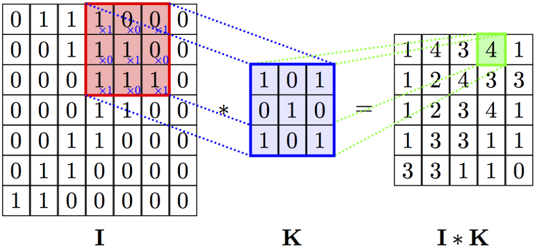
Convolution **flips** the filter, whereas correlation does not.

Pictorial Illustration

A convolution operation always involves 3 steps: flip-shift-add.



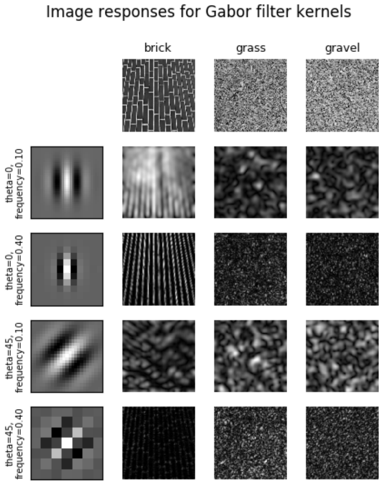
Most tutorials you see on the internet are correlations.



A Few Things You Need to Know about Convolution

- The core of convolution is the concept of **linear shift invariant** (LSI).
- A system $\mathcal{T}$ is LSI if
  - $\mathcal{T}(af_1(\boldsymbol{x}) + bf_2(\boldsymbol{x})) = a\mathcal{T}(f_1(\boldsymbol{x})) + b\mathcal{T}(f_2(\boldsymbol{x}))$
  - Let $g(\boldsymbol{x}) = \mathcal{T}(f(\boldsymbol{x}))$. Then for any $\boldsymbol{\xi}$, $f(\boldsymbol{x} + \boldsymbol{\xi}) \mapsto g(\boldsymbol{x} + \boldsymbol{\xi})$.
- Convolution is the only operation that allows LSI.
- **Eigen-functions** of a convolution operation are the Fourier series.
- The flip operation is necessary to define the Fourier series.
- This can be dated back to Pierre-Simon Laplace (1749-1827) and Joseph Fourier (1768-1830), with 200 years of work in real / functional analysis.
- Convolution with large filters are always implemented by **Fast Fourier Transforms**.
- Convolution can be performed at the **speed of light**! Put a mask at the focal plane of the lens. It will give you the convolution of the mask and the image (in the Fourier domain). `https://www.youtube.com/watch?v=4Eg0Tbk601s`

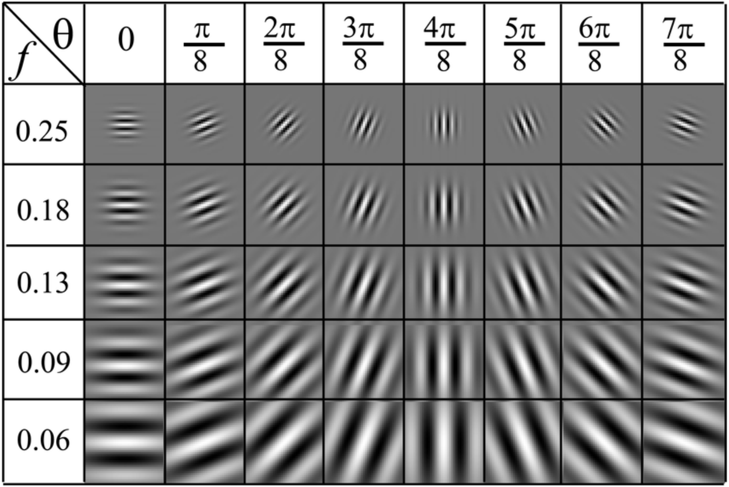
Effect of Convolution / Correlation



https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_gabor.html

Examples of Filters

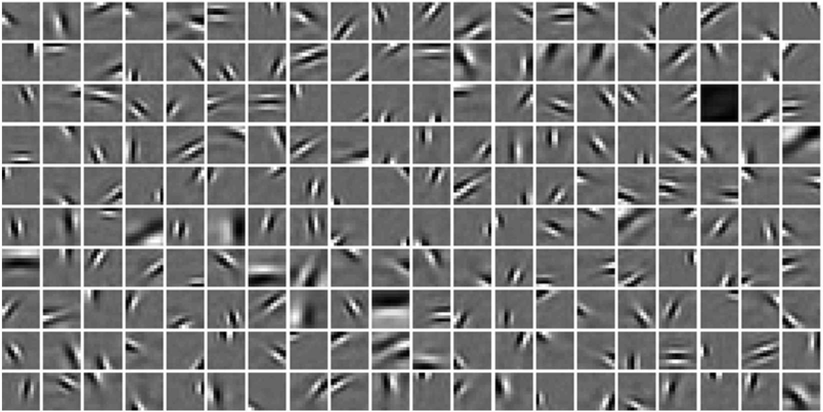
Gabor Filter



https://www.researchgate.net/figure/Real-parts-of-the-Gabor-filter-bank-Generated-for-different-combinations-of-th-in_fig5_292671765

Examples of Filters

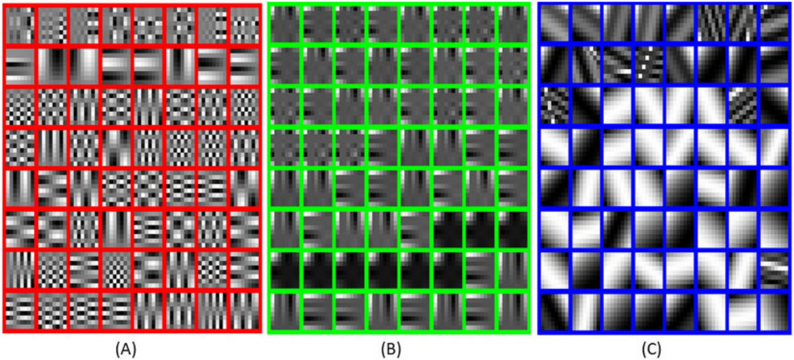
Another Gabor Filter



https://www.quora.com/How-are-Gabor-filters-implemented-in-visual-area-V1-in-the-brain

Examples of Filters

KSVD Filters



https://www.researchgate.net/figure/Basis-functions-used-by-a-KSVD-The-KSVD-based-dictionary-elements-or-atoms-mostly_fig6_336133323

Outline

Feature Analysis

- Lecture 7 Principal Component Analysis (PCA)
- Lecture 8 Hand-Crafted and Deep Features

This Lecture

- Little History of Feature Extractions
- Convolution
  - What is convolution (if you don't know what it is yet)?
  - Some interesting facts about convolution
- SIFT and HOG
  - Gaussian derivatives
  - Pyramid
  - Histogram of oriented gradients
- Deep Features
  - What are they?
  - How to use them?

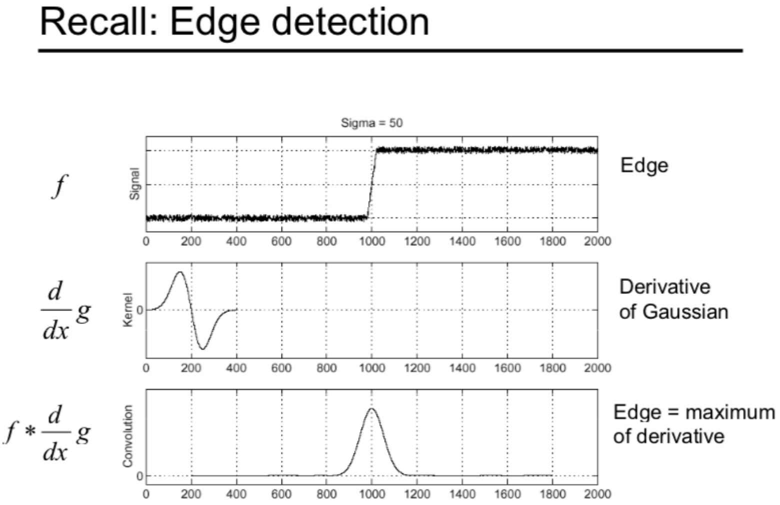
SIFT

- SIFT = Scale-invariant feature transform.
- Proposed by David Lowe in 1999.
- Idea: Convolve the image with the **2nd order derivative of a Gaussian**.
- Vary the radius of the Gaussian. Locate the radius that maximizes the response.
- What makes SIFT so powerful? The derivative of Gaussian filter extracts the scale:

$$G(\boldsymbol{x}, k\sigma) - G(\boldsymbol{x}, \sigma) \approx (k-1)\sigma^2 \nabla^2_{\boldsymbol{x}} G,$$

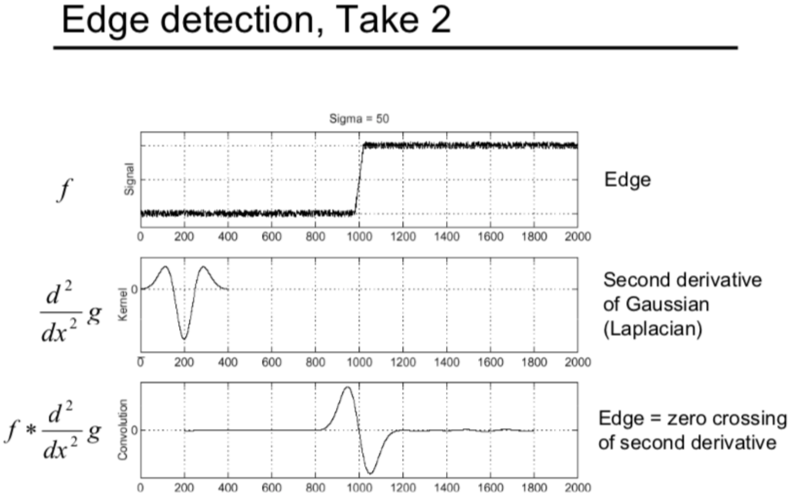
  where $G(\boldsymbol{x}, \sigma) = (1/(2\pi\sigma^2)) \exp\{-\|\boldsymbol{x}\|^2/(2\sigma^2)\}$.
- Output of SIFT: A set of **locations** where there are "**blobs**".
- Can be used for registering images.

Gaussian Filter for Edge Detection



https://towardsdatascience.com/sift-scale-invariant-feature-transform-c7233dc60f37

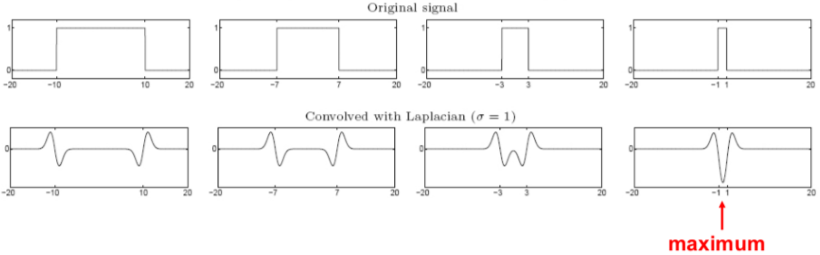
2nd Order Gaussian Derivative



https://towardsdatascience.com/sift-scale-invariant-feature-transform-c7233dc60f37

Locating Blobs

- Edge = ripple
- Blob = superposition of two ripples



Spatial selection: the magnitude of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is "matched" to the scale of the blob

https://towardsdatascience.com/sift-scale-invariant-feature-transform-c7233dc60f37

Scale Pyramid

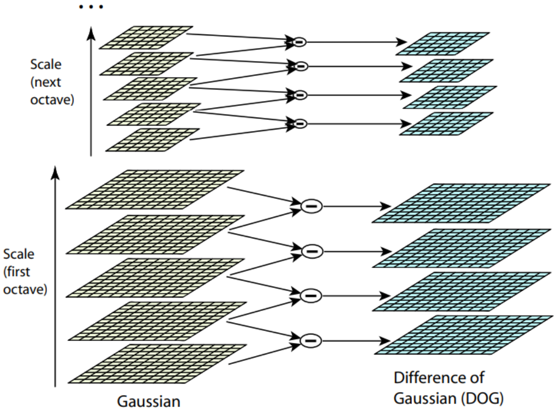
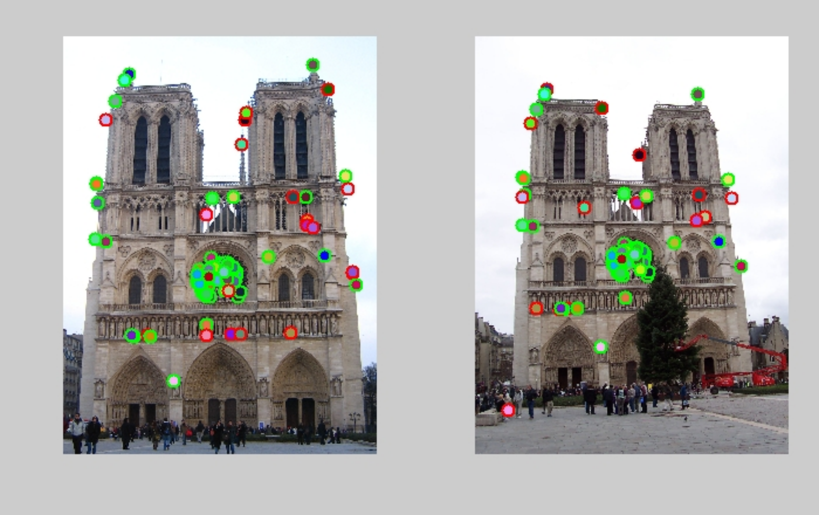


Figure 1: For each octave of scale space, the initial image is repeatedly convolved with Gaussians to produce the set of scale space images shown on the left. Adjacent Gaussian images are subtracted to produce the difference-of-Gaussian images on the right. After each octave, the Gaussian image is down-sampled by a factor of 2, and the process repeated.

https://people.eecs.berkeley.edu/~malik/cs294/lowe-ijcv04.pdf

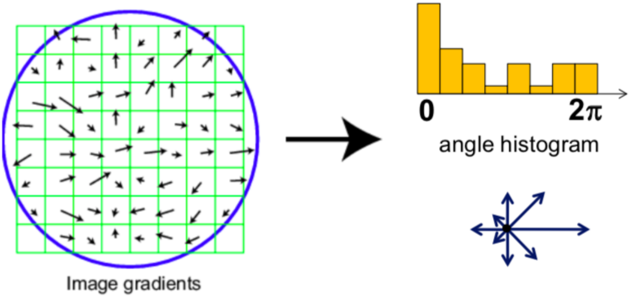
Example



http://cs.brown.edu/courses/cs143/2013/results/proj2/rroelke/

SIFT + HOG

- HOG = Histogram of Oriented Gradient
- Used to "encode" the detected blobs
  - Take 16x16 square window around detected feature
  - Compute edge orientation (angle of the gradient - 90°) for each pixel
  - Throw out weak edges (threshold gradient magnitude)
  - Create histogram of surviving edge orientations



https://towardsdatascience.com/sift-scale-invariant-feature-transform-c7233dc60f37

Outline

Feature Analysis

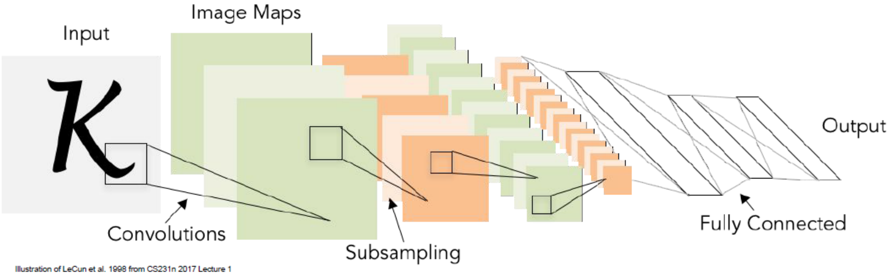
- Lecture 7 Principal Component Analysis (PCA)
- Lecture 8 Hand-Crafted and Deep Features

This Lecture

- Little History of Feature Extractions
- Convolution
  - What is convolution (if you don't know what it is yet)?
  - Some interesting facts about convolution
- SIFT and HOG
  - Gaussian derivatives
  - Pyramid
  - Histogram of oriented gradients
- Deep Features
  - What are they?
  - How to use them?

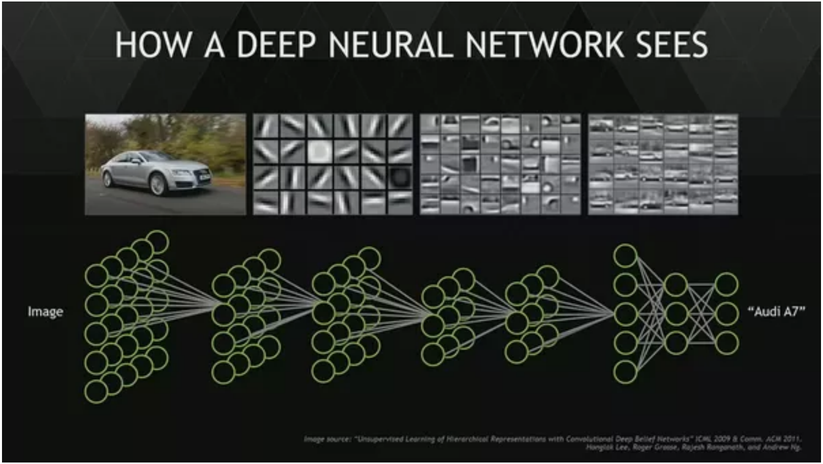
Deep Features

- Create a convolutional neural network (See Deep Learning courses).
- Investigate the features extracted at different stages of the network.



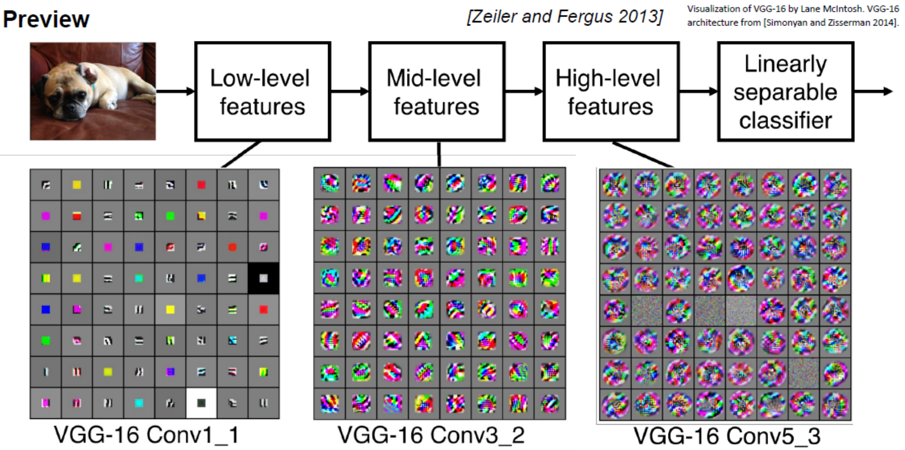
Source: Stanford CS 231n Lecture Note

Deep Features



https://www.analyticsvidhya.com/blog/2018/03/essentials-of-deep-learning-visualizing-convolutional-neural-networks/cnn_filters/

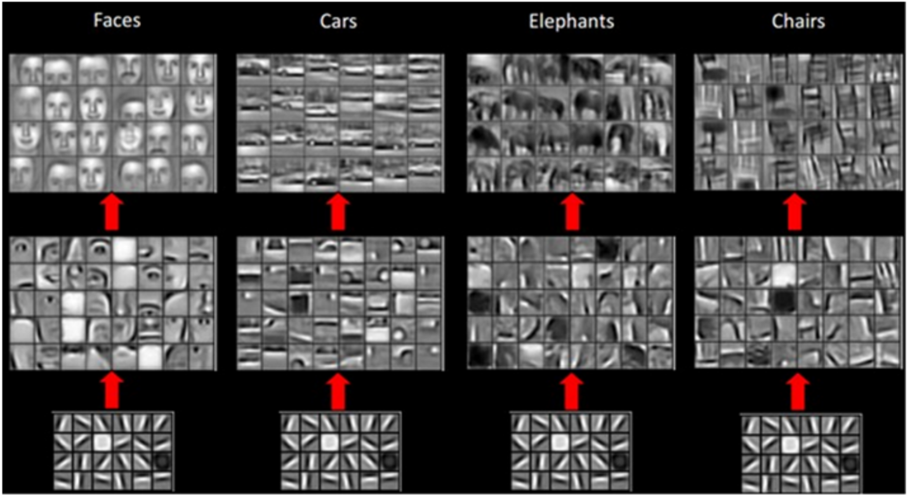
Hierarchical Representations



Source: Stanford CS 231n Lecture Note

Deep Features

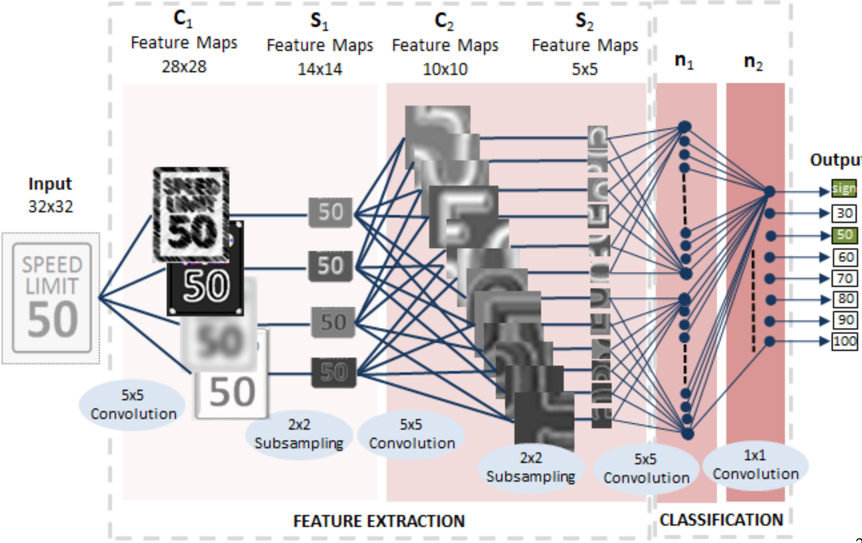
- Going through the layers, the network learns different representations



https://stats.stackexchange.com/questions/146413/why-convolutional-neural-networks-belong-to-deep-learning

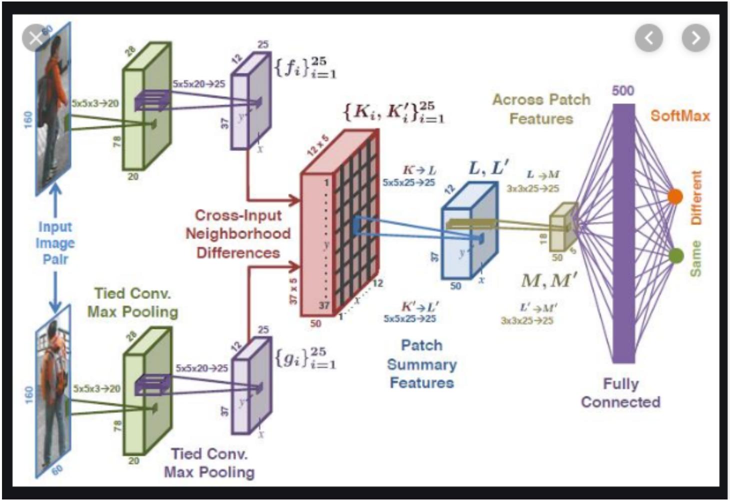
Hierarchical Representations

https://e2e.ti.com/blogs_/b/behind_the_wheel/archive/2018/02/08/ai-in-automotive-practical-deep-learning



Combining Features

http://www.fubin.org/research/Person_ReID/Person_ReID.html

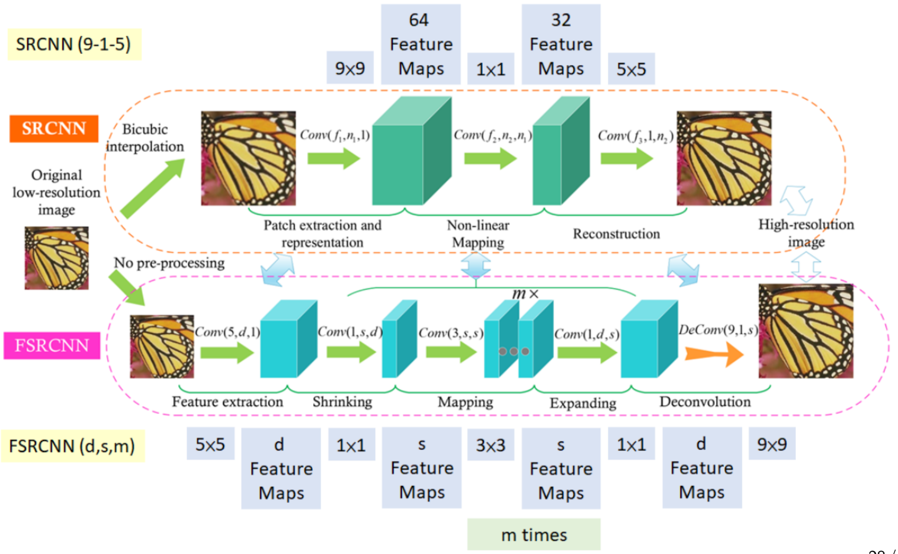


Using Deep Features for kNN

https://github.com/cvjena/semantic-embeddings

Using Deep Features for Super-Resolution

https://towardsdatascience.com/review-fsrcnn-super-resolution-80ca2ee14da4



Reading List

Convolution

- Oppenheim and Willsky, Signals and Systems, Chapter 2.
- ECE 637 Image Processing 1
  https://engineering.purdue.edu/~bouman/ece637/notes/

SIFT and HOG

- ECE 661 Computer Vision https://engineering.purdue.edu/kak/computervision/ECE661Folder/Lecture9.pdf
- Lowe's original SIFT paper https://people.eecs.berkeley.edu/~malik/cs294/lowe-ijcv04.pdf
- Blog on HOG https://www.learnopencv.com/histogram-of-oriented-gradients/

Deep Features

- Stanford CS 231n http://cs231n.stanford.edu/