Lecture Notes for
Machine Learning in Python

Professor Eric Larson

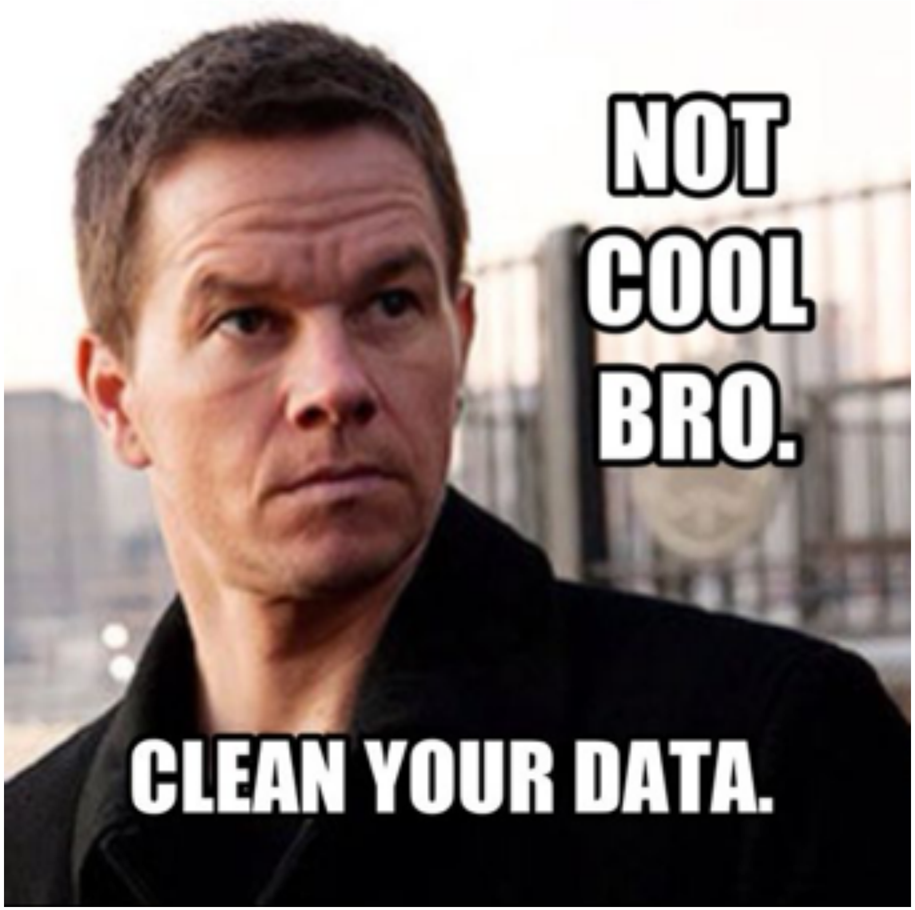
Week Three, Lecture B

Class Logistics and Agenda

- Next Week: Project Work Week
  - and I am out of town all week…
  - lecture after that: ICA1
- Agenda
  - Finish Dimensionality Reduction
  - Common Feature Extraction Methods for Images

synchronous

Dimensionality Reduction (Continued)

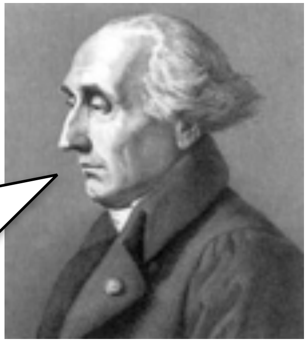


Dimensionality Reduction: LDA

- PCA tell us variance explained by the data in different directions, but it ignores class labels
- Is there a way to find “components” that will help with **discriminate** between the classes?

$$\arg\max_{comp.} \frac{\sum \text{differences between classes}}{\sum \text{variance within classes}}$$

- called Fisher’s discriminant
- …but we need to solve this using using *Lagrange multipliers* and gradient-based optimization
- which we haven’t covered yet



I invented Lagrange multipliers… and …*nothing impresses me…*

covered in stats

Dimensionality Reduction: LDA versus QDA

$$\underset{comp.}{\arg\max} \frac{\sum \text{differences between classes}}{\sum \text{variance within classes}}$$

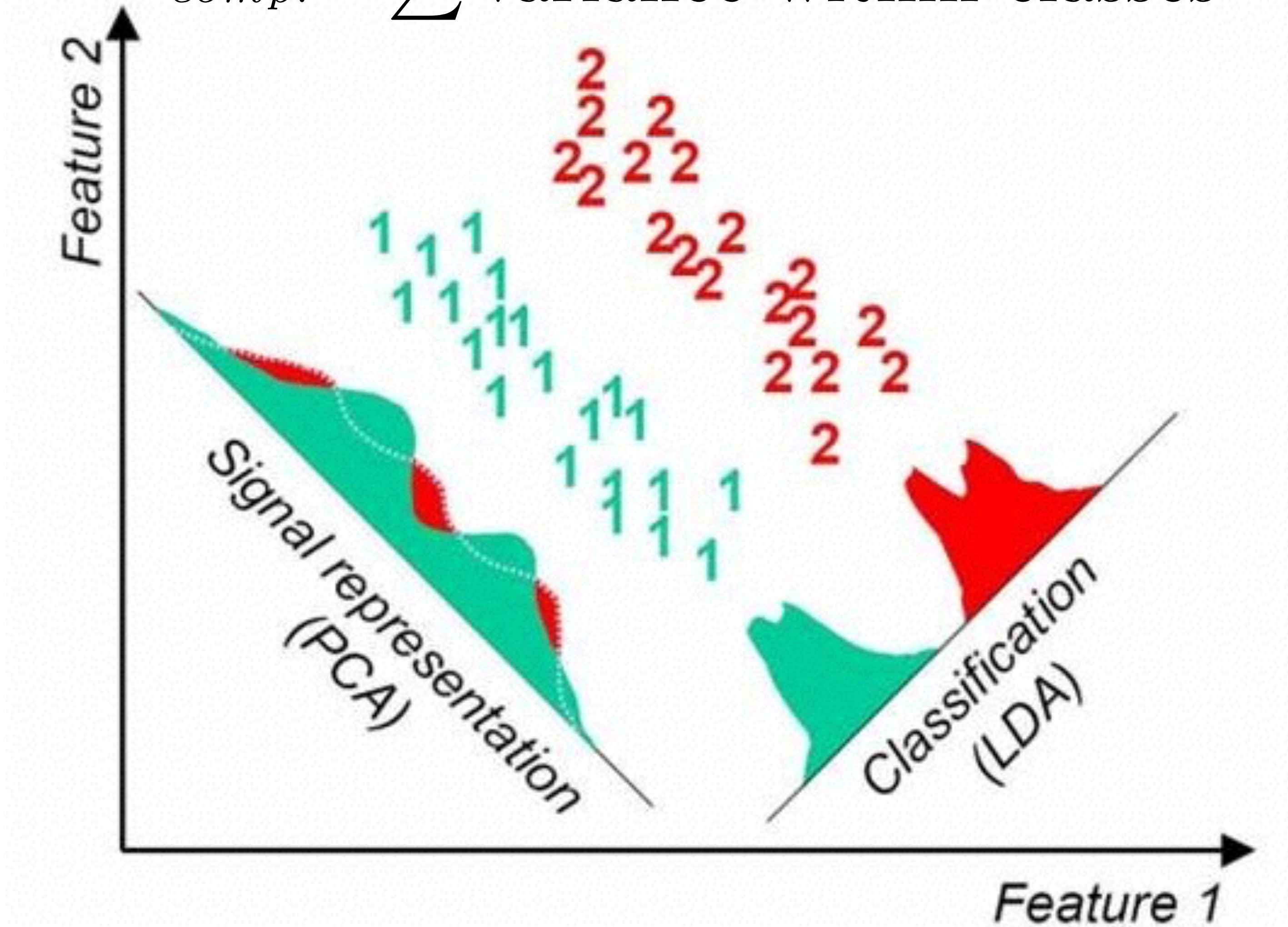


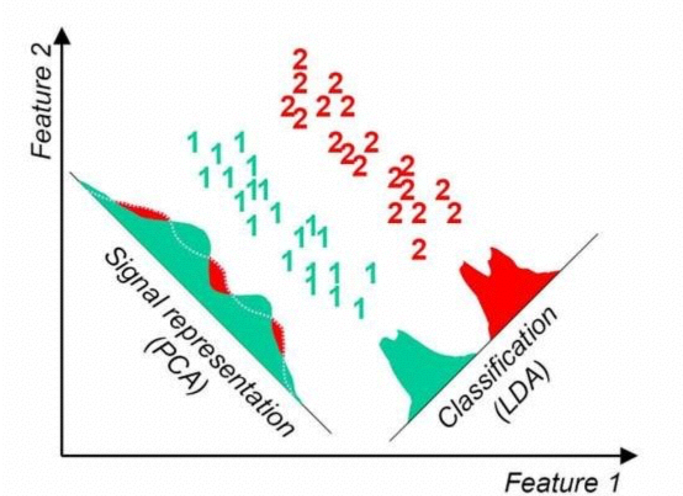
image source: 不想飞翔，不是因为没有翅膀，而是失去了梦想

covered in stats

Dimensionality Reduction: LDA versus QDA

$$\underset{comp.}{\arg\max} \frac{\sum \text{differences between classes}}{\sum \text{variance within classes}}$$

- "*differences between classes*" is calculated by trying to separate the **mean value** of each **feature** in each **class**
- Linear discriminant analysis:
  - assume the covariance in each class is the same
- Quadrature discriminant analysis:
  - estimate the covariance for each class



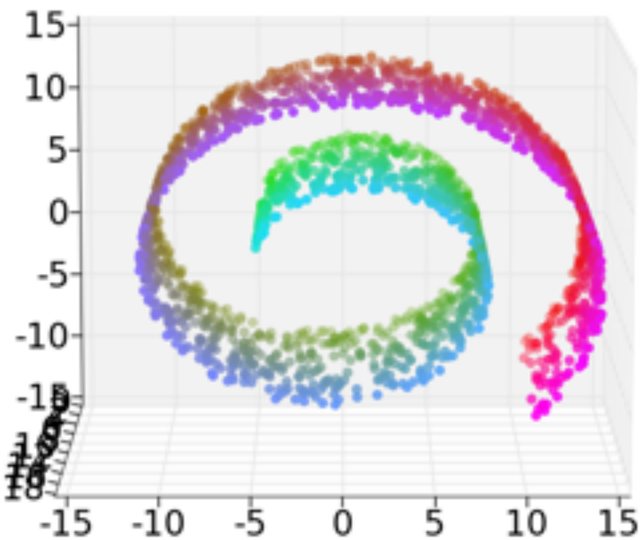
covered in stats

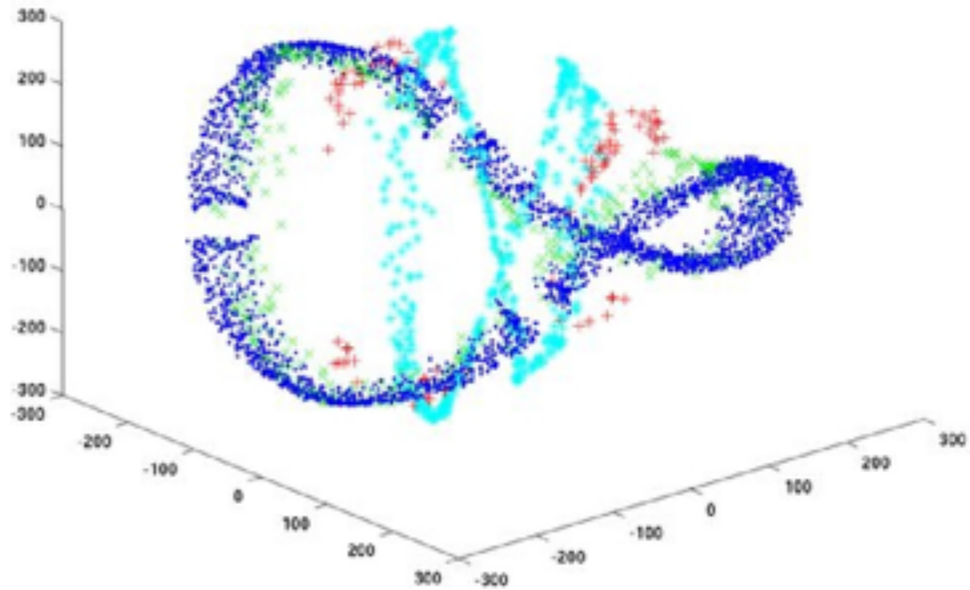
Self Test ML2b.2

LDA only allows as many components as there are unique classes in a dataset.

A. True
B. False

Dimensionality Reduction: non-linear

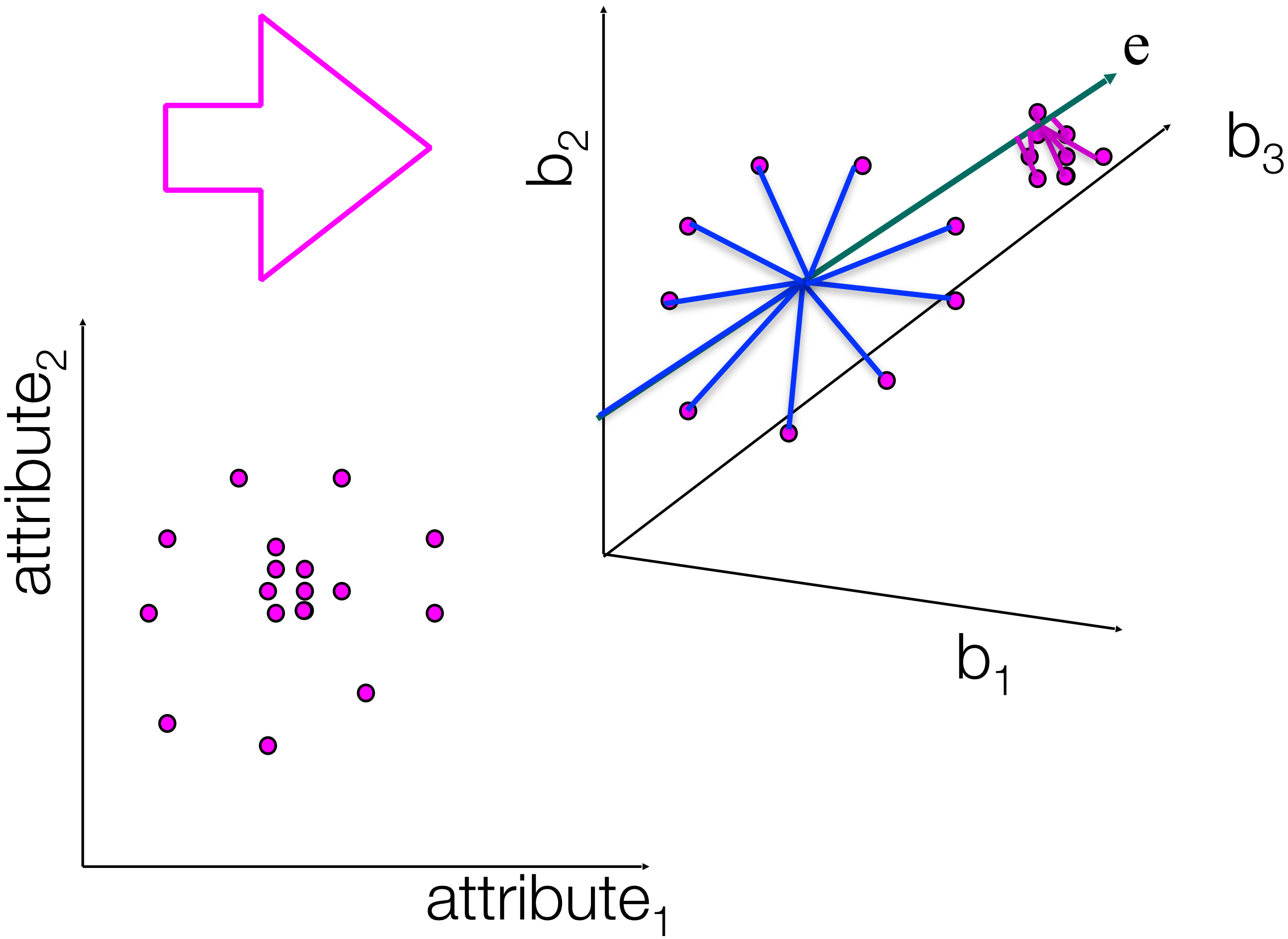




- Sometimes a **linear transform** is not enough
- A powerful non-linear transform has seen a resurgence in past decade: **kernel PCA**

Kernel PCA

- Project to higher dimensional space
- Employ principal components
- Apply transform in higher dimensional space



37.1	-6.7	-3.2
-6.7	43.9	1.45
-3.2	1.45	12.1

	B1	B2	B3
1	66	33.6	0.3
2	54	26.6	0.4
3	69	23.3	-4
4	73	28.1	-5.6
5	61	43.1	0.23
6	62	25.6	-5

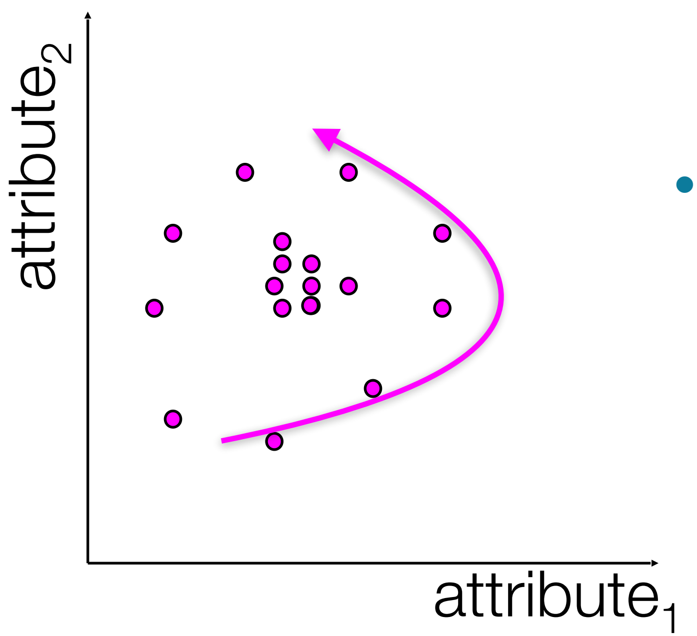
Kernel PCA

kernel: defines what the dot product is in higher dimensional space

some kernels have corresponding transformations with **infinite dimensions**!!

- **Just the dot product**

37.1	-6.7	-3.2
-6.7	43.9	1.45
-3.2	1.45	12.1

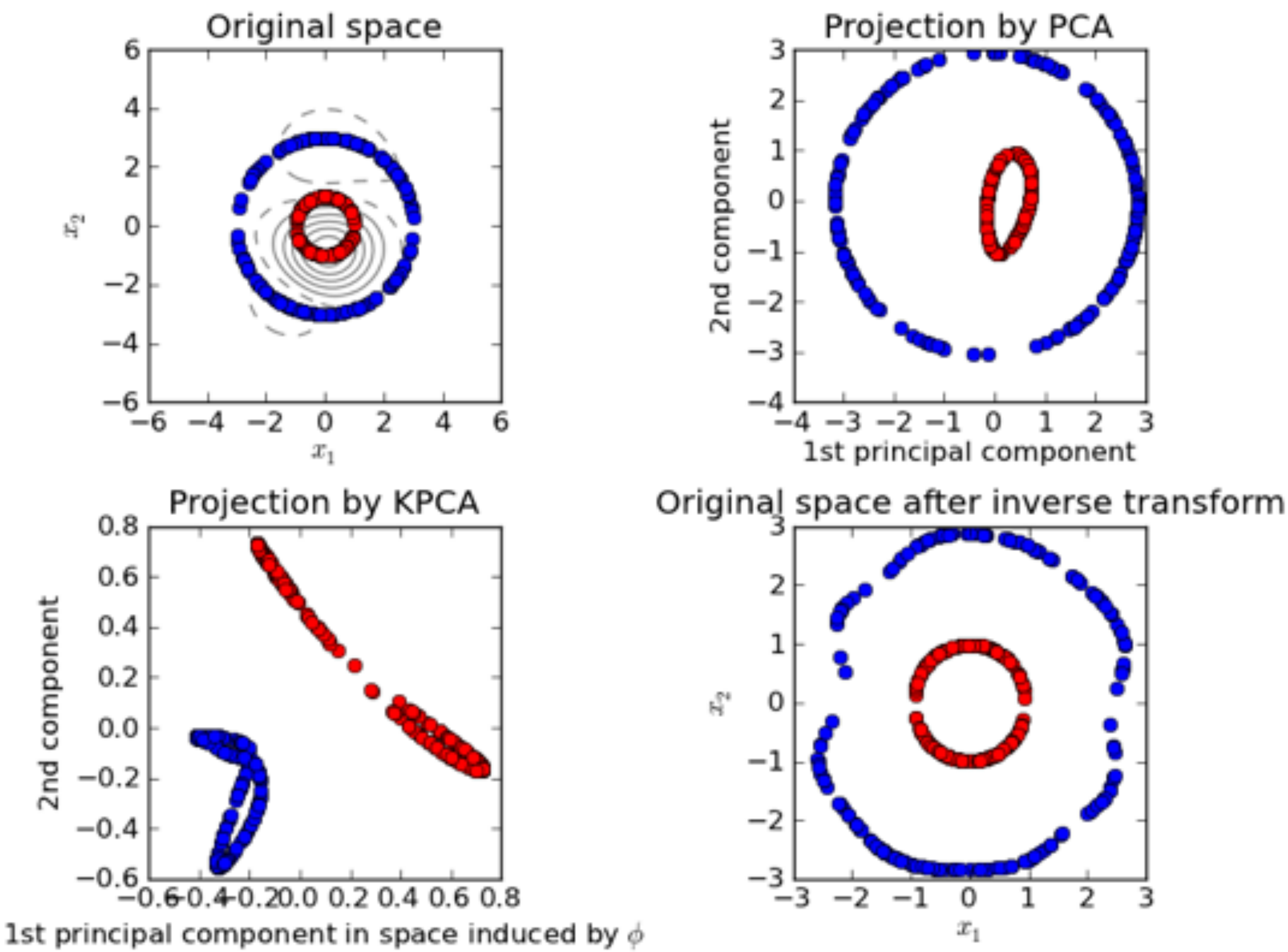


- **Key insight**: don't need to know the actual transformation vectors

	B1	B2	B3
1	66	33.6	0.3
2	54	26.6	0.4
3	69	23.3	-4
4	73	28.1	-5.6
5	61	43.1	0.23
6	62	25.6	-5

section 1: moved down

Kernel PCA



section 1: moved down

Demo

Dimension Reduction

PCA

LDA

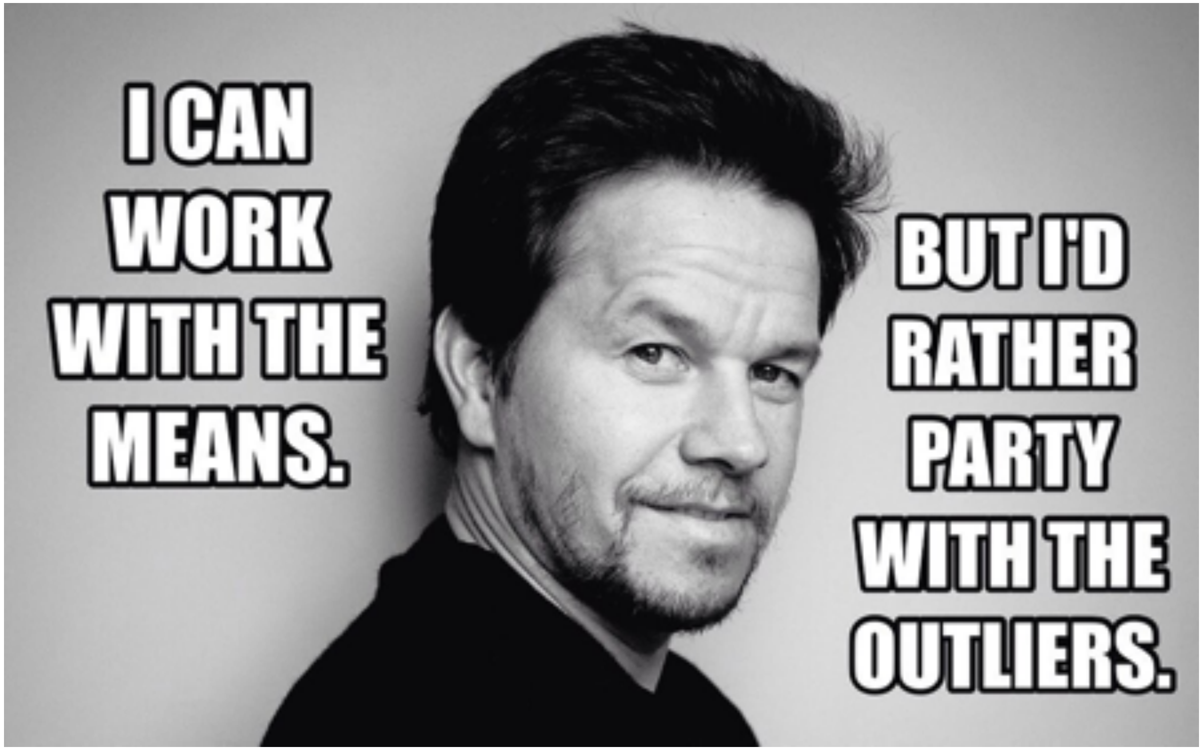
Other Tutorials:

http://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_vs_lda.html#example-decomposition-plot-pca-vs-lda-py

http://nbviewer.ipython.org/github/ogrisel/notebooks/blob/master/Labeled%20Faces%20in%20the%20Wild%20recognition.ipynb

section 2: screen moved down

Image Processing and Representation

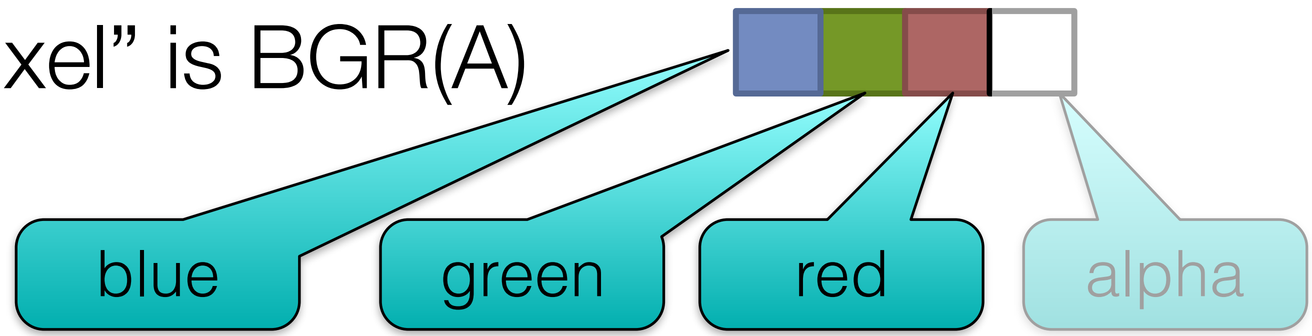


What is image processing

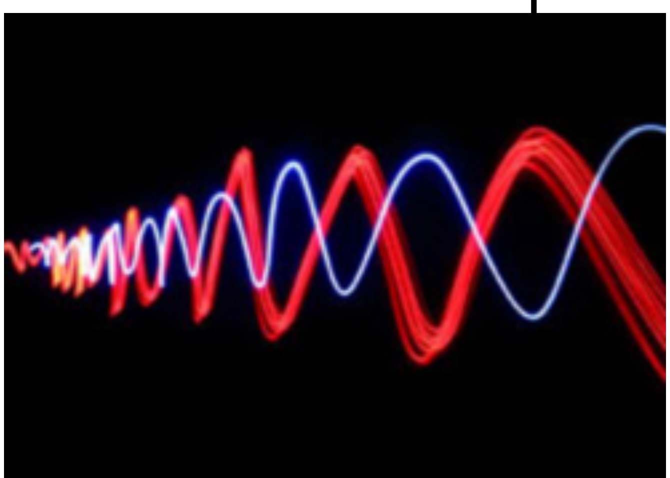
- the **art** and **science** of manipulating pixels
  - combining images (blending or compositing)
  - enhancing edges and lines
  - adjusting contrast, color
  - warping, transformation
  - filtering
  - features extraction

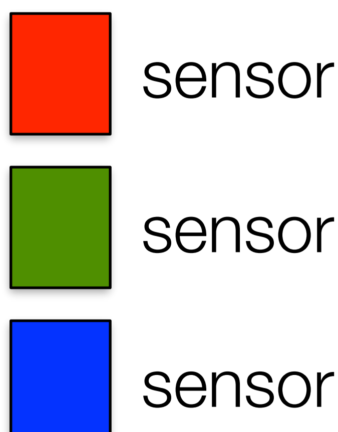
Images as data

- an image can be represented in many ways
- most common format is a matrix of pixels
  - each “pixel” is BGR(A)



- used for capture and display





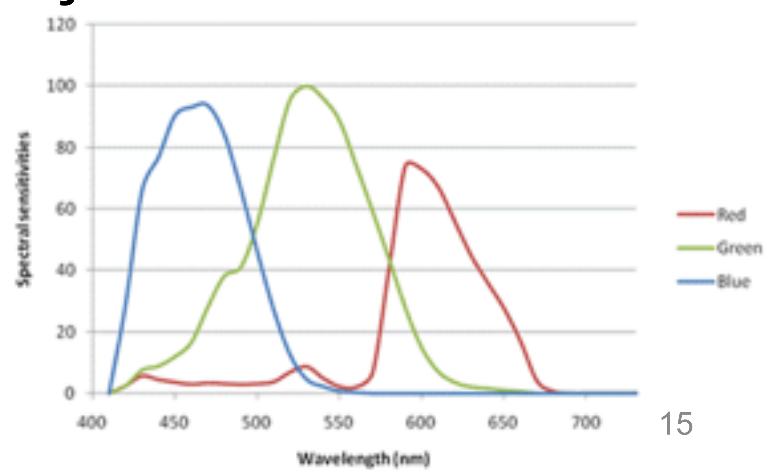


Image Representation

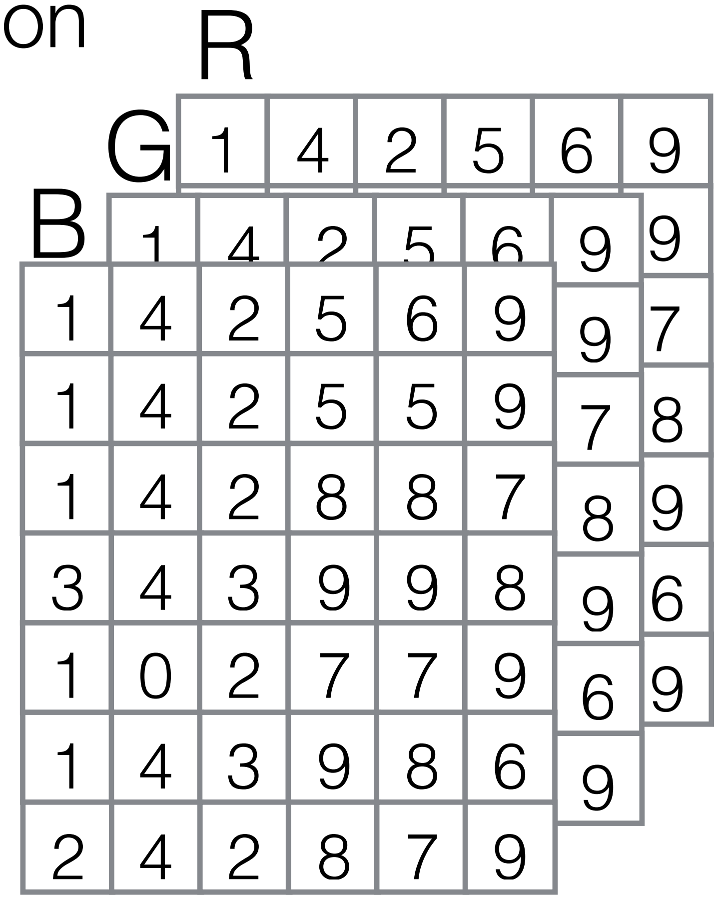
- need a compact representation

- **grayscale**
  0.3*R+0.59*G+0.11*B,
  "luminance"

gray

1	4	2	5	6	9
1	4	2	5	5	9
1	4	2	8	8	7
3	4	3	9	9	8
1	0	2	7	7	9
1	4	3	9	8	6
2	4	2	8	7	9

Numpy Matrix
image[rows, cols]



Numpy Matrix
image[rows, cols, channels]

Image Representation, Features

Problem: need to represent image as table data

1	4	2	5	6	9
1	4	2	5	5	9
1	4	2	8	8	7
3	4	3	9	9	8
1	0	2	7	7	9
1	4	3	9	8	6
2	4	2	8	7	9

Image Representation, Features

Problem: need to represent image as table data

Solution: row concatenation

Row 1	1	4	2	5	6	9	1	4	2	5	5	9	1	4	2	8	8	7	3
Row 2	1	4	2	8	8	7	3	4	3	9	9	8	1	4	2	5	5	9	1
…																			
Row N	9	4	6	8	8	7	4	1	3	9	2	1	1	5	2	1	5	9	1

Demo

Dimension Reduction with Images

Images Representation

Randomized PCA

Kernel PCA

section 2: screen moved down

Features of Images

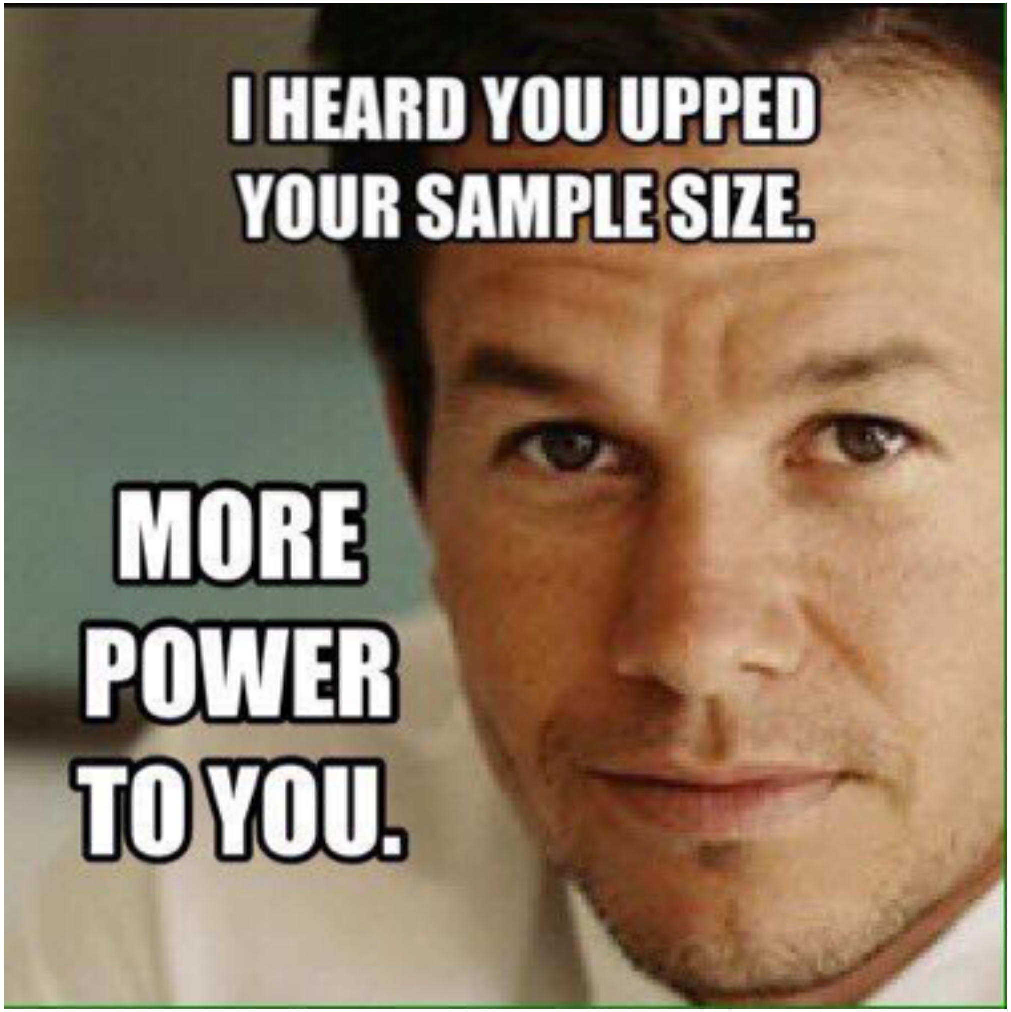
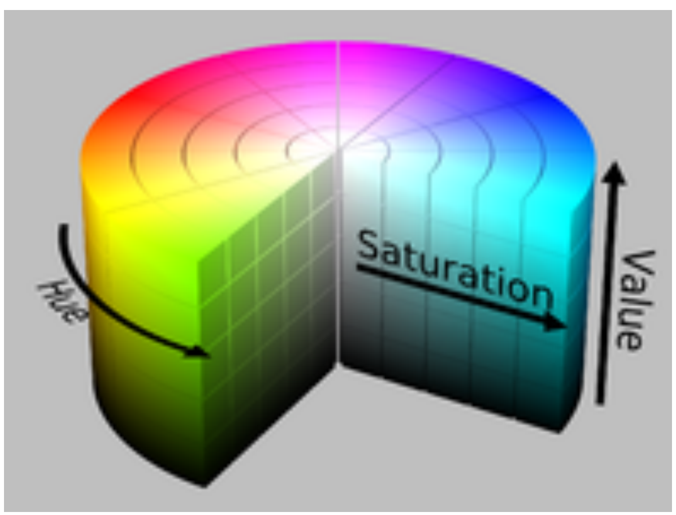
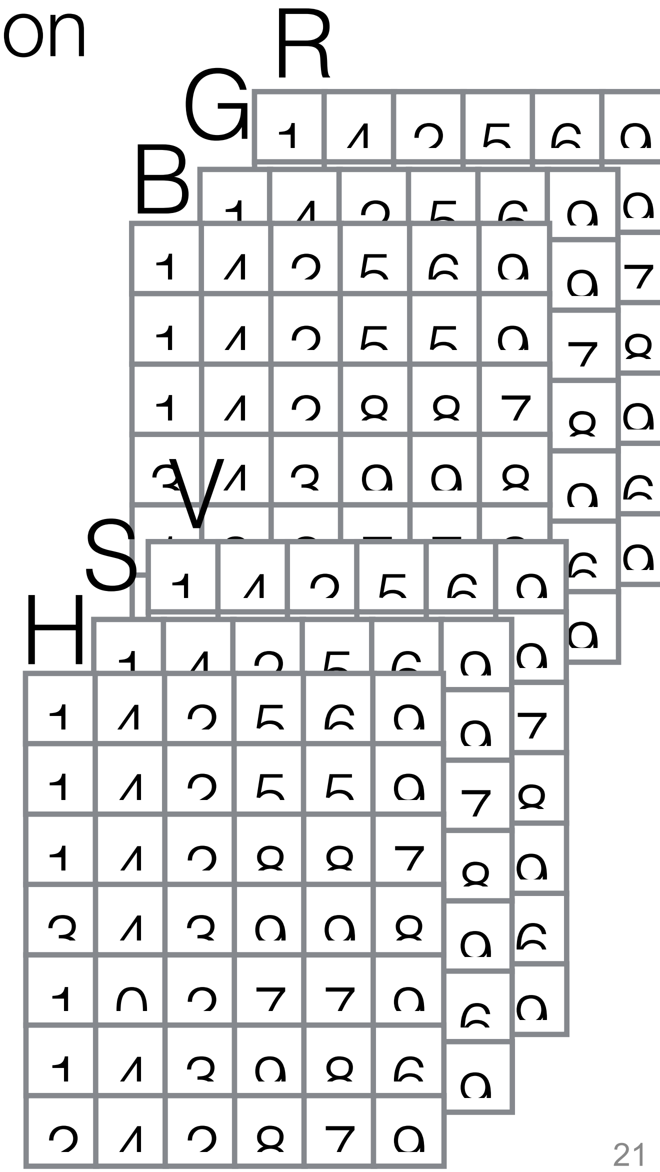


Image Representation

- need a compact representation

- **hsv**
  - what we perceive as color (ish)
    - hue: the color value
    - saturation: the richness of the color relative to brightness
    - value: the gray level intensity



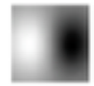


Common operations

- the gradient (2D derivative)

I

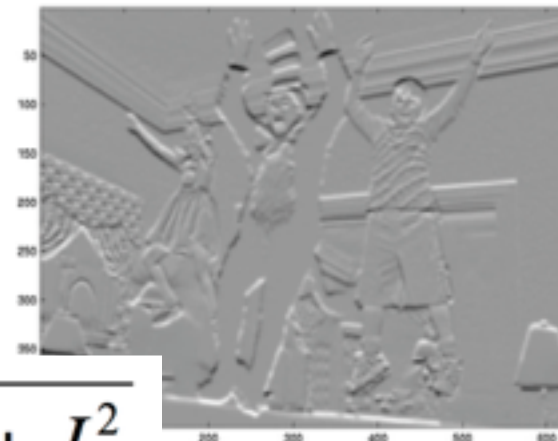
filter



J_x

filter

J_y

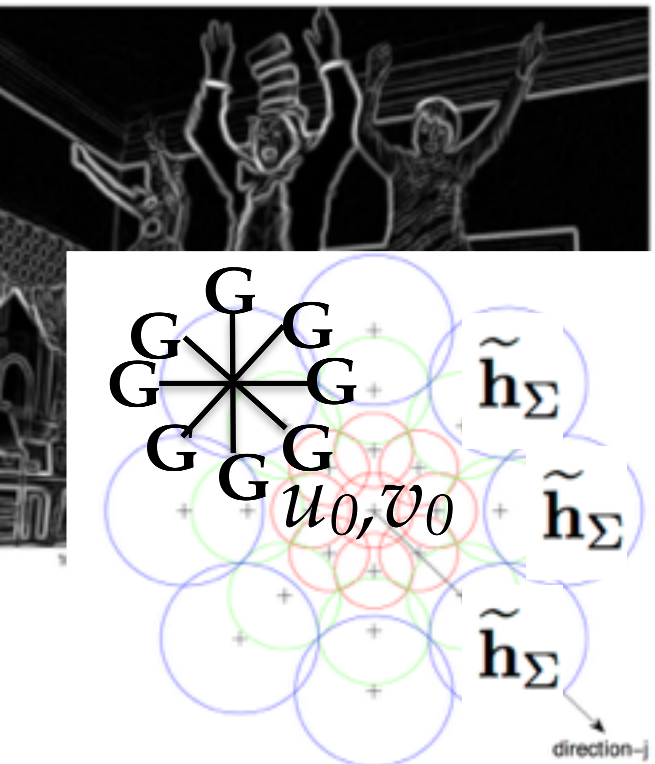


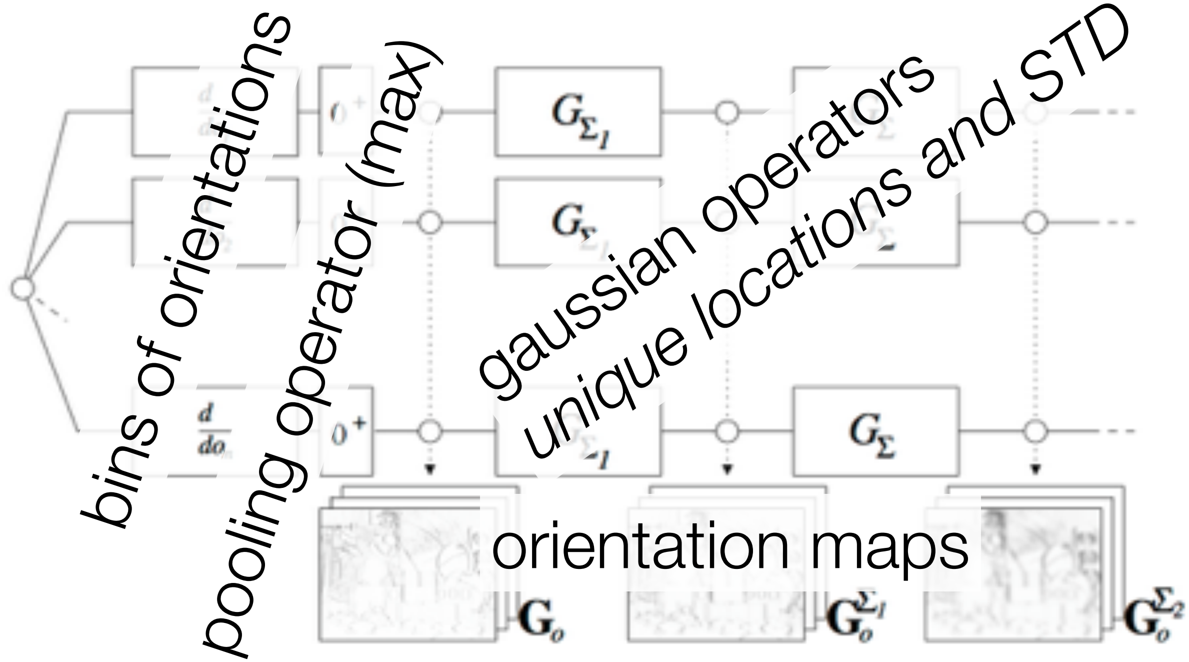
$$\|\nabla J\| = \sqrt{J_x^2 + J_y^2}$$

$$O = \tan^{-1}(Jx / Jy)$$

images: Jianbo Shi, Upenn

Common operations: DAISY





take normalized histogram at point u,v

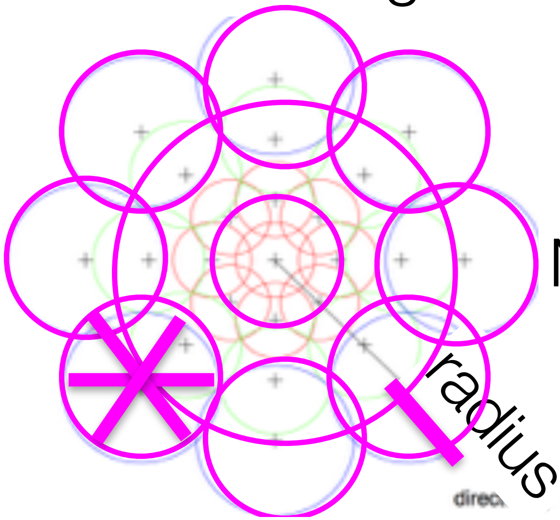
$$\widetilde{\mathbf{h}}_\Sigma(u,v) = \left\| \left[\mathbf{G}_1^\Sigma(u,v),\ldots,\mathbf{G}_H^\Sigma(u,v)\right]^\top \right\|$$

$$\begin{aligned}
&\mathcal{D}(u_0,v_0) = \\
&\Big[\ \widetilde{\mathbf{h}}_{\Sigma_1}^\top(u_0,v_0), \\
&\quad \widetilde{\mathbf{h}}_{\Sigma_1}^\top(\mathbf{l}_1(u_0,v_0,R_1)),\cdots,\widetilde{\mathbf{h}}_{\Sigma_1}^\top(\mathbf{l}_T(u_0,v_0,R_1)), \\
&\quad \widetilde{\mathbf{h}}_{\Sigma_2}^\top(\mathbf{l}_1(u_0,v_0,R_2)),\cdots,\widetilde{\mathbf{h}}_{\Sigma_2}^\top(\mathbf{l}_T(u_0,v_0,R_2)),
\end{aligned}$$

Tola et al. “Daisy: An efficient dense descriptor applied to wide-baseline stereo.” Pattern Analysis and Machine Intelligence, IEEE Transactions

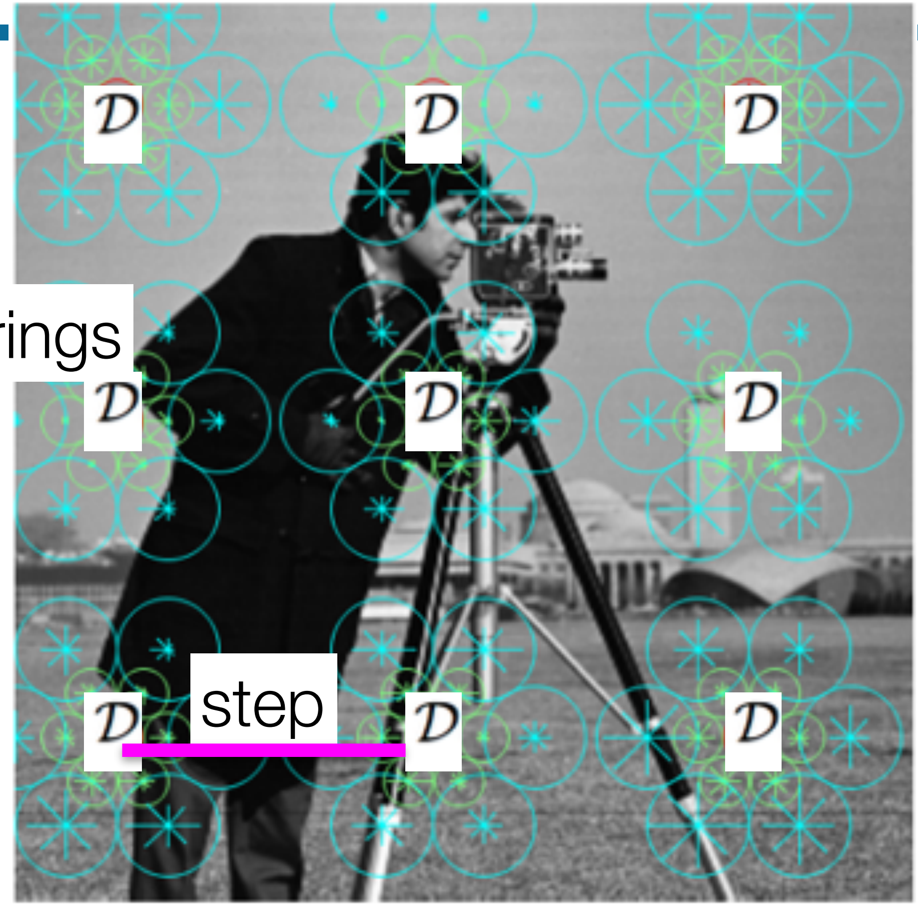
Common operations: DAISY

Num histograms



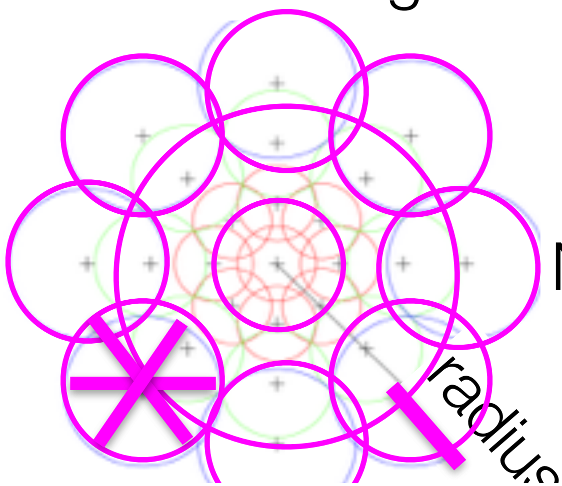
Num rings

num orientations (bins)



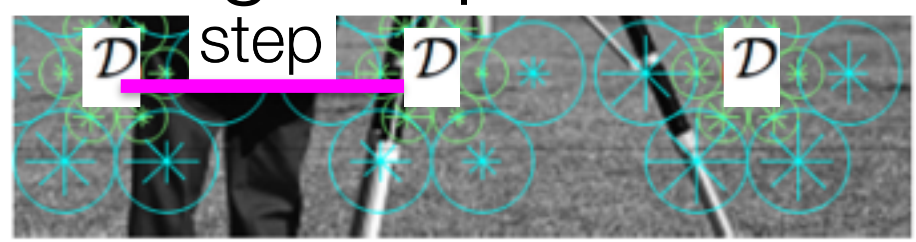
Params:
step, radius, num rings,
num histograms per ring,
orientations per histogram

Common operations: DAISY



num

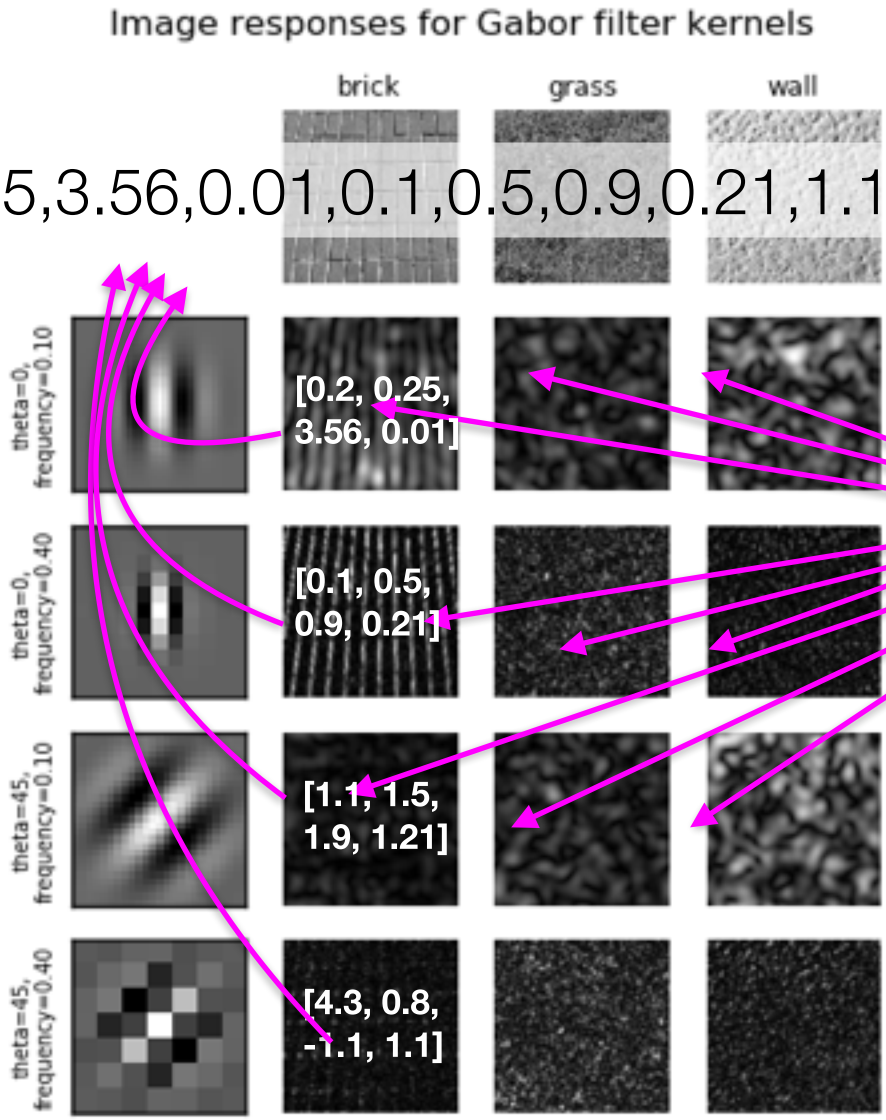
Bag of Features Image Representation

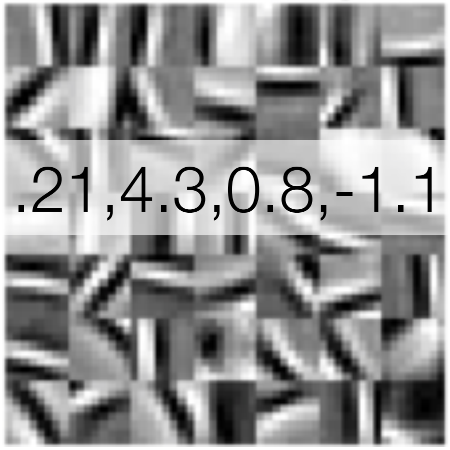


Params:
step, radius, num rings,
num histograms per ring,
orientations per histogram

Common operations: Gabor filter Banks (if time)

- common used for texture classification





[0.2,0.25,3.56,0.01,0.1,0.5,0.9,0.21,1.1,1.5,1.9,1.21,4.3,0.8,-1.1,1.1]

mean
std
skew
kurtosis

Concatenate statistics of different filters into one feature vector

Demo

More Image Processing

- Gradients
- DAISY
- Gabor Filter Banks

Other Tutorials:

http://scikit-image.org/docs/dev/auto_examples/

For Next Lecture

- There is no lecture next week!!
- Project work week:
  - Work on Lab One and Turn it in on Time.
- I am actually out of town (Germany)
- But email me your issues and I will try to get back when I can…

synchronous