

CS598PS - Machine Learning for Signal Processing

Introduction & Linear Algebra Basics

25 August 2020

* We'll start 5min late to facilitate slow logins

What do I need to know?

- I won't assume you know machine learning or signal processing
 - You are here to learn, not to know!
- Be comfortable with the basics
 - Some linear algebra, some probability
 - If you are rusty that's ok, you are here to learn

What is this class?

- Is it a signal processing class?
 - What is signal processing?
- Is it a machine learning class?
 - What is machine learning?

Signal Processing

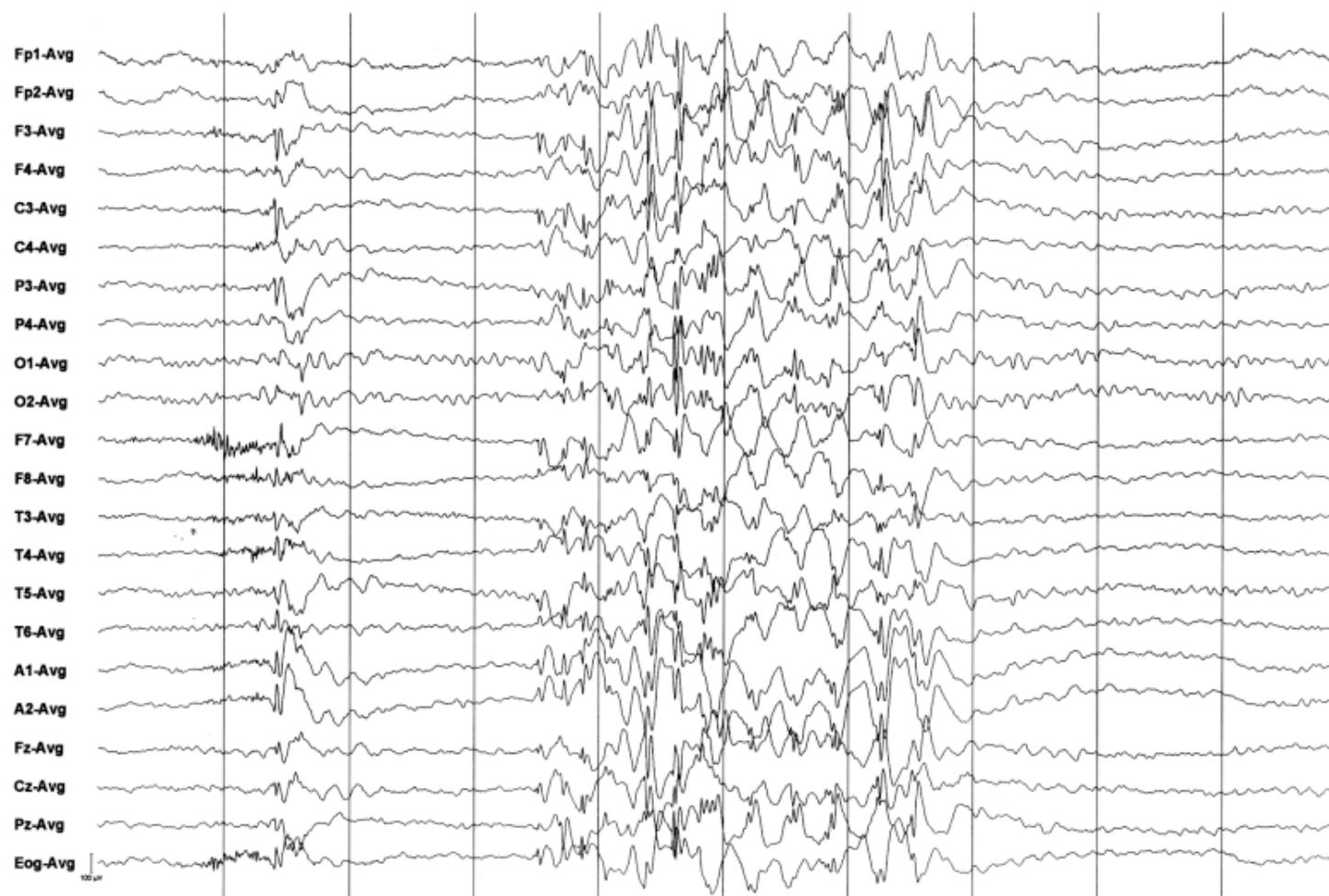
- The study of capturing, processing and manipulating *signals*
- What is a signal?

Signals (as far as this class goes)

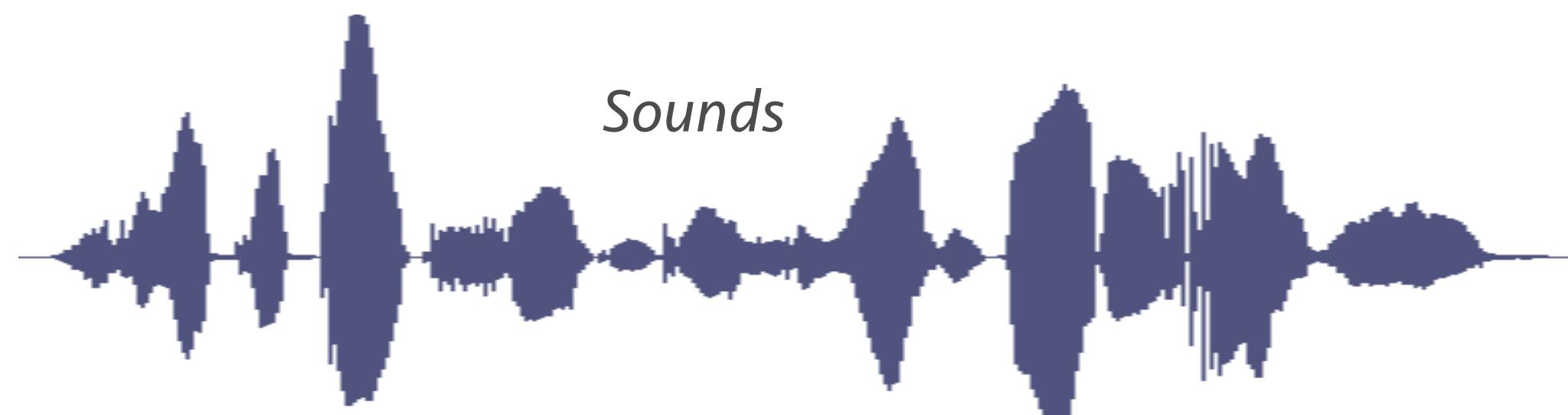
- *Structured collections of measurements that convey information*



Images

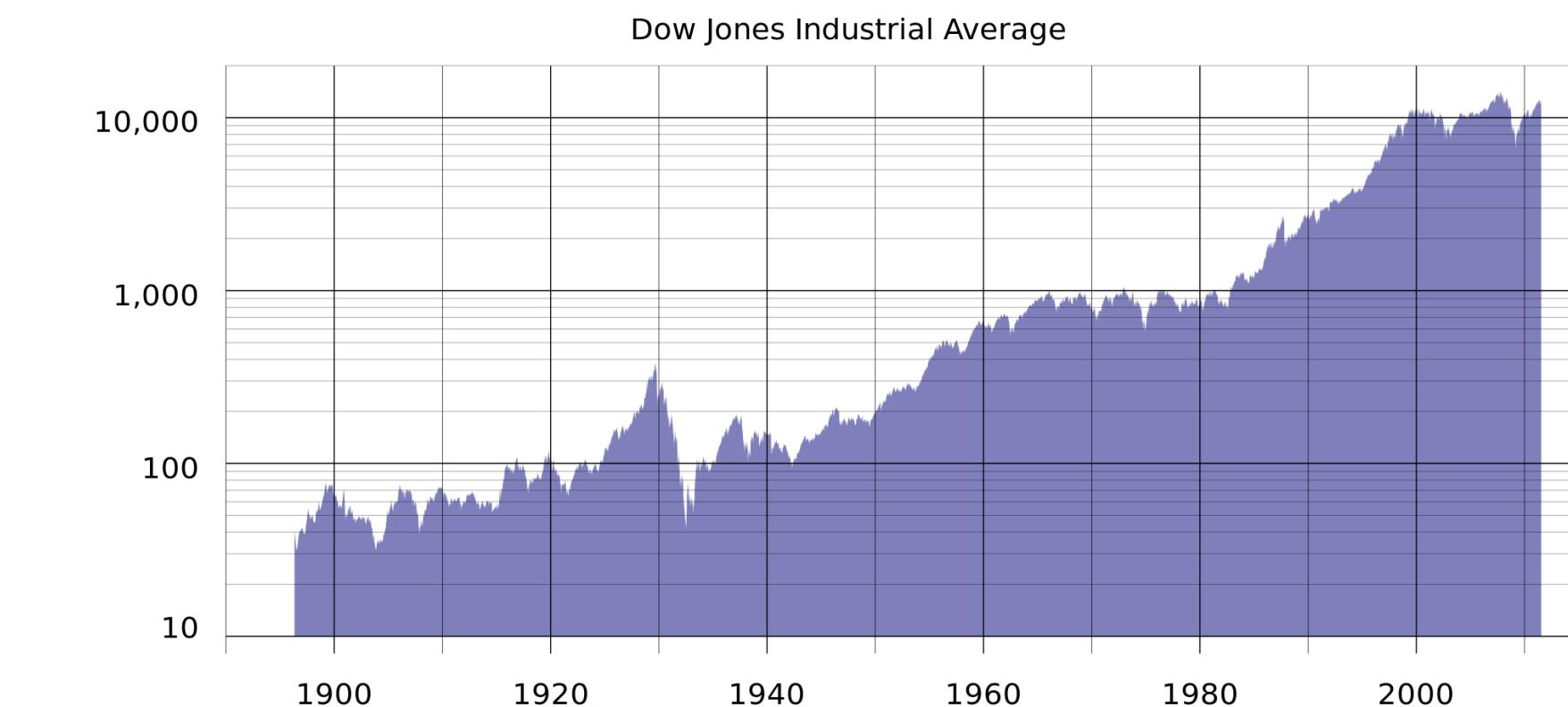


Brains!!



Sounds

Stocks



Dow Jones Industrial Average

Machine Learning

- The study of discovering and extracting information from *data*
- For data we will use signals

Why bother?

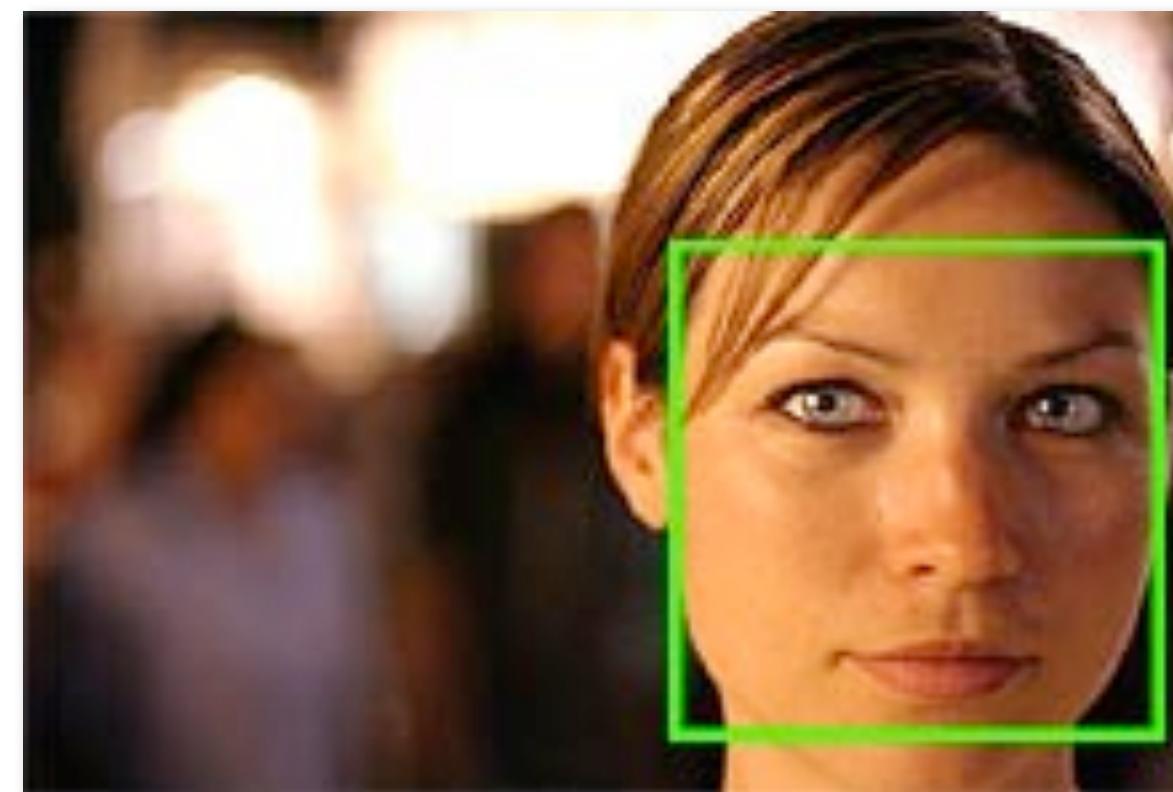
- Traditional signal processing doesn't really care (much) about its input's content
- Traditional machine learning is not very signals-friendly

MLSP

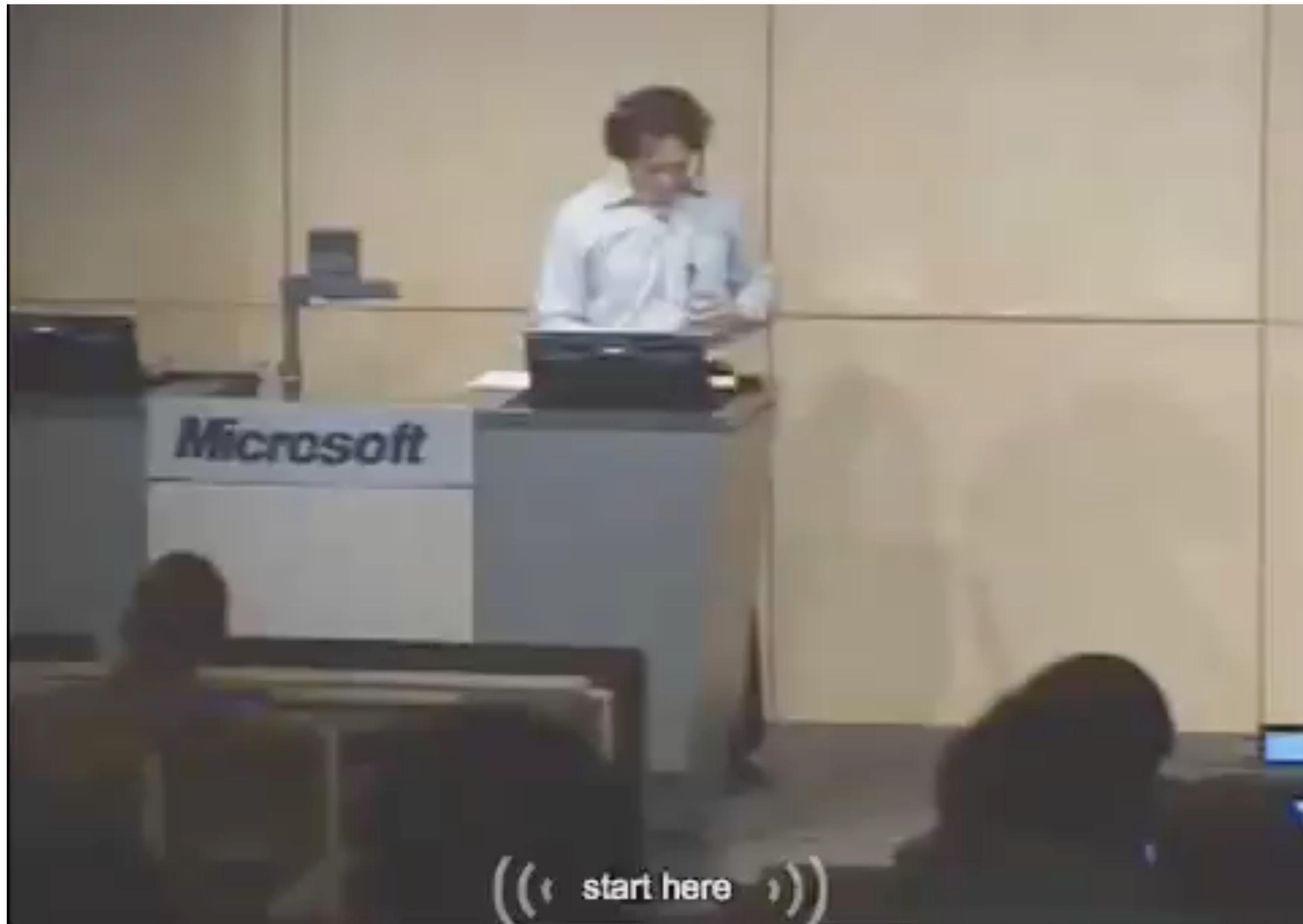
- MLSP combines both disciplines to perform learning on signals data
- Many examples of MLSP in the real-world

Face recognition

- Found in cameras and photo software



Speech recognition

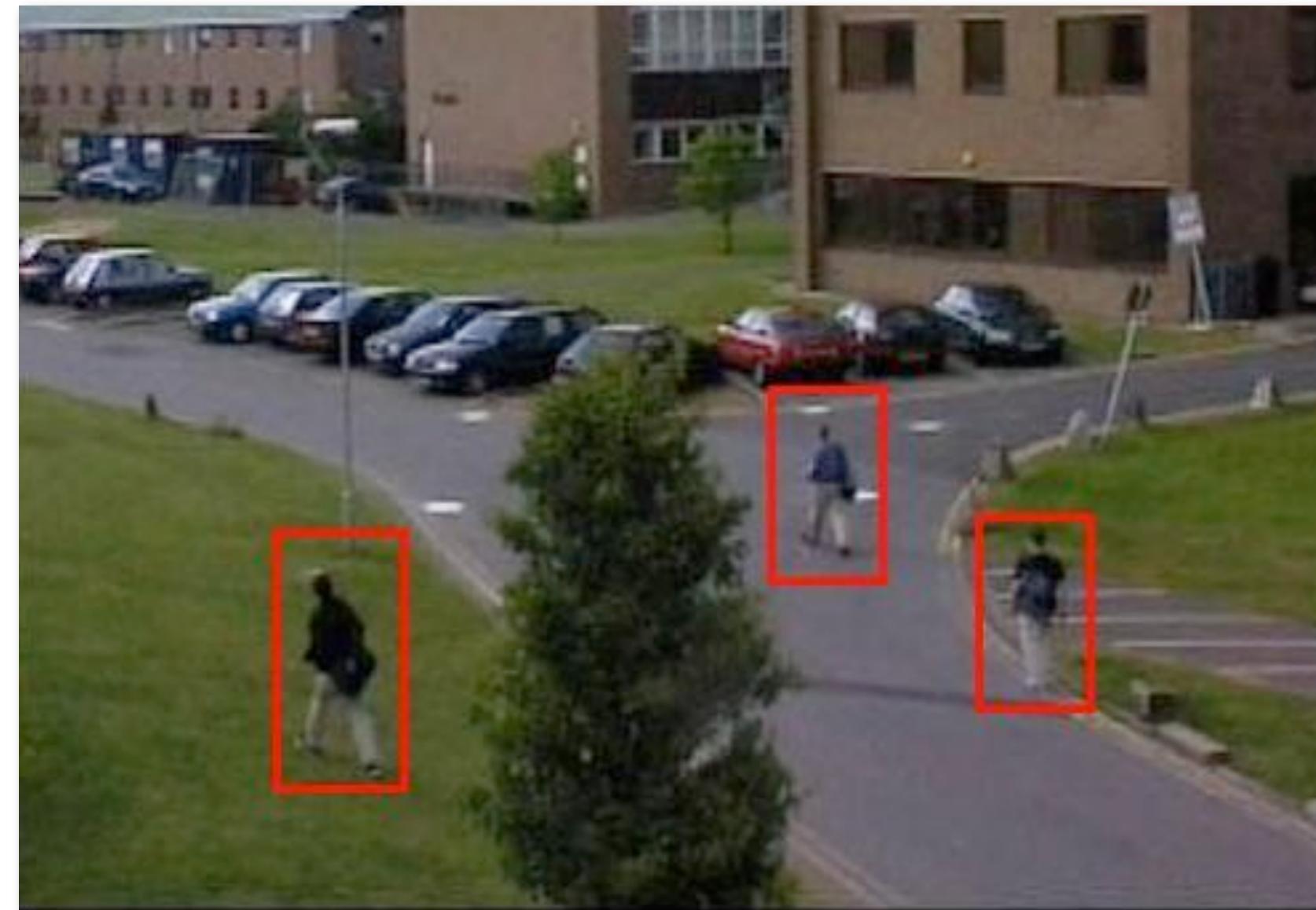


Surveillance/monitoring

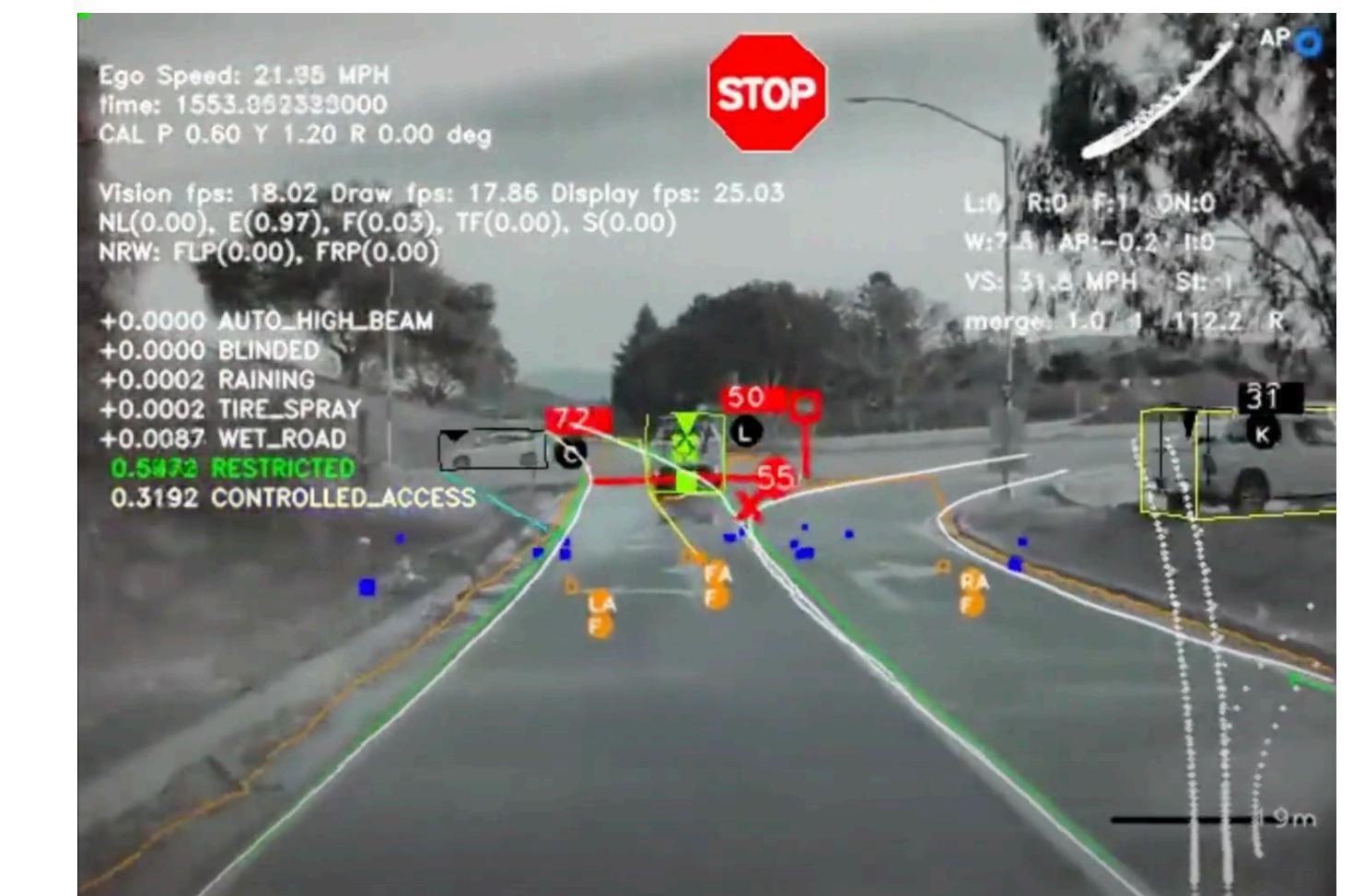
- Detection of specified objects / activity



Gunshot detection



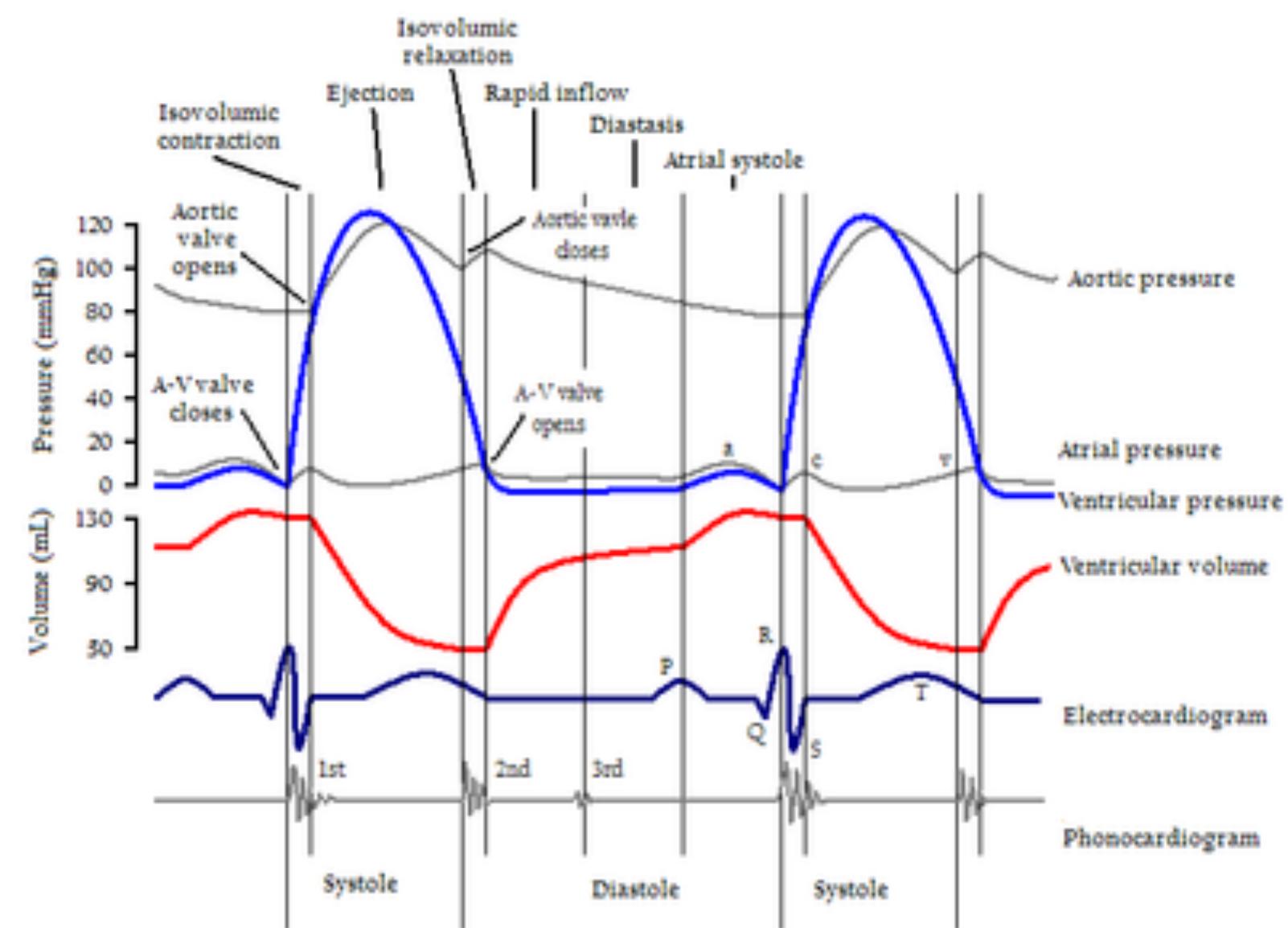
Pedestrian detection



Self-driving cars

Bio-signals

- Interpreting our body's data



Heartbeat processing



Brain Machine Interfaces

Many other applications

- Biometrics
- Music Information Retrieval
- Robotics
- Gesture-based UIs
- Condition monitoring (power grids, structures, automotive, ...)
- Financial data mining
- Many more ...

*All of these have been
class projects in this class!*

About this course

- Heavy on practical applications
 - Please bring your own domain problem in class!
- Won't go excruciatingly deep on theory
 - We'll skip convergence proofs, etc.
 - Many more courses here that cover all that
 - We'll go fast. Each lecture could be an entire topic
- Our goal is getting intuition and real-world experience

Syllabus: the basics

- Covering the basics:
 - Part 1: Linear Algebra and Probability
 - Part 2: Signals Theory
 - Part 3: Representations and Features

Syllabus: machine learning review

- Elements of machine learning
 - Part 4: Unsupervised learning basics
 - Part 5: Detection and classification
 - Part 6: Time-series and dynamical models

Syllabus: The fun stuff

- Applications and theory:
 - Matrix/Tensor Factorizations, Bag Models
 - Manifolds and Embedding
 - Deep Learning
 - Graph Signal Processing and Machine Learning
 - ICA, MIMO models, Arrays, Sensor Fusion
 - Compressive Sensing and Sparsity
 - Computer Vision, Speech Recognition
 - Music/Audio Informatics
 - Bio/Brain-signals
 - ...

Your part

- Problem sets: 40% of the grade
 - 3 to 5 throughout class, will be mostly machine problems
- Final project: 50% of the grade
 - Mid-semester: Proposal due
 - Last 1-2 weeks: Presentations and/or posters
- Remaining 10% of grade
 - Be active in class, ask questions, make sure I know who you are!

The final project is “conference style”

- Teams of 2-3 students (no more than 3, no less than 2)
 - Start making friends now!
- Mid-term: Abstract submission
 - Short abstract describing the problem you want to solve and how you plan to
- Week ~13: Paper Submission
 - 4-6 page paper
 - Peer reviewed by all of you
 - All papers will be accepted and presented

Communications

- We have a [piazza.com](#) page
 - Look for CS598PS
 - I will upload lectures, problem sets, data, links, etc.
 - Use it for questions, discussions, finding project-mates, etc.
 - Will be used for all class announcements
 - (so sign up if you aren't in it)
- We will also use Slack for quickie questions
 - Use signup link from class web page
 - Slack doesn't have a good issues tracking mechanism, so if you really want an answer use piazza for your questions

Course staff: Who am I?

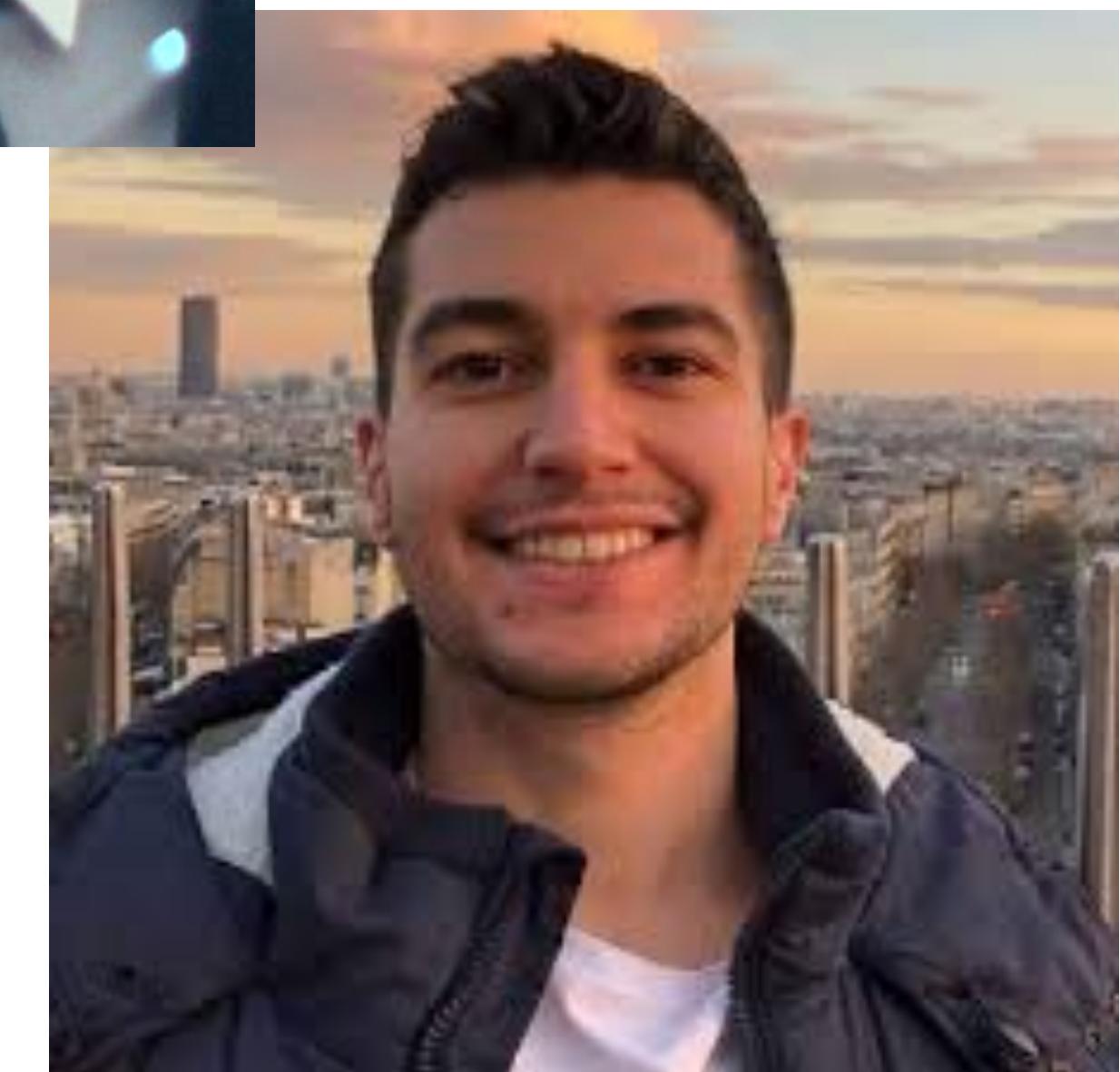
- Instructor: Paris Smaragdis (CS & ECE)
 - paris@illinois.edu
 - Office: ~~Siebel Center 3231~~ Home
 - Send me email/PM if you want to meet
- Interested in machine perception, computational audition
- Previously chaired the IEEE MLSP Technical Committee, the IEEE SPS Data Science Initiative, and the ICA/LVA community
- I've written many MLSP papers, and created many MLSP products

Who are the TAs?

- Zhepei Wang & Thymios Tzinis
 - zhepeiw2@illinois.edu
 - etzinis2@illinois.edu
- TA office hours will be online
 - Using zoom/slack
 - We will be adding them in the class calendar
- Some advice
 - Seek help early, avoid mad rush before deadlines
 - Make use of Piazza/Slack, you'll get answers faster



← Zhepei



Thymios →

Who are you?

- Name, department, undergrad?
 - What are your interests in this area?
 - Hint: you will need a project-mate (take notes!)
 - We usually do this live, but maybe you can take the time to quickly introduce yourself to the class in the chat window
 - Name, department, research interest, etc.

Final administrative note

- Enrollment constraints
 - Yes, they are annoying, if you need a waiver let me know
 - I do not foresee problems with class capacity
- If you are not formally registered yet:
 - Please add yourself to piazza and slack
 - Keep up with the assignments, we won't accept late ones

Linear algebra refresher

- Linear algebra is the math tool du jour
 - Compact notation
 - Convenient set operations
 - Used in all modern texts
 - Interfaces well with MATLAB, numpy, R, TensorFlow, etc.
- We will use a lot of it!!

Scalars, Vectors, Matrices, Tensors

0th order tensor

$$\chi$$

1st order tensor

$$\chi = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

2nd order tensor

$$\mathbf{X} = \begin{bmatrix} \mathbf{x} & \dots & \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & & x_{N,1} \\ \vdots & \ddots & \vdots \\ x_{1,M} & & x_{N,M} \end{bmatrix}$$

1st dimension
2nd dimension

3rd order tensor

$$\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_K\} = \begin{bmatrix} x_{1,1,1} & & x_{N,1,1} \\ \vdots & \ddots & \vdots \\ x_{1,M,1} & & x_{N,M,1} \\ & \ddots & \ddots \\ & & x_{1,1,K} \\ & & \vdots \\ & & x_{1,M,K} & \dots & x_{N,M,K} \end{bmatrix}$$

1st dimension
2nd dimension
3rd dimension

How will we see these?

- 1D signals (e.g. sounds) will be vectors

$$\mathbf{x}^\top = \begin{bmatrix} x(0) & \dots & x(T) \end{bmatrix} = \left[\begin{array}{c} \text{blue audio waveform} \end{array} \right]$$

- 2D signals (e.g. images) will be matrices

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & & x_{N,1} \\ \vdots & \dots & \vdots \\ x_{1,M} & & x_{N,M} \end{bmatrix} = \left[\begin{array}{c} \text{gray image matrix} \end{array} \right]$$

- 3D data will be videos, etc ...

Element-wise operations

- Addition/subtraction

$$\mathbf{a} \pm \mathbf{b} = \mathbf{c} \rightarrow a_i \pm b_i = c_i$$

- Multiplication (Hadamard product)

$$\mathbf{a} \odot \mathbf{b} = \mathbf{c} \rightarrow a_i b_i = c_i$$

- Other times denoted as $\mathbf{a} \circ \mathbf{b}$
- No named operator for element-wise division
 - Just use Hadamard with reciprocal \mathbf{b}

Transpose

- Transpose
 - Change rows to columns (and vice versa)

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}, \quad \mathbf{x}^\top = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 & x_3 \\ x_2 & x_4 \end{bmatrix}, \quad \mathbf{x}^\top = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}$$

- Hermitian (conjugate transpose)

- Notated as \mathbf{X}^H
- In addition to transposing we also take the complex conjugate of each entry
 - If you have complex-valued data this makes a big difference!

Visualizing transposition

- Mostly pointless for 1D signals

$$\mathbf{x} = \begin{bmatrix} \text{blue waveform} \\ \text{blue waveform} \\ \text{blue waveform} \\ \text{blue waveform} \end{bmatrix}, \quad \mathbf{x}^\top = \begin{bmatrix} \text{blue waveform} & \text{blue waveform} & \text{blue waveform} & \text{blue waveform} \end{bmatrix}$$

- Swap dimensions for 2D signals

$$\mathbf{x} = \begin{bmatrix} \text{gray image} & \text{black image} \\ \text{black image} & \text{gray image} \end{bmatrix}, \quad \mathbf{x}^\top = \begin{bmatrix} \text{gray image} \\ \text{black image} \end{bmatrix}$$

Reshaping operators

- The vec operator
 - Unrolls elements column-wise
 - Or in order of dimensions for tensors
 - Useful for getting rid of matrices/tensors

$$\text{vec}(\mathbf{x}) = \text{vec} \begin{bmatrix} x_1 & x_3 \\ x_2 & x_4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

- The vec-transpose
 - For (p) of an $m \times n$ matrix make each column a $p \times (m/p)$ matrix
 - Useful for inverting vec and getting rid of tensors $\mathbf{X} = \text{vec}(\mathbf{X})^{(M)}, \mathbf{X} \in \mathbb{R}^{M \times N}$

$$\mathbf{x}^{(2)} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \\ x_{41} & x_{42} \\ x_{51} & x_{52} \\ x_{61} & x_{62} \end{bmatrix}^{(2)} = \begin{bmatrix} x_{11} & x_{31} & x_{51} \\ x_{21} & x_{41} & x_{61} \\ x_{12} & x_{32} & x_{52} \\ x_{22} & x_{42} & x_{62} \end{bmatrix}$$

Trace and diag

- Matrix trace
 - Sum of diagonal elements
- The diag operator

$$\text{tr}(\mathbf{X}) = \text{tr} \begin{bmatrix} x_{11} & & & & x_{1N} \\ \vdots & \ddots & & & \vdots \\ x_{N1} & & \ddots & & x_{NN} \\ \vdots & & & \ddots & \vdots \end{bmatrix} = \sum_i x_{ii}$$

$$\text{diag}(\mathbf{x}) = \text{diag} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 & 0 \\ 0 & x_2 \end{bmatrix} \quad \text{Make a diagonal matrix from a vector}$$

$$\text{diag}^{-1} \begin{bmatrix} x_1 & a \\ b & x_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \text{Extract diagonal as a vector}$$

Norms

- 2-norm:

$$\|\mathbf{x}\| = \sqrt{\sum_i |x_i|^2}$$

← Often lazily written as $|x|$ as well

- p -norms:

$$\|\mathbf{x}\|_p = \left(\sum_i |x_i|^p \right)^{1/p}$$

- Frobenius norm:

$$\|\mathbf{X}\|_F = \sqrt{\sum_{i,j} |x_{i,j}|^2} = \sqrt{\text{tr}(\mathbf{X} \cdot \mathbf{X}^H)}$$

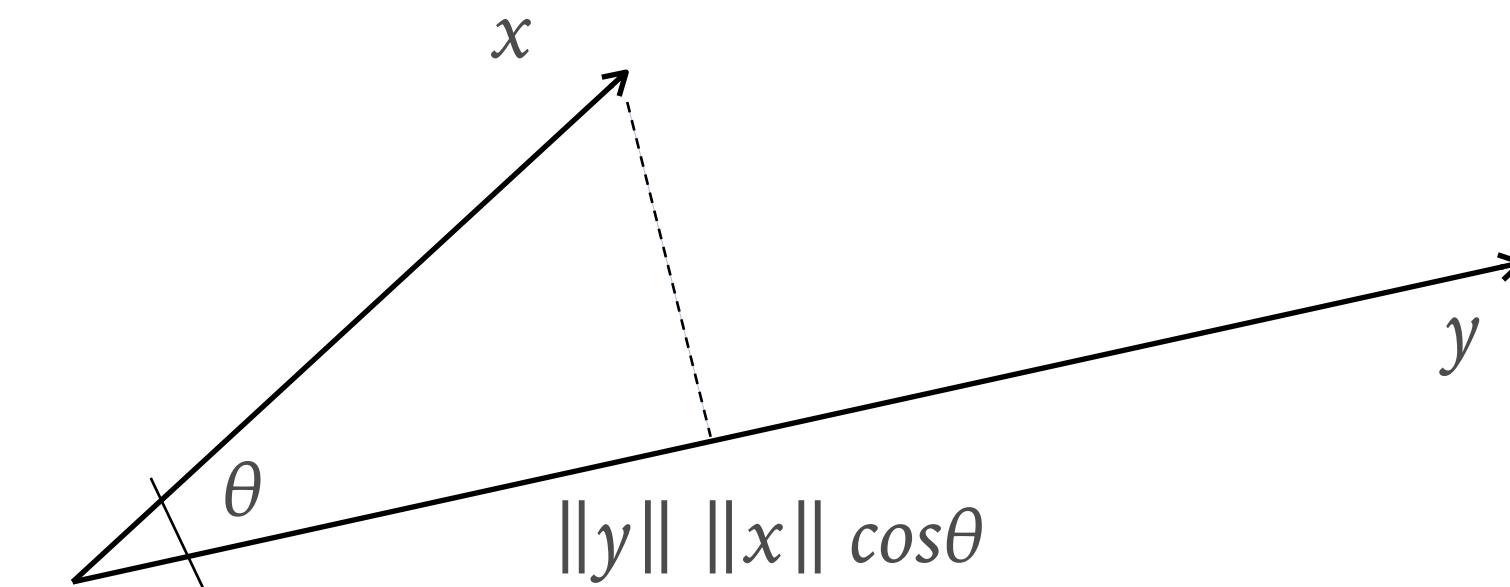
The dot product

- Dot product
 - Shorthand for multiply and accumulate

$$\mathbf{x}^\top \cdot \mathbf{y} = \sum_i x_i \cdot y_i = \|\mathbf{x}\| \cdot \|\mathbf{y}\| \cos \theta$$

- Geometry
 - For unit vectors:

$$\theta = \arccos(\mathbf{x}^\top \cdot \mathbf{y})$$



- Great tool for checking out similarity

Matrix-vector product

- Generalizing the dot product

$$\mathbf{X} \cdot \mathbf{y} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \mathbf{x}_3^\top \end{bmatrix} \cdot \mathbf{y} = \begin{bmatrix} \mathbf{x}_1^\top \cdot \mathbf{y} \\ \mathbf{x}_2^\top \cdot \mathbf{y} \\ \mathbf{x}_3^\top \cdot \mathbf{y} \end{bmatrix} = \begin{bmatrix} \sum x_{1,i} \cdot y_i \\ \sum x_{2,i} \cdot y_i \\ \sum x_{3,i} \cdot y_i \end{bmatrix}$$

- \mathbf{X} must have as many columns as \mathbf{y} has elements
 - Does not commute!
- Useful for computing multiple dot products
 - Pack all vectors that you want to multiply in a matrix

Matrix-matrix product

- Between two matrices:

$$\mathbf{X} \cdot \mathbf{Y} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \mathbf{x}_3^\top \end{bmatrix} \cdot \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^\top \cdot \mathbf{y}_1 & \mathbf{x}_1^\top \cdot \mathbf{y}_2 & \mathbf{x}_1^\top \cdot \mathbf{y}_3 \\ \mathbf{x}_2^\top \cdot \mathbf{y}_1 & \mathbf{x}_2^\top \cdot \mathbf{y}_2 & \mathbf{x}_2^\top \cdot \mathbf{y}_3 \\ \mathbf{x}_3^\top \cdot \mathbf{y}_1 & \mathbf{x}_3^\top \cdot \mathbf{y}_2 & \mathbf{x}_3^\top \cdot \mathbf{y}_3 \end{bmatrix}$$

- \mathbf{X} must have as many columns as \mathbf{Y} has rows
 - $(M \times N) = (M \times K) \cdot (K \times N)$
 - Remember this operation doesn't commute!
- All linear operations can be represented as a matrix product
 - We'll be seeing that a lot!

Matrix products

- # of output rows == left matrix # of rows
- # of output columns == right matrix # of columns

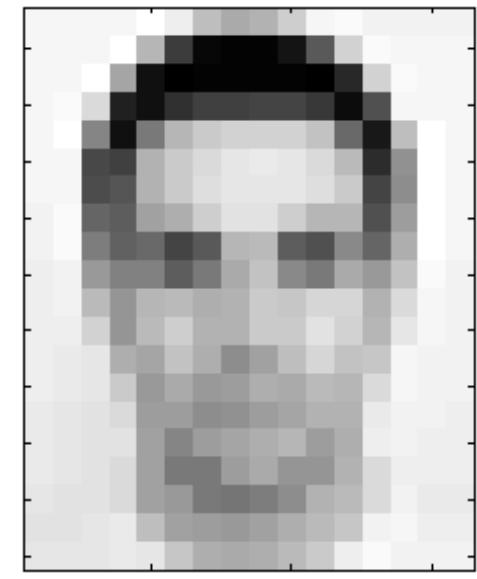
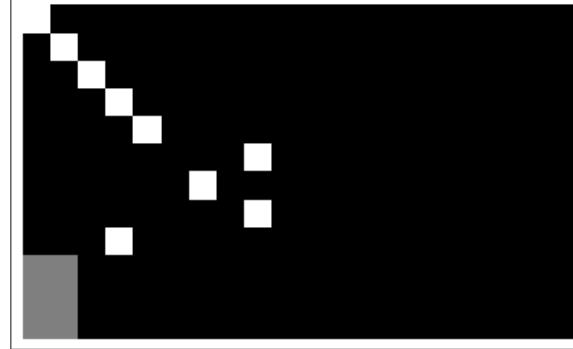
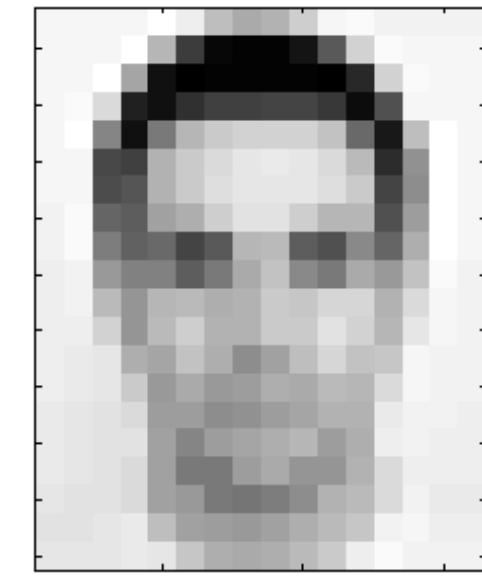
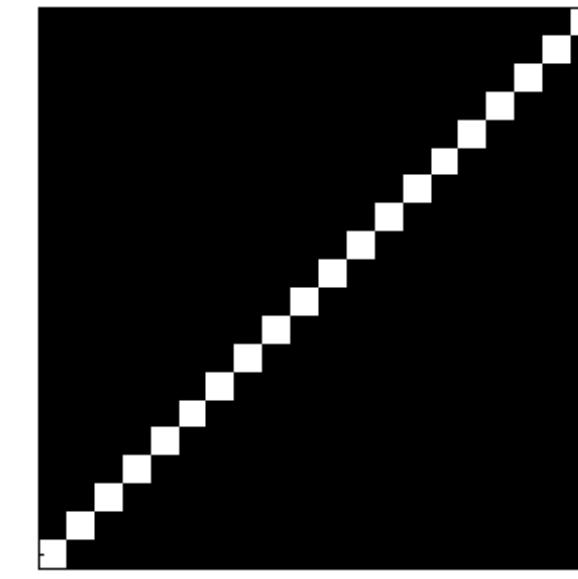
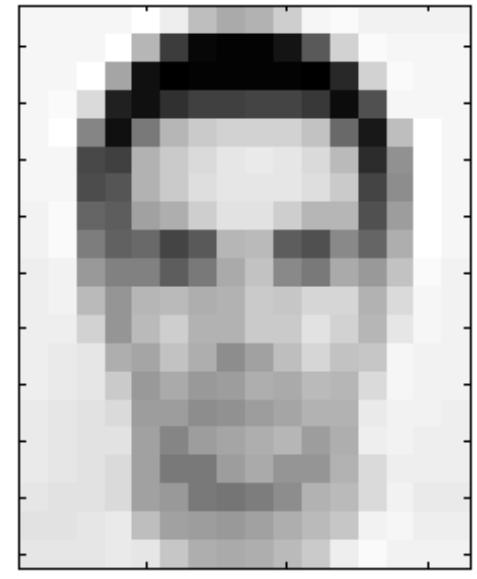
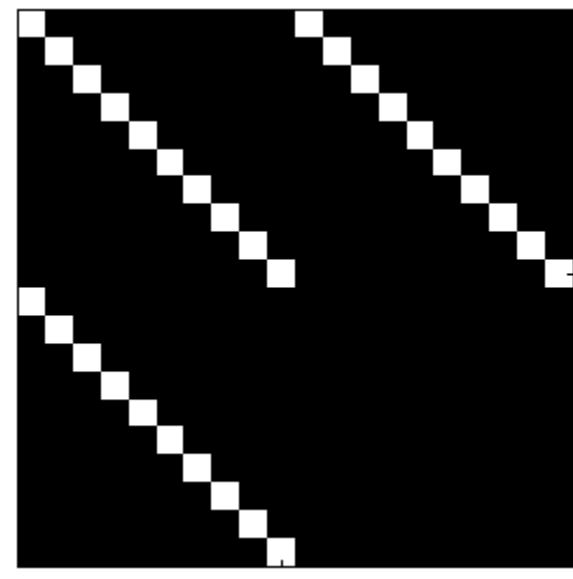
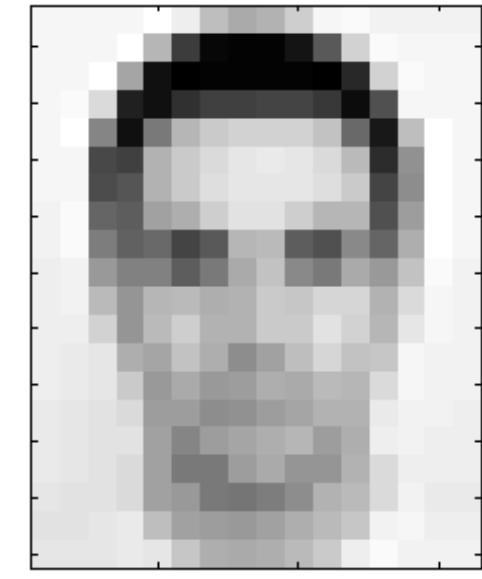
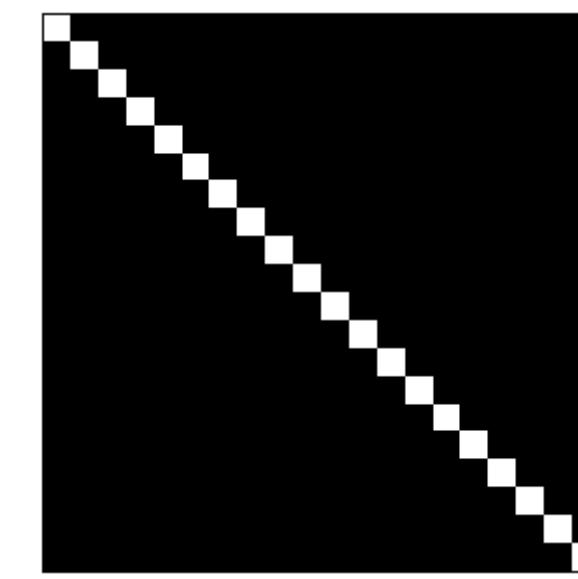
$$\begin{matrix} & \text{Fat} \\ \text{Short} & \end{matrix} = \begin{matrix} \text{Short} \\ & \text{Fat} \end{matrix}$$

$$\begin{matrix} & \text{Thin} \\ \text{Short} & \end{matrix} = \begin{matrix} \text{Short} \\ & \text{Thin} \end{matrix}$$

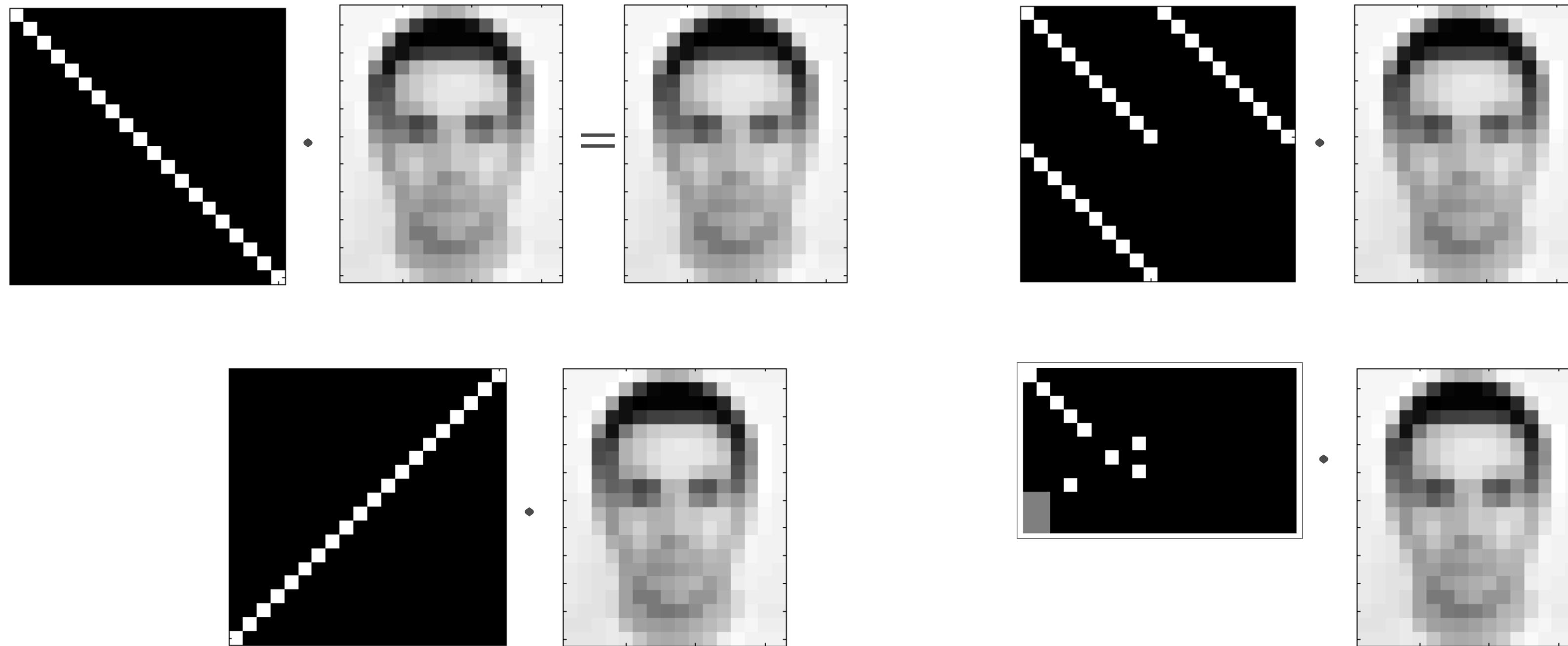
$$\begin{matrix} & \text{Fat} \\ \text{Tall} & \end{matrix} = \begin{matrix} \text{Tall} \\ & \text{Fat} \end{matrix}$$

$$\begin{matrix} & \text{Regular} \\ \text{Short} & \end{matrix} = \begin{matrix} \text{Short} \\ & \text{Regular} \end{matrix}$$

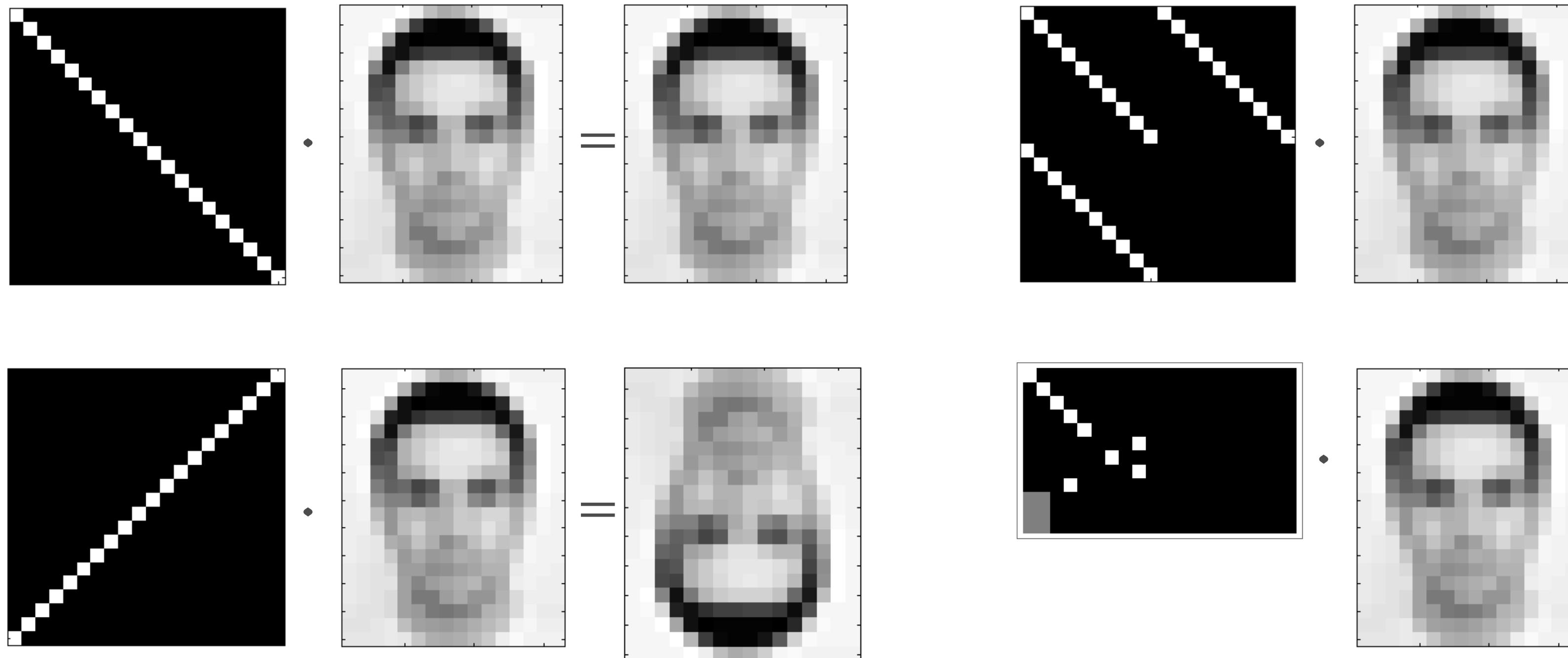
Visualizing the matrix product



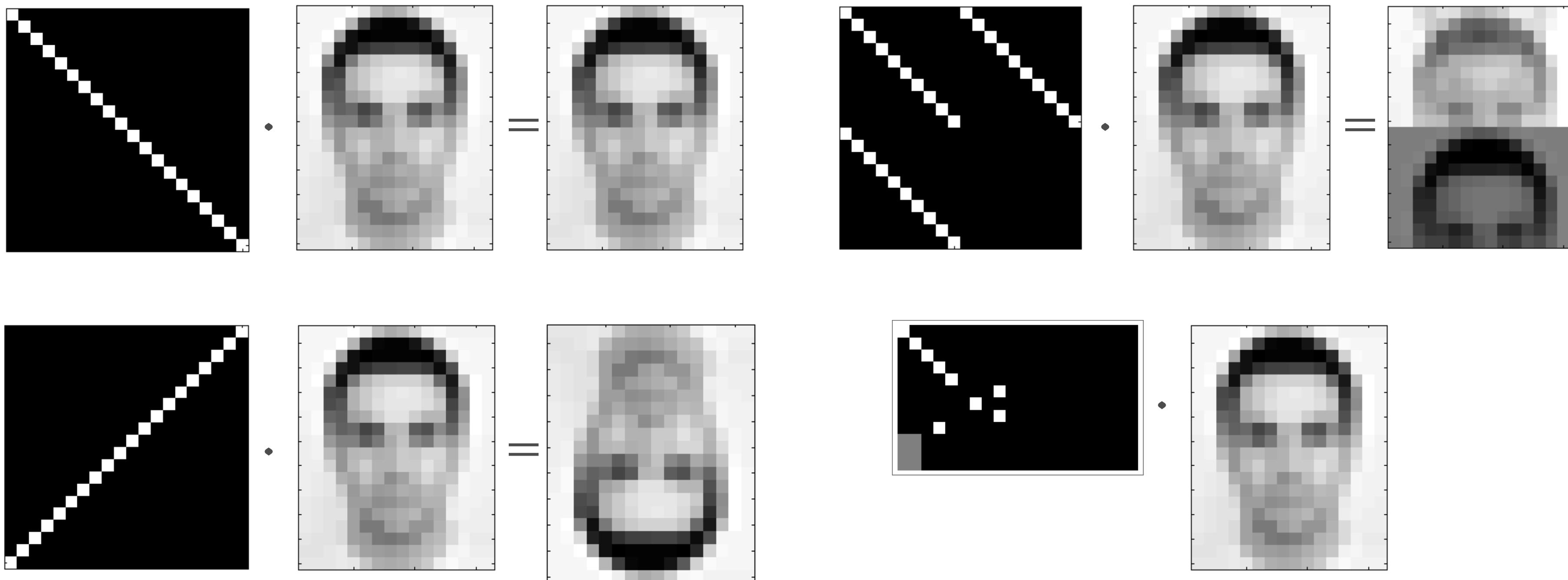
Visualizing the matrix product



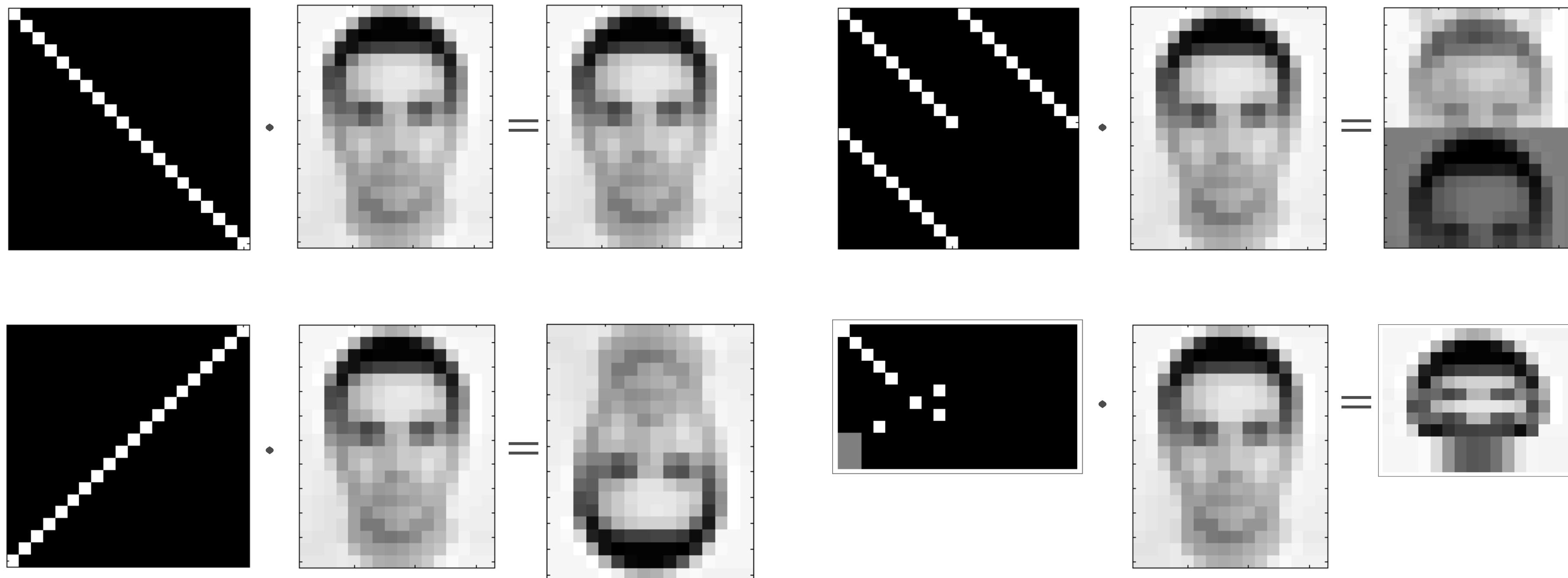
Visualizing the matrix product



Visualizing the matrix product



Visualizing the matrix product



Kronecker product

- A bit more complex
 - Replicate and multiply right matrix with each scalar of left matrix

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \otimes \mathbf{Y} = \begin{bmatrix} x_{11} \cdot \mathbf{Y} & x_{12} \cdot \mathbf{Y} \\ x_{21} \cdot \mathbf{Y} & x_{22} \cdot \mathbf{Y} \end{bmatrix}$$

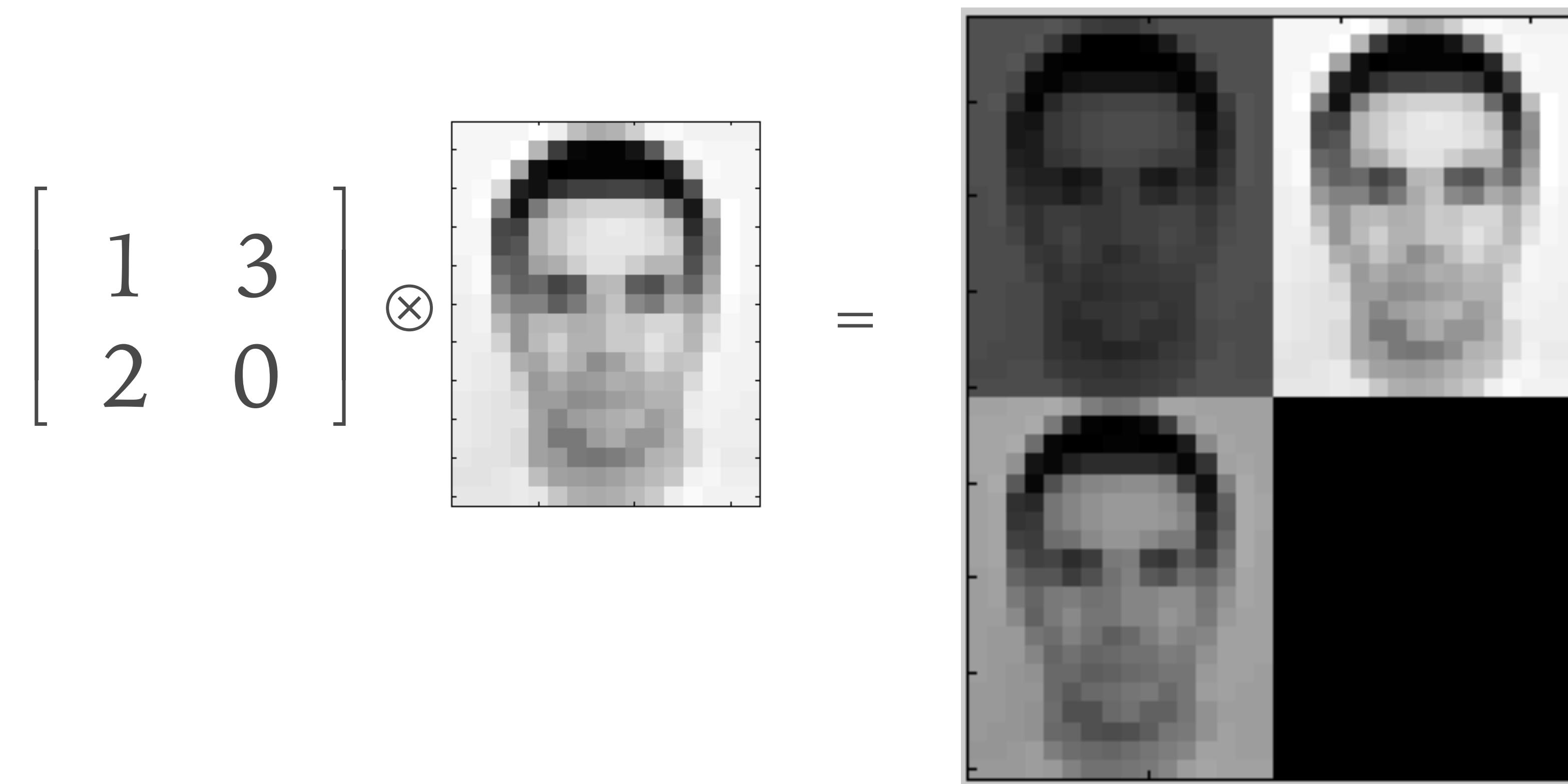
- Useful result:

$$\text{vec}(\mathbf{X} \cdot \mathbf{Y} \cdot \mathbf{Z}) = (\mathbf{Z}^\top \otimes \mathbf{X}) \text{vec}(\mathbf{Y})$$

Visualizing Kronecker

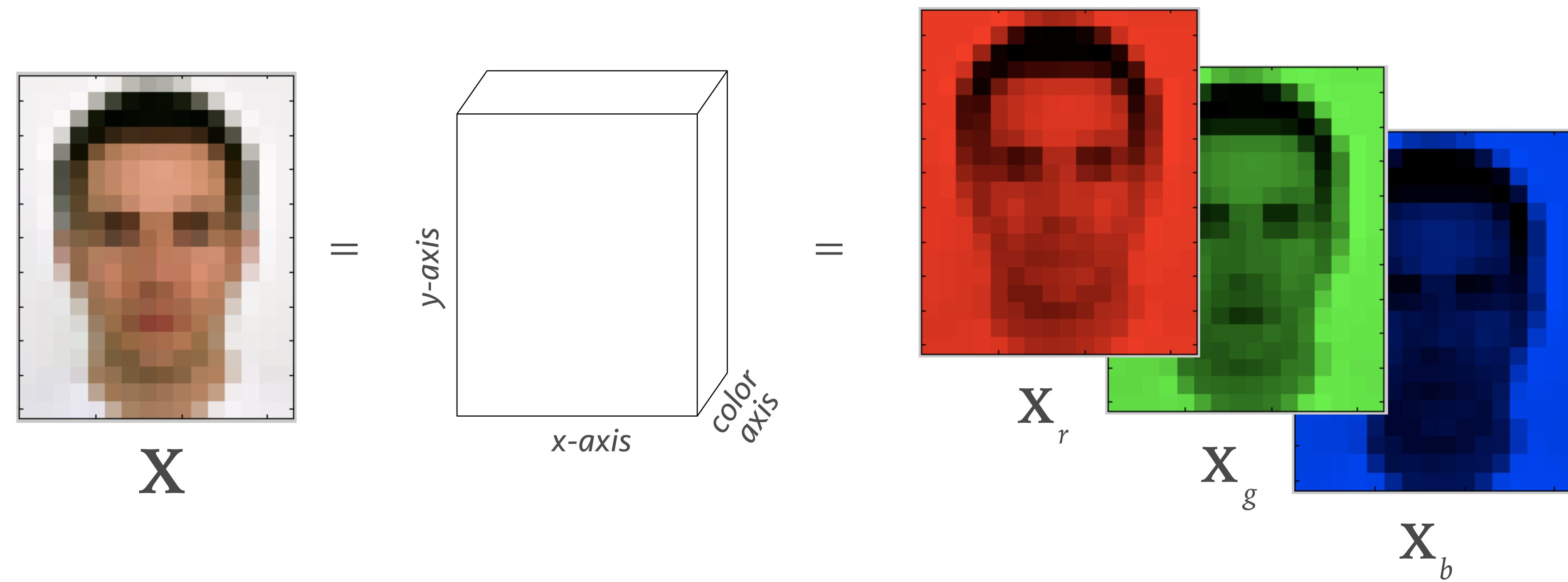
$$\begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} \otimes \begin{array}{c} \text{A 4x4 grayscale image of a face} \\ \text{with a black border} \end{array} = ?$$

Visualizing Kronecker



Dealing with tensors

- Using Kronecker products and the `vec` operator we can perform *multilinear transforms*
 - Tensor example with RGB images:



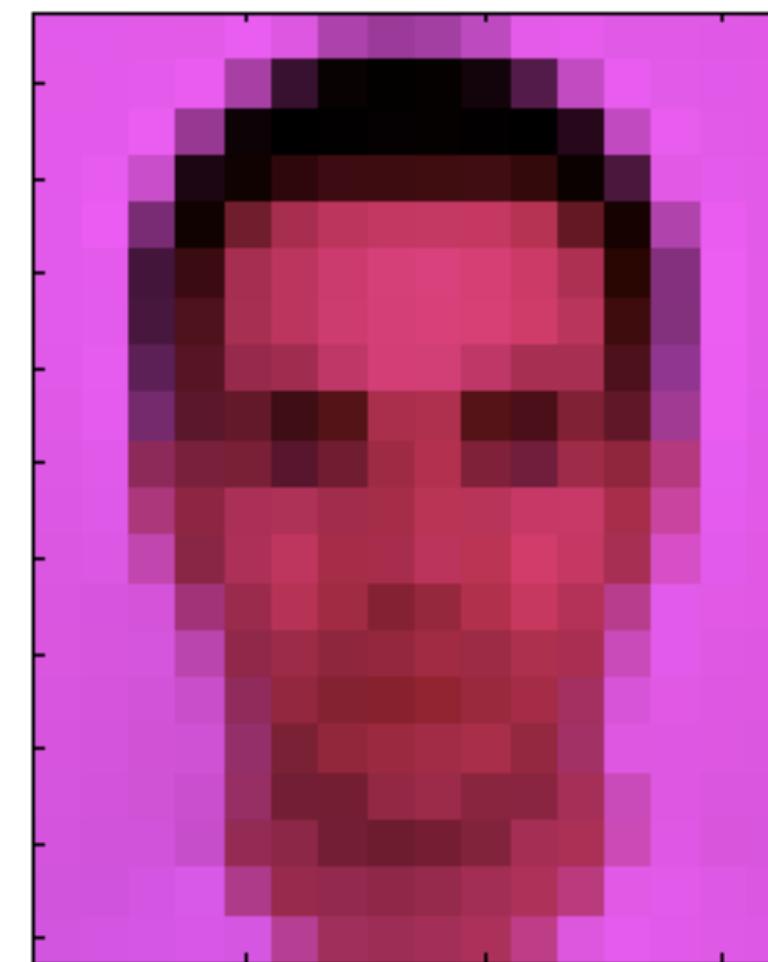
Mixing the colors

- Define color, horizontal, and vertical mixing

$$\left(\underset{3^{rd} \ dim}{\mathbf{C}^T} \otimes \underset{2^{nd} \ dim}{\mathbf{H}^T} \otimes \underset{1^{st} \ dim}{\mathbf{V}^T} \right) \cdot \text{vec}(\mathbf{X})$$

- Example: color mixing

$$\left(\text{diag} \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \otimes \mathbf{I} \otimes \mathbf{I} \right) \text{vec}(\mathbf{X}) =$$



- Caution: Dimensions will quickly get out of hand this way

Special matrices

- Symmetric:

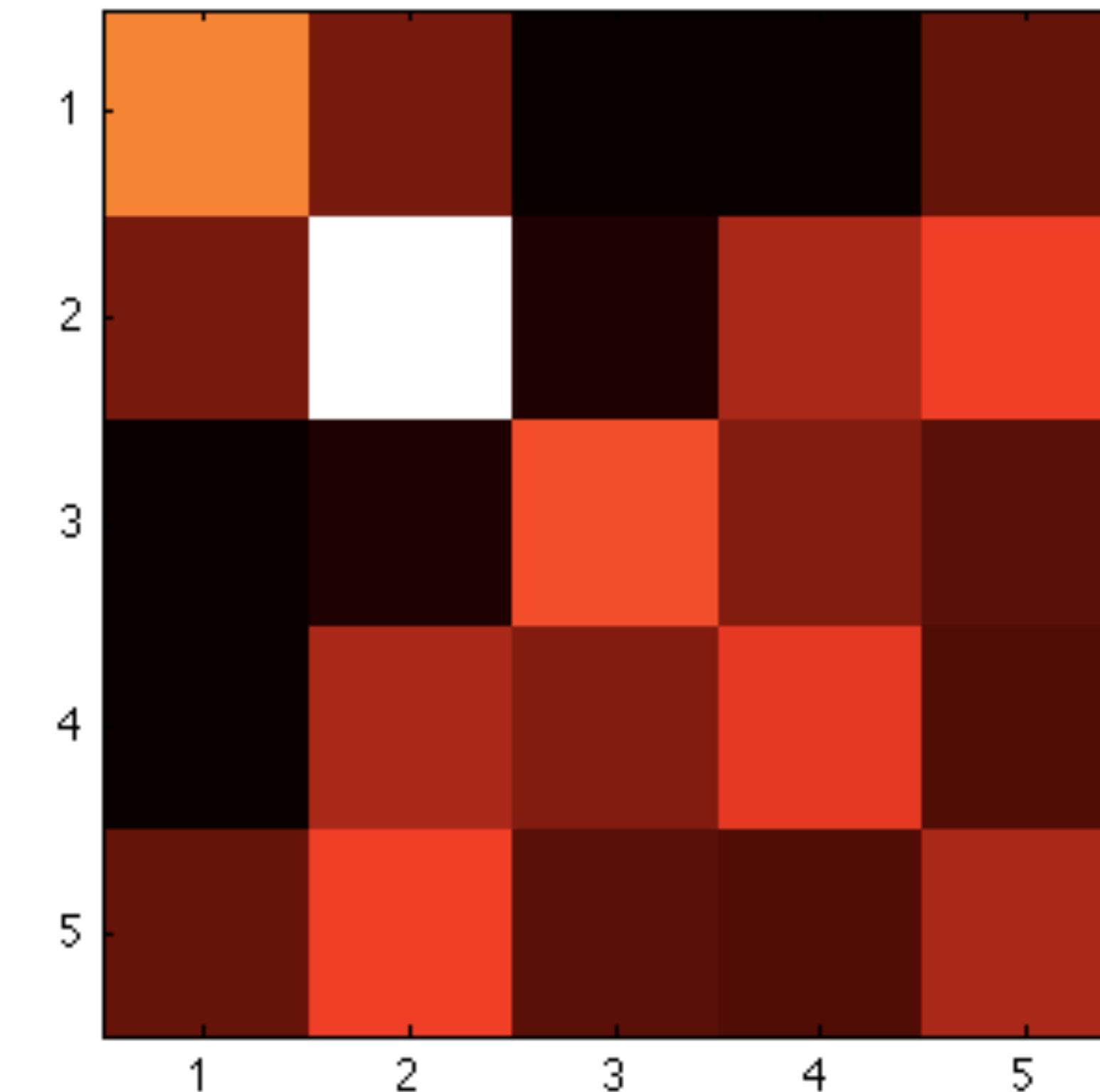
$$\mathbf{X} = \mathbf{X}^\top \Rightarrow x_{ij} = x_{ji}$$

- Positive definite

- If: $\mathbf{y}^\top \cdot \mathbf{X} \cdot \mathbf{y} > 0, \forall \mathbf{y}$
- Also symmetric

- Orthonormal:

$$\mathbf{X}^\top \cdot \mathbf{X} = \mathbf{X} \cdot \mathbf{X}^\top = \mathbf{I}$$



Matrix inverse

- “Undoes” a matrix multiplication
 - Only for square matrices
 - Not all matrices are invertible
 - must be a full-rank matrix
 - Remember, in matrix multiplication order matters
- $$\mathbf{X}^{-1} \cdot \mathbf{X} = \mathbf{I}$$
- $$\mathbf{X}^{-1} \cdot \mathbf{X} \cdot \mathbf{Y} = \mathbf{Y}$$
- $$\mathbf{Y} \cdot \mathbf{X} \cdot \mathbf{X}^{-1} = \mathbf{Y}$$
- $$\mathbf{X} \cdot \mathbf{Y} \cdot \mathbf{X}^{-1} \neq \mathbf{Y}$$

Matrix pseudoinverse

- Also known as Moore-Penrose (or MP) pseudoinverse
 - For an $m \times n$ matrix \mathbf{X} , pseudoinverse is $n \times m$ matrix \mathbf{X}^+

$$\mathbf{X} \cdot \mathbf{X}^+ \cdot \mathbf{X} = \mathbf{X}$$

$$\mathbf{X}^+ \cdot \mathbf{X} \cdot \mathbf{X}^+ = \mathbf{X}^+$$

$$(\mathbf{X} \cdot \mathbf{X}^+)^T = \mathbf{X} \cdot \mathbf{X}^+$$

$$(\mathbf{X}^+ \cdot \mathbf{X})^T = \mathbf{X}^+ \cdot \mathbf{X}$$

- We'll be seeing this operation a lot, it's essentially least squares

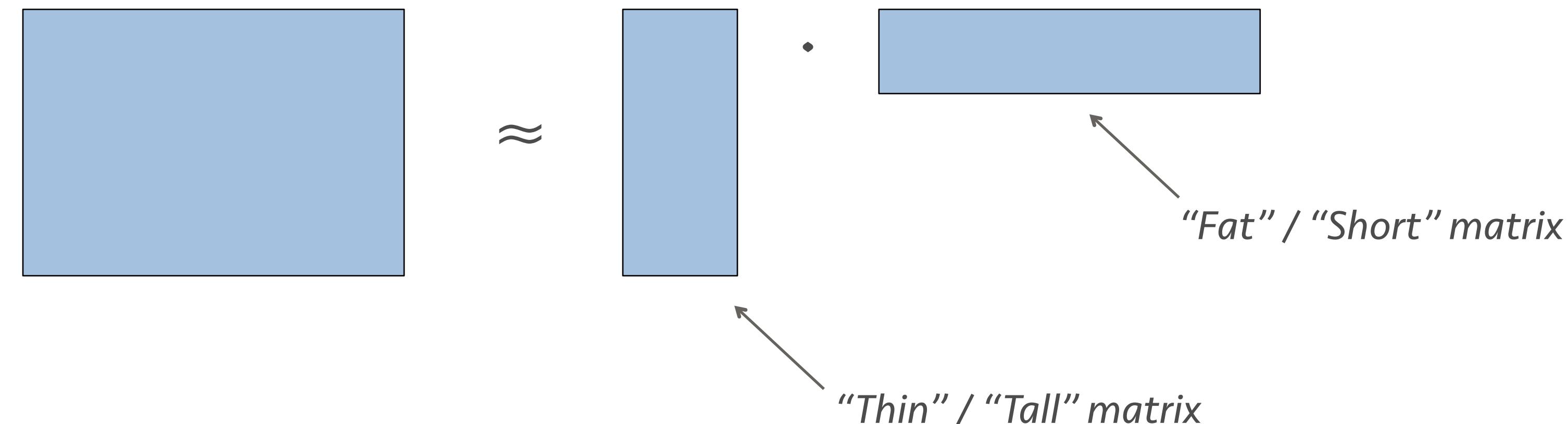
$$\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \rightarrow \mathbf{x} = \mathbf{A}^+ \cdot \mathbf{y}$$

Low rank approximations

- Use smaller matrices to describe a large matrix

$$\mathbf{Y} \approx \mathbf{A} \cdot \mathbf{X}$$

- With \mathbf{Y} being $m \times n$, \mathbf{A} being $m \times r$, \mathbf{X} being $r \times n$, and $r < m$



Eigenanalysis

- Eigenvectors and eigenvalues

- For an $n \times n$ matrix \mathbf{X}

$$\mathbf{X} \cdot \mathbf{V} = \mathbf{V} \cdot \mathbf{L}$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{bmatrix}$$

$$\mathbf{L} = \text{diag} \begin{bmatrix} \lambda_1 & \cdots & \lambda_n \end{bmatrix}$$

- \mathbf{V} is $n \times n$ and contains the eigenvectors \mathbf{v}_i
 - It will be an orthogonal matrix for positive (semi-)definite matrix inputs
 - \mathbf{L} is $n \times n$ contains the eigenvalues λ_i

The Singular Value Decomposition (SVD)

- Similar decomposition to eigenanalysis

- For a matrix \mathbf{X}

$$\mathbf{U}^\top \cdot \mathbf{U} = \mathbf{I}$$

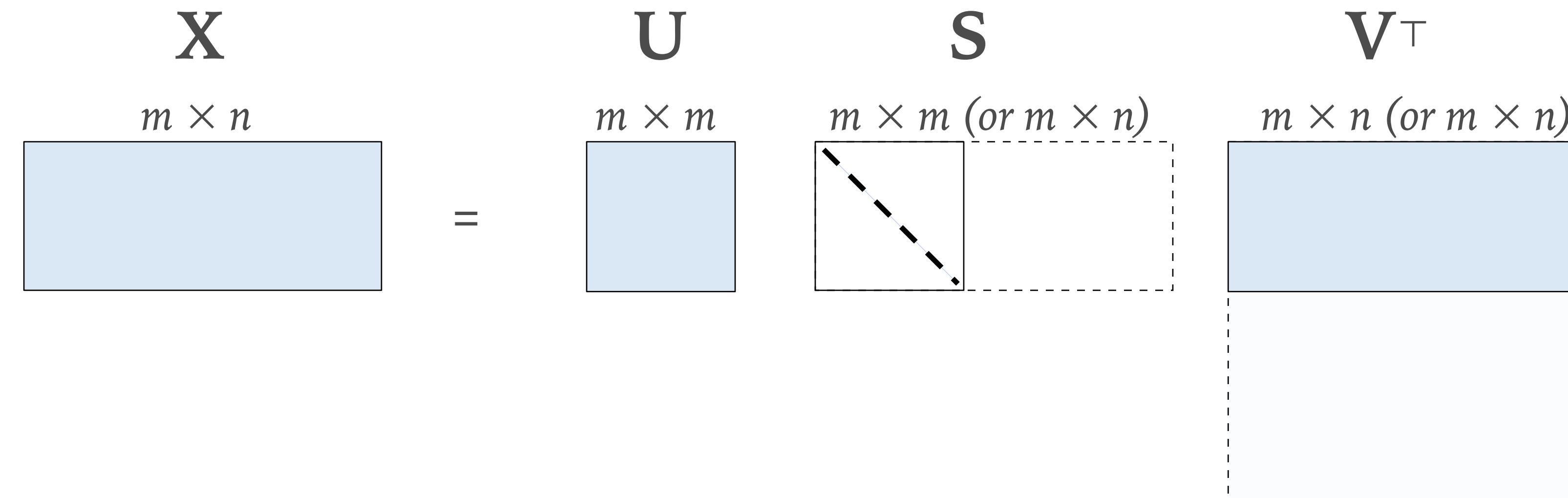
$$\mathbf{X} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^\top \quad \text{where} \quad \mathbf{V}^\top \cdot \mathbf{V} = \mathbf{I}$$

$$\mathbf{S} = \text{diag}(\sigma_i)$$

- \mathbf{U}, \mathbf{V} are orthonormal
 - Contain the left and right singular vectors of \mathbf{X}
 - \mathbf{S} is diagonal
 - Contains on its diagonal the singular values σ_i

Visualizing the SVD

- Comes in two versions, *full* and *economy*
 - The only difference is the size of the matrices
 - The numerical approximation is the same
 - By truncating the columns of \mathbf{S} we can make a low-rank approx.



Recap

- What's this class about?
 - Signals, learning, fun, etc ...
- Linear algebra basics
 - Algebraic operations, norms, decompositions, linear transforms, form manipulations, tensors, special matrices, ...

What now?

- Skim through this material for now
 - We'll be seeing it in context soon
 - e.g., what are the eigenvectors of image matrices?
- Reading material
 - Old and new algebra useful for statistics
 - <http://research.microsoft.com/en-us/um/people/minka/papers/matrix/>
 - The Matrix Cookbook
 - http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3274
 - Best book ever on matrix calculus (use school's library to get PDF):
 - <https://onlinelibrary.wiley.com/doi/book/10.1002/9781119541219>

Questions?

- Slides are online on our piazza page
- Lecture recording will soon be up as well

Next lecture

- Review of:
 - Probability theory
 - Bayes theorem, probability rules, Bayes nets
 - Basic distributions, transformations of RV's
 - Statistics
 - Basic measures, independence, information
 - Parameter estimation intro
 - Maximum likelihood, MAP, Bayesian, EM intro