CSSE463: Image Recognition Day 20

Project Teams:

Beating Captchas:	David H, Dan E, Joe F, Matt B
Whiteboard Scribbles:	Gunnar H, Andrew T, Sean C, Noah M
???:	Hazen H, Alex T, Donnie W, Andy Y
Yelp Restaurant:	Nathan C, Faye L, Alex L, Addison W
SubwayCam:	Chris B, Jonathan J, Kassandra S, Andy M
OCR with OpenCV:	Misato M, Bo P, Min S, Zhihao X
Pokemon Type Rec:	Sam B, Tai E, Orion M, Josh M

Term project next steps

- From the **Lit Review specification**.
  - Goal: Review what others have done. Don't re-invent the wheel.
  - Read papers!
  - Summarize papers
  - Due next Friday (Extensions available on request)

CSSE463: Image Recognition Day 20

- Today: Lab for sunset detector

- Next week:
  - Monday: Midterm Exam
  - Tuesday: sunset detector lab (due Weds. 11:00 pm)
  - Thursday: k-means clustering
  - Friday: lab 6 (k-means)

Exam prep

- Bright blue roadmap sheet
- Exam review slides (courtesy reminder)

Last words on neural nets/SVM

How does svmfwd compute y1?

y1 is just the weighted sum of contributions of individual support vectors:
d = data dimension, e.g., 294, σ = kernel width.

$$y1 = \sum_{i=1}^{numSupVecs} \left(svcoeff_i * e^{(-1/d\sigma)*\|x - sv_i\|^2} \right) + bias$$

numSupVecs, svcoeff (alpha) and bias are learned during training.
Note: looking at which of your training examples are support vectors can be revealing! (Keep in mind for sunset detector and term project)

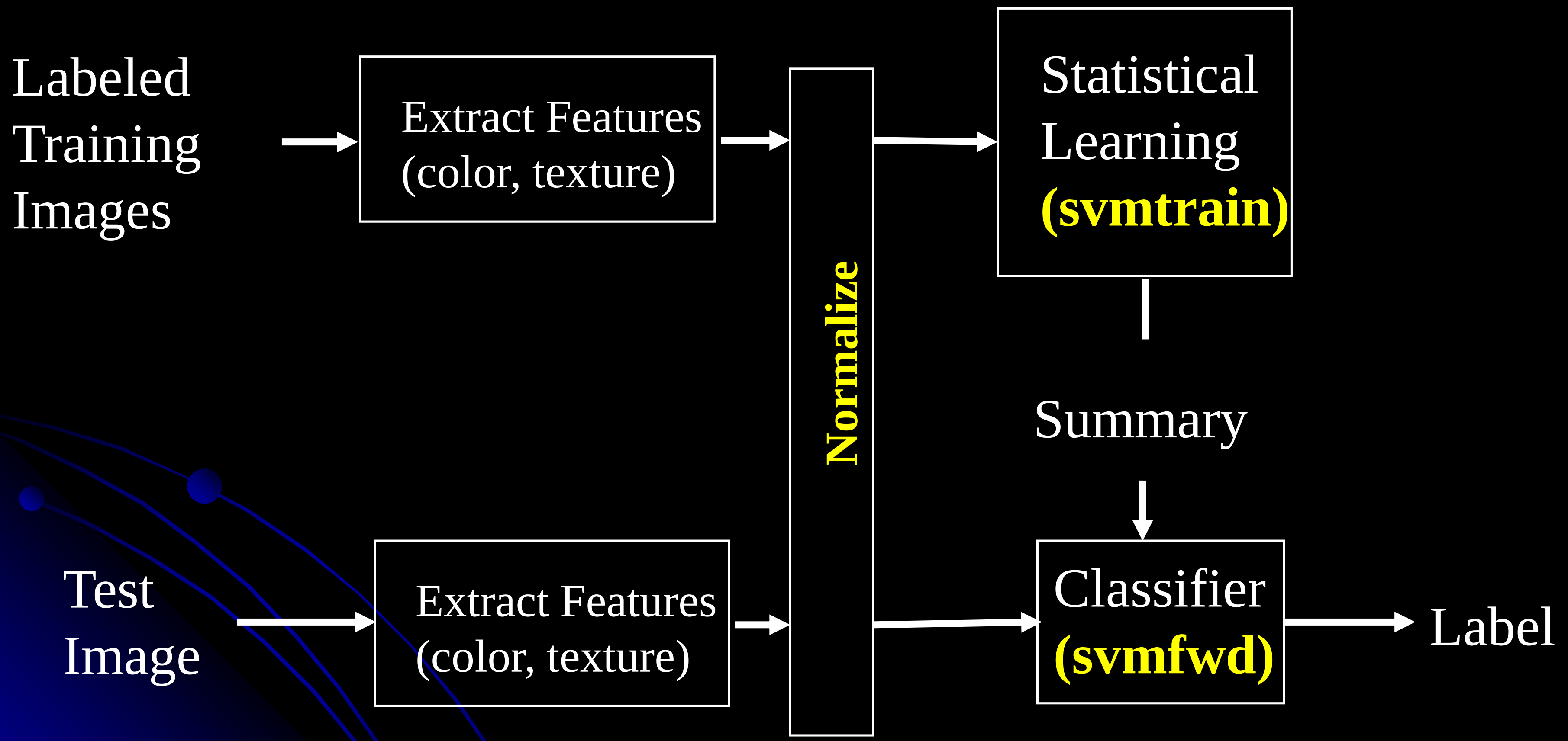
- Much easier computation than training
- Was easy to implement on a device without MATLAB (ea smartphone)

SVMs vs. Neural Nets

- SVM:
  - Training can take a *long* time with large data sets due to choosing parameters…
  - But the classification runtime and space are *O(sd)*, where *s* is the number of support vectors, and d is the dimensionality of the feature vectors.
  - In the worst case, *s* = size of whole training set (like nearest neighbor)
    - Overfitting is occurring: can use with accuracy to choose classifier
  - But no worse than implementing a neural net with *s* perceptrons in the hidden layer.
  - Empirically shown to have good generalizability even with relatively-small training sets and no domain knowledge.
- Neural networks:
  - can tune architecture. Lots of parameters!

Q3 on old SVM quiz

Common model of learning machines



Sunset Process

- Loop over 4 folders of images
  - Extract features
- Normalize
- Split into train and test and label
- Save
- Loop over kernel params
  - Train
  - Test
  - Record accuracy, #sup vec
- For SVM with param giving best accuracy,
  - Generate ROC curve
  - Find good images
  - Do extension
- I suggest writing as you go