Final Project: 1 Slide
Reminder: Final Project

- 1-slide from group in 2 weeks Wednesday October 14:
  - 2/30 of final project grade
- Slide should cover 3 aspects
  1. **Problem you intend to work on either:**
     - App that helps WPI students cope with COVID
     - Points awarded for difficulty, components used (location, sensor, camera, ML)
     - If games, must gamify solution to real world problem
  2. **Why this problem is important**
     - E.g. WPI students can’t find places to study during COVID
  3. **Summary of envisioned mobile app (?) solution**
     1. E.g. Mobile app tracks availability of study spaces. WPI students can check using app
- You can:
  - Bounce ideas of me (email, or in person)
  - Change idea any time
Final Project: Difficulty Score

- Project execution, presentation, paper: 80%
- Project difficulty score: 20%
- Mobile Components and Android UI (4 points each)
  - Every 5 Android screens (A maximum of 8 points can be earned for the UI)
  - Playback audio/video
  - Maps, location sensing
  - Camera: simply taking pictures
- Ubiquitous Computing Components & Android UI (6 points each)
  - Activity Recognition, sensor programming, step counting
  - GeoFencing, Mobile Vision API: e.g. Face/barcode detection/tracking
- Machine/Deep Learning (10 points each)
  - Machine/deep learning (i.e. run study, gather data or use existing dataset to classify/detect something)
  - Program Android, machine learning/deep learning components
Reminder: Detection vs Recognition

Ref: Adriana Kovashka, Introduction to Vision (CS 1674), U. of Pittsburgh, fall 2016

- Detection: Find (draw squares around) all faces in image
- Recognition: Whose face is in the image
Uses of Face Detection

- Detect Faces in surveillance images
- Cameras use it to detect faces in a picture
- Facebook: detect faces in an image, recognize them
Types of Face Detection Algorithms


- Knowledge-based: Rules based on human knowledge. E.g. relative positions and distances of parts (nose, eyes, mouth, etc)

- Feature-based: Extract visual attributes (color, texture, shape), train machine learning classifier to distinguish facial vs. non-facial regions

- Template: Divide face into parts (e.g. mouth, eyes), compare/match parts to templates (or standard face patterns)

- Appearance-based: Generates templates from representative set of faces
Face Detection: Voila Jones Algorithm

- Most popular method, published in CVPR 2001 (top computer vision conference)
- Windows based: Draw candidate windows, decides if it contains face or not
- First fast, real time, still used today
- Challenge?
  - Lots of pixels in image (millions)
  - But faces are rare (0 – 10 per image)
  - Window evaluation must be computationally efficient, fast
  - But also low false positive rate ( < 1 in 1 million)
  - False Positive: Say it’s a face but it’s not
Voila Jones: 4 Stages

1. Haar Feature Selection
2. Creating an Integral Image
3. Adaboost Training
4. Cascading Classifiers
Step 1: Haar Feature Selection

- Features: rectangular patterns
- Calculate sums of pixel values within rectangles
- Example:
  - Overlay feature A over image
  - Calculate (sum of pixels in white rectangle) – (sum of pixels in gray rectangle)
- So what?
  - All faces have common properties
  - E.g.
    - Eye region is darker than upper cheeks
    - Bridge of nose is brighter than eyes
Step 2: Integral Image

- Efficient way to calculate sum of pixel values within rectangles
  - Calculate (sum of pixels in white rectangle) – (sum of pixels in gray rectangle)

![Integral image diagram]
Step 3: AdaBoost Classifier

- Want to overlay various scales, and positions of the 4 feature types over the face
  - Determine which specific shapes, positions, scales are discriminative
- Lots of permutations and combinations: E.g. 180,000 features per 24 x 24 pixel window
- Use an AdaBoost classifier (Machine Learning)
Boosting: Training

Initially, weight each training example equally

In each boosting round:

a) Find the weak learner that achieves the lowest weighted training error
   (Find dividing line that has lowest total distance from all the points in the training set)
Boosting: Training

Ref: Adriana Kovashka, Introduction to Vision (CS 1674), U. of Pittsburgh, fall 2016

- In each boosting round:
  a) Find the weak learner that achieves the lowest weighted training error
  b) Increase weights of training examples misclassified by current weak learner
     (Increase weights of examples on wrong side of dividing line)
Boosting: Training

Ref: Adriana Kovashka, Introduction to Vision (CS 1674), U. of Pittsburgh, fall 2016

● Round 2: Repeat a) and b) again to find weak classifier 2
  a) Find the weak learner that achieves the lowest weighted training error
     (Note: weighted examples results in different dividing line)
**Boosting: Training**

Ref: Adriana Kovashka, Introduction to Vision (CS 1674), U. of Pittsburgh, fall 2016

- Round 2: Repeat a) and b) again to find weak classifier 2
  a) Find the weak learner that achieves the lowest weighted training error
  b) Increase weights of training examples misclassified by current weak learner
     (Increase weights of examples on wrong side of dividing line)
Round 3: Repeat a) and b) again to find weak classifier 3
Final classifier is combination of weak classifiers.

i.e. for each point to be classified, test against:

- Classifier 1
- Classifier 2
- Classifier 3... etc

Different algorithms, formulas for re-weighting and combining weak learners

- E.g. AdaBoost
Step 3: AdaBoost Classifier

- Examine windows within an image
- What to determine if there’s a face in that window
- Overlay 4 feature patterns over various parts of the window
- Find which of 4 feature patterns + locations that indicate that window contains face
  - Patterns + position + scale that discriminate face vs. non-face used as weak classifiers
  - Use Boosted combination of multiple weak classifiers as final classifier
  - Form cascade of weak classifiers, reject negatives quickly
Step 4: Cascade of Classifiers

- Form cascade: Use weak classifiers one after the other
- Reject non-faces quickly by placing features with low false negative rates early on

Voila Jones Algorithm: Summary

- Training: slow
- Test: fast
  - Integral image
  - Cascade of classifiers

Train with 5K positives, 350M negatives
Real-time detector using 38 layer cascade (0.067s)
Voila Jones: Results
Face Recognition: Uses

- Device security:
  - Recognize owner’s face, use as their device password

- Identify genetic disorders:
  - Analyze, compare faces to databases of people with various disorders
  - E.g. Face2gene app, DeepGestalt software

- Prevent shoplifting
  - Recognize past shoplifters, notify owner if they visit

- Check underage alcohol/tobacco buyers
  - Automatically assess buyers age from their face

- Security in schools
  - Recognize criminals, sex offenders

- Airline ticketing:
  - Use your face as your ticket
Android Face Recognition/Face Unlocking

- Mostly used for security, recognize owners face in place of password
  - Not definitely more secure but faster, more convenient (no forgotten passwords)
- Basic version compares stored picture with picture captured by front-facing camera
- Thief can fool system by presenting a picture of owner
- Available on Pixel 4, well executed
  - Cannot be fooled by similar people, photos
Face Recognition: Triplet Loss

- Uses neural networks a lot these days
  - Ref: Schroff et al, 2015, FaceNet: A Unified embedding for face recognition and clustering

- Given:
  - Anchor: reference image
  - Positive: Positive example
  - Negative: Negative example

- Defines loss (objective function) that learns encoding (0101000...) such that:
  - Anchor and positive close together
  - Anchor and negative far apart
Triplet Loss: Training Set

- Millions of sets of 3 (Anchor, Positive, Negative)
- Neural network learns encoding and to distinguish faces

Training set using triplet loss

Anchor  Positive  Negative
Android Networking
Android Networking Alternatives

- Android provides various alternative methods of network connection
  - Sockets: low level, multi-purpose
  - URLConnection: Higher level, over the web
  - HttpURLConnection: Web, uses the HTTP protocol
References

- Android Nerd Ranch, 1st edition
- Busy Coder’s guide to Android version 4.4
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014
- Adriana Kovashka, Introduction to Vision (CS 1674), U. of Pittsburgh, fall 2016
- Divyansh Dwivedi, Face Detection For Beginners, https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9
References

- Network programming with Android, https://slideplayer.com/slide/8471042/
- Convolutional Neural Networks Course, Deeplearning.ai, Coursera