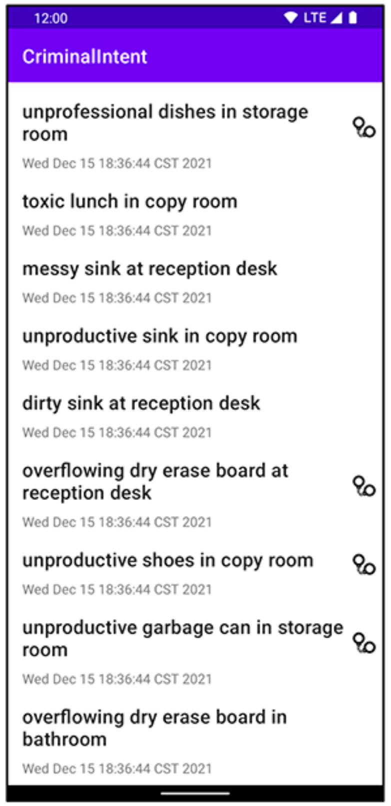
CS 528: Mobile and Ubiquitous Computing
Lecture 5b: Facial Analysis: How it works

Emmanuel Agu

Criminal Intent
ANR Skipped Chapters

Chapter 13: Fragment Navigation

- Fragment navigation used to get list and details parts of **CriminalIntent** working together
- Uses Navigation jetpack library
- When user presses item in list of crimes:
  - Navigation library swaps out **CrimeListFragment**
  - Replaces it with new instance of **CrimeDetailFragment**



12:00 LTE

CriminalIntent

unprofessional dishes in storage room
Wed Dec 15 18:36:44 CST 2021

toxic lunch in copy room
Wed Dec 15 18:36:44 CST 2021

messy sink at reception desk
Wed Dec 15 18:36:44 CST 2021

unproductive sink in copy room
Wed Dec 15 18:36:44 CST 2021

dirty sink at reception desk
Wed Dec 15 18:36:44 CST 2021

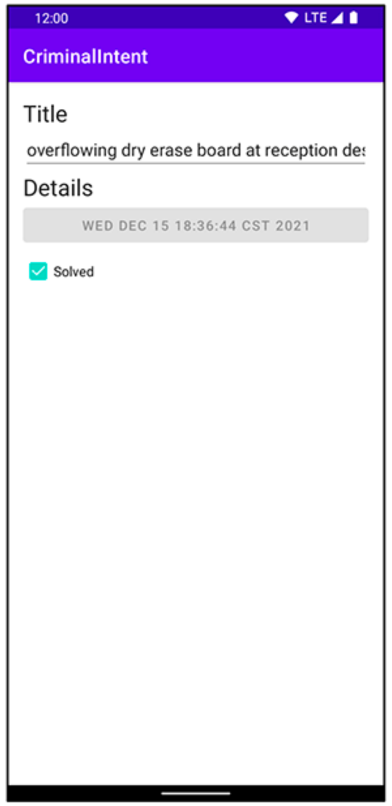
overflowing dry erase board at reception desk
Wed Dec 15 18:36:44 CST 2021

unproductive shoes in copy room
Wed Dec 15 18:36:44 CST 2021

unproductive garbage can in storage room
Wed Dec 15 18:36:44 CST 2021

overflowing dry erase board in bathroom
Wed Dec 15 18:36:44 CST 2021

Press Crime



12:00 LTE

CriminalIntent

Title

overflowing dry erase board at reception des

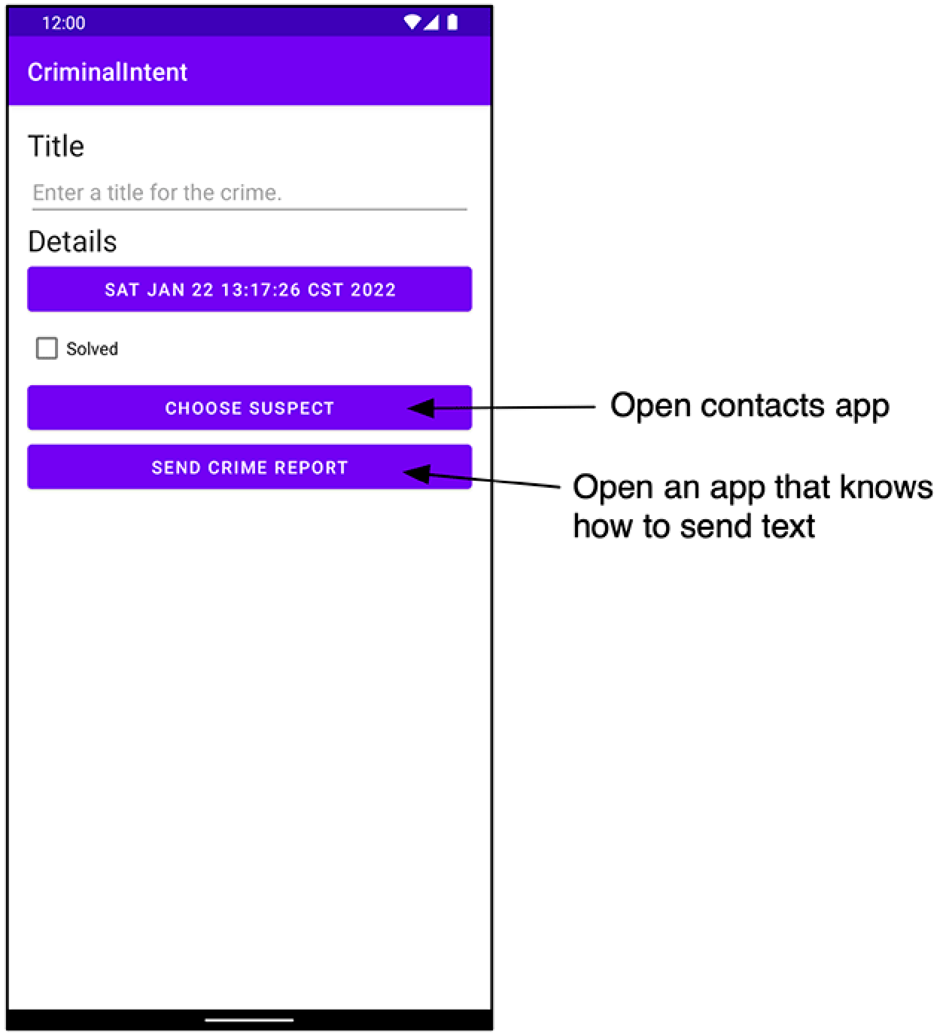
Details

WED DEC 15 18:36:44 CST 2021

✓ Solved

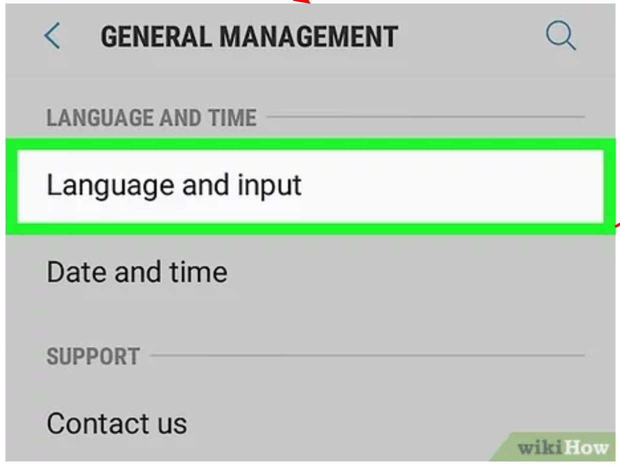
Chapter 16: Implicit Intents

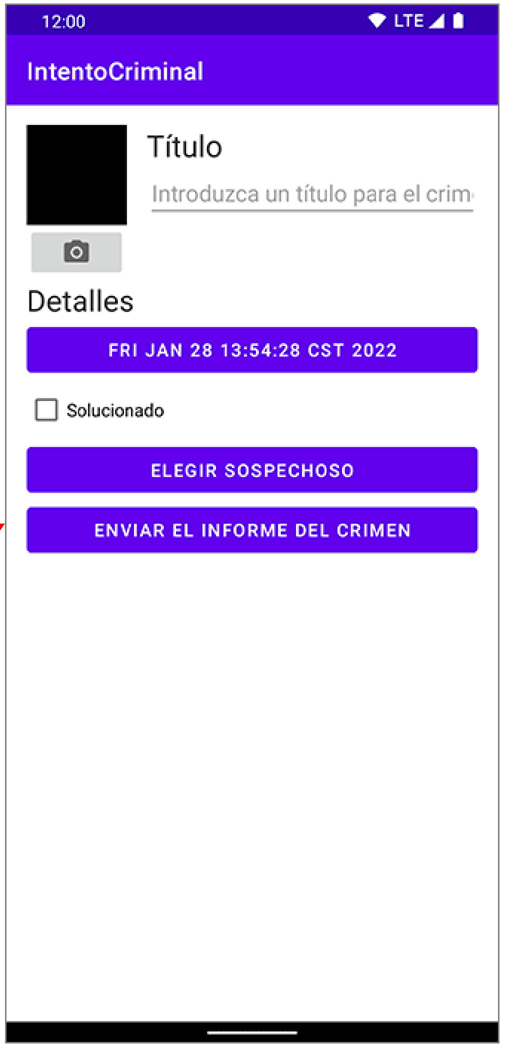
- In **CriminalIntent**:
  - Enable picking a suspect for a crime from user’s contact list
  - Use implicit intent to send a text-based report of a crime


12:00
CriminalIntent
Title
Enter a title for the crime.
Details
SAT JAN 22 13:17:26 CST 2022
Solved
CHOOSE SUSPECT
SEND CRIME REPORT
Open contacts app
Open an app that knows how to send text

Chapter 18: Localization

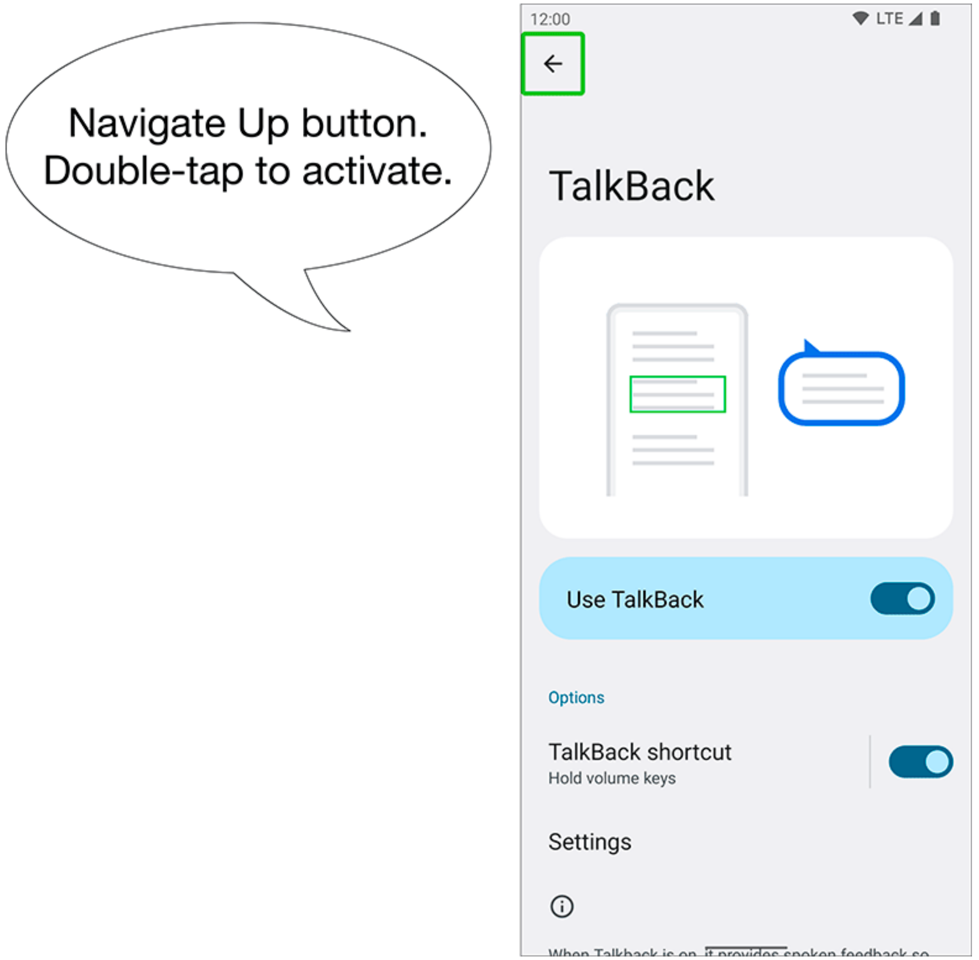
- **Localization:** provide app's resources in various languages
- In **CriminalIntent,** localize text that user sees so that app can be read in Spanish or English
  - Create a Spanish version of strings.xml
  - When device language is set to Spanish, android runtime uses Spanish strings.




12:00
LTE
IntentoCriminal
Título
Introduzca un título para el crim
Detalles
FRI JAN 28 13:54:28 CST 2022
Solucionado
ELEGIR SOSPECHOSO
ENVIAR EL INFORME DEL CRIMEN

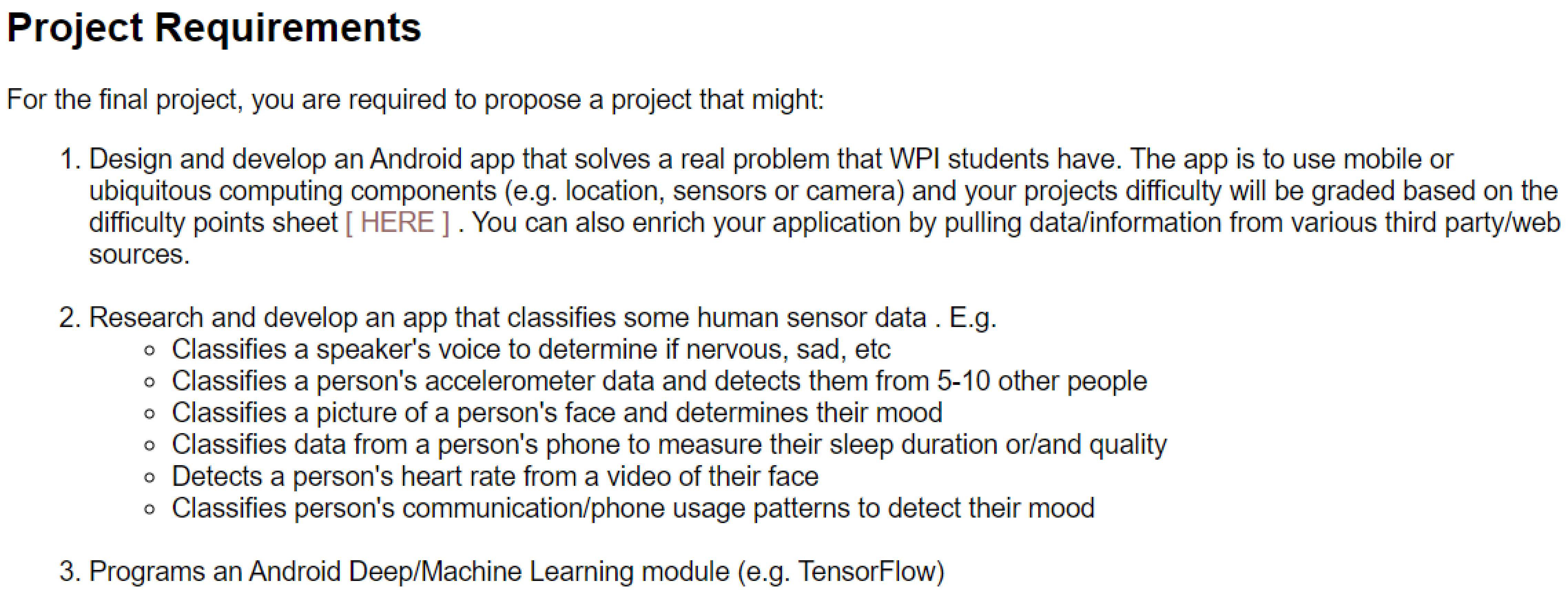
Chapter 19: Accessibility

- **Accessibility:** Makes app usable by people with impaired vision, mobility or hearing
- **Goal:** Make **CriminalIntent** more accessible for people with impaired vision
  - By reading text out loud using Talkback
- **Talkback:** Android screen reader made by Google, speaks out screen’s content depending on what user is doing
- E.g. when app opened and + new crime clicked, Talkback speaks out “New Crime. Double-tap to activate”



Final Project

Final Project: 3 main flavors of projects



Project Requirements

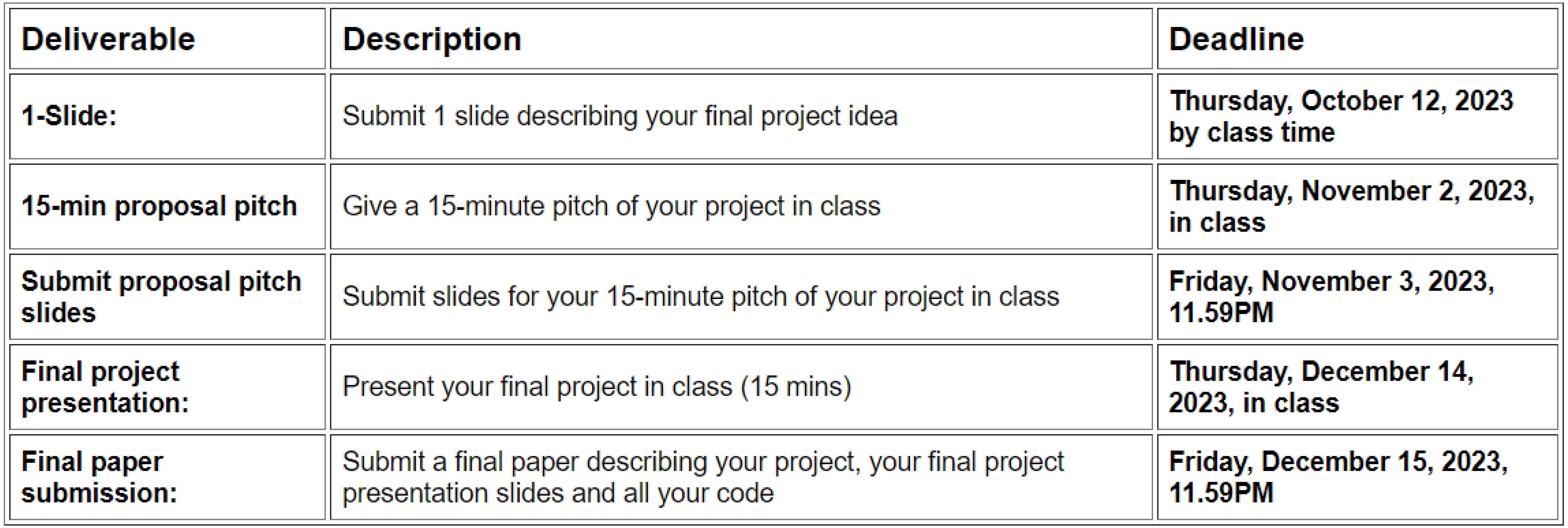
For the final project, you are required to propose a project that might:

1. Design and develop an Android app that solves a real problem that WPI students have. The app is to use mobile or ubiquitous computing components (e.g. location, sensors or camera) and your projects difficulty will be graded based on the difficulty points sheet [HERE] . You can also enrich your application by pulling data/information from various third party/web sources.

2. Research and develop an app that classifies some human sensor data . E.g.
   - Classifies a speaker's voice to determine if nervous, sad, etc
   - Classifies a person's accelerometer data and detects them from 5-10 other people
   - Classifies a picture of a person's face and determines their mood
   - Classifies data from a person's phone to measure their sleep duration or/and quality
   - Detects a person's heart rate from a video of their face
   - Classifies person's communication/phone usage patterns to detect their mood

3. Programs an Android Deep/Machine Learning module (e.g. TensorFlow)

Final Project: Timeline

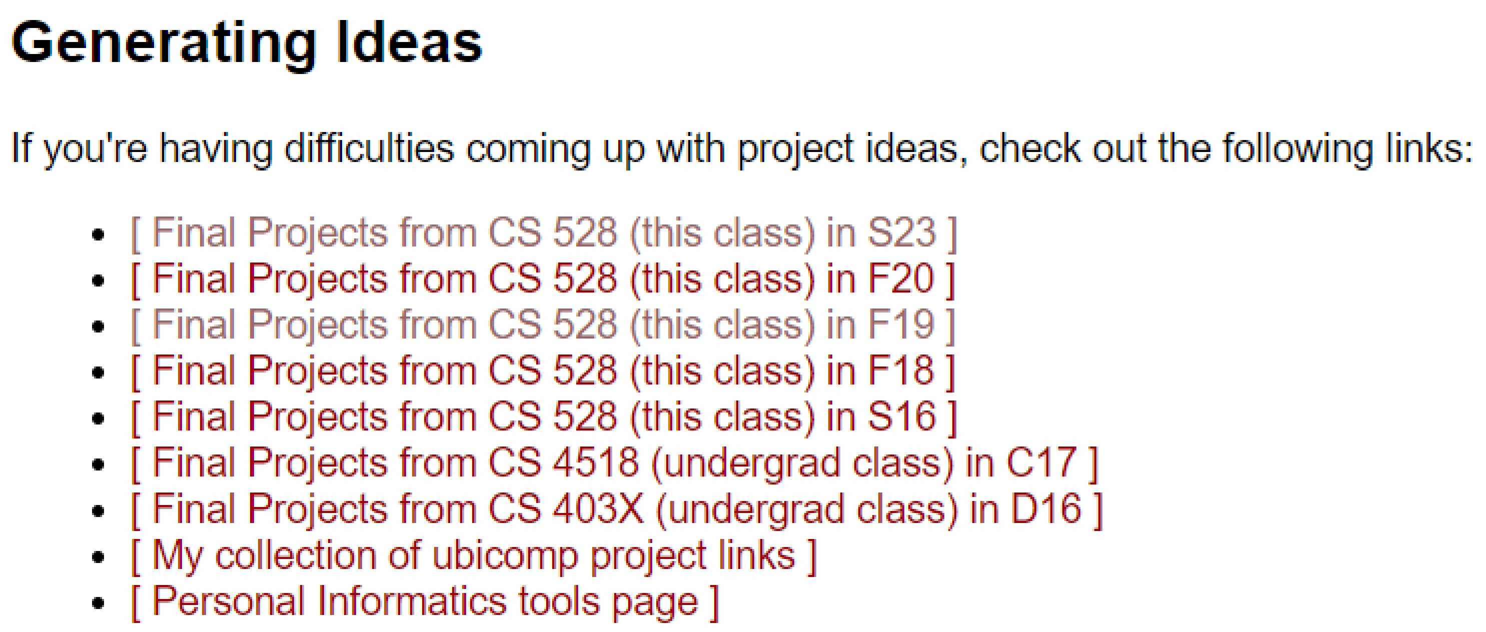


Deliverable	Description	Deadline
1-Slide:	Submit 1 slide describing your final project idea	**Thursday, October 12, 2023 by class time**
15-min proposal pitch	Give a 15-minute pitch of your project in class	**Thursday, November 2, 2023, in class**
Submit proposal pitch slides	Submit slides for your 15-minute pitch of your project in class	**Friday, November 3, 2023, 11.59PM**
Final project presentation:	Present your final project in class (15 mins)	**Thursday, December 14, 2023, in class**
Final paper submission:	Submit a final paper describing your project, your final project presentation slides and all your code	**Friday, December 15, 2023, 11.59PM**

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: taking pictures
- **Ubiquitous Computing Components & Android UI (6 points each)**
  - Activity Recognition, sensor programming, step counting
  - GeoFencing, ML Kit modules: 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

Final Project: Generating Ideas/Past Projects



Generating Ideas

If you're having difficulties coming up with project ideas, check out the following links:

- [Final Projects from CS 528 (this class) in S23]
- [Final Projects from CS 528 (this class) in F20]
- [Final Projects from CS 528 (this class) in F19]
- [Final Projects from CS 528 (this class) in F18]
- [Final Projects from CS 528 (this class) in S16]
- [Final Projects from CS 4518 (undergrad class) in C17]
- [Final Projects from CS 403X (undergrad class) in D16]
- [My collection of ubicomp project links]
- [Personal Informatics tools page]

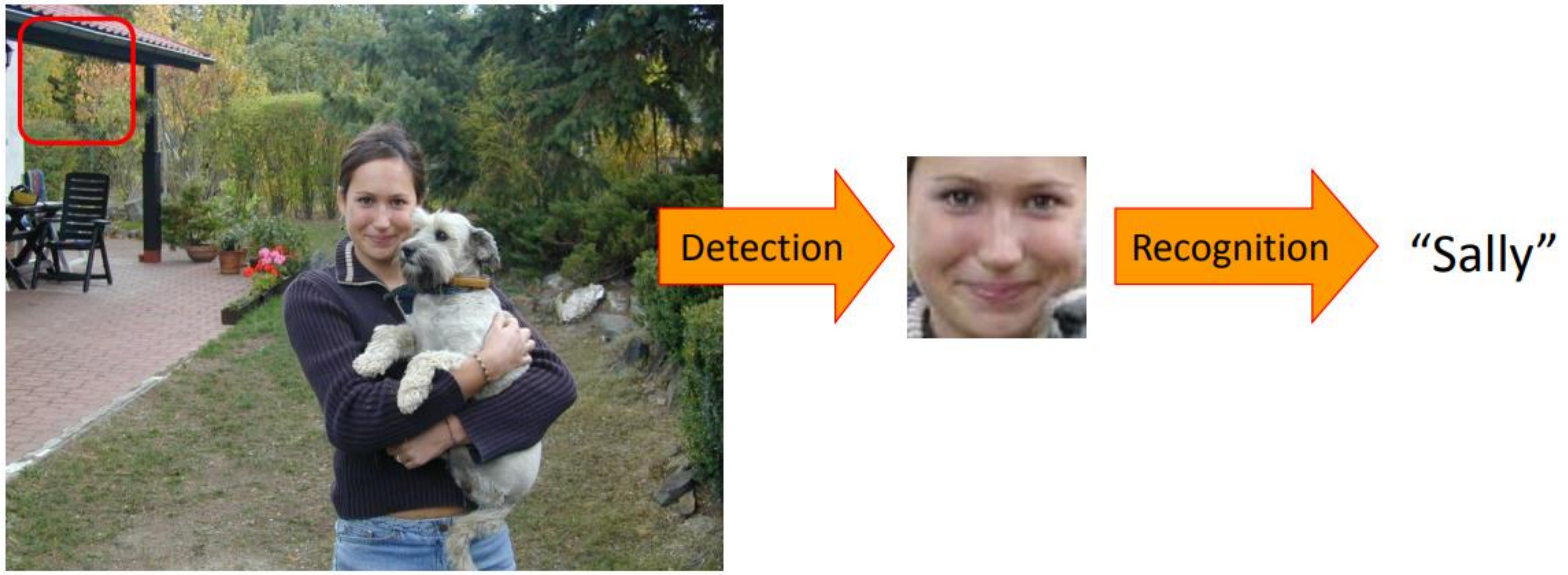
Final Project: 1-slide

- 1-slide from group due on Thursday, October 12
  - 2/30 of final project grade
- Slide should cover 3 aspects
  1. **Problem you intend to work on either:**
     - App that solves a real problem that WPI students face
     - 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 want to find study space close to end of term
  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

Reminder: Detection vs Recognition

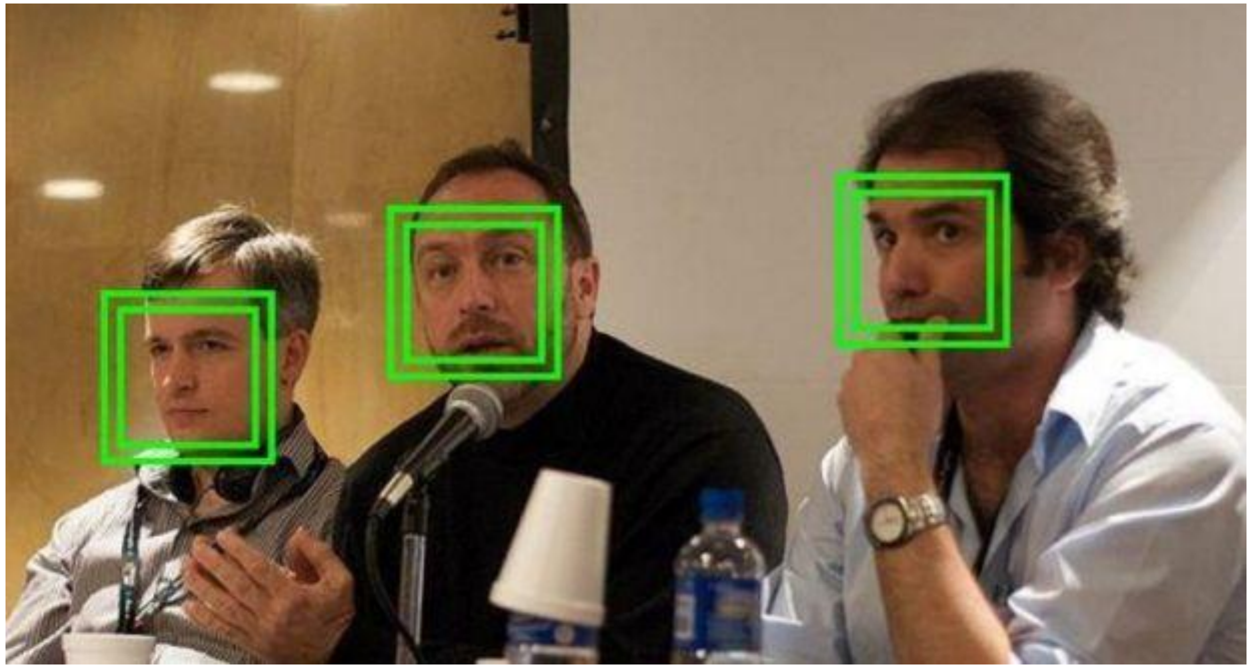
Ref: Adriana Kovashka, Introduction to Vision (CS 1674), U. of Pittsburgh, fall 2016

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



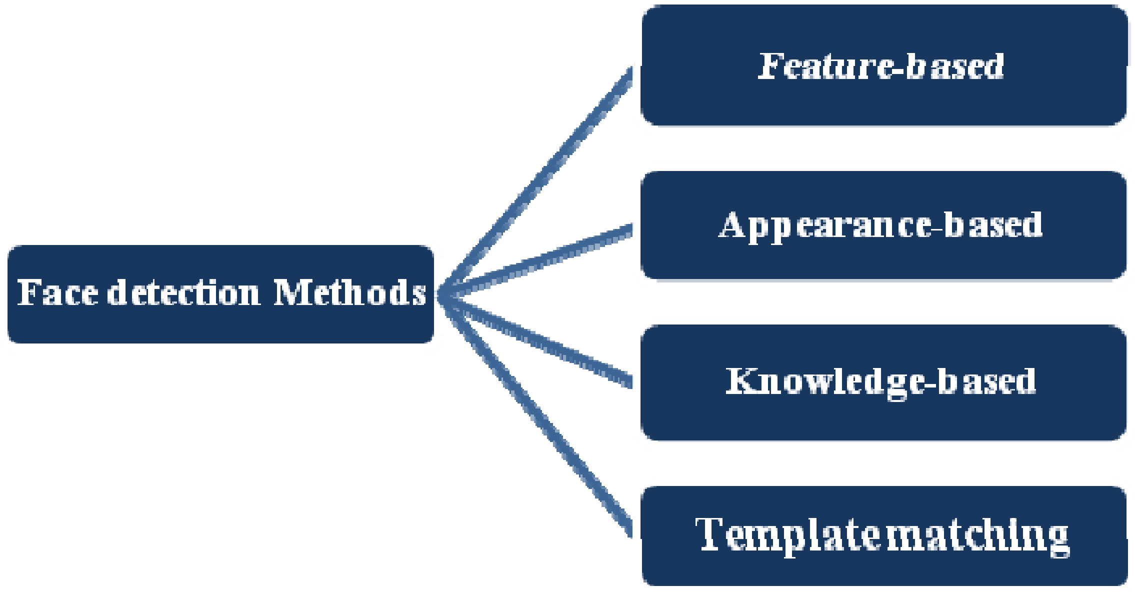
Uses of Face Detection

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



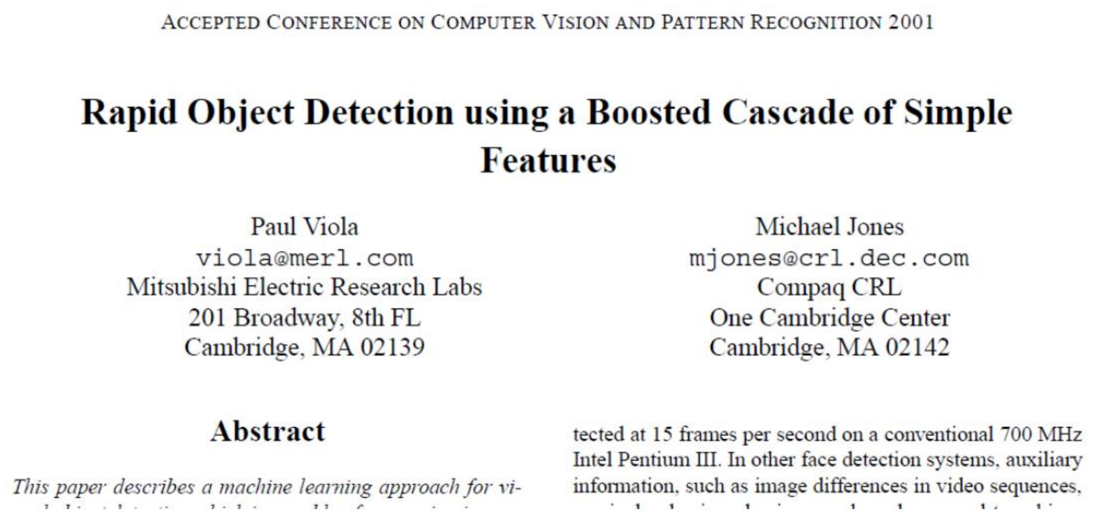
Types of Face Detection Algorithms

- Categorization by Yang, M.H., Kriegman, D.J. and Ahuja, N., 2002. Detecting faces in images: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(1), pp.34-58.
- **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



ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

Michael Jones
mjones@crl.dec.com
Compaq CRL
One Cambridge Center
Cambridge, MA 02142

Abstract

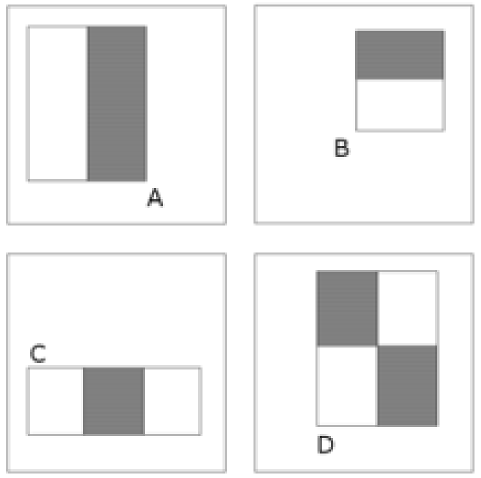
This paper describes a machine learning approach for vi-

tected at 15 frames per second on a conventional 700 MHz Intel Pentium III. In other face detection systems, auxiliary information, such as image differences in video sequences,

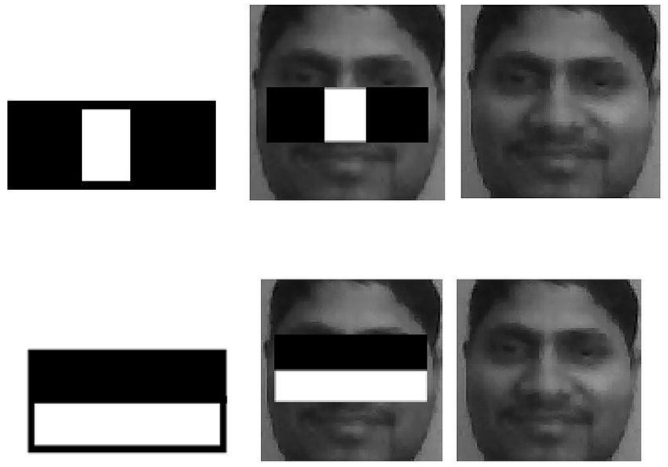
Voila Jones: 4 Stages

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

Step 1: Haar Feature Selection

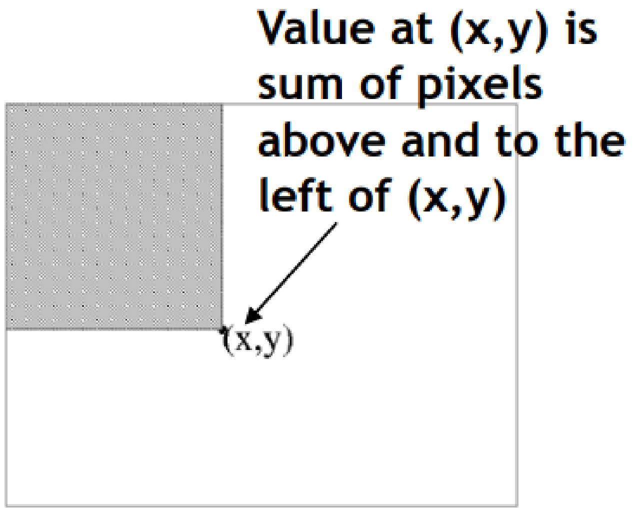


- Features: rectangular patterns
- Calculate sums of pixel values within rectangles
- Example:
  - Overlay feature A over image
  - Calculate (sum of pixels in white rectangle) – (sum of pixels in gray rectangle)
- So what?
  - All faces have common properties
  - E.g.
    - Eye region is darker than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

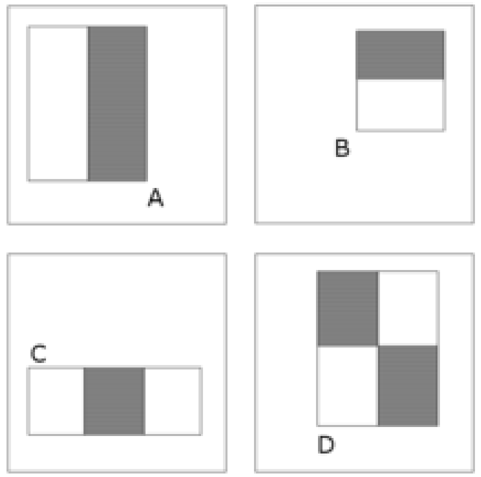


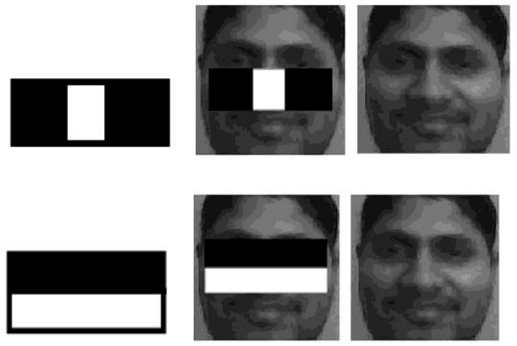
Step 2: Integral Image

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


Value at (x,y) is sum of pixels above and to the left of (x,y)

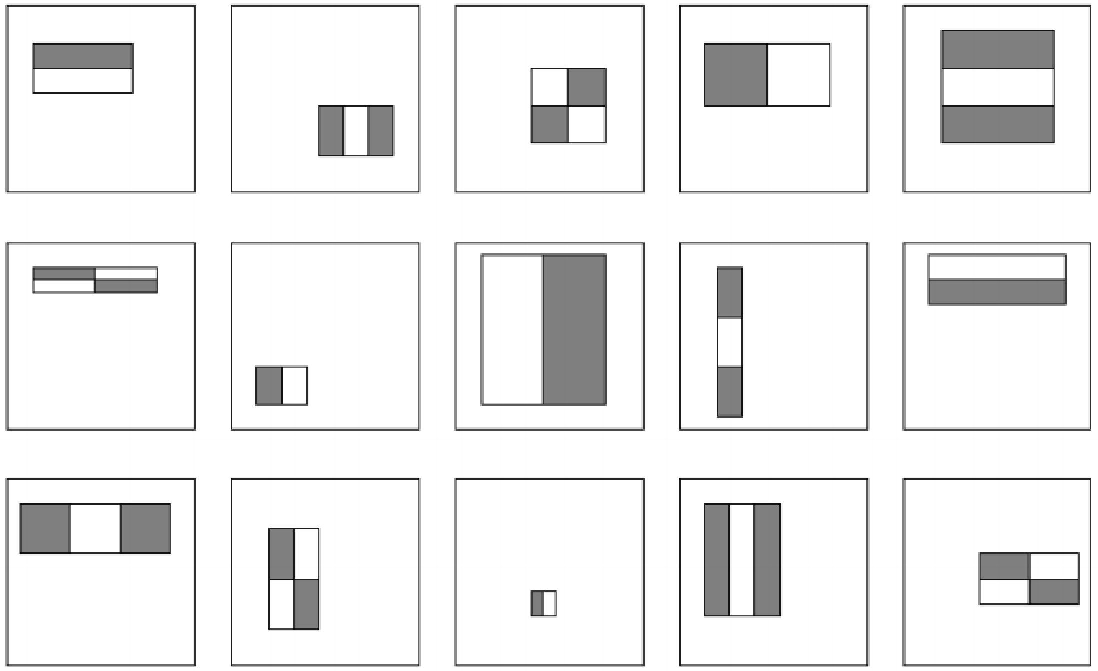
Integral image





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)

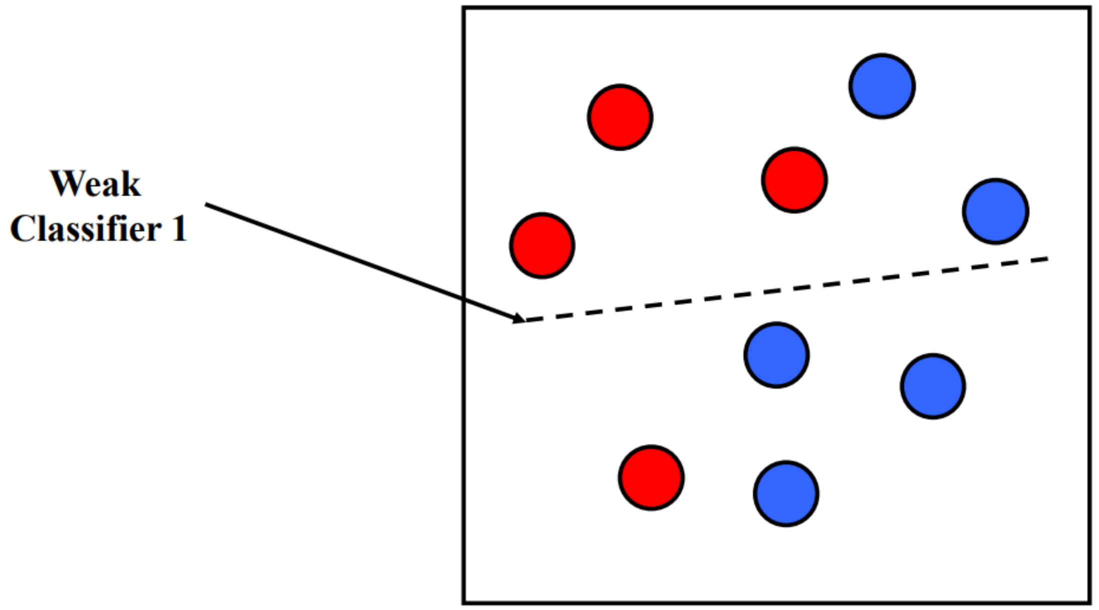


Boosting: Training

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

- Initially, weight each training example equally
- In each boosting round:
  a) Find the weak learner that achieves the lowest weighted training error

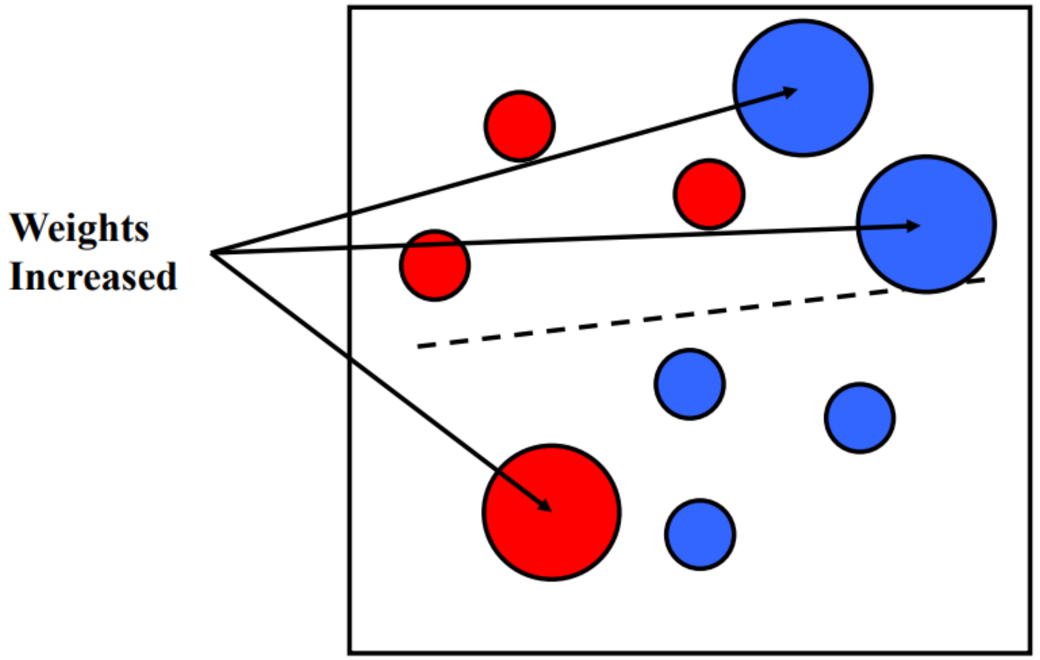
  (Find dividing line that has lowest total distance from all the points in the training set)



Boosting: Training

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

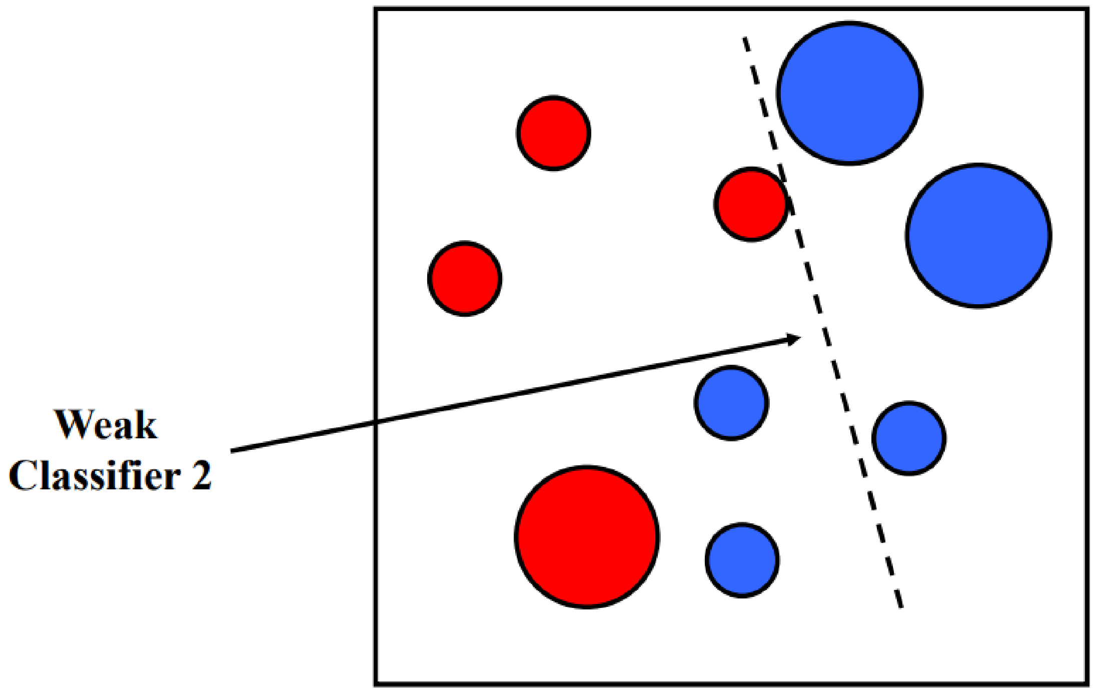
- In each boosting round:
  a) Find the weak learner that achieves the lowest weighted training error
  b) **Increase weights of training examples misclassified by current weak learner (Increase weights of examples on wrong side of dividing line)**



Boosting: Training

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

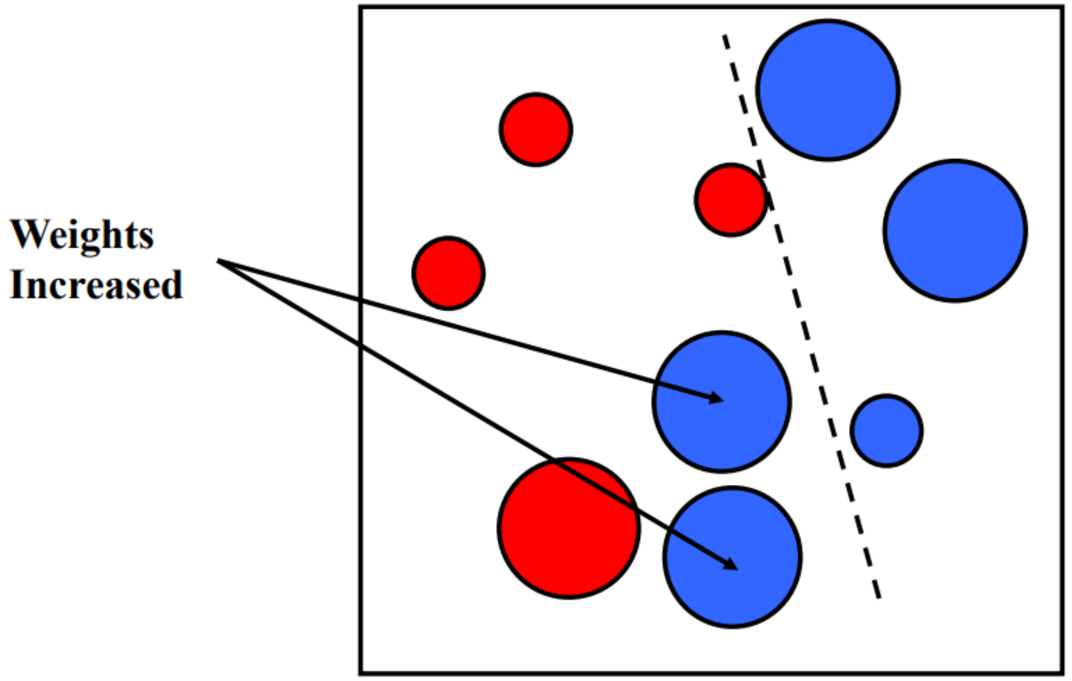
- Round 2: Repeat a) and b) again to find weak classifier 2
  a) **Find the weak learner that achieves the lowest weighted training error**
  (Note: weighted examples results in different dividing line)



Boosting: Training

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

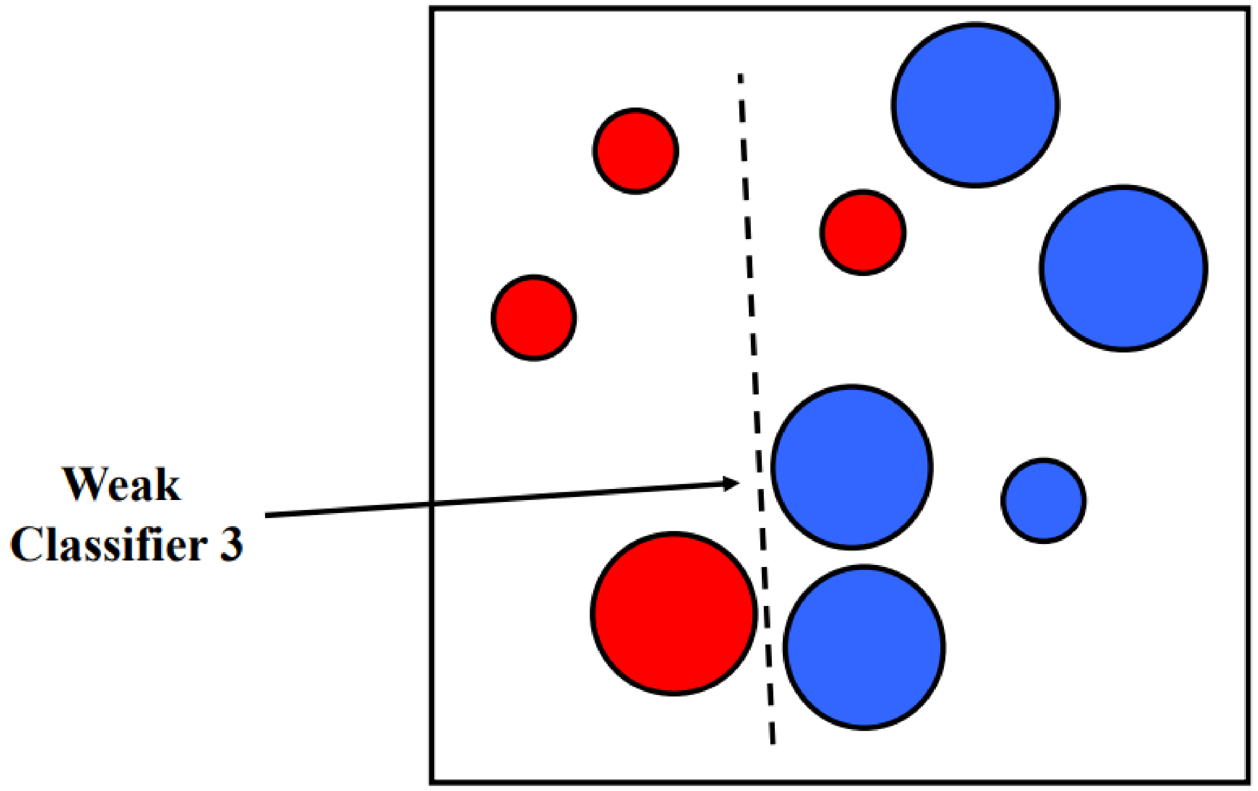
- Round 2: Repeat a) and b) again to find weak classifier 2
  - a) Find the weak learner that achieves the lowest weighted training error
  - b) **Increase weights of training examples misclassified by current weak learner (Increase weights of examples on wrong side of dividing line)**



Boosting: Training

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

- Round 3: Repeat a) and b) again to find weak classifier 3

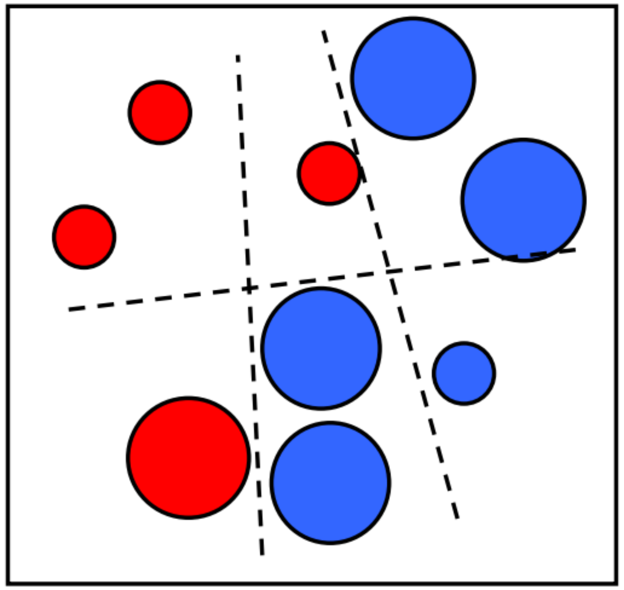


Boosting: Training

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

- Final classifier is combination of weak classifiers.
- i.e. for each point to be classified, test against:
  - Classifier 1
  - Classifier 2
  - Classifier 3… etc

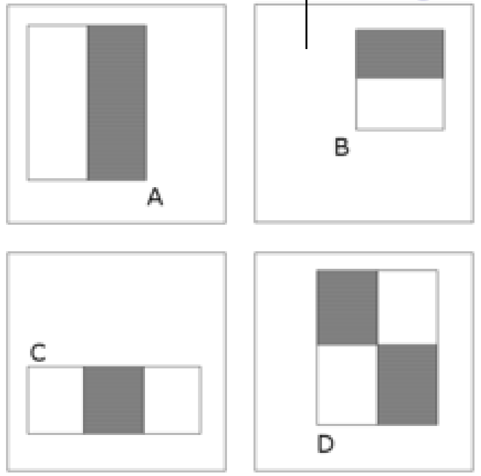
Final classifier is a combination of weak classifiers

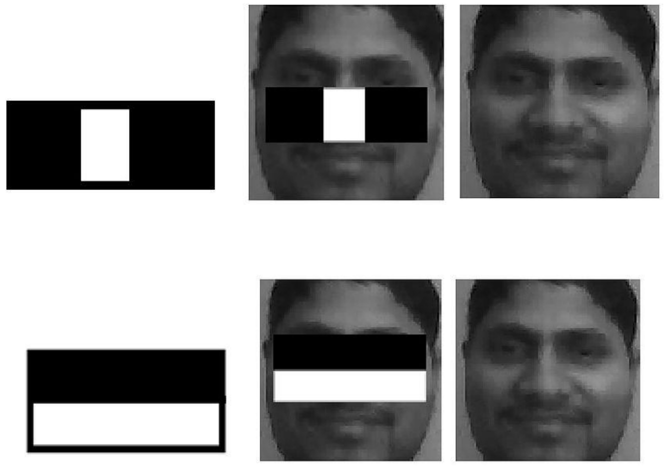


- Different algorithms, formulas for re-weighting and combining weak learners
  - E.g. AdaBoost

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

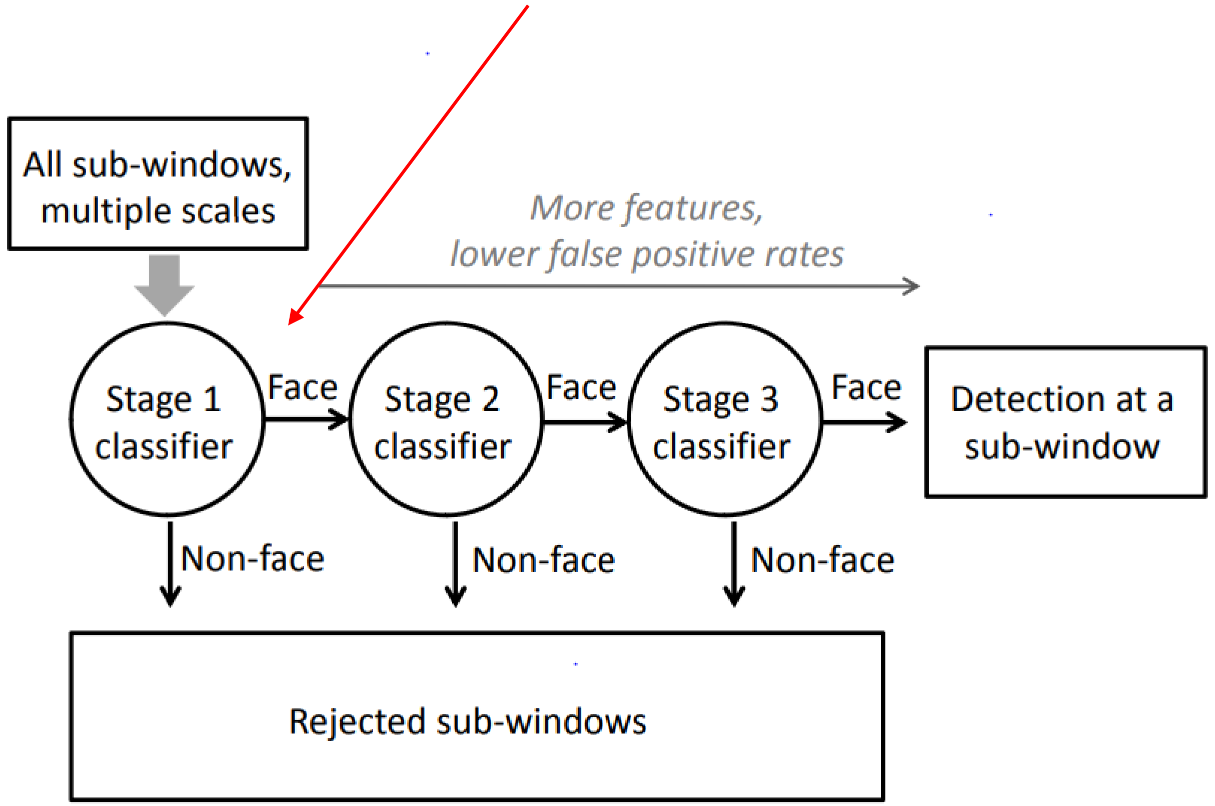




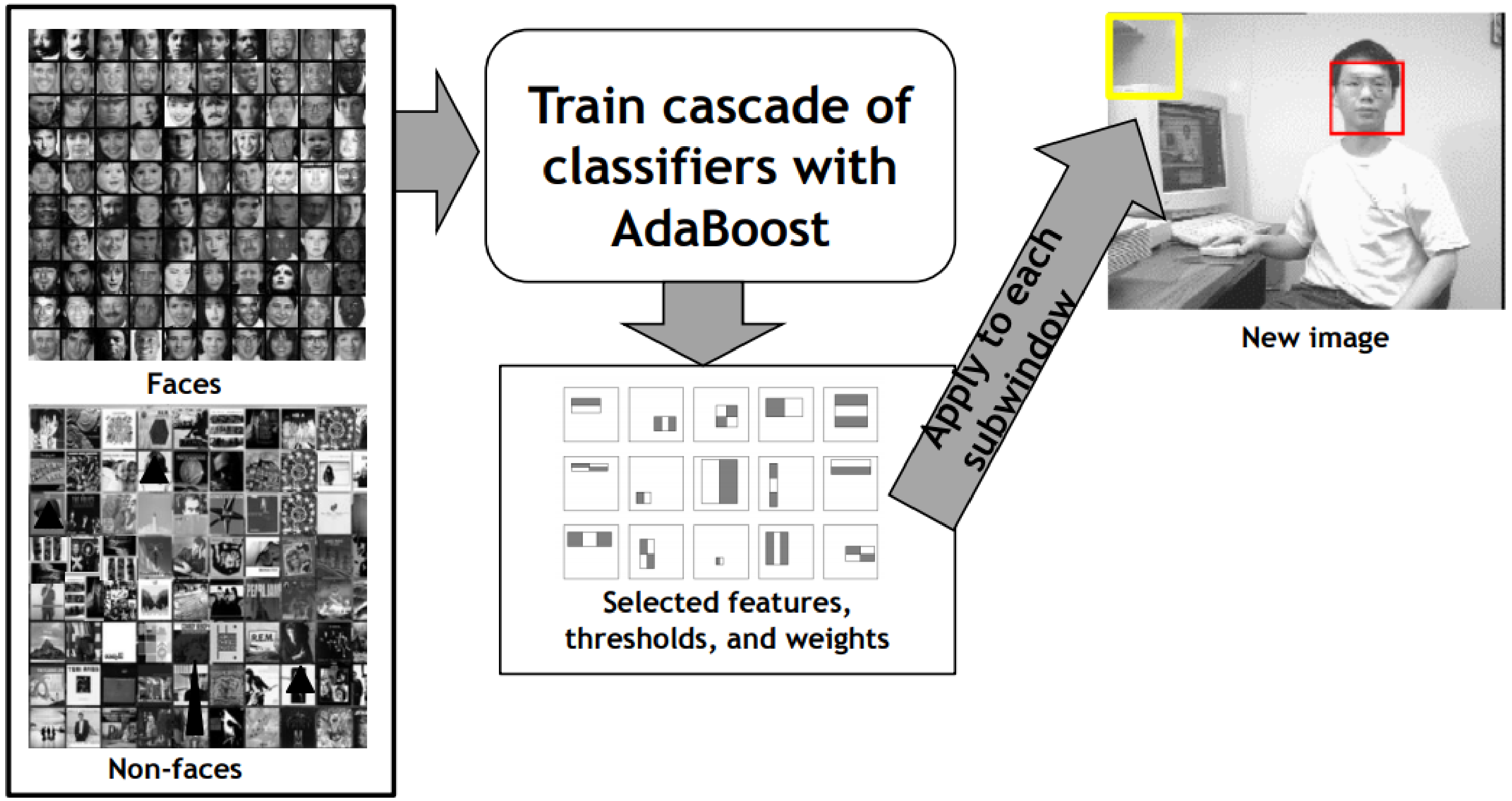
Step 4: Cascade of Classifiers

Ref: Grauman, K. and Leibe, B., 2011. Visual object recognition. *Synthesis lectures on artificial intelligence and machine learning*, 5(2), pp.1-181.

- 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


Faces
Non-faces
Train cascade of classifiers with AdaBoost
Selected features, thresholds, and weights
Apply to each subwindow
New image

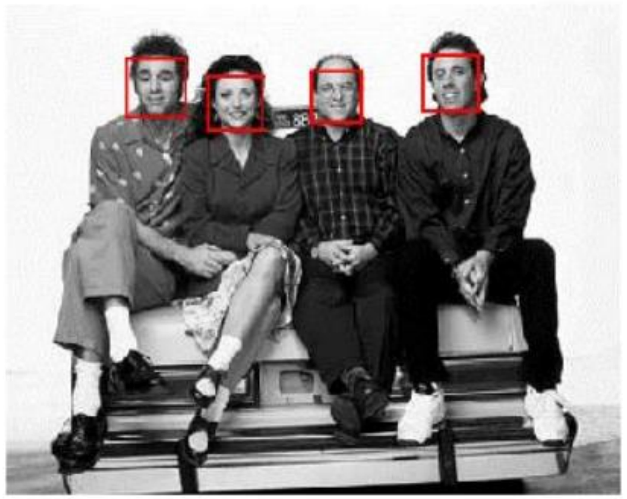
Train with 5K positives, 350M negatives

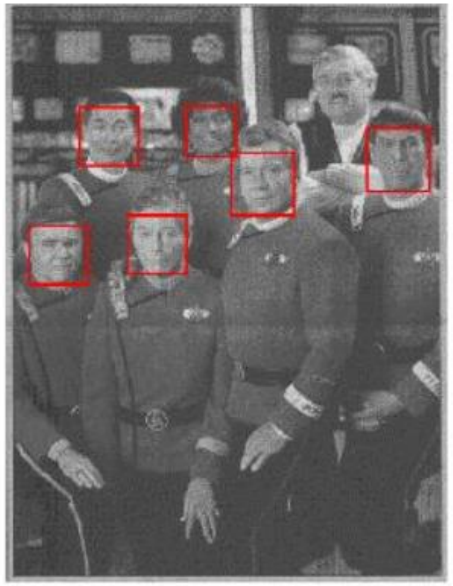
Real-time detector using 38 layer cascade (0.067s)

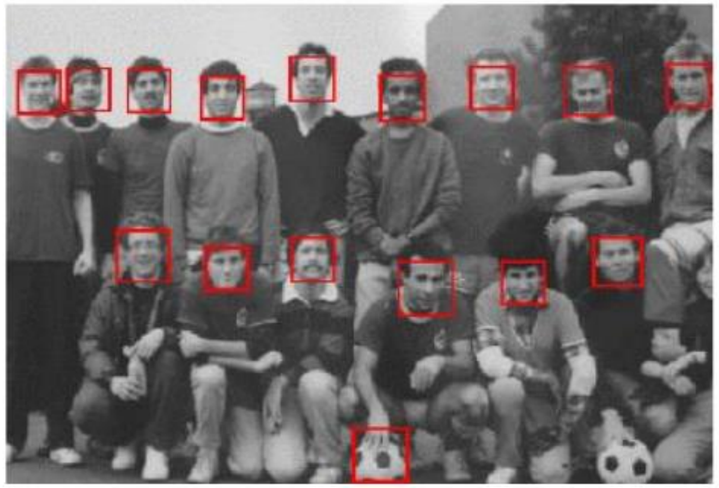
Training: slow
Test: fast

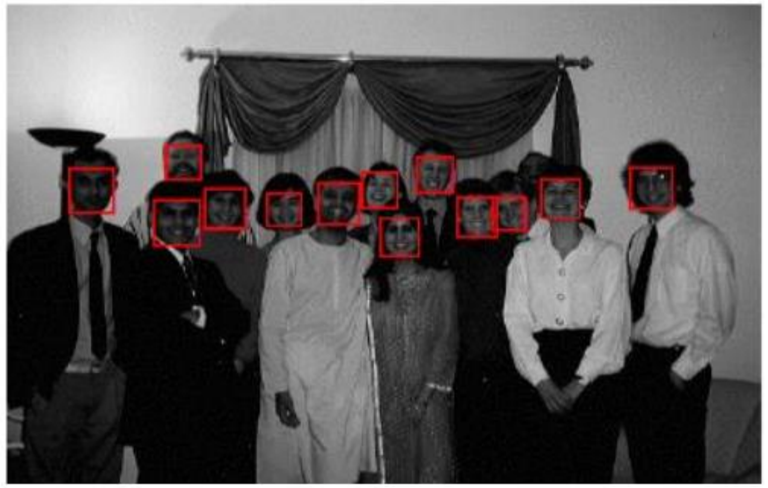
- Integral image
- Cascade of classifiers

Voila Jones: Results



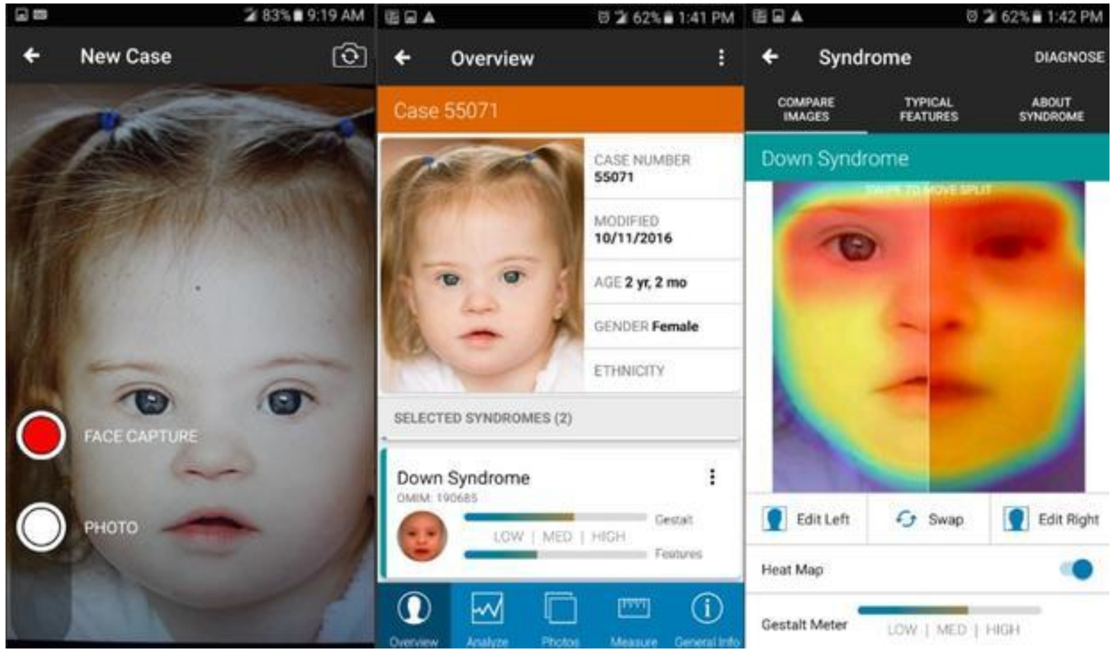






Face Recognition: Uses

- 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



Case 55071

CASE NUMBER 55071

MODIFIED 10/11/2016

AGE 2 yr, 2 mo

GENDER Female

ETHNICITY

SELECTED SYNDROMES (2)

Down Syndrome

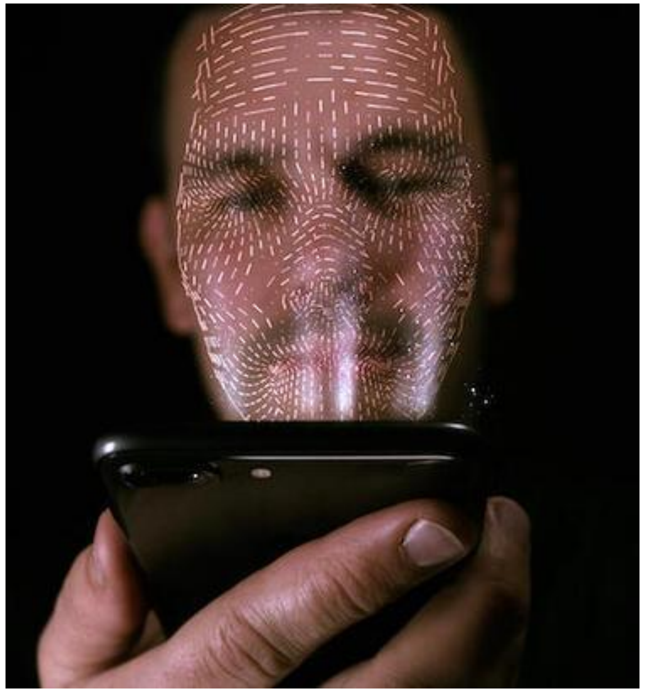
OMIM: 190685

Gestalt LOW | MED | HIGH Features

Down Syndrome

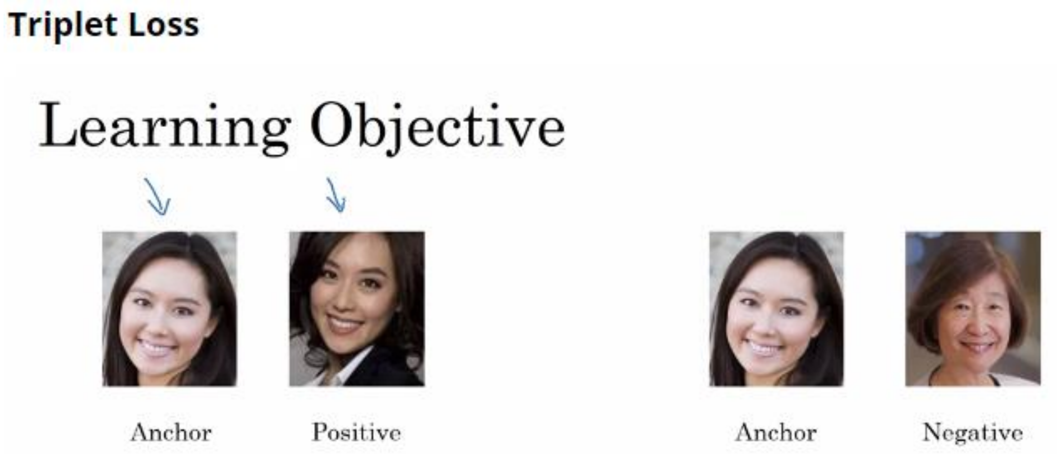
Android Face Recognition/Face Unlocking

- 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

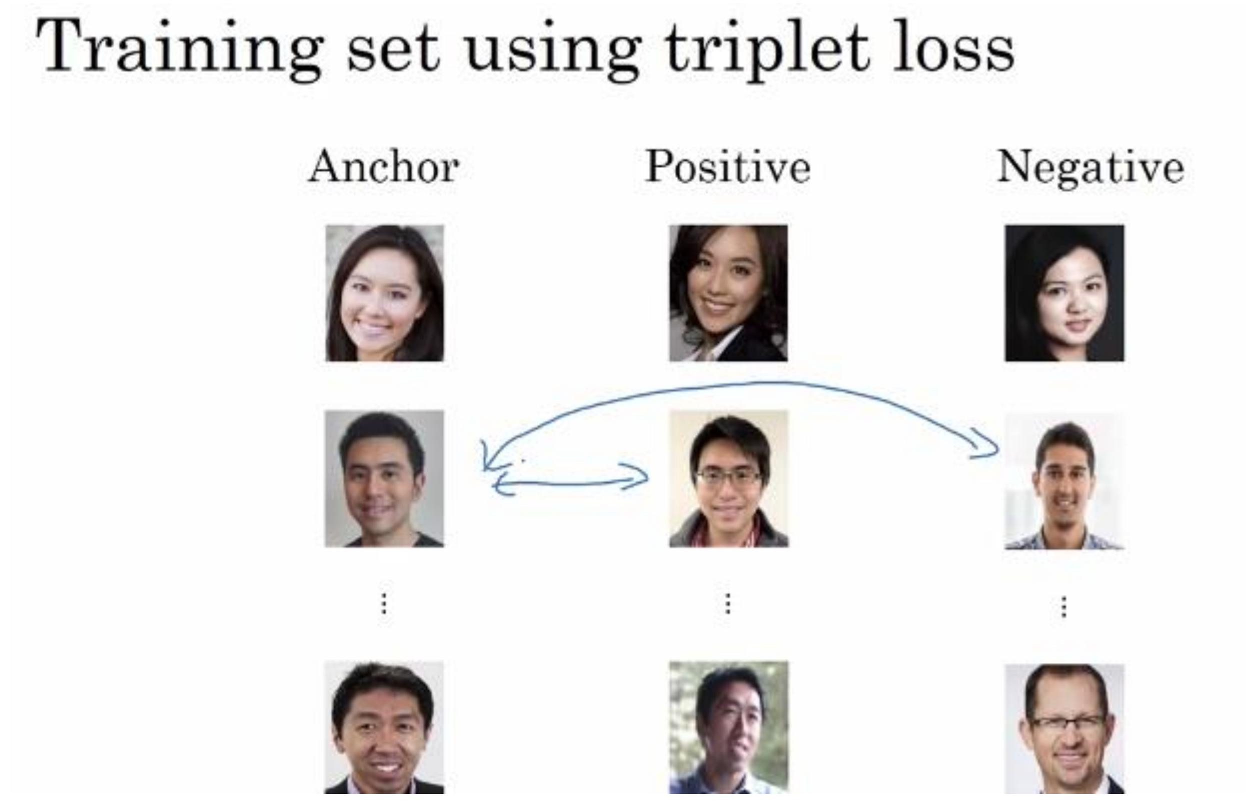
- Uses neural networks a lot these days
  - Ref: Schroff *et al*, 2015, FaceNet: A Unified embedding for face recognition and clustering
- Given:
  - Anchor: reference image
  - Postive: Positive example
  - Negative: Negative example



- 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



References

- Android Nerd Ranch, 5th edition
- Google Android Tutorials
- 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
- The complete guide to Facial recognition, Panda Security, October 11, 2019, https://www.pandasecurity.com/mediacenter/panda-security/facial-recognition-technology/
- Voila Jones Object Detection Framework Wikipedia page, https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework

References

- Network programming with Android, https://slideplayer.com/slide/8471042/
- Android Networking, https://developer.android.com/training/basics/network-ops
- Convolutional Neural Networks Course, Deeplearning.ai, Coursera