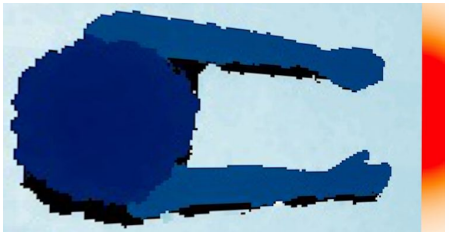
Detecting person's direction of interest



George Barvinok
Marta Didych
Denys Filippov
Yurii Ostapchuk
Andrii Palyha

UCU
Data Science
Summer School

Ricker Lyman Robotics

Supervised by:
Oles Dobosevych

Project

- Proposed by Ricker Lyman Robotic
- Customers behaviour analysis
- Retail domain

Given

- 3 days
- 5 people
- some data
- @dobosevych

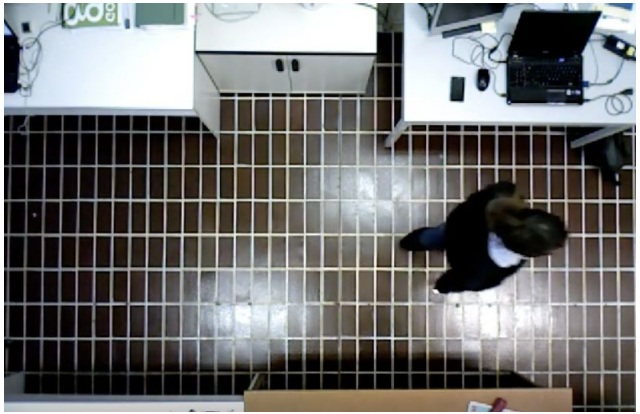
Goals

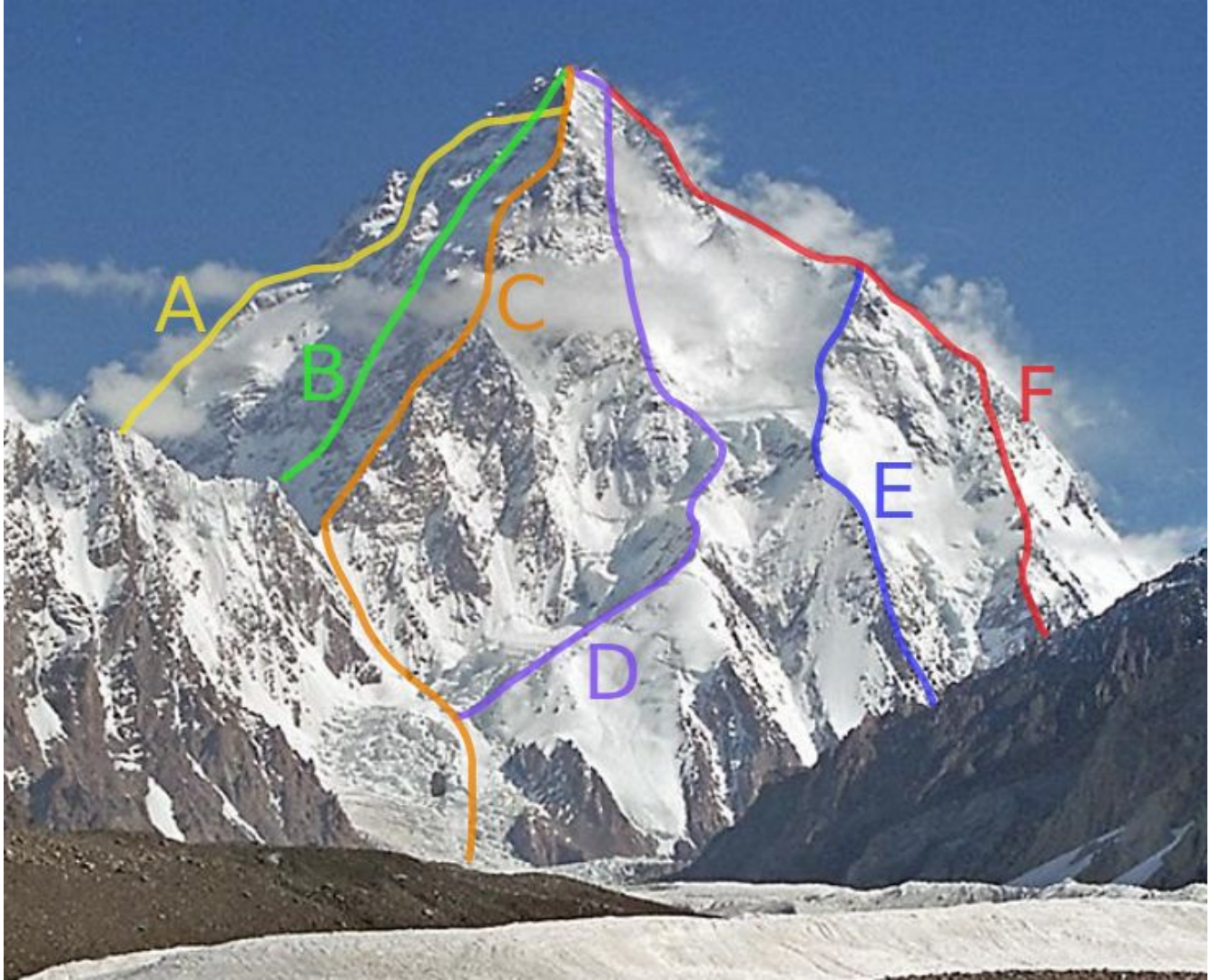
- Try and analyse different CV techniques / learn something
- Build a PoC

Dataset

Top-view cameras

- Stanford dataset (https://www.albert.cm/projects/viewpoint_3d_pose/)
  - Depth cameras
  - Labeled joints
- Politecnica delle Marche (http://vrai.dii.univpm.it/re-id-dataset)
  - Depth & Colored
  - No labels




A
B
C
D
E
F

OpenCV Background subtraction + Hough Circles

Real-time solution!

- Background subtraction
- Mrophological operations to remove noise
- Bounding box
- Histogram equalization Blur
- Hough Circles
- CNN for head detection

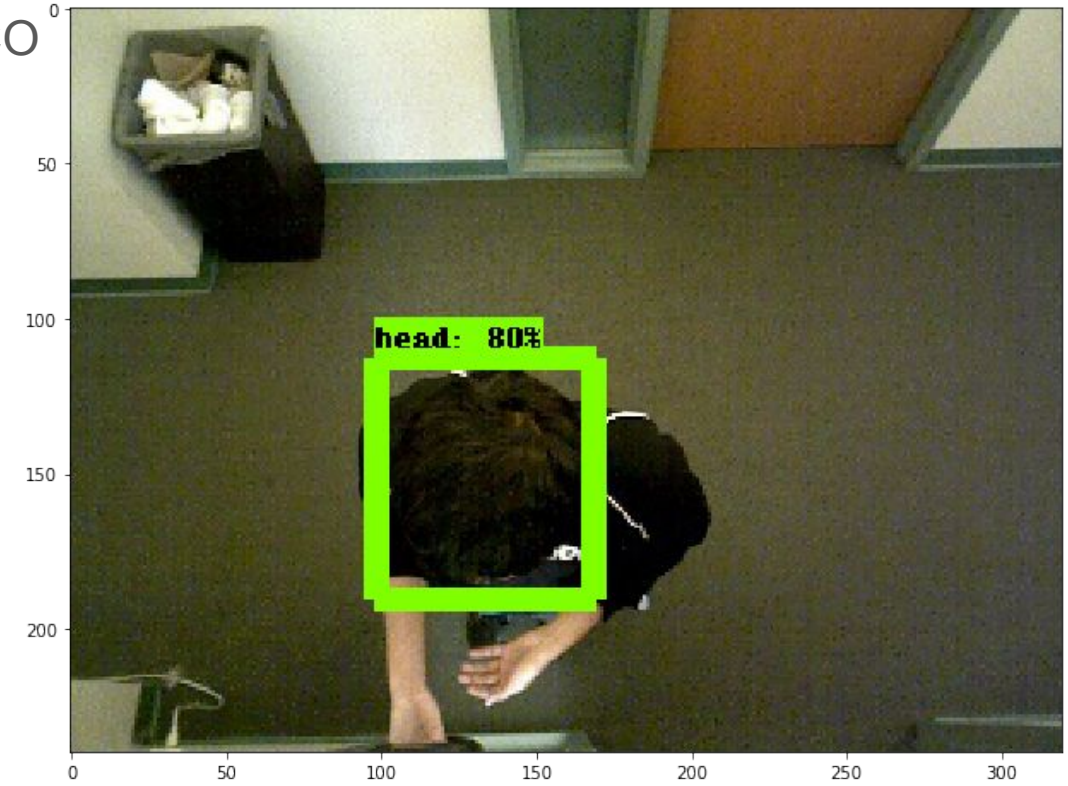
https://drive.google.com/open?id=1p41pD4zp_8R4tLsumD4aoo-5cl-QVHgI

Tensorflow Object Detection API

Pretrained SSD Mobilenet on COCO

Trained on ~250 images

Head detection

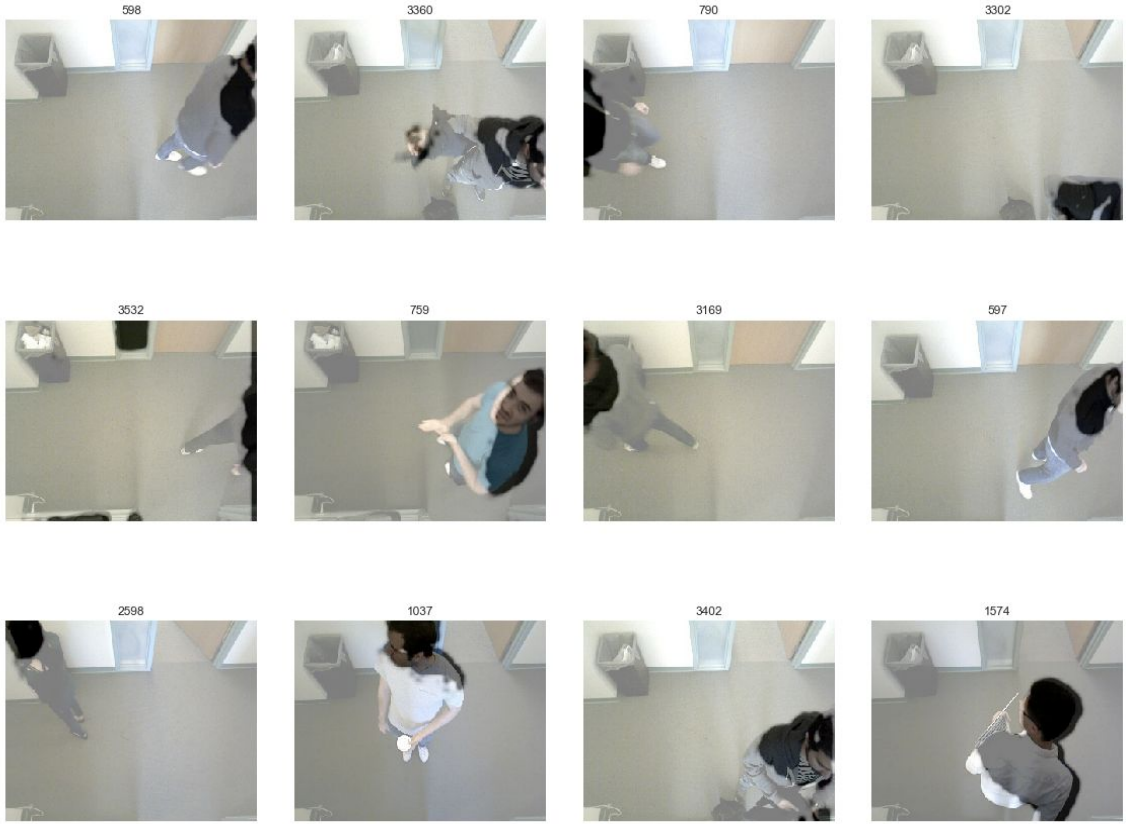


Head segmentation with U-net

Masks received from **depth** maps and trained using VGG U-net

Train on **colored** images

Still not enough data



Predicted masks from U-net


I'M TELLING YOU
YOUR'RE GONNA NEED MORE DATA
memegenerator.net

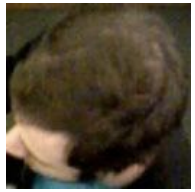
Putting things together

Prepare labeled **dataset** +

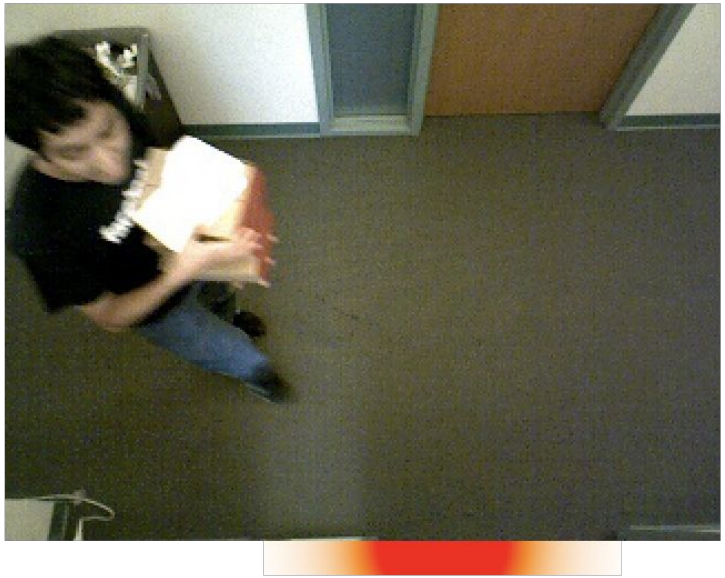
TF Object Detection API - **detect people heads** +

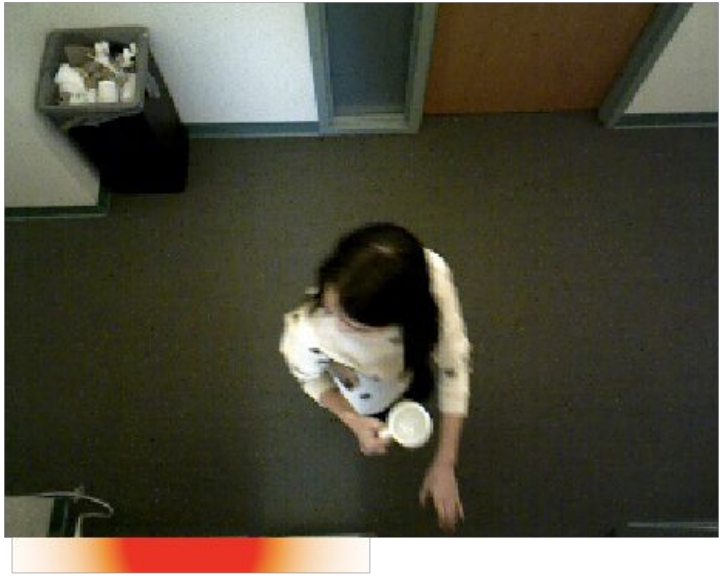
CNN Regression - **detect head direction** +

OpenCV - **visualize gaze gradient**

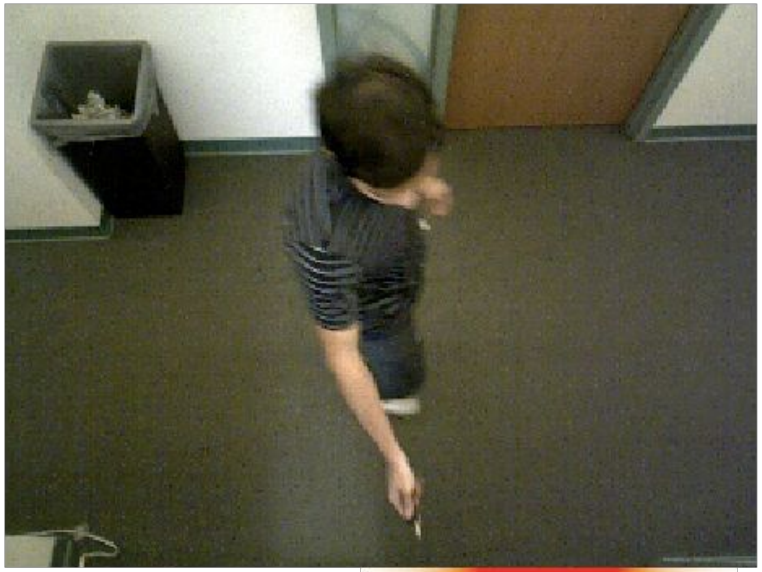


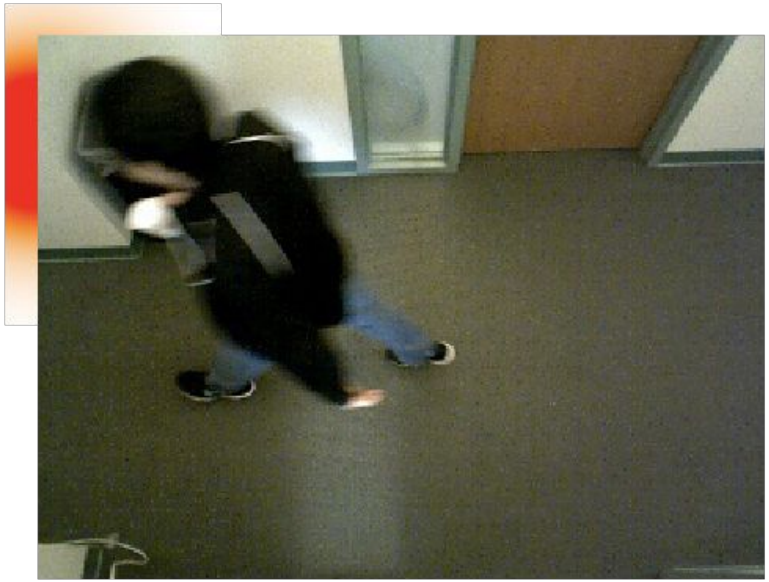
Result - region of interest



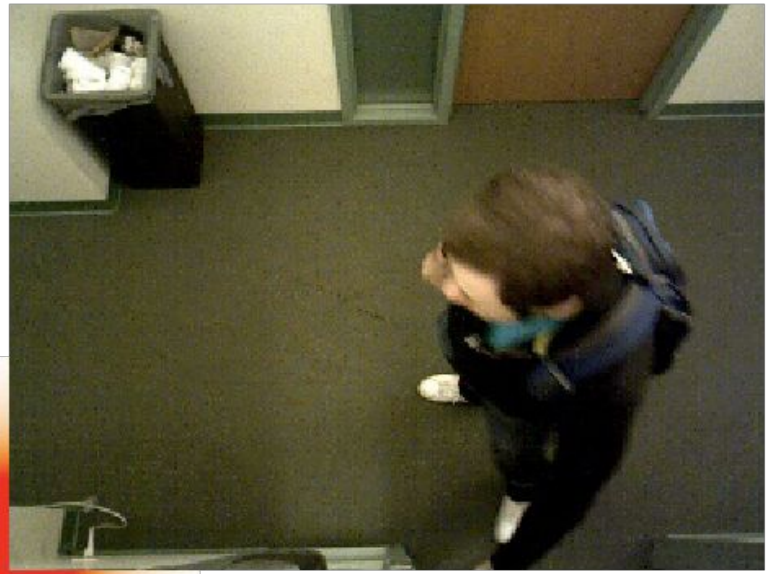


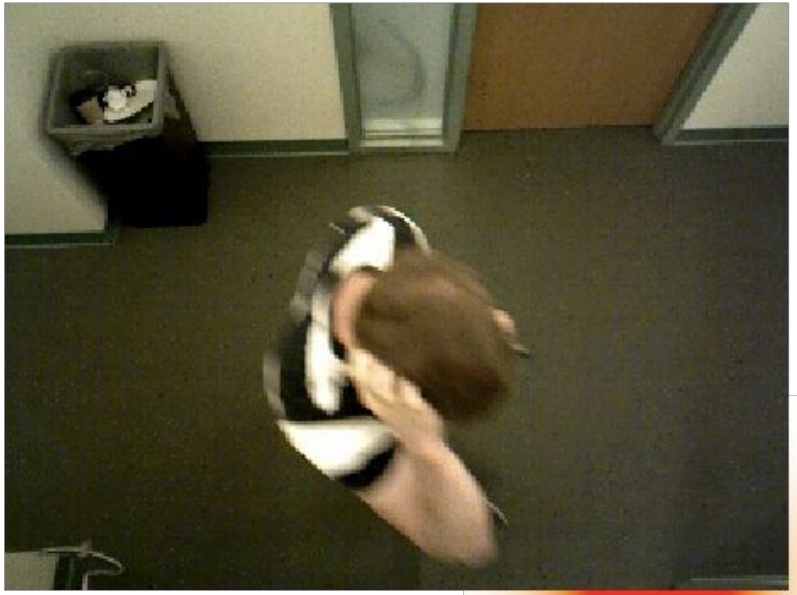
Result - region of interest





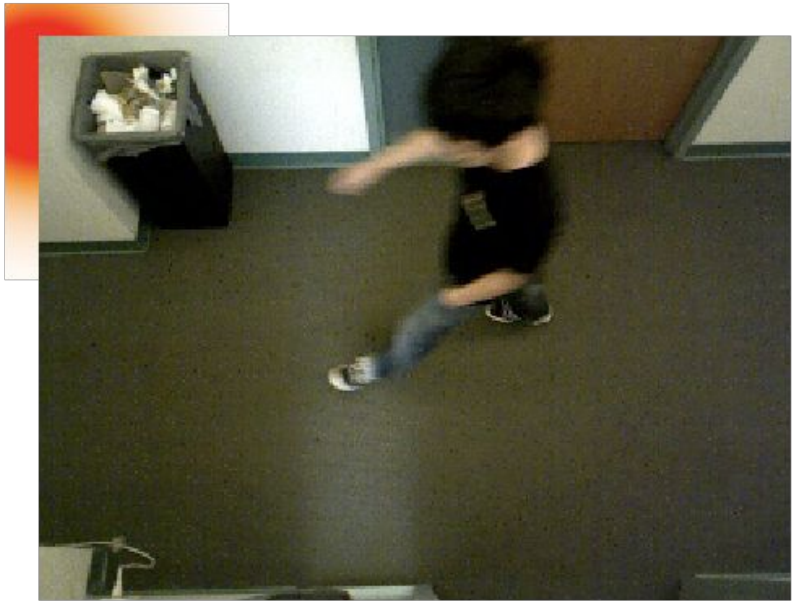
Result - region of interest

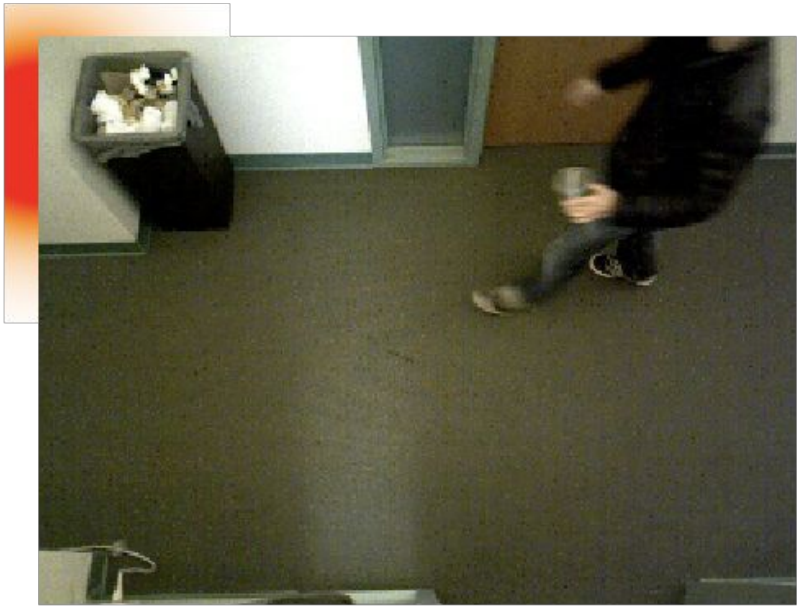




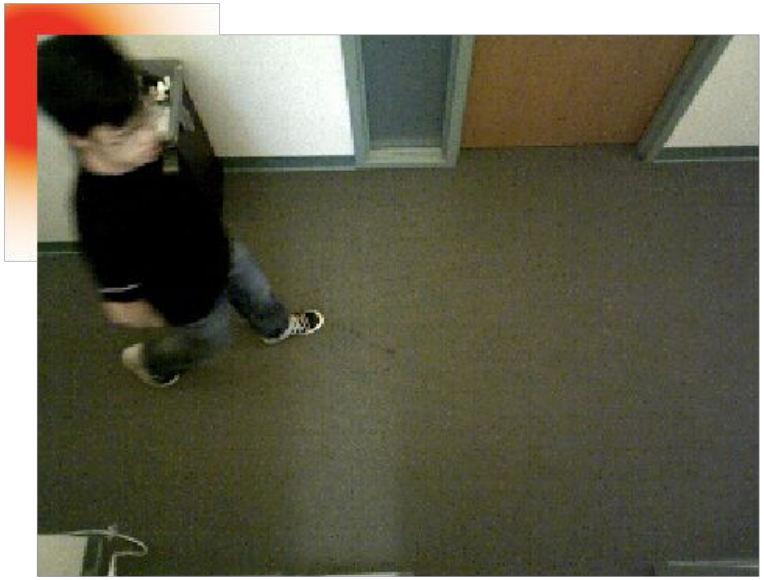
Result - region of interest

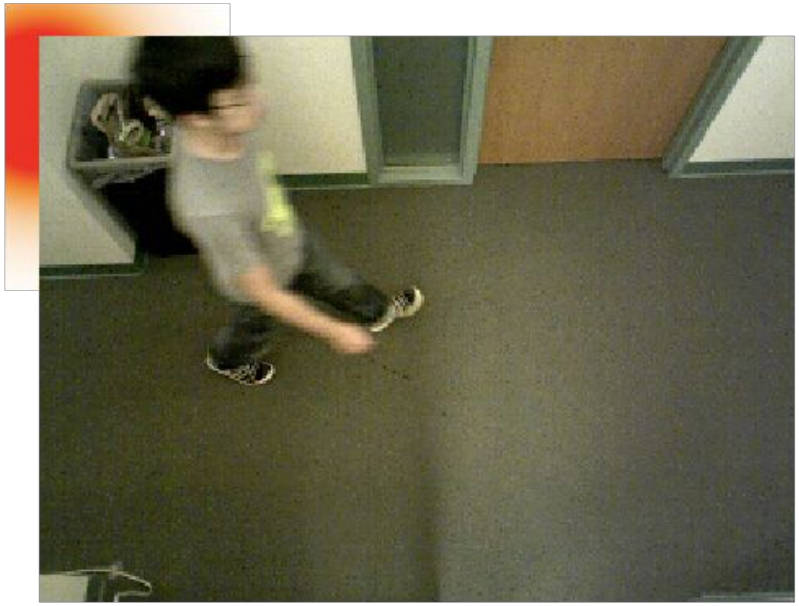
Result - region of interest





Result - region of interest





Technologies

Python 3, OpenCV 3.4, Tensorflow 1.9, Keras 2.1

Google Cloud Platform for training on GPU

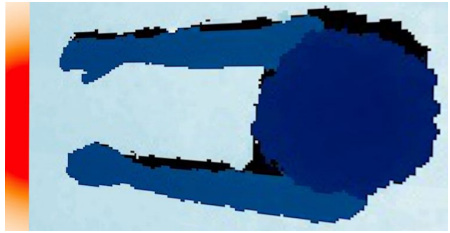
Hough Circles

U-net custom / VGG

Tensorflow Object Detection API

YOLO, SSD, Mobilenet

Summary / Lessons Learnt



- **Dataset** - is the key, need more labeled data
- Classical CV is not enough but can improve quality
- Need much more optimizations for real-time solution
    - Right now - ~2 frames per second

What’s next?

- Prepare more data
- Different approach
  - whole body segmentation instead of head only
  - use more classical CV for preprocessing data
- Involve capturing from different angles
- Different models, hyperparameter tuning

THANKS