Egocentric Height Estimation

Jessie Finocchiaro[1], Aisha Urooj Khan[2], Ali Borji[2]

(jfinocchiaro@mocs.flsouthern.edu, aishaurooj@gmail.com, aborji@crcv.ucf.edu)

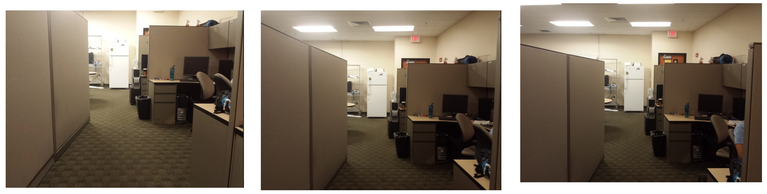
Florida Southern College[1], University of Central Florida[2]

1. Problem

- Estimate the height of a camera in an egocentric video.

Difficulties

- Since video is egocentric, the camera wearer is generally not in the frame
- Image background is variable



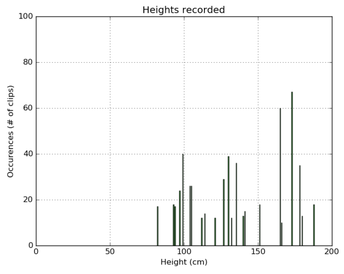
Images taken from a person's waist, chest, and head, respectively, while standing in the same spot.

2. Proposed Method

- Collect videos of 10 people walking with a camera at their waist, their chest, and their head to simulate three different people.
- Train a Support Vector Machine to classify rough heights from which a video was recorded based on different image descriptors.
- Explore various deep learning techniques to estimate height as a continuous measurement

2.1 Data collection

- Ten people recorded 6 videos each. 2 from the waist, chest, and head each.
  - One video had a static background (no movement or other people)
  - One video had a dynamic background (unrestricted background movement)
- Height in centimeters for each camera was recorded.



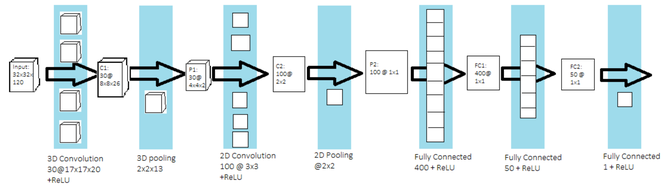
Distribution of camera heights with video length taken into consideration

2.2 Support Vector Classifier

- Trained a SVC on different discretizations of the images
  - Which camera (3 bins)
  - 25 cm bins of height (Chance = 0.2)
  - 10 cm bins of height (Chance = 0.09)
- Trained on different sets of frames
  - 1st frame in each video
  - First 5, 10, 20 frames in video
  - Every 15th frame

2.3 Temporal Convolutional Neural Network

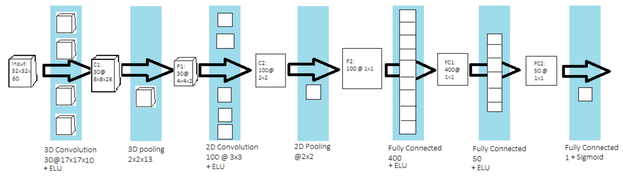
- 3D Convolutional Neural Network with Sparse Optical Flow as the input
- Frame rate is normalized to 15 fps
- x and y flow vectors are concatenated on the temporal dimension



The Network Architecture is based heavily on that from Peleg's paper [1] for Egocentric Action Recognition.

2.4 Spatial Convolutional Neural Network

- 3D Convolutional Neural Network with grayscale images as the input
- Similar network design to Temporal Network



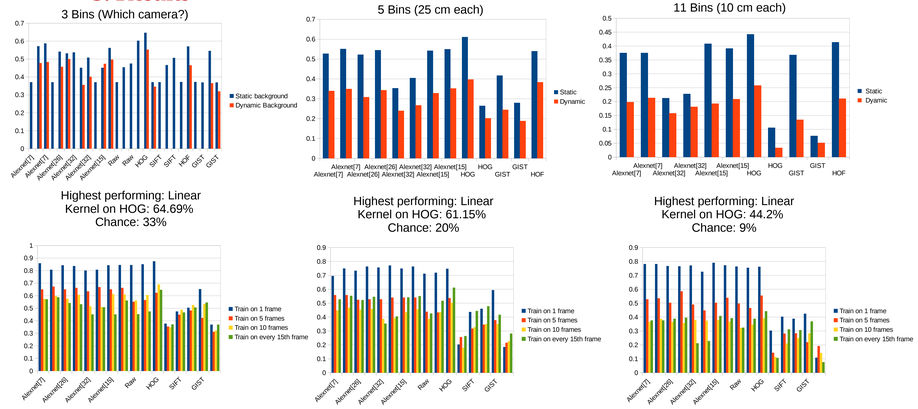
2.5 Two-stream Convolutional Neural Network

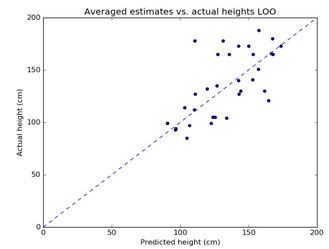
- Take pre-trained networks from 2.3 and 2.4 and concatenate output
- Use a fully connected layer to use both temporal and spatial information to yield one estimate.
- Concatenated at two levels
  - Connected after final output
  - Connected at 50 neurons in each network

Intuition

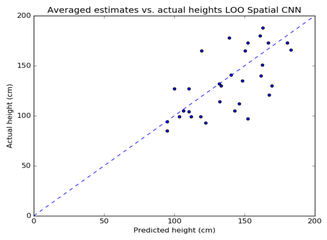
- Height cannot be learned solely on optical flow patterns nor spatial information.

3. Results

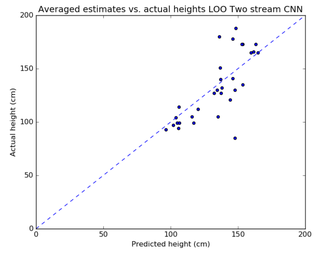




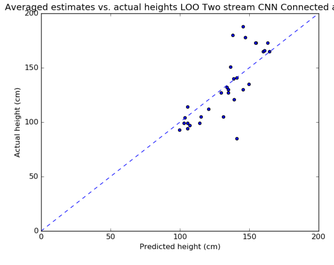
Temporal results with Leave One Out Testing



Spatial results with Leave One Out Testing



Two stream CNN results on Leave One Out Testing



Two stream CNN results (connected at 50 neurons) on Leave One Out Testing

	Mean Absolute Error (cm)	Mean Squared Error	r^2
Temporal Network	18.03 cm	549.28	0.4557
Spatial Network	19.51 cm	602.54	0.4267
Two-stream Network- Connected at final output	14.53 cm	323.98	0.5773
Two-stream network- connected at 50 neurons	**14.04 cm**	**303.48**	**0.6505**

4. Acknowledgements

We would like to thank Dr. Borji, Dr. Shah, and Dr. Lobo for their feedback and support, as well as the National Science Foundation (NSF) for granting us the opportunity to have the REU program at the University of Central Florida.

5. References

[1]Poleg, Yair, Ariel Ephrat, Shmuel Peleg, and Chetan Arora. "Compact CNN for Indexing Egocentric Videos." 2016 IEEE Winter Conference on Applications of Computer Vision (WACV) (2016): n. pag. Web.

[2] Poleg, Yair, Tavi Halperin, Chetan Arora, and Shmuel Peleg. "EgoSampling: Fast-forward and Stereo for Egocentric Videos." 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015): n. pag. Web.