Facial Recognition and Learning with Embedded Computing on Raspberry Pi

by
Kelly Deckelman Brandon and Jigar Jadav
SMARTER NYU-Poly

Acknowledgements

- Prof. Vikram Kapila – for the opportunity
- Jared Frank – for all the help, support, and laughs
- Valentin Siderskiy – for KNN
- David Lopez - for his insight on Single Board Computers
- The Grad Students, High School Students,

CAESAR's Mental Issues ☺

- Two problems associated with the "brain":

1. Processing and control – embedded computing (Hardware)

2. Face recognition and learning – algorithms and data structures (Software)

Face Recognition & Learning (FRL)

Recognition

- Detect face in video frame
- Compare face to existing stored images
- Predict label for detected face

Learning

- Analyze strength of prediction
- Conditions for weak prediction
  - A new perspective of a known
  - A never-seen-before face

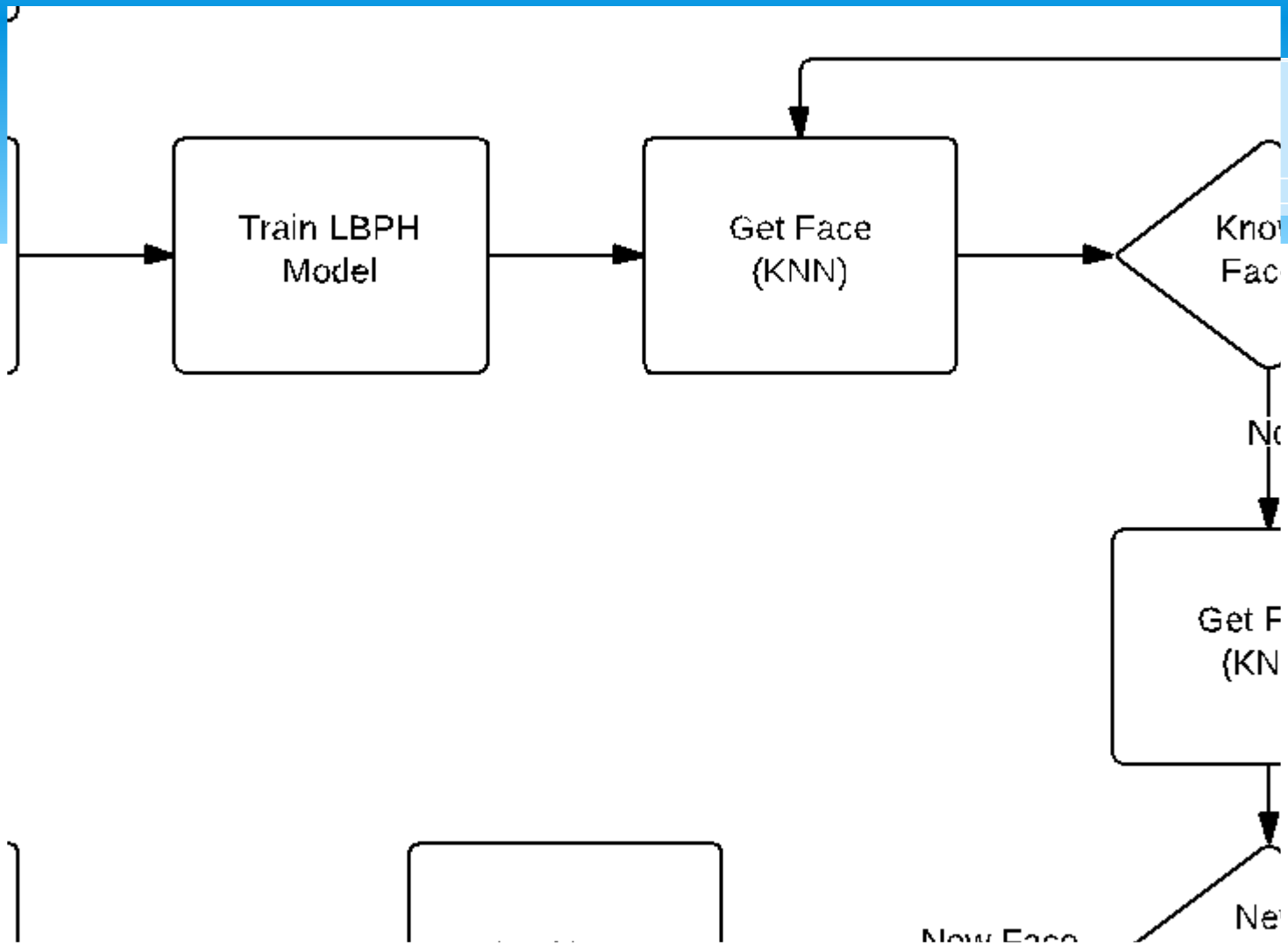
FRL Current Conditions

- Laptop running on Ubuntu (Linux)
- Using OpenCV and Fisherfaces
- Detecting two known subjects with confusion

FRL Proposed Solution

- Use the Local Binary Pattern Histogram (LBPH) model from OpenCV instead of Fisherfaces
- Use a modified K Nearest Neighbors (KNN) algorithm to improve recognition and introduce Learning
- Use a Threshold and comparison of subsequent KNN blocks to tweak and further improve recognition and learning

System Flow



FRL Challenges

- Threshold may vary from location to location depending on lighting (can this be automated?)
- Improving recognition of a face by storing a new image without prompt is risky
  - A face in the wrong face file ruins recognition for both faces
  - Currently such a file can only be manually deleted

FRL Testing

- System is currently setup for testing
- Tests run less than 1 minute (as seen in demo)
- Participants may be recorded numerically for anonymity
- Some testing run this morning

Hardware - Embedded Computing on Single Board Computer (SBC) Raspberry Pi (Rpi)

- Run Kelly’s code on the Raspberry Pi without any modifications
- Hardware options:
  - Gumstix - bigger learning curve, smaller community, expert users
  - Raspberry Pi - smaller learning curve, bigger community, beginner level
- Winner - Raspberry Pi

Current Hardware Setup

- Laptop physically connected to CESAR
- Laptop running face detection program
- Not mobile

Proposed Hardware Solution

- Using a Single Board Computer like Raspberry Pi
- Connect wirelessly on WI-FI for any control
- Overclocking the Rpi to speed up face detection on Rpi
- Use openGL library for image processing (at software level)

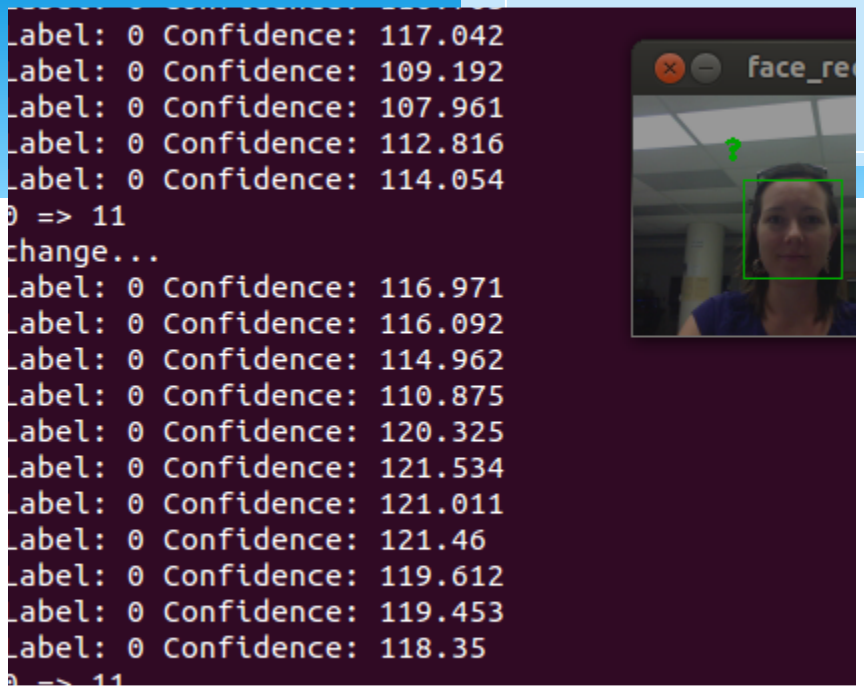
Single Board Computer Challenges

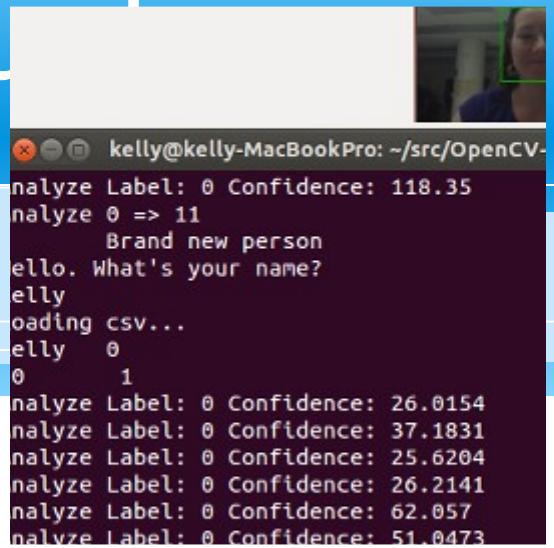
- Started with GumStix - 4 weeks not enough
- Raspberry Pi - Broken board
- SSH and VNC wireless connection
- Compiling openCV libraries
- Unable to support 640x480 resolution
- Tweaking the hardware:
  - Overclocking the CPU
  - Reassigning RAM to GPU and CPU

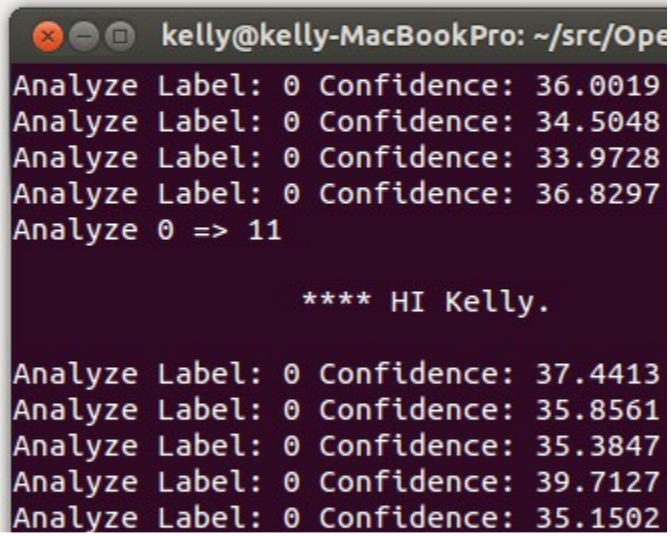
Rpi Testing

- Rpi is currently setup to run on the follow specs:
- CPU: Overclocked to 900Mhz
- RAM: Evenly split between CPU and GPU
- No SSH or VNC connections
- Unmodified face detection algorithm of Kelly running at 160x120 resolution

FRL Demo on Ubu

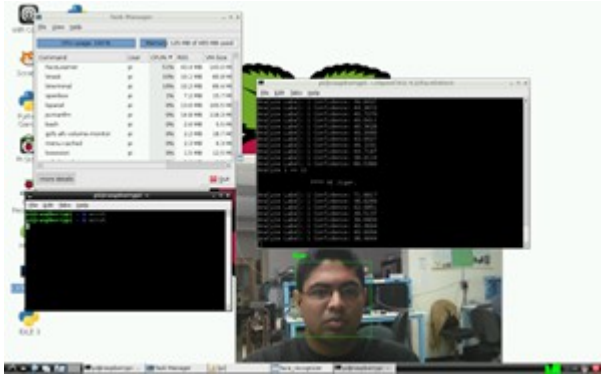






Rpi - Setup and Proof of Concept

Raspberry Pi Setup



Rpi - Proof of concept

Next Steps

- Understanding limitations of Single Board Computers (SBC) and look at alternate SBCs
- Test FRL for learning & recognition efficacy, and possible changing variables: K, Threshold, Recognition Model
- Modify the FRL to run on SBC
- Modified CPU to specifically run processor hungry

Questions