



Identity Checker with FPGA

Team C7

Junye (Hans) Chen (junyec)

Sheng-Hao Huang (shenghah)

Andy Shen (asshen)

Application Area



Functionality

Facial Detection

Facial Recognition

Areas

Software+Hardware (FPGA)

Software

Why FPGA?

Significant parallelism in Image processing algorithms

Solution Approach - Overall

Software



Laptop with Camera



Django
Framework
(webapp)



Viola-Jones
Algorithm
(detection)



Eigenface
Algorithm
(recognition)



KC-705 FPGA Board



Viola-Jones Algorithm
(detection)

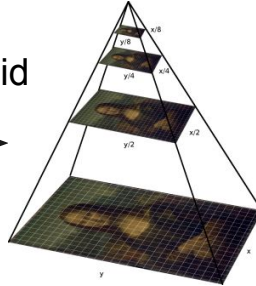
Hardware

Solution Approach - Facial Detection

Classification Pipeline



Build image pyramid
(new!)



All subwindows

Stage 1

pass

fail

NOT face

Stage 2

pass

fail

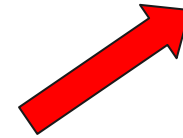
Stage 3

pass

Further Stages

Face
Found!

Training:
Pre-Trained Weights
from OpenCV



Solution Approach - Facial Recognition



Yale Faces Database
15 subjects, 166 images

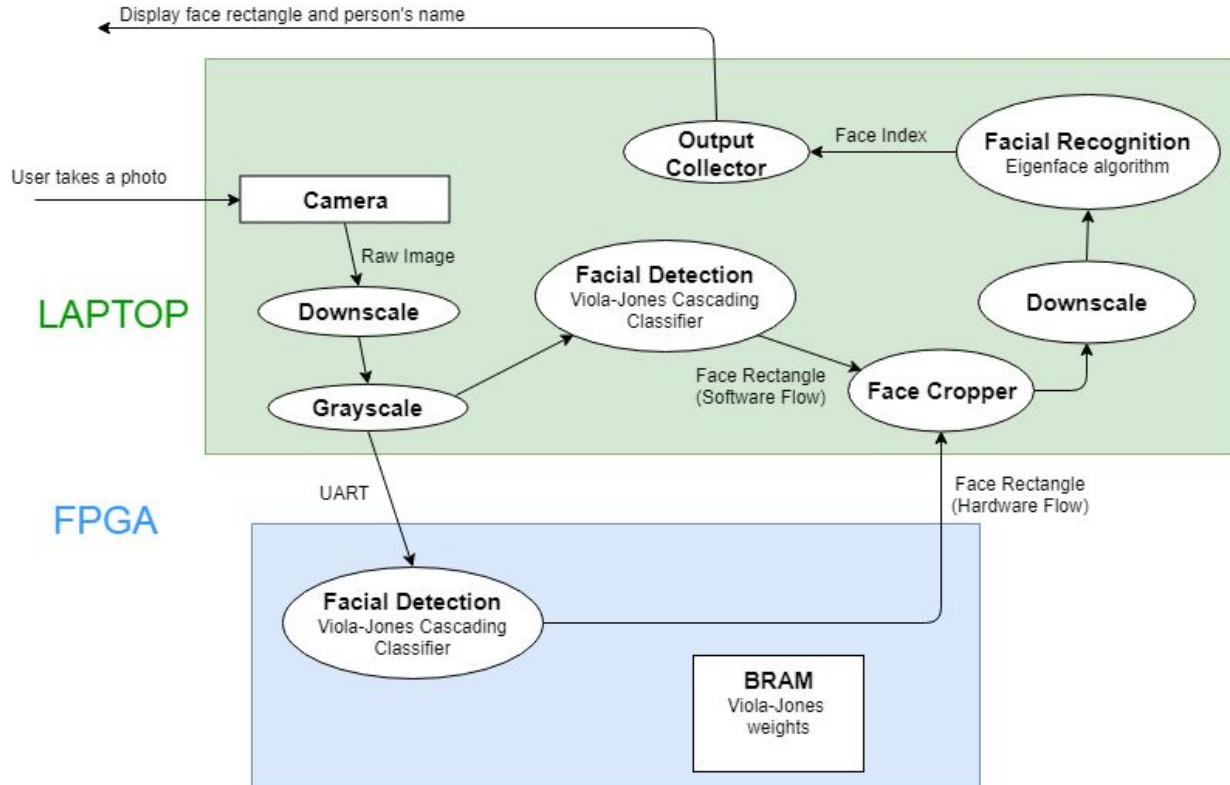
Training

1. Prepare a face database with 8 20x20 images per subject.
2. Use a set of eigenvectors to represent how each face differs from the mean face.
3. The eigenvectors are called “eigenfaces”. They measure the “variance” of the face database.

Classification

1. Project new test image onto eigenfaces. This tells us how different the test image is from the mean face.
2. Find a face that differs the same way as the input image. That face is the closest match.

Block Diagram



Complete Solution - Add Face



Identity Checker



Add to Database

Check Identity

Identity Checker

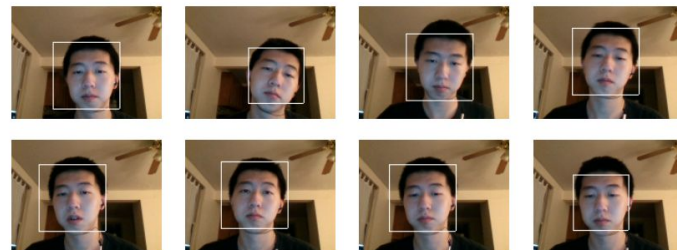
Enter your name for the Identity Checker:

Junye

Chen

Submit

Identity Checker



Complete Solution - Recognize Face



Identity Checker

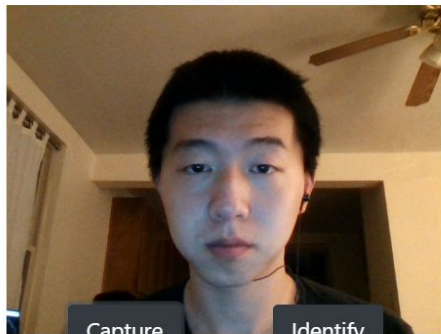
Add to Database

Check Identity

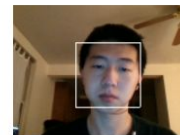


Identity Checker

Finished



Identity Checker



You must be: Junye Chen

Restart

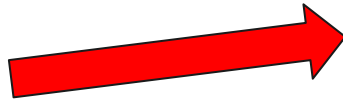
Metrics and Validation - Correctness



Yale Faces Database
15 subjects, 166 images

Facial Detection

- Goal: 80% accuracy
 - “box” the face if there is one
 - Reject if no face
- **166** images
- **163** Faces correct
- **98%** accuracy
- Good lighting



Facial Recognition

- Goal: 80% accuracy
 - Identify person’s name, assuming face is in database
- 163 images, $8 \cdot 15 = 120$ training images
- **43** test images
- **37** correctly recognized
- **86%** accuracy

Metrics and Validation - Speedup

Target: 0.05s (detect one face on software)

FPGA Baseline: 0.17s

Goal: 5x speedup

Optimization Step	Improvement
Faster clock: 50MHz => 200MHz	0.17s => 0.0826s
Loop pipeline	0.0826s => 0.037s
Loop unroll	0.037s => 0.031s

Result: 1.6x speedup

Project Management

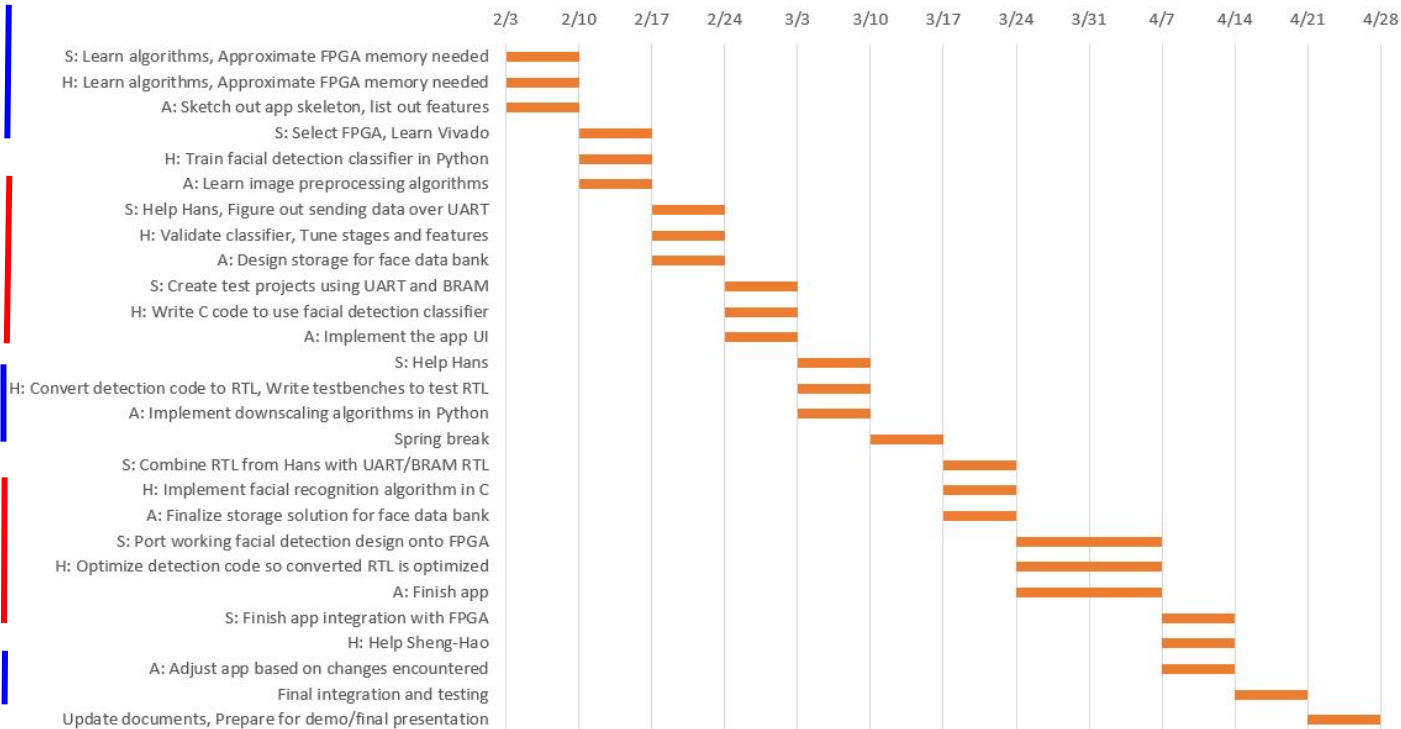
Research + Design

Software Facial Detection

Integration

Software Facial Recognition

Integration



FPGA
UART
connection

FPGA Facial
Detection
(longer than
expected)

Lessons Learned



- Always have contingency plans
- Identify dependencies between tasks, and follow schedule closely
- Diversification of team members is very important