Indian Institute Of Technology,
Bombay

CS 101

Autumn 2014

Team Diary and Journal

Keshav Srinivasan
Kshitij Bajaj
Kumar Ayush
Reebhu Bhattacharya

October 19, 2014

Introduction

We aim to develop an image processing application targeted to real-world applications such as attendance, emotion detection, psych-analysis, flora-fauna classification. Combined with tools from opencv, we also look forward to harness sound processing packages in conjugation with out application.

- **6th October 6:00 PM** Reebhu, Ayush, Kshitij and Keshav decide to form Team for CS 101 Project and decide to meet the next day to discuss the topics. Everyone is told to each come up with their own ideas to discuss the next day.
- **7th October 1:00 PM** The team held their first meeting. Kumar Ayush is chosen as leader of the team. Every member contributes ideas for the project, of which a few ideas were short-listed. The section below summarises the ideas that were discussed. finally a topic was chosen via an objective approach. The ideas section below summarises the ideas that were discussed. We finally chose an idea via a objective approach to rank the ideas. Details are summarised in the said section.

Ideas

- Mathematical program like ganit++ to find roots of an equation, integration/differentiation, plot, simultaneous equations, linear algebra and many other mathematical functions
- games like: quarto, chain reaction, etc
- Chat-bot
- Sach ka Saamna (Truth Teller)
- Minecraft Server related programs
- Circuit Simulator
- Video Attendance and Facial Recognition
- Processor Simulator
- Information Extraction (NLP)
- Map and GPS(Global Positioning System) for IIT-Bombay

Top contenders amongst these ideas were Circuit Simulator (CS), Video Attendance (VA), GPS map for IIT-Bombay (GP).
There were set three critera for choosing one out of these: Complexity

(C), Practical Use (P), Flexibility (F)
All team members gave marks out of 10 to all three projects for each criterion mentioned above, which is summarised in the table below:

-	CS	VA	GP
C	33	35	33
P	33	37	37
F	25	25	32

The marks were then weighted as: P(0.5), C(0.2), F(0.3).
GPS for IIT-Bombay won by this method, but was vetoed by Reebhu. He felt that it is too simple and easy to do. The team agreed to discard the idea and further discussed the other 2 topics.

- **7th October 3:00 PM** After prolonged deliberation over the two topics, the team unanimously agreed on the Video Attendance and Facial Recognition. We decided to wrap up the meeting after deciding upon the topic.

- **8th October 2:16 PM** The team met up the second time to discuss installation and trial of openCV. We setup openCV 2.4.9 with Visual Studio 2013, and run some sample codes from the internet, such as loading an image to various color forms and getting input from a web-cam. We also self-implemented a ColorDodge blend algorithm to test the waters Everyone familiarises themselves with the OpenCV and each told to read up on the software.

- **12th October 9:00 PM** All team except Keshav met. He had to go home. This meeting was a total waste since we messed up the settings on Visual Studio '13 and had to reinstall it in order to get a simple Face Recognition code which ultimately gave undesired output. Being optimistic, we learnt what not to do.

- **13th October 8:50 PM** Reebhu gets on to a code snippet. His task is to take a human potrait and crop it so that only the face remains.

- **13th October 9:02 PM** The rest of the members move on to discuss points for SRS (Software Requirements Specification) and SDD (Software Design Description). The points will be compiled into the documents by Keshav and Kshitij respectively. The sections below summarize the discussions and notes we have for the two documents

- **13th October 9:23 PM** Reebhu successfully completes the task assigned He is asked to write a comment for every thing under the assumption that nothing is obvious. The code he has written is good and can be used in the main program as well.
- **13th October 9:59 PM** We discussed a lot for SRS and SDD, but not all. We'll continue the discussion tomorrow. Meanwhile, Reebhu has some doubts. The whole team moves to figure the code part out.
- **17th October 10:11 PM** The team assembles. Reebhu is learning unsupervised learning algorithms, hidden Markov models and other texts of relevance to the project. Ayush is experimenting with reading frames over a network webcam. Kshitij and Keshav are writing the SRS and draft user manual.
- **17th October 11:39 PM** Keshav and Kshitij is about to finish the documentation for SRS and SUD. Reebhu finished reading some of the material he sorted out for himself. He has derived a list of various methods that can be used to achieve the main task of face recognition.
- **19th October 10:53 AM** Reebhu started analysis of various possible algorithms. Kshitij was assigned a coding job. Keshav and Ayush started polishing reports.
- **19th October 3:45 PM** We have decided to use a pseudo-HMM algorithm which has given 100 percent results with the ORL database. Polishing of all documents are done. We move on to quickly compile the Project Report

SRS Notes

Introduction

Purpose

- face recognition would help develop a video attendance package
- simplifies attendance, less human effort of any sort
- easily cut off attendance if someone walks out after giving attendance
- unless someone hides his face, or prints a poster of his friend for a proxy

System Overview

- identifies and seperates faces from a crowd
- facial recognition
- marks attendance in a database
- has central connectivity to a network database
- uses intel's opencv for other purposes
- blinking triggers attendance, avoids proxy via a poster [possible feature]
- sound recognition to confirm in case of confusion
- 32-bit windows based
- gui to initialize application
- updates database regularly to account for evolution

References

- give opencv links here
- some papers by microsoft research, one of the first ones to develop tracking
- wikipedia

Overall Description

Product Perspective

- user interface is a terminal
- ask for model and other parameters
- hopefully we can write a gui if we get time
- hardware interfaces involve hd webcam
- processor atleast 2 GHz, 2 GB RAM
- large data storage facility to store reference images

- software interfaces ,windows x64 7 or above, windows SDK, .NET framework
- hopefully the latest versions
- opencv libraries
- communication interfaces, we need high-speed LAN
- memory constraints none
- as suitable as spending resources on biometric

Constraints, assumptions and dependencies

- low reliability if poor quality images
- face recog algo not very good - theoretical limit on accuracy
- lighting in environment- affects quality of image
- people in iit rarely take haircut, we might take the portion below eyebrows, but then they rarely do a shave as well
- background disturbances e.g. someone wearing a skin colored tee

Software System Attributes

- Fine! we admit it. Not reliable at all.
- but that is under the assumption that everything is automatic
- if we deduct the automatic element in it and have the user show his straight face before a camera and say a voice message such as 'yo yo',this might work better
- easy to implement, no problems in availibility of equipment and software resources
- security we write a code, nobody dares to hack it Period.
- The hardware is as easy to maintain as any database is these days
- portability is not a problem, all you need is a webcam and network connectivity.

SDD Notes

Structure Viewpoint

- the code is divided into specific modules that do stepwise work
- the integrity and reliability of each step is crucial to the next step
- each step shall be built by different members of the team
- this ensures that the code is as strong as the weakest snippet - I dunno how that's in our favor
- the planned parts are summarized in the following portion:-

  Parts

  - getting the faces out of the video
  - cropping them
  - adjusting brightness-contrast etc
  - scaling, rotation - standard transforms of image
  - setting up the data for model
  - setting up data storage
  - setting up network connectivity
  - training the model and getting the output
  - gui interface for setting model parameters, editing database, admin stuff etc
  - webcam and other hardware setup