Engineering Notebook

Monday, September 21, 2020 5:15 PM

Engineering Notebook:

The Engineering notebook is an ongoing electronic journal of your activities on the project. Content that needs to be covered in this document includes (but not limited):

- The specific component of the project you worked on (code, hardware component, etc.)
- The specific problem you solved
- The specific document you worked on
- The components you tested

Each week, we will randomly select one or two people per team and review their notebook for our evaluation. Therefore, there is a possibility that throughout the semester, your notebook will be evaluated 2-4 times. In addition, you will be required to submit your final version of the notebook on November 25th. *If the notebook is not available at the moment of random selection, it will be consider as a no submission*. These will be reviewed at the end of any class meeting times.

Initial Research

Friday, September 11, 2020 10:04 PM

KNN

- KNN stands for k-nearest neighbors
- Is a supervised machine learning algorithm that relies on labeled input data to learn a function that produces an appropriate output when given new unlabeled data.
- The KNN algorithm assumes that similar things exist in close proximity.
- Calculations
  - Distance between two points
    - C = $\sqrt{a^2 + b^2}$
    - C=$\sqrt{(x_A - x_B)^2+(y_A - y_B)^2}$
- The KNN Algorithm
  - Load the data
  - Initialize K to your chosen number of neighbors
  - For each example in the data
    - Calculate distance between the query example and the current example from the data.
    - Add the distance and the index of the example to an ordered collection
  - Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
  - Pick the first K entries from the sorted collection
  - Get the labels of the selected K entries
  - If regression, return the mean of the K labels
  - If classification, return the mode of the K labels
- Note:
  - As we decrease the value of K to 1, our predictions become less stable.
  - As we increase the value of K, our predictions become more stable due to majority voting / averaging, and thus, more likely to make more accurate predictions (up to a certain point).

From <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>

SVM

- SVM stands for Support Vector Machines
- Is a supervised learning models
- SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.
  - The objective of applying SVMs is to find the best line in two dimensions or the best hyperplane in more than two dimensions in order to help us separate our space into classes.
- Looks at the extremes and draws a decision boundary (aka hyperplane)
- D- and D+ the section in the middle (0) is the margin or hyperplane

Vision Statement/Meeting 9/12

Saturday, September 12, 2020 11:58 AM

Vision Statement

To create a system that provides a patient with representative faces of different nose types that align with the measurements of their face, therefore allowing the patient to select which nose is attractive for their rhinoplasty procedure.

I created - vision statement

Notes: Faces similar to their own / representative different type of noses and select which one is attractive will fit on a face that looks like your face

http://openimaj.org/tutorial/intro-facial-feature-extraction.html

Stand up 9/15

Tuesday, September 15, 2020 2:31 PM

Trying to

Doing now:

Me & Chris - working up uploading photo using python script, get that photo and display it to you using script

We are not trying to pick most attractive nose

We want face that matches the patients face. We want to show these options to the customer.

Database w/ actors/models - **not our main goal to define beauty'(Akbas)**

We just want to find bodies that match (jacob)

Akbas to do - send us some more librarys
3d scans from web application to use as our patient

2d face scan (input) the output needs to be matching faces

Backlog
We made changes for the app that we want to do application through

We are using python app

What the product will be at the end of the semester?(Akbas -

Display 5 faces to show patient.
These noses look good (select 2?_

Say they select 1 we should be able to tell what changes need to be done to meet that selected nose.

*****We need to tell them what will be different (size - space from bridge,)

Summary of how its different - giving hints to the surgeon to tell them how to change the users nose to the new one.

(the difference between the n

Python Scripting

Friday, September 18, 2020 7:32 PM

Task 1: create python script
script 1: name file yourself and save to my repository file location
Script 2: don't ask user for name it saves a s default and pushes it to the
Note: No longer needed to do script 2 (9/18/20 and 9/19/20)

Useful links:

https://www.geeksforgeeks.org/python-pil-image-open-method/

Python Scripts

Created on Wed Sep 16 09:39:52 2020

@author: Vjord

script 1: name file yourself and save to my repository file location
Script 2: don't ask user for name it saves a s default and pushes it to the

note: repositoryjo-victoria/imagedatabase

"""

"script 1"

```
#!/usr/bin/python
import cgi, os
import cgitb; cgitb.enable()
form = cgi.FieldStorage()
# Get filename here.
fileitem = form['filename']
# Test if the file was uploaded
if fileitem.filename:
   # strip leading path from file name to avoid
   # directory traversal attacks
   fn = os.path.basename(fileitem.filename)
   open('/tmp/' + fn, 'wb').write(fileitem.file.read())
   message = 'The file "' + fn + '" was uploaded successfully'
else:
   message = 'No file was uploaded'
print(message)
```

```
# -*- coding: utf-8 -*-
"""
Created on Fri Sep 18 19:58:39 2020

@author: Vjord
"""

# "Base code to open an image at a fixed location"
# from PIL import Image

#read the image
# im1 = Image.open(r"C:\Users\Vjord\Pictures\avionlogo.PNG")

#show images
# im1.show()


"code to allow user to select a photo"
import cv2
import os

def load_images_from_folder(folder):
    images = []
    for filename in os.listdir(folder):
        img = cv2.imread(os.path.join(folder,filename))
        if img is not None:
            images.append(img)
    return images
```

JSON Files Task

Monday, September 21, 2020 5:11 PM

Task: Open JSON files and find the functionality
GitHub Info: Research javascript files sent by akbas #11 (From <https://app.zenhub.com/workspaces/se-450-5f53121d05c784001ae8505a/issues/king-of-kong/machine-learning-nose-jobs/11>)

How to open JSON research
https://medium.com/@antonmedv/how-to-work-with-json-from--f5a10cbe2121
https://stackoverflow.com/questions/8553200/how-do-i-install-and-use-a-json-editor-in-eclipse
https://sourceforge.net/projects/eclipsejsonedit/
https://marketplace.eclipse.org/marketplace-client-intro?mpc_install=945

Features.json (sample script)

```
var fpoints = [
 {
  "id": "FirstPoint",
  "name": "1st Point",
  "imageFile": "feature_point_first_point-frontal.png",
  "description": "Mark any point needed to calculate a distance, an angle or a ratio.",
       "usedInMeasurements": ["Distance_First_Second", "Angle_First_Second_Third",
       "Ratio_First_Second_Third_Fourth"],
       "xVal": "",
       "yVal": "",
       "zVal": "",
  "type": "Points"
 },"usedInMeasurements": ["Angle_First_Second_Third", "Ratio_First_Second_Third_Fourth"],
       "xVal": "",
       "yVal": "",
       "zVal": "",
  "type": "Points"
 },
```

Description:

- Features.json
  - Labels:
    - Id - same as name
    - Name - same as id
    - imageFile - image name/ can classify what position the photo is in.
    - Description - Mark any point needed to calculate a distance, an angle or a ratio
    - usedInMeasurements - values for distance, angle, and ratio
    - xVal
    - yVal
    - zVal
    - Type

- Female_Head_3-2006
    - Contains data/info for the features.json file
- Measuremaents.json
    - Filled in information
    - Formula to calculate: Pythagorean's theorem. C^2=a^2 +b^2

Standup 9/22

Tuesday, September 22, 2020 2:33 PM

W/ Akbas

Vision statement critique:
Add to the vision statement something that relates to:

- What changes need to be done on this version
- What nose they can select
- Since you have measurements you can say you can

Calculate difference on the nose bridge
Goal for this semester - what we find about the nose is calculate measurements and print them

Need to match proportions of the face (like generating a whole new image)

Suggestion (Akbas) : give option on diff faces

The faces look alike, but there will be proportion differences.

Just want draft of changes that need to be done. [Akbas]

Product Backlog critique

- He only sees 4 tasks as backlog items
- Backlog needs to show everything we are planning to do
- Then look at a few issues
- Github tutorial - need to put what is on which was sprint
- Need to fix backlog and show how to display it.

Anton

- Face clustering

Chintan

- Cv2
- pip3 install opencv-python

Standup 9/24

Thursday, September 24, 2020 2:33 PM

67-8 = want to match the features and not the nose (== total of 59 features to detect items)

Match points that they have and put them in the json files (****DO THIS***)
JSON files contain the point

What measurements we take will match those in the measurements.json file
8 (features on nose)
59 (on general face)

Standup 9/29

Tuesday, September 29, 2020 2:32 PM

Chintan
- Able to spit out code for each face in CSV file for 1600 pictures

Victoria
- JSON
  - Helpful links:
    - https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON#:~:text=JavaScript%20Object%20Notation%20(JSON)%20is,page%2C%20or%20vice%20versa).
    - https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Basics

Jacob
- Nothing now, will integrate stuff to python

Anton
- Created algorithm

Chris
- Working/researching algorithm
- Learning how to do python app

What will we show during demo (Akbas)?
- Show dlib/algorithm to show how measurements will be captured
- Consider as --- skeleton
- Okay with seeing what we want to show
- For the sprint we were working on this and we want to show this
- After this sprint he will see facial points

We changed structure of design class
- This semester he wants us to focus on implementation
  - Focus on the product

Best tool is ratios

After Team Meeting
- I'll work on SDS
- Jacob and Chris will look into the python app
  - Are assigning themselves it in github
- Anton will host demo on his computer for Thursday's demo

SDD

Wednesday, September 30, 2020 11:40 AM

Updated the following sections of the SDD

1. INTRODUCTION

2. Purpose and Scope

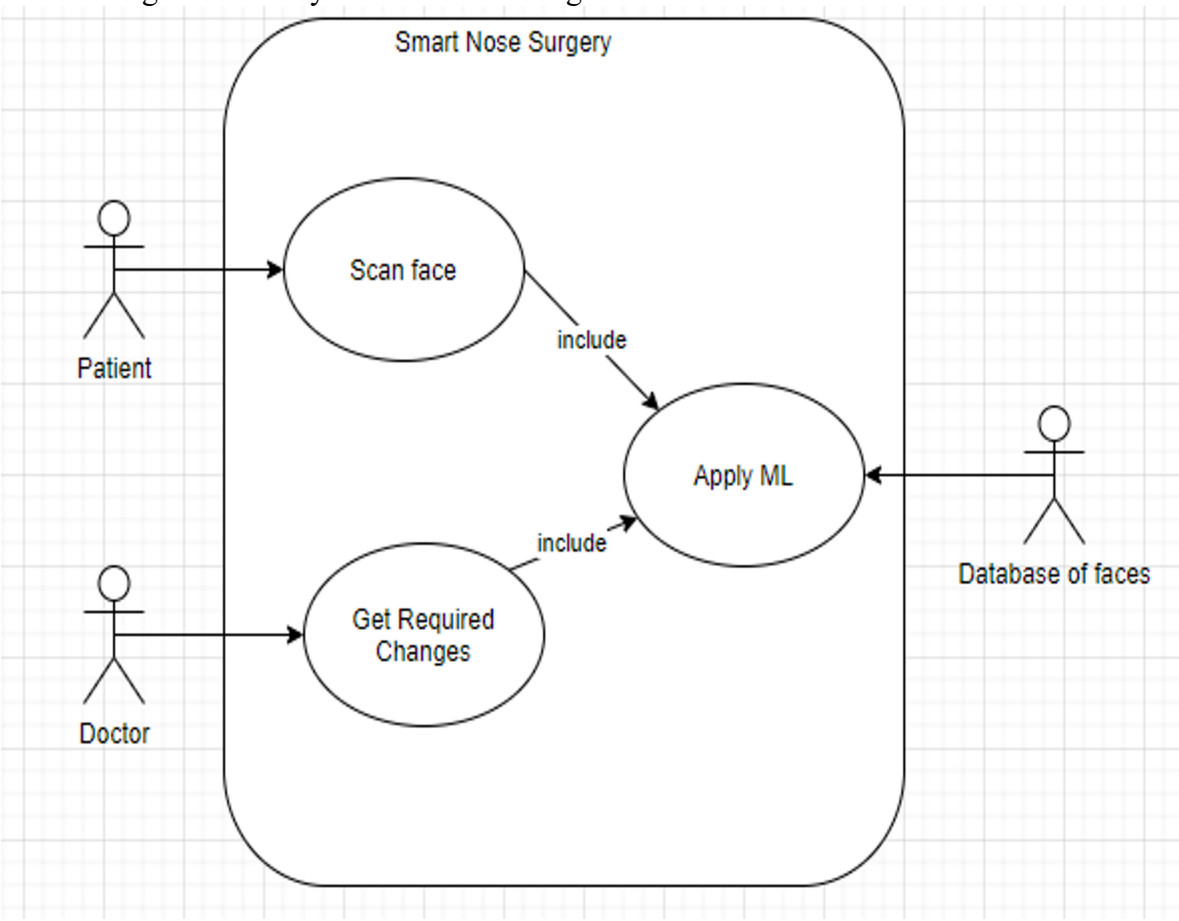
This document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, and external interfaces for Machine Learning Algorithm for Rhinoplasty (M-LAR) system.

3. Project Executive Summary

This section provides a description of the Machine Learning Algorithm for Rhinoplasty (M-LAR) system from a management perspective and an overview of the framework within which the conceptual system design was prepared.

4. System Overview

The system will provide a patient with representative similar faces with different nose types that align with the measurements of their face, therefore allowing the patient to select which nose is preferred for their rhinoplasty procedure. Below is the high-level architecture and context diagram of the system's overall design.



5. Document Organization

This document is designed to describe the structure of the Machine Learning Algorithm for Rhinoplasty (M-LAR) system.The following sections will provide information on what the product does, limitations, interactions, interfaces, hardware and software designs, and security.

6. Project References

1. System Requirements Specification

7. Glossary

1. M-LAR - Machine Learning Algorithm for Rhinoplasty (M-LAR) system
2. "Patient" and "User" are used as interchangeable terms.
3. KNN - "k-nearest neighbors", a machine learning algorithm that, when given a test point represented by a point on the coordinate plane, will find the k points in the data set, also in the coordinate plane, that are nearest to the test point.

4. SYSTEM ARCHITECTURE

This section describes an overview of the hardware and software architecture for the M-LAR system and subsystems.

8. System Hardware Architecture

In this section, describe the overall system hardware and organization. Include a list of hardware components (with a brief description of each item) and diagrams showing the connectivity between the components. If appropriate, use subsections to address each subsystem.

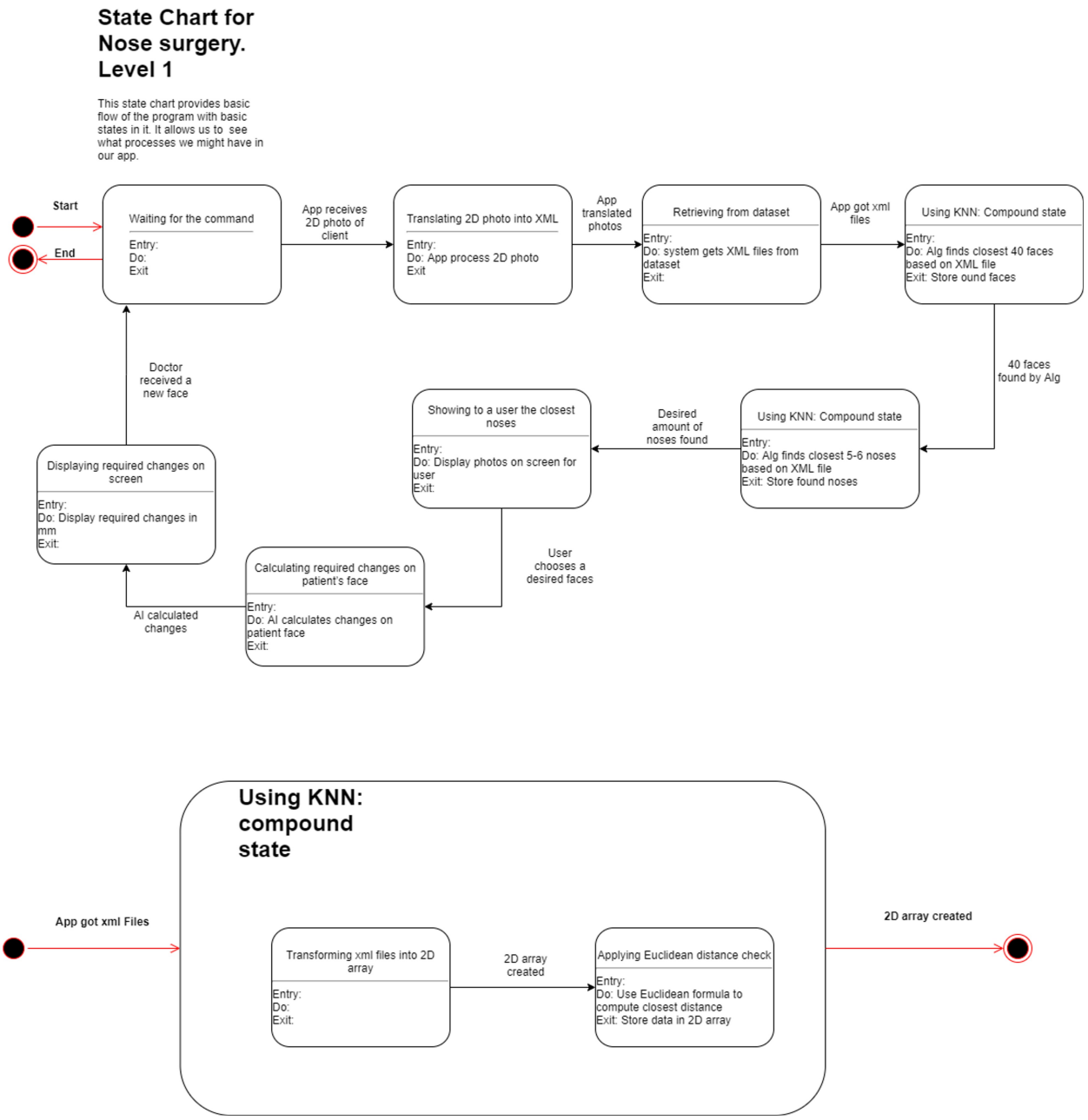
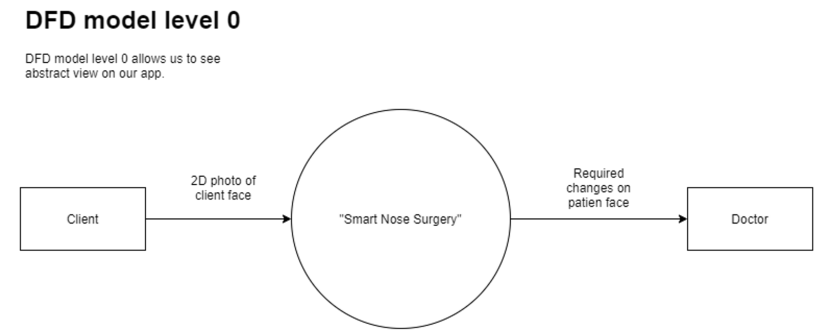


Figure 1: State Chart Diagram

Figure 1 describes what states and conditions will be cycled through in the system.



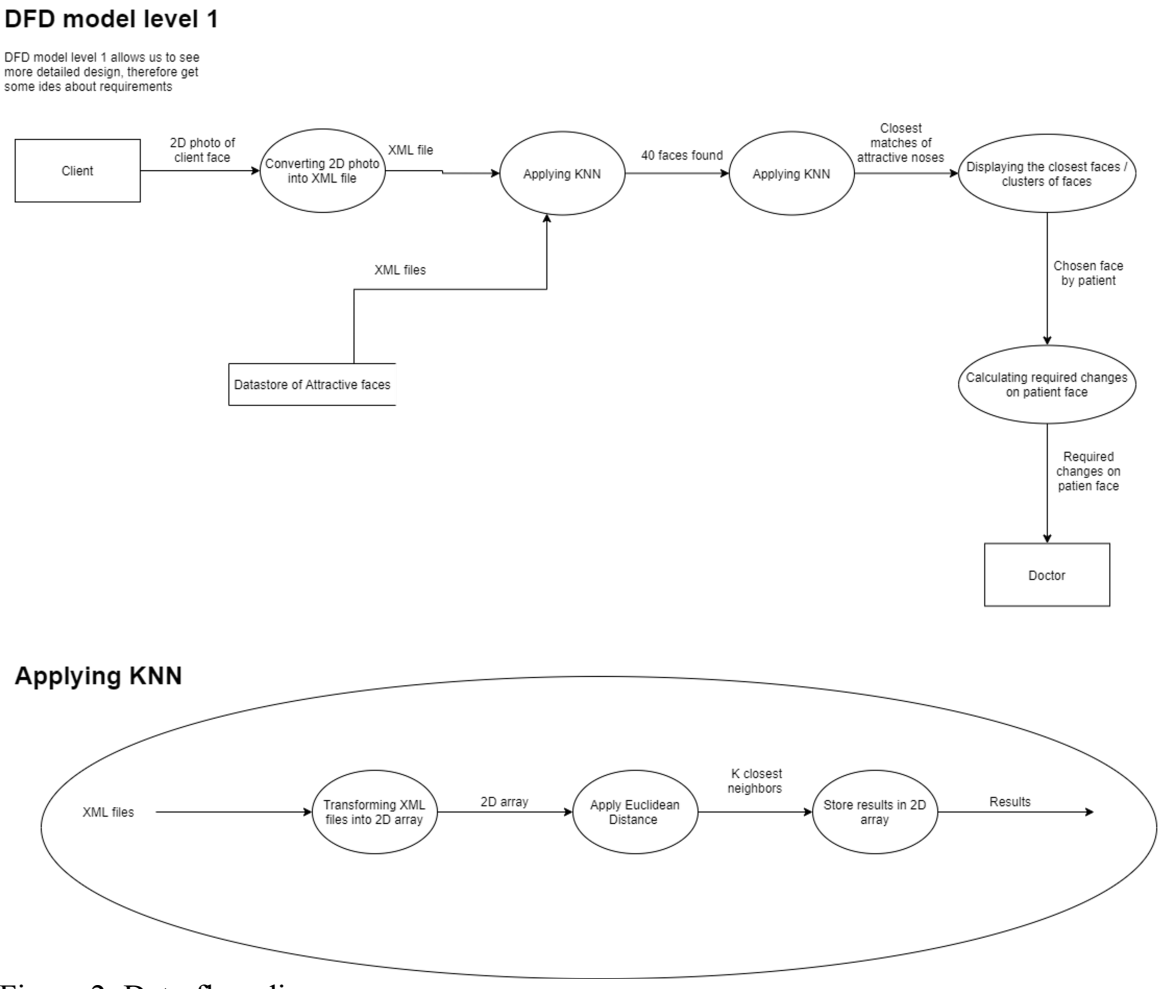


Figure 2: Data-flow diagram

Figure 2 describes the transfer of data in the system between the client and the doctor.

Demo Day - class

Thursday, October 1, 2020 2:38 PM

Notes:

Doesn't like listing 40 faces

Achieve dimensionally -
Update to 3D -
40 faces but can update
Looking at Euclidean distance is nice, now lets look at clusters.

Similar faces in groups, when you are proposing changes it doesn't have to go from location x to

After meeting
Chris - likes

Stand up

Tuesday, October 6, 2020 2:22 PM

What will the product be at the end of sprint 2

- Goals: 1)Backend algorigthm, you give an algorithm and it does knn. Assign Z score based on mean. 2)Python app - display on grid
- In a demo what will we see in the sprint
    - [Write here in the notebook]
    - Do scenario of what will happen during this time
    - Here is what we are planning
    - Rough outline of what we're doing
    - Share link w/ Akbas
    - Z-Score of the face
        - Statistical method/numerical value method

Demo Sprint 2
Steps:

- Log into designed python app (save for sprint 3)
- Ask user for a photo
    - Options:
        - Program access users' computer
        - User uploads a photo
        - ~~Define what position we would like the photo to be~~
            - Front facing
                - Profile(distant future maybe)
- Program processes the user photo
    - During process the system
        - Run dlib face detector
        - Run statistical analysis (Z-Score?) - give users' face a z-score
        - Uses KNN to define the points for nearest neighbor
            - Faces passed down for clustering
- Display a grid of faces that have been recommended for the user

  Demo KNN and clustering working on its on.

Show how many faces KNN recommends
Run script for uploading photo
Face detector using dlib
Output through KNN
Show grid of all of the faces
"cluster algorithm"
Give photo --> see representative faces

Later task - redo csv files

After standup meeting

A - look for svm, meet with akbas,

Create a program that does subtraction
Logging in and security

Standup 10/12 w/Tayler

Monday, October 12, 2020 12:07 PM

SRS:
Separate table of contents
But who did each section and put in change log
Definitions and acronyms in alphabetical order
Graphs were find
We missed a section from template
Add back shalls
Rework on formatting (numbering vs - bullets)

SDD:
Put caption under our diagrams
Bullet points in glossary

I did sprint demo outline. Todo: begin 3D faces

Not edit image but return calculations

Standup 10/8

Thursday, October 8, 2020 2:24 PM

Stick to KNN and SVM for sprint 2

Didn't like that the users

SRS

Wednesday, October 14, 2020 11:14 PM

Updated SRS for Version 2

Updates includes

- Formatting
    - Bulleting
    - Numbering
- Wording "the system shall"
-

Standup 10/20

Tuesday, October 20, 2020 2:18 PM

Class - goals for next sprint, video describe project

Video is replacement of live demo

Question: do we need to readapt out demo outline

Stand-up

Demo is done, fixing/fixed SRS, 3D library

Chris - satisficed with how the library look. Got the app working to take the picture.

Will it be ready for the demo?

- Yes - can use KNN

Anton - faces rearrange

Show how to demonstrate its working - test plans

Victoria -

Grade scale from Jafer
Consider video- spell out what we have done since sprint 1.

Instead of KNN algorithm - do virtual piece

Chintan - faces

Jacob - going into todays meeting what to work on next

DEMO IS NEXT THURSDAY
If we want to do a dry run let akbas now

But engineering notebook on wiki page??

After Meeting 2:45 - :

Main things we need to do
Clustering

Easy way to do K #

Redo the csv (height/width/nose point)

Get knn algorithm on app

Clustering - get clustering to work

Facial feature.py
Get csv reader from testerapp. (put two together

```
def calculateFacialSize(self):
    width = self.facialCords[16][0] - self.facialCords[0][0] # top right(17) - top left(1)
    height = self.facialCords[8][1] - self.facialCords[27][1] # bottom center(9) - top center(28)
    return (width , height)
```

Standup 10/22/20

Thursday, October 22, 2020 2:16 PM

Regular Class Meeting

- Sprint 2 demo is next Thursday
- Peer evaluations due Oct.28
- Put engineering notebook on Github
- SDD/SRS full document due

Standup

- Chintan
  - Thing not working
  - Need to
- Chris
  - Looking for images/
- Jacob
  - Trimming down csv
- Anton
  - Test cases
- Victoria
  - CSV
- Notes for Akbas:
  - Demo next week
  - Any problems in the team try to solve process for fixing it
  - Try to equalize/fair workload
  - Wants a trailer of the demo on Tuesday

Afternoon:

- Talked about ratios last meeting
  - Issues about using width and height originally
- How are we doing the csv files
  - By names - ratios
  - Run only one to calculate data - KNN
  - "one line code"
- Chris meet with Tayler to push back epic
- What's left
  - Demo trailer - chris needs to finish
  - CSV - chintan fix
  - Jacob - read the file
  - Trailers - make presentation (Victoria - contact Anton)

- Soon as csv is done - need to fix look at stuff
- Diagrams
- Ratio compute (jacob)
- Main app (chris)

Team Meeting

Saturday, October 24, 2020 3:50 PM

Time 3-3:50pm (10/24/20)

[UPDATE THIS LATER]

Send

Trailer – show a clip of the app

Send video and do voice over it

Capturing works in cv2 (open camera and capture image

Do ppt (Same type of ppt and with diff models)

Show app running

Why push back issue

Clean something up

Anton fixing code

graph

Meeting with Tayler and Akbas

Monday, October 26, 2020 10:59 AM

Date: 10/26/20
Time: 11am - 11:15

Discussion:

- process of moving an issue in the sprint backlog into sprint three, specifically issue #14
- Demo for Thursday

Notes:

Issue #14 - developing and implementing a clustering algorithm.

We spent this sprint doing KNN and ratios (optimization)

Akbas: Good thing would be to explain plan to finish that task during the demo on Thursday
*We are okay to push it back to sprint 3

Thursday Demo:
Prepare ppt presentation
Chris version/Anton demo

My task: Do demo voice over

If we display female faces, needs to upload a female face {future thing we need to have/consider}

Are there any steps in between to show how we got to these five faces - make it clear - figure

Akbas - likes our interface

Stand up 10/27

Tuesday, October 27, 2020 2:40 PM

Class:
Survey: bit.ly/coemidcourse
SDS & SRS & Demo due Thursday

Standup

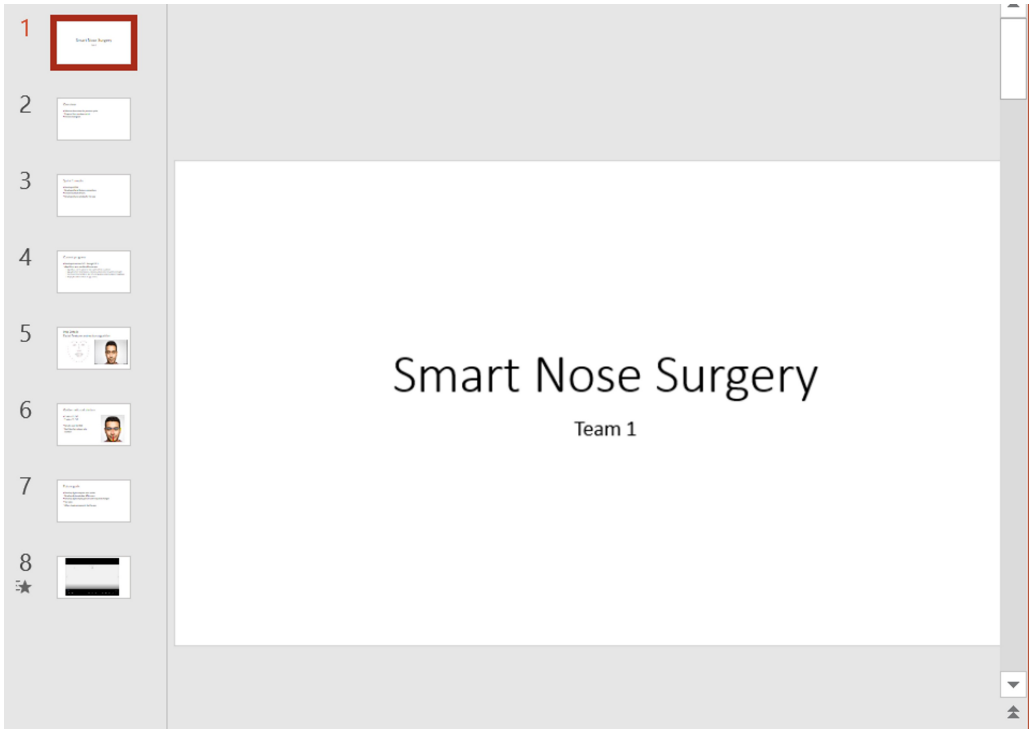
Victoria
- SDS feedback update
- Demo finalization

Chintan
- Face ratios
- *golden ratio

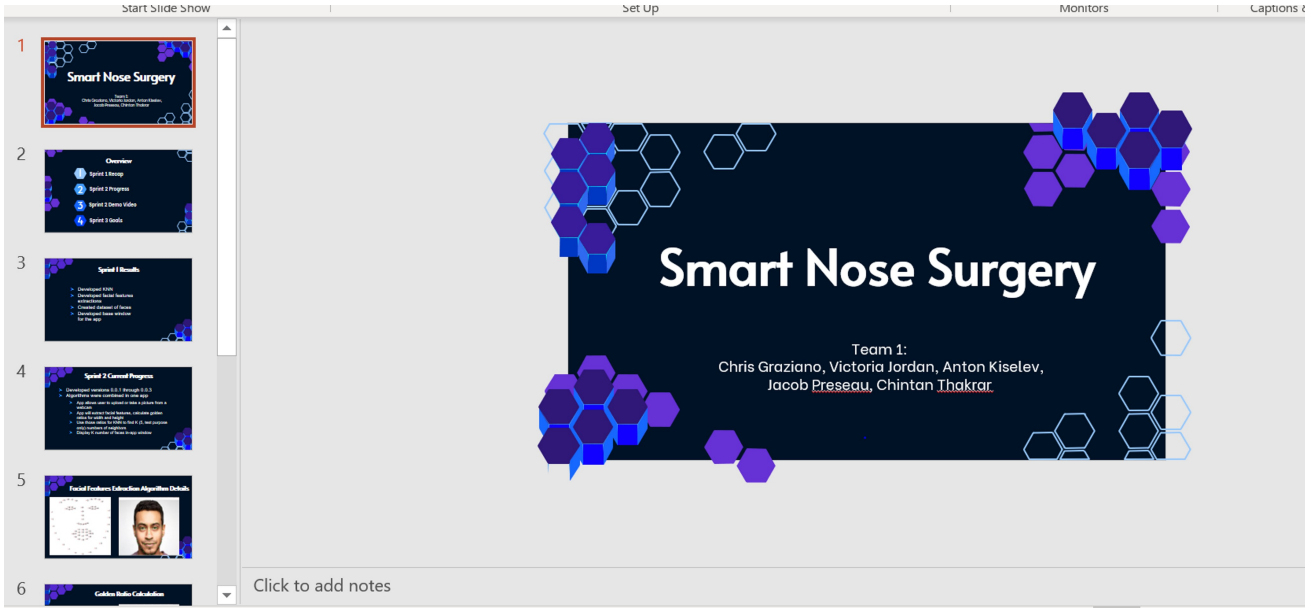
Group Meeting (2:46pm - 3:13pm
Review sprint 1, what we've done through sprint 2, voice over video for chris' video

Demo Contribution (11:00pm - 12:00am)
- Demo voiceover
  - Edit using Zoom


zoom_1

- Revamp original powerpoint for demo
  - Style, wording, inserting demo
  - Original:
    - 

  - New:


Start Slide Show
Set Up
Monitors
Captions
1
Smart Nose Surgery
2
Overview
1 Sprint 1 Recap
2 Sprint 2 Progress
3 Sprint 2 Demo Video
4 Sprint 3 Goals
3
Sprint 1 Results
4
Sprint 2 Current Progress
5
Facial Features Extraction Algorithm Details
6
Golden Ratio Calculation
Smart Nose Surgery
Team 1:
Chris Graziano, Victoria Jordan, Anton Kiselev,
Jacob Preseau, Chintan Thakrar
Click to add notes

10/29/30

Thursday, October 29, 2020 2:17 PM

Class:

- Peer Evaluations were due Yesterday
- Deliverables (PPT, SDS v2, SRS v2) put into the deliverables folder

Demo Notes

After Demo Recap

- Work on clustering
- 3 - clustering
- 2 - ____
- Calculating differences

Chintan - nose clustering

Quick Meeting 11/2

Monday, November 2, 2020 4:46 PM

Discuss more about modelling the nose to get more points. Realized we need angles from the side
Need a new dataset
Standardized in terms of distance and size

Categories	Nasal shape	Nasal index
Hyperleptorrhine	Very Narrow	40 – 54.9
Leptorrhine,	Long and Narrow	55 – 69.9
Mesorrhine	Moderate shape	70 – 84.9
Platyrrhine	Broad and Short	85 – 99.9
Hyperplatyrrhine	Very broad/Wide	≥ 100

$$\text{Nasal Index} = \frac{\text{Nasal Breadth}}{\text{Nasal Height}} \times 100$$

Reference

https://www.researchgate.net/figure/The-mean-nasal-height-values-among-Bini-male-and-female-children_fig1_332912694

IDEA

Take frontal picture and side picture to make calculations more precise

Chris - ratio calculator

Clustering and ratios todo

Research Clustering

Tuesday, November 3, 2020 9:37 AM

Article 1:

Abstract: Determining Facial Beauty Using Artificial Intelligence

From <https://journals.lww.com/prsgo/Fulltext/2018/08001/Abstract__Determining_Facial_Beauty_Using.228.aspx>

Notes:

- METHODS: Photographic analysis consisted of 1192 pre- and post- rhinoplasty photos of women as well as 139 photos of actresses, all of whom are listed as the most beautiful women of all time per IMDB. All photos are frontal shots with the face in a neutral pose. Using a pre-trained deep convolutional network algorithm, the photos were embedded with 128 vectors for clustering analysis. Phenotyping analysis of the pre-rhinoplasty photos was conducted via parameterized Gaussian mixture models optimized via Bayesian Information Criteria (BIC) for expectation-maximization. Furthermore, facial averages were generated via a Delaunay triangulation using the 68 landmarks and facial similarity scores were computed via similarity score of two faces by computing the squared L2 distance between their representations.
- Key Points -
    - Using a pre-trained deep convolutional network algorithm
    - 128 vectors for clustering analysis
    - Phenotyping analysis of pre-rhinoplasty

Stand Up 11/3/20

Tuesday, November 3, 2020 2:16 PM

Class Announcements

- Test Plan V1 is due 11/10 (next Tuesday)
- SRS/SDS final due 11/19
- Final Demo & Test Plan Final due 11/24
- Test Plan
  - Should have started on test plans from having documents on requirements and design
  - See what tests were given and which tests were passed
  - Expecting a longer document
- Bonus Point
  - Survey
- Grades
  - Receive grades for sprint 2
    - Engineering notebook
    - Peer evaluation
    - Demo
- Goal for today
  - Sprint retrospective
  - Define goals for sprint 3

Stand up

- We should do retrospective
- Chintan
  - Him and tony have done nose work
  - Points to categorize the nose
- Tony
  - Height and length of the nose
  - Akbas reminder - when looking for similarities
- Victoria
  - Research on clustering algorithm
  - Paper on using postoperative noses
- Chris
  - Working ahead to program to calculate user nose (Euclidean distance)
  - Turn labels into buttons
- Jacob
  - User selection

My TODO:
Send paper to chintan
Fix test plan

Test Plan

Thursday, November 5, 2020 11:19 AM

Start time: 11:00am - 11:30pm

Goal: Template revision - updating Anton's original version into the template provided by instructors

Merging document

Saturday 11/7/20
Start time: 4:40pm
End Time: 6:45pm
Tasks

2. Functional Scope

Defined characteristics for our scope

3.2. System Testing Entrance Criteria
Defined options for testing readiness

3.3 Testing Types
3.3.2 Functional Testing
Stating that are requirements are found in our SRS

3.4 Suspension Criteria and Resumption Requirements
Fixing the wording of things

5. Traceability Matrix & Defect Tracking 3
5.1 Traceability Matrix 3
Adding in the full requirements produced from our SRS to this requirements table.

5.2 Defect Severity Definitions
Updated table

SmartNose…
(1)

Tuesday 11/10
11:05 -

Update the traceability table

Stand up 11/5/20

Thursday, November 5, 2020 2:30 PM

Standup

What are our main tasks

Anton
-found new paper that there are 7 clustering for noses

Chintan
-assist with categorizing … with Anton

Jacob
-face landmark file (looking into opening this)
--> help with overlay of the face

Chris
-since we can't get face predictor open at a roadblock since Tuesday.

Victoria
-Test plan

Akbas: Since we meet the other day I don't have anything left to ask
-->Anton - showed UI development

Stand Up 11/10/20

Tuesday, November 10, 2020 2:18 PM

Class

- Test plan due today
- SRS Final and SDS Final is due 11/19
- Final test plan due 11/24
- UPDATE: Final demo and product delivery due 12/9 between 8 -10am
- UPDATE: The documents due by thursday of study week (11/26 - test plan/SRS/SDS)
- For demo - everyone demos in front of the entire class during this time

Standup

- SRS/SDS feedback given today
- Chris
    - Working on display like anton had in balsamic
    - Making tk frames into grids
    - HICCUPS:
        - Clustering was one level
        - Anton
            - Being working on the faces and clustering since Sunday
                - Akbas
                    - Put analysis in our documentation
- Jacob
    - Working on dummy program
- Victoria
    - Finished v1 of test plan
    - Shift to new task

IDEA: LOOK FOR SYSTEMS THAT CAN PULL FACIAL POINTS FOR PEOPLE'S PROFILE PHOTOS

Implement clustering frame?

Meeting w/Chris

Thursday, November 12, 2020 12:12 PM

Meet with Chris
Change KNN fram to a cluster fram

Knn - nearst neighbor and print results

Deposit

Tony structured the clustering algorithm

Make an array

Clusterarray = clusteralg(first40faces)

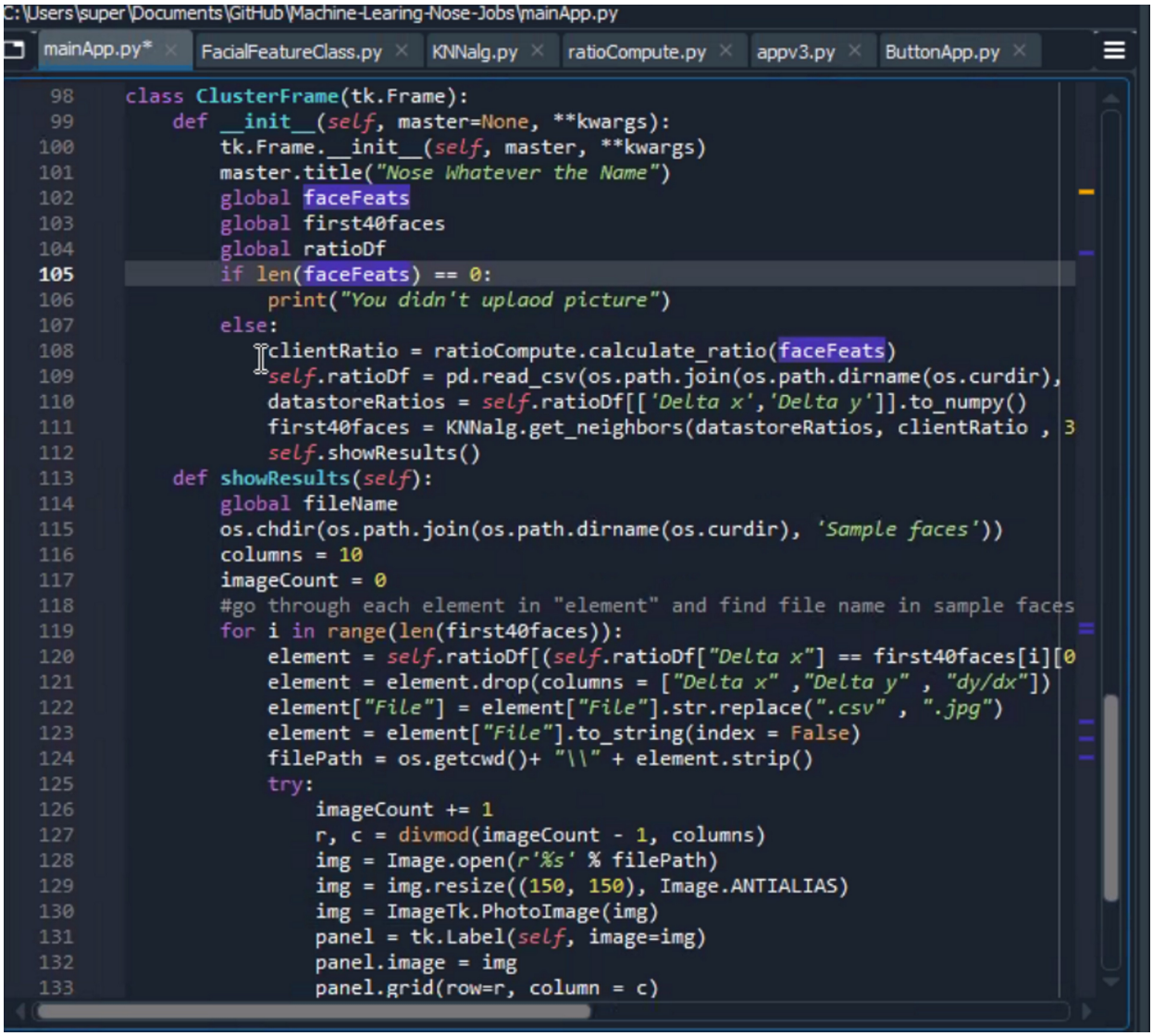
For I in range(len(cluster array)

Display the results from cluster array
-want groups of 7 - 7 faces

Could probably hard code

2/3 rows 3 colums
Row 1 - full
Row 2 - full
Row 3 - 1 face in column 2

In each column go put in a button and each face

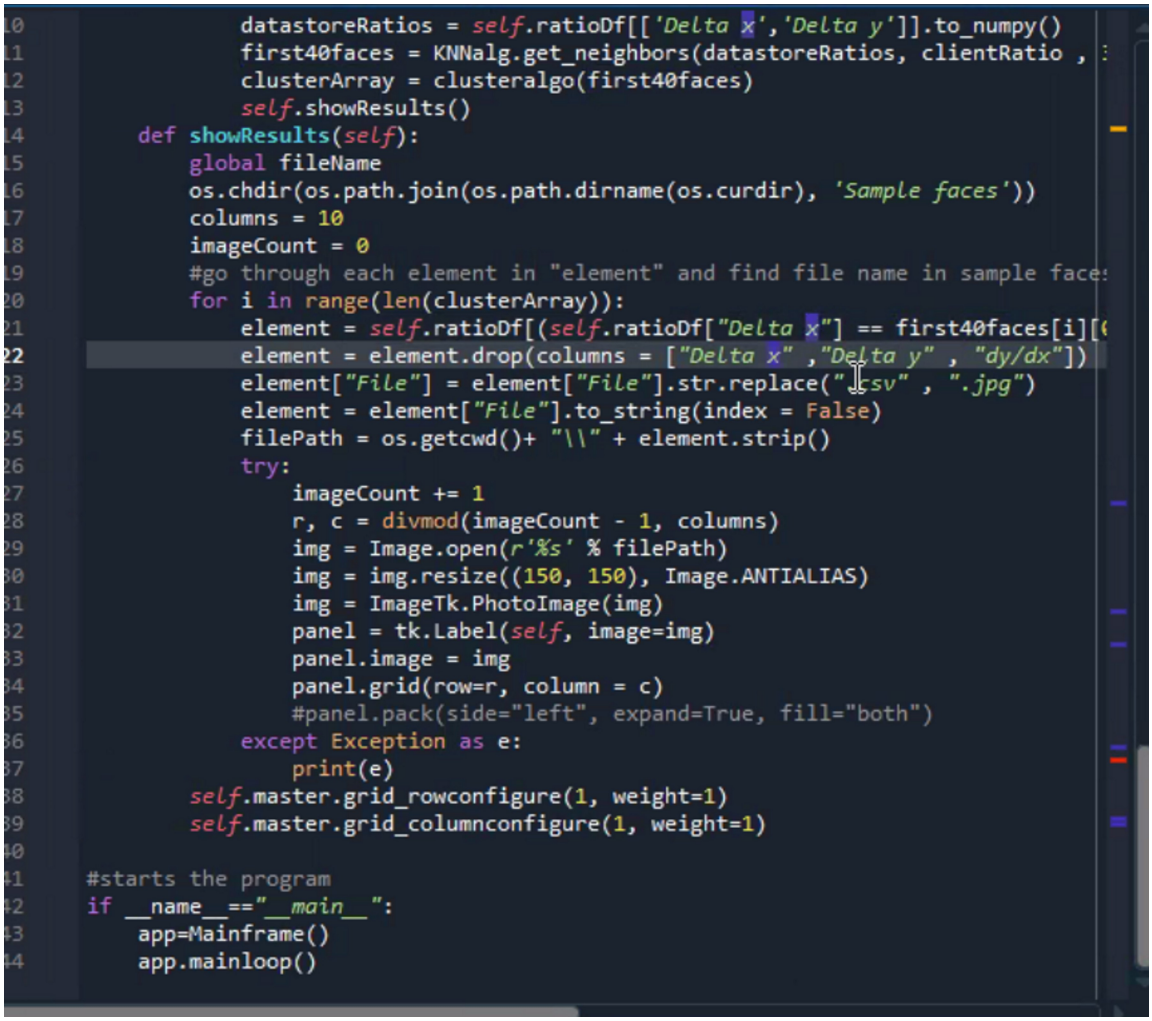


Element is what he is displaying here
Begin 1st 40 faces and begin with ratios

May be able to keep element

Kt frame grid

Button chose cluster # (svg or means)



```
            datastoreRatios = self.ratioDf[['Delta x','Delta y']].to_numpy()
            first40faces = KNNalg.get_neighbors(datastoreRatios, clientRatio ,
            clusterArray = clusteralgo(first40faces)
            self.showResults()
    def showResults(self):
        global fileName
        os.chdir(os.path.join(os.path.dirname(os.curdir), 'Sample faces'))
        columns = 10
        imageCount = 0
        #go through each element in "element" and find file name in sample face
        for i in range(len(clusterArray)):
            element = self.ratioDf[(self.ratioDf["Delta x"] == first40faces[i][
            element = element.drop(columns = ["Delta x" ,"Delta y" , "dy/dx"])
            element["File"] = element["File"].str.replace(".csv" , ".jpg")
            element = element["File"].to_string(index = False)
            filePath = os.getcwd()+ "\\" + element.strip()
            try:
                imageCount += 1
                r, c = divmod(imageCount - 1, columns)
                img = Image.open(r'%s' % filePath)
                img = img.resize((150, 150), Image.ANTIALIAS)
                img = ImageTk.PhotoImage(img)
                panel = tk.Label(self, image=img)
                panel.image = img
                panel.grid(row=r, column = c)
                #panel.pack(side="left", expand=True, fill="both")
            except Exception as e:
                print(e)
        self.master.grid_rowconfigure(1, weight=1)
        self.master.grid_columnconfigure(1, weight=1)

#starts the program
if __name__=="__main__":
    app=Mainframe()
    app.mainloop()
```

```
def euclidean_distance(point1, point2):
    return np.sqrt(np.sum((point1-point2)**2))


# In[132]:


def defineCluster(facial_coordinates):
    #height_of_nose = euclidean_distance(facial_coordinates[0], facial_coordina
    #top_cord = facial_coordinates.loc[0]["Y"]
    #bottom_cord = facial_coordinates.loc[6]["Y"]
    height_of_nose = euclidean_distance(facial_coordinates.loc[0]["Y"], facial_
    width_of_nose  = euclidean_distance(facial_coordinates.loc[12]["X"], facial_
    nasal_index = nasalIndex(height_of_nose,width_of_nose)
    if(nasal_index <= 39.99):
        print("Overly narrow nose")
    elif(nasal_index <= 54.99):
        print('Very narrow nose')
    elif(nasal_index <= 69.99):
        print('Narrow nose')
    elif(nasal_index <= 84.99):
        print('Medium nose')
    elif(nasal_index <= 99.99):
        print('Broad nose')
    elif(nasal_index <= 114.99):
        print('Very broad nose')
    else:
        print("Overly broad nose")


# In[133]:
```

Need a representative face for each cluster

What I'm trying to do - doesn't exist

Stand Up 11/12/20

Thursday, November 12, 2020 2:15 PM

Class:

- Extended deadlines
- Deliverables

Stand Up

- Anton
  - Work on my diagrams
  - Working with chintan on new data set
  - Combine narrow noses
- Chintan
  - Working with anton on data set -
- Jacob
  - Working on calculating changes
  - 194 predictor works (needs to know)
- Chris
  - Grid stuff
  - Helping Victoria with clustering
- Victoria
  - Implement clustering frame
  - Show clustering of face for app
  - Issues: part of the clustering algorithm isn't done
  - Another frame to the app

After Meeting:

- Chris working on button location
- Doesn't have to be mathematically accurate
- Clustering Algorithm
  - We have 40 faces from KNN
    - It already has its clusters (broad nose, medium nose etc)
  - I can take file names of the cluster and use that in the area
  - CSV reader inside main app
  - "if clusters has any faces" = pull faces
  - Only make frames for the files that have those characteristics

- Make the csv reader
- Reader the third column
- Make an array
- Chris - can make frames for the array
- Chintan - recommend using dictionaries

Stand Up 11/17/20

Tuesday, November 17, 2020 6:20 PM

Reference Notes from last stand up meeting

Stand Up 11/19/20

Thursday, November 19, 2020 2:20 PM

Class:
No major updates

Stand up:

Anton
- Try different implementation
- Jaw line/face height

Chintan

Chris
- Entry box for user to specify face
- Working with Victoria on csv

Victoria
- Change the csv

What's left
- Prepare demo
- Documentation
- Final result output

  Look at points on the nose and show an image of the points on where they should be. Overlays the nose

Akbas

Reread our previous documentation and double check edits, add in additional features

Pixel - this semester
Real measurements - next semester

11/21/20

Saturday, November 21, 2020 9:30 AM

Tasks: Reorganize array to make corresponding CSV file
9:15am - 9:52am

CSV sample

File	Cat	val
51079.jpg	Broad nose	91.66667
04732.jpg	Broad nose	86.36364
55579.jpg	Broad nose	88.09524
22265.jpg	Broad nose	89.3617
24728.jpg	Medium nose	80
69523.jpg	Medium nose	82.6087
64753.jpg	Medium nose	80.76923
69752.jpg	Medium nose	84.78261
11460.jpg	Medium nose	84.31373
09833.jpg	Broad nose	91.83673
69693.jpg	Medium nose	77.08333
66245.jpg	Broad nose	95.55556

The categories for the final arrays - {

Broad Nose
Medium Nose
Narrow Nose
Overly Broad Nose
Very Broad Nose

Code updates
File Name: mainApp.py

```
class ClusterFrame(tk.Frame):
    def __init__(self, master=None, **kwargs):
        tk.Frame.__init__(self, master, **kwargs)
        master.title("Nose Whatever the Name")
        global faceFeats
        global first40faces
        global ratioDf
        filename = "golden_ratio.csv"
        clusterBroadNose = []
        clusterOverlyBroadNose = []
        clusterVeryBroadNose = []
        clusterMediumNose = []
        clusterNarrowNose = []
        if len(faceFeats) == 0:
            print("You didn't uplaod picture")
```

```
                    if row[3] == "Broad":
                        clusterBroadNose.append(row[3])
                    if row[3] == "Overly Broad":
                        clusterOverlyBroadNose.append(row[3])
                    if row[3] == "Very Broad":
                        clusterVeryBroadNose.append(row[3])
                    if row[3] == "Medium":
                        clusterMediumNose.append(row[3])
                    if row[3] == "Narrow":
                        clusterNarrowNose.append(row[3])
                    line_count += 1
 self.showResults(clusterBroadNose, clusterOverlyBroadNose, clusterVeryBroadNose, clusterMediumNose
 showResults(self,clusterBroadNose, clusterOverlyBroadNose, clusterVeryBroadNose, clusterMediumNose
 global fileName
```

Originals :

```
class ClusterFrame(tk.Frame):
    def __init__(self, master=None, **kwargs):
        tk.Frame.__init__(self, master, **kwargs)
        master.title("Nose Whatever the Name")
        global faceFeats
        global first40faces
        global ratioDf
        filename = "golden_ratio.csv"
        clusterOneArray = []
        clusterTwoArray = []
        clusterThreeArray = []
        clusterFourArray = []
        clusterFiveArray = []
```

```
                    else:
                        if row[3] == "very broad":
                            clusterOneArray.append(row[3])
                        if row[3] == "very broad":
                            clusterTwoArray.append(row[3])
                        if row[3] == "very broad":
                            clusterThreeArray.append(row[3])
                        if row[3] == "very broad":
                            clusterFourArray.append(row[3])
                        if row[3] == "very broad":
                            clusterFiveArray.append(row[3])
                        line_count += 1
        self.showResults(clusterOneArray, clusterTwoArray, clusterThreeArray, clusterFourArray, clusterFiveArray)
    def showResults(self,clusterOneArray, clusterTwoArray, clusterThreeArray, clusterFourArray, clusterFiveArray):
```

Stand Up 11/24

Tuesday, November 24, 2020 2:20 PM

Class
- Sprint 2 peer evals
- Make a video explaining the product and different videos - add this video link in your powerpoint for the class.
- During demo can go over code implementation
- Will wait until the beginning of the next semester to do a different project
- Recommendation: schedule an additional meeting with the customer (akbas) before dec.9th

Stand Up:
Victoria
- Update CSV frame (Buttons & check empty statements)

Any questions?
No

Tasks till Dec. 9
Chintan
- Doesn't specifically know (will check with the group)

Chris
- Finishing cluster frame implementation

Anton
- Scaling pixel (convert millimeters)

Jacob
- Same boat as Chintan
- Maybe do documentation

Victoria
- Work with Chris
- Do the csv frame

Akbas suggestions
- Set tasks timeline to know what we need to do by Dec.9
- Spend enough time practicing the presentation
- To answer Chris question - we all could continue on doing this project - then probably work on stuff during the break.
- What do we think about the last sprint? What could we do better? If we wanted to access our own team?
    - Chintan - we have a fantastic team. The way we specialize in something is good. We did a fantastic job in splitting the work. What could we do better? A slightly more even distribution of work.
    - Chris - our organization was ad hoc, could have been planned a bit better. If it really didn't work truly we would have stopped. What to do better - improved/managed the scrum board better. Could have been more efficiently.
    - Victoria - need more balance of work
    - Anton - fix scrum board

Chris - finish app this weekend
(would recommend - looking to distance frame - class that expects to receive an image)
Chintan - honing the algorithms
Anton & Jacob - finalize distance calculators, overlay faces, pixel to millimeters.
Victoria - documentations.

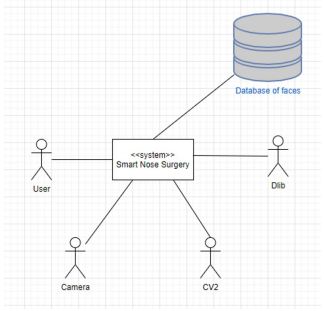
Documentation Review & Finalization

Thursday, November 26, 2020 7:02 PM

SDS (7:15pm - 9:15pm)

- Updating Sections
  - 1.2.1 System Overview (Rewrite)
  - 1.2.3 Future Contingencies
    - Addressed feedback questions (

      Akbas, M. Ilhan - What specific needs for this purpose?
    - Best fit that the needs originally mention would be for future goals
  - 2.2 System Software Architecture
    - Added section description for Figure 1



      - Figure 1 displays a context overview of the M-LAR system. The system consists of five (5) crucial components: Database of faces, Dlib, CV2, Camera, and User. The database of faces contains all of the images we have available for the facial ratio algorithm to pull 40 faces from. The database includes 2D frontal images of both men and women. Dlib assigns the facial coordinates to a given image. CV2 is used for image detection and capturing the users 2D frontal photo by accessing the users' computer camera. The user provides access to the camera and a canvas to pinpoint facial coordinates on as well as selecting a nose candidate from the nose clustering.
    - Expanded section description of Figure 2
  - 5.1 Interface Architecture
    - Removed old figure and descriptions
    - Updated description of new figure. (Renamed from figure 6 to figure 5)
  - Ordering figure numbers
  - Adjust table of contents

SRS (**11/26/20**: 9:20pm - 10:00, 10:30 - 11:24pm; **11/27/20**: 1:30pm - 2:48, 4:22, 5:05pm- 8:35pm)

- Update Sections
  - 3.2 Stakeholders
    - Added additional stakeholders (Clinic, health care provider, hospital, Pharmaceutical Company
  - Clarification on follow template feedback -

    "Check out the template for the formatting of the Functional and Nonfunctional requirements"
    - Using template table to fix the entire section 4 (system requirements)

No: <unique requirement number>
Statement: <the "shall" statement of the requirement>
Source: <source of the requirement>

Dependency: <list each other requirement on which satisfaction of this requirement depends. (May be "None")>
Conflicts: <list each other requirements with which this requirement conflicts. (May be "None")>
Supporting Materials: <list any supporting diagrams, lists, memos, etc.>
Evaluation Method: <How can you tell if the completed system satisfies this requirement? >
Revision History: <who, when, what>

- 4.1.1. The system shall allow a user to scan their face with a camera.
- 4.1.2. The system shall be able to transform the picture of the user's face into a set of coordinates representing facial landmark measurements.
- 4.1.3. The system shall contain coordinate files with an array of 67 variables that can be used to uniquely characterize each face in the database and that of the user.
- 4.1.4. The system shall ask a user to retake the photo in case coordinate detection fails.
- 4.1.5. The system shall be able to get the files representing faces from the database.
- 4.1.6. The system shall be able to apply the KNN algorithm on the files in the database.

Test Plan (11/27/20: 8:40pm - 9:00pm, 10:15pm -

- The feedback we received was for the wrong document.
  - Based off of Tony's initial test plan (File Name: Test_Plan_SE450.doc) instead of the detailed test plan we submitted (File Name: SmartNoseSurgery_SystemTestPlan_v1.docx) that matched the given template.
- Updating overall document based on common issues from our other documentation
- Update sections
  - 5.2 Defect Severity Definitions
    - Added Section description for clarity based on feedback "good to mention these are examples"
  - 4. Execution Plan
    - Defined section description
    - Moved table located in the appendix to this section.
  - 5.1 Traceability Matrix
    - Added in options for the pass/fail column to be updated later.
  - 6. Environment
    - Updated listed environment components
  - 7. Assumptions
    - Created section of assumptions based on the test plan only instead of the project as a whole.

Regarding our test plan, we make the following assumptions:

- A test case passes if the actual output aligns with the expected output.
- DRs are satisfied after running the system more than 15 times.
- The overall execution of the system is successful if the user can take a photo, the photo is processed and returns with facial coordinates, the system displays five nose clusters, the user can selected their desired nose, and the system returns calculations for the required changes the doctor needs to make on the users' face to provide a quality rhinoplasty.