Facial Analyzer for Rhinoplasty (Option 9)

Priscilla Carbo, Marvin Cazeau, Jared Curtis, Alessandra Oo, Maree Sky

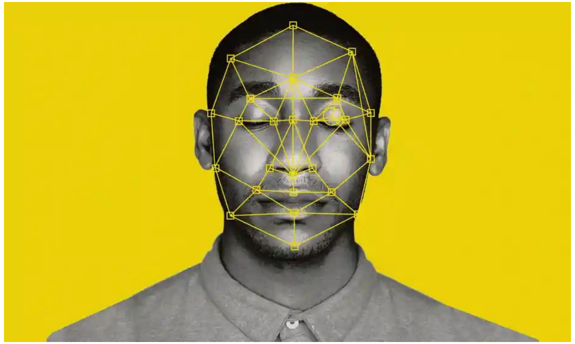
Introduction

- Analyze a patient who is about to get Rhinoplasty using ML (Machine Learning).

- Main goals:
  - Capture 3D image of a patient's face
  - Analyze the 3D image and mark significant landmarks on face
  - Compare images and landmark ratios using ML
  - Find new ML algorithm to implement

- Suggest potential changes in facial features.

- Allows patients to choose the closet match.



Benefits/Importance

- Provides patients with the opportunity to see the changes before the surgery.
- Assists Plastic surgeons by making surgery easier and more accurate
- Presents patients with multiple applicable options to choose from.
- Enhances the overall practice of Plastic surgery, benefiting hospitals that provide that service
- Potential applications for other reconstructive surgery.

Dependencies

Dependencies would include

- The Blender software
- Database we use to store face data
- Previous group's work
- IPhone X or later
- Bellus 3D Photo Application
- Digitized Rhinoplasty Web App

Assumptions

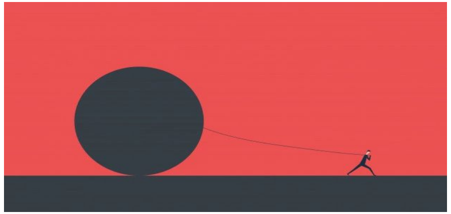
Previously in our 1st sprint we assumed that the client would

- Submit a face with no altercations
- Clients would know how to submit 3D image

2nd sprint Assumptions made were

- ML AI would automatically output recommended changes into a dataset
- All Dataset changes would be stored & saved into an SQL server

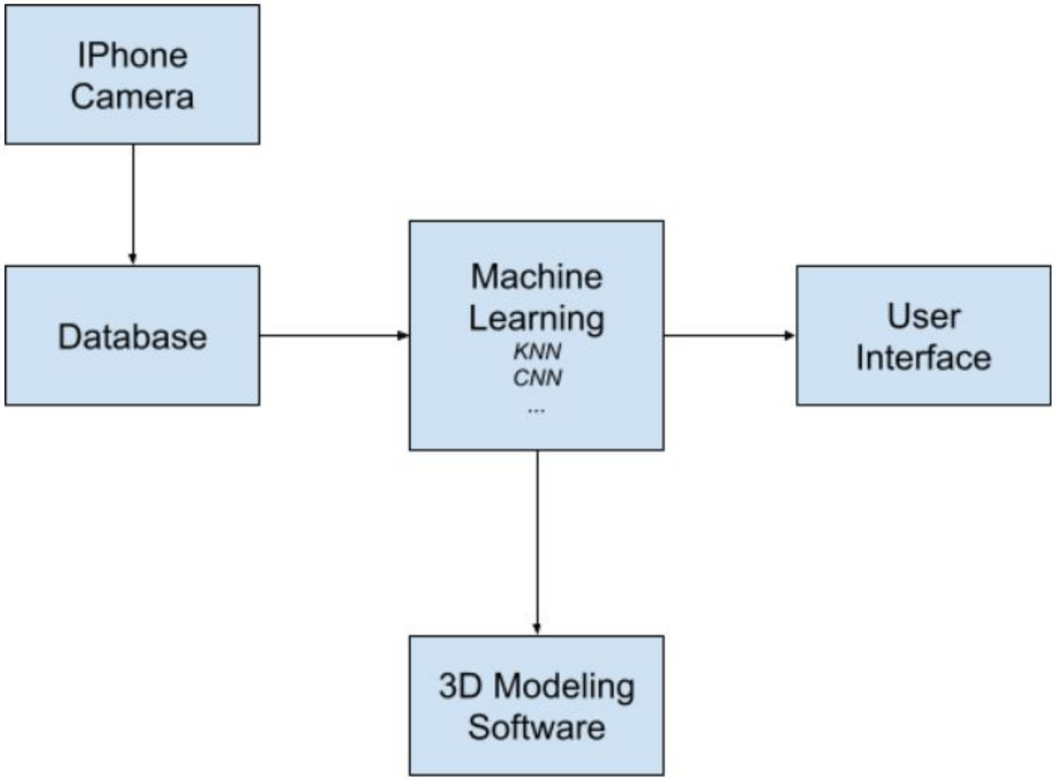
General Constraints



Constraints we faced during this sprint:

- Editing hardpoints to achieve desired results and having that information saved
- Sourcing and implementing a CNN ML algorithm compared to a KNN previously used
- Ensuring that uploading a 3D render is easy and accessible for everyone
- Sourcing more faces for our dataset

Software Class Diagram



Interfaces / Exports

- CNN Algorithm : Design

kNeighbors - DataFrame

Index	patient	distance
15	PatientE	4.19372e-10
12	PatientA	0.1023
24	PatientK	0.114711

face_group_1	10/28/2021 11:40 AM	File folder
face_group_2	10/28/2021 11:40 AM	File folder
face_group_3	10/28/2021 11:41 AM	File folder
face_group_4	10/28/2021 11:41 AM	File folder
face_group_6	10/28/2021 11:41 AM	File folder
face_group_7	10/28/2021 11:41 AM	File folder
face_group_8	10/28/2021 11:41 AM	File folder
face_group_9	10/28/2021 11:41 AM	File folder
face_group_10	10/28/2021 11:41 AM	File folder
face_group_11	10/28/2021 11:41 AM	File folder
face_group_12	10/28/2021 11:41 AM	File folder
face_group_13	10/28/2021 11:42 AM	File folder
face_group_14	10/28/2021 11:42 AM	File folder
face_group_15	10/28/2021 11:42 AM	File folder
face_group_16	10/28/2021 11:42 AM	File folder

Interfaces / Exports

- CNN Algorithm : Output

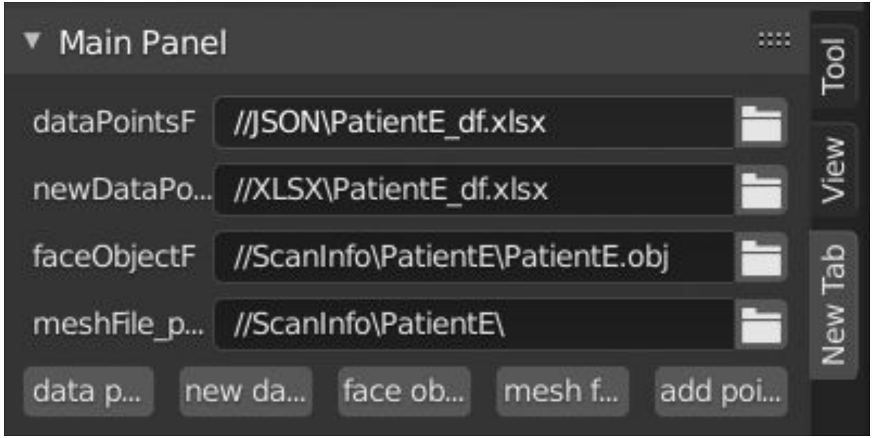
Interfaces / Exports

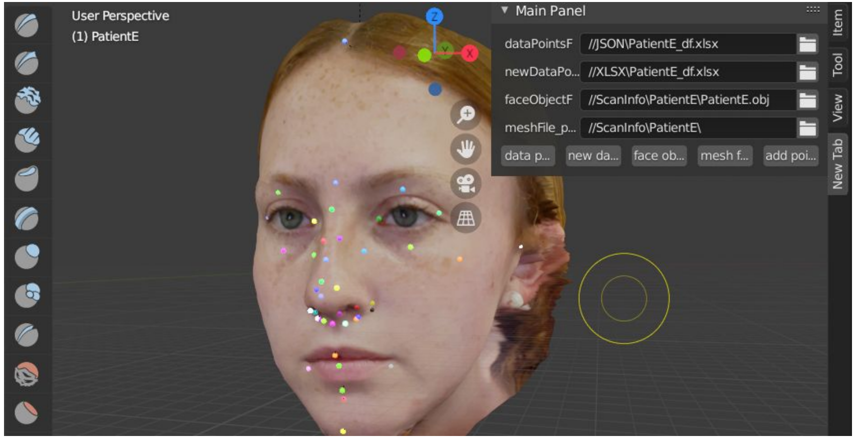
```
ImagePath='PATH TO IMAGE OF FACE'
test_image=image.load_img(ImagePath,target_size=(64, 64))
test_image=image.img_to_array(test_image)

test_image=np.expand_dims(test_image,axis=0)

result=classifier.predict(test_image,verbose=0)
#print(training_set.class_indices)

print('####'*10)
print('Prediction is: ',ResultMap[np.argmax(result)])
```

```
The Number of output neurons:  16
LENGTH OF TRAINING SET :  8
Epoch 1/10
8/8 [==============================] - 3s 329ms/step - loss: 101.3777 - accuracy:
0.0820 - val_loss: 4.4661 - val_accuracy: 0.1230
Epoch 2/10
8/8 [==============================] - 2s 307ms/step - loss: 3.1708 - accuracy: 0.1230
- val_loss: 2.7321 - val_accuracy: 0.0943
Epoch 3/10
8/8 [==============================] - 2s 277ms/step - loss: 2.7752 - accuracy: 0.1352
- val_loss: 2.4770 - val_accuracy: 0.2623
Epoch 4/10
==========================] - 2s 277ms/step - loss: 2.5125 - accuracy: 0.2213
ss: 2.2680 - val_accuracy: 0.2910
10
==========================] - 2s 292ms/step - loss: 2.3069 - accuracy: 0.2541
ss: 1.9911 - val_accuracy: 0.4016
10
==========================] - 2s 282ms/step - loss: 1.9669 - accuracy: 0.4303
ss: 1.6360 - val_accuracy: 0.5164
10
==========================] - 2s 294ms/step - loss: 1.6617 - accuracy: 0.5574
ss: 1.3091 - val_accuracy: 0.6107
10
==========================] - 2s 298ms/step - loss: 1.2887 - accuracy: 0.5984
ss: 1.0115 - val_accuracy: 0.7049
Epoch 9/10
8/8 [==============================] - 3s 334ms/step - loss: 1.2632 - accuracy: 0.6230
- val_loss: 1.2715 - val_accuracy: 0.6270
Epoch 10/10
8/8 [==============================] - 2s 326ms/step - loss: 1.0526 - accuracy: 0.6803
- val_loss: 0.3311 - val_accuracy: 0.9098
###### Total Time Taken:  0 Minutes ######
########################################
Prediction is:  face4
```

Interfaces / Exports

- Rewritten Blender Software





Lesson learned

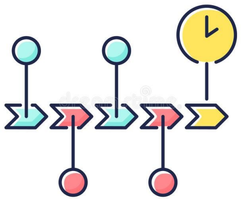
The main thing we learned throughout this sprint was to adapt to changes in the timeline. While we had initially stated that we wanted to get certain tasks done this sprint, we found that we had to spend more time on others, which forced us to adapt and resolve the issues that we faced.

That being said, we also learned how to better allocate time for each of the tasks we wanted to complete in case we did need to spend more time on it.

We'll acknowledge this throughout the last sprint to finish strong on our project and set up our future plans for the spring semester.

Project Timeline

- Basic Project Timeline
  - Sprint 3:
    - Implement Machine Learning models
    - Populate facial image database for ML models
    - Update and change SDD document for CNN models instead of KNN
    - Test and comparing Facial analyzer ML algorithms for accuracy and efficiency of outputs



Questions?

Thank you