캡스톤디자인 5조

안면 비대칭 진단 앱

요구사항 분석서 및

\* UML 및 Sequence Diagram \*

컴퓨터공학과 15011038 채수현

컴퓨터공학과 16012686 David Wahid

컴퓨터공학과 17011471 전찬의

목차

1. 문제설명서
2. 유스케이스 다이어그램
3. 요구사항 분석
   1. 요구사항 목록
   2. 요구사항 명세
      1. R001 (Input Image)
      2. R002 (이미지 전처리)
      3. R003 (Landmark 추출)
      4. R004 (Face Mid-Line)
      5. R005 (Image Gradient)
      6. R006 (Feature 추출)
      7. R007 (Machine Learning)
      8. R008 (비대칭 결과 반환)
      9. R009 (그래프 정보)
      10. R010 (비대칭 점수 정보)
4. 클래스 설계 명세서
   1. 클래스 다이어그램

4.2 클래스 다이어그램 정의

1. 시퀀스 명세서

5.1 시퀀스 다이어그램

5.2 시퀀스 다이어그램 정의

**1. 안면 비대칭 진단 모듈 문제설명서**

**(Facial Asymmetry Diagnosis Module Problem Statement)**

**안면 비대칭 진단 모듈은** 얼굴 사진을 App으로부터 전달받아 이미지 전처리를 통해 머신 러닝에 사용될 Feature들을 추출하고 비대칭정도가 담긴 정보를 반환해주는 모듈이다. 입력 받는 얼굴 사진은 jpg형식으로, App에서 정면 얼굴 사진이라고 판단된 사진을 입력 받는다. 반환되는 값들은 Json파일로 전달되며 사용자에게 출력할 그래프에 대한 정보와 비대칭 점수에 대한 정보를 포함한다.

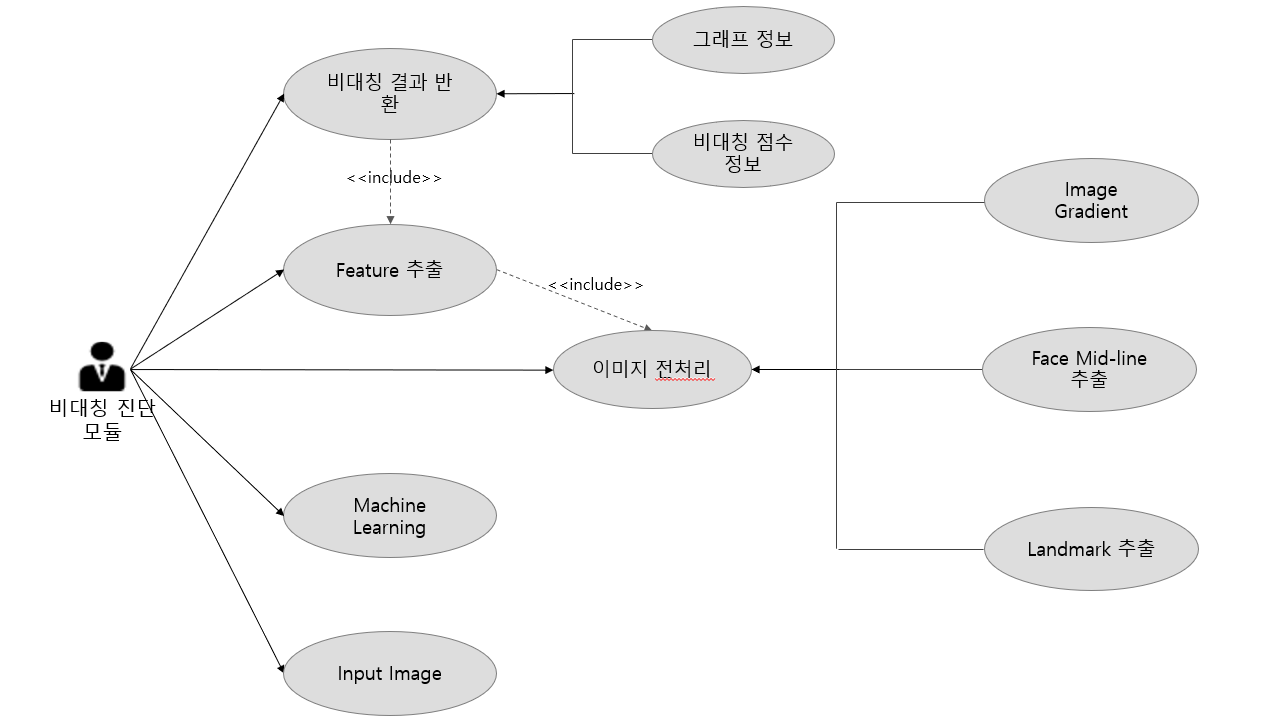
**모듈의 이미지 전처리는** 크게 3가지를 수행한다. 먼저 얼굴의 **표식점(Landmark)[[1]](#endnote-1))**을 추출하여 그 좌표를 저장한다. 다음으로 표식점의 일부분을 이용하여 얼굴의 **중간선(Face Mid-Line)[[2]](#endnote-2)**을 출력하고 중간선을 이용해 128x128의 이미지를 저장한 다. 마지막으로 Sobel필터를 사용하기 위해 Gray Scale이미지로 나타내고 이미지에 대한 Gradient (X, Y) 정보들을 저장한다.

**Machine Learning의 Feature들은** 크게 7가지로 분류한다. Sobel 이미지를 이용한 **DD(density difference), EOS(edge orientation similarity)[[3]](#endnote-3)**[2]값, 표식점들을 이용한 **CFR(CranioFacialRatio),** Length, Area, Angle값들과 그에 대한 여러 통계수치 그리고 얼굴 부위별 면적차이를 이용한다.

**비대칭 결과 반환은** 그래프에 출력할 정보와 비대칭인지 아닌지 판단할 수 있는 비대칭 점수에 대한 정보를 포함한다. 그래프정보는 얼굴의 부위별 면적차이가 포함된 정보이며 비대칭 점수는 DD와 EOS 그리고 비대칭 진단 결과가 포함된 정보이다.

.

**2. 유스케이스 다이어그램**



**3. 요구사항 분석**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 요구사항 분석서 | | 작성일자 | 2020.10.29 | 버전 | 2.0.0 |
| 사업명 | 비대칭 진단 모듈 | 작업단계 | 요구사항 분석 | 작업명 | 요구사항 분석 |

**3.1 요구사항 목록**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 모듈 | | | | |
| 번호 | 분류 | 요구사항명 | 상세내용 | 난이도 |
| R001 | Manage | Input Image | App으로부터 정면사진을 입력 받는다. | 하 |
| R002 | Manage | 이미지 전처리 | Machine Learning에 필요한 Feature들을 추출하기 위해 이미지 전처리과정을 수행한다. | 중 |
| R003 | Manage | Landmark 추출 | DLib을 이용하여 얼굴 이미지에 눈썹, 눈, 코, 입, 턱선을 포함한 68개의 점을 추출한다. | 중 |
| R004 | Manage | Face Mid-Line | Landmark의 c1,c2,c3의 점을 이용하여 얼굴의 중간선을 뽑아내고 중간선을 중심으로 128x128의 이미지를 추출한다. | 하 |
| R005 | Manage | Image Gradient | 추출한 Mid-Line이미지를 이용하여  Gray Scaling을 수행하고 Scaling된  이미지의Gradient(X,Y)를 계산한다. | 중 |
| R006 | Manage | Feature 추출 | Sobel필터를 이용한 DD, EOS값과  Landmark를 이용한 각도,넓이,길이에 대한 통계 값들을 Feature로 추출한다. | 상 |
| R007 | Manage | Machine Learning | Feature들과 저장된 머신러닝 Model을 이용하여 비대칭인지 아닌지 Test한 결과Label을 구한다. | 상 |
| R008 | Manage | 비대칭 결과 반환 | 사용자에게 제공될 사진에 대한 비대칭 결과들을 Json파일로 반환한다. | 하 |
| R009 | Manage | 그래프 정보 | Mid-Line이미지를 이용하여 얻은 얼굴의 부위별 면적차이가 담긴 정보 | 하 |
| R010 | Manage | 비대칭 점수 정보 | DD와 EOS값을 이용한 비대칭 점수 정보와 Model Test의 결과로 얻은 비대칭 진단 결과가 담긴 정보 | 하 |

**3.2 요구사항 명세**

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | Image Input | 요구사항번호 | R001 |
| 요구사항구분 | Manage | 작성자 | 전찬의 |
| 요구사항설명 : App으로부터 정면사진을 입력 받는다. | | | |
| 해결방안 :  ① App 으로부터 정면으로 판단된 얼굴사진인 jpg파일을 서버로 전송 받는다.  ② Module에 저장한다. | | | |
| 위험요소 | App에서 찍힌 사진 또는 갤러리에서 불러온 사진의 head pose estimation이 미리 수행되어 서버로 전송 되어야 한다. | | |
| 설계시 고려사항 | 서버 관리 | | |
| 관련요구사항 | 정면 얼굴 사진 | | |
| 시나리오 | 1. App에서 진단 모듈이 저장된 Server로 이미지를 전송한다.  2. 진단 Module의 IO클래스의 객체에 이미지를 저장한다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 이미지 전처리 | 요구사항번호 | R002 |
| 요구사항구분 | Manage | 작성자 | 전찬의 |
| 요구사항설명 : Machine Learning에 필요한 Feature들을 추출하기 위해  이미지 전처리과정을 수행한다. | | | |
| 해결방안 :  ① Machine Learning에 필요한 Feature들을 추출하기 위해 Landmark를 추출한다.  ② 얼굴의 중간선을 추출한다.  ③ Gray Scale이미지 변환과 Gradient값을 구한다. | | | |
| 위험요소 |  | | |
| 설계시 고려사항 | 이미지 전처리의 순서를 지켜야 한다. | | |
| 관련요구사항 | Landmark추출, Face Mid-Line, Image Gradient | | |
| 시나리오 | 1. 저장된 이미지의 Landmark를 추출한다.  2. Landmark정보를 이용하여 얼굴의 중간선을 추출한다.  3. 중간선을 이용하여 중간선이 중심을 지나고 이미지의 선분에 대하여 평행한 128x128이미지를 저장한다. ( (ex) -> Mid-Line Image)  4. 중간선 이미지를 Gray Scaling하고 그 이미지에 대하여 Gradient 값을  구한다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | Landmark 추출 | 요구사항번호 | R003 |
| 요구사항구분 | Manage | 작성자 | 전찬의 |
| 요구사항설명 : DLib과 openCV를 이용하여 얼굴 이미지에 눈썹, 눈, 코, 입, 턱선을 포함한 6개의 점을 추출한다. | | | |
| 해결방안  ① Dlib과 OpenCV를 이용한 landmark를 추출하는 python 오픈소스를 이용한다. | | | |
| 위험요소 |  | | |
| 설계시 고려사항 |  | | |
| 관련요구사항 | Input Image | | |
| 시나리오 | 1. 모듈의 IO클래스에 저장된 이미지와 Landmark.py코드를 이용하여 얼굴의 68개 표식점을 추출한다.  2. IO클래스에 표식점의 좌표정보를 저장한다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | Face Mid-Line | 요구사항번호 | R004 |
| 요구사항구분 | Manage | 작성자 | 전찬의 |
| 요구사항설명 : Landmark의 c1,c2,c3의 점을 이용하여 얼굴의 중간선을 뽑아내고 중간선을 중심으로 128x128의 이미지를 추출한다. | | | |
| 해결방안 :  ① **Diversity in Faces[[4]](#endnote-4)** 논문에서 사용된 face mid-Line과 같이 중간선을 추출한다.  ② 중간선을 중심으로 얼굴이 담긴 128x128이미지를 추출한다. | | | |
| 위험요소 | 신뢰성 있는 Head pose estimation이 수행되어야 한다.  App으로 전달받은 사진은 모두 같은 픽셀로 scaling 되어 있어야 한다. | | |
| 설계시 고려사항 | 128x128 이미지에 얼굴이 충분히 들어가는지 확인해야 한다. | | |
| 관련요구사항 | Input Image | | |
| 시나리오 | 1. Landmark의 c1과 c2의 중점 a와 c3를 이은 직선으로 중간선을 구한다.  2. 중간선에 평행하도록 이미지를 회전변환 하고 표식점을 업데이트 한다.  3. 중간선을 중심으로 128x128픽셀에 대한 정보를 IO클래스에 저장한다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | Image Gradient | 요구사항번호 | R005 |
| 요구사항구분 | Manage | 작성자 | 전찬의 |
| 요구사항설명 : 추출한 Mid-Line이미지를 이용하여 Gray Scaling을 수행하고 Scaling된 이미지의Gradient(X,Y)를 계산한다. | | | |
| 해결방안 :  ① 저장된 Mid-Line이미지를 Gray Scaling 한다.  ② 이미지의 Gradient X, Gradient Y 정보를 저장한다. | | | |
| 위험요소 |  | | |
| 설계시 고려사항 |  | | |
| 관련요구사항 | Face Mid-Line Image | | |
| 시나리오 | 1. IO클래스의 객체에 저장된 Mid-Line이미지를 Gray Scaling한다.  2. Gray Scaling된 이미지의 Gradient X, Gradient Y를 계산한 결과를 해당 객체에 저장한다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | Feature 추출 | 요구사항번호 | R006 |
| 요구사항구분 | Manage | 작성자 | 전찬의 |
| 요구사항설명 : Sobel필터를 이용한 DD, EOS값과 Landmark를 이용한 각도,넓이,길이에 대한  통계 값들을 Feature로 추출한다. | | | |
| 해결방안 :  ① Feature를 크게 DD, EOS, Length, Area, Angle, CFR로 나눈다.  ② 각각의 Feature에 대하여 결과값을 계산하여 저장한다. | | | |
| 위험요소 |  | | |
| 설계시 고려사항 | 다형성 | | |
| 관련요구사항 | 이미지 전처리 | | |
| 시나리오 | 1. 6개 Feature들의 객체를 생성한다.  2. DD, EOS클래스에 이미지의Gradient(X,Y)값을 생성자를 통해 입력 받는다.  3. DD클래스에서는 Sobel filter를 이용한 magnitude 이미지, EOS클래스에서는 theta이미지를 클래스 객체에 저장한다.  4. Length, Area, Angle클래스와 CFR클래스에 Landmark정보를 생성자를 통해 입력 받는다.  5. Length, Area, Angle클래스에서는 왼쪽 얼굴 오른쪽 얼굴에 대한 Landmark를 나누어서 클래스 객체에 저장한다.  6. 오버라이딩된 메소드(calResult())를 통해 각 Feature들 각각의 결과값 계산을 수행하며 그 과정은 아래와 같다.  6-1. DD, EOS클래스에서는 Sobel Filter를 적용한 이미지를 이용하여 중간선 기준으로 왼쪽 오른쪽에 대응되는 mag 또는 theta 값의 차이의 평균을 저장한다.  6-2. CFR클래스 에서는 **Diversity in Faces**논문에서 활용된 CranioFacialRatio와 같은 정보들을 계산하여 저장한다.  6-3. Length, Area, Angle 클래스에서는 왼쪽 표식점과 오른쪽 표식점에서 각각 2개, 4개, 3개씩 대칭되는 점들을 뽑아 길이, 면적, 각도를 계산하고 그 차이를 저장하며 차이값들에 대한 통계수치를 계산하여 저장한다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | Machine Learning | 요구사항번호 | R007 |
| 요구사항구분 | Manage | 작성자 | 전찬의 |
| 요구사항설명 : Feature들과 저장된 머신 러닝 Model을 이용하여 비대칭인지 아닌지 Test한  결과Label을 구한다 | | | |
| 해결방안 :  ① 안면 비대칭에 대한 T/F Label이 붙은 이미지와 이미지에서 추출한 Feature들을 이용하여 Machine Learning Model을 훈련시킨다.  ② 훈련된 Model을 서버에 저장하고 App사용자의 얼굴사진에서 얻어진 Feature들을 활용해 Test한 결과를 저장한다. | | | |
| 위험요소 | Model을 Training시키기 위하여 의학적으로 안면 비대칭을 진단받은 얼굴 사진이 여러장 필요함 | | |
| 설계시 고려사항 | Test 속도 | | |
| 관련요구사항 | 사용자의 얼굴 사진, Feature추출, Machine Learning Model | | |
| 시나리오 | 1. 추출한 Feature들과 서버에 저장된 Model을 이용하여 비대칭인지 아닌지 Test한 결과를 Machine Learning 객체에 저장한다. | | |

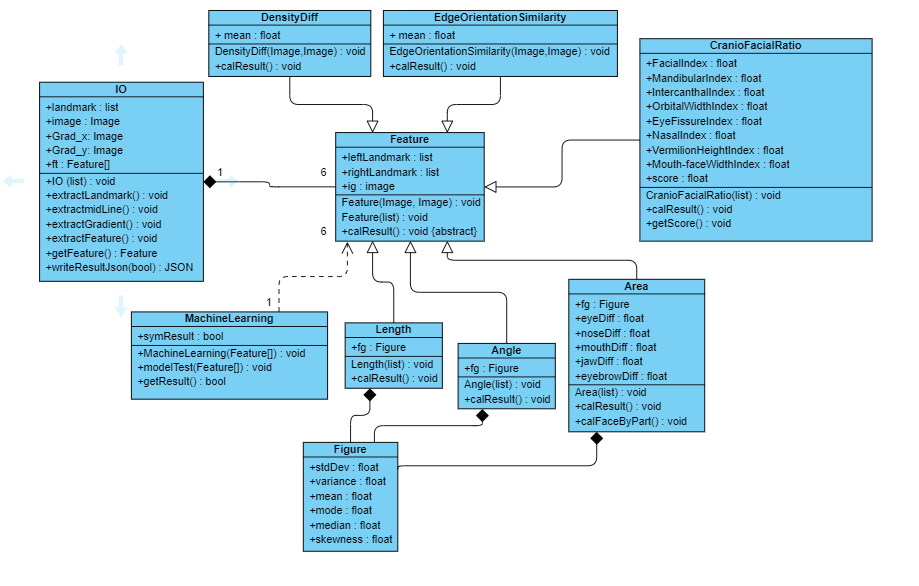
|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 비대칭 결과 반환 | 요구사항번호 | R008 |
| 요구사항구분 | Manage | 작성자 | 전찬의 |
| 요구사항설명 : 사용자에게 제공될 사진에 대한 비대칭 결과들을 Json파일로 반환한다.  안면 비대칭 진단 결과를 얻을 수 있다. | | | |
| 해결방안 :  ① 사용자에게 보여줄 정보를 Json으로 반환한다. | | | |
| 위험요소 |  | | |
| 설계시 고려사항 |  | | |
| 관련요구사항 | 그래프 정보, 비대칭 점수 정보 | | |
| 시나리오 | 1. 그래프에 출력할 정보를 Json파일에 추가한다.  2. 비대칭 점수에 관한 정보와 비대칭 진단 결과에 대한 정보를 Json파일에 추가한다.  3. Json 파일을 반환한다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 그래프 정보 | 요구사항번호 | R009 |
| 요구사항구분 | Manage | 작성자 | 전찬의 |
| 요구사항설명 : Mid-Line이미지를 이용하여 얻은 얼굴의 부위별 면적차이가 담긴 정보 | | | |
| 해결방안 :  ① Area클래스에서 부위별 면적차이를 계산하여 저장한다. | | | |
| 위험요소 |  | | |
| 설계시 고려사항 |  | | |
| 관련요구사항 | 중간선에 의해 업데이트 된 Landmark | | |
| 시나리오 | 1. Area클래스에서 FaceByPart()를 수행하여 중간선을 기준으로 눈썹, 눈, 코, 입, 턱선의 면적의 차이를 Area 클래스의 객체에 저장한다. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 비대칭 점수 정보 | 요구사항번호 | R010 |
| 요구사항구분 | Manage | 작성자 | 전찬의 |
| 요구사항설명 : DD, EOS값과 CFR데이터를 이용한 비대칭 점수 정보와 Model Test의 결과로 얻은 비대칭 진단 결과가 담긴 정보 | | | |
| 해결방안 :  ① 계산된 DD, EOS값과 CFR에서 얼굴 비율에 대한 데이터를 이용하여 비대칭 점수를 저장한다  ② Model Test로 얻은 비대칭 진단 결과값을 저장한다. | | | |
| 위험요소 |  | | |
| 설계시 고려사항 | 점수를 정하기 위하여 DD, EOS와 CFR의 데이터에 대한 가중치 설정 | | |
| 관련요구사항 | DD, EOS, CFR클래스의 데이터값 | | |
| 시나리오 | 1. DD와 EOS의 평균값과 CFR의 얼굴비율에 대한 값이 담긴 변수 8가지를 이용하여 비대칭 점수를 계산하여 IO클래스의 객체에 저장한다.  2. Model Test로 얻은 비대칭 진단 결과는 Machine Learning클래스의 객체멤버에 저장된 값을 사용한다. | | |

5. 클래스 설계 명세서

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **D1** | **클래스 설계 명세서** | | | | | |
| 시스템명 | | 안면 분석 프레임워크 | 작성일지 | 2020. 10. 29 | 버전 | 2.0 |
| 단계명 | | 분석 |

5.1 클래스 다이어그램

5.2 클래스 다이어그램 정의

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 설계 ID | SS\_DCD\_001 | 설계 클래스명 | IO | |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| landmark | public | list | 없음 | landmark 저장 |
| image | public | Image | 없음 | 이미지 저장 |
| Grad\_x | public | Image | 없음 | sobel 필터의 결과에 대한 gradients |
| Grad\_y | public | Image | 없음 | sobel 필터의 결과에 대한 gradients |
| ft | public | Feature[] | 없음 | landmark를 통해 추출된 Feature.  Length, Angle, Area, DD, EOS, CranioFacialRatio 객체가 저장됨 |
|  | | | | |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| IO | public | 없음 | double | 이미지를 읽어 Image에 저장 |
| extractLandmark | public | 없음 | void | landmark추출 |
| extractmidLine | public | 없음 | void | Landmark를 이용하여 midlline추출, midline 기준으로 사진 회전, 사진을 잘라서 저장 |
| extractgradient | public | 없음 | void | extractmidLine에서 호출, 이미지를 GrayScale출력 후 sobel 필터의 Gradient x, Gradient y 구함 |
| extractFeature | public | 없음 | void | Length, Angle, Area, DD, EOS, CranioFacialRatio 객체를 생성하고 결과값을 계산을 수행 |
| getFeature | public | 없음 | Feature[] | Feature 반환 |
| writeResultJson | public | bool | JSON | 결과(부위별 면적 차, CranioFacialRatio score, 비대칭 진단 결과) Json 형태로 출력 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 설계 ID | SS\_DCD\_002 | 설계 클래스명 | Feature | |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| leftLandmark | public | list | 없음 | 왼쪽 Landmark |
| rightLandmark | public | list | 없음 | 오른쪽 Landmark |
| ig | public | Image | 없음 | 이미지 |
|  | | | | |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| Feature | public | Image, Image | void | sobel 필터를 위한 gradient\_x, gradient\_y 전달 후 sobel 이미지 저장 |
| Feature | public | list | void | landmark를 받아서 leftLandmark, rightLandmark 초기화 |
| calResult {abstract} | public | 없음 | void | 특징들을 계산하는 추상 메소드 (Length, Angle, Area, DD, EOS, CranioFacialRatio class에서 재정의됨) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 설계 ID | SS\_DCD\_003 | 설계 클래스명 | Figure | |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| stdDev | public | float | 0 | 표준편차 |
| variance | public | float | 0 | 분산 |
| mean | public | float | 0 | 평균 |
| mode | public | float | 0 | 최빈값 |
| median | public | float | 0 | 중앙값 |
| skewness | public | float | 0 | 왜도 |

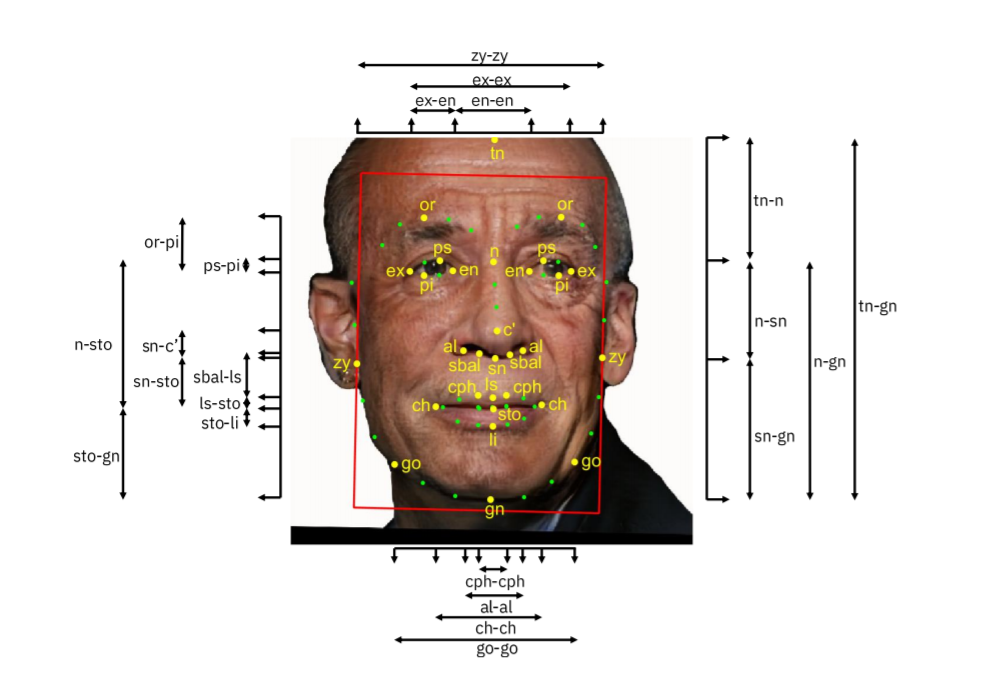
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 설계 ID | SS\_DCD\_004 | 설계 클래스명 | Length (Feature 상속) | |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| fg | public | Figure | 없음 | 통계 수치들을 저장하는 Figure 객체 |
|  | | | | |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| Length | public | list | void | landmark를 부모 클래스인 Feature 클래스에 leftLandmark와 rightLandmark로 나누어서 저장 |
| calResult | public | 없음 | void | leftLandmark와 rightLandmark에서 각 두 개를 선택하여 그 landmark를 지나는 직선의 길이를 비교하는 계산 수행 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 설계 ID | SS\_DCD\_005 | 설계 클래스명 | Angle (Feature 상속) | |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| fg | public | Figure | 없음 | 통계 수치들을 저장하는 Figure 객체 |
|  | | | | |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| Angle | public | list | void | landmark를 부모 클래스인 Feature 클래스에 leftLandmark와 rightLandmark로 나누어서 저장 |
| calResult | public | 없음 | void | leftLandmark와 rightLandmark에서 각 세 개를 선택하여 그 landmark를 지나는 직선의 각도를 비교하는 계산 수행 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 설계 ID | SS\_DCD\_006 | 설계 클래스명 | Area (Feature 상속) | |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| fg | public | Figure | 없음 | 통계 수치들을 저장하는 Figure 객체 |
| eyeDiff | public | float | 0 | 좌우 눈 면적 차이 |
| noseDiff | public | float | 0 | 좌우 코 면적 차이 |
| mouthDiff | public | float | 0 | 좌우 입 면적 차이 |
| jawDiff | public | float | 0 | 좌우 턱 면적 차이 |
| eyebrowDiff | public | float | 0 | 좌우 눈썹 면적 차이 |
|  | | | | |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| Area | public | list | void | landmark를 부모 클래스인 Feature 클래스에 leftLandmark와 rightLandmark로 나누어서 저장 |
| calResult | public | 없음 | void | leftLandmark와 rightLandmark에서 각 네 개를 선택하여 그 Landmark들이 이루는 면적의 차 계산 수행 |
| calFaceByPart | public | 없음 | void | 왼쪽과 오른쪽 얼굴 부위별 (눈, 눈썹, 코, 입, 턱) 면적 차 계산 수행 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 설계 ID | SS\_DCD\_007 | 설계 클래스명 | | DensityDiff | | |
| 속성 | | | | | | |
| 속성명 | 가시성 | 타입 | | 기본값 | | 설명 |
| mean | public | float | | 0 | | 좌우 얼굴의 sobel 이미지 간의 DensityDiff값들의 평균 |
|  | | | | | | |
| 오퍼레이션 | | | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | | 반환타입 | | 설명 |
| DensityDiff | public | Image, Image | | void | | 파라미터의 gradient\_x, gradient\_y를 통해 부모 클래스인 feature의 ig에 DensityDiff의 이미지 저장 |
| calResult | public | 없음 | | void | | 좌우 얼굴의 sobel 이미지 간의 DensityDiff 계산 수행 |
| 설계 ID | SS\_DCD\_008 | 설계 클래스명 | EdgeOrientationSimilarity | | | |
| 속성 | | | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | | 설명 | |
| mean | public | float | 0 | | 좌우 얼굴의 sobel 이미지 간의 EdgeOrientationSimilarity 값들의 평균 | |
|  | | | | | | |
| 오퍼레이션 | | | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | | 설명 | |
| EdgeOrientationSimilarity | public | Image, Image | void | | 파라미터의 gradient\_x, gradient\_y를 통해 부모 클래스인 feature의 ig에 EdgeOrientationSimilarity의 이미지 저장 | |
| calResult | public | 없음 | void | | 좌우 얼굴의 sobel 이미지 간의 EdgeOrientationSimilarity 계산 수행 | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 설계 ID | SS\_DCD\_009 | 설계 클래스명 | cranioFacialRatio | |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| FacialIndex | public | float | 0 | (n − gn)/(zy − zy) |
| MandibularIndex | public | float | 0 | (sto − gn)/(go − go) |
| IntercanthalIndex | public | float | 0 | (en − en)/(ex − ex) |
| OrbitalWidthIndex | public | float | 0 | (ex − en)/(en − en) |
| EyeFissureIndex | public | float | 0 | (ps − pi)/(ex − en) |
| NasalIndex | public | float | 0 | (al − al)/(n − sn) |
| VermilionHeightIndex | public | float | 0 | (ls − sto)/(sto − li) |
| Mouth-faceWidthIndex | public | float | 0 | (ch − ch)/(zy − zy) |
| score | public | float | 0 | cranioFacialRatio 점수 |



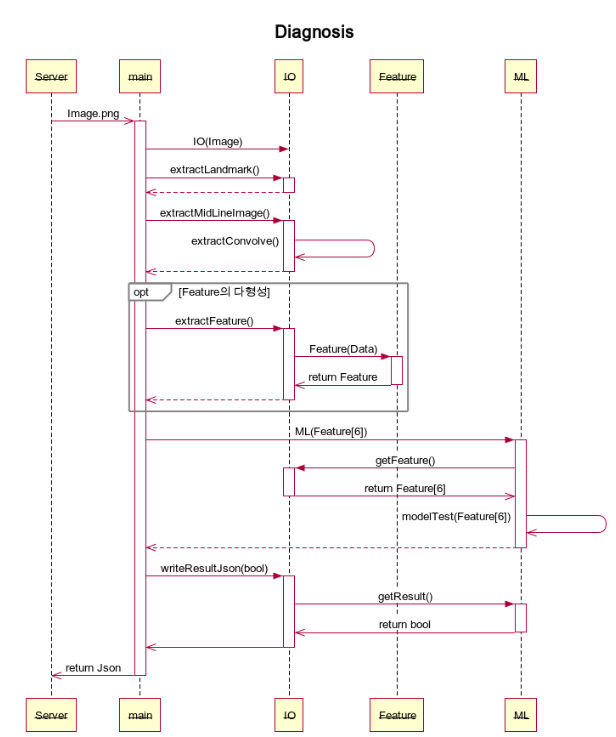
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| cranioFacialRatio | public | 없음 | void | landmark를 부모 클래스인 feature에 저장 |
| calResult | public | 없음 | void | 좌우 얼굴 간의 FacialIndex, MandibularIndex, IntercanthalIndex, OrbitalWidthIndex, EyeFissureIndex, NasalIndex, VermilionHeightIndex, Mouth-faceWidthIndex의 차 계산 수행 후 점수를 매겨서 score에 저장 |
| getScore | public | 없음 | void | score 출력 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 설계 ID | SS\_DCD\_010 | 설계 클래스명 | MachineLearning | |
| 속성 | | | | |
| 속성명 | 가시성 | 타입 | 기본값 | 설명 |
| symResult | public | bool | 없음 | 결과(부위별 면적 차, cranioFacialRatio score, 비대칭 진단 결과) |
|  | | | | |
| 오퍼레이션 | | | | |
| 오퍼레이션명 | 가시성 | 파라미터 | 반환타입 | 설명 |
| MachineLearning | public | Feature[] | void | Feture 객체에 들어있는 Length,Angle, Area,DD,EOS, CranioFacialRatio 들의 계산 값들을 machineLearing의 feature로 사용하여 model을 Test |
| modelTest | public | Feature[] | void | user의 사진을 통해 얻어진 사진의 feauture를 Traing된 모델로 결과 추출 후 symResult에 저장 |
| getResult | public | 없음 | bool | 비대칭 판단 결과 출력 |

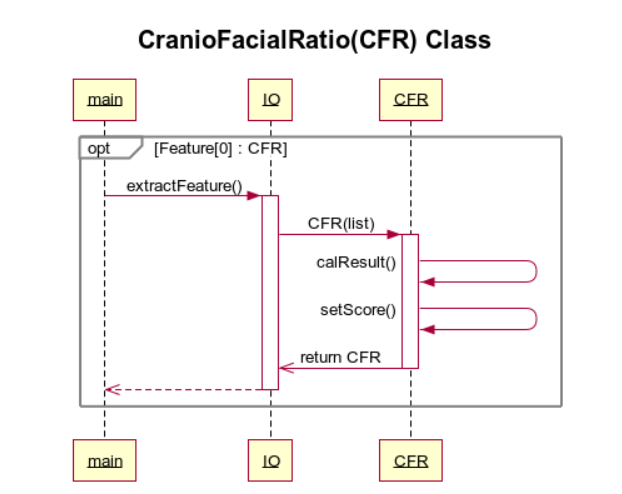
6. 시퀀스 명세서

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **D2** | **시퀀스 명세서** | | | | | |
| 시스템명 | | 안면 비대칭 진단 앱 | 작성일지 | 2020. 10. 29 | 버전 | 2.0 |
| 단계명 | | 분석 |

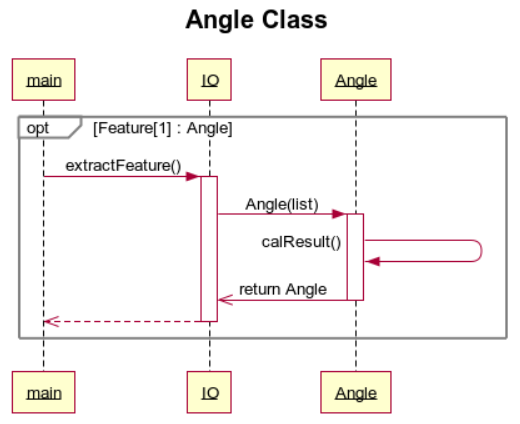
6.1 시퀀스 다이어그램

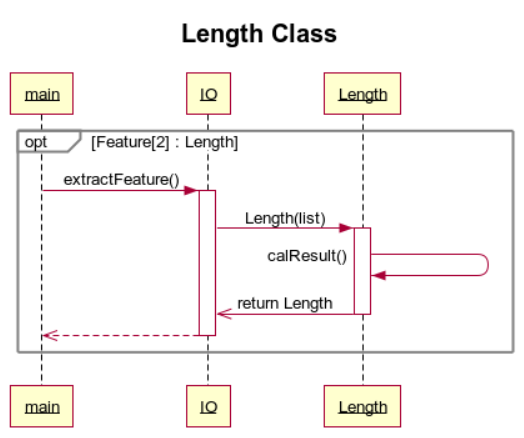
[AM\_001]

[AM\_002]

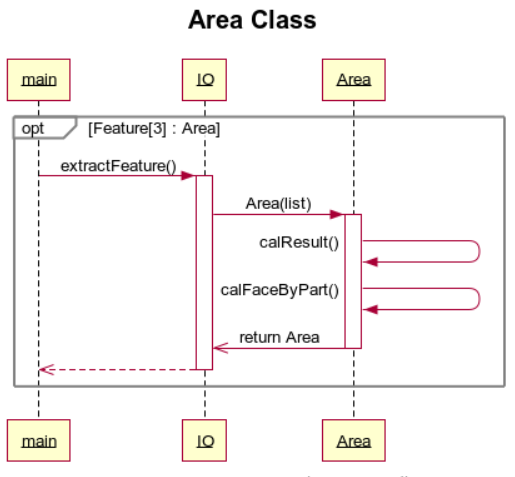


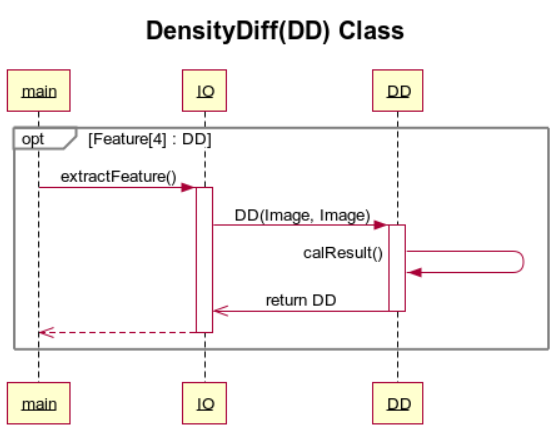
[AM\_003]



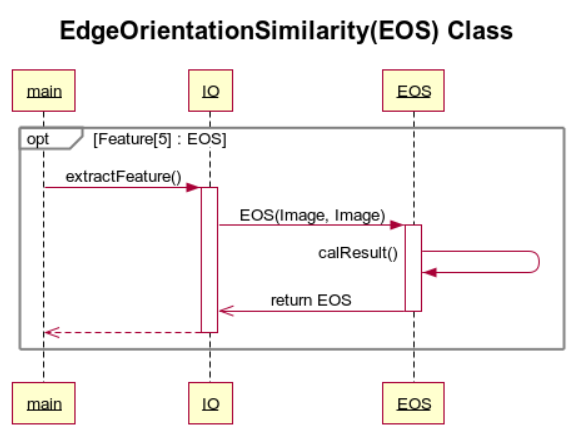
[AM\_004]

[AM\_005]



[AM\_006]

[AM\_007]



6.2 시퀀스 다이어그램 정의

|  |  |  |
| --- | --- | --- |
| 시퀀스 ID | 시퀀스 명 | 시퀀스 설명 |
| AM\_001 | Diagnosis | 정면사진을 입력받아 머신러닝을 이용하여 사용자에게 전달할 그래프 정보와 비대칭 진단 점수를 Json으로 반환하는 과정 |
| AM\_002 | opt1 : CFR | IO 클래스에서 다형성을 이용한 Feature객체 생성과정 중 CFR객체를 반환하는 과정 |
| AM\_003 | opt2 : Angle | IO 클래스에서 다형성을 이용한 Feature객체 생성과정 중 Angle객체를 반환하는 과정 |
| AM\_004 | opt3 : Length | IO 클래스에서 다형성을 이용한 Feature객체 생성과정 중 Length객체를 반환하는 과정 |
| AM\_005 | opt4 : Area | IO 클래스에서 다형성을 이용한 Feature객체 생성과정 중 Area객체를 반환하는 과정 |
| AM\_006 | opt4 : DD | IO 클래스에서 다형성을 이용한 Feature객체 생성과정 중 DD객체를 반환하는 과정 |
| AM\_007 | opt4 : EOS | IO 클래스에서 다형성을 이용한 Feature객체 생성과정 중 EOS객체를 반환하는 과정 |

1. DLib과 openCV 이용하여 얼굴의 Landmark 추출 [↑](#endnote-ref-1)
2. [1] Y. Liu, K. L. Schmidt, J. F. Cohn, and R. L. Weaver. Facial asymmetry quanti\_cation for

   expression invariant human identi\_cation. In IEEE Face Gesture Recognition, 2002 [↑](#endnote-ref-2)
3. [2] Michele Merler, Nalini Ratha, Rogerio Feris, John R. Smith. **Diversity in Faces**. April 10, 2019 [↑](#endnote-ref-3)
4. [2]의 논문내용 [↑](#endnote-ref-4)