

<창의종합설계1 중간보고서>

# OpenCV를 이용한 Gender Classification

Team Q 박충훈, 박상문, 방지우

엄현상 교수님

황창호 담당자님 (티맥스소프트)

# Contents

<b>1. Abstract.....</b>	<b>3</b>
<b>2. Introduction.....</b>	<b>3</b>
<b>3. Background study .....</b>	<b>4</b>
3-1) 관련 접근방법 / 기술 분석 .....	4
3-2) 프로젝트 개발 환경 .....	8
<b>4. Goal/Problem &amp; Requirement.....</b>	<b>8</b>
<b>5. Approach.....</b>	<b>8</b>
<b>6. Architecture .....</b>	<b>9</b>
6-1) Architecture Diagram .....	9
6-2) Architecture Description .....	9
<b>7. Implementation Spec .....</b>	<b>10</b>
7-1) Input/Output Interface .....	10
7-2) Inter-Module Communication Interface.....	10
7-3) Modules .....	10
<b>8. Current Status .....</b>	<b>11</b>
<b>9. Future Work .....</b>	<b>11</b>
<b>10. Division &amp; Assignment of Work.....</b>	<b>12</b>
<b>11. Schedule.....</b>	<b>12</b>
<b>◆ Appendix .....</b>	<b>12</b>
a. Preprocessing.....	12
b. PCA module.....	13
c. Landmark module.....	14
d. Test module .....	14

## 1. Abstract

이 프로젝트에서는 OpenCV를 이용하여 주어진 얼굴이미지 데이터베이스를 통해 성별을 구분할 수 있는 주된 attribute가 무엇인지 찾아내고, 이를 통해 새로운 얼굴이미지 input이 주어졌을 때 해당 이미지의 성별을 추정할 수 있는 Gender Classification 프로그램을 작성한다. 우리의 목표는 얼굴이미지 데이터베이스에서 주된 attribute를 적절한 과정을 거쳐 추출하여 성별을 추정하는 모델의 정확도를 최대화시키는 것이다. 또한 이를 바탕으로 성별을 추정하기 위한 모델을 생성하는 과정에서 걸리는 시간을 최소화하면서 신뢰도 높은 모델을 생성할 수 있는 코드를 구현하려 한다. 마지막으로, 사용자가 생성한 모델에 얼굴이미지 input을 주어 결과를 쉽게 확인할 수 있도록 웹 프로그래밍을 통해 UI를 구현할 것이다.

## 2. Introduction

이미지 처리 기술은 중앙처리장치, 대용량 저장 장치의 발달과 함께 정확도와 속도 등 성능에서 많은 발전을 이루었다. 단순히 이미지의 사이즈나 색조, 명암 등을 변경하는 것을 넘어서 이미지 자체가 가지고 있는 정보를 추출하고 이를 통해 의미 있는 결과를 낼 수 있는 다양한 기술들이 등장했다.

Gender classification은 이러한 이미지 처리 기술과 머신러닝 기술이 결합하면서 등장한 분야이다. Gender classification에서는 인간이 누군가의 Gender를 판별할 때와 유사한 방법을 사용한다. 인간은 대상의 Gender를 판별할 때 대상의 얼굴에서 다양한 정보를 취합한다. 눈, 코, 입 등의 신체 기관이 어떻게 생겼는지 에서부터 이들이 얼굴 상에 어떻게 배치되어있는지, 얼굴의 윤곽은 어떠한 지 등 다양한 정보를 얻는다. 이를 종합하여 대상의 Gender를 판별한다. 또한 이러한 경험이 쌓여서 Gender를 판별하는 기준이 굳어지고, 이를 바탕으로 더 빠르게 Gender를 판별하게 된다.

Gender classification에서도 인간과 마찬가지로 얼굴이미지로부터 주된 attribute를 추출한다. 적절한 attribute를 추출하기 위해 주성분 분석(PCA), Geometric-Based Feature Extraction(GBFE) 등의 방법론을 사용한다. 이렇게 추출한 attribute를 바탕으로 얼굴이미지 input의 Gender를 판별하기 위한 모델을 세운다. 여기서는 K-Nearest Neighbor(KNN), Support Vector Machine(SVM) 등의 알고리즘을 사용하여 모델링을 할 것이다. 이제 이렇게 세워진 모델을 바탕으로 얼굴 이미지 input의 Gender를 판별한다.

우리는 이미지 데이터가 Gender 정보로 변환되는 과정을 직접 구현해보고, 더 효율적이고 정확한 Gender classification 프로그램을 개발할 것이다. 이를 위해 PCA와 GBFE 두 가지 방법으로 attribute를 추출하고, 이 정보들 각각에 대하여 classification 알고리즘을 통해 모델을 생성할 것이다. 그리고 이 둘을 종합하여 더 높은 성능의 Gender classification을 구현할 것이다.

### 3. Background study

#### 3-1) 관련 접근방법 / 기술 분석

##### • OpenCV(Open Source Computer Vision Library)

OpenCV는 Intel에서 개발한 Computer Vision 오픈 소스 라이브러리로, 여기서 Computer Vision이란 컴퓨터가 알고리즘을 사용하여 사람처럼 물체 또는 글씨를 인식하기 위한 CS 분야 기술을 말한다. OpenCV는 실시간 연산이 가능하도록 연산의 효율성을 중점으로 설계되었으며 멀티 코어 프로세서를 활용할 수 있다. 보통 머신 러닝과 함께 사용되는 경우가 많다.



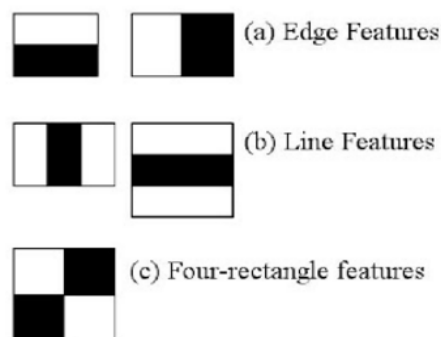
OpenCV는 다음의 5개의 구성 요소를 포함하고 있다.

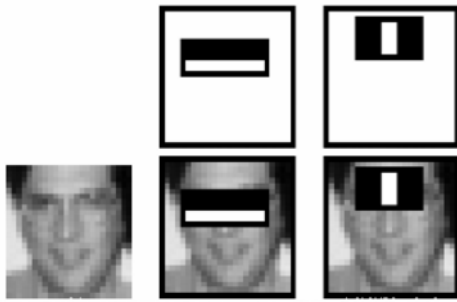
1. cv : 기본적인 영상처리와 고수준 컴퓨터 비전 알고리즘
2. mll : 머신 러닝 알고리즘
3. highGUI : 입출력과 관련된 함수들 및 영상과 비디오를 불러오고 저장하는 기능
4. CXCORE : 기본 자료구조와 알고리즘
5. CVAux : 얼굴 인식과 같은 실험적인 알고리즘

이와 같은 알고리즘들은 사람의 얼굴을 인식하고 감지하며 움직임을 추적하는 것을 가능토록 한다. 또한 이미지들을 연결하여 전체 장면을 만들어내는 streetview도 OpenCV를 통해 만들어낼 수 있다. OpenCV는 C와 C++로 구현되어 있고, C, C++, Python, Java, Matlab 등의 언어를 지원하며 Windows, Linux, OS X, 안드로이드, iOS 플랫폼을 지원한다.

##### • Haar Feature-based Cascade Classifier

물체를 인식하기 위해 쓰이는 대표적인 기법 중 하나인 Haar feature-based cascade classifier는 다양한 크기와 형태의 흑백 영역에 대한 픽셀 값의 평균 차를 통해 특징을 판단, 파악하는 기법이다. 이는 연산과정이 간단하여 빠른 얼굴 검출에 적합하다. 오른쪽의 그림은 Haar-like feature의 몇 가지 예시들이다. 각각의 feature는 흑색 영역 안의 픽셀 값들의 합과 백색 영역 안의 픽셀 값들의 합의 차이를 통해 얻어진 하나의 값을 의미한다.

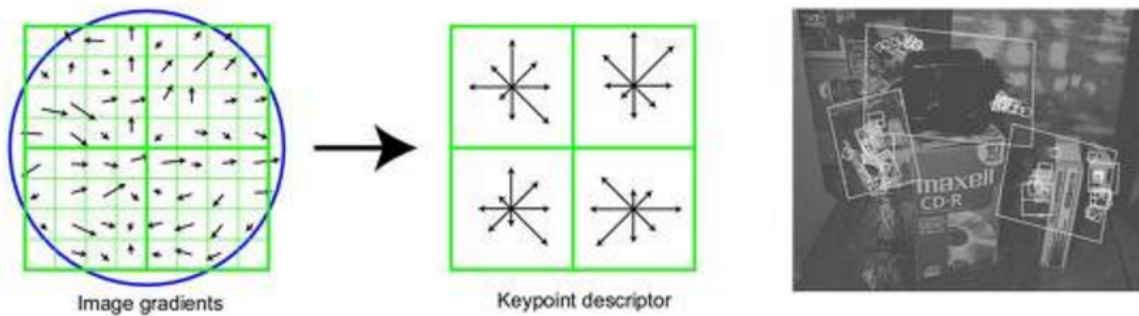




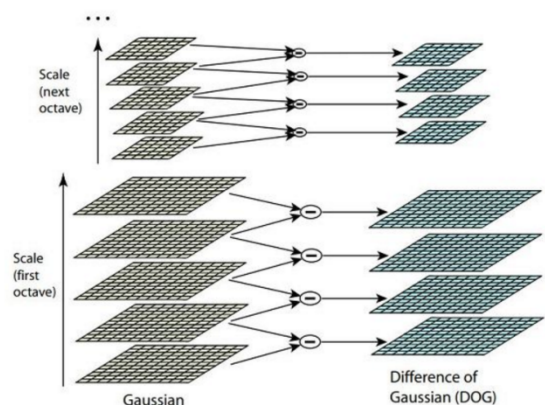
이와 같은 feature들을 다양한 위치와 크기로 수백, 수천 개를 조합한다면 사람의 얼굴에서 각 특징들을 추출해낼 수 있다. 예를 들어 사람의 눈을 떠올려보자. 사람의 얼굴에서 상대적으로 눈은 들어가 있는 형태를 가지기 때문에 비교적 어둡게 보인다. 마찬가지로 사람의 코는 얼굴에서 상대적으로 튀어나온 형태를 가지기 때문에 주변에 비해 비교적 밝은 특징을 갖는다. 위의 그림은 Haar-like feature을 이용해 사람의 얼굴에서 눈과 코를 인식하는 과정을 대략적으로 나타낸 것이다.

### • SIFT(Scale-Invariant Feature Transform)

앞서 설명한 Haar feature-based cascade classifier는 기본적으로 물체의 기하학적 정보를 유지하며 특정 영역 안에서의 밝기 차를 이용하는 기법이기 때문에 그 영역 안에서의 물체의 위치 변화는 어느 정도 커버할 수 있다. 하지만 물체가 회전된 경우에는 Haar를 이용한 특징 검출이 힘들다. SIFT는 이를 보완할 수 있는 크기와 회전에 불변하는 특징을 추출하는 영상 인식 기법이다. 영상에서 코너 또는 글자, 형상 있는 모양 등을 특징점이라 할 수 있는데, SIFT를 통해 영상의 크기와 회전에 불변하는 특징점을 추출할 수 있다. SIFT 특징 벡터는 특징점 주변의 영상을 4\*4 블록으로 나누고 각 블록에 속한 픽셀 값들의 방향과 크기에 대한 히스토그램을 구해 연결한 128차원 벡터를 말한다. 이를 통해 이미지의 윤곽 등을 인식할 수 있다.

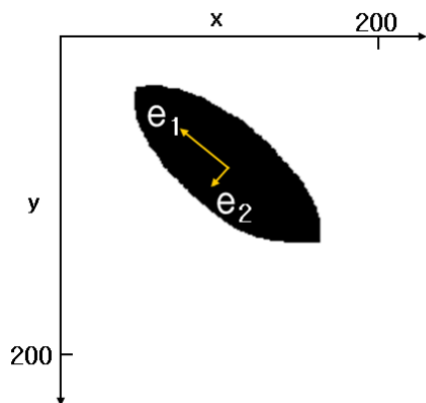


SIFT 특징점은 Laplacian 함수값을 사용하여 추출할 수 있다. 주로 속도 문제로 인해 Dog(Difference of Gaussian)을 이용해 각 스케일 별 Laplacian을 근사적으로 계산하는 방법을 사용한다. DoG는 입력 영상에 Gaussian 필터를 점진적으로 적용한 이미지들간의 subtraction 영상을 의미한다. 이는 이론적으로 Laplacian of Gaussian 필터를 적용한



LoG와 거의 동일한 결과를 보인다. 이렇게 얻어진 DoG 피라미드에서 극대 또는 극소점을 찾으면 SIFT의 특징점이 된다.

#### • PCA(Principal Component Analysis)



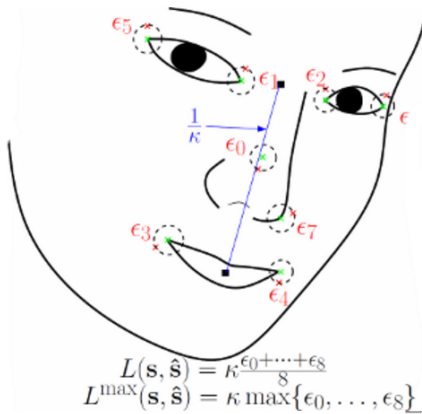
PCA는 분포되어 있는 데이터들의 principal component를 찾아주는 방법이다. 왼쪽의 그림에서 데이터의 특징을 가장 잘 나타낼 수 있는 벡터는  $e_1$ 과  $e_2$ 일 것이다.  $e_1$ 과  $e_2$ 의 방향과 크기를 통해 우리는 이 데이터 분포가 어떤 형태인지 가장 단순하고 효율적으로 파악할 수 있다. 즉,  $e_1$ 과  $e_2$ 는 principal component라 할 수 있다.

가장 대표적인 예로 얼굴인식(face recognition)에 PCA를 응용할 수 있다. 먼저 인식 대상이 되는 사람들의 얼굴 샘플들에 대해 PCA를 통해  $k$ 개의 주요 eigenface들을 구한 후 각 개인들을 eigenface로 근사했을 때의 근사계수를 저장한다. 여기서 eigenface는 얼굴 이미지를 통해 얻은 주성분 벡터들을 다시 이미지로 해석한 것을 말한다. 이후 입력 데이터  $x$ 가 들어왔을 때, 이를  $k$ 개의 eigenface로 근사한 근사계수가 미리 저장된 개인별 근사계수들 중 가까운 계수를 조사하여  $x$ 를 식별할 수 있다.

다음 그림은  $k$ 개의 eigenface만으로 원본 얼굴을 복원한 결과들이다.  $k$ 가 클수록 개인 고유의 얼굴 특성이 강화되는 것을 알 수 있다.



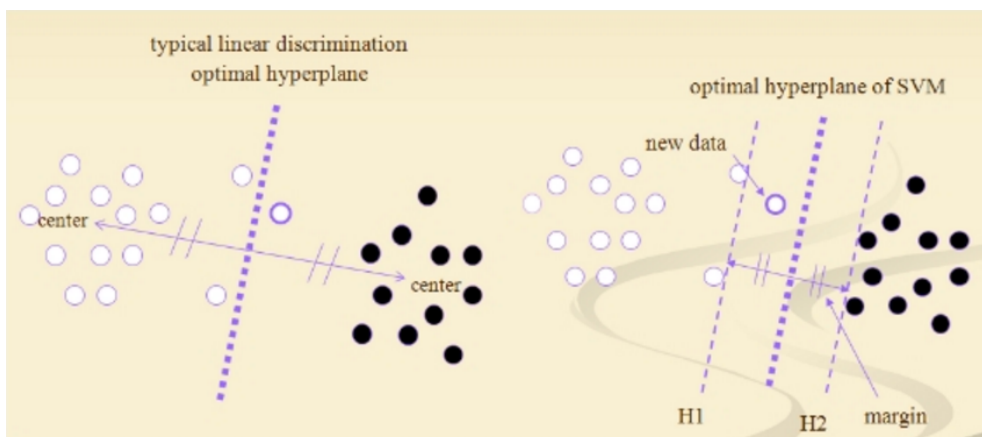
### • Geometric Based Feature Extraction



사람의 얼굴에는 눈과 눈 사이의 거리, 눈썹의 위치, 코 끝의 모양, 입매 등과 같은 여러 가지 특징들이 있다. 이러한 얼굴의 특징들을 추출하는 알고리즘을 Geometric based feature extraction이라 하고, 특징들을 특별히 facial landmark라 한다. facial landmark는 얼굴의 위치를 특정 위치로 정렬시키는 역할을 한다. 때문에 정렬 이후의 모든 이미지들의 facial landmark의 위치들은 대부분 동일한다. 이를 통해 얼굴 인식 알고리즘의 효율을 더 높일 수 있다.

### • SVM(Support Vector Machine)

SVM은 기계학습 알고리즘의 일종이다. 기계학습 알고리즘이란 주어진 데이터들에 대해 기계학습을 수행하여 판별 기준을 구축함으로써 새로운 데이터를 입력 받았을 때 그 데이터를 판별 기준에 따라 판별하는 과정을 말한다. 다음은 SVM의 기본적인 원리를 나타낸 그림이다.



우리의 목표는 흰색 원과 검은색 원을 구분하는 것이다. 왼쪽은 일반적인 판별분석방법이다. 두 그룹 안의 원들 각각에 대해 데이터 간 거리를 측정하여 그 중 최적의 초평면을 구함으로써 흰색과 검은색 그룹을 나누는 것이다. 반면 오른쪽의 SVM에서는 두 그룹 사이의 경계에 있는 데이터에 초점을 둔다. 두 그룹 사이의 경계인 H1과 H2의 선을 그은 후 그 가운데에 새로운 선을 만들어 최적의 초평면을 구하는 것이다. H1과 H2 사이에 새롭게 등장한 new data는 일반적인 판별분석방법에서는 검은색 원으로 잘못 판별되지만, SVM에서는 흰색 원으로 정상적으로 판별됨을 확인할 수 있다.

### 3-2) 프로젝트 개발 환경

운영체제	Ubuntu 16.04.1 LTS
개발 언어	Python 2.7.12
라이브러리	OpenCV 3.1

## 4. Goal/Problem & Requirement

이번 프로젝트의 목표는 사람의 사진을 넣으면 남녀를 구분해 알려주는 프로그램을 만드는 것이다. 사용자가 테스트 하기 위해 웹 브라우저를 이용하여 웹을 제작할 것이다. 다양한 조건의 사진에 대해서 전 처리를 통해 variant 를 최소화 해야 하며, 적절한 분류 알고리즘을 선택하여 최적의 결과를 도출 해야 한다. 이미지의 scale과 조명에 상관없이 alignment, crop, rotation 등 적절한 전 처리가 필요 하며, gender detection을 위한 feature를 적절하게 정의 해야 한다. 또 여러 기계학습 알고리즘 중에서 성능을 비교하여 적절한 모델을 선택해야 한다. 훈련 시간과 테스트 시간이 알려진 평균적 요구 치를 충족해야 한다. 빅데이터 분석의 주요 문제 중 하나인 Overfitting을 피하기 위한 방법론을 제시하여 이를 적용하여야 한다

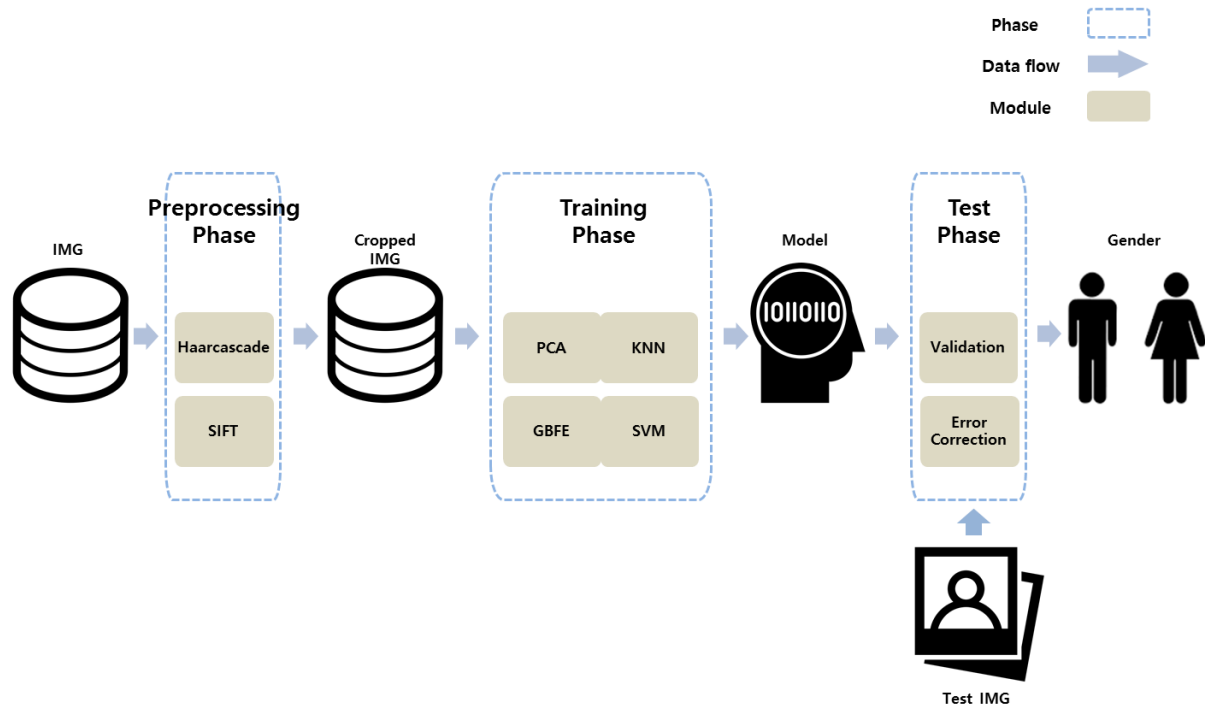
## 5. Approach

이번 프로젝트는 크게 세 단계로 나누어 접근해보려 한다. 첫 번째 단계는 얼굴을 추출해내는 과정이다. 사진이 주어졌을 때, OpenCV에서 제공하는 알고리즘들을 기반으로 얼굴을 찾아낸다. 이 때, Haar Feature-based Cascade Classifier를 이용해 사람의 얼굴만을 찾아내 crop한다. 또한, 빛의 양 또는 얼굴이 회전한 각도에 따라 얼굴이 다르게 인식될 수 있으므로 grey normalization을 통해 빛의 밝기를 표준화하고 SIFT를 이용해 얼굴 인식에 용이한 이미지로 전처리 해준다. 두 번째는 추출한 이미지를 이용해 기계를 학습시키는 과정이다. PCA와 Geometric Based Feature Extraction의 두 가지 방법을 이용해 남녀 얼굴의 특징을 추출해내고 SVM을 이용해 기계를 학습시킨다. 마지막으로 여러 가지 샘플들을 통해 테스트를 함으로써 목표를 달성한다.



## 6. Architecture

### 6-1) Architecture Diagram



### 6-2) Architecture Description

Gender classification은 크게 Preprocessing, Training 그리고 Test phase로 나뉜다.

우선 Preprocessing phase로 시작한다. 얼굴이미지 데이터베이스로부터 이미지 파일들을 읽어 온 뒤, 이로부터 attribute를 추출하는 알고리즘들이 사용할 Cropped 이미지 데이터베이스를 생성한다. 이미지에서 haarcascade 알고리즘을 이용하여 대상의 얼굴을 찾아낸다. 그리고 얼굴 부분을 cropped 이미지 크기로 잘라내고(200x200), SIFT를 사용하여 전처리 한다. 또한 cropped 이미지로부터 attribute를 잘 추출해내기 위해 histogram Equalization을 수행한다. 이렇게 만들어진 cropped 이미지를 모아서 새로운 데이터베이스를 생성하는 것으로 Preprocessing phase가 종료된다.

다음으로 Training phase를 수행한다. 여기서는 PCA, GBFE 모듈을 통해 cropped 이미지 데이터베이스로부터 attribute를 추출하는 과정과, 이를 기반으로 classification 모델을 생성하는 과정이 연속적으로 수행된다. Preprocessing phase에서 생성된 cropped 이미지 데이터베이스를 각 모듈의 input으로 넣어 Gender에 대한 attribute를 추출한다. 각 모듈로부터 추출된 attribute를 기반으로 classification 모델을 생성하는데, 현재 우리는 PCA 모듈에서 추출된 attribute는 KNN으로, GBFE

모듈에서 추출된 것은 SVM으로 모델링을 했다. 이는 각 classification 모델을 시험해보고, 더 나은 모델링 방법을 적용하기 위함이다. 모델이 생성되면 Training phase가 종료된다.

마지막으로 Test phase에서는 Training phase에서 생성된 모델을 이용해 input으로 들어온 얼굴 이미지의 성별을 추정한다. 우선 input 얼굴이미지를 Preprocessing 모듈을 통해 모델이 판별할 수 있는 형태로 변환한다. 그리고 모델 안에 변환된 이미지를 넣고 해당 이미지가 남자와 여자 중 어떤 곳에 속하는지 결정하는 과정이 수행된다. 현재 우리는 두 가지 모델을 통해 Gender를 판별하는데, 각 모듈이 추정한 결과를 바탕으로 Error Correction 함수에 넣어 정확도를 높인다. 최종적으로 나온 Gender 정보를 통해 input 얼굴이미지를 Gender classification 할 수 있게 된다.

## 7. Implementation Spec

### 7-1) Input/Output Interface

#### • Input

이번 프로젝트에서는 사용자의 얼굴 사진이 input 으로 들어올 것이다. 웹 페이지를 활용하기로 했으므로 가장 처음 input은 사용자가 현재 가지고 있는 사진을 웹 서버에 업로드 하는 것이다. 사진을 선택한 후, 업로드 버튼을 누르면 input이 끝나게 된다.

#### • Output

Output은 input 으로 들어온 이미지 파일을 전 처리하고, 미리 학습시킨 Model에 통과 시켜서 나온 값이다. 남자, 여자, 알 수 없음의 3가지가 나올 수 있고, 이렇게 나온 값을 웹 페이지 화면에 보여줌으로써 프로세스가 완료된다.

### 7-2) Inter-Module Communication Interface

현재 프로젝트에서 Preprocessing, Training, Test 은 굉장히 분절된 모듈들이다. 즉각적인 반응이 그다지 필요 없는 모듈이기 때문에 정교한 인터페이스는 불필요하다. 전처리 과정에서 만들어지는 DB를 Training과 Test 모듈에서 가져오는 방식의 인터페이스를 구현하였다.

### 7-3) Modules

이 프로그램의 모듈은 크게 Preprocessing, Training, Test phase에서 사용되는 모듈들로 구분할

수 있다. Traing phase에서의 모듈은 PCA, GBFE 모델링 모듈로 나뉜다. Preprocessing은 제일 첫 단계로서 사진의 전처리를 담당한다. input으로 들어온 사진은 조명 및 촬영 각도에 따라 다양한 변이가 있을 수 있고, 이는 정확도의 감소를 야기한다. 따라서 평균적인 상태의 이미지로 변환을 해야 학습의 능률을 높일 수 있다. 이를 위해 histogramlqualization, contrast 등 다양한 알고리즘을 통해 균일한 밝기와 대조로 만들고, 넓은 부분 중에서 얼굴 부분만을 잘라낸다. Training 은 전처리된 이미지에서 attribute를 얻어 학습을 진행하는 단계로, LandMark 와 PCA 모듈로 나뉘진다. LandMark 는 얼굴에서 눈 코 입 과 같은 object들 사이의 관계 정보를 추출하여 학습을 진행한다. 모듈 내에 있는 searchLandMark 함수를 통해 이미지에서 원하는 object좌표를 추출하고, makeFeature 함수로 attribute를 만든다. 학습은 SVM을 사용한다. PCA 모듈에서는 이미지에 바로 PCA를 적용해서 attribute를 뽑아낸다. 학습에서는 K-NN 알고리즘을 활용한다. 위의 2가지 방식으로 학습시킨 모델은 pickle 을 통해 저장한다. Test 모듈은 학습된 모델을 통해서 들어온 이미지가 여성인지 남성인지 판단하는 곳이다. Pickle을 통해 저장된 모델을 가져와서 전처리 후 들어온 이미지를 넣으면, 판단 결과가 나오게 된다.

## 8. Current Status

사진을 넣으면 밝기와 대조를 보정해주고, 얼굴을 잘라내는 Preprocessing 과정을 완성하였다. PCA와 Landmark 방식을 통한 attribute 추출 및 학습도 진행 중이다. 다만 전처리 한 이미지에서 눈이나 코 입 등의 위치를 찾는 과정에서 오차가 심해 연구를 진행 중이다.

## 9. Future Work

높은 정확도로 얼굴 내의 object들을 찾는 알고리즘을 구현해야 한다. 그리고 많은 데이터들을 K-NN, SVM, CNN 등의 여러 모델에 학습시키고 결과를 비교하여 가장 효율적인 모델을 찾아내야 한다. Test 결과와 실제 Label을 비교하여 전처리 과정에서 누락되거나, 오염된 정보가 없는지 체크하고 전처리 과정을 보완해야 한다.. 그리고 성능을 높이기 위해 조원 각자 DB를 구축하기로 하였다. 한국인 으로 남녀 각각 10명에 사람당 10장씩 총 200장의 DB를 만들기로 하였다. 또한 최종 시연을 위해 웹 서버를 구축해야 한다. Test가 이원화 되어있어서 불필요한 작업이 진행되고 있어서 이를 하나로 합치는 작업이 필요하다.

## 10. Division & Assignment of Work

항목	담당자
사전 지식 이해 / 개발 환경 구축	팀 전원
Image에서 face recognition	박상문, 박충훈
Face image Normalization	박충훈, 방지우
Face feature 추출	방지우, 박상문
Face DB에 대해 기계 학습	팀 전원
테스트 및 개선, 업데이트	팀 전원
최종 모듈 interface 구현	팀 전원

## 11. Schedule

내용	11월				12월	
	1	2	3	4	1	2
사전 지식 이해 개발 환경 구축	완료					
Image에서 face recognition	완료					
Face image Normalization	완료					
Face feature 추출	○					
Face DB에 대해 기계 학습	○					
테스트 및 개선, 업데이트		○				
최종 모듈 interface 구현			○	○	○	○

## ◆ Appendix

### a. Preprocessing

- cv::cvtColor(image, mat\_gray, CV\_BGR2GRAY)

cvtColor 함수는 첫 번째 인수로 지정된 이미지를 세 번째 인수 형태로 바꾼 후, 두 번째 인수로 돌려 주는 함수이다. 이 때, 세 번째 인수는 옵션이며, CV\_BGR2GRAY는 흑백으로 바꾸어 주는 옵션이다.

- cv::equalizeHist(image, result)

equalizeHist 함수는 첫 번째 인수로 지정된 이미지를 histogram equalization 처리한 이후, 두 번째 인수로 돌려 주는 함수이다.

- detect(image, cascade)

detect 함수는 image로부터 얼굴을 인식하는 함수이다. cascade classifier 또는 Haarcascade는 영상에서 특정 형태의 물체를 찾고자 할 때 사용되는 방법이다.

- cv::createCLAHE clipLimit, tileGridSize)

createCLAHE 함수는 Contrast Limited Adaptive Histogram Equalization 방법을 이용하여 grayscale image를 equalize한다. clipLimit는 contrast의 임계값이며, tileGridSize는 histogram equalization을 위한 grid의 크기이다. 이 크기를 이용해 image는 균등한 크기의 tile로 나뉘어져 equalization이 이루어진다.

## **b. PCA module**

- PCA(), NearestNeighbor(dist\_metric, k), PredictableModel(feature, classifier)

PCA 모델을 생성하기 위한 생성자들이다. PCA, NearestNeighbor를 통해 생성한 객체를 PredictableModel의 인자로 넣어 모델을 계산하기 위한 객체를 생성한다. NearestNeighbor에서 dist\_metric은 classifier를 할 때 거리 값을 계산하는 방법이고, k는 KNN 알고리즘에서 어떤 개체를 classification할 때, 기준으로 삼을 가까운 개체들의 수를 나타낸다.

- PCA::compute(img, img\_class\_index)

이미지와, 해당 이미지가 어떤 class에 속하는 지 (여기서는 male=0, female=1) 나타내는 이미지 class index list를 전달하여 PCA eigenvector를 계산한다. 그리고 이렇게 생성된 모델을 반환한다.

- PCA::extract(img)

이미지를 numpy array로 변환하고, project 함수의 인자로 넘겨 해당 결과값을 반환한다.

- PCA::project(img)

계산된 eigenvector와 이미지의 평균값을 통해 이미지에서 PCA 요소를 계산한 뒤 반환한다.

### c. Landmark module

- find\_eyes(img, classifier)

양 눈의 위치를 반환하는 함수 이다.

- find\_nose(img, classifier)

코의 위치를 반환하는 함수이다.

- find\_mouth(img, classifier)

입의 위치를 반환하는 함수이다.

- trans\_data(raw\_data)

입, 눈, 코의 위치를 Input으로 받아 적절한 attribute 들을 생성해 반환하는 함수 이다.

- sklearn::preprocessing::scale(data)

데이터들을 Input으로 받아 표준화시켜 반환한다.

- sklearn::svm::SVC::fit(X, Y)

설정한 SVC 모델에 데이터 X, label Y를 학습시킨다.

### d. Test module

- MV::validate(img, img\_class\_index)

MV로 객체의 함수로, 분류하고자 하는 이미지와 이미지 클래스 인덱스를 인자로 받는다. 우선 이미지들을 모델에 따라 male, female로 분류한다. 그리고 img\_class\_index에 따라 각 이미지가 잘 분류되었는지 확인한 뒤, 분류 정확도를 (분류에 성공한 이미지 수)/(전체 이미지 수)로 계산한다.

- sklearn::svm::SVC::score(X, Y)

학습 시킨 모델에 test set X, label Y로 테스트를 진행한다.