

# 영상처리

CNN 을 이용한 얼굴, 나이 예측 : 시청연령제한

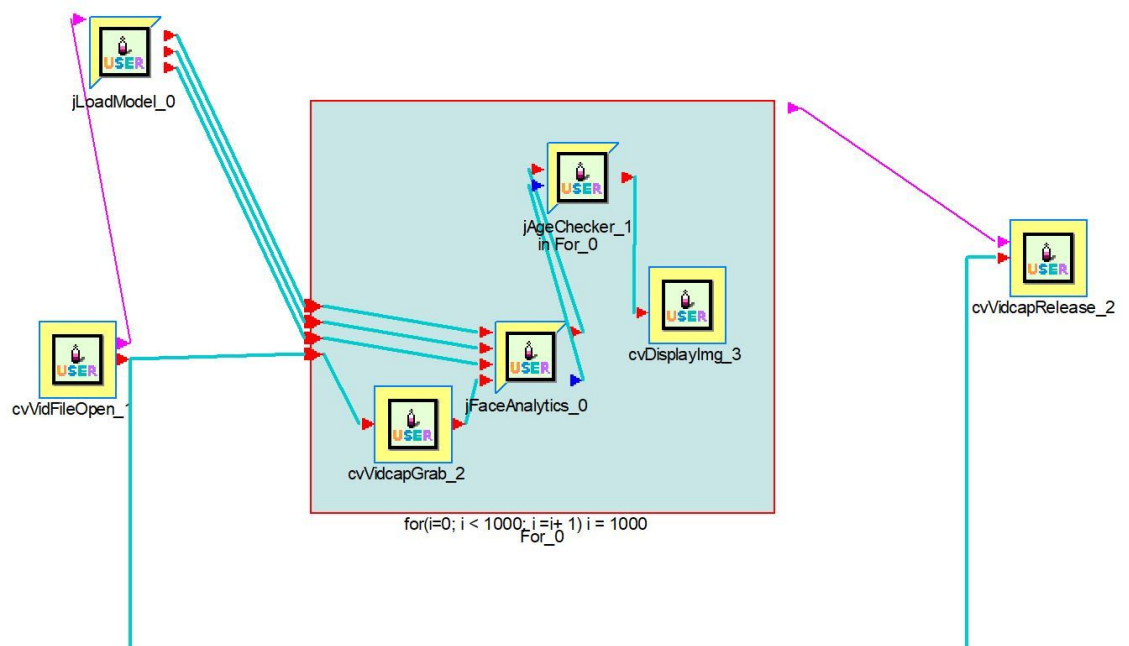
조혁준

## 목적

- 웹캠이나 동영상으로 부터 연속적으로 입력되는 디지털 영상에서 opencv 함수를 이용하여 얼굴 영역을 검출하여 연령을 검출하는 머신러닝을 통해 학습된 모델 기반 나이 및 성별을 예측한다.
- 예측한 나이를 기반으로 시청연령을 일정 나이를 기준으로 시청을 제한한다.

## 내용

- 전체적인 시스템 구조를 Figure 1. 과 같은 비주얼 프로그램으로 설계하였다.



(Figure. 1)

- Figure 1. 의 jLoadModel 은 KScOpenCvUtils.h 에서 Figure 2. 와 같이 정의한 UserDefineArray 를 이용하여 모델을 읽는 함수이다.
- 다음과 같이 UserDefineArray 를 적용하는 이유는 웹캠이나 동영상과 같이 프레임을 for loop 내부에서 읽고 처리하는데 있어서 for loop 내부에서 다음과 같은 모델을 읽을 경우 프로그램의 속도측면에서 성능 저하가 발생하기때문이다. 즉, for loop 밖에서 모델을 읽어와 코드의 중복성을 줄이기 위함이다.

```
class JoModelReader {
private:
    int typeID;

public:
    void InitializeID() { typeID = 2014104152; }
    bool RightDataType() { return (typeID == 2014104152); }
    int GetTypeID() { return typeID; }
    void Release() {}
    cv::dnn::Net face_Model;
    cv::dnn::Net gender_Model;
    cv::dnn::Net age_Model;
};
```

(Figure. 2)

- Figure 3. 의 코드는 UserDefineArray 를 이용하여 모델의 경로를 이용해 opencv 의 함수를 사용하여 모델을 읽는 함수의 코드이다.
- 이를 이용하여 다양한 모델을 읽어와 적용할 수 있다.
- 본 프로젝트에서는 face 모델과 age 모델을 사용하였다.
- face 를 검출하는 모델은 SSD(Single Shot Detector) 알고리즘을 이용하여 학습한 모델이다. SSD 는 입력 영상에서 특정 객체의 클래스와 위치, 크기 정보를 실시간으로 추출할 수 있는 객체 검출 딥러닝 알고리즘이다.
- SSD 알고리즘은 기본적으로 다수의 클래스 객체를 검출 할 수 있지만 해당 프로젝트에서는 오직 얼굴 객체의 위치와 크기를 알아내도록 훈련된 학습 모델을 사용하였다.
- Figure 4. 는 SSD(Single Shot Detector) 의 architecture 이다. 기본 구조나 보조 구조에서 얻은 feature map 들은 각각 다른 convolutional filter 에 의해 결과값을 얻게 된다.
- $m * n$  을  $p$  채널을 가지고 있는 feature map 은 각 위치 마다  $3 * 3 * p$  kernel 들 을 적용할 수 있으며, 각 kernel(filter) 은 카테고리 점수나 bounding box offset 점수를 알려주게 된다.

```

int CDECL jLoadModel(KScUserDefineArray* outFaceModel, KScUserDefineArray* outGenderModel, KScUserDefineArray* outAgeModel)
{
    JoModelReader *facePtr = NULL;
    JoModelReader *genderPtr = NULL;
    JoModelReader *agePtr = NULL;

    // Model path.
    // Face model.
    string faceProto = "C:\\Program Files (x86)\\MTES2.0\\FMS\\JoHyukJun\\models\\opencv_face_detector.pbtxt";
    string faceModel = "C:\\Program Files (x86)\\MTES2.0\\FMS\\JoHyukJun\\models\\opencv_face_detector_uint8.pb";

    // Age model.
    string ageProto = "C:\\Program Files (x86)\\MTES2.0\\FMS\\JoHyukJun\\models\\age_deploy.prototxt";
    string ageModel = "C:\\Program Files (x86)\\MTES2.0\\FMS\\JoHyukJun\\models\\age_net.caffemodel";

    // Gender model.
    string genderProto = "C:\\Program Files (x86)\\MTES2.0\\FMS\\JoHyukJun\\models\\gender_deploy.prototxt";
    string genderModel = "C:\\Program Files (x86)\\MTES2.0\\FMS\\JoHyukJun\\models\\gender_net.caffemodel";

    facePtr = AttachNewObjToUDA(facePtr, outFaceModel);
    facePtr->face_Model = readNetFromTensorflow(faceModel, faceProto);

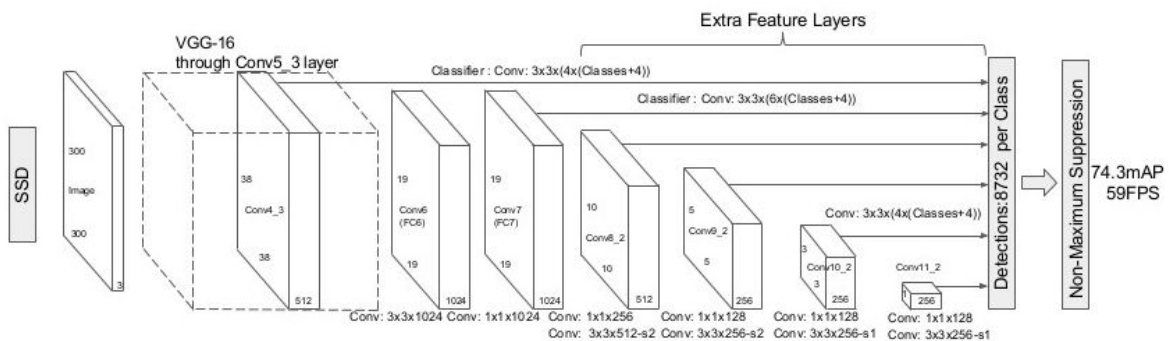
    genderPtr = AttachNewObjToUDA(genderPtr, outGenderModel);
    genderPtr->gender_Model = readNet(genderModel, genderProto);

    agePtr = AttachNewObjToUDA(agePtr, outAgeModel);
    agePtr->age_Model = readNet(ageModel, ageProto);

    return TRUE;
}

```

(Figure. 3)



(Figure. 4)

- Figure 5. 는 Figure 1. 의 jFaceAnalytics 에서 얼굴을 검출하고 나이를 예측하는데 필요한 모델들을 불러오는 함수들이다.  
jFaceAnalytics 이 나이와 얼굴을 검출하는데 필요한 모델을 사용하는데 있어서 jLoadModel 에서 읽어온 모델들을 호출하는 역할을 수행한다.
- jFaceAnalytics 에서 얼굴을 검출하는 과정에서 net 객체에 입력으로 사용하기 위하여 blob 객체를 생성해야한다 blob 객체는  $1 * 3 * 300 * 300$  형태의 4차원 행렬이며 이를 통해 생성된 net 객체는  $1 * 1 * N * 7$  사이즈의 4차원 행렬을 출력한다. 해당 행렬의 첫 두 개의 차원 크기는 항상 1이며 네 번째 차원 크기는 항상 7이다. N 은 검출된 얼굴

후보의 영역 개수를 나타내며 최대 200 까지 검출 가능하다. 1, 2 차원 크기는 항상 1이므로 3, 4 차원만 이용하여 새로운 2차원 행렬이 구성 가능하다. 이렇게 새로 구성한 2차원 행렬의 0번과 1번 열에는 항상 0과 1로 저장된다. 2번째 열에는 face confidence 가 저장되며 해당 confidence 는 0부터 1 사이의 실수이다. 1에 가까운 값을 가질 수록 얼굴일 가능성이 높다고 판단한다. 3번부터 6번 열에는 얼굴 영역 사각형의 좌측 상단 좌표(x1, y1) 와 우측 하단 좌표(x2, y2) 가 저장된다. 이와 같은 사각형 좌표는 영상의 가로와 세로 크기를 1로 정규화하여 저장된 좌표이므로 실제 픽셀 좌표는 영상의 가로 및 세로 크기를 곱해서 계산해야한다. 또한 각 행은 face confidence 가 높은 순서대로 저장되며 최종적으로 행을 읽으면서 face confidence 를 임계값을 기준으로 출력한다.

```
// Get the face model.
JoModelReader *faceNet = NULL;
faceNet = GetNewObjFromUDA(faceNet, faceModel); // Convert UserDefinedType to Net Type
if (faceNet == NULL) { // if NULL, pointer is empty or different data type
    ::MessageBox(NULL, "ERROR: Type Mismatched or Memory not Initialized", "jFaceAnalytics", MB_OK);
    return FALSE;
}

// Get the gender model.
JoModelReader *genderNet = NULL;
genderNet = GetNewObjFromUDA(genderNet, genderModel); // Convert UserDefinedType to Net Type
if (genderNet == NULL) { // if NULL, pointer is empty or different data type
    ::MessageBox(NULL, "ERROR: Type Mismatched or Memory not Initialized", "jFaceAnalytics", MB_OK);
    return FALSE;
}

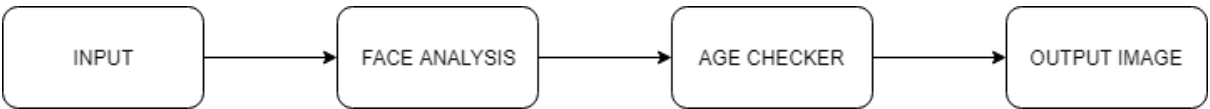
// Get the age model.
JoModelReader *ageNet = NULL;
ageNet = GetNewObjFromUDA(ageNet, ageModel); // Convert UserDefinedType to Net Type
if (ageNet == NULL) { // if NULL, pointer is empty or different data type
    ::MessageBox(NULL, "ERROR: Type Mismatched or Memory not Initialized", "jFaceAnalytics", MB_OK);
    return FALSE;
}
```

(Figure 5.)

- Figure 6. 은 나이를 식별하는 모델에 사용된 네트워크의 architecture 이다.
- 나이 추정을 위한 초기 방법은 얼굴 형상의 다른 측정 사이의 비율을 계산하는 것에서 기초한다. 즉, 일단 얼굴 특징(눈, 코, 입, 턱 등)이 localize 되고 그 크기와 거리를 측정하면 그 사이의 비율을 계산하여 직접 만든 규칙에 따라 얼굴을 다른 연령 범주로 분류하는 데 사용한다.
- CNN 은 나이에 따라 분류된 제약 없는 dataset 의 작은 크기를 고려하더라도 분류 결과를 사용하는데 사용될 수 있다.
- 적용된 모델의 단순성은 더 많은 훈련 데이터를 사용할 경우 더 정교한 시스템으로 개선할 수 있다.


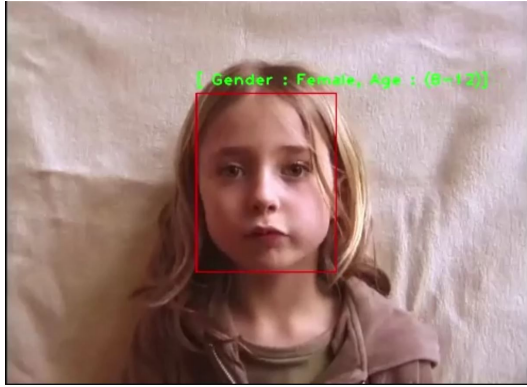


- jAgeChecker에서는 jFaceAnalytics를 통해 파싱된 나이 값과 영상을 가지고 나이 값이 설정한 임계값에 도달하지 못할 경우 영상을 출력하지 않는 분기 문의 역할을 수행한다.



(Figure 7.)

실행 결과 및 분석

| Original  | Face Analysis  |
|---|--|
|  |  |

(Figure. 8)

- Figure 8. 에서 볼 수 있듯이 jFaceAnalytics를 통해 Original 영상에서 얼굴의 위치 정보(Bound Box)와 성별, 나이 정보를 확인 및 추출할 수 있었다.
- Figure 8. 에서 원본 영상에서 실제 나이는 7세이며 나이를 예측한 결과는 8세에서 12세의 범위로 추론하는 것으로 보아 정확도가 높게 추출된 것을 확인 할 수 있었다.

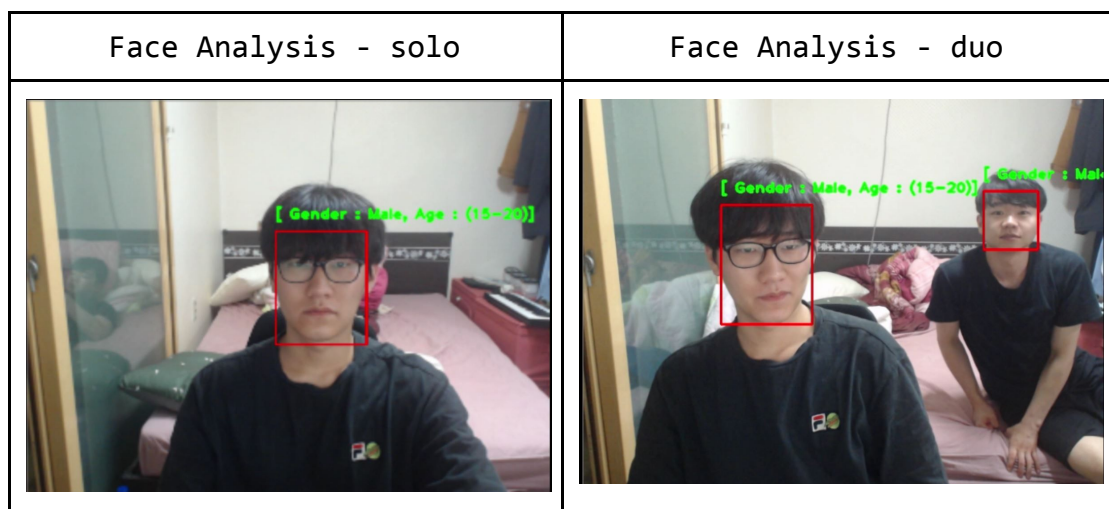
| Original | Face Analysis |
|----------|---------------|
|----------|---------------|





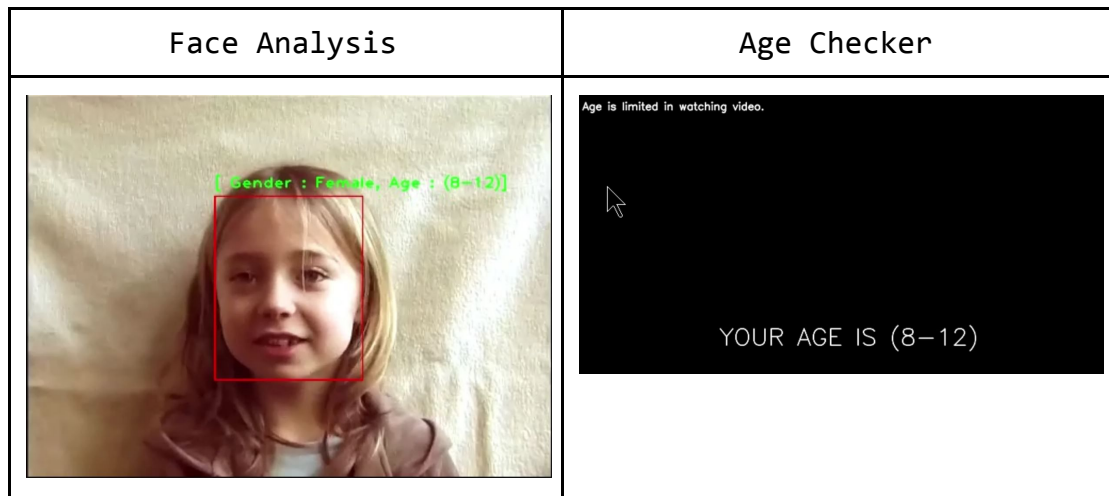
(Figure 9.)

- 반면 얼굴의 위치정보는 대부분의 경우 정확하게 인식하는 반면 나이와 성별을 판별하는 모델의 정확도는 다소 낮은 것 또한 확인 할 수 있었다.
- Figure 9. 에서 원본 영상의 나이는 7세인데 작성한 함수가 추출한 나이는 25세에서 32세로 판단한 것을 확인 할 수 있었다.
- 위와 같이 나이와 성별을 적절히 추론하지 못하는 경우는 대부분 입력 영상의 얼굴의 표정이 일그러지는 것과 같이 정상적인 표정이 아닐 경우 발생하는 것을 확인 할 수 있었다.



(Figure 10.)

- Figure 10. 에서는 입력 영상에 등장하는 인물의 수 가 변화하더라도 얼굴의 위치 정보, 성별 정보, 나이 정보 모두 추론하는 것을 확인 할 수 있었다.
- 화면상에서 다양한 각도와 위치에서도 얼굴의 위치정보를 획득하는데에는 높은 성능을 보여준 것을 확인했다.



(Figure 11.)

- Figure 11. 은 얼굴을 분석하여 나이를 추출하여 최종적으로 시청을 제한하는 영상이다.
- Figure 11. 에서 원본 영상이 나이는 7세이며 얼굴 분석을 통해 추론된 나이는 8세에서 12세이다 jAgeChecker 에서 나이의 임계값을 15세 이하로 설정하였을 경우 “Age is limited in watch video.” 란 문구와 함께 현재 인식되고 있는 나이를 중앙에 “YOUR AGE IS \$AGE\_VAL” 과 같이 출력하는 것을 확인 할 수 있다.

## 결론

- Original Image 에서 SSD 알고리즘을 통해 학습한 모델을 이용하여 얼굴을 인식하는 과정에서는 어려움이 없었으나 for loop 내부에서 얼굴을 못찾았을 경우 예외처리 과정에서 return FALSE 할 경우 프로그램이 동작하지 않는 문제를 해결하려 여러 방면으로 시도했지만 찾지 못해 어려움을 겪었다.
- 나이와 성별을 추론하는 CNN 기반으로 학습된 모델을 통해 입력 영상에서 나이와 성별을 추론하는 과정에서 얼굴 부분이 crop 된 이미지의 인물의 표정과 각도에 따라 변동하는 과정이 있었다. 해당 부분은 모델을 dataset 을 추가하여 fine tuning 과정을 거칠 경우 추론하는 과정에서 정확도의 성능이 상승할 것으로 예상된다.
- 나이와 성별을 예측하는 것에 착안하여 광고 추천과 같은 다양한 어플리케이션으로 개발할 수 있을 것으로 예상되며 머신러닝을 통해 직접



목적에 맞는 모델을 생성하여 다양한 목적으로 사용할 수 있을것으로 기대 된다.

## **참조**

- 황선규. OpenCV 4로 배우는 컴퓨터 비전과 머신 러닝. (주) 도서출판 길벗.
- <https://junjiwon1031.github.io/2017/09/08/Single-Shot-Multibox-Detector.html>
- <https://github.com/eveningglow/age-and-gender-classification>
- Gil Levi and Tal Hassner. Age and Gender Classification using Convolutional Neural Networks. Department of Mathematics and Computer Science The Open University of Israel.