**캡스톤 디자인 I**

**종합설계 프로젝트**

|  |  |
| --- | --- |
| 프로젝트 명 | Auto Blur With Object Detection |
| 팀 명 | Bblur |
| 문서 제목 | 중간보고서 |

|  |  |
| --- | --- |
| **Version** | 1.4 |
| **Date** | 2019-04-18 |

|  |  |
| --- | --- |
| **팀원** | 김 용욱 (조장) |
| 김 대희 |
| 권 보경 |
| 이 나영 |
| 채 승훈 |
| **지도교수** | 김 영만 교수 |

|  |
| --- |
| **CONFIDENTIALITY/SECURITY WARNING**  이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 “Auto blur with object detection”를 수행하는 팀 “Bblur”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “Bblur”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다. |

**문서 정보 / 수정 내역**

|  |  |
| --- | --- |
| **Filename** | 중간보고서-AutoBlurWithObjectDetection.doc |
| **원안작성자** | 김용욱, 김대희, 권보경, 이나영, 채승훈 |
| **수정작업자** | 김대희,이나영,권보경,김용욱 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 수정날짜 | 대표수정자 | Revision | 추가/수정 항목 | 내 용 |
| 2019-04-13 | 김대희 | 1.0 | 최초 작성 | 모든 항목 대략 작성 |
| 2019-04-14 | 이나영 | 1.1 | 추가 작성 | 모바일 어플리케이션 개발 |
| 2019-04-15 | 권보경 | 1.2 | 추가 작성 | 모바일 어플리케이션 개발 |
| 2019-04-16 | 김용욱 | 1.3 | 추가 작성 | 누락 내용 추가 |
| 2019-04-18 | 김용욱 | 1.4 | 추가 작성 | 전체적인 내용 수정 |
|  |  |  |  |  |
|  |  |  |  |  |

**목 차**

[1 프로젝트 목표 4](#_Toc6520355)

[2 수행 내용 및 중간결과 5](#_Toc6520356)

[2.1 계획서 상의 연구내용 5](#_Toc6520357)

[2.1.1 영상 파일 입력 및 출력 5](#_Toc6520358)

[2.1.2 딥러닝(YOLO) 모델 사용 및 학습 5](#_Toc6520359)

[2.1.3 블러 처리 6](#_Toc6520360)

[2.2 수행내용 7](#_Toc6520361)

[2.2.1 영상 파일 입력 및 출력 7](#_Toc6520362)

[2.2.2 딥러닝(YOLO) 모델 사용 및 학습 7](#_Toc6520363)

[2.2.3 블러 처리 7](#_Toc6520364)

[3 수정된 연구내용 및 추진 방향 8](#_Toc6520365)

[3.1 수정사항 8](#_Toc6520366)

[3.1.1 특정 인물의 얼굴을 제외한 블러 처리 기능 8](#_Toc6520367)

[3.1.2 브랜드 로고 블러 처리 기능 9](#_Toc6520368)

[3.1.3 모바일 어플리케이션 개발 9](#_Toc6520369)

[3.1.4 결과물 출력 10](#_Toc6520370)

[3.1.5 바디 블러 기능 삭제 10](#_Toc6520371)

[4 향후 추진계획 11](#_Toc6520372)

[4.1 향후 계획의 세부 내용 11](#_Toc6520373)

[4.1.1 특정 인물의 얼굴을 제외한 블러 기능 11](#_Toc6520374)

[4.1.2 브랜드 로고 블러 기능 12](#_Toc6520375)

[4.1.3 모바일 어플리케이션 개발 12](#_Toc6520376)

[4.1.4 결과물 출력 14](#_Toc6520377)

[5 고충 및 건의사항 14](#_Toc6520378)

# 프로젝트 목표

이 프로그램은 입력(실시간 웹캠, 이미지 파일, 영상 파일)에서 특정인을 제외한 얼굴을 블러 처리하여 출력해준다. 또한 모바일 어플리케이션을 개발해서 이미지나 영상 파일에 대해 서버를 통한 블러 처리를 실행할 수 있도록 한다. 추가적으로 브랜드 로고를 인식하여 블러 처리된 결과물을 생성하는 것을 목표로 한다.

# 수행 내용 및 중간결과

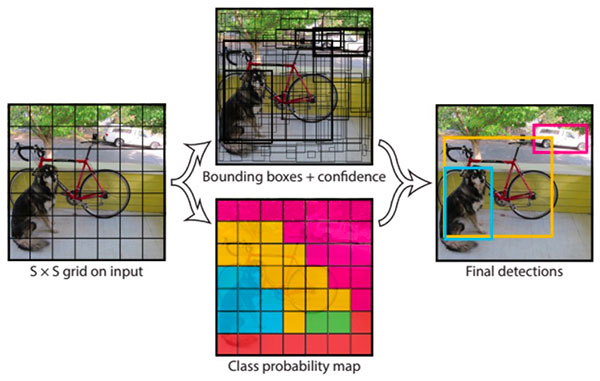
## 계획서 상의 연구내용

### 영상 파일 입력 및 출력

사용자가 지정한 파일을 입력으로 받으면 OpenCV 라이브러리를 이용하여 입력을 프레임마다 나누어 수정할 수 있도록 한다. 또한 프레임마다 나누어 수정한 영상의 이미지들을 다시 영상으로 출력해준다.

### 딥러닝(YOLO) 모델 사용 및 학습

영상의 특정 영역을 탐색하여 특정 얼굴 또는 브랜드 로고를 찾아 블러 처리를 자동화하는 프로그램을 개발하기 위해서는 이미지의 어떤 부분이 얼굴인지 또는 브랜드 로고 인지 파악하는 기능이 필요하다. 따라서 우리는 기존의 영상처리 기술보다 더 효율적이라고 알려져 있는 딥러닝을 활용한 Object Detection에 주목했다. Object detection은 특정한 이미지에서 물체가 존재하는지 인식, 분류하고, 또한 인식한 Object가 이미지 상 어디에 위치하는지를 탐색하는 알고리즘이다. 아래 이미지는 Object detection 모델 중 하나인 YOLO가 이미지 탐색에 대해 어떻게 접근하는지 대략적으로 나타낸다.



**그림 1 YOLO 이미지 탐색 과정**

Object detection은 2013년의 R-CNN부터 시작되어 Fast R-CNN, Faster R-CNN, YOLO v1, SSD 등을 거쳐 현재까지 발전해왔다. 그중 Faster R-CNN 까지는 속도 면에서 영상에서 전통적인 프레임 기준인 30fps에 많이 미달하는 모습을 보여줬다. 하지만 YOLO라는 모델부터 TITAN X 그래픽카드를 사용하여 실행했을 때 최대 45fps까지도 나오기 시작했다. 이는 실시간 영상에도 적용할 수 있을 정도다. 따라서 우리가 원하는 영상 처리를 자동화하는 프로그램에서 Object detection을 사용해야 효율적으로 개발할 수 있다.

### 블러 처리

**그림 2 이미지 모자이크 처리**

Object detection을 사용해서 목표의 적절한 이미지 상 좌표를 얻는다. 얼굴에 대한 블러 처리를 하는 경우에는Face Recognition 라이브러리를 사용하여 그 좌표 내의 얼굴에서 특징을 추출하고, 미리 저장해둔 사진 얼굴들의 특징들과 비교하여 다른 사람이라고 인식하면 블러 처리해준다. 브랜드 로고에 대하여 블러 처리를 하는 경우에는 얻은 좌표내의 특징과 학습시켜놓은 데이터와 비교하여 브랜드 로고로 인식하면 블러 처리를 해준다. 블러 처리는 OpenCV에서 제공하는 픽셀값들을 정렬 한 후에 픽셀의 중간 값으로 채워버리는 medianBLur를 이용하여 진행한다.

## 수행내용

### 영상 파일 입력 및 출력

OpenCV를 이용하여 입력을 프레임마다 나누어 수정할 수 있도록 구현했다. 또한 프레임마다 나누어 수정한 영상의 이미지들을 다시 영상으로 출력해준다. 그런데 openCV에는 음성에 관한 기능이 없어서 실시간 웹캠이 아닌 음성이 있는 영상 파일에는 ffmpeg 툴을 이용해야 한다. 순서는 다음과 같다.

**순서도 1 영상 파일 입출력 및 음성 처리**

### 딥러닝(YOLO) 모델 사용 및 학습

먼저 현재까지 있었던 딥러닝 기반 객체인식 연구 중 우리 프로젝트에 가장 적절한 알고리즘이 무엇인지 연구했다. 연구 내용은 Faster-RCNN 계열부터 SSD 알고리즘, Mask-RCNN 알고리즘 등이었다. 우리 프로그램에서는 실시간 웹캠을 블러 처리하고, 특히나 추가 구현 목표에서 목표 얼굴을 학습시켜 누구의 얼굴인지 분류를 하기 위해서는 속도가 빠른 것이 중요하다. 따라서 기존 생각대로 YOLO v3 모델이 가장 적절한 것으로 판단했다.

그래서 YOLO v3 모델 중에서 얼굴만 찾도록 학습 되어있는 모델을 찾아서 사용하고 있다. 얼굴을 생각보다 잘 인식해서 따로 학습시킬 일은 없었다. 그러나 추가된 구현 목표 중 브랜드 로고를 인식하는 모델은 YOLO v3 기존 모델에서 분류기 부분만 전이학습 시켜보고 있다.

### 블러 처리

우선 오브젝트 디텍션 및 얼굴분류를 통해 학습되지 않은 데이터에 대하여 좌표를 얻고, 그 좌표의 이미지에 대해서 블러 처리를 한다. 블러 처리는 OpenCV에서 제공하는 픽셀 값들을 정렬한 후에 픽셀의 중간 값으로 채워버리는 medianBLur를 이용하여 진행한다.

# 수정된 연구내용 및 추진 방향

## 수정사항

### 특정 인물의 얼굴을 제외한 블러 처리 기능

**그림 3 특정 인물 얼굴 제외 블러**

영상에서 특정 인물의 얼굴을 제외하고 나머지 얼굴을 블러 처리하는 기능을 추가했다. 특정 인물의 얼굴을 학습시키기 위해 양쪽 측면의 얼굴 사진 2장, 아래와 위로 기울인 얼굴 사진 2장, 정면 사진 1장을 찍는다. 실시간 웹캠에서 블러 처리한 영상을 보여주거나 또는 영상 파일을 편집할 수 있다.

개인 방송에서 진행자 얼굴을 제외하고 나머지를 블러 처리를 할 수 있다면, 길거리에서 라이브 방송을 할 때 행인들의 초상권을 보호하는데 더 유용하게 쓰일 수 있어서 기능을 추가하게 되었다.

### 브랜드 로고 블러 처리 기능

**그림 4 브랜드 로고 블러**

유명한 브랜드들의 로고를 탐색하여 블러 처리할 수 있는 기능이다. TV 방송 프로그램에서 공식 간접광고가 아닌 브랜드의 로고는 편집할 때 하나씩 다 가려야 한다. 심지어 인기 있는 개인 방송인들도 간접광고가 아닌 브랜드 로고를 검정색 테이프로 가리기도 한다. 프로그램을 사용하면 브랜드 로고를 자동으로 블러 처리해주므로 인력의 낭비를 방지할 수 있는 효과를 기대하고 있다.

### 모바일 어플리케이션 개발

전문적으로 영상을 다루는 사람이 아닌 일반인들도 간단하게 사용할 수 있는 안드로이드용 어플리케이션이다. 벚꽃축제, 콘서트, 식당 등 사람들이 많은 장소에서 찍은 사진을 바로 sns에 올릴 때 유용할 것이라고 판단하여 추가적으로 개발을 진행하게 되었다. 블러 처리를 할 서버를 따로 두어 어플리케이션이 무겁지 않게 하고, 카메라 기능을 추가하여 사진을 찍는 과정을 포함한 전 과정을 어플 내에서 해결할 수 있도록 한다.

### 결과물 출력

사용자가 객체 탐색 기준을 쉽게 선택할 수 있도록 하고, 입력할 영상 파일을 쉽게 지정할 수 있는 등의 기능을 포함하는 UI를 구현한다. 실시간 웹캠 영상에 대해서는 사용자 PC를 RTSP서버로 사용해서 사용자 PC의 웹캠 영상을 AWS 서버로 전송한 뒤, 블러 프로세스를 거친 후에 웹 페이지를 통해 처리된 영상을 재생한다. 사용자는 본인 pc에서 웹 페이지에 접속해서 영상을 볼 수 있다.

### 바디 블러 기능 삭제

사람 전체를 찾아서 블러 처리하는 기능은 얼굴을 찾는 기능이 잘 구현된다면 상대적으로 필요성이 떨어진다고 판단되어 삭제하기로 했다.

# 향후 추진계획

## 향후 계획의 세부 내용

### 특정 인물의 얼굴을 제외한 블러 기능

**순서도 2 특정 얼굴을 학습하여 나머지와 분류하기**

파이썬의 face\_recognition 라이브러리와 YOLO 알고리즘을 사용한다. Face\_recognition은 얼굴을 탐색한 후 그 얼굴의 특징을 학습되어 있는 얼굴들과 비교해서 차이의 정도를 계산한다. 단계적으로는 목표 얼굴을 학습시키고, 이미지에서 모든 얼굴을 탐색한다. 그리고 탐색한 얼굴과 학습되어 있는 얼굴들의 특징을 비교하여 그 중 가장 비슷한 학습된 얼굴을 도출해낸다. 일정 threshold를 줘서 가장 닮은 얼굴이더라도 일정 수치보다 차이가 크다면 모르는 얼굴로 판단하게 된다.

이 라이브러리에도 얼굴 탐색 기능은 있지만 YOLO보다 기능은 떨어진다. 따라서 우리는 더 완성도 높은 프로그램을 개발하기 위해 얼굴을 탐색하는 부분을 YOLO로 대체하여 구성하고 있다.

### http://image.ntua.gr/iva/datasets/flickr_logos/dataset1_bboxes.jpg브랜드 로고 블러 기능

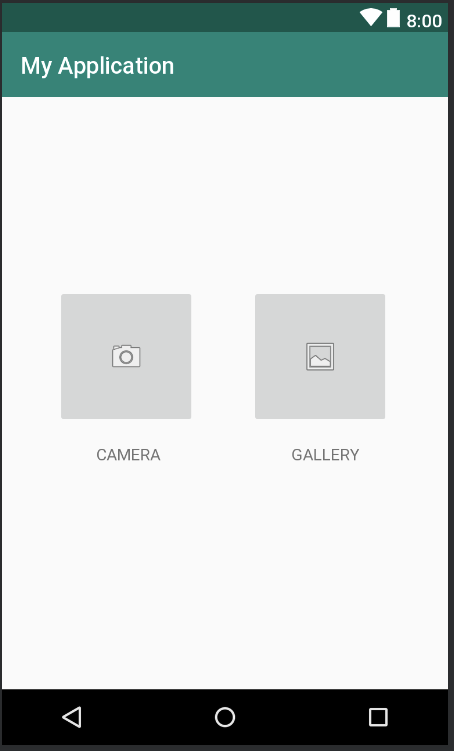
**그림 5 Flickr Logos 27 데이터셋**

국내 브랜드의 로고를 모아 놓은 데이터셋을 찾을 수 없어 Flickr Logos 27 dataset[[1]](#footnote-1)을 사용하여 학습시키려고 한다. keras 기반으로 yolo v3 모델을 전이 학습시키는데, 대략 4500여개의 데이터셋이 있으므로 classifier 부분만 새로 학습시키려고 한다. 모델 학습 환경은 학교에서 제공되는 KMUSW 딥러닝 클라우드를 이용한다.

### 모바일 어플리케이션 개발

Android Studio 개발환경을 통해 안드로이드 버전 어플리케이션을 개발할 예정이다. 어플리케이션 내에서의 블러 처리 과정은 프로그램에서 블러 처리가 진행되는 과정과 동일하며 다음과 같다.

**순서도 3 어플리케이션 블러처리 과정**

**그림 6 어플리케이션 예시**

어플리케이션에서 파일을 선택하는 방법으로는 먼저 사진 혹은 동영상을 찍는 방법이 있다. 이는 카메라 기능을 추가하여 사진이나 1분 미만의 짧은 영상을 찍어서 업로드하는 방법이다. 두번째로는 핸드폰 사진첩을 연결하여 저장되어 있는 사진 혹은 영상을 업로드 하는 방법이다. 파일 업로드 시 묶음으로 1장 이상의 파일을 처리할 수 있도록 한다.

두가지 방법 중 하나를 선택하여 업로드 된 파일을 확인한 후, 해당 파일로 처리를 원한다면 ‘선택’ 버튼을 누른다. ‘취소’ 버튼을 누른다면 다시 파일을 선택할 수 있는 창으로 되돌아간다. 다음 단계에서는 블러 처리를 원하지 않는 대상에 대한 사진을 추가한다. 모든 사람을 처리하기를 원한다면 이 과정은 생략될 수 있다. 서버에 전송된 파일은 블러 처리를 진행한다. 서버에서는 파일에서 처리를 원하지 않는 대상을 제외한 나머지 인물들의 얼굴을 가리게 된다. 블러 처리가 완료된 파일들을 미리보기로 확인할 수 있으며, 저장을 원하는 경우에 한하여 파일들을 사진첩에 일괄 다운로드 한다. (이때, 블러 처리가 완벽하지 않은 부분을 수동으로 처리할 수 있는 기능을 추가한다.)

### 결과물 출력

사용자가 객체 탐색 기준을 쉽게 선택할 수 있도록 하고, 입력할 영상 파일을 쉽게 지정할 수 있는 등의 기능을 포함하는 UI를 구현한다. 실시간 웹캠 영상에 대해서는 사용자 PC를 RTSP서버로 사용해서 사용자 PC의 웹캠 영상을 AWS 서버로 전송한 뒤, 블러링 프로세스를 거친 후에 웹 페이지를 통해 처리된 영상을 재생한다. 사용자는 본인 pc에서 웹 페이지에 접속해서 영상을 볼 수 있게 한다.

# 고충 및 건의사항

다 같이 회의할 공간이 부족합니다.

1. <http://image.ntua.gr/iva/datasets/flickr_logos/> [↑](#footnote-ref-1)