



국민대학교
전자정보통신대학
컴퓨터공학부


캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	동영상 연령제한 검열
팀 명	YouHi
문서 제목	결과보고서

Version	1.3
Date	2020-JUN-09

팀원	이태훈 (조장)
	이인평
	이주형
	김성수
	김민재


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

문서 정보 / 수정 내역

CONFIDENTIALITY/SECURITY WARNING	
<p>이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 "동영상 연령제한 검열"를 수행하는 팀 "YouHi"의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 "YouHi"의 팀원들의 서면 허락 없이 사용되거나, 재가공 될 수 없습니다.</p>	


Filename	최종보고서-동영상 연령제한 검열doc
원안작성자	이태훈, 이인평, 이주형
수정작성자	이태훈, 이인평, 이주형

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2020-06-06	이주형	1.0	최초 작성	
2020-06-08	이태훈	1.1	내용 수정	연구/개발 내용 및 결과물 목록 수정
2020-06-08	이인평	1.2	내용 수정	개발/활용 기술 수정
2020-06-09	이태훈	1.3	최종 마무리	자기평가, 테스트 케이스 수정


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

목 차

1	개요	5
1.1	프로젝트 개요	5
1.2	추진 배경 및 필요성	5
1.3	관련 기술 조사	9
2	개발 내용 및 결과물	12
2.1	목표	12
2.2	연구/개발 내용 및 결과물	13
2.2.1	영상 1차 검열 - Instance Segmentation	13
2.2.2	영상 2차 검열 - Video Classification	15
2.2.3	영상 2차 검열 - Object Detection, Image Classification	17
2.2.4	음성 인식	19
2.2.5	자막 추출	19
2.2.6	형태소 분석 및 욕설 검열	20
2.2.7	Front End	22
2.2.8	Back End	24
2.2.9	정확도 측정	25
2.3	시스템 요구사항	26
2.3.1	시스템 기능 요구사항	26
2.3.2	시스템 비기능(품질) 요구사항	27
2.4	시스템 구조 및 설계도	27
2.5	활용/개발된 기술	28
2.5.1	영상 검열	28
2.5.2	음성 검열	30
2.5.3	자막 검열	32
2.5.4	Front End	32
2.5.5	Back End	33
2.6	현실적 제한 요소 및 그 해결 방안	33
2.6.1	하드웨어	33
2.6.2	소프트웨어	34
2.6.3	기타	34
2.7	결과물 목록	34
2.8	기대효과 및 활용방안	34
3	자기평가	36
4	참고 문헌	37
5	부록	38

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

5.1	사용자 매뉴얼	38
5.2	운영자 매뉴얼	40
5.2.1	AWS Gatewap API 설정하기	40
5.2.2	AWS Lambda 함수 설정하기	41
5.2.3	AWS EC2 인스턴스 생성하기	43
5.3	테스트 케이스	45

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

1 개요

1.1 프로젝트 개요

본 프로젝트는 YouTube 등 동영상 업로드 플랫폼들의 미성년자 시청 불가 영상에 대한 검열 시스템의 단점을 보완한 시스템을 개발하는 것을 목표로 한다. 현재 동영상 업로드 플랫폼들의 영상 검열 시스템은 사용자의 신고 혹은 운영자의 모니터링을 통한 수동 검열로 이루어진다. 이러한 방식은 새로 업로드된 영상을 즉각적으로 검열 할 수 없으며 사용자의 신고를 받거나 운영자의 모니터링에 의해 검열 되기 전까지 청소년들에게 무방비 상태로 노출된다. 이를 방지하기 위하여 영상이 업로드되기 전에 사전 검열을 하여 가이드라인(YouTube Community GuideLine)에 위배되는 내용이 영상에 포함되어 있을 경우, 청소년 시청 불가 콘텐츠로 분류한다. 사전 검열 과정을 통해 청소년에게 유해한 콘텐츠가 노출되는 것을 예방하는 것을 목표로 한다.


1.2 추진 배경 및 필요성

영상 시청 시 사용할 수 있는 동영상 플랫폼에는 다양한 종류들이 있다. 그 중에서도 YouTube 플랫폼은 2019년 7월 기준 타 플랫폼 대비 가장 많은 순이용자를 보유하고 있다. 리서치 전문업체 닐슨코리아클릭의 통계에 따르면 YouTube의 2019년 7월 기준 이용자 수는 2,631만명에 달했고 이는 전년 대비 6.7% 증가한 수치이다. 따라서 우리는 YouTube를 기준으로 진행한다.

주요 OTT 월간 순이용자 (단위=명)			
구분	2018년 7월	2019년 7월	증감률 (%)
유튜브	2,466만	2,631만	6.7
넷플릭스	41만	185만	342
틱톡	87만	180만	107
옥수수	313만	336만	7.1
네이버TV	323만	193만	-40
폭	76만	139만	83

※ 자료=닐슨코리아클릭

<동영상 플랫폼 이용자 현황>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

YouTube의 시스템은 접근성이 뛰어나 누구나 쉽게 자신이 원하는 동영상을 시청하거나 업로드할 수 있다는 장점이 있다. 하지만 동영상 업로드에 관한 제한이 없어서 일부 사용자가 올린 부적절한 내용의 동영상은 제재없이 그대로 미성년자에게 노출되기도 한다. 이는 청소년들에게 부정적인 영향을 주며 청소년 범죄와 같은 사회적 문제로 이어질 수 있다. 현재 YouTube는 이를 예방하기 위해 다양한 가이드 라인을 사용자에게 제시하고 있고, 가이드 라인에 적합하지 않은 영상들은 삭제하거나, 연령 제한을 통해 미성년자들이 시청하지 못하도록 하였다.

정책 및 안전


YouTube를 사용한다는 것은 전 세계 사람들과 함께하는 커뮤니티의 일원이 된다는 것입니다. YouTube의 새롭고 멋진 커뮤니티 기능을 사용하려면 일정 수준 이상의 신뢰가 필요합니다. 수많은 YouTube 사용자들이 이러한 신뢰를 존중하고 있으며 여러분도 책임감 있는 모습을 보여주시리라 믿습니다. 아래 가이드라인을 준수하시면 YouTube를 모두에게 즐겁고 신나는 공간으로 유지할 수 있습니다.

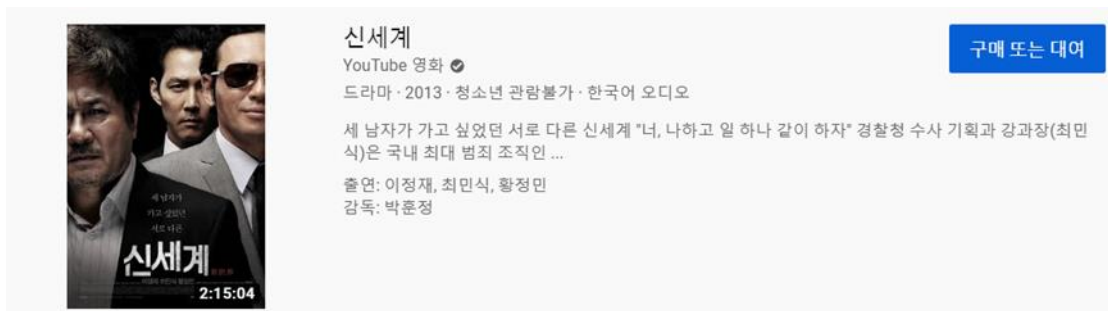
YouTube에 표시되는 콘텐츠가 모두 마음에 들 수는 없을 것입니다. 콘텐츠가 부적절하다고 생각하시면 YouTube팀에서 검토할 수 있도록 **신고 기능**을 통해 신고해 주세요. YouTube팀에서는 신고된 콘텐츠를 연중무휴 검토하여 커뮤니티 가이드의 위반 여부를 판단합니다.

<현재 Youtube 영상 검열 시스템>

현재 YouTube에서는 가이드라인을 준수하지 않는 영상에 대하여 YouTube 운영자들의 모니터링과 사용자들의 신고를 통해서 검열하고 있으며 일부 영상에 대해서는 "노란딱지"라는 YouTube 자체 AI시스템이 검열을 진행한다. 하지만 운영자의 모니터링과 사용자의 신고를 통한 검열 방법은 처리 과정이 길다는 단점이 존재하며 자체 AI시스템은 사용자가 자신의 영상 중 어떤 부분이 가이드라인을 위반하였는지 직접 확인이 불가능하여 오분류 발생시 확인 및 대처가 어렵다는 한계가 있다.

이러한 문제들이 제대로 해결되지 않아 몇몇 영상들은 미성년자에게 부적합한 영상임에도 불구하고 미성년자에 대한 제약이 걸려있지 않은 채로 청소년들에게 노출되고 있다. 영화 <신세계>는 청소년 관람 불가 등급의 영화이다 YouTube에서도 또한 이에 대한 내용을 확인할 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08




<영화 신세계 - Youtube 제공>

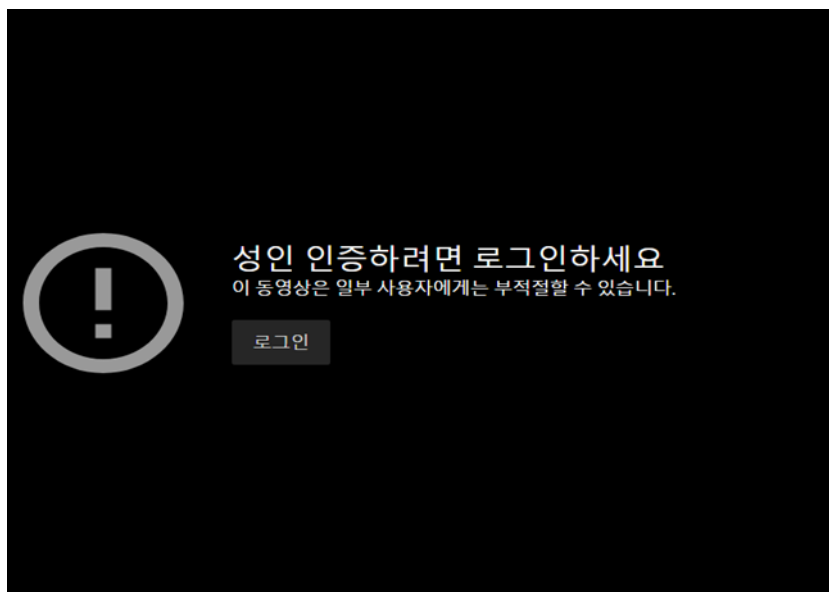
그러나 해당 영화에서 선정적이거나 폭력적인 장면이 고스란히 담겨있는 장면들을 편집해 놓은 영상들은 누구나 시청 가능하게 설정되어 있고, Youtube 정책에 의해 검열되지 않았다. 이러한 영상들 중, 400만이 넘는 조회수를 기록한 영상도 존재했다.



<신세계 명장면 - Youtube 제공>

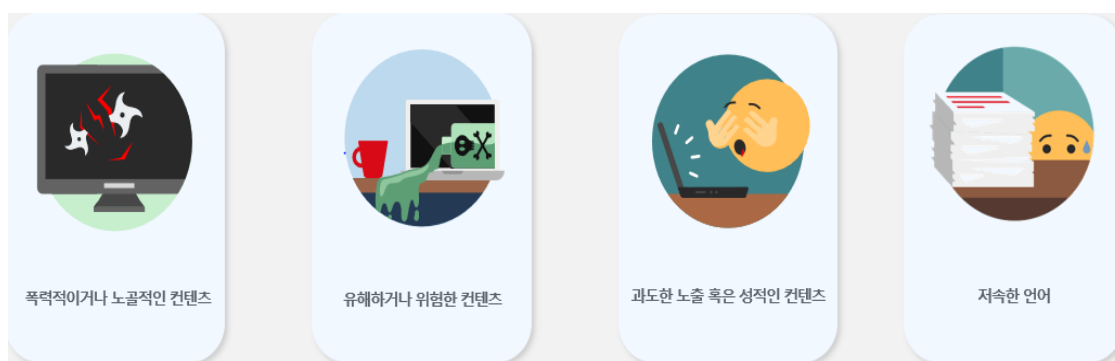
미성년자에게 부적합한 영상은 다음과 같이 개제되어야 한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08



<미성년자 시청 불가 영상 – Youtube 제공>

Youtube 내 많은 영상들이 미성년자에게 부적합한 내용을 담고 있음에도 불구하고, 성인 인증과 같은 별도의 절차 없이 손쉽게 접근이 가능하다. 많은 미성년자들이 별다른 제한없이 청소년 유해물에 접근하고 있음을 알 수 있다. 따라서 본 프로젝트에서는 미성년자들에게 초점을 맞추어 해당 연령대가 시청하기에 부적절한 영상에 대하여 ‘19세 이상 시청가능’ 조건 없이는 영상을 업로드 할 수 없도록 하는 자동 검열 시스템을 개발하기로 했다. 검열의 기준은 Youtube 내의 이용정책 – 연령별 등급제한, 방송통신위원회 내용을 참고하였다.



<Youtube 내 주요 연령제한 가이드라인>

	노출	성행위	폭력	언어	기타
4등급	성기노출	성범죄 또는 노골적인 성행위	잔인한 살해	노골적이고 외설적인 비속어	1. -마약사용조장 -무기사용조장 -도박 2. -음주조장 -흡연조장
3등급	전신노출	노골적이지 않은 성행위	살해	심한 비속어	
2등급	부분노출	착의상태의 성적접촉	상해	거친 비속어	
1등급	노출복장	격렬한 키스	격투	일상 비속어	
0등급	노출없음	성행위없음	폭력없음	비속어없음	

※ 등급기준의 명칭은 방송통신심의위원회 SafeNet 등급기준으로 정함.

<방송통신심의위원회 SafeNet 등급기준>

구분	노출	성행위	폭력	언어
전체가 (초등학생가)	1등급	0등급	1등급	0등급
12세 이상 (중학생가)	2등급	2등급	2등급	1등급
15세 이상 (고등학생가)	2등급	2등급	3등급	2등급
19세 이상 (성인가)	3등급	3등급	4등급	4등급


<방송통신심의위원회 연령별 권장사항>

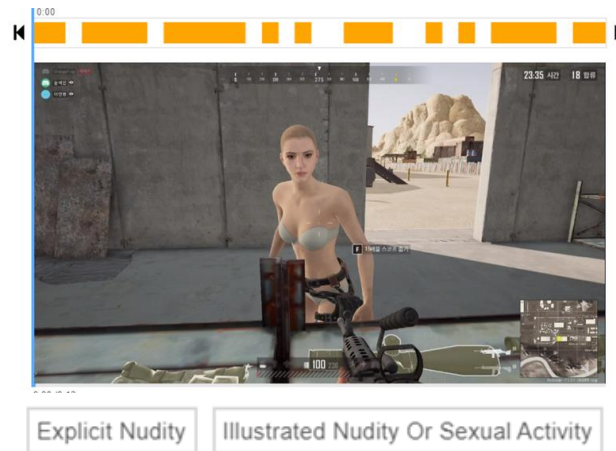
1.3 관련 기술 조사

현재 유해 콘텐츠를 분류하는 기능을 가진 툴은 다음과 같다.

- 1) Amazon ReKognition - Unsafe Content Detect
- 2) Google Cloud PlatForm, Vision API - Unsafe Content Detect

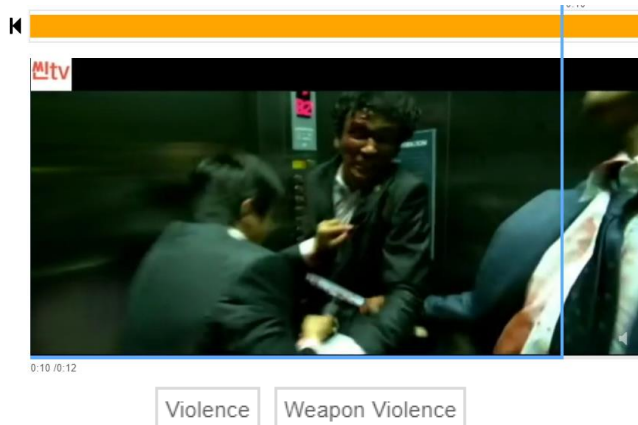
그러나 각 툴의 기능은 19세 연령 기준을 통한 검열과는 확연히 다른 방식이다. 아래 화면과 같이 Amazon의 경우 속옷이나 수영복을 입고 있는 것과 Adult(19세 이상 시청가능한 노출 장면이나 음란한 장면)의 경우를 동일한 라벨로 취급한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

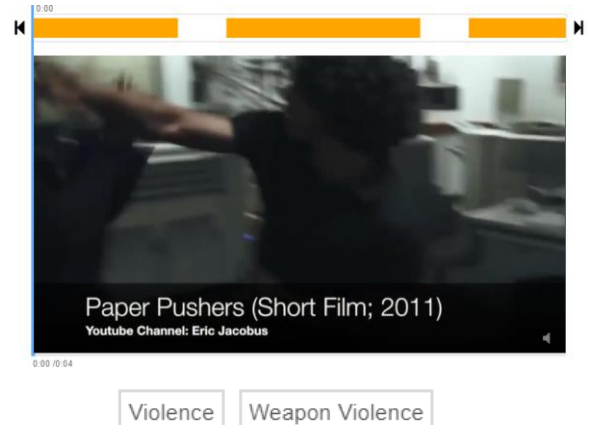


<속옷을 입고있는 장면에서 도출한 라벨>

반면에 Google API는 Adult 장면만 인식하지만, 두 툴 모두 폭력적인 장면의 경우 19세 이상의 기준으로 적용하고 있지 않다. 현재 YouTube 이용 정책과 한국방송통신위원회의 기준에 따르면 잔인한 장면(칼 또는 흉기 등으로 사람을 명확하게 찌르거나 베어 피가 표현되는 장면)으로 19세 관람 기준으로 설정하고 있다. 그러나 아래 화면과 같이 Amazon Rekognition과 Google API 모두 단순 격투 장면(UFC, 15세 이상 관람가능한 격투씬)과 19세 이상 시청가능한 잔인한 장면을 같은 라벨로 취급하기 때문에 분류가 사실상 불가능하다.




<잔인한 장면에서 도출한 라벨>



<15세 격투씬에서 도출한 라벨>

또한 API 특성상 Custom이 불가능에 가깝기 때문에, 각 툴을 이용해 19세 영화 및 영상을 분류하는 것은 불가능하다고 볼 수 있다. 마찬가지로 우리가 검열하는 만 18세 이상 플레이 가능한 게임의 경우에서 등장하는 총을 맞아 피를 흘리는 신체가 등장하는 장면도 검열하지 못하기 때문에, 그 다양성 또한 제한된다고 볼 수 있다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

따라서 YouHi 프로젝트는 앞서 설명했던 선정적이고 음란한 장면과 잔인한 장면(19세 이상 게임 포함) 그리고 흡연과 욕설을 검열할 수 있도록 Scope를 확장하고 정확도를 높여 현재 존재하는 영상 플랫폼에게 더 효과적으로 적용될 수 있게 하였다.

YouHi 프로젝트는 음성 텍스트에서 욕설의 유무를 판단하기 위해 텍스트를 구성하는 하나의 단어를 수치화 하는 단어 임베딩(Word Embedding) 기법 중 하나인 FastText를 이용할 것이다. Word2Vec, FastText, Glove 등 단어 임베딩에는 많은 기법들이 존재하지만 이 중에서도 FastText를 선택한 이유는 다른 기법들과 다르게 텍스트의 최소 단위를 '어휘'가 아닌 '어휘를 구성하고 있는 n-gram'단위로 보기 때문이다. 이러한 방법은 어휘를 구성하는 모든 n-gram 벡터의 평균 벡터를 어휘 임베딩으로 설정할 수 있고 따라서 어휘 안에 포함된 다양한 요소를 n-gram 수준에서 학습하여 동일한 의미를 갖는 어휘가 문법적인 규칙에 따라 변화하는 패턴을 학습하기 용이하다. 또한 동일한 양의 텍스트 데이터에서 더 많은 정보를 활용하기 때문에, 더 적은 양의 학습 데이터로도 높은 성능을 낼 수 있게 하고, 등장 빈도 수가 적은 단어도 참고할 수 있는 경우의 수가 많아지므로 다른 기법들보다도 정확도가 높게 측정된다.

욕설을 검열하기 위해서는 STT로 얻은 텍스트를 FastText 모델의 Test case로 설정해야 하는데 만약 FastText만을 이용한다면 test case의 단위는 문장을 띄어쓰기를 기준으로 나눈 문자열이 될 것이다. 하지만 이러한 방법으로 test를 진행할 경우 여러가지 의미가 합쳐진 단어들이 test case가 되기 때문에 정확한 검열이 어려울 수 있다. 따라서 YouHi 프로젝트는 Test case의 단위를 일정한 의미가 있는 가장 작은 말의 단위인 형태소로 설정하여 의미 판단에 있어 정확도를 높이기로 하였다. 형태소는 kharii라는 형태소 분석기를 이용하여 생성할 것이며 FastTet 모델은 형태소를 입력으로 받아 욕설인지 아닌지 판단하게 된다. 이로 인해 기존 fastText만을 이용하여 텍스트에서 욕설을 검열하는 것보다도 더 정확하고 높은 검열 성능을 보여줄 수 있을 것이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

2 개발 내용 및 결과물

2.1 목표

YouTube 및 실시간 방송 플랫폼, 방송통신위원회 등 여러 기준들을 참고하여 동영상에 미성년자 시청 불가능 장면/욕설이 있는지 판단한다.

동영상에 미성년자 시청 불가능 장면/욕설이 있는지 판단되면 해당 장면이 어떤 가이드라인을 위반했는지 알려준다.

현재 YouTube에 존재하는 다양한 가이드라인 중 선정성, 폭력성, 모방성(흡연, 욕설 등)에 대한 가이드라인을 충족시키는 지 중점적으로 확인하는 검열을 실시한다. 또한 폭력성과 모방성에 대한 사례가 매우 많아 다음과 같이 한정해서 설정했다.

- 1) "칼에 찔리는 장면이 명확하게 보이는 경우"
- 2) "담배를 입에 물고 흡연하는 장면이 명확하게 표출될 경우"
- 3) "19세 미만 이용 불가 게임에서 총을 맞아 피가 튀기는 신체"
- 4) "여성의 상반신이 노출되는 장면"
- 5) "지나치게 상대를 비하하거나 과도한 욕설이 표현될 경우"

또한 현재 YouTube "노란 딱지" 정책 기준의 모호성과 불공정성에 대한 문제를 해결하기 위하여 사용자가 직접 자신의 영상 중 어떤 부분이 부적합한지 확인이 가능하게 하여 크리에이터들의 영상 제작에 있어서 효율성을 높인다.(여기서 "노란 딱지"는 YouTube에 존재하는 기존 AI 영상 검열 장치이다.)

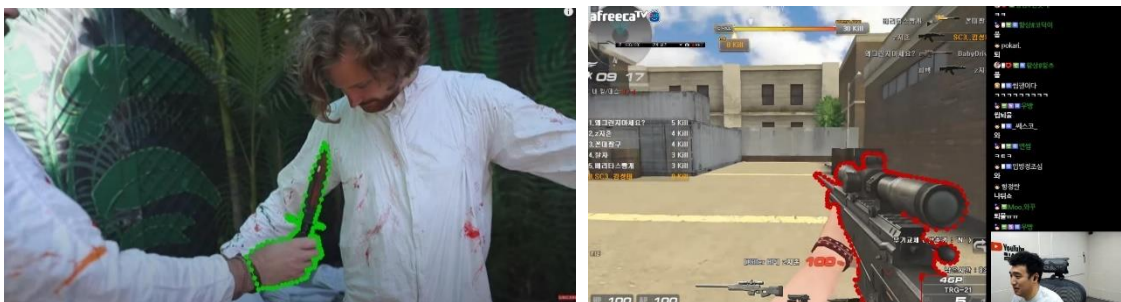
우리 프로젝트의 최종 목표는 다양한 영상 플랫폼에서 이 시스템을 사용할 만큼 정확도를 높이고, 많은 시간이 소요되지 않게 하는 것이다. 또한 이 시스템을 이용해 연령 제한 기준을 충족시키지 못하는 많은 영상들 중 90%를 줄이는 것이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

2.2 연구/개발 내용 및 결과물

2.2.1 영상 1차 검열 - Instance Segmentation

Knife, Smoke, Gun, Adult를 각각 Labelme 툴을 이용해 Annotation을 제작하여 데이터셋을 구축하였다. 각 라벨 당 70장의 이미지가 사용되었으며, Annotation은 이미지의 이름과 동일한 이름을 가지는 json 파일로 생성된다.




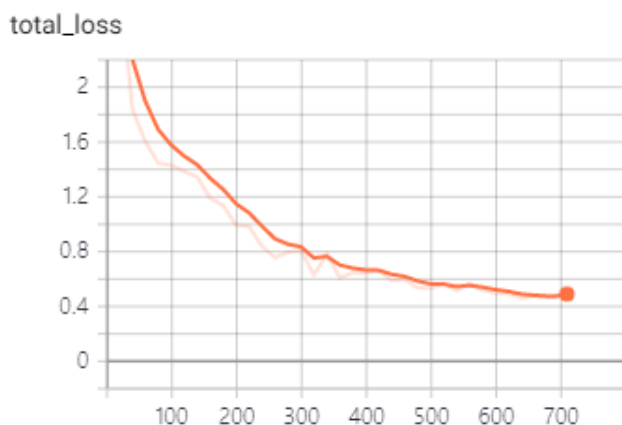
<Knife, Gun, Smoke Annotation>

칼 같은 경우 담배와 비슷하게 인식하여 오분류되는 결과가 많이 발생하여 손을 포함시켜서 하나의 Segmentation으로 구성하였다.

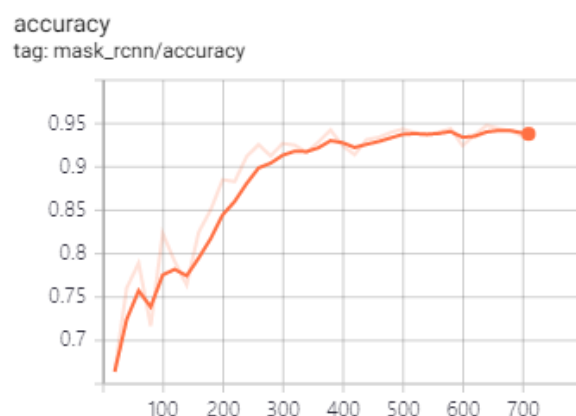
모델은 Mask_RCNN_FPN_101_3x 모델을 사용하였고, Learning Rate는 0.0025, Iteration은 710으로 설정하여 학습을 진행하였다. 예측 값이 최소 0.5이상(threshold) 결과만 출력시키며 가장 높은 값을 보이는 것을 해당 이미지에서의 라벨로 선정하였다.

아래 화면은 설정한 Hyper Parameter로 학습을 진행한 화면이다. Loss 값과 Prediction의 Accuracy 값, Iteration에 따른 학습 결과를 출력시켰다.

 <div> 국민대학교 컴퓨터공학부 캡스톤 디자인 I </div>	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08



<Loss 값>



<정확도>

```

[05/21 19:45:26 d2.utils.events]: eta: 0:04:14 iter: 379 total_loss: 0.645
[05/21 19:45:42 d2.utils.events]: eta: 0:03:59 iter: 399 total_loss: 0.647
[05/21 19:45:57 d2.utils.events]: eta: 0:03:44 iter: 419 total_loss: 0.660
[05/21 19:46:13 d2.utils.events]: eta: 0:03:29 iter: 439 total_loss: 0.589
[05/21 19:46:29 d2.utils.events]: eta: 0:03:14 iter: 459 total_loss: 0.595
[05/21 19:46:44 d2.utils.events]: eta: 0:02:59 iter: 479 total_loss: 0.535
[05/21 19:47:00 d2.utils.events]: eta: 0:02:43 iter: 499 total_loss: 0.533
[05/21 19:47:16 d2.utils.events]: eta: 0:02:28 iter: 519 total_loss: 0.562
[05/21 19:47:31 d2.utils.events]: eta: 0:02:12 iter: 539 total_loss: 0.516
[05/21 19:47:47 d2.utils.events]: eta: 0:01:57 iter: 559 total_loss: 0.568
[05/21 19:48:02 d2.utils.events]: eta: 0:01:41 iter: 579 total_loss: 0.520
[05/21 19:48:18 d2.utils.events]: eta: 0:01:26 iter: 599 total_loss: 0.495
[05/21 19:48:33 d2.utils.events]: eta: 0:01:10 iter: 619 total_loss: 0.490
[05/21 19:48:49 d2.utils.events]: eta: 0:00:55 iter: 639 total_loss: 0.460
[05/21 19:49:05 d2.utils.events]: eta: 0:00:39 iter: 659 total_loss: 0.471
[05/21 19:49:20 d2.utils.events]: eta: 0:00:24 iter: 679 total_loss: 0.466
[05/21 19:49:36 d2.utils.events]: eta: 0:00:08 iter: 699 total_loss: 0.479
[05/21 19:49:45 d2.utils.events]: eta: 0:00:00 iter: 709 total_loss: 0.510


```

<학습 진행도>

학습이 끝나고 테스트 데이터셋으로 예측을 진행한 결과이다.



<Knife 예측 결과>

	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08



<Smoke 예측 결과>




<Gun 예측 결과>

Adult 라벨의 경우 2차 검열 과정인 Image Classification을 거치기 때문에, 인식된 객체만 큼 ROI를 설정해 이미지를 추출하여 저장한다.

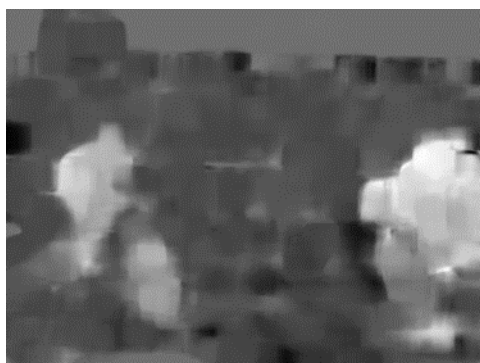
2.2.2 영상 2차 검열 - Video Classification

영상 2차 검열은 라벨별로 2가지의 알고리즘으로 작동한다. 첫번째로는 Video Classification 이다. Smoke 라벨과 Knife 라벨에 대한 영상 검열이 해당된다. Video Classification은 Two Stream Convolution Network 모델로 진행한다. Two Stream Convolution Network 모델은 Spatial Stream ConvNet과 Temporal Stream ConvNet으로 각각 예측된 결과를 Average, Conv Fusion을 통해 결합한다. 먼저 Spatial Stream ConvNet는, 하나의 동영상 프레임에서 작동되며 해당 프레임에서 사람의 Action을 인식한다. 즉, 정적 이미지에서 결과를 예측하는 것이다. 반면 Temporal Stream ConvNet은 일정 길이의 이미지(프레임)를 취합한 내용을 통해 예측하는 것이다. Optical Flow ConvNet이라고도 불리는데, 그 이유는 RGB 이미지가 아닌 Optical Flow로 표현된 이미지로 학습과 예측을 진행하기 때문이다. Optical Flow는 Vertical Flow와 Horizontal Flow로 구성되어있다. 따라서 학습과 예측을 진행하기 전에 RGB이미지를 Vertical Flow, Horizontal Flow로 표현해야 한다. Vertical Flow, Horizontal Flow는 다음과 같이 표현된다.

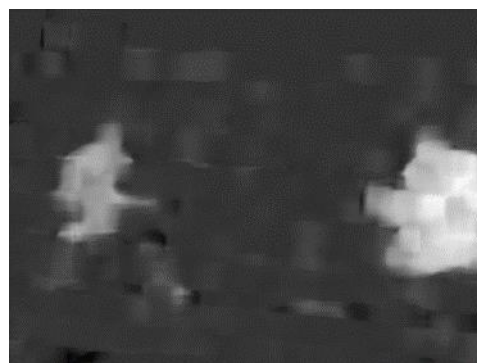
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08



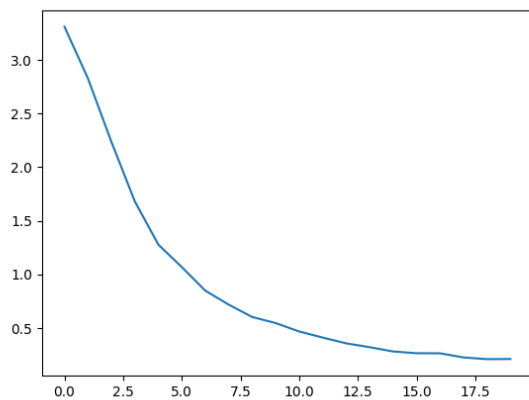
<Original Image>



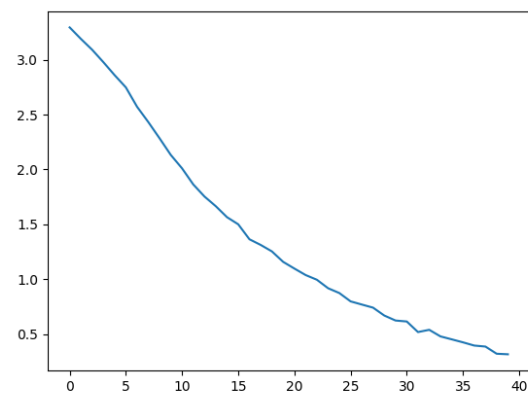
<Horizontal>




<Vertical>



<Spatial Stream Model 학습>

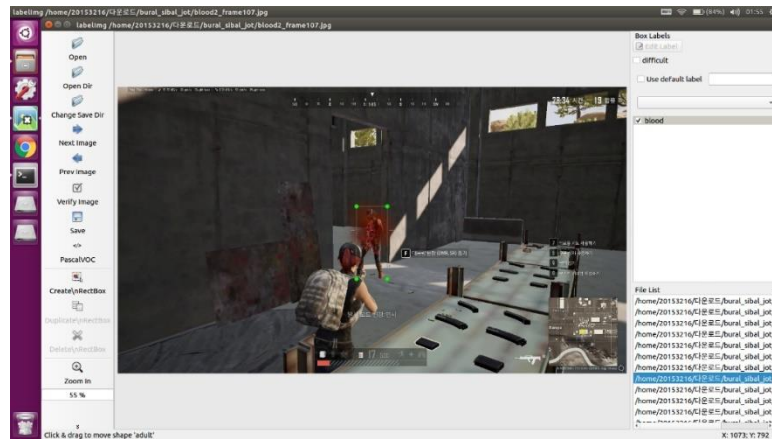


<Temporal Steam Model 학습>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

2.2.3 영상 2차 검열 - Object Detection, Image Classification

영상 2차 검열의 두번째 알고리즘인 Image Classification 과정이다. Blood 라벨과 Adult 라벨에 대해 진행하는데, Blood 라벨은 Image Classification을 진행하기 전에 Object Detection 을 추가적으로 진행한다. Object Detection은 만 18세 이용 불가 게임에서 등장하는 총을 맞고 피를 흘리는 신체를 Detect 하기위해 진행한다. Object Detect는 중간 계획과 동일하게 Faster-RCNN으로 Detect하며 해당 라벨의 데이터셋 구축은 Labellmg 툴을 이용해 구축했다.




<Labellmg를 이용한 데이터셋 구축>

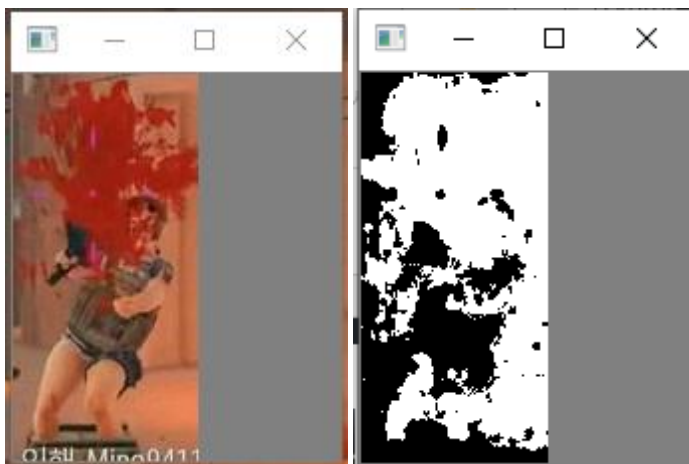


<오분류>

<정분류>

그러나 가장 우측 사진처럼 올바르게 분류하는 경우도 있지만, 좌측 2개의 사진처럼 일반 캐릭터를 Detect하는 오분류가 발생하기때문에 추가적인 Image Classification 진행을 위해 ROI로 캐릭터만 추출하고, HSV 필터를 적용해 빨간색 픽셀이 임계값 이상 있는 이미지들만 고려한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08



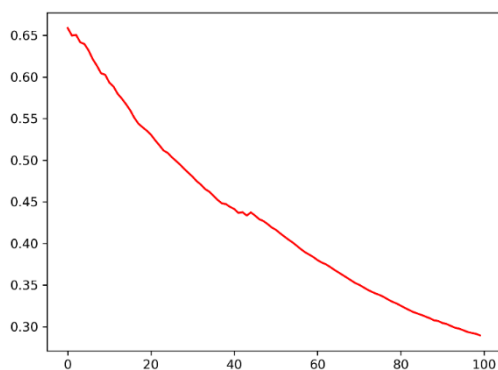
<ROI 추출 및 HSV 필터>

이후 잘라진 ROI 사진으로 Image Classification을 진행한다.

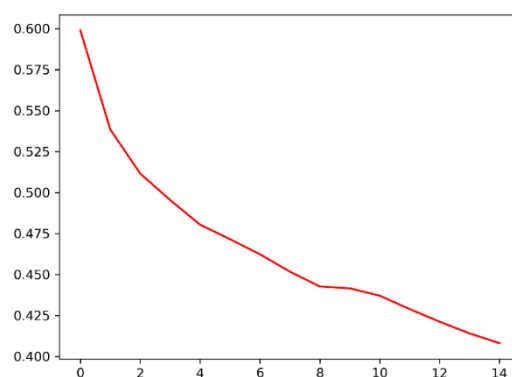
Image Classification은 Blood 라벨과 Adult 라벨에서 대해서 진행되는데, 앞서 추출된 ROI 이미지를 가지고 혈흔과 빨간 색의 옷, 나체와 살색의 옷의 구분이 가능하도록 했다. Adult와 Blood의 경우 동일한 모델을 사용했지만, Epoch와 Learning Rate, Batch Size를 다르게 설정하여 보다 정확도를 높였다.

Adult의 경우 100번의 Epoch 중 88번째 Epoch가 가장 뛰어난 성능을 보였으며, Batch Size는 8, Learning Rate는 0.0005로 설정하였다.


Blood의 경우 15번의 Epoch 중 10번째 Epoch가 가장 뛰어난 성능을 보였으며, Batch Size는 10, Learning Rate는 0.001로 설정하였다.



<Adult 라벨 학습>

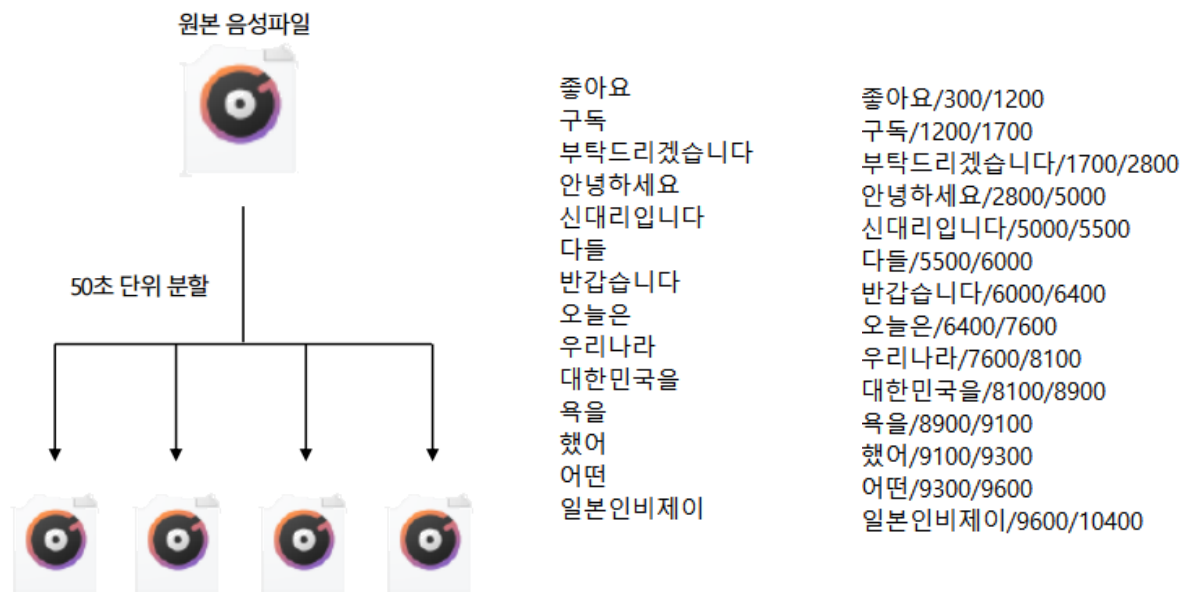


<Blood 라벨 학습>

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

2.2.4 음성 인식

음성 인식은 Google Speech To Text API를 사용하여 음성에서 텍스트를 추출한다. 그러나 Google Speech To Text API에서 길이가 1분 이상인 음성은 유료 서비스로 제공하고 있기 때문에, 음성파일을 50초 단위로 분할하여 API를 호출한다.



STT(Speech To Text) 는 잡음처리와 특징 추출과정을 거친 음성 데이터가 언어모델과 음향 모델이 결합된 디코더를 통과한 후 문장 형태로 출력되면 이를 텍스트로 저장한다.

그리고 음성 마스킹 기능을 위해 발화 시간을 추출하는 API의 기능을 사용해 발화 시간이 담긴 텍스트 파일을 저장한다.

2.2.5 자막 추출

자막 추출은 영상을 프레임 단위로 나눈 사진에 Google Vision API를 적용시켜 이루어진다. 해당 API를 통해 자막을 텍스트로 변환한 후 자연어 부분과 이미지 내의 해당 단어 좌표를 구분 지어 저장한다.

	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

오늘은 축구 역사상 기1장 화려한 플레이를 보여준 선수, +(68,228) (274,228) (91,230) (91,244) (84,244)/축구+(95,229) (109,229) (109,244) (95,244)/역사 (143,243)/1+(148,230) (149,230) (149,243) (148,243)/장+(151,229) (158,229) (192,230) (216,230) (216,244) (192,244)/를+(219,230) (225,230) (225,244) (: (271,228) (271,247) (255,247)/,+ (273,243) (274,243) (274,245) (273,245)/

오늘은 축구 역사상 1장 화려한 플레이를 보여준 선수, +(68,226) (274,226) (2 (91,244) (84,244)/축구+(95,229) (109,229) (109,244) (95,244)/역사상+(114,2 (148,230) (158,230) (158,244) (148,244)/화려한+(163,229) (187,229) (187,24 (225,230) (225,244) (219,244)/보여준+(230,230) (253,230) (253,244) (230,24 (273,246)/

<자막의 단어 별 좌표를 포함하고 있는 파일>

또한 프레임에서 자막을 추출하다 보니 동일한 자막이 굉장히 많이 나온다. 따라서 동일한 자막을 모두 좌표를 추출하여 텍스트 파일에 적으면 Overhead가 굉장히 커지기 때문에 Cosine 유사도를 통해 문장의 유사도를 확인하여 비슷한 문장이면 고려하지 않는 알고리즘을 구현했다.

```
def get_cosine(vec1, vec2):
    intersection = set(vec1.keys()) & set(vec2.keys())
    numerator = sum([vec1[x] * vec2[x] for x in intersection])

    sum1 = sum([vec1[x] ** 2 for x in list(vec1.keys())])
    sum2 = sum([vec2[x] ** 2 for x in list(vec2.keys())])
    denominator = math.sqrt(sum1) * math.sqrt(sum2)

    if not denominator:
        return 0.0
    else:
        return float(numerator) / denominator

text1 = "오늘은 축구 역사상 기1장 화려한 플레이를 보여준 선수,"
text2 = "오늘은 축구 역사상 1장 화려한 플레이를 보여준 선수,"


vector1 = text_to_vector(text1)
vector2 = text_to_vector(text2)

cosine = get_cosine(vector1, vector2)

print("Cosine:", cosine)
Cosine: 0.7499999999999999
```

2.2.6 형태소 분석 및 욕설 검열

저장된 텍스트파일은 khaiii api 를 이용해 모든 문장을 형태소 단위로 분리한다. khaiii 형태소 분석기를 빌드 및 설치한 후 Python 바인딩을 하여 Python 인터프리터에서 사용할 수 있도록 한다. 그 후 형태소 분석을 위해 khaiii 형태소 분석기를 이용하는 Python 파일을 실행한다. 인자는 Speech To Text로 추출된 텍스트가 저장되어 있는 텍스트 파일을 주도록 한다. 실행 결과는 다음과 같다.

 <div> <p>국민대학교</p> <p>컴퓨터공학부</p> <p>캡스톤 디자인 I</p> </div>	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

```

kmsuw@kmsuw-ThinkPad-T440: ~/바탕화면/fastText/mywork/text_file
kmsuw@kmsuw-ThinkPad-T440:~/바탕화면/fastText/mywork/text_file$ cat test_tokenized khaiii.txt
안녕/NNG 하/XSA 세요/EC 오늘/NNG 은/JX 암기법/NNG 영상/NNG 을/JKO 들고오/VV 앓/EP 어요/EC
여러분/NP 들/XSN 께서/JKS 정말/MAG 정말/MAG 많이/MAG 요청/NNG 하/XSV 여/EC 주/VX 시/EP 어서/EC
이렇/VA 게/EC 찍/VV 어/EC 보/VX 게/EC 되/VV 앓/EP 습니다/EC 사실/MAG 저/NP 는/JX
암/VV 기/NNG 는/JX 시간/NNG 투자/NNG 와/JC 반복/NNG 이/VCP 라고/EC 생각/NNG 을/JKO 하/VV 여요/
EC 근데/MAJ 이것/NP 이/JKS 마치/MAG 다이어트/NNG 는/JX
식이/NNG 요법/NNG 과/JC 운동/NNG 이/VCP 다/EC 이런/MM 말/NNG 이/JX 랑/JKB 똑같/VA 지/EC 앓/VX
아요/EC 왜냐면/MAG
kmsuw@kmsuw-ThinkPad-T440:~/바탕화면/fastText/mywork/text_file$

```

출력 결과는 분석된 형태소와 해당 형태소에 대한 품사를 나타내고 띄어쓰기로 구분 되어 있다. 줄 바꿈은 인자로 받았던 텍스트 파일과 동일하게 적용된다. 위 출력결과에 있는 형태소들은 '/' 와 자신의 품사를 띄어쓰기 구분없이 붙여서 나타내기 때문에, 한글 데이터로만 학습된 FastText 모델에서는 테스트할 수 없는 데이터이다. 따라서 형태소 뒤에 붙은 '/품사'를 없애 주는 데이터 전처리 과정을 거쳐야 한다. 데이터 전처리 과정은 다음과 같이 실시한다. split 함수를 이용하여 하나의 line에 존재하는 형태소들을 각 하나의 원소로 갖는 List를 얻는다. 그리고 find 함수를 사용하여 형태소에서 '/'가 존재하는 위치를 파악한 후 그 위치의 앞부분들을 List의 원소로 저장한다. 각 line마다 이러한 전처리를 진행하여 한글로만 이루어진 형태소 데이터를 원소로 갖는 List를 출력하면 다음과 같다.


```

kmsuw@kmsuw-ThinkPad-T440: ~/바탕화면/fastText/mywork
kmsuw@kmsuw-ThinkPad-T440:~/바탕화면/fastText/mywork$ python3 voice_filter.py --input ./text_file/test_tokenized khaiii.txt
['안녕', '하', '세요', '오늘', '은', '암기법', '영상', '을', '들고오', '앓', '어요']
['여러분', '들', '께서', '정말', '정말', '많이', '요청', '하', '여', '주', '시', '어서', '이렇',
'게', '찍', '어', '보', '게', '되', '앓', '습니다', '사실', '저', '는']
['암', '기', '는', '시간', '투자', '와', '반복', '이', '라고', '생각', '을', '하', '여요', '근',
'데', '이것', '이', '마치', '다이어트', '는']
['식이', '요법', '과', '운동', '이', '다', '이런', '말', '이', '랑', '똑같', '지', '앓', '아요',
', '왜냐면']
kmsuw@kmsuw-ThinkPad-T440:~/바탕화면/fastText/mywork$

```

FastText 모델을 학습시키기 위한 텍스트 데이터셋은 한국어 위키디피아 텍스트와 웹 크롤링으로 수집한 욕설 텍스트로 이루어져 있다. 기존 데이터셋은 한글을 제외한 문자들이 다수 존재하기 때문에 한글을 제외한 문자들을 텍스트로부터 제외하는 전처리 과정을 진행한다. 그 결과로 생성되는 한글로만 이루어진 텍스트 데이터셋은 FastText skip-gram 모델의 학습 데이터로 이용된다. 데이터 전처리는 Python에서 제공하는 re 모듈을 이용하여 진행하였다.

학습이 완료된 FastText 모델은 분석된 형태소와 욕설 간의 코사인 유사도를 확인하는데

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

사용된다. 음성 검열에 있어 이 과정이 적합한지 판단하기 위해 3가지 경우에 대해 각각 코사인 유사도의 정확도를 판단하였는데, 첫 번째 경우는 욕설과 욕설이 아닌 형태소, 두 번째는 욕설과 욕설, 마지막 세 번째 경우는 욕설과 욕설과 비슷하지만 욕설이 아닌 형태소이다. 각 경우에 대해 계산한 코사인 유사도는 다음과 같다.


```
kmusw@kmusw-ThinkPad-T440:~/바탕화면/fastText/mywork$ python3 cs_sim.py
'안녕' | '시발' : 0.42002043
'병' | '시발' : 0.73076606
'시발점' | '시발' : 0.30771598
```

첫 번째 경우를 살펴보면, 욕설과 욕설이 아닌 형태소의 코사인 유사도는 0.420으로 0.5 이하인 값을 보였다. 그에 비해 욕설과 욕설의 코사인 유사도를 보면 0.730으로 매우 높은 값을 갖는데, 이는 다른 자음과 모음으로 이루어져 있다 하더라도 학습 데이터에 따라 욕설로 판단되었기 때문에 이와 같은 결과를 얻게 된 것이다. 마지막으로 세 번째 경우 '시발점'이라는 단어는 '시발'이라는 욕설을 포함하고 있으나 첫 출발을 하는 지점이라는 의미를 갖고 있기 때문에 유사도는 0.307이라는 낮은 값을 갖게 되었다.

위의 결과에 따라 FastText 모델을 이용한 코사인 유사도 측정은 음성 검열에 있어 충분히 적합하다는 것을 알 수 있다.

2.2.7 Front End

Upload



동영상을 '여기' 끌어다 놓거나
'여기'를 클릭하세요.

도움말 및 제안사항

- 필터링에 소요되는 시간은 약 10분이며, 사용자의 인에 따라 달라질 수 있습니다.
- 재생 시간이 1분 미만인 영상만 업로드 가능합니다.
- 업로드 파일 형식은 mp4로 제한합니다.
- 업로드를 완료한 후, Clear 버튼을 통해 다른 영상을 업로드할 수 있습니다.

업로드

검열

.mp4 영상만이 허용됩니다.


영상 재생시간이 1분을 초과하였습니다.

하나의 영상만 업로드 가능합니다.

검열을 완료하였습니다.

동영상을 업로드하기 위해 동영상을 등록하고 업로드 하는 공간이다. 파일을 정상적으로 등록하고, 업로드 버튼이 활성화되고 업로드 버튼을 눌러 업로드가 완료되면 검열 버튼이 활성화되어 검열을 진행할 수 있다.


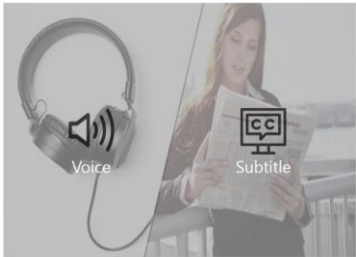
파일을 정상적으로 등록하지 못할 때, 등록되었을 때, 검열이 완료되었을 때 등 모든 이벤

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

트에 따라 알림창을 띄워 사용자에게 알려준다.

검열이 완료되면 Video / Voice & Subtitle 버튼을 클릭해 결과를 확인해볼 수 있다. 결과는 모달(Modal)창을 띄워 보여주며, 영상 검열의 경우 라벨 박스를 클릭하여 쉽게 결과를 확인할 수 있고, 음성 검열의 경우 욕설이 탐지된 구간과 빼-처리가 완료된 음성파일을 들을 수 있다. 자막 검열의 경우 해당 자막에 바운딩 박스가 그려져서 출력된다.

Filter

영상 검열 결과


4 전체 결과

0 Adult

0 Blood

0 Knife

4 Smoke



Adult

Blood

Knife

Smoke


0:01 ~ 0:04

0:04 ~ 0:07

0:07 ~ 0:10

확인

음성, 자막 검열 결과




음성에서 욕설이 검열되지 않음

0:00 / 0:00

- 검열된 음성은 볼륨이 0으로 설정되어 들을 수 없습니다.
- 원본 오디오를 볼륨 100%로 설정하여 다시 확인하십시오.

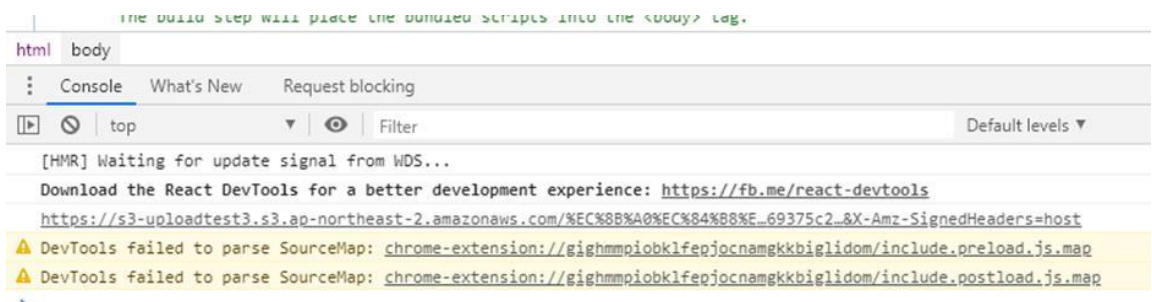
확인

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

2.2.8 Back End

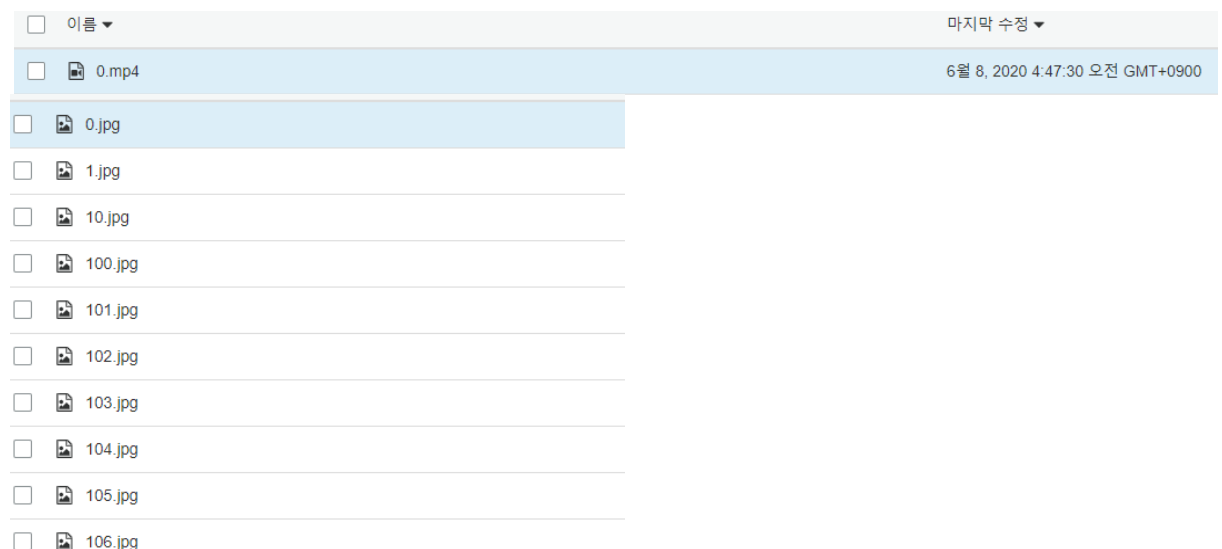
딥러닝 모델 학습과 모델의 영상, 음성 검열 과정에 필요한 Amazon EC2 instance, Amazon S3, Amazon Lambda, Amazon Gateway API를 생성한다. 웹 페이지의 배포를 위해 웹서버를 구축한다. 웹서버, 웹페이지, AWS의 원활한 상호작용을 위해 Socket 통신을 활용한다.

동영상이 업로드 되는 공간, 업로드 된 동영상의 프레임 추출, 딥러닝 모델에 넣는 작업을 위한 Amazon S3, Amazon Lambda, Amazon Gateway API를 이용한다. 단순 S3에 업로드하는 방식을 채택할 경우, IAM 사용자의 Access Key 및 Secret Key가 노출될 가능성이 높으므로 Gateway Api와 Lambda를 통해 업로드를 진행한다. S3에 업로드가 완료되면 Lambda 함수에서는 프레임을 추출하고, 전처리를 진행하는 코드를 실행한다.



<Pre-Signed Url>

여러 사용자들이 동시에 업로드하는 환경을 고려해 소켓 통신으로 고유번호를 할당한다. 고유번호를 할당 받은 파일 이름으로 S3에 업로드하고 프레임을 추출한다.



2.2.9 정확도 측정

본 프로젝트는 유해한 내용을 유해하다고 판단하는 것과 유해하지 않은 내용을 유해하지 않다고 판단해야 하기 때문에, 기술적 완성도와 시스템의 필요성을 평가 받으려면 정확도가 가장 중요하다. Precision / Recall 매트릭스를 통해 정확도를 측정하고 개선하는 방향으로 진행한다. 다음과 같이 Precision / Recall 매트릭스를 구성한다.

- True Positive(TP) : 실제 True인 정답을 True라고 예측 (정답)
- False Positive(FP) : 실제 False인 정답을 True라고 예측 (오답)
- False Negative(FN) : 실제 True인 정답을 False라고 예측 (오답)
- True Negative(TN) : 실제 False인 정답을 False라고 예측 (정답)

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

$$(Precision) = \frac{TP}{TP + FP} \quad (Recall) = \frac{TP}{TP + FN}$$

- **Precision(정밀도)** : 모델이 True라고 분류한 것 중에서 실제 True인 것의 비율

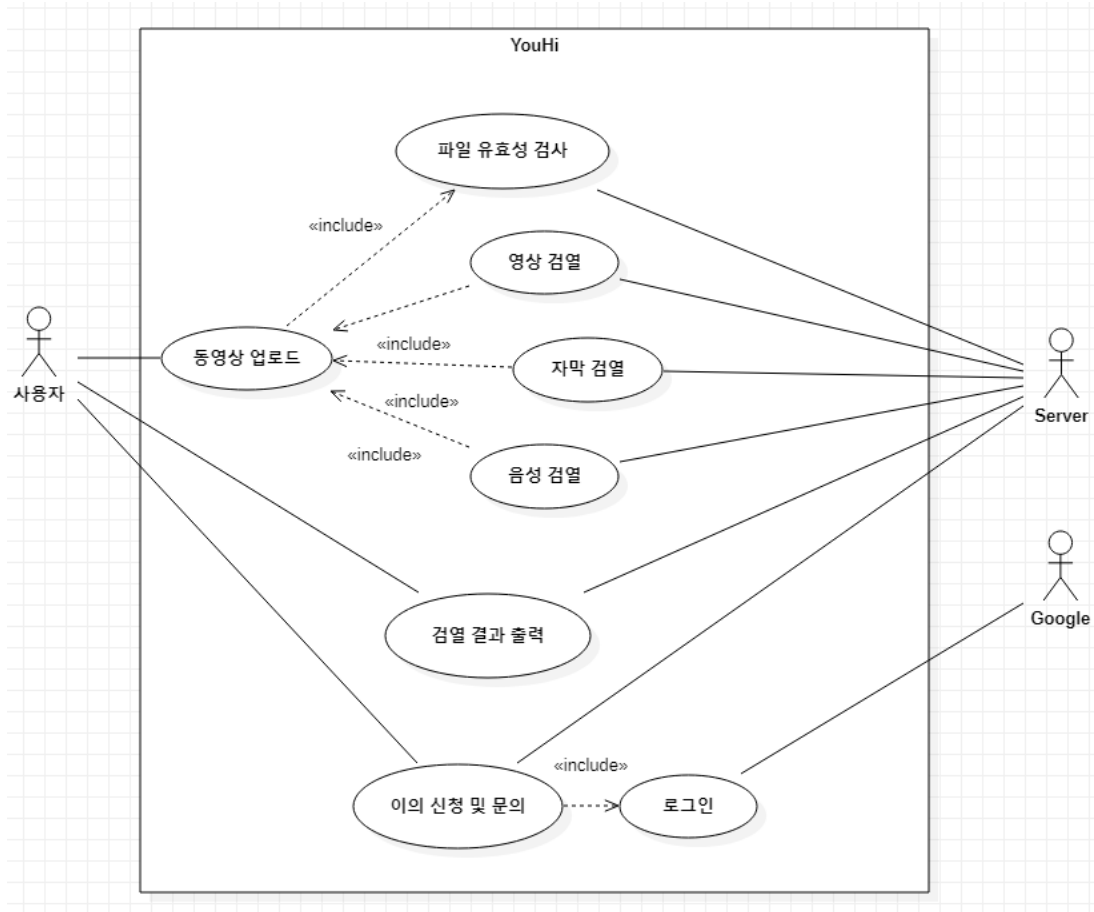
$$(Precision) = \frac{14}{14 + 2} = 87.5\%$$

- **Recall(재현율)** : 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율


$$(Recall) = \frac{14}{14 + 6} = 70\%$$

2.3 시스템 요구사항

2.3.1 시스템 기능 요구사항



- 웹페이지는 동영상을 서버에 업로드한다. (완료)
- 웹페이지는 영상 및 음성에 대한 검열 결과를 출력한다. (완료)
- 웹페이지에서 사용자의 의견을 수신할 수 있게 사용자가 구글 계정으로 로그인한다.(완료)
- 웹페이지는 로그인 된 사용자의 의견을 수신한다. (완료)
- 서버는 업로드 된 동영상의 확장자를 확인하고 저장한다. (완료)
- 서버는 업로드 된 동영상에서 음성 및 프레임을 추출한다. (완료)
- 서버는 프레임에서 자막을 추출한다. (완료)
- 서버는 음성에서 텍스트 파일을 추출한다. (완료)
- 서버는 추출된 음성 및 프레임을 딥러닝 모델에 넣어 그 결과를 저장한다. (완료)
- 서버는 로그인 된 사용자의 의견을 송신한다. (완료)

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

2.3.2 시스템 비기능(품질) 요구사항

사용성 : 사용자가 한 눈에 사용방법을 알 수 있도록 UI 를 제작한다 **(완료)**

검열 결과를 한 눈에 알아볼 수 있도록 가시화한다 **(완료)**

신뢰성 : 업로더들이 검열 결과를 납득할 수 있게 정확하게 검열을 한다.

기능성 : 사용자들의 편의성을 위해 다양한 파일확장자의 업로드를 가능하게 설계한다.**(완료)**

효율성 : 검열에 소요되는 시간은 최대 10 분이 넘어가지 않도록 설계한다. **(미완료)**

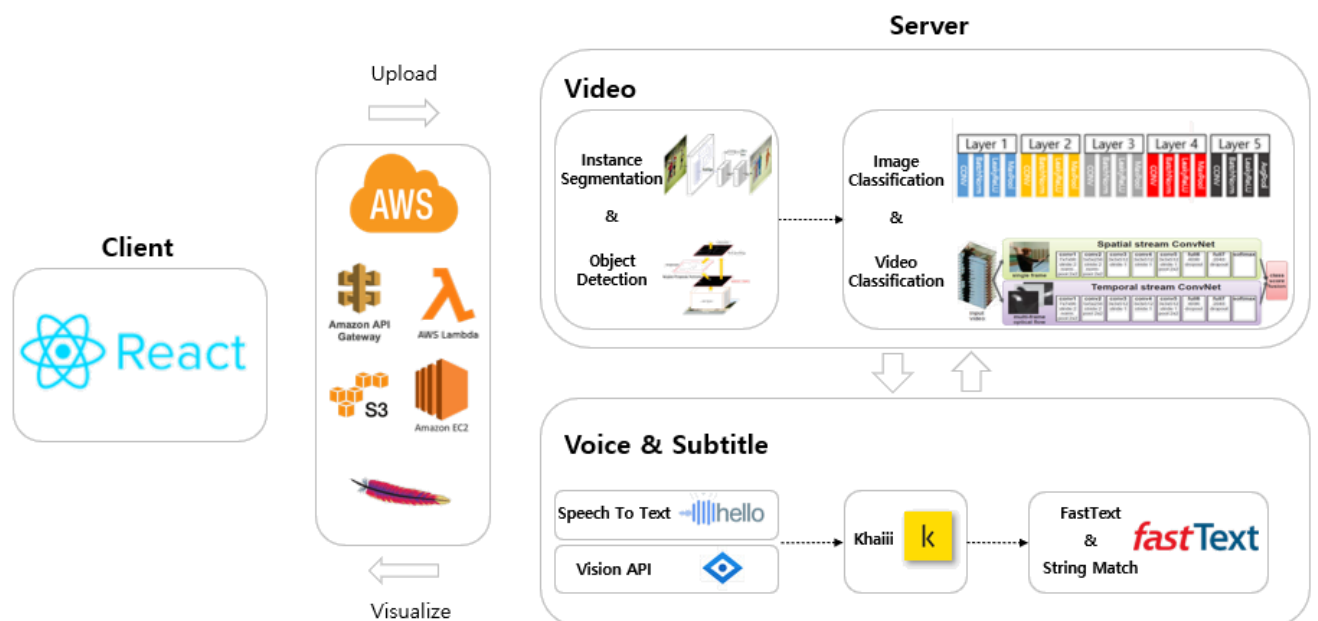
이식성 : 동영상 업로드 기능 및 검열 결과 출력을 웹페이지에서 구현함으로써 다양한 플랫폼(스마트폰, 태블릿 PC 등)에서 사용가능 하게 한다.**(완료)**


서버를 AWS 를 이용해서 구현함으로써 다른 컴퓨터나 환경에서도 쉽게 구축할 수 있게 한다. **(완료)**

보안성 : 보안성을 위해 검열이 끝난 데이터는 즉시 클라우드에서 삭제한다. **(완료)**

서버기능의 보안성을 위해 mp4,avi 형태의 파일만 업로드 가능하도록 제한한다. **(완료)**

2.4 시스템 구조 및 설계도



 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

영상을 사용자가 업로드할 때, 업로드 되는 공간, 업로드 이후 처리되는 일은 모두 Amazon Web Service 를 사용해서 진행된다. Amazon Web Service 를 포함한 시스템 구조에 대한 설명은 다음과 같다.

- 1 사용자가 영상을 업로드하기 위해 등록하면, Apache 웹서버에서 구동되고 있는 소켓 서버에서 고유 번호를 부여한다.
- 2 등록된 고유 번호로 파일명을 변경하고, S3 에 업로드하기 위해 Pre-Signed Url 을 Gateway API 를 호출함으로써 가져온다.
- 3 얻은 Pre-Signed Url 을 통해 S3 버킷에 업로드하고, 업로드 완료되는 이벤트에 따라 Lambda 함수가 동작되며, 프레임을 추출한다.
- 4 프레임 추출이 완료되면, 영상과 추출된 프레임을 서버에서 다운로드 받는다.
- 5 영상 검열과 음성 검열을 진행한다.
- 6 영상 검열과 음성 검열이 모두 완료된 이후 웹브라우저로 시각화한다.

2.5 활용/개발된 기술


2.5.1 영상 검열

Instance Segmentation

Instance Segmentation(객체 분할)이란 경계 상자(Bounding Box) 안에 존재하는 실제 고유 객체들이 갖는 영역만을 탐지하는 기술이다. 주어진 이미지 내 각 위치 상의 픽셀들을 하나씩 조사하며, 현재 조사 대상인 픽셀이 어느 특정한 라벨에 해당하는 사물의 일부인 경우 해당 픽셀의 위치에 그 객체를 나타내는 '값'을 표기하는 방식으로 예측 결과물을 생성하게 되고, 반대로 조사 대상 픽셀이 어느 클래스에도 해당하지 않는다면 어떠한 값도 부여하지 않게 된다.

Instance Segmentation에 사용되는 모델은 R-CNN 계열에 속하는 Mask R-CNN이다.

프로젝트에서 Instance Segmentation 기술을 활용한 이유는 오탐률, 즉 유해하지 않은 것을 유해하다고 판단하는 확률을 줄이기 위해서이다. Instance Segmentation을 활용하지 않았을 때는 Video Classification 단계에 매우 다양한 영상들이 Test Case가 되었고 그만큼

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

오탐률은 높게 측정되었다. 하지만 Instance Segmentation을 활용한 후로 칼, 담배, 총, 여성의 가슴이 객체로써 탐지된 프레임에 갖는 영상만 Video Classification 단계의 Test Case가 되도록 하였기 때문에, 그 결과 오탐률은 매우 감소했고 프로젝트의 Precision은 증가하게 되었다.

Object Detection

Object Detection(객체 탐지)는 이미지에서 찾고 싶은 관심 객체를 배경과 구분해 식별하는, 딥러닝을 통한 자동화 기법이다. 우리는 Deep Learning을 이용한 Object Detection을 진행한다. 여기에는 1-Stage Detector와 2-Stage Detector로 나눌 수 있는데, 1-Stage Detector는 YOLO, SSD 계열이고, 2-Stage Detector는 R-CNN 계열이다. 우리 프로젝트에서는 2-Stage Detector에 해당되는 Faster R-CNN을 사용하는데, 이는 속도보다는 정확도를 고려하였을 때 Faster R-CNN이 더 높기 때문이다.

Video Classification

Video Classification은 Two Stream Convolution Network를 사용했다.

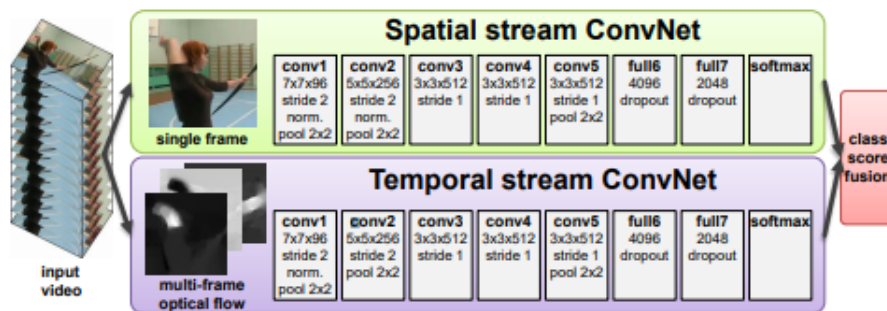



Figure 1: Two-stream architecture for video classification.

구조는 위와 같으며 2.2.3과 설명이 동일하다.

Image Classification

Image Classification에 사용되는 CNN모델의 구조이다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

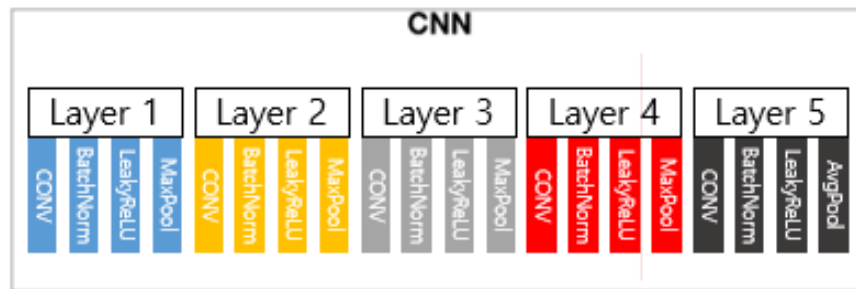



Image Classification이란 딥러닝을 통해 시각적 내용에 따라 이미지를 분류하는 것을 말한다. 딥러닝 신경망의 한 종류인 CNN(Convolutional Neural Network)을 이용하며 이미지를 입력으로 받고 그에 대한 class 값과 입력 이미지가 특정 class라는 것에 대한 확률 값을 출력한다. CNN은 크게 input layer, output layer, hidden layers로 나눌 수 있고, hidden layers는 보통 convolution 연산을 입력 이미지에 적용 후 다음 층으로 전달하는 Convolutional layers, CNN의 출력을 결정하는 ReLU layers, 뉴런 군집의 출력을 단일 뉴런으로 결합하여 데이터의 차수를 감소시키는 Pooling layers, 이전 층의 모든 노드를 다음 층의 모든 노드에 연결하는 Fully connected layer으로 구성된다.

2.5.2 음성 검열

Google Speech API

딥러닝 기반의 음성인식 기술은 중요한 지식원(Knowledge Source)인 단어 시퀀스에 확률을 할당(assign)하는 일을 하는 언어모델, 그리고 언어의 소리단위를 딥러닝을 통해 학습하여 어떤 단어가 어떤 소리로 나는지를 확률적으로 변환하는 음향모델을 사용해 음성 신호로부터 문자 정보를 출력한다. 그리고 높은 확률을 보인 단어를 출력하게 된다. Google Speech API는 음성 데이터로부터 잡음 처리와 특징 추출과정을 거친 후, 언어모델과 음향모델이 결합된 디코더를 통과한 후 결과를 문장 형태로 출력하게 된다.

영상으로부터 추출된 음성을 검열하기 위해서는 음성을 텍스트 파일로 나타낼 수 있어야 했고, 우리 팀은 테스트한 Speech API 중 가장 높은 정확도를 보인 Google Speech API를 활용하여 이를 구현했다. Google Speech API가 제공하는 기능 중 출력한 문장의 단어마다 발화 시작, 종료 시간을 제공하는 기능이 있었기 때문에 원본 음성 파일에 욕설이 발견된 부분을 묵음 처리하기에 매우 용이했다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

Khaiii


Khaiii는 카카오에서 개발한 세 번째 형태소 분석기로써, 텍스트 형태의 데이터를 형태소 단위로 분리한다. 기계학습 기반의 알고리즘을 이용하여 이를 구현하게 되는데, 형태소 분석은 자연어 처리를 위한 가장 기본적인 전처리 과정이기 때문에 빠른 시간 내에 이루어져야 하므로 신경망 알고리즘들 중 빠른 속도로 진행되는 Convolutional Neural Network(CNN)을 사용한다.

음성 검열을 구현하기 위해서는 음성으로부터 추출된 텍스트 파일에서 욕설을 검열할 수 있어야 한다. 이 과정에서 효과적으로 욕설을 검열하려면 텍스트 내용을 더 이상 분석할 수 없는 가장 작은 말의 단위인 형태인 형태소로 나타낸 후 검열을 진행해야 했고, 따라서 우리 팀은 음성 검열에 Khaiii 형태소 분석기를 활용하기로 결정하였다. Khaiii는 다른 형태소 분석기들과는 다르게 신경망 알고리즘의 앞단에 기분석 사전, 뒷단에 오분석 패치라는 두 가지 사용자 사전 장치를 제공하는데, 이러한 두 기능으로 형태소 오분석 확률을 줄일 수 있고 본 프로젝트에 맞게 형태소 분석을 진행할 수 있다.

FastText

FastText는 단어를 벡터화 하는 Word2Vec과 비슷한 매커니즘을 가진다. 그러나 어휘를 최소 단위로 보는 Word2Vec과는 다르게 FastText는 텍스트의 최소 단위를 어휘를 구성하는 글자 n-gram으로 설정하였다. 즉 FastText는 어휘를 구성하고 있는 n-gram마다 하나씩 벡터값을 할당하고, 어휘를 구성하는 모든 n-gram 벡터의 평균 벡터를 어휘 임베딩으로 한다. 이러한 알고리즘을 통해 동일한 데이터 안에서 더 많은 정보를 활용할 수 있고, 또한 n-gram을 이용한 임베딩을 통해 여러 벡터값을 조합하여 학습 시 존재하지 않았던 어휘(OOV)에 대한 임베딩도 만들 수 있기 때문에 자주 등장하지 않는 생소한 어휘에 대해서도 준수한 성능을 보인다.

우리 팀은 Khaiii 형태소 분석기 결과물인 형태소들이 욕설인지 검열하기 위해서 FastText 임베딩 기법을 활용하였다. 많은 임베딩 기법들이 존재했지만 FastText 임베딩 기법은 위에 설명했듯이 가장 준수한 성능을 보였기 때문에 욕설 검열에 활용하기에 가장 적합하다고 생각하였다. FastText 모델 학습 및 개발 내용은 2.2 연구/개발 내용 및 결과물을 통해 살펴볼 수 있기 때문에 중복 기술을 피하기 위해 생략한다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

2.5.3 자막 검열

Google Vision API

광학 문자 인식(Optical character recognition; OCR)은 사람이 쓰거나 기계로 인쇄한 문자의 영상을 이미지 스캐너로 획득하여 기계가 읽을 수 있는 문자로 변환하는 것이다. 우리 팀은 자막이 존재하는 영상에서 자막에 있는 욕설을 검열하기 위해서는 프레임에서 문자를 찾을 수 있는 OCR 기술을 이용해야 했고, Google Vision API를 이용하여 이를 구현하였다.


Google Vision API는 TEXT_DETECTION과 DOCUMENT_TEXT_DETECTION이라는 두 가지 주석 특징에서 광 문자 인식(OCR)을 지원한다. TEXT_DETECTION은 임의의 이미지에서 텍스트를 감지하고 추출하며, DOCUMENT_TEXT_DETECTION도 이미지에서 텍스트를 추출하지만, 응답이 밀집된 텍스트와 문서에 맞게 최적화된다. 우리 팀은 이러한 Google Vision API를 영상의 프레임마다 적용시켜, 자막을 텍스트로 변환한 후 자연어 부분과 이미지 내의 해당 단어 좌표를 저장하고 이를 이용하여 원본 프레임에 존재하는 욕설 자막에 Bounding Box 처리를 한다.

2.5.4 Front End

ReactJS

ReactJS는 사용자 인터페이스를 만들기 위한, Facebook에서 제공해주는 JavaScript 라이브러리이다. JavaScript로 작성되는 Component 기반이며, 이러한 Component들을 조합하여 손쉽게 어플리케이션 UI를 만들 수 있다. 그리고 다양한 형식의 데이터를 앱 안에서 손쉽게 전달할 수 있고, DOM과는 별개로 상태를 관리할 수 있다. ReactJS는 또한 앱의 각 상태에 대한 간단한 뷰만 설계하기 때문에, 어플리케이션에서 필요로 하는 데이터가 변경됨에 따라 적절한 컴포넌트만 효율적으로 갱신하고 렌더링을 실시한다.

우리 팀은 프론트 엔드 개발을 ReactJS를 통해 진행하였다. 앞서 설명한 ReactJS의 장점과 같이 프로젝트의 웹 UI를 쉽고 효율적으로 개발할 수 있었고, 검열 결과 데이터, 백 엔드와의 소켓 통신 등 다양한 형식의 데이터를 활용하여 프론트 엔드를 구현할 수 있었다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

2.5.5 Back End

Amazon Web Service

2.2.8 항목과 동일한 내용이다. 중복을 피하기 위해 작성을 생략한다.

Apache Server

Apache 서버를 통해 웹페이지를 배포한다.


2.6 현실적 제한 요소 및 그 해결 방안

2.6.1 하드웨어

1. 모든 딥러닝 모델(Video Classification, FastText)을 학습시키기 위해서는 일반적인 로컬(데스크탑 PC, 노트북 등)에서는 RAM 메모리와 GPU 메모리의 양이 절대적으로 부족해 원활하게 진행되지 않는다. 따라서 우리는 AWS 의 Deep Learning AMI 를 이용해 학습 및 테스트를 진행할 것이다.
2. 딥러닝 모델이 5 개가 작동되기 때문에, 충분한 성능을 보이는 인스턴스에서도 여러 사용자가 이용할 때 제대로 동작하지 않기 때문에, 서버 인스턴스를 2 개로 나누어서 영상 검열과 음성 및 자막 검열을 담당하게 분산시켜서 부담을 줄였다.
3. Internet Explorer 에서는 우리가 만든 웹페이지 기능이 정상적으로 작동되지 않을 수 있다. 각종 라이브러리가 제대로 적용되지 않기 때문이다. 이를 크리에이터들이 Chrome 에서만 접속하여 이용할 수 있도록, 주의사항을 작성해 해결한다.

2.6.2 소프트웨어

1. 단순 AWS EC2 인스턴스 도메인으로 여러 이용자가 대용량 File Upload 를 진행할 시 서버의 속도가 느려질 뿐만 아니라, 용량의 제약에도 큰 영향을 받아 문제가 생긴다. 따라서 AWS S3 를 통해 이를 해결한다. AWS S3 는 무제한 클라우드로서 파일을 업로드하거나 다운로드할 때 S3 을 이용하면 여러 제약사항이 사라진다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

2. '빨간 원숭이의 해'(예: 2016 년)를 뜻하는 '병신년'과 시작점을 뜻하는 '시발점'등 욕설이 아님에도 불구하고, 욕설과 같은 글자들을 오분류하는 문제점이 발생한다. 따라서 FastText 딥러닝 모델을 이용하여, 문맥을 고려함으로써 해결한다.

3. 영상에서 프레임을 추출하고 각 프레임마다 딥러닝을 진행하기 때문에 검열 시간이 오래 걸린다. 따라서 우리는 웹페이지에서 데모 서비스로 제공하는 것이기 때문에, 1 분 이상 영상을 제한함으로써 검열 시간이 10 분 이내로 이루어질 수 있게 설정하였다.

4. avi 등 mp4 파일 형식이 아닌 영상 업로드시 썸네일 표시나 검열 결과 표시 등 웹브라우저 환경에서 작동되지않는다. 따라서 mp4 파일 형식으로 업로드 파일을 제한함으로써 이를 해결한다.

2.6.3 기타

1. 모든 Youtube 가이드라인을 맞추어서 검열을 진행하기에는 그 데이터셋 구축하는 시간과 노력이 상당하고, 단순 Video Classification 으로 검열할 수 없어서, 가이드라인 중 4 개에 충족되게끔 목표를 설정했다.

2.7 결과물 목록


2.2 ~ 2.6 항목까지 연구/개발 내용과 함께 결과물을 모두 서술하였다. 중복을 피하기 위해 작성을 생략한다.

2.8 기대효과 및 활용방안

검열 시스템은 Youtube 의 가이드 라인에 맞춰서 영상과 음성을 검열한다. 이는 현재 신고 기반으로 이루어지는 수작업 검열 과정보다 효율적으로 작동된다. 업로드 이전에 일어나는 자동 검열 시스템이므로 다양한 기대효과와 활용방안이 있다.

1. 검열 시스템을 적용함으로써 청소년에게 부적합한 영상들을 일차적으로 검열할 수 있다. 이는 동영상 업로드 플랫폼에서 실시하는 신고 기반 검열 시스템보다 더 많은 영상을 검열할 수 있고, 작업량과 비용 측면에서 효율적이다.

2. 추후 Youtube 이외의 다양한 실시간 스트리밍 서비스(Twitch, Afreeca TV 등)에서도 효과적으로 사용할 수 있다. 각 스트리밍 서비스들은 운영진이 직접 실시간 모니터링과


 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

시청자들의 신고를 통해 제재를 가하는데, 모든 과정이 수동 작업으로 이루어지기 때문에 효과적으로 이루어지지 않는다. 따라서 방송되고 있는 장면들을 실시간으로 딥러닝 모델에 넣어 검열할 수 있다.

4. 현재는 미성년자들을 대상으로 부적합한 영상들을 검열하는 시스템을 목표로 하지만, 데이터셋이 많아지고 다양한 라벨에 대해서 학습을 시킨다면, 교통사고, 자연 재해, 길거리 싸움, 집단 구타 및 학대 등 검열 대상을 확대시켜 활용이 가능하다. 또한 유해한 동영상의 업로드를 제한함으로써 사이버 범죄 예방에 도움을 준다.

5. 현 음성 검열 시스템은 단순 욕설 단어에 대해서만 진행되지만, 데이터셋을 확대시켜 학습시킨다면 욕설을 포함하지 않은 비방 목적의 문장 또한 검열이 가능하다.

6. 이 시스템이 적극적으로 활용된다면, 동영상 업로드 플랫폼에 대한 사람들의 신뢰도와 인식의 향상에 도움이 된다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08


3 자기평가

1. 영상 검열과 음성 및 자막 검열을 하기위해 단순히 인공지능 모델들을 차용해서 사용하는 것이 아닌, 여러가지 모델들을 직접 테스트해보고 인공지능과 인공지능을 연결해 모델별 결과를 취합하여 정확도를 높이는 등 정확도를 높이기 위해 많은 노력을 기울였고 성과 자체도 Precision / Recall 매트릭스의 높은 정확도로 우수하게 표현됐다.
2. 웹페이지에서 동영상을 업로드할 때와 검열을 진행할 때, 최대한 사용자들의 편의를 위해 Notification 을 띄워 현재 진행 중인 상황을 알려주고, 이해하기 편하게 UI 를 제공했다. 즉, 단순히 업로드가 가능하고 검열이 진행되는 웹페이지가 아닌 사용자의 편의를 최대한 고려하여 설계되었다.
3. S3 에 업로드할 때 편하게 SECRET KEY 와 ACCESS ID 를 코드에 입력하여 업로드하는 것이 아닌 최대한 보안을 유지하기 위해 Amazon Gateway API 를 이용하여 Pre Signed URL 을 통해 업로드한다. 또한 인스턴스에서 딥러닝 모델이 여러 개가 작동됨으로써 이미 서버 부담이 큰 상태에서 프레임 추출을 진행하기에는 무리라고 판단하여, S3 에서 프레임을 추출하도록 Lambda 함수를 구현했다. 이 모든 과정을 진행하면서 Amazon Web Service 를 적극 활용하여 백엔드를 구축할 수 있었다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

4 참고 문헌


번호	종류	제목	출처	발행년도	저자	기타
1	기사	[NDC2018] "스 111 발" 도 잡아내는 옥설 탐지기, 딥러닝으로 만들기	http://www.inven.co.kr/webzine/news/?news=198156	2018	정필권	
2	기사	넷플릭스 국내이용자 185 만...1 년새 4 배 증가	https://m.mk.co.kr/uberin/read.php?sc=30000001&year=2019&no=678422	2019	박창영	
3	규정집	방송언어 가이드라인	방송통신심의위원회	2019	방송통신심의위원회	
4	법령	청소년보호법	국가법령정보센터	2018	여성가족부	
5	YouTube	YouTube 정책 및 안전	Youtube	2020	YouTube	
6	깃허브	Tutorial for video classification/ action recognition using 3D CNN/ CNN+RNN on UCF101	https://github.com/HHTse ng/video-classification	2019	Huan-Hsin Tseng	
7	깃허브	Kakao Hangul Analyzer III	https://github.com/kakao/khaiii	2019	Kakao	
8	깃허브	FastText for Korean	https://github.com/skyer9/FastTextKorean	2019	sky9	

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

5 부록

5.1 사용자 매뉴얼

Upload



도움말 및 제안사항


- 필터링에 소요되는 시간은 약 10분이며, 사용자의 인터넷 환경에 따라 달라질 수 있습니다.
- 재생 시간이 1분 미만인 영상만 업로드 가능합니다.
- 업로드 파일 형식은 mp4로 제한합니다.
- 업로드를 완료한 후, Clear 버튼을 통해 다른 영상을 업로드할 수 있습니다.

업로드

검열

Upload

파일 등록



도움말 및 제안사항


- 필터링에 소요되는 시간은 약 10분이며, 사용자의 인터넷 환경에 따라 달라질 수 있습니다.
- 재생 시간이 1분 미만인 영상만 업로드 가능합니다.
- 업로드 파일 형식은 mp4로 제한합니다.
- 업로드를 완료한 후, Clear 버튼을 통해 다른 영상을 업로드할 수 있습니다.

test_smoke.mp4

업로드

검열

Upload



검열


도움말 및 제안사항

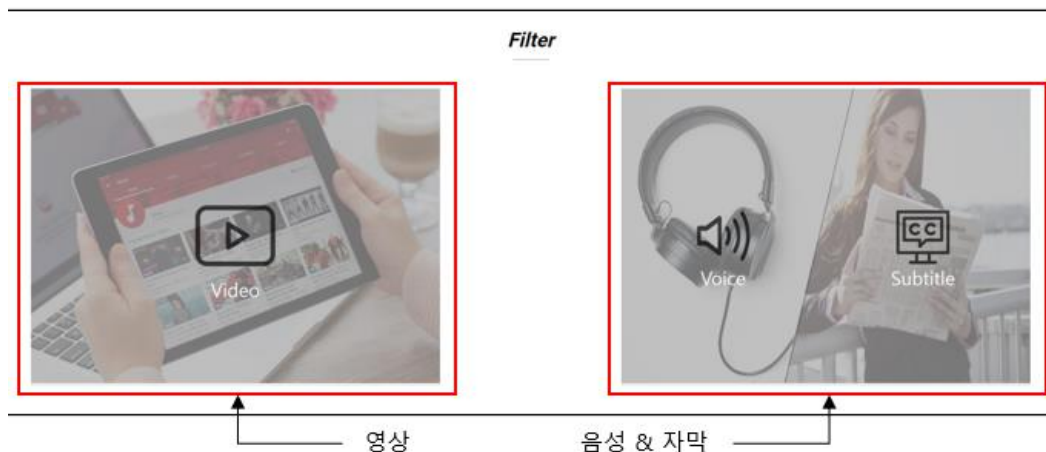
- 필터링에 소요되는 시간은 약 10분이며, 사용자의 인터넷 환경에 따라 달라질 수 있습니다.
- 재생 시간이 1분 미만인 영상만 업로드 가능합니다.
- 업로드 파일 형식은 mp4로 제한합니다.
- 업로드를 완료한 후, Clear 버튼을 통해 다른 영상을 업로드할 수 있습니다.

test_smoke.mp4

Clear

검열

 <div> <p>국민대학교</p> <p>컴퓨터공학부</p> <p>캡스톤 디자인 I</p> </div>	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08



영상 검열 결과


4
전체 결과

0
Adult

0
Blood

0
Knife

4
Smoke



확인

▶ Adult

▶ Blood

▶ Knife

▼ Smoke

0:01 - 0:04

0:04 - 0:07

0:07 - 0:10

결과 확인

음성, 자막 검열 결과




확인

업로드

'생각중'	0:10
'시뮬'	0:12
'생각중'	0:13
'생각중'	0:16
'생각'	0:20
'생각생각'	0:21
'생각'	0:24
'생각생각중'	0:27
'생각중'	0:28
'생각'	0:29
'생각중'	0:30

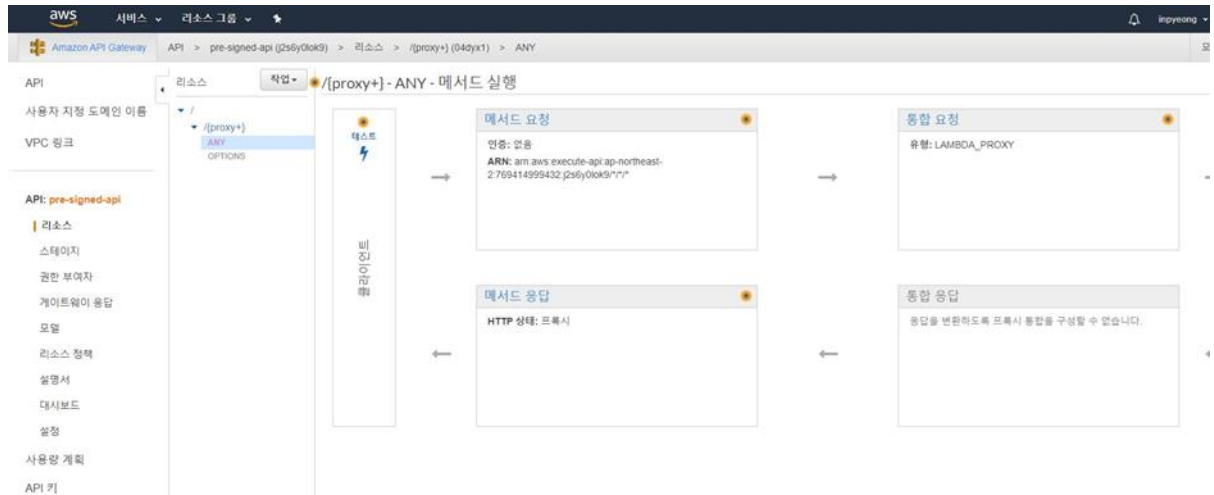
▶ 0:00 / 0:14

결과 확인

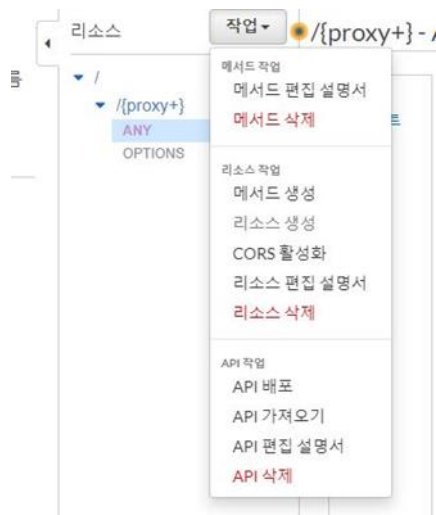
 <div> 국민대학교 컴퓨터공학부 캡스톤 디자인 I </div>	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

5.2 운영자 매뉴얼


5.2.1 AWS Gatewap API 설정하기

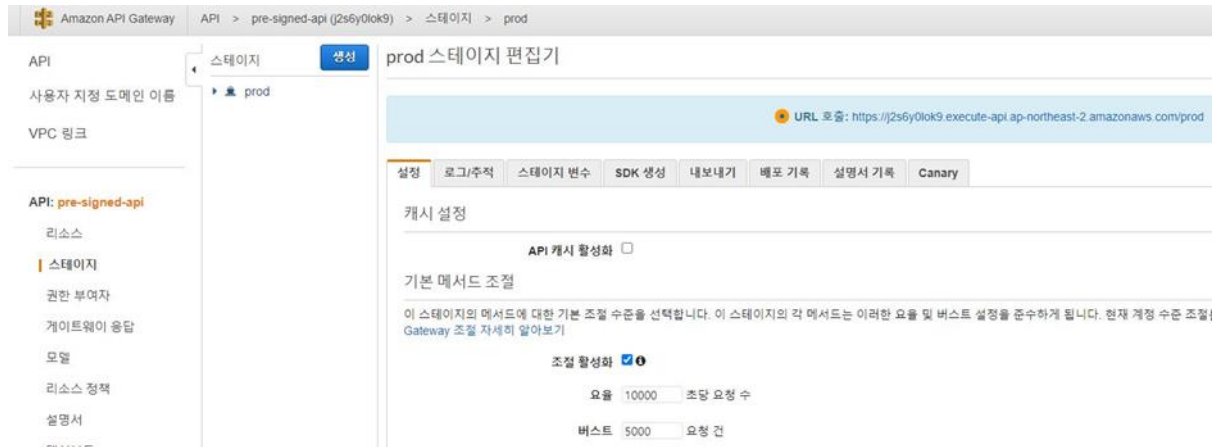


presigned url 을 만들어서 전송해주는 API 를 생성합니다. 메소드는 ANY 로 생성하여 Get 메소드로 요청할 수 있게 합니다.



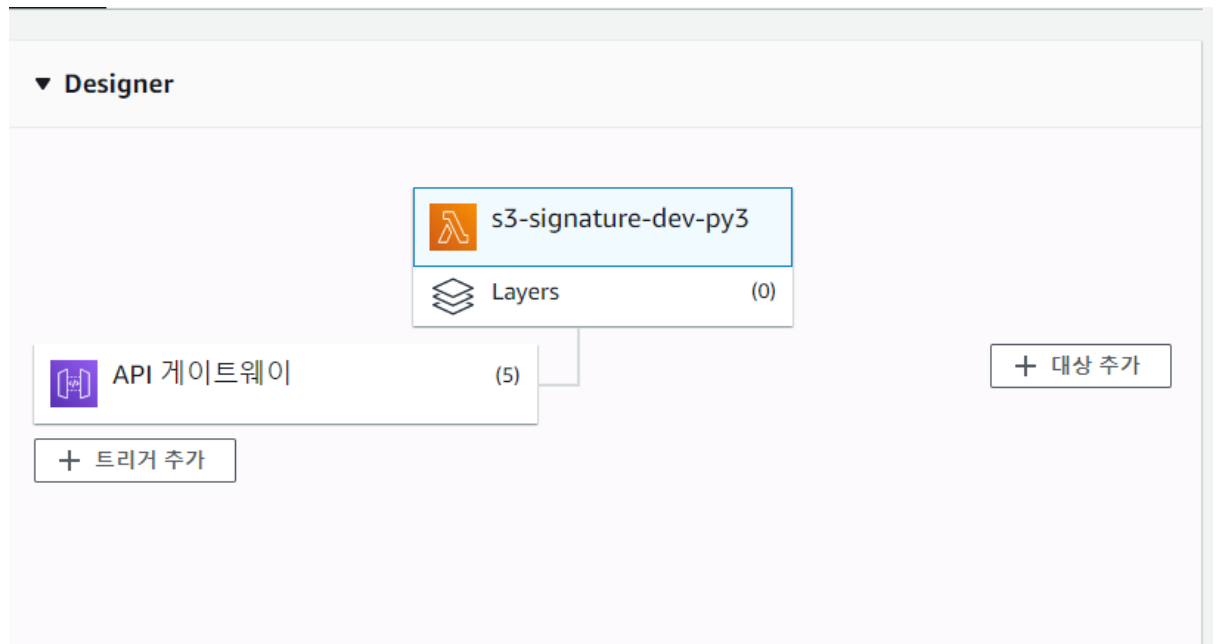
작업 버튼을 눌러 CORS 활성화를 설정합니다. 모든 칸에 "*"를 입력해 CORS 를 모두 허용해줍니다. 이후 API 배포 버튼을 클릭해 API 를 배포합니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08




5.2.2 AWS Lambda 함수 설정하기

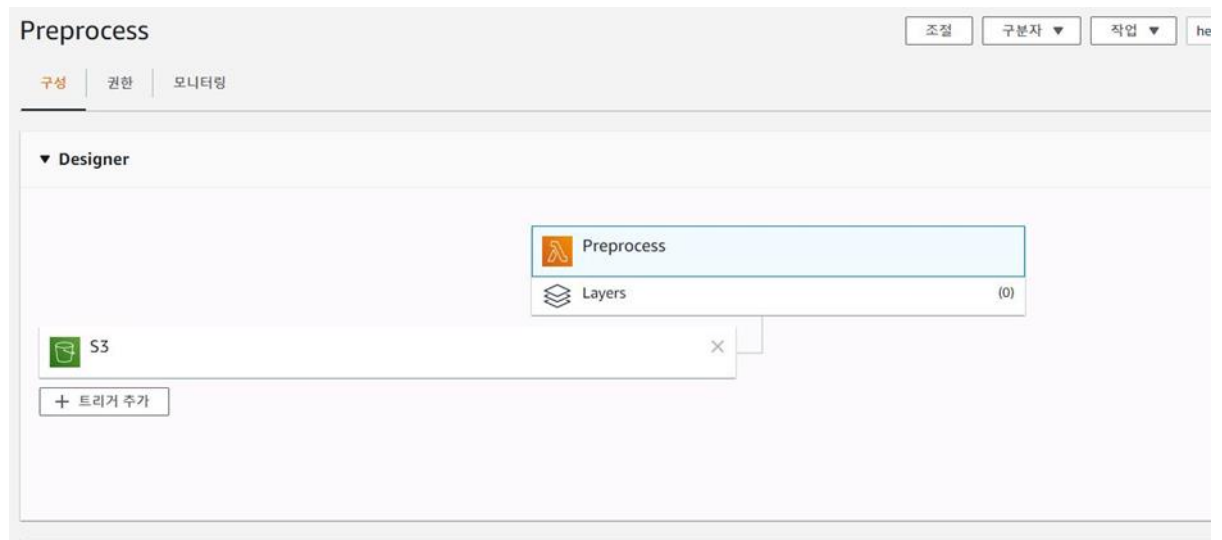
Lambda 함수를 생성하고 다음과 같이 생성한 Gateway API 를 트리거로 설정해줍니다.



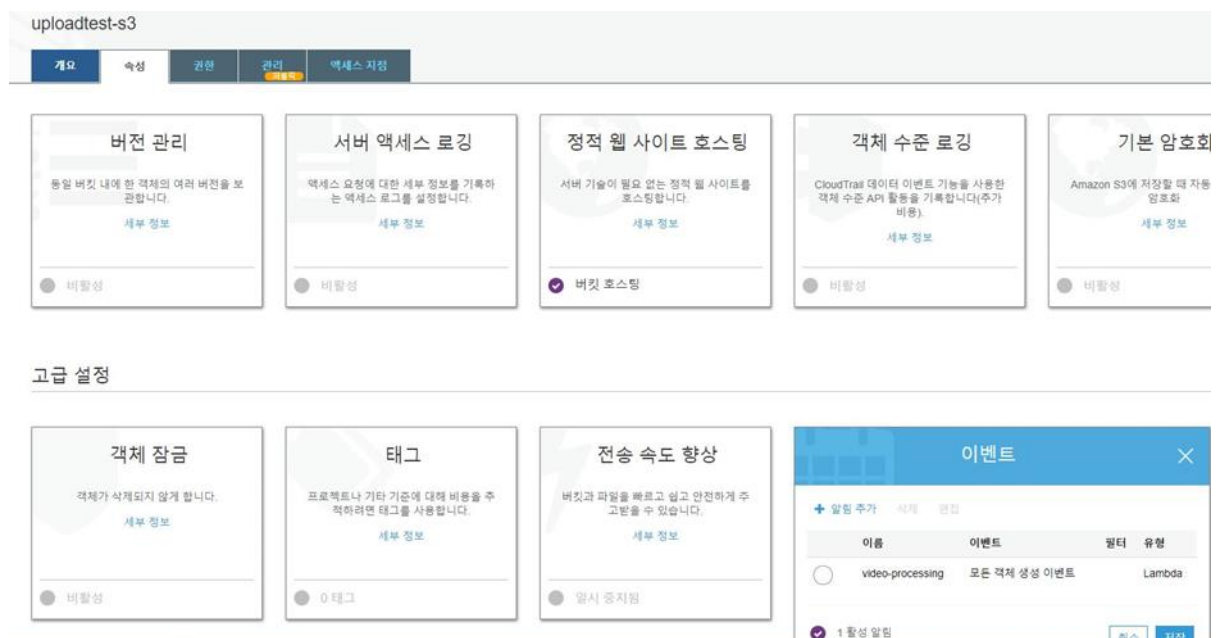
Lambda 코드는 클론 받은 폴더의 aws 에 pre-signed-url 폴더의 lambda_function.js 입니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

S3 에 동영상 파일이 업로드되면 프레임을 추출하는 Lambda 함수를 생성합니다. S3 버킷을 트리거로 등록합니다.



Lambda 함수의 코드는 클론 받은 폴더의 aws/preprocessing 폴더의 lambda_function.py 입니다. 또한 단순히 lambda_function.py 만 등록하면 되는 것이 아닌 해당 라이브러리들을 모두 압축해서 업로드해야하기 때문에, 필요한 라이브러리들을 설치하여 압축해서 업로드합니다.



위의 Lambda 함수에서 트리거로 등록한 S3 버킷 설정에서 이벤트를 클릭해 모든 객체 생성 이벤트로 Lambda 함수 실행 조건을 등록합니다.

5.2.3 AWS EC2 인스턴스 생성하기



웹서버와 영상 검열을 진행하는 EC2 를 생성합니다. p2.xlarge 를 선택하는 것이 좋으며, Ubuntu DeepLearning AMI 인스턴스를 생성합니다.

▼ 퍼블릭 IPv4 주소 ▼	할당 ID ▼	연결된 인스턴스 ▼
13.209.93.181	eipalloc-022079e2c93b7687e	i-0b8ce60173082d48e 🔗


외부에서 소켓을 연결하거나, Jupyter Notebook, 혹은 아파치 웹서버로 배포되는 웹페이지 접속시 IP 주소를 사용하는데, IP 주소가 매번 바뀌기 때문에 탄력적 IP 주소를 설정합니다.

인바운드 규칙			
유형	프로토콜	포트 범위	소스
HTTP	TCP	80	0.0.0.0/0
HTTP	TCP	80	::/0
사용자 지정 TCP	TCP	8888	0.0.0.0/0
사용자 지정 TCP	TCP	8888	::/0
SSH	TCP	22	0.0.0.0/0
SSH	TCP	22	::/0
사용자 지정 TCP	TCP	4567	0.0.0.0/0
사용자 지정 TCP	TCP	4567	::/0

생성한 EC2 인스턴스의 보안 규칙을 다음과 같이 설정하여 외부에서 접속 가능하게 합니다. 포트 번호 4567 은 소켓 통신에 사용되고, 8888 은 Jupyter Notebook 에서 사용됩니다.

i-042b92e2ce39c874e	p2.xlarge	ap-northeast-2c	● running
▼ 퍼블릭 IPv4 주소 ▼	할당 ID ▼	연결된 인스턴스 ▼	
3.34.55.213	eipalloc-026eccb530c611407	i-042b92e2ce39c874e 🔗	

마찬가지로 음성 및 자막을 검열하는 인스턴스도 생성합니다. 위 인스턴스와 마찬가지로 탄력적 IP 주소를 설정합니다. 또한 보안 규칙도 위와 동일하게 설정합니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	동영상 연령제한 필터링	
	팀 명	YouHi	
	Confidential Restricted	Version 1.3	2020-JUN-08

첫번째 인스턴스에서 `sudo apt-get install apache2` 를 입력해 아파치 웹서버를 구축합니다.

`/etc/apache2/sites-available/000-default.conf` 에 Header set Access-Control-Allow-Origin "*" 를 적어줍니다.

`sudo apt install php php-mysql` 을 입력해 php 를 설치합니다.

`cd /var/www/html/`을 입력해 html 로 이동한 후 프로젝트를 클론받습니다. 클론 폴더에서 `src/web-front` 로 이동해 `sudo npm install` 과 `sudo npm run build` 를 입력해 `build/static` 과 `index.html` 을 생성합니다. 이후 `/var/www/html/`에 `build/static` 과 `index.html` 을 넣습니다.

`static` 폴더를 `sudo chmod 777 static` 명령어로 권한을 열어줍니다.(외부에서 파일을 생성하고 폴더를 이동시키기 위함)

클론 폴더에서 `pip install -r video_requirements.txt` 를 입력하여 필요한 패키지를 모두 설치합니다.

클론 폴더 `video-backend` 에서 `nodejs server.js` 명령어를 통해 소켓 서버를 열어줍니다.

두번째 인스턴스에서 `voice_classification` 폴더를 생성하고 프로젝트를 클론 받습니다. `src/voice-backend` 의 내용물을 모두 `voice_classification` 에 넣습니다. 이후 `pip install -r voice_requirements.txt` 를 입력해 필요한 패키지를 모두 설치합니다.

클론 폴더 `voice-backend/server` 에서 `nodejs server.js` 명령어를 통해 소켓 서버를 열어줍니다.

5.3 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과
스크립트	Python	영상 검열을 진행한다.	segmentation.py를 실행한다.	영상에서 칼에 찔리는 장면, 담배를 피는 장면, 여자 상반신이 노출되는 장면의 위치를 알려준다.	성공
스크립트	Python	자막이 있는 이미지에 욕설 표시	자막이 있는 이미지와 해당 이미지에 매칭된 욕설 좌표값이 있을 때 subtitle.py를 실행한다.	자막에 포함된 욕설에 바운딩 박스를 그려준다.	성공
API	API	S3에 영상을 보안을 유지하면서 업로드한다.	API Gateway API를 호출하는 URL을 Get 요청한다.	S3에 업로드할 수 있는 Pre Signed URL을 얻는다.	성공
API	API	인스턴스가 아닌 S3에서 동영상 전처리 작업을 수행한다.	Lambda 함수에 트리거로 설정된 S3 버킷에 동영상을 업로드한다.	Lambda 함수에 명시된 버킷에 추출된 프레임들이 저장된다.	성공
스크립트	Python	음성 검열을 진행한다.	main.py를 실행한다.	한국어 음성이 포함된 영상에서 욕설을 추출한다.	성공
스크립트	React, JavaScript	웹의 build/static 폴더 안에 존재하는 검열 결과 텍스트 파일을 파싱하여 React Modal에 시각화한다.	검열이 완료된 후 검열 결과 확인 박스를 클릭하여 렌더링되는 React Modal을 확인한다.	영상, 음성, 자막 검열 결과가 성공적으로 React Modal에 시각화되었다.	성공