



국민대학교  
소프트웨어융합대학  
소프트웨어학부

# 캡스톤 디자인 I

## 종합설계 프로젝트


프로젝트 명	영상 품질 개선 공유 플랫폼
팀 명	Memories
문서 제목	Memories 중간보고서

Version	0.6
Date	2020-APR-21

팀원	김병조(조장)
	곽윤희
	김수진
	이혜진
	권우철
지도교수	이재구 교수님

### CONFIDENTIALITY/SECURITY WARNING


이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 “영상 품질 개선 공유 플랫폼”을 수행하는 팀 “Memories”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “Memories”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

문서 정보 / 수정 내역


Filename	Memories 중간보고서.pdf
원안작성자	김병조, 곽윤혁, 김수진, 이혜진, 권우철
수정작업자	김병조, 곽윤혁, 김수진, 이혜진, 권우철

수정날짜	대표수 정자	Revision	추가/수정 항목	내 용
2020-04-11	권우철	0.1	최초 작성	표지, 머리글, 프로젝트 목표
2020-04-13	김병조 외 4명	0.2	수정	수행내용
2020-04-17	김병조 외 4명	0.3	수정	수행내용, 고충 및 건의사항
2020-04-18	김병조 외 4명	0.4	수정	향후 추진 계획
2020-04-19	김병조 외 4명	0.5	수정	수행내용, 전체적인 내용 검토 및 수정
2020-04-21	김병조 외 4명	0.6	수정	고충 및 건의사항 등 수정

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

## 목 차

<b>프로젝트 목표</b>	<b>4</b>
<b>수행 내용 및 중간결과</b>	<b>4</b>
계획서 상의 연구내용	4
2.1.1 품질 개선 모듈	4
수행내용	8
2.2.1 품질 개선 모듈	8
2.2.2 서버	9
2.2.3 웹	10
2.2.4 결과물 목록 및 진행사항	16
<b>3 수정된 연구내용 및 추진 방향</b>	<b>18</b>
3.1 수정사항	18
<b>향후 추진계획</b>	<b>20</b>
향후 계획의 세부 내용	20
<b>고충 및 건의사항</b>	<b>20</b>
<b>참고문헌</b>	<b>21</b>

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	영상 품질 개선 공유 플랫폼	
	팀 명	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

## 1 프로젝트 목표

본 프로젝트는 사용자들이 다시 보고 싶은 그 시절 그 장소를 검색하여 생생한 화질로 복원된 영상들을 보며 추억을 떠올리거나, 함께 공유하고 싶은 영상을 업로드하여 고화질로 변환된 영상을 모든 사용자가 다 같이 볼 수 있게끔 할 수 있는 플랫폼을 제공하는 것을 목적으로 한다. 기존 영상이 흑백이라면 컬러 영상으로, 60fps 미만의 영상이라면 frame interpolation을 추가하여 60fps 영상으로 복원하고, 1080p 미만의 저화질 영상이라면 1080p의 고화질 영상으로 바뀌어서 기존보다 더 자연스럽고 생생한 영상을 만드는 것을 목표로 한다. 플랫폼을 완성하여 사용자들에게 제공한다면, 기존 타 사이트에서 제공하는 품질 개선 기능을 일일이 찾으러 다니지 않고, 한 사이트에서 일괄적으로 품질 개선 할 수 있다.

사용자가 접속할 수 있는 웹 앱을 만들어야 하며, 서비스에 접속했을 때 사용자는 서비스가 무엇을 하는 플랫폼인지 쉽게 알 수 있어야 한다. 또한 장소와 시간으로 검색이 가능해야 하고 새로운 영상 업로드가 간편해야 한다. 본인이 업로드한 영상 관리도 간편하게 할 수 있어야 한다.


서버는 많은 사용자가 한 번에 영상을 업로드해도 견딜 수 있도록 scalable하게 설계되어야 하며, 영상 품질 향상을 위한 여러 가지 모델(Colorization, Video Frame Interpolation, Super resolution)이 존재해야 한다. 또한 업로드된 영상을 품질 향상 모델들에 전달해주기 전에 알맞은 포맷으로 전처리할 수 있어야 한다. 현 프로젝트는 사용자가 영상을 업로드 했을 때, 40분 이내에 품질 개선이 완료되는 것을 목표로 한다.

## 2 수행 내용 및 중간결과

### 2.1 계획서 상의 연구내용

#### 2.1.1 품질 개선 모듈

##### 1) Colorization 모듈


 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

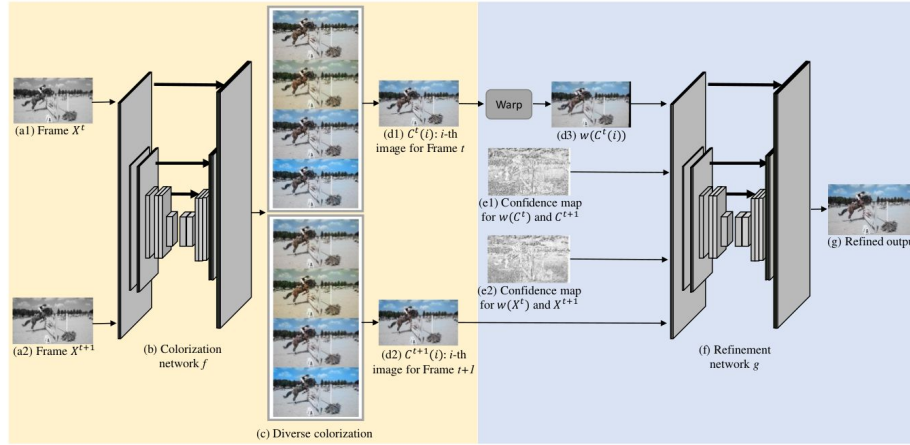
컬러화(Colorization)는 흑백 영상을 그럴듯한 색깔을 입혀서 컬러영상으로 만드는 것이 목적이다. Colorization의 종류는 참조 이미지 기반의 Colorization과 자동(Automation) Colorization가 있다. 참조 이미지 기반의 Colorization은 주어진 이미지 색깔을 참조하여 색상을 입히는 것이고, 자동(Automation) Colorization은 학습한 모델이 자동으로 그럴듯한 색깔을 만들어내는 것이다. 참조 이미지 기반의 Colorization은 참조 이미지의 색깔에 한정해서 Colorization을 하므로 다채로운 색깔을 보여줄 수 없으리라 판단하였다. 그래서 우리는 자동(Automation) Colorization을 선택하였다.

자동 비디오(Automation Video) Colorization에서 중요하게 다루는 것은 색깔이 입혀진 프레임들 사이의 자연스러움을 살려야 한다는 것이다.  $t$  시간의 프레임과  $t+1$  시간의 프레임이 전혀 다른 색깔이 된다면 영상이 자연스럽지 못할 것이다. 이런 문제를 해결하기 위해서 프레임간의 Optical Flow와 일관성(Coherence)을 고려한 기술이 필요하다. 조사한 결과, Automation Colorization이면서 Optical Flow를 고려한 Fully Automatic Video Colorization with Self-Regularization and Diversity(CVPR 2019) 논문이 지금 문제를 해결하기 위해 가장 적합하다고 생각하여 선택했다.

밑의 그림이 Colorization 모델 아키텍처이다. 이 논문의 모델을 살펴보면, 크게 2개의 네트워크로 구성되어 있다. 하나는 Colorization 네트워크이며, 이 네트워크에서 흑백인 프레임을 입력으로 받아 출력으로 컬러화된 프레임이 나온다. 다른 모델들과 다른 점은 1대 1 결과가 아닌 1대 다 결과라는 것이다. 즉, 하나의 프레임을 넣으면, 색깔 또는 느낌이 다른 결과들이 나온다. 다양한 결과를 내뱉음으로써 기존 Colorization의 문제점인 결과의 애매모호함을 해결하려고 하였다. 하지만 우리 연구에서는 하나의 결과만 사용할 것이다.

또 하나의 네트워크는 Refinement Network이다. 이 네트워크는 시간적 일관성(Temporal Consistency)을 지키기 위해서 만든 네트워크이다. 시간적 일관성(Temporal Consistency)은 연속하는 두 개의 프레임에서 일치하는 물체가 비슷한 색깔이 아닐 때를 뜻한다. 이렇게 두 개의 네트워크를 지나고 나면 컬러화된 하나의 프레임이 나오게 된다.


 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

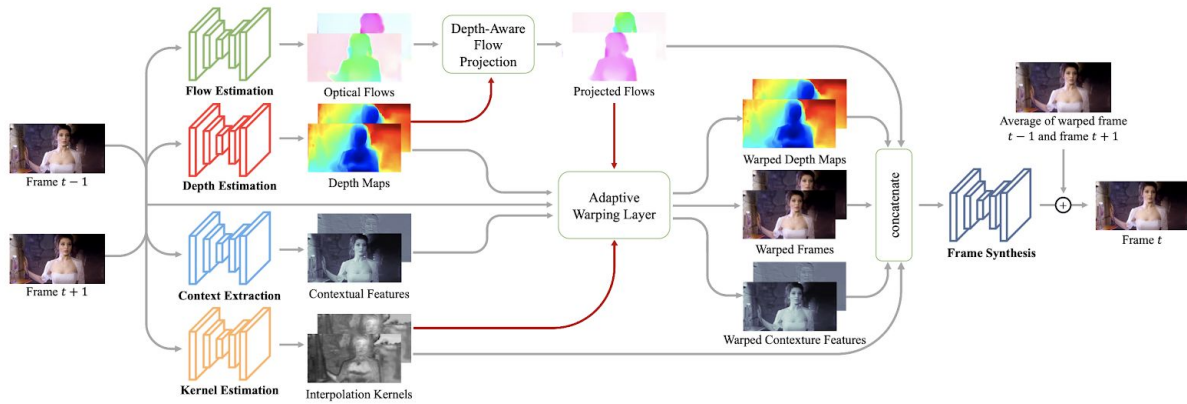


〈Colorization architecture 구조〉

## 2) Frame Interpolation 모듈

프레임 보간(Frame Interpolation)은 비디오에서 두 프레임 사이의 프레임들을 생성하는 것을 말한다. Video Frame Interpolation 모델을 개발하기 위해 두가지의 최신 연구를 참고했다. 첫번째는 "Super SloMo: High quality estimation of multiple intermediate Frames for video interpolation."이고 두번째는 "Depth-Aware Video Frame Interpolation."이다. Frame Interpolation은 슬로우 모션 영상 생성, 애니메이션 제작 작업 간소화 등 적용할 분야가 많다. 그러나 기존의 많은 연구에서는 물체의 폐색(Occlusion)이나 큰 움직임(large object motion)이 발생하였을 때 품질이 좋지 않다는 문제점이 존재했다. 또한 기존의 연구들은 Frame Interpolation을 할 때 임의의 프레임 개수를 생성해 내는 것이 매우 까다롭고 제약이 많았다. 두 프레임 사이에 여러 프레임을 생성하고 싶다면 해당 알고리즘을 재귀적으로 적용하여야 했다. 따라서 병렬화가 힘들었으며  $2^N - 1$ 의 개수의 프레임만 생성할 수 있었다. 하지만 첫번째 참고논문인 Super SloMo는 두 프레임 사이에 임의의 개수의 프레임을 생성할 수 있으며 병렬화가 비교적 간단하다. 또한 End-to-End로 처리가 가능하다. 두번째 논문인 DAIN은 이보다 더 나아가 큰 움직임을 잘 캐치하기 위해 Coarse-to-Fine 전략을 사용하였다. 폐색(Occlusion)을 해결하기 위해서는 여러 방향으로 Optical Flow를 계산한 뒤 추출한 Depth Map을 고려해서 집합시킨다. 기존 연구들에 비해 더 깔끔한 Motion Boundary가 고려된다.

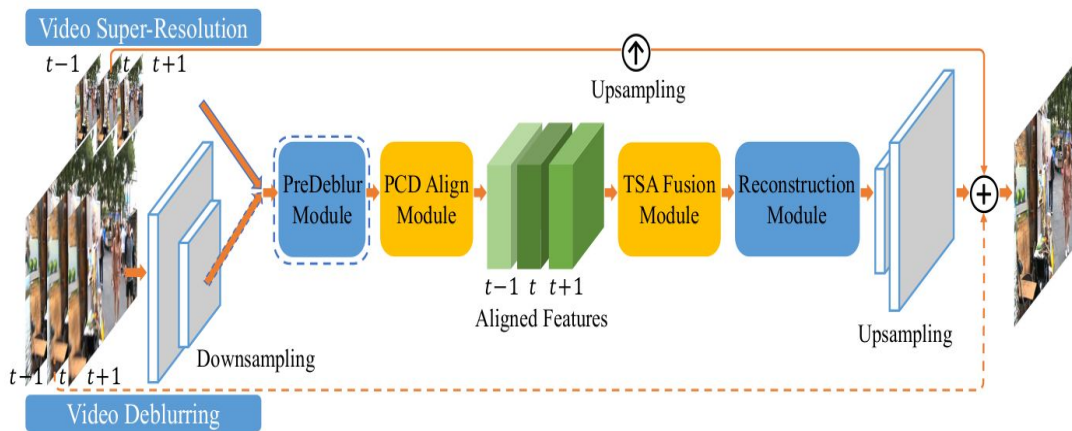
 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	영상 품질 개선 공유 플랫폼	
	팀 명	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21



< DAIN architecture 구조 >


위는 DAIN의 구조이다. 두 장면간의 Optical Flow와 Depth Map을 이용하여 Flow Projection을 Depth-Award Flow Projection Layer에서 수행한다. 여기서 나온 집합된 Optical Flow를 가지고 입력 프레임들과 Depth Map, 계층적 특징을 Warp시키며, Warp된 세가지 결과물들과 투영된 흐름, Local Interpolation Kernel을 이용하여 최종 결과(Interpolated 프레임)를 만든다.

### 3) Super Resolution 모듈



< Super Resolution architecture의 구조 >

초해상화(Super Resolution)은 저해상도 입력을 고해상도로 변환해 주는 것을 목적으로 한다. Super Resolution에는 비디오 초해상화(Video Super Resolution)와 이미지 초해상화(Image

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

Super Resolution)이 있다.

비디오도 여러 프레임이 모여 만들어진 것이기 때문에 Image Super Resolution을 사용해서 각 프레임의 해상도를 높이는 방법도 가능하다고 생각을 했었다. 하지만 Image Super Resolution은 한 장의 사진에 대해서만 저화질을 고화질로 변환을 하면 되지만 비디오의 경우에는 변환하려는 프레임의 앞/뒤 프레임의 시공간적 정보도 고려해야 한다. 즉, 비디오 프레임을 Image Super Resolution을 사용할 경우 프레임을 각각 봤을 때는 잘 처리된 것처럼 보여도 비디오로 합쳤을 때는 중간중간 어색한 부분이 있으리라 판단했다. 그래서 우리는 Image Super Resolution의 가능성을 접어두고, Video Super Resolution 기술을 선택했다. 변형 가능한 컨볼루션을 기반으로 한 PCD Align 모듈과 TSA Fusion 모듈을 사용하여 다른 모델들보다 복잡한 행동이 들어 있는 비디오를 더 잘 처리하는 EDVR 모델을 선택했다.

이 논문의 모델은 처음 7개의 Low Video 프레임을 입력으로 한다. 그리고 계산해야 하는 값을 줄이기 위해 먼저 Downsampling을 진행하고 PreDeblur 모듈을 거쳐 입력 프레임의 흐릿한 부분은 없애는 역할을 한다. 그런 다음 PCD Align 모듈을 거쳐 다른 모델들과 다르게 복잡한 동작이 있는 부분을 조금 더 부드럽게 처리하는 역할을 한다. 그런 다음 TSA 모듈에서는 기준 프레임과 인접 프레임 사이에서의 시간적 요소를 융합한다. 이미지와 달리 비디오는 앞 프레임과 뒤 프레임 사이의 시공간적 요소를 고려해야 하기 때문이다. 마지막으로 앞서 작업한 정렬(Align)과 융합(Fusion) 모듈을 통해 Reconstruction 모듈에서 재구성하고 Upsampling을 하여 1개의 고해상도 프레임을 출력해준다. 결과적으로 입력 해상도의 4배 값이 출력 해상도로 출력된다.


## 2.2 수행내용

### 2.2.1 품질 개선 모듈

#### 1) Colorization 모듈

Colorization 모듈은 Fully Automatic Video Colorization with Self-Regularization and Diversity(CVPR 2019) 논문의 구현체를 사용하였다. 기존 구현체를 그대로 사용할 경우, Cuda 환경이나 Package Dependency 문제가 있어서 코드가 실행되지 않는다. 그 문제를 해결하기 위해서 텐서플로우 1.2.0 버전을 2.0.0 버전에서 사용할 수 있도록 코드를 수정하였다. 기존 네트워크에서는 tf.contrib.slim layer를 사용하지만 2.0.0에서는 지원하지 않기 때문에, 컨볼루션 층을 tf.layers.conv2d로 바꾸었다. 사전 훈련 파라미터 또한 slim.conv2d와 layers.conv2d의 저장되는 파라미터 이름이 다르기 때문에 변경하는 스크립트를 구현하였다. 그리고 원래 모듈에는



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

Colorization 네트워크와 Refinement 네트워크가 사용되지만, 사전 훈련(Pre-trained) 파라미터는 Refinement 네트워크가 빠져있는 상태로 학습이 되었다. 아직 학습을 시키지 않아 사전 훈련(Pre-trained) 파라미터를 사용하였고, Colorization 네트워크만 사용되어 컬러 영상이 부자연스러워 보일 수가 있다. 중간평가 이후에 전체 모델을 사용하여 학습시킨 파라미터를 사용할 것이며 이 문제는 해결될 것으로 보인다.

## 2) Frame interpolation 모듈


Video Frame interpolation 모듈은 DAIN(Depth-Award Video Frame Interpolation)이라는 논문의 구현체를 사용하였다. 하지만 현재 그래픽 카드 메모리의 용량 한계로 중간 프레임을 3개 이상 생산해 낼 수 없는 문제점이 발생하여 60fps의 요구사항을 항상 맞출 수는 없게 되었다.

## 3) Super Resolution 모듈

Super Resolution 모듈은 EDVR: Video Restoration with Enhanced Deformable Convolutional Networks라는 논문의 구현체를 사용했다. 베타 버전에서 사전에 훈련된 파라미터를 가지고 테스트를 해보았다. 아직 제대로 훈련된 파라미터를 사용하지 않아서 그런지 개인이 가지고 있는 영상으로 테스트를 해본 결과 글씨나 흐려짐(Blur) 처리가 제대로 이루어지지 않는 현상이 발견되었다. 이 문제에 대해서는 직접 데이터를 모아서 훈련을 시킨 다음 정확하게 다시 테스트해볼 예정이다. 또한 현재는 Colab에서 코드를 테스트하고 있어 cuda memory 용량 때문에 360p 영상까지만 테스트가 가능하다. 이 문제는 직접 학습을 시킬 때부터 T4 서버에서 훈련을 시킬 것이기 때문에 문제가 해결될 것으로 보인다.

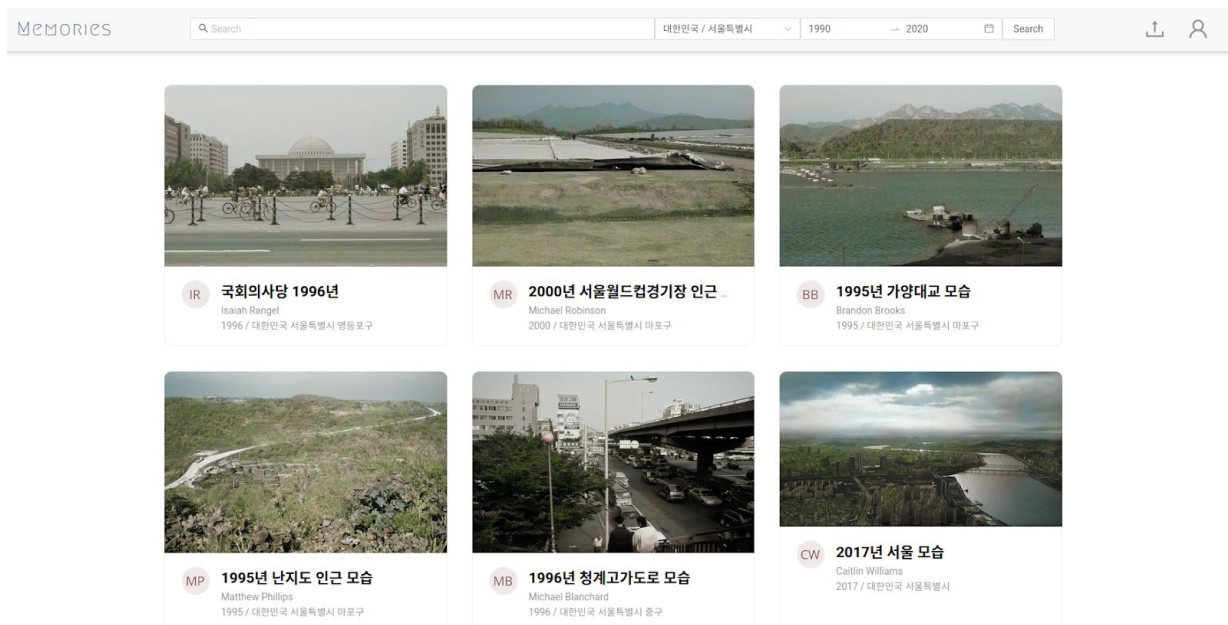
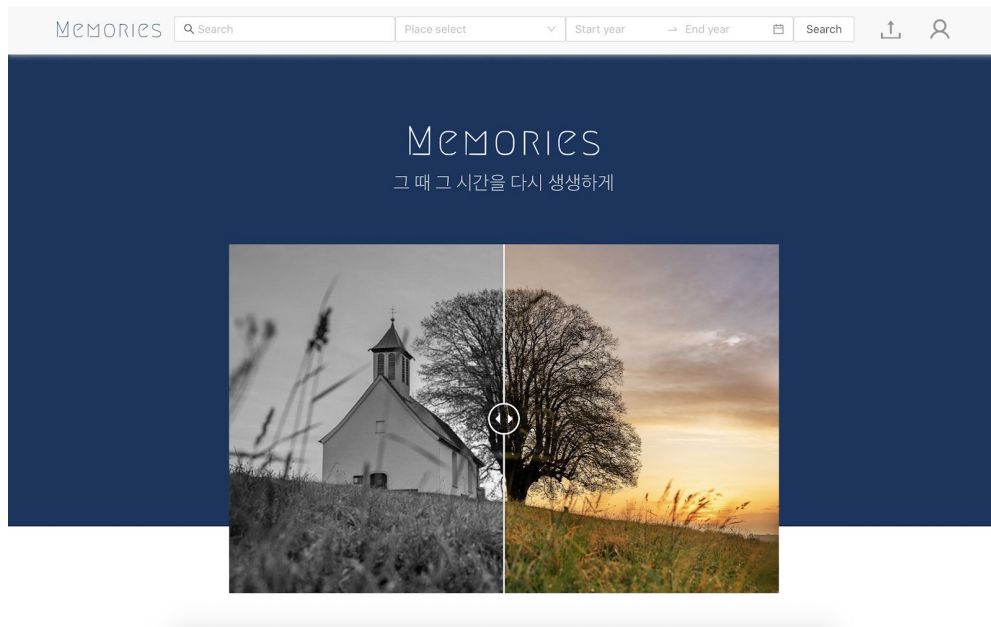
### 2.2.2 서버

베타 버전에서는 클라우드 서비스를 사용하지 않고 주요 기능들을 구현하는 것을 목표로 하였다. 품질 향상 모듈을 쉽게 호출하기 위해 Python으로 서버를 만들고자 Flask와, Aridane라는 GraphQL server framework를 사용하여 클라이언트가 정보를 요청할 수 있는 API 서버를 구성하였다. 클라이언트는 GraphQL schema에 따라 GraphQL로 서버에 미디어 조회, 서치, 미디어 업로드, 삭제 등의 요청을 보낼 수 있다.


 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

## 2.2.3 웹

### < HomePage > - 소개 페이지

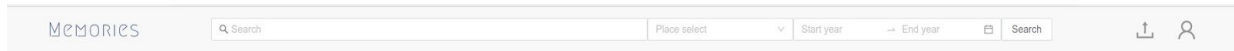


사용자가 홈페이지에 접속하면 첫 화면으로 페이지에 대한 간단한 소개와 품질 개선 모듈을 통해

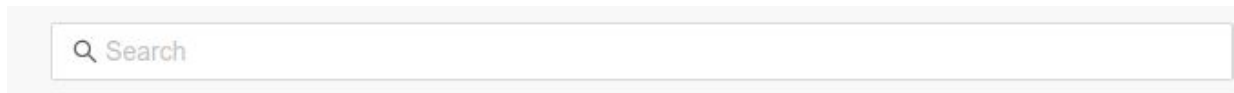
 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
	프로젝트 명	영상 품질 개선 공유 플랫폼	
	팀 명	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

개선된 이미지를 볼 수 있다. 스크롤을 내리면 제작자가 추천하는 영상을 볼 수 있다.

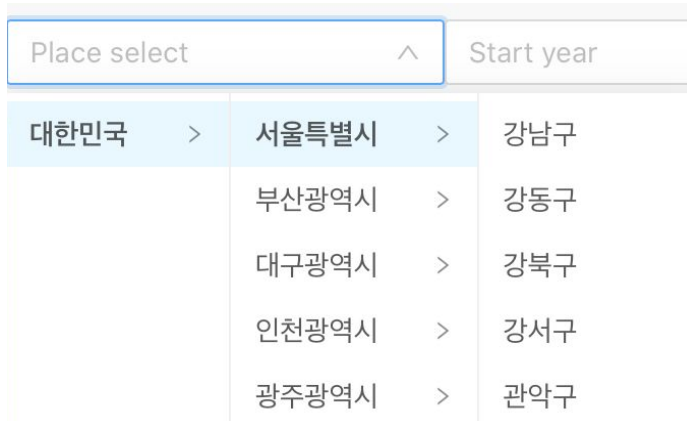
< TopNavbar > - 검색 기능을 구현한 컴포넌트




TopNavbar에선 Home으로 이동할 수 있는 로고와 제목, 장소, 연도를 검색할 수 있는 SearchBar, 업로드 페이지와 사용자 페이지로 이동할 수 있는 버튼이 있다. 영상을 검색할 때, 3가지 사항을 모두 기입하거나 모두 기입하지 않거나, 별개의 사항만을 기입하더라도 검색할 수 있도록 구현하였다.



Search Input에는 제목을 입력한다. 제목을 입력하지 않고 검색을 하면 장소와 연도 범위의 조건에 맞는 모든 내용의 영상이 검색된다.



Search Place Select에는 검색하려는 영상의 장소를 선택한다. default 값은 대한민국이며 전국의 모든 시, 구를 선택할 수 있다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
	프로젝트 명	영상 품질 개선 공유 플랫폼	
	팀 명	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

Start year

→ End year

Search

2020-2029

2030-2039

2019	2020	2021	2029	2030	2031
2022	2023	2024	2032	2033	2034
2025	2026	2027	2035	2036	2037
2028	2029	2030	2038	2039	2040

Search Year Select에는 검색하려는 영상의 연도를 입력할 수 있다. 범위를 지정하여 시작연도부터 끝 연도까지로 지정할 수 있으며, 예를 들어 시작연도가 1980년, 끝 연도가 2000년이면 1980년~2000년까지의 영상이 검색된다.

< MediaListPage > : 검색한 미디어들을 보여주는 페이지


MEMORIES

Search

대한민국 / 서울특별시

1990 → 2020

Search




IR

국회의사당 1996년

Isaiah Rangel

1996 / 대한민국 서울특별시 영등포구




MR

2000년 서울월드컵경기장 인근

Michael Robinson

2000 / 대한민국 서울특별시 마포구




BB

1995년 가양대교 모습

Brandon Brooks

1995 / 대한민국 서울특별시 마포구




MP

1995년 난지도 인근 모습

Matthew Phillips

1995 / 대한민국 서울특별시 마포구




MB

1996년 청계교가도로 모습

Michael Blanchard

1996 / 대한민국 서울특별시 중구




CW

2017년 서울 모습

Caitlin Williams

2017 / 대한민국 서울특별시

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

SearchBar에서 입력한 title, location, date 정보를 쿼리문을 통해 서버에 전달한다. 입력한 정보에 맞는 title, id, location, date, author, isProcessing 등을 서버에서 받아 온 후, MediaCard(쿼리문을 통해 받아온 정보들을 하나의 형식으로 보여주는 카드)에 전달하여 형식에 맞게 각 카드를 출력한다. 받아온 정보들을 카드 안에 출력하며, 카드들은 페이지 넓이에 맞춰 반응형으로 사용자에게 제공된다.

< MediaDetailPage > : 하나의 미디어를 보여주는 페이지




CM 2014년 청계천 모습  
Christopher Meyers

대한민국 서울특별시 종로구  
2014

delete

2014년 청계천

미디어 리스트 페이지에서 원하는 미디어를 선택하면, 서버에 GraphQL 쿼리문을 통해 선택한 미디어의 정보를 요청한다. 서버에서 정보를 받아와 미디어 디테일 페이지에서 보여준다. 이 페이지에서 미디어를 스트리밍하고 미디어에 대한 설명, 유저 정보 등 미디어의 정보를 자세히 볼 수 있다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

## < UploadPage > : 미디어를 업로드하는 페이지

Click or drag file to this area to upload

Title:

Title

Location:

대한민국

Date:

Select year

Description:

Description

Submit

Click or drag file to this area to upload

Title:

Title

Location:

대한민국

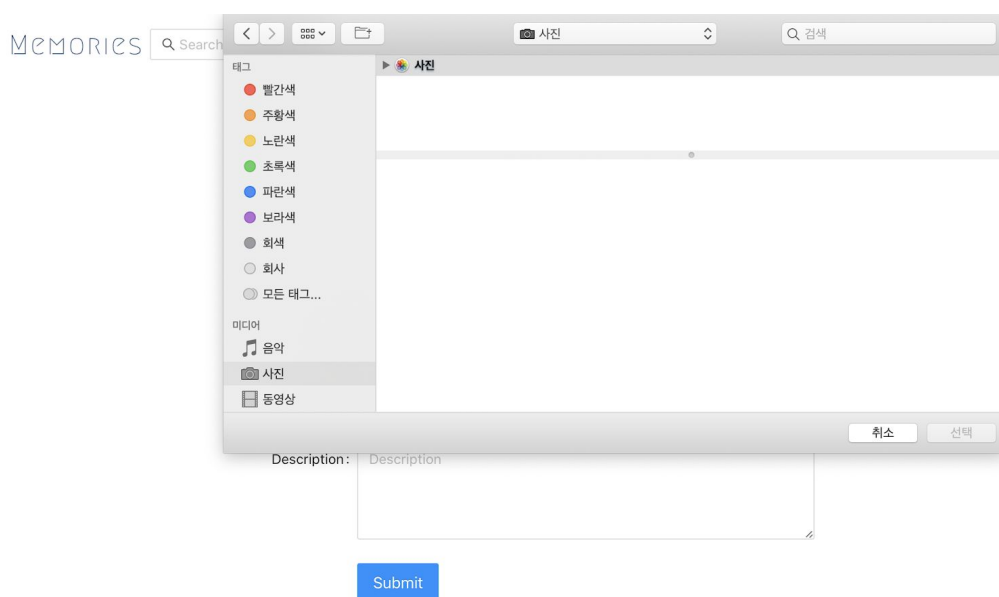
Date:

Select year


Description:

Description

Submit





 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	중간보고서		
	프로젝트 명	영상 품질 개선 공유 플랫폼	
	팀 명	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

test.mp4

Title:

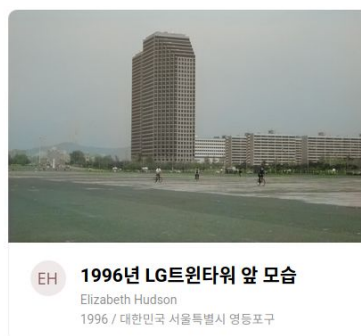
Location:


Year:

Description:

[Submit](#)

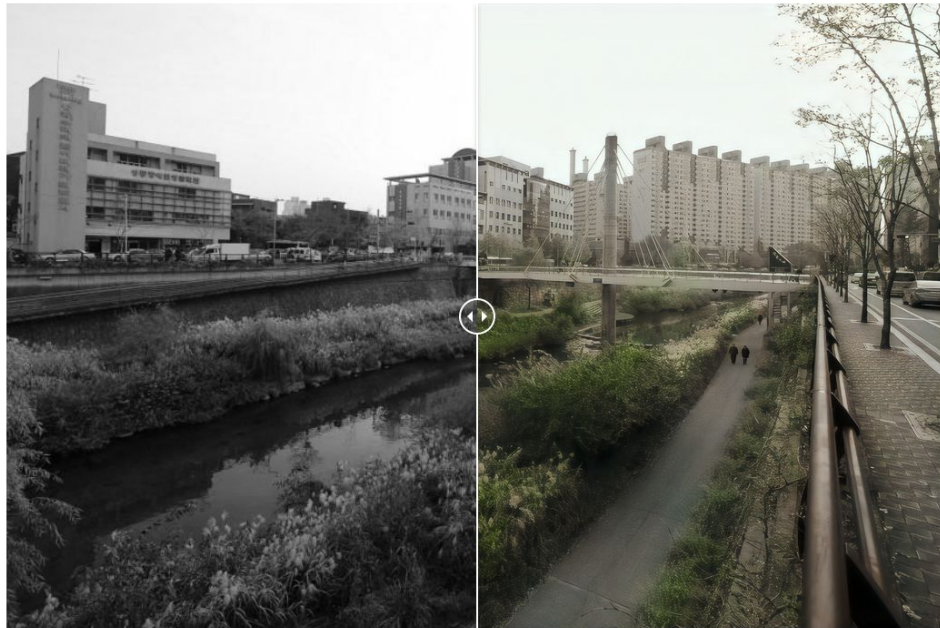
상단 오른쪽에 있는 업로드 버튼을 눌러 추억의 영상을 업로드 할 수 있다. 원하는 영상을 drag 하거나 선택해서 추가하고 제목, 장소 등을 작성해 submit 버튼을 누르면 사용자가 작성한 내용을 GraphQL mutation으로 서버에 보낸다.



 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	영상 품질 개선 공유 플랫폼	
	팀 명	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

### 〈 업로드한 영상(4번째)이 변환중인 사진 〉

변환이 완료되면 리스트 페이지에서 확인할 수 있으며 변환 중이면 다음과 같이 loading으로 처리가 되어 클릭하지 못한다. 변환이 완료되면 클릭하여 영상을 확인 할 수 있다.



NB test  
Nathan Bennett

대한민국 서울특별시 강남구  
2010

delete

test

### 〈 변환 완료한 영상을 클릭했을 때의 미디어 디테일 페이지 예시 〉


#### 2.2.4 결과물 목록 및 진행사항

대분류	소분류	기능	형식	진행 현황
딥러닝 모델	Colorization 모델	Colorization을 수행한다	DNN	진행중
	Frame Interpolation 모델	Frame interpolation을 수행한다.	DNN	진행중
	Super Resolution 모델	super resolution을 수행한다.	DNN	진행중
	서비스 소개	서비스를 소개한다.	웹페이지	진행중





웹 앱	동영상 스트리밍	동영상을 스트리밍한다.	웹페이지	완료
	로그인 & 회원가입	회원가입 및 로그인을 수행한다.	웹페이지	진행중
	영상 업로드	원하는 영상을 업로드하고 서버에 전송한다.	웹페이지	완료
	영상 검색	원하는 영상을 검색한다.	웹페이지	완료
	영상 뷰	전체 영상 또는 조건에 맞는 영상들을 보여준다.	웹페이지	완료
	영상 삭제	원할 때에, 자신이 올린 영상을 삭제할 수 있다.	웹페이지	완료
	사용자 페이지	자신이 올린 영상을 볼 수 있다.	웹페이지	진행예 정
	영상 관리 기능	사용자가 영상을 관리할 수 있는 기능을 제공해야 한다.	함수	진행중
	회원 가입	회원 정보를 받아 DB에 저장한다.	함수	진행예 정
	로그인	회원 정보를 받아 로그인시킨다.	함수	진행예 정
	검색 기능	사용자가 검색하려는 메타 정보를 받아 적절한 검색 결과를 반환한다.	함수	완료
	댓글 게시	사용자가 댓글을 게시하면 DB에 저장한다.	함수	진행예 정
서버	웹 앱 전달	사용자가 웹사이트에 접속하면 웹 앱을 서빙한다.	함수	진행중
	영상 전처리	영상이 들어오면 전처리를 한다.	함수	진행중
	영상의 소리 분리	영상을 편집하기 전에 영상에서 소리를 분리한다.	함수	진행중
	영상의 소리 합성	완성된 영상에 소리를 합성한다.	함수	진행중
	영상 품질 변환	딥러닝 모델을 수행하여 업로드한 영상의 품질을 향상시킨다.	함수	진행중
	영상 저장	품질 향상된 영상을 AWS S3에 저장한다.	함수	진행중

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

## 3 수정된 연구내용 및 추진 방향

### 3.1 수정사항

#### 1. 품질 향상 모듈 실행 순서 고정


계획서상에서는 사용자가 업로드한 영상을 전처리 과정에서 판별하여, 조건에 맞게 모듈을 적용하는 것이었다. 만약에 흑백 영상이라면 Colorization 모듈을 적용하고, 60fps미만이라면 Frame Interpolation을 적용하고, 해상도가 1,080p미만이라면 Super Resolution 모듈을 적용하게 된다. 이렇게 조건별로 모듈을 적용하려고 할 때 영상을 전처리 과정에서 판별해준 뒤, 조건에 해당하는 모듈에 넣어줘야 하는 복잡성이 있다. 앞서 언급한 방식으로 구현할 경우 고려해야 할 조건이 27가지였다. 그리고 각 모듈의 입력을 살펴봤을 때, 실행 속도와 같이 각기 다른 속성을 가지고 있다는 것을 고려하여, 모듈의 실행 순서를 고정했고, 영상의 속성을 구분하지 않고 Colorization, Video Frame Interpolation, Super Resolution의 순서로 3가지 모듈을 적용하는 것으로 하였다.

#### 2. 품질 향상 모듈에서 데이터 저장 포맷 규칙 적용

3가지 모델들을 순서대로 거치다 보니 입력값을 모아놓은 폴더와 출력값을 모아놓은 폴더를 규칙으로 만들어야 한다. 우선 전처리 과정을 거쳐 영상을 흑백으로 변환한 영상 프레임들 {fps}\_{00000000}.png 이름으로 순서대로 color\_input 폴더에 저장하며 영상의 소리는 sound\_input 폴더에 저장한다. Colorization에서 나온 출력값들은 VFI\_input 폴더에 저장한다. Video Frame interpolation에서 나온 출력값들은 SR\_input으로 저장되며 Video Super Resolution에서 나온 출력값들은 model\_output에 저장하도록 규칙을 정했다.

#### 3. 품질 향상 모듈의 입력 데이터 제한

이미지가 아닌 비디오이기 때문에, 프레임의 개수가 비디오의 길이에 따라 대폭 증가한다. 1분 길이의 30fps 비디오라도 프레임으로 바꾸면 1,800개의 프레임이 생성된다. 게다가 Video Frame Interpolation을 거치게 되면 프레임 수는 더 늘어나게 되는데 이를 모델에서 처리하려면 변환하는데 시간이 많이 필요하다. 현재 확실한 기준을 정하지 않았으나, 추후 영상의 길이를 2분 미만 또는 3분 미만으로 입력 비디오의 길이를 제한할 예정이다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

#### 4. 업로드 디폴트 내용 변경

기존 계획은 웹에서 사용자가 업로드를 할 때 영상을 촬영한 장소와 시기를 무조건 입력을 해야만 한다는 것이었다. 하지만 사람들이 예전에 찍었던 정보를 정확하게 모두 기억하는 건 어려움이 있다고 판단을 하였다. 따라서 장소는 대한민국만 입력을 해도 되고, 8도 선에서 기억을 한다면 대한민국 서울특별시, 대한민국 경상북도 등으로 설정을 해서 업로드를 할 수 있도록 수정했다. 또한 시기도 ‘년-월-일’의 형식에서 연도만 입력하면 업로드가 가능하도록 수정했다.

#### 5. API 서버에 사용되는 프로그래밍 언어를 임시적으로 변경


전처리와 딥러닝 모델을 사용하는 품질 향상 모듈을 쉽게 호출하기 위해 Python으로 서버를 만들고자 Flask와, Aridane라는 GraphQL Server 프레임워크를 사용하여 클라이언트가 정보를 요청할 수 있는 API 서버를 구성하였다.

#### 6. 베타버전 클라우드 배포 연기

베타 버전에서는 컨셉의 가능성을 실험하고 Baseline을 구축하고 사용자가 서비스를 사용할 수 있도록 웹 서비스를 구축하는 것에 집중했다. 현재는 인공지능 연구실의 서버를 이용하여 배포하고 있다. 추후에 서비스를 클라우드로 배포하고 공개할 예정이다.

#### 7. 이미지 품질 향상 모듈 추가

처음에는 비디오만을 한정지어서 계획을 세웠기에 예정대로 진행하였다. 그러나 품질 향상 모듈을 적용하는 중에 비디오를 프레임으로 나눠서 입력으로 넣기 때문에 이미지도 가능하겠다는 의견이 나왔다. 이미지는 Frame Interpolation을 사용하지 않고 Colorization과 Super Resolution만 적용하기 때문에 파이프라인이 비디오보다 조금 더 간단하다. 현재 진행상황으로는 이 이미지 품질 향상 모듈을 추가하여, 이미지 업로드와 이미지 품질 향상 후 변환이 가능한 상태이다. 비디오는 현재 품질 향상 모듈 적용은 완료하였지만, 품질이 저조하고 시간이 오래 걸리는 문제가 있다. 추후에 분산처리나 모델 경량화 등등 모듈 최적화를 통하여 비디오 품질 향상 모듈의 성능을 개선시키고자 한다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	영상 품질 개선 공유 플랫폼	
	<b>팀 명</b>	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

### 3 향후 추진계획

#### 3.1 향후 계획의 세부 내용

현재까지 미디어 품질 향상 모듈의 Baseline을 구현하였으며, 사용자들이 서비스를 사용할 수 있는 웹 서비스를 개발하여 품질 향상 모듈과 연결하였다. 웹서비스에는 인트로, 검색, 업로드, 삭제 등의 기능이 구현되어있으나, 향후 댓글 기능과 유저 시스템(가입, 유저 페이지)등을 추가하려한다. 또한 미디어 품질 향상 모듈 Baseline을 구현하였으나 품질이 기대보다 저조하여 다음과 같은 계획을 갖고 향후 일정을 진행하려고한다.

##### 1. Training Dataset 수집

현재 각 모델의 사전 훈련(Pre-Trained) 파라미터를 사용하고 있는 데, 이젠 우리만의 훈련 데이터셋을 수집하여 각 모델을 직접 훈련을 시켜 성능이 향상될 수 있도록 계속해서 실험해보려한다.

##### 2. 여러 연구들을 참고해서 실험 및 개선

각 기술당 두가지 정도의 논문을 참고하고 실험해보았는데 좀 더 다양한 연구들을 참고해서 실험을 진행해보고 더 발전시켜 성능을 향상시킬 예정이다.


##### 3. 또한 품질 향상 모듈의 각 작업 실행 순서를 다양하게 바꿔서 실험해보려한다.

또한 품질 향상 모듈의 속도가 느리다. 현재는 360p, 1분, 30fps의 원본 영상 기준으로 품질 향상에 180분이 소요될 것으로 예상된다. 변환 시간을 줄일 필요가 있기에 향후에는 병렬처리, 모델 경량화를 실시해보려한다.

뿐만아니라 모델 품질 향상 모듈과의 수월한 연결을 위해 파이썬으로 빠르게 만들었는데, 서버를 JavaScript기반의 기술스택으로 Rebuilding하고 클라우드 서비스에 올릴 예정이다.

### 4 고충 및 건의사항

- 각 팀별로 주어진 발표 시간이 부족해서 프로젝트에 대해 충분히 설명하지 못한 것 같습니다. 시간이 충분하다면 더 상세하게 저희가 해온 노력을 설명할 수 있을 것 같습니다.
- 개발에 필요한 하드웨어의 성능이 부족하여 더 좋은 하드웨어가 있으면 좋겠습니다.

 국민대학교 소프트웨어학부 캡스톤 디자인 I	중간보고서		
	프로젝트 명	영상 품질 개선 공유 플랫폼	
	팀 명	Memories	
	Confidential Restricted	Version 0.6	2020-APR-21

## 5 참고문헌

번호	종류	제목	출처	발행년도	저자
1	논문	EDVR: Video Restoration with Enhanced Deformable Convolutional Networks	<a href="https://arxiv.org/abs/1905.02716">https://arxiv.org/abs/1905.02716</a>	2019.05.07	Xintao Wang, Kelvin C.K. Chan
2	논문	Depth-Aware Video Frame Interpolation	<a href="https://arxiv.org/abs/1904.00830">https://arxiv.org/abs/1904.00830</a>	2019.04.01	Wenbo Bao, Wei-Sheng Lai
3	논문	Fully Automatic Video Colorization With Self-Regularization and Diversity	<a href="http://openaccess.thecvf.com/content_CVPR_2019/html/Lei_Fully_Automatic_Video_Colorization_With_Self-Regularization_and_Diversity_CVPR_2019_paper.html">http://openaccess.thecvf.com/content_CVPR_2019/html/Lei_Fully_Automatic_Video_Colorization_With_Self-Regularization_and_Diversity_CVPR_2019_paper.html</a>	2019	Chenyang Lei, Qifeng Chen