창직종합설계프로젝트1

B611155 이유진

B511064 박성춘

B511157 이현수

B511170 장형주

<functional requirement list>

1. 이미지, 동영상 촬영(또는 가져오기) 후 어플에 업로드

   1.1 사용자는 찍은 고화질 이미지 그대로 어플에 업로드

2. Key frame extraction

   2.1 이미지인 경우는 2번 생략하고 3번으로 넘어감

   2.2 (1.2 usecase)에서 받은 동영상은 프레임별로 이미지 생성 (인풋 데이터가 동영상이므로 클러스터링 할 때 너무 오래 걸린다고 생각되면 동영상 자체를 인코딩해서 해상도를 낮춘다.)

   2.3 프레임 간 유사도를 측정하여 clustering을 함.

   2.4 각 클러스터의 frame 수가 전체 frame 수의 10%이상인 클러스터에서만 key-frame 추출 후 서버로 전송

3. Instance mask

   3.1 (2.4 usecase)에서 전송한 key frame을 받아 각 frame 별로 R-CNN을 이용하여 instance mask 추출 -> masked images

4. Insta-GAN

   4.1 (3.1 usecase)에서 받은 원본 이미지에서 뽑은 masked image와 원본 이미지 세트를 입력으로 넣어 사진 변환

5. Cartoon-GAN

5.1 (4.1 usecase)에서 받은 이미지를 만화풍으로 변환하여 어플로 전송

6. Make cartoon

6.1 5.1 usecase 에서 넘어온 결과 이미지의 개수에 해당하는 layout(1~10개의 layout)을 사용자에게 보여주고 고르게 함

6.2 사용자가 고른 layout에 맞춰 이미지를 넣고 만화를 만들어서 사용자에게 보여줌과 동시에 말풍선 list도 보여줘서 원하는 대사를 넣을 수 있게 함

<Non-functional requirement list>

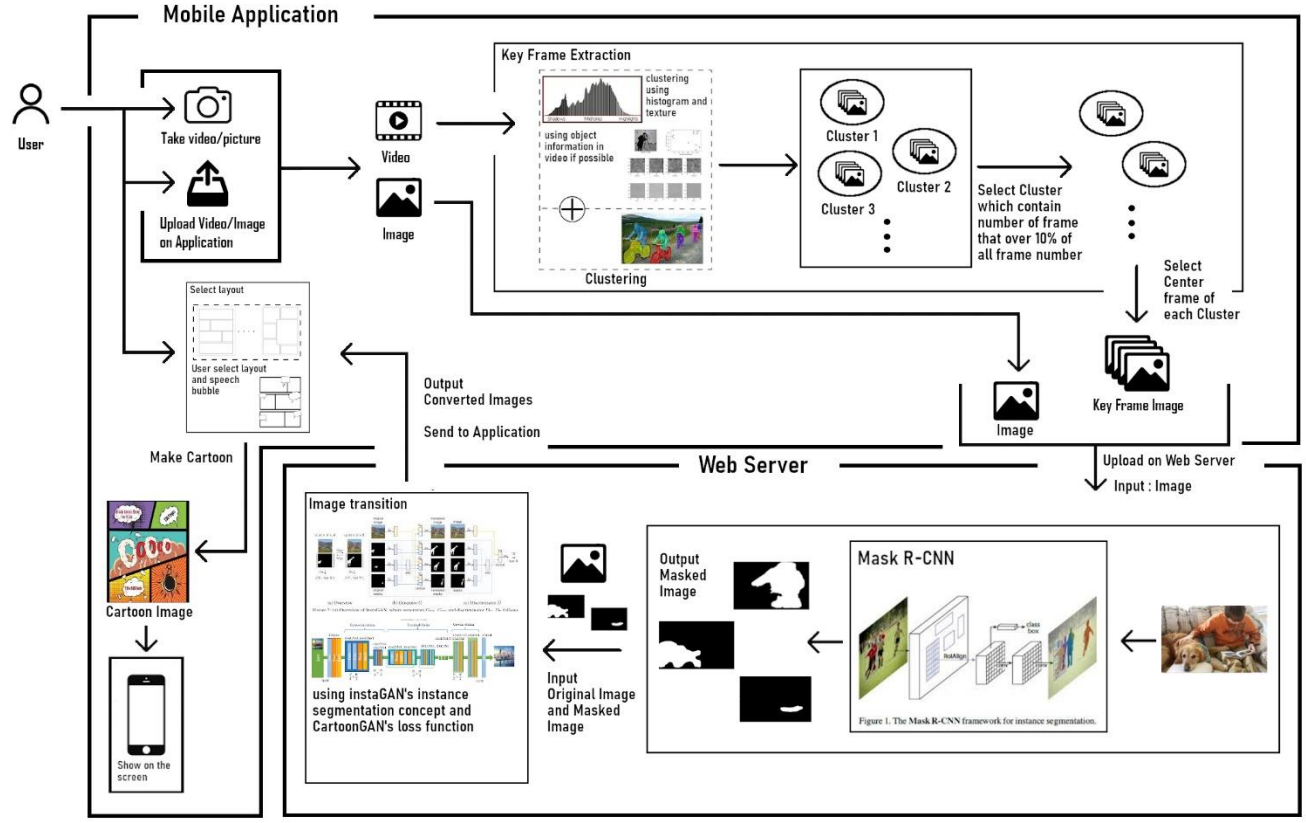
1. Insta-GAN에서 인스턴스 모양을 바꿀 때 사람이 바꾸고 싶은 인스턴스(코끼리)와 결과 인스턴스(기린)를 일일이 정해줄 수 없음. (논문에서는 테스트 이미지가 한 장이므로 바꾸고자 하는 마스크를 넣어줄 수 있지만 본 프로젝트는 어플이 여러 장을 자동으로 바꾸어야 함)

   → 개발할 때 모델에 미리 학습시켜 놓은 몇가지의 만화풍.

   → 사진 유형에 관계 없이 모델이 학습한 만화풍으로 변환할 수 있음.

2. 어플에 컷 개수별로 layout과 말풍선을 넣어둬야 함

<구체화한 시스템 도식화>


Mobile Application
User
Take video/picture
Upload Video/Image on Application
Video
Image
Key Frame Extraction
clustering using histogram and texture
using object information in video if possible
Clustering
Cluster 1
Cluster 2
Cluster 3
Select Cluster which contain number of frame that over 10% of all frame number
Select Center frame of each Cluster
Key Frame Image
Image
Select layout
User select layout and speech bubble
Output Converted Images
Send to Application
Make Cartoon
Web Server
Upload on Web Server
Input : Image
Image transition
using instaGAN's instance segmentation concept and CartoonGAN's loss function
Input Original Image and Masked Image
Output Masked Image
Mask R-CNN
Figure 1. The Mask R-CNN framework for instance segmentation.
Cartoon Image
Show on the screen

<개발 환경>

구분		항목	세부내용
S/W 개발 환경	OS	Windows /Linux	Frontend, Backend 개발
	개발 환경 (IDE)	Android studio	Backend, IOS, Android 어플 개발 IDE/ Editor
		Jupyter notebook	Python 딥러닝 모델 개발 IDE
		Pycharm	
	개발 도구	Source tree	효율적인 git관리를 위한 프로그램
		AZURE	프로젝트 운영 서버
	개발 언어	python	딥러닝 모델 개발 언어
		Kotlin	어플리케이션 개발 언어

<시뮬레이션 계획>

구분	추진내용	추진일정									
		2 월	3 월	4 월	5 월	6 월	7 월	8 월	9 월	10 월	11 월
계획	관련 분야 조사										
	논문 분석										
분석	개발 환경 설정										
	개발 일정 계획										
설계	서버 기능 설계										
	어플리케이션 설계										
	모델 설계										
개발	어플 제작										
	이미지 추출 모델 구현										
	마스크 추출 모델 구현										
	이미지 변환 모델 구현										
	만화풍 변환 모델 구현										
테스트	성능 검증을 통한 시스템 고도화										
	원활한 시나리오 테스트										
종료	최종 결과물 및 보고서										

시뮬레이션 계획

고려할 사항들

1. Train Data Set 준비
2. 다음의 알고리즘들 개별 성능 평가
    - Key Frame Extraction.
    - Mask R-CNN
    - InstaGAN
    - CartoonGAN

1. Train Data Set 준비

60분 기준 동영상의 프레임 수 => 60*60*30=108,000프레임 이미지 추출 가능.

1초당 한프레임 추출해서 Train Data로 사용시 4,800프레임 이미지 추출 가능

만약 원하는 만화풍의 Train 이미지가 부족할 시 좌우반전 이미지를 사용하여, 부족한 Train Data 보충.

CartoonGAN의 경우 논문에서 약 8,000장의 이미지를 사용한 것으로 나와 있어, Train Image가 8000장 정도 넘기도록 동영상에서 프레임 추출해서 사용.

2. 다음의 알고리즘들 개별 성능 평가

- Key Frame Extraction

    Key Frame Extraction 논문에서도 나와 있듯 알고리즘의 성능 평가를 객관적으로 실시하기엔 어려움. 따라서 테스트 input video로 나온 output frame이 Key Frame인지 주관적으로 사람이 평가함.

- Mask R-CNN

    사람 인스턴스를 마스크로 세그멘테이션하는 모델을 학습시켰다고 할 때, Test Data Set으로 사람이 포함된 Image Set과 사람이 포함되지 않은 Image Set으로 테스트하여, 사람 인스턴스를 제대로 세그멘테이션하는지 확인. 이후 다양한 사람이 포함된 Image Set으로 테스트하여 세그멘테이션 성능 테스트함.

- InstaGAN

  Mask R-CNN 성능평가 후, 결과가 좋을 때 무작위 이미지로 마스크 세그멘테이션과 input image를 input으로 하여, 성능 테스트함. Content Loss에 가중치 변수가 3개 있는데, 3개 값 조절해가면서 성능을 테스트함.

$$\mathcal{L}_{\text{InstaGAN}} = \underbrace{\mathcal{L}_{\text{LSGAN}}}_{\text{GAN (domain) loss}} + \underbrace{\lambda_{\text{cyc}}\mathcal{L}_{\text{cyc}} + \lambda_{\text{idt}}\mathcal{L}_{\text{idt}} + \lambda_{\text{ctx}}\mathcal{L}_{\text{ctx}}}_{\text{content loss}},$$

- CartoonGAN

  프로젝트에서 가장 중추가 되는 딥러닝 모델로 성능 테스트가 가장 중요한 부분임.

  위 InstaGAN과 비슷하게 input image를 input으로 하여 성능 테스트함. 최종 Loss에 가중치 변수가 존재하며, 가중치 변수의 값을 조절해가면서 성능을 테스트함.

  CartoonGAN 논문에서 배경사진만 테스트하고, 인물 사진에 관한 테스트는 없었는데, adversarial Loss에서 윤곽선이 제거된 만화 이미지 Ej에 대한 오차를 추가하여 만화스러운 이미지로 변환은 가능하나, GAN자체의 shape 변형엔 취약한 점으로 인물 변환에 있어서 낮은 성능을 보일 것으로 예상됨.

  따라서 위의 성능 테스트 결과가 좋지 못할 때, InstaGAN과 결합하여 인스턴스 shape 변형이 약한 것을 보완할 수 있도록 테스트해 볼 예정.

  또한 위 논문에서는 Adversarial Loss로 기존 Loss에 윤곽선이 제거된 만화 이미지 Ej에 관한 오차 함수를 추가했는데, output의 결과를 높일 수 있는 추가적인 오차 함수를 정의할 수 있으면, 추가하여 추가 성능 테스트를 실시해볼 예정임.