

Your Cartoon Is: Real Video to Cartoon

이현수, 장형주, 박성춘, 이유진

Computer Engineering, Hongik University,
Seoul, Korea

ABSTRACT

딥 러닝을 이용한 영상 및 이미지 분석과 변형에 대한 연구는 나날이 발전하고 있다. 시각적 데이터를 어떤 목적으로 다루는지에 따라 다양한 딥러닝 모델을 만들 수 있다. 본 논문에서는 영상으로부터 사용자가 원하는 이미지만을 추출하고, 추출한 이미지의 화풍을 원하는 형태로 변환한다.

이를 위해 yolo 모델을 이용해 object detection을 수행하여 영상에서 원하는 object가 포함된 이미지를 뽑아내고, 변환한 Cartoon-GAN 모델로 이미지를 만화풍으로 변환하는 과정을 수행한다. 결과는 기존의 Key-Frame Extraction을 사용했을 때와 본 논문에서 제시하는 방법을 사용했을 때 뽑히는 이미지를 비교한다. 또한 만화 같은 이미지를 얻기 위해 Cartoon-GAN을 변경시킨 과정과 그에 따른 최종 결과를 보인다.

1. INTRODUCTION

딥 러닝이 발전함에 따라 다양한 문제를 딥 러닝으로 해결하려는 노력이 이어지고 있다. 그 중 영상과 이미지 분석 및 변형에 관한 연구 또한 꾸준히 발전하고 있다. 이 분야에서는 특히 적대적 생성 신경망인 GAN [8]이 많이 이용되고 있다. 요즘 큰 문제로 대두되고 있는 Deep-Fake 또한 GAN을 이용한 결과이다. GAN은 생성모델을 통해 진짜같은 가짜를 만들어내며, 판별 모델을 통해 생성자가 만들어낸 가짜와 진짜를 구분하면서 적대적으로 학습하고, 이를 통해 정확도를 향상시킨다. 이와 같은 GAN의 특성은 이미지 분류에서 데이터가 부족한 경우에도 효과적인 해결책을 제시한다. [5] 영상, 이미지 분류 및 변환에서 기능을 추가하여 더 좋은 성능을 내는 파생 GAN에 대한 연구가 활발히 진행되어왔다. 구체적인 예로, 정답 레이블에 해당하는 이미지를 따로 학습시키지 않아도 이미지 변환을 할 수 있는 cycle-GAN [9]과 이미지의 화풍을 변화시키는 pix-to-pix [3] 등이 있다.

이처럼 GAN을 이용하여 효과적인 데이터 변환을 위해 본 논문에서는 만화적 이미지를 생성하는 것에 특화된 cartoon-GAN [1]에 기초하여 동영상에 컷 톤 형식으로 만들어보고자 했다. 동영상에서 스토리가 있는 만화를 자연스럽게 만들기 위해서는 동영상에서 사람이 보기에 의미 있는 프레임을 뽑아야 하고, 뽑은 프레임을 만화풍으로 바꾸기 위해서는 만화적 특징을 살리는 것이 중요하다.

본 논문에서는 사용자가 만화로 바꾸고자 하는 동영상은 자연환경만이 담겨있는 영상보다는 사람 객체가 나오는 동영상일 것이라는 가정에 기초하여 동영상에서 사람이 있는 컷을 중점적으로 뽑기 위해 object detection을 수행한 뒤 object가 있는 컷에 가중치를 두어 key-frame을 추출하였다. 추출한 frame에 사용자가 선택한 만화를 나타내는 스티커를 붙인 후 Cartoon-GAN에 넣어 윤곽선을 강조한 만화로 변환한다. Cartoon-GAN 모델은 참고한 논문에서 손실함수의 가중치와 hyperparameter를 변경하여 원하는 결과를 얻었다. 전체 과정을 어플리케이션으로 구현하여 편리하게 사용할 수 있도록 하였다.

2. RELATED WORK

2.1 Key Frame Extraction

동영상에서 하이라이트 이미지를 추출해내기 위한 방법을 제시한다. 6가지의 Key-Frame Extraction 방법 중 해당 논문은 Clustering을 사용하는 방법을 제시한다. Clustering을 사용한 Key Frame Extraction은 구현이 비교적 쉽고, 속도가 빨라 실시간으로 적용할 수 있다는 이점이 있다.

Key Frame Extraction 과정은 다음과 같다. 우선 Global Level Histogram을 사용하여 프레임 간의 색

상 유사도를 측정하고, GLCM(Grey-Level Co-occurrence Matrix)을 사용하여 프레임 간의 텍스처(질감) 유사도를 측정한다.

최종적으로 두 유사도 측정값을 가중치를 사용하여 최종 유사도 측정값으로 통합한다. 이 최종 유사도 측정값을 임계 값과 비교하여 Cluster의 밀집도를 제어한다. 최종 유사도 측정값이 임계 값보다 낮다면 이는 기존 Cluster에 추가되기에는 유사도 값이 낮다는 것을 의미하므로, 새로운 Cluster를 생성한다.

Clustering이 끝나면, Cluster들 중 각 Cluster가 갖는 Frame 수가 전체 Frame 수의 10%가 넘으면 해당 Cluster가 비중 있는 Cluster라고 판단하여 해당 Cluster의 대표 Frame을 Key Frame으로 선택한다. 해당 Cluster의 대표 Frame은 Cluster 내의 Frame의 HSV와 GLCM의 중심에 가장 가까운 Frame을 Key Frame으로 선택한다.

2.2 Cartoon-GAN

실제 이미지를 만화 스타일의 이미지로 변환하는 것을 목표로 하는 모델로, GAN 모델을 기초로 한다.

만화 스타일의 이미지는 실제 이미지와 다른 고유한 특성이 존재하며, 현재 널리 사용되는 GAN 모델의 손실함수로는 이를 처리하기 힘들다. Cartoon-GAN에선 이를 위해, 기존 손실함수를 기반으로 윤곽선에 대한 손실도를 포함하는 손실함수를 정의하여 해당 모델의 성능을 높였다.

추가적으로, Cartoon-GAN에서는 기존의 GAN 모델이 비선형이기 때문에, 무작위로 초기화할 경우 지역 최소점에 갇혀 원하는 방향으로 학습되지 않을 가능성이 커, 이를 방지하기 위해 Semantic Content Loss를 사용하여 사전에 생성자를 학습하였고, 이것이 결과에 유의미한 차이가 있음을 보였다.

2.3 YOLO (you look only once)

YOLO는 본 논문에서 Key-Frame Extraction 시에 이미지 속의 Object를 탐지하기 위해 사용하는 모델로, Object를 탐지할 때 전체 이미지를 보고 맥락적인 정보를 Encoding하여 Background Error가 적으며, Object의 일반적인 특징을 학습하기 때문에 새로운 Input 이 들어와도 망가질 확률이 적다. 또한 Detection을 단일 회귀 문제로 정의하기 때문에 속도가 매우 빠르다.

3. MODEL STRUCTURE

3.1 전체 프레임워크 설명

Figure 3-1은 전체 시스템의 구조이다. 사용자가 어플리케이션에 동영상상을 업로드하면 동영상에서 프레임을 뽑아 이미지를 만들고 그 프레임속의 색깔과 명암으로 유사도를 측정하여 클러스터링을 한다. Object 가중치를 고려하여 클러스터를 대표하는 Key-Frame을 뽑고 이렇게 뽑은 Key-Frame을 선택한 만화에 맞게 꾸민 후 서버로 전송하면 화풍을 변화시키는 모델인 Cartoon-GAN에 입력으로 넣어 사진을 변환한다. 이렇게 변환된 이미지를 다시 어플리케이션으로 보내 사용자가 고른 레이아웃에 맞추어 만화로 만든다.

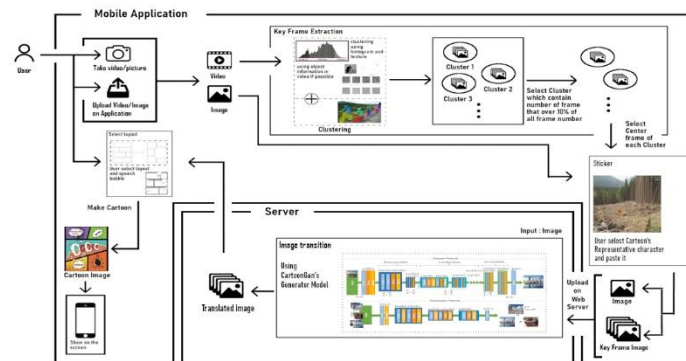


Figure 3-1. 전체 구상도

3.2 Unsupervised Clustering을 사용한 Video key Frame Extraction

Key Frame Extraction을 위해 Clustering을 이용한다. Video의 서로 다른 Frame $\{f_1, f_2, f_3, \dots, f_N\}$ 들을 각 frame들 사이의 공통된 color bins수와 GLCM정보를 활용하여 유사도 측정 서로 다른 M개의 Cluster $\{C_1, C_2, C_3, \dots, C_M\}$ 로 분류하였다 [4]. 유사도 측정은 기존 논문의 이미지의 Color bin과 GLCM texture를 사용하는 $Combine_{simi}$ 을 사용하였다. [4]

또한 만화라는 스토리 유지를 위해 모든 cluster와 유사도를 측정하는 기존의 clustering 방식과는 다르게 새로운 frame을 직전 시간대의 cluster와 비교를 진행하게 함으로써 비슷한 장면이 다른 시간대에 등장하더라도 그 정보가 유지되도록 구현하였다.

Clustering 방식만을 사용하면 key Frame들이 촬영된 시간에 의존한다는 문제가 있다. 이러한 의존도를 낮춰주고 원하는 장면들을 많이 뽑아 내기 위해 object detection을 추가하였다.

Figure 3-2는 object detection이 추가된 key frame extraction을 보여준다. Clustering 방식으로 분류해낸 Frame들에서 각 Cluster의 중심 frame을 Object Detection 모델에 넣어 object label을 뽑아낸다. 이후 (1)의 수식을 이용하여 Key Frame을 뽑아내는 구조이다.

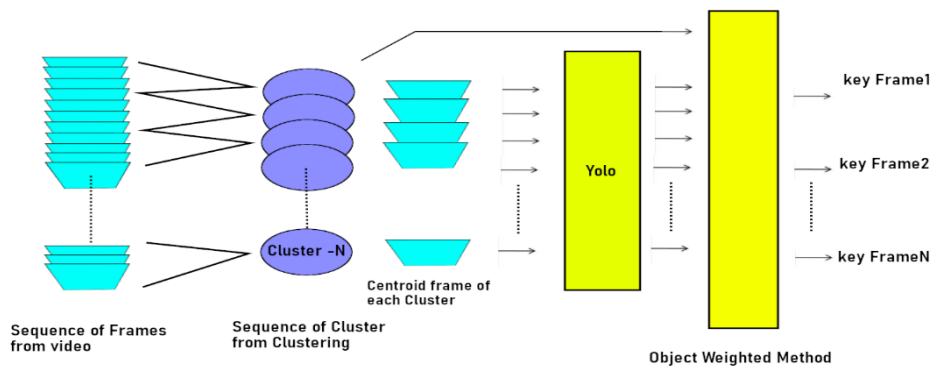


Figure 3-2. Object Detection을 적용한 Key Frame Extraction

key Frame을 뽑는 기준에 사용할 식은 다음과 같다.

$$\alpha_K = N_k + \sum W_K(i) \quad (1)$$

N_k 은 k번째 Cluster 안의 frame수를 나타내며 $W_K(i)$ 는 k번째 Cluster의 대표 Frame에 있는 object 중 i번째 object의 가중치를 나타낸다. 따라서 α_K 는 k번째 Cluster 안의 Frame의 수와 대표 Frame의 모든 object들의 가중치를 합한 값이다. W는 여러 번의 실험을 통해 적절한 값을 찾아야 한다. 본 논문에서는 ‘사람’에 가중치를 두었으며 $W = 3$ 정도로 진행하였다.

α_K 가 특정 임계 값(전체 frame 수 * δ)을 넘으면 해당 Cluster의 대표 Frame을 Key Frame으로 뽑아낸다. δ 는 key frame extraction을 진행하며 실험적으로 적절한 값을 구하여야 한다. 퍼센트 기준 값이 너무 낮으면 중복된 frame이 많이 나오며 퍼센트 기준 값이 너무 높으면 원하는 frame이 나오지 않을 수 있다 [4]. 본 논문에서는 $\delta = 0.035$ 정도로 두었다.

Object detection 모델로는 빠른 속도를 위해 tiny-YOLOv4(You Only Look Once)를 사용했으며 COCO(large-scale object detection, segmentation, and captioning dataset)를 이용하여 미리 학습된 모델을 사용하였다. [7].

3.3 Cartoon-GAN

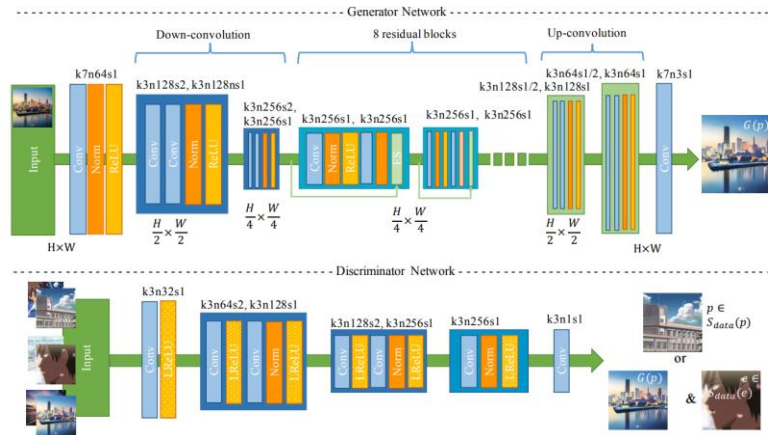


Figure 3-3. Cartoon-GAN 구조도

어플리케이션에서 보낸 하이라이트 이미지들을 사전에 학습된 만화 스타일로 변환하기 위해 사용되는 모델이다. 만화 스타일로 이미지를 변환시키는데 초점을 맞춘 GAN 기반의 모델로, 변환된 이미지가 뚜렷한 윤곽선과 만화 같은 색감을 갖는 것이 특징이다. 기존 GAN의 입력으로 source 이미지와 target 이미지를 갖는 것에 추가로 target 이미지(만화 이미지)에서 윤곽선을 Blur 처리한 이미지 또한 source 이미지에 포함시켜 학습시킨다. 이러한 학습 방법의 결과로 변환된 이미지가 뚜렷한 윤곽선 이미지를 갖도록 할 수 있다. 또 다른 특징으로는 source 이미지와 target 이미지의 특성이 달라 source 이미지의 오브젝트들이 변환되면서 형태가 뭉개지는 것을 방지하기 위해 사전에 학습된 VGG19 net을 사용하여 bottleneck feature를 손실 함수를 계산하는데 사용하여 학습 시 오브젝트의 형태를 잘 유지하도록 하였다.

$$L_{con}(G, D) = \mathbb{E}_{p_i \sim S_{data}(p)} \left[\left\| VGG_1(G(p_i)) - VGG_1(p_i) \right\|_1 \right] \quad (2)$$

또한, 모델의 가중치 값들이 무작위로 initialize된 특성으로 인하여 제대로 학습되지 않는 점을 해결하기 위하여 학습을 진행하기 전 L_{con} 만을 사용하여 모델의 가중치 값들이 올바르게 initialize 되도록 하여 제대로 학습된 결과를 얻을 수 있었다.

최종 프로젝트에 사용된 모델 학습을 위해, source 이미지 데이터 셋으로 kaggle의 Flickr 30k Dataset 중 15,000장을 학습에 사용하고, target 이미지 데이터 셋으로 애니메이션 ‘톰과 제리’ 영상에서 프레임 추출하여 약 19,000장을 학습에 사용하였다.

또한, Cartoon-GAN 학습에 추가로 포함되는 데이터 셋인 윤곽선을 Blur 처리한 이미지를 사용하기 위해 target 이미지를 Canny Edge Detection을 사용하여 윤곽선을 검출하고, 검출된 윤곽선을 5x5 Gaussian kernel을 사용하여 윤곽선을 Blur 처리하여 데이터 셋으로 사용하였다.[2]

3.4 Application

사용자의 편의성을 위해 전체 과정을 어플리케이션으로 구현하여 접근성을 높였다. 어플리케이션으로 서비스하는 과정에서 변환 시간을 줄이기 위해 어플리케이션에서 Key Frame Extraction 과정을 구현하였다. 어플리케이션은 크게 이미지 한 장을 변환하는 것과 영상을 이미지로 나누어 변환하는 과정의 두가지 방식으로 나뉜다.

첫번째 방식인 이미지 한 장을 변환하는 과정은 다음과 같다. 사용자가 앨범에서 이미지를 업로드하고 Socket 통신을 이용하여 서버에 전송을 한다. 서버에서 받은 이미지를 변환시키고 결과 이미지를 다시 App으로 반환한다. 사용자는 반환된 이미지에 드래그 앤 드롭 방식으로 스티커를 추가할 수 있다.

두번째 방식인 영상을 변환하는 과정은 다음과 같다. 사용자가 앨범에서 영상을 업로드한다. 영상

이 업로드가 되면 App에서는 영상을 Frame별로 나누어 C++로 구현된 Key-Frame Extraction을 이용하여 Key Frame을 뽑아낸다. 또한, 어플리케이션에서는 반환된 이미지들을 만화형식으로 편집할 수 있는 기능인 콜라주 기능과 변환된 이미지에 스티커를 추가할 수 있는 기능을 제공한다. 뽑혀진 Key Frame들 중 사용자가 원하는 특정 이미지에 스티커를 추가할 수 있다. 스티커 추가 단계를 마무리하고 난 후, 사용자는 원하는 이미지들을 선택하여 Socket 통신을 이용하여 서버에 전송하고 서버는 이미지들을 순차적으로 변환하여 변환된 이미지들을 반환한다. 다음 단계로 사용자가 특정 콜라주 레이아웃을 선택하여 임의로 원하는 결과 이미지를 드래그 앤 드롭으로 추가할 수 있다.

최종적으로 사용자가 편집한 이미지를 저장할 수 있는 기능을 제공하는 것으로 어플리케이션의 동작은 마무리된다.



Figure 3-4. 이미지 변환 과정(왼)과 영상 변환 과정(오)

4. EXPERIMENT

4.1 Key-Frame Extraction

Key-frame 정의의 주관성으로 인해 key-frame extraction에 이용 가능한 균일한 평가 방법이 없다. 따라서 영상을 선택할 때 특정 object를 확실히 포함하는 영상을 선택한다.

기존의 방법과 차별을 두기 위해 object-detection 방법을 사용하여 특정 object가 있는 프레임에 가중치를 두는 방법을 채택하였다. 이를 평가하기 위해 두 경우의 결과를 비교한다.

실험에 사용되는 동영상은 30 초, 1 분, 2 분의 길이를 가지는 영상들이다. 이 영상들은 본 연구의 목적에 맞게 사람이라는 object가 많이 나온다. 최종적으로 3 가지의 동영상을 가지고 object detection을 적용했을 때와 적용하지 않았을 때의 결과 사진과 시간 차이를 비교한다.

Table 1. 기존 Unsupervised Clustering 과 Ours 의 안드로이드 환경에서의 측정 결과

Video Length	No Object majority.			Object majority is 3.		
	Key-Frame Extraction Time	Total Run Time	The number of pictures	Key-Frame Extraction Time	Total Run Time	The number of pictures
30 sec	2.631s	14.684s	7	5.643s	17.673s	9
60 sec	6.212s	34.241s	11	17.738s	46.273s	12
120 sec	12.386s	59.274s	5	18.661s	65.245s	6

이 논문에서 구현한 Object majority 방식이 Yolo 를 사용하는 부분 때문에 Extraction Time 이 더 길어진다. 하지만 기존 Unsupervised 방식과 다르게 Cluster 를 비교하는 부분에서 시간이 단축되어 영상의 길이가 길어질수록 차이가 좁혀지는 것을 볼 수 있으며 120 sec 이후 더욱 긴 영상을 input 으로 넣었을 때 Object majority 방식이 더 효율적일 것으로 예상된다.

4.2 Cartoon-GAN

추출된 하이라이트 이미지들을 만화 스타일로 변환하기 위해 Cartoon-GAN 모델을 약 5,000장의 Flickr 데이터셋, 약 9,000장의 만화 이미지로 학습한 결과, 발생한 문제점이 있었다. 변환된 결과 이미지 중 만화 스타일로 변환되지 못하고 검게 변하는 현상이 발생했다.

해당 문제의 해결을 위해, 원인이 생성자(Generator)와 판별 자(Discriminator)의 학습 비율이라 판단하여 학습 비율을 1:3으로 변경하여 학습을 다시 진행하였다. 학습 결과 검게 변하는 현상은 발생하지 않았으나 마찬가지로 학습이 제대로 진행되지 못하였다.



Figure 4-1. 학습이 제대로 진행되고 있지 않은 과정

학습 비율이 원인이 아니라고 판단되어, 내부 모델 구조 중 손실 함수 계산에 사용된 VGG19 Net 을 GoogleNet(Inception v3)[3]으로 변경하여 다시 학습을 진행하였으나 문제는 해결되지 않았다. 이를 통해 문제의 원인이 모델 구조가 아닌 학습에 사용된 데이터 셋이라 판단하여 데이터 셋을 변경하는 과정을 거쳤다. 우선 실제 이미지 데이터 셋의 크기를 약 5,000장에서 15,000장으로 만화 이미지 데이터 셋의 크기를 약 9,000장에서 20,000장으로 늘렸다. 또한 윤곽선을 Blur 처리하는 전 처리 과정에서 직접 전 처리된 샘플 이미지의 결과를 확인하며 가장 좋은 결과를 내는 Gaussian kernel의 크기를 5x5를 사용하여 학습을 진행하였고, 해당 문제를 해결하면서 가장 좋은 결과를 내는 모델을 학습할 수 있었다.



Figure 4-2. 학습이 제대로 진행된 이후의 예시

5. CONCLUSIONS

5.1 key-frame extraction 결과

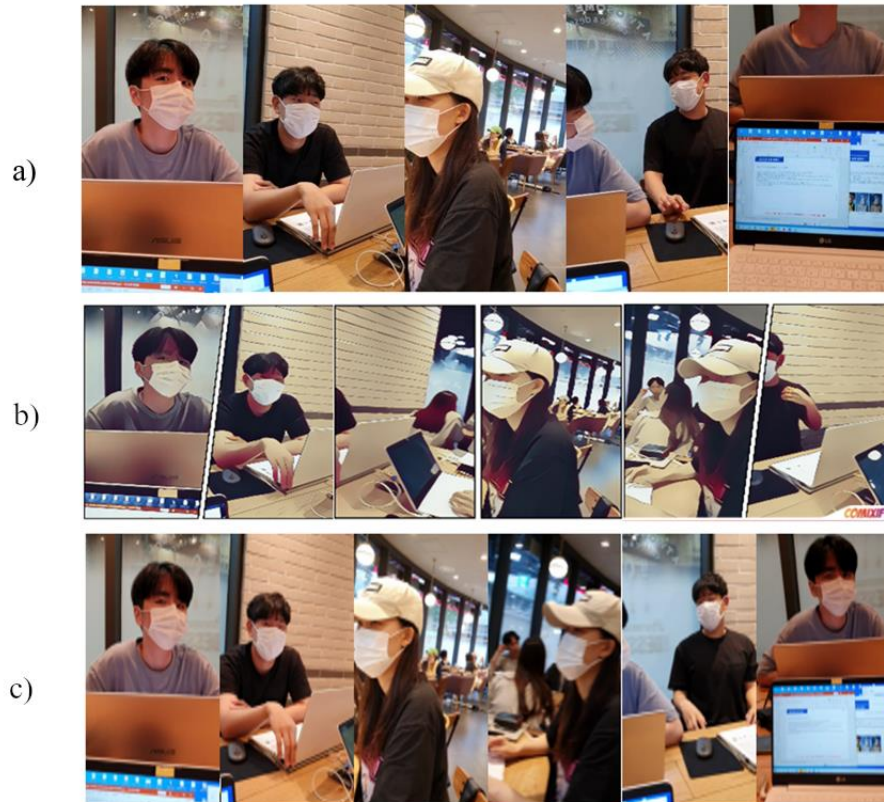


Figure 5-1. a) Unsupervised Clustering, b) Comixify's method, c) Ours

예시 이미지에서 보다시피 객체 가중치 덕에 빠른 장면 전환 도중에도 ‘사람’이 들어간 사진이 잘 나오는 것으로 보인다. 아래의 모자를 쓴 사람의 사진을 보면 서로 유사도의 차이가 있어 서로 다른 Cluster로 분류되었지만 한 장은 장면의 촬영 정도가 일정 수준에 미치지 못하여 기존 방법으로는 추출되지 않은 것을 볼 수 있다. 마지막 사진의 경우 기존방식에서는 맨 앞장의 사진과 유사하여 같은 Cluster로 들어가게 되 사람이 함께 나온 사진은 추출되지 않은 것으로 보인다. 반면 이 논문에서 구현한 Clustering 방식으로는 시간대를 구분하여 유사한 사진도 다시 추출된 것을 볼 수 있다. Comixify's DSN(Deep Summarization network)을 사용한 방식과 비교해 보았을 때에도 비슷한 결과가 나옴을 볼 수 있었다.

5.2 Cartoon-GAN 결과

약 15,000장의 실제 이미지와 약 20,000장의 만화 이미지, 해당 20,000장의 만화 이미지를 Gaussian Kernel을 사용하여 윤곽선을 Blur 처리한 20,000장의 이미지를 사용하여 학습을 진행하였다.

학습을 진행하기 전 3. MODEL STRUCTURE 섹션에서 기술한 Initialization(Pre-train) 과정을 10 회의 Epoch 만큼 진행하였다. Figure 5-2 는 Initialization 과정 중 변환된 이미지들이다. Initialization 과정이 끝난 후 같은 데이터 셋을 사용하여 총 100 회의 Epoch 만큼 학습을 진행하였으나 학습이 끝난 후 검증 이미지를 비교한 결과 Epoch 30 과 50 에서의 학습된 모델이 어플리케이션에 적용 시 가장 좋은 결과를 내어 해당 모델을 프로젝트 서비스 시에 사용하였다. Figure 5-3 은 학습이 끝난 모델의 검증 이미지의 변환 전, 변환 후 이미지와 실제 어플리케이션 서비스 시 변환된 이미지들이다.



Figure 5-2. Initialization(Pre-train) 과정

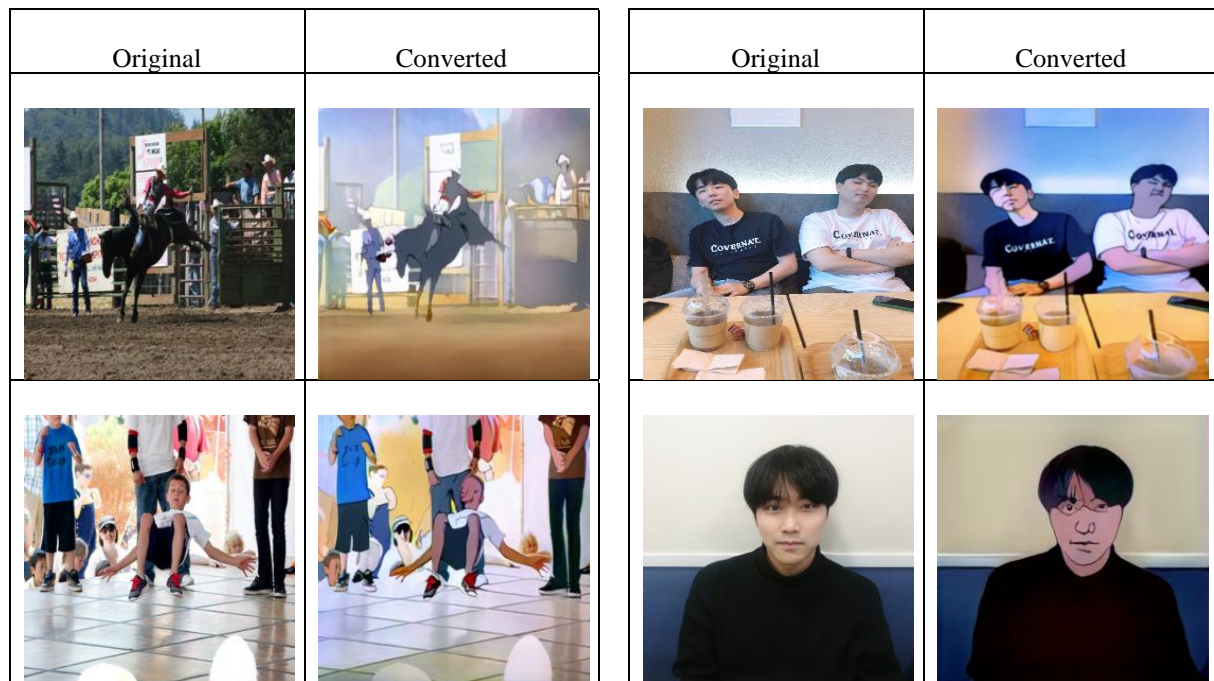


Figure 5-3. 검증 이미지 (원) 어플리케이션 서비스 (우)

학습된 모델을 서비스 시 사용하기 위해 어플리케이션과 소켓 통신을 통해 이미지를 입력 받고 변환 후 다시 어플리케이션에 전송하도록 API 형태로 제공하여 구동될 수 있도록 하였다. 서비스하면서 변환된 이미지들을 통해 몇 가지 결과를 도출할 수 있었다. 첫째로 이미지에 사람(오브젝트)가 한 명만 있거나 이미지가 대체로 단순한 경우 만화 스타일로 잘 변환되었으나 오브젝트가 많아지거나 이미지가 복잡해지는 경우 만화 스타일로 변환되는 정도가 낮았다. 이는 L_{con} 에 사용된 VGG19Net이 해당 이미지에서 오브젝트 탐지 에러로 인한 결과로 추측되며, VGG19Net을 추가로 학습하거나 낮은 에러율을 보이는 네트워크로 구조를 변경하면 해결될 것이라 추측된다.

또한 이미지 변환 시 얼굴의 영역이 작거나 하는 등의 경우 얼굴의 이목구비가 제대로 변환되지 못하고 단일화되어 변환되는 경우가 발생하였다. 이는 모델의 입력 해상도가 256x256으로 상대적으로 이미지의 크기가 작아서 발생한 문제로 보이며, 더 높은 해상도의 이미지를 입력으로 하도록 모델의 구조를 바꾸고, 같은 해상도의 이미지로 학습을 진행하면 해결할 수 있을 것으로 보인다. 현 프로젝트에서는 하드웨어의 한계로 부득이하게 위 해상도를 사용하였다.

5.3 Application 결과

Figure 5-4는 전체 과정을 Application으로 구현하고 실제 동영상을 넣어 구동한 모습이다.

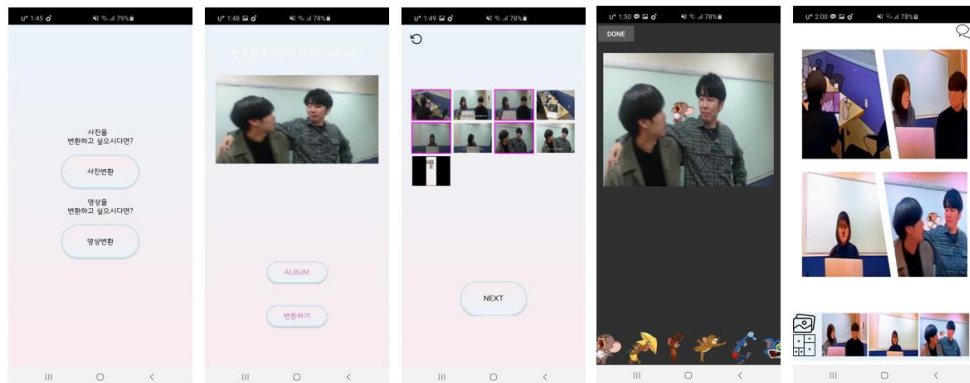


Figure 5-4. Application 시연 이미지 (동영상 부분)

7. REFERENCES

- [1] Chen, Y., Lai, Y. K., & Liu, Y. J. (2018). Cartoongan: Generative adversarial networks for photo cartoonization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9465-9474).
- [2] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679-698.
- [3] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).
- [4] Janwe, N. J., & Bhoyar, K. K. (2016). Video key-frame extraction using unsupervised clustering and mutual comparison. *International Journal of Image Processing (IJIP)*, 10(2), 73-84.
- [5] Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621.
- [6] Pęsko, M., Svystun, A., Andruszkiewicz, P., Rokita, P., & Trzcinski, T. (2019). Comixify: Transform Video Into Comics. *Fundamenta Informaticae*, 168(2-4), 311-333.
- [7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
- [8] Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2018). Self-attention generative adversarial networks.

arXiv preprint arXiv:1805.08318.

- [9] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).