

일반논문 (Regular Paper)

방송공학회논문지 제24권 제6호, 2019년 11월 (JBE Vol. 24, No. 6, November 2019)

<https://doi.org/10.5909/JBE.2019.24.6.1076>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

## 미디어 아카이브 구축을 위한 등장인물, 사물 메타데이터 생성 시스템 구현

조성만<sup>a)</sup>, 이승주<sup>a)</sup>, 이재현<sup>a)</sup>, 박구만<sup>a)†</sup>

### Implementation of Character and Object Metadata Generation System for Media Archive Construction

Sungman Cho<sup>a)</sup>, Seungju Lee<sup>a)</sup>, Jaehyeon Lee<sup>a)</sup>, and Gooman Park<sup>a)†</sup>

#### 요 약

본 논문에서는 딥러닝을 적용하여 미디어 내의 등장인물 및 사물을 인식, 메타데이터를 추출하고 이를 통해 아카이브를 구축하는 시스템을 개발하였다. 방송 분야에서 비디오, 오디오, 이미지, 텍스트 등의 멀티미디어 자료들을 디지털 콘텐츠로 전환하기 시작한지 오래 되었지만, 아직 구축해야 할 자료들은 방대하게 남아있다. 따라서 딥러닝 기반의 메타데이터 생성 시스템을 구현하여 미디어 아카이브 구축에 소모되는 시간과 비용을 절약 할 수 있도록 하였다. 전체 시스템은 학습용 데이터 생성 모듈, 사물 인식 모듈, 등장인물 인식 모듈, API 서버의 네 가지 요소로 구성되어 있다. 미디어 내에서 등장인물 및 사물을 인식하여 메타데이터로 추출할 수 있도록 딥러닝 기술로 사물 인식 모듈, 얼굴 인식 모듈을 구현하였다. 딥러닝 신경망을 학습시키기 위한 데이터를 구축하기 용이하도록 학습용 데이터 생성 모듈을 별도로 설계하였으며 얼굴 인식, 사물 인식의 기능은 API 서버 형태로 구성하였다. 1500명의 인물, 80종의 사물 데이터를 사용하여 신경망을 학습시켰으며 등장인물 테스트 데이터에서 98%, 사물 데이터에서 42%의 정확도를 확인하였다.

#### Abstract

In this paper, we introduced a system that extracts metadata by recognizing characters and objects in media using deep learning technology. In the field of broadcasting, multimedia contents such as video, audio, image, and text have been converted to digital contents for a long time, but the unconverted resources still remain vast. Building media archives requires a lot of manual work, which is time consuming and costly. Therefore, by implementing a deep learning-based metadata generation system, it is possible to save time and cost in constructing media archives. The whole system consists of four elements: training data generation module, object recognition module, character recognition module, and API server. The deep learning network module and the face recognition module are implemented to recognize characters and objects from the media and describe them as metadata. The training data generation module was designed separately to facilitate the construction of data for training neural network, and the functions of face recognition and object recognition were configured as an API server. We trained the two neural-networks using 1500 persons and 80 kinds of object data and confirmed that the accuracy is 98% in the character test data and 42% in the object data.

Keyword : Deep learning, Face recognition, Archive, Metadata, Media

Copyright © 2016 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

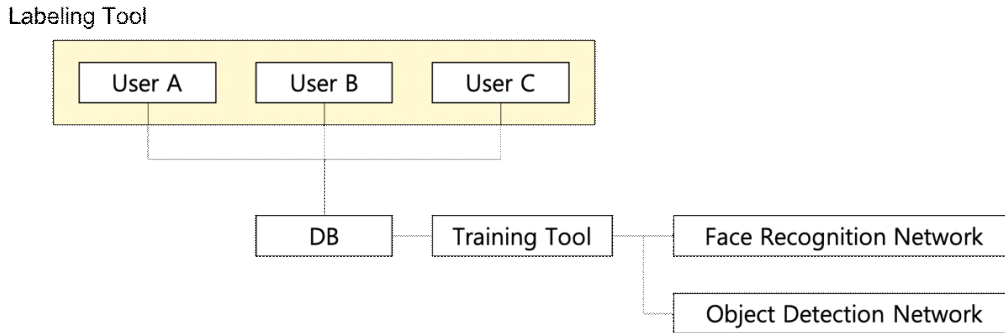


그림 1. 전체 시스템 구조도  
 Fig. 1. Overall System Architecture

## 1. 서론

최근 빅데이터와 딥러닝의 발달로 인하여 방송, 통신, 보안 등의 다양한 영역에서 딥러닝 기반의 솔루션을 구축하기 위한 다양한 연구를 진행하고 있다<sup>[1]</sup>. 딥러닝 기반의 서비스를 위해서는 빅데이터의 구축이 필요하다. 방송 분야에서도 마찬가지로 디지털 콘텐츠를 활용하여 미디어 아카이브를 구성하고 이를 사용한 딥러닝 기반의 서비스를 연구하고 있다. 하지만 이런 미디어 아카이브를 만드는데 수작업이 필요하기 때문에 시간과 비용이 많이 소모되게 된다<sup>[2]</sup>. 따라서 미디어 아카이브를 만들기 위한 메타데이터 확보의 자동화 작업은 매우 중요하다.

딥러닝 기반의 시스템을 자동화에 사용하기 위해서는 아직 사람의 수작업이 많이 필요하다. 데이터의 품질에 따라 딥러닝 기반의 시스템의 성능이 크게 좌우될 수 있기 때문에 데이터의 정제과정에서 엄청나게 많은 양의 시간과 노력이 필요하다. 하지만, 기존 사용되던 규칙 기반의 시스템

에 비하여 딥러닝 기반의 시스템이 성장가능성이 더 높고 하드웨어의 발달 및 신규 연구자의 다량 유입으로 인하여 성장속도는 더욱 더 증가하고 있다.

이에 본 논문에서는 미디어 아카이브를 만들기 위한 딥러닝 기반의 메타데이터 생성 시스템을 구현하였다. 전체 시스템은 학습용 데이터 생성 모듈, 사물 인식 모듈, 얼굴 인식 모듈, API 서버의 네 가지 요소로 구성되어 있다. 얼굴 인식 모듈에서는 Facenet<sup>[3]</sup>을 사용하였고, 사물 인식 모듈에서는 SSD<sup>[4]</sup>를 사용하였다. CNN을 활용한 기존 얼굴인식 연구로는 Facenet, Center-loss<sup>[5]</sup>, Sphreface<sup>[6]</sup> 등이 있다. CNN을 활용한 사물 검출의 경우에는 속도가 빠른 1스태이지 기반의 YOLO<sup>[7]</sup>, SSD가 있고, 속도가 약간 느리지만 정확도가 우수한 R-CNN<sup>[8]</sup> 계열의 연구가 있다. 본 논문에서는 얼굴 인식 모듈로 Facenet을 사용하였는데 학습 데이터의 양이 충분히 많은 경우에는 Facenet의 경우가 정확도가 제일 높기 때문이다. 사물 인식 모듈에서는 YOLO에 비하여 작은 물체를 검출을 잘 하면서 R-CNN계열의 네트워크보다 속도가 빠른 SSD를 사용하였다.

a) 서울과학기술대학교 일반대학원 미디어IT공학과(Seoul National University of Science and Technology)

‡ Corresponding Author : 박구만(Gooman Park)

E-mail: gmpark@seoultech.ac.kr

Tel: +82-2-970-6430

ORCID: <https://orcid.org/0000-0002-7055-5568>

※ This study was supported by the Research Program funded by the SeoulTech(Seoul National University of Science and Technology).

· Manuscript received July 4, 2019; Revised August 29, 2019; Accepted September 4, 2019.

## II. 전체 시스템 구조도

그림 1은 전체 시스템 구조도를 나타낸다 학습용 데이터 생성 모듈(Labeling tool, LT), 학습용 데이터로 학습을 진행시키기 위한 학습 모듈(Training tool), 등장인물 데

이터를 구축하기 위한 얼굴 인식 신경망(Face recognition network), 사물 데이터를 구축하기 위한 사물 인식 신경망(Object recognition network), API 서버(Metadata generation server, MGS)으로 구성되어 있다.

학습용 데이터 생성 모듈은 다수의 사용자가 동시에 데이터 작업을 수행할 수 있도록 하기 위하여 원격지에 학습 데이터 베이스를 구축하는 형태로 구성하였다. 이렇게 함으로써 다수의 사용자가 데이터 작업을 진행을 하는 것이 가능해지고 이를 통해 학습용 데이터를 구축하는데 소요되는 시간을 단축시킬 수 있다.

### III. 학습용 데이터 생성 모듈

딥러닝 기반의 서비스를 위해서는 성능 좋은 딥러닝 신경망이 필요하다. 그리고 성능 좋은 신경망을 만들기 위해서는 양질의 학습 데이터가 필요하다<sup>[3]</sup>. 양질의 학습 데이터를 수집하고 가공하기 위하여 학습용 데이터 생성 모듈을 별도로 구현하였으며, MTCNN<sup>[4]</sup>, k-means clustering<sup>[5]</sup>, SSD<sup>[6]</sup>을 활용하였다. 미디어 내 등장인물의 학습용 데이터를 생성하기 위하여, 미디어 내에서 MTCNN을 활용하여 사람의 얼굴을 검출하는 알고리즘을 사용하였다. 또한 사물 데이터를 생성하기 위하여, 미디어 내에서 SSD를 사용하여 사물을 검출하는 알고리즘을 사용하였다.

### 1. 등장인물 데이터 생성

학습용 등장인물 데이터를 수집하기 위하여, 영상에서 30 프레임 단위의 이미지를 추출하고 이미지에서 MTCNN으로 얼굴을 검출하는 방법을 사용하였다. 미디어 내에서 인접한 프레임 사이에서의 등장인물 정보는 유사한 정보이기 때문에 30프레임을 주기로 샘플링 하는 방법을 사용하였다.

위의 방법으로 얼굴을 검출하여 이미지로 저장시킨 후에, 이를 Inception-Resnet v2<sup>[7]</sup>를 사용하여 특징을 추출하였다. 이 후에 추출된 특징점을 Chinese whisper<sup>[8]</sup>으로 클러스터링 하였다. 그림 2는 클러스터링을 활용한 데이터 생성 작업의 구조를 나타낸다.

학습용 등장인물 데이터를 생성할 때, 영상 기반의 클러스터링 작업이 아닌 단일 이미지를 입력으로 하는 경우를 위하여 단일 이미지를 입력으로 받아 학습 데이터를 생성해주는 모드도 추가하였다. 단일 이미지 모드는 사용자가 수작업으로 데이터 정보를 입력해주는 것을 기반으로 하지만, 이미 등장인물 인식 네트워크가 알고 있는 인물의 경우 자동으로 인물의 이름과 위치를 표시해주는 기능을 추가하여 사용자가 쉽게 학습 데이터를 생성할 수 있도록 하였다. 그림 3은 클러스터링을 한 이후의 결과 사진을 보여준다. 그림 4는 단일 이미지 모드의 과정을 나타낸다. 사용자가 직접 데이터를 입력해주는 형태로 이루어지지만, 이미 알고 있는 사물, 사람의 경우 자동으로 사물, 사람의 이름과 위치를 표시해준다.

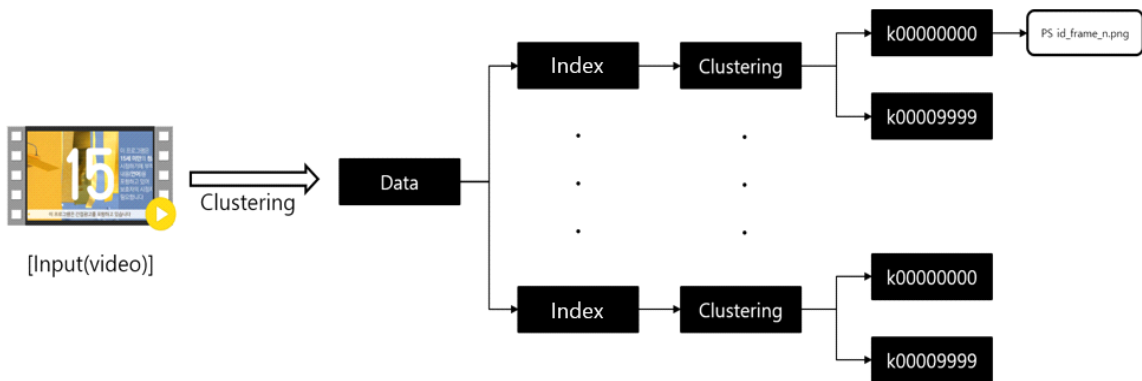


그림 2. 클러스터링을 사용한 데이터 수집 작업  
Fig. 2. Data gathering process using clustering



그림 3. 클러스터링 결과 예시  
 Fig. 3. Clustering results example

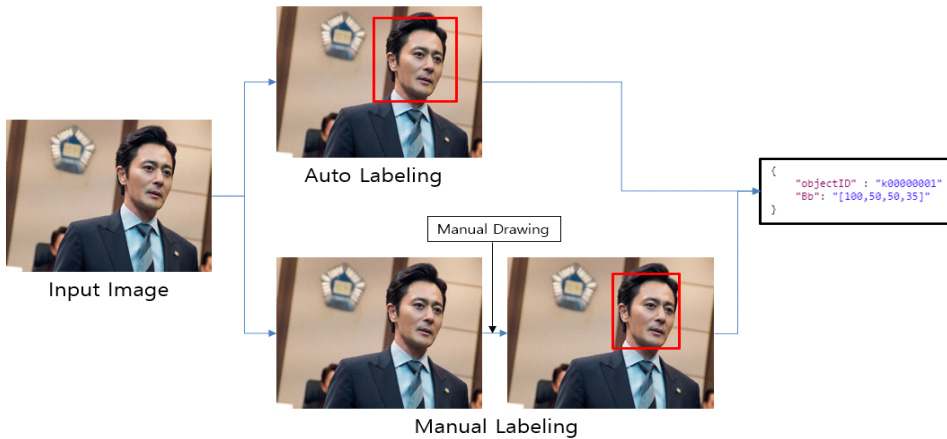


그림 4. 단일 이미지 데이터 생성 과정  
 Fig. 4. Process of generating single image data for training

## 2. 사물 데이터 생성

학습용 사물 데이터의 경우에는 별도의 데이터 수집과정은 거치지 않았다. COCO 데이터셋<sup>[9]</sup> 80종과 VOC 데이터셋<sup>[10]</sup> 20종을 활용하여 80종의 사물에 대하여 학습을 시켰다. COCO dataset을 기준으로 같은 레이블에 넣을 수 있는 이미지를 VOC에서 찾아 이미지를 추가로 더해주는 방법을 사용했다. 등장인물 데이터 수집의 경우와 마찬가지로 사용자가 직접 사물의 이미지와 종류를 학습시킬 수 있지만, 이미 알고 있는 사물의 경우 자동으로 사물의 이름과 위치를 표시해주는 기능을 추가하였다.

## Ⅳ. 딥러닝 기반의 등장인물 인식 모듈

딥러닝 기반의 얼굴 인식 알고리즘을 사용하여 미디어 내의 등장인물을 인식하고, 인식된 결과를 바탕으로 메타 데이터를 저장하기 위해서는 얼굴 인식 알고리즘의 정확도가 높아야 한다. 얼굴 인식 알고리즘의 정확도가 낮으면 생성된 데이터를 검토하는데 걸리는 시간이 전체 과정을 수작업으로 하는 방법과 크게 다르지 않을 수 있다. 얼굴 인식분야의 경우에 테스트 클래스가 closed-set이 아닌 open-set 문제로 여겨지기 때문에, Soft-max로 학습시킨 신경망을 사용하여 특징을 추출하고 이를 SVM, NN 등으로

분류하는 방법을 사용한다. 본 논문에서는 위의 방법을 차용하였고, Facenet과 SVM을 사용하였다.

시와 테스트 시의 시스템의 흐름을 나타낸다.

### 1. 학습, 테스트 시의 신경망 구성

학습 과정에서는 학습용 데이터 생성 모듈을 사용하여 구성된 데이터를 사용하여 Facenet을 학습시켰다. 실제 테스트에서는 Facenet에서의 feature embedding network 부분만 사용하여 feature embedding network를 통과한 feature를 SVM으로 분류하는 형태의 구조로 구성하였다. feature embedding network는 Inception-Resnet-V2를 사용하였다. Adam optimizer를 사용하였고, weight-decay는 5e-4, learning rate는 0.001로 학습을 진행하였다. 그림 5는 학습

### V. 딥러닝 기반의 사물 인식 모듈

딥러닝 기반의 사물 인식 모듈도 얼굴 인식 모듈과 마찬가지로 정확도가 보장되어야 한다. 따라서 사물 인식을 위한 딥러닝 신경망으로 성능이 검증된 SSD를 사용하였다. 사물 인식 신경망은 COCO 데이터셋 +VOC 데이터셋에서 80종의 클래스를 추출하여 학습하는데 사용하였다. 그림 6은 SSD를 사용하여 사물을 인식한 결과 이미지를 나타낸다. SSD 학습에는 learning rate를 0.001, weight decay: 5e-4 로 설정하여 학습을 진행하였다.

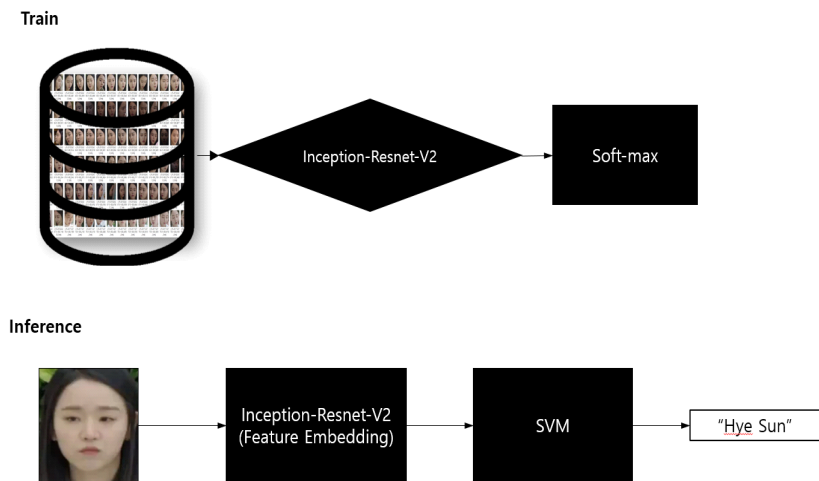


그림 5. 신경망의 학습, 테스트 구조도  
 Fig. 5. Train, test architecture of neural network



그림 6. 사물 이미지 인식 결과  
 Fig. 6. Object image recognition result

사물 인식 모듈과 등장인물 인식 모듈 모두에서 사람이라는 객체에 대하여 검출을 하기 때문에 두 모듈에서 사람 객체에 대하여 서로 다른 결과 값을 반환했을 때에는 등장인물 인식 모듈의 결과값을 기준으로 판단하였다. 그림 6은 서버에서 반환하는 json 파일 형태의 예시이다. frameIdx는 frame의 index 정보를 나타내며, ObjectID, Bb는 해당 클래스의 ID와 Bounding box 정보를 나타낸다. Bounding box는 x, y, width, height로 구성된다.

## VI. API 서버

API 서버는 딥러닝 기반의 사람 인식 모듈과 사물 인식 모듈의 기능을 탑재하고 있다. 사용자는 단일 이미지, 복수 이미지, 영상의 3가지 형태로 서버에 결과를 요청할 수 있으며, 요청을 받은 이미지, 영상에서의 인물, 사물 메타데이터를 json 형태로 반환해준다. 서버는 REST API 형태로 구현하였다. 그림 7은 REST API 서버의 결과를 나타낸다.

## VII. 실험

실험은 크게 학습 데이터의 분포에 따른 신경망의 정확도 비교, 학습 데이터 검증 실험, API 서버의 네트워크별 추론 속도 비교, 클라우드 서비스와의 성능 비교의 4가지를 실시하였다.

### 1. 학습 데이터의 분포에 따른 신경망의 정확도 비교

학습 데이터의 클래스 간의 균형도가 신경망의 정확도에 미치는 영향을 파악하기 위하여, 학습 데이터의 분포를 다양하게 변형해가며 테스트를 시도하였다. 표 1은 학습 데이터 클래스 분포에 따른 신경망의 정확도를 나타낸다. FE (Feature Extraction network)를 클래스별로 450장씩 배치하고 분류기를 450장씩 학습했을 때의 정확도가 제일 좋은 성능을 보였다.

첫번째 줄에서, FE와 분류기에서의 데이터 수는 450장이상인 클래스를 450장으로 맞춰주었다는 것을 의미한다. 450장 보다 적은 이미지 데이터를 450장으로 맞춰주는 작

표 1. 학습 데이터 클래스 분포에 따른 신경망의 정확도  
 Table 1. Inter-class balancing test accuracy

FE training data number (each class)	Mean	Std	Classifier training data number (each class)	Accuracy
450	313.84	159.28	450	0.981
			400	0.962
400	287.84	137.53	450	0.966
			400	0.975
300	230.76	92.93	450	0.939
			300	0.966
200	169.3	52	450	0.883
			200	0.939

업은 수행하지 않았다. 따라서 학습 데이터가 450장보다 적은 경우가 존재할 수 있고 그렇기 때문에 추출하는 학습데이터의 수를 변경할수록 데이터의 평균과 표준편차가 달라지게 된다.

미디어 내에서 등장인물의 비중이 다르기 때문에 30프레임 단위로 샘플링을 하게 되었을 때 등장인물의 클래스별 학습 데이터의 수의 불균형이 나타났지만, 인위적으로 데이터의 수를 맞춰주는 작업은 별도로 수행하지 않았다.

그림 8은 FE를 통과한 데이터의 분포를 나타낸다. FE를 통과한 이후의 Feature에 PCA<sup>[17]</sup>를 적용하여 차원을 줄이고, t-SNE<sup>[18]</sup>를 사용하여 시각화 하였다. 왼쪽의 그림은 50개 클래스에 대하여 100개씩 샘플링을 하였을 때, 오른쪽 그림은 50개 클래스에 대하여 450개씩 샘플링을 하였을 때를 의미한다.

```

{
  "frameIdx" : 6240,
  "result": [
    {
      "objectID" : "k00002432",
      "Bb": [439, 67, 142, 105]
    },
    {
      "objectID" : "k00002433",
      "Bb": [798, 203, 136, 121]
    }
  ]
}
    
```

그림 7. API 서버 반환 형태  
 Fig. 7. API server return format



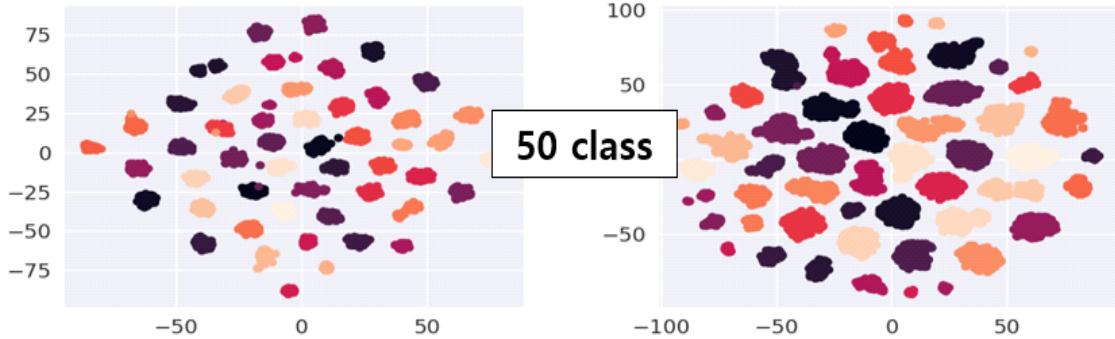


그림 8. 학습데이터 분포도  
Fig. 8. Training data distribution

### 2. API 서버의 네트워크별 속도 비교

API 서버의 속도 비교를 위하여 SD, HD, FHD, UHD의 4종류의 영상과 내부망, 외부망, 핫스팟 3종류의 네트워크를 비교하며 실험을 진행하였다. 그림 9는 네트워크의 종류 및 영상의 종류에 따른 네트워크의 속도 실험 결과이다. 내부망을 사용하는 경우 FHD 영상 기준으로 한 장의 이미지 전송에 0.3초 정도 소요되는 것을 확인하였다. 서버에서의 얼굴인식, 사물인식 등의 이미지 처리 속도는 FHD일 경우에 1장당 1초 이내인 것을 확인하였다.

### 3. 클라우드 서비스 성능 비교

완성된 등장인물 인식 신경망 및 사물 인식 신경망의 성능

을 비교하기 위하여 클라우드 API와 성능 비교 실험을 실시하였다. 표 2, 3은 클라우드 API와의 비교 실험 결과를 나타낸다. 등장인물 정확도 실험에서 AWS 나 NAVER의 경우 1500 클래스의 테스트 데이터에서 학습되지 않은 클래스가 많기 때문에 정확도가 낮게 측정되었다. 전체 시스템의 속도를 위하여 MobileNet<sup>[19]</sup> 기반의 SSD를 사용하다

표 2. 클라우드별 등장인물 인식 정확도 테스트  
Table 2. Character recognition accuracy test

Cloud	# of class	# of test data	Accuracy (%)
NAVER CFR API	1500	31k	26
AWS Rekognition			22
Ours			98

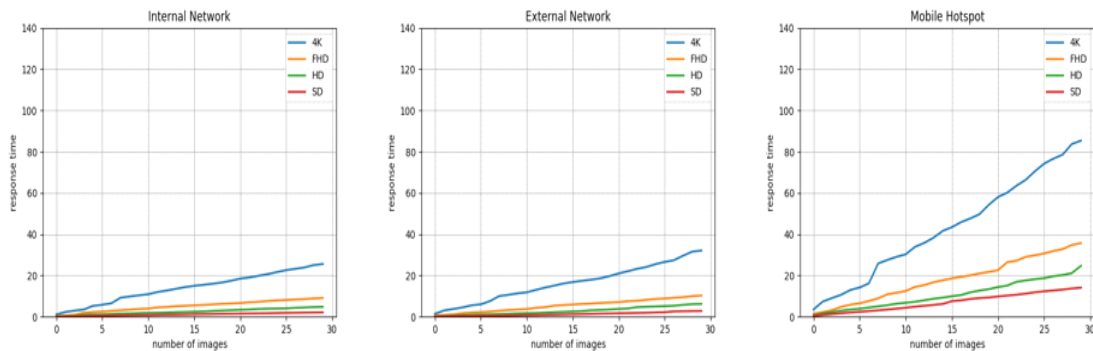


그림 9. 네트워크 및 영상의 종류에 따른 네트워크 속도 비교  
Fig 9. Network speed comparison according to network and image type

표 3. 클라우드별 사물 인식 정확도 테스트  
 Table 3. Object recognition accuracy test

Cloud	# of class	# of test data	Accuracy (%)
Google vision API	30	600	46
AWS Rekognition			62
Ours			41

보니, 기존 클라우드 시스템인 AWS, Google vision API 보다는 정확도가 좋지 않은 것을 확인할 수 있었다.

### Ⅷ. 결 론

본 논문에서는 미디어 아카이브를 구축하기 위하여 미디어 내에서 등장인물, 사물의 메타데이터를 추출하는 시스템에 대하여 소개하였다. 전체 시스템은 API 서버 형태로 구현하여 접근성을 용이하게 하였다. 딥러닝 기반의 방법론을 사용하여 높은 정확도를 보이는 메타데이터 추출 시스템을 구현하였으며, 클라우드와의 성능 비교 테스트를 통해 수집된 데이터 내에서는 등장인물의 경우 더 좋은 정확도를 보이는 것을 확인했다. 사물인식 모듈에서 시스템의 속도를 위하여 MobileNet 기반의 SSD를 사용하였기 때문에 정확도가 AWS보다 낮았다. 추후 네트워크를 변경하여 정확도를 향상하는 실험을 진행할 계획이다.

### 참 고 문 헌 (References)

[1] I.Choi, H.Song, S.Lee, J.Yoo, "Facial Expression Classification Using Deep Convolutional Neural Network", Journal of Broadcast Engineering, Vol.22, No.2, March 2017.  
 [2] H.Jun, G.Hyun, K.Lim, W.Lee, H.Kim, "Big Data Preprocessing for Predicting Box Office Success", KISE Transactions on Computing Practices, Vol.20, No.12, pp.615-622, December 2014.  
 [3] F.Schroff, D.Kalenichenko, J.Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 815-823, 2015.

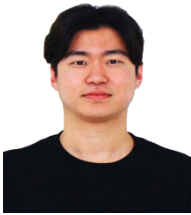
[4] W.Liu, D.Anguelov, D.Erhan, C.Szegedy, S.Reed, "SSD: Single shot multibox detector", arxiv:1512.02325, 2015.  
 [5] Y.Wen, K.Zhang, Z.Li, Y.Qiao, "A discriminative feature learning approach for deep face recognition", In European Conference on Computer Vision (ECCV), pages 499-515, 2016.  
 [6] W.Liu, Y.Wen, Z.Yu, M.Li, B.Raj, L.Song, "SphereFace: Deep Hypersphere Embedding for Face Recognition", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017  
 [7] J.Redmon and A.Farhadi, "You only look once: Unified, real-time object detection", In CVPR, 2016.  
 [8] R.Girshick, J.Donahue, T.darrell, J.malik, "Rich feature hierarchies for accurate object detection and semantic segmentation", In CVPR, 2014.  
 [9] Y.Kim, J.Woo, J.Lee, J.Shin, "High-quality data collection for machine learning using block chain", Journal of the Korea Institute of Information and Communication Engineering, Vol.23, No.1, pp.13-19, Jan 2019.  
 [10] K.Zhang, Z.Zhang, Z.Li, Y.Qiao, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks", IEEE SIGNAL PROCESSING LETTERS, Vol.23, No.10, October 2016  
 [11] W.Kiri, C.Claire, R.Seth, S.Stefan, "Constrained K-means Clustering with Background Knowledge", Proceedings of the Eighteenth International Conference on Machine Learning, pp.574-584, 2001  
 [12] Szegedy C, Ioffe S, Vanhoucke V, "Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning". In: AAAI. vol. 4, pp. 12, 2017.  
 [13] Chris Biemann, "Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems", TextGraphs-1 Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing, pp.73-80, 2006.  
 [14] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár, "Microsoft COCO: Common Objects in Context", arxiv : 1405.0312, 2015.  
 [15] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A., "The PASCAL Visual Object Classes (VOC) Challenge", International Journal of Computer Vision, 88(2), 303-338, 2010  
 [16] C. J. C. Burges, "A tutorial on Support vector machines for pattern recognition", Data mining and knowledge discovery 2, pp. 121-167, 1998.  
 [17] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction", IEEE Transactions on Automatic Control, pp. 17-32, Feb 1981.  
 [18] L. Maaten, G. Hinton, "Visualizing data using t-SNE", Journal of Machine Learning Research 9, pp. 2579-2605, Nov 2008.  
 [19] Andrew G.Howard, M.Zhu, Bo Chen, D.Kalenichnko, W.Wang, T.Weyand, M.Andreetto, H.Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", CoRR, abs/1704.04861, 2017.



---

— 저 자 소 개 —

---



**조 성 만**

- 2018년 2월 : 서울과학기술대학교 전자IT미디어공학과 공학사
- 2018년 3월 ~ 현재 : 서울과학기술대학교 일반대학원 미디어IT공학과 석사과정
- ORCID : <https://orcid.org/0000-0002-7922-9638>
- 주관심분야 : 객체 검출, 얼굴 인식, 딥러닝, 컴퓨터비전



**이 승 주**

- 2016년 2월 : 상명대학교 컴퓨터소프트웨어공학과 공학사
- 2018년 9월 ~ 현재 : 서울과학기술대학교 일반대학원 미디어IT공학과 석사과정
- ORCID : <https://orcid.org/0000-0001-9212-4411>
- 주관심분야 : 객체 검출, 딥러닝, 컴퓨터비전



**이 재 현**

- 2018년 2월 : 수원대학교 전자공학과 공학사
- 2018년 3월 ~ 현재 : 서울과학기술대학교 일반대학원 미디어IT공학과 석사과정
- ORCID : <https://orcid.org/0000-0001-9212-4411>
- 주관심분야 : 객체 검출, 딥러닝, 컴퓨터비전



**박 구 만**

- 1984년 2월 : 한국항공대학교 전자공학과 공학사
- 1986년 2월 : 연세대학교 전자공학과 석사
- 1991년 2월 : 연세대학교 전자공학과 박사
- 1991년 3월 ~ 1996년 9월 : 삼성전자 신호처리연구소 선임연구원
- 2016년 1월 ~ 2017년 12월 : 서울과학기술대학교 나노IT디자인융합대학원 원장
- 1999년 8월 ~ 현재 : 서울과학기술대학교 전자IT미디어공학과 교수
- 2006년 1월 ~ 2007년 8월 : Georgia Institute of Technology Dept.of Electrical and Computer Engineering, Visiting Scholar
- ORCID : <https://orcid.org/0000-0002-7055-5568>
- 주관심분야 : 컴퓨터비전, 지능형실감미디어