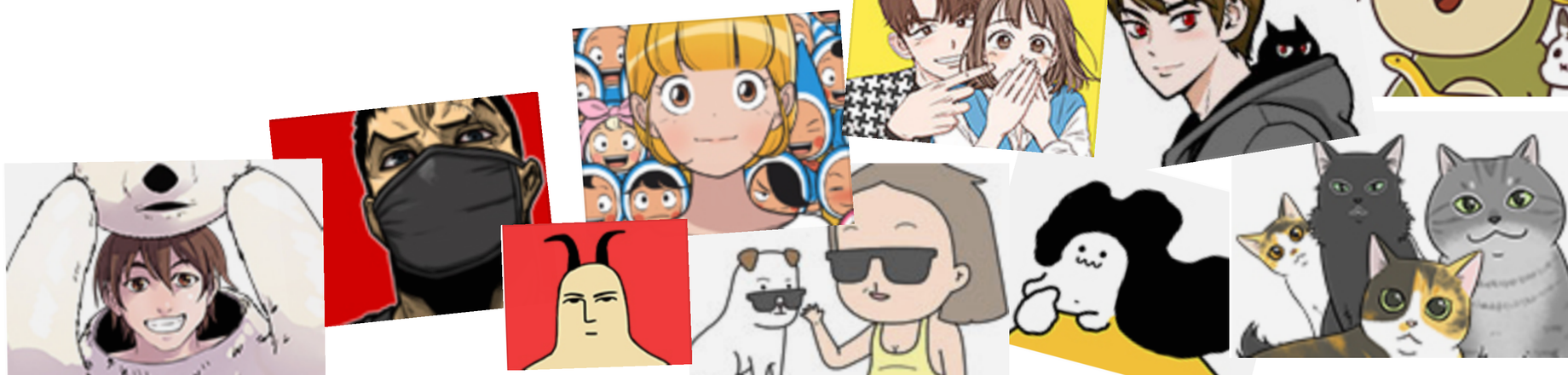
비슷한 그림체의 웹툰 찾아내기

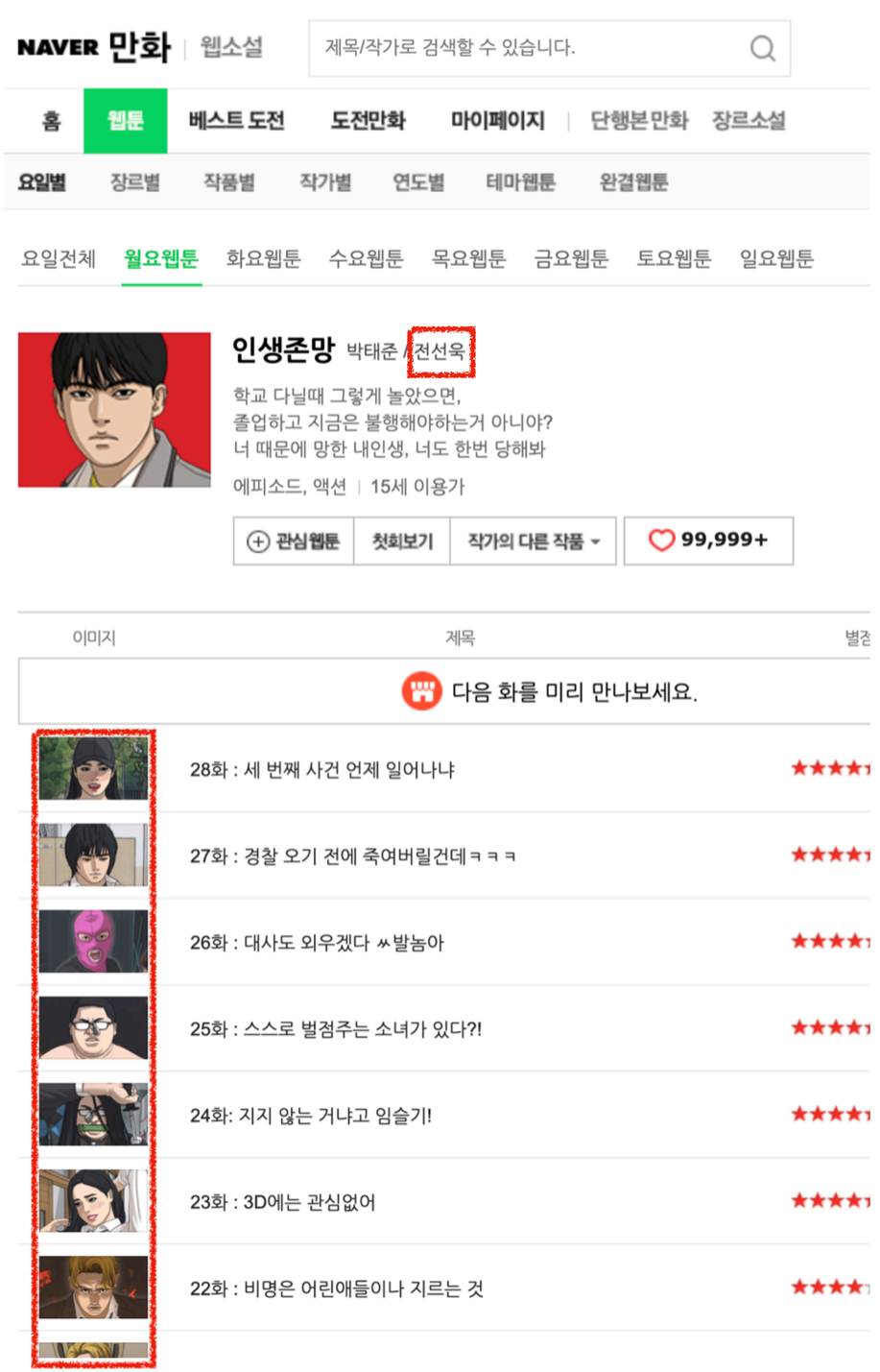
이진주
조대선
송이준

2020.05.29



Intro

- ✓ CNN을 이용한 이미지 classification
- ✓ 수집 데이터 : 네이버 전체 웹툰 thumbnail
- ✓ 라벨 : 작가명



순서도

데이터 수집(크롤링)

✳ 네이버 전체 웹툰 thumbnail

CNN 모델링

✳ image size 71x42

✳ 라벨 수 : 315개 (작가이름)

✳ 하이퍼 파라미터 튜닝

Thumbnail 수 200 이상 CNN 모델링

✳ Image size 35x21

✳ 라벨 수 : 33개 (작가이름)

✳ 소요 시간 절약

✳ RAM 절약 (augmentation)

프로젝트 진행

Image Augmentation

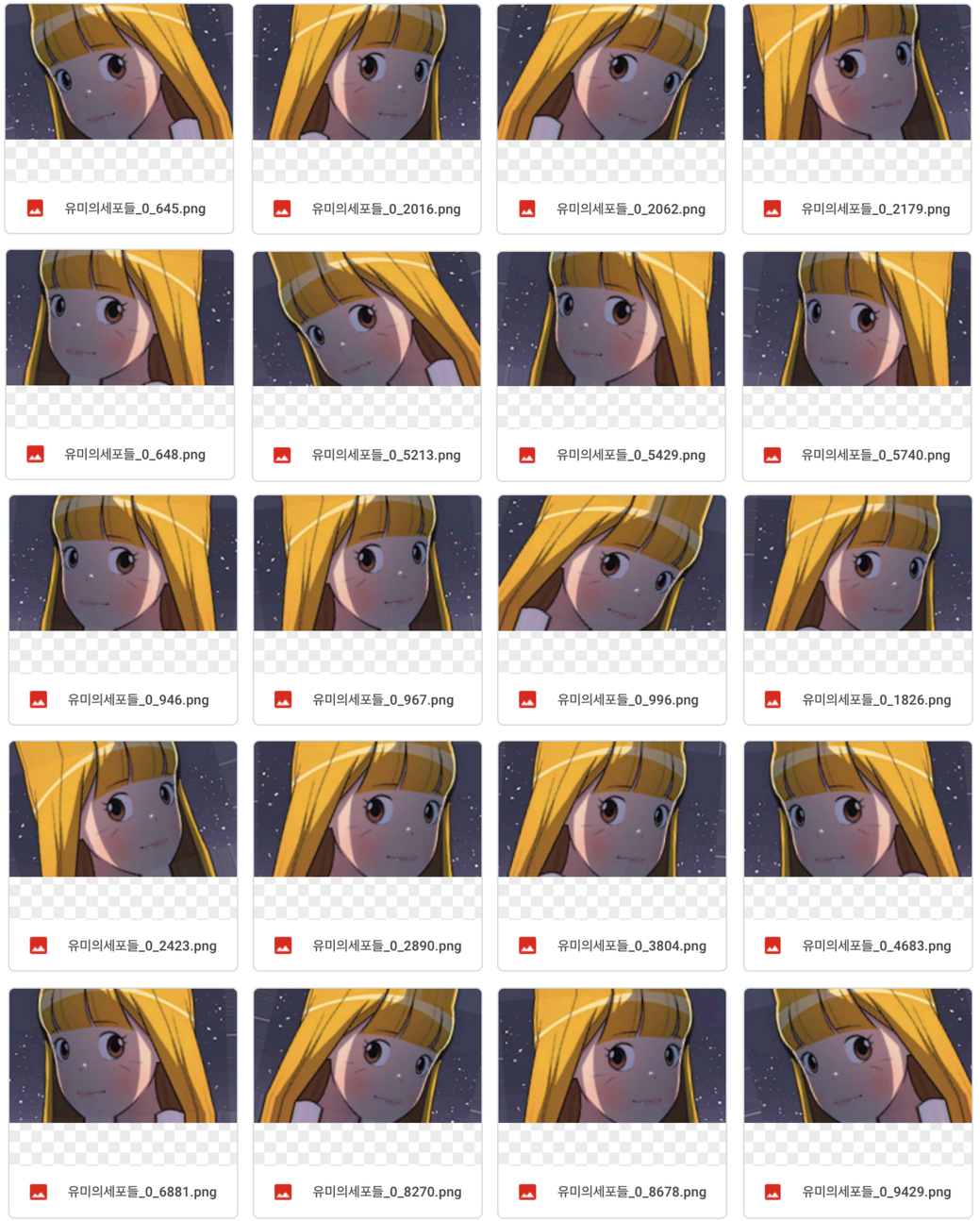
- ✳ 데이터 부족 문제를 해결하기 위한 방안
- ✳ 이미지에 임의의 변형을 가해 훨씬 많은 이미지로 학습하는 효과
- ✳ 과적합 방지

```
train_generator = ImageDataGenerator(
    zoom_range=0.05,
    rotation_range=30,
    width_shift_range=0.05,
    height_shift_range=0.05,
    horizontal_flip=True,
    fill_mode='nearest')

train_generator.fit(X_train)
train_iterator = train_generator.flow(X_train, Y_train, batch_size=len(X_train))
```

```
N = 50 # 오그멘테이션 할 배수
print("total N : ", N)

X_train_new = []
Y_train_new = []
for i in range(N):
    x_train_new, y_train_new = train_iterator.next()
    X_train_new.append(x_train_new)
    Y_train_new.append(y_train_new)
```



CNN

CNN 이란?

Convolutional Neural Network = CNN

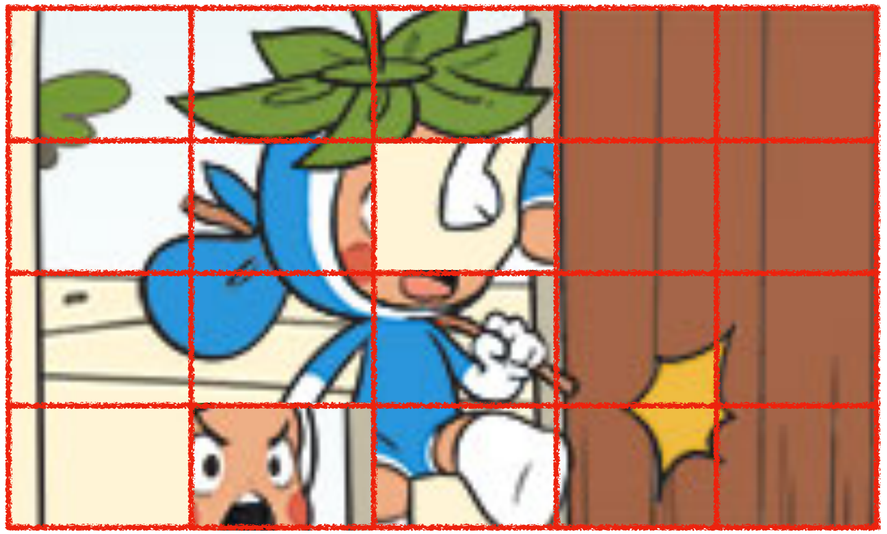
일반적인 Neural Network 앞 부분에 Convolution layer를 추가하여

이미지에서 패턴을 찾아내 분류하는 딥러닝 알고리즘

Why CNN?

이미지 구분 → 공간 정보 파악

(O)

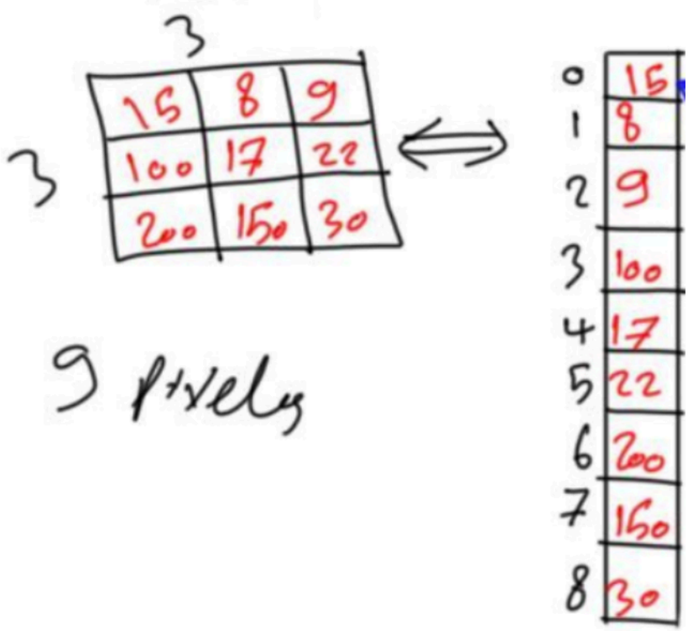


(X)

‘눈’은 얼굴 주변의 픽셀들과 함께 있을 가능성이 높다

이미지의 특징을 주변 특징들과의 관계 속에서 파악하기 위함

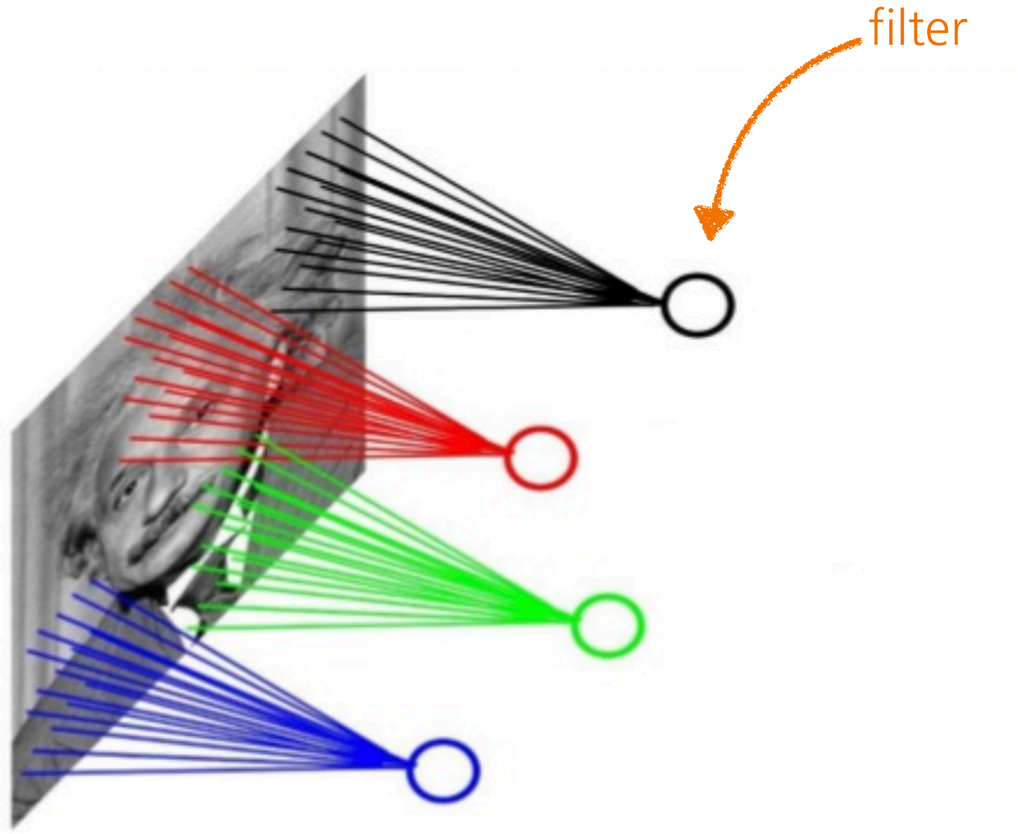
이미지의 어떤 특징을 그 주변 특징들과의 관계 속에서 파악하려면?



이미지를 곧바로1차원으로 변환하면
각각의 픽셀들이 독립적인 feature가 되어버려서,
공간정보를 잃게 된다

공간정보를 유지한 채로 이미지를 구성하는 각 특징들의 가중치를 계산하려면?

matrix 자체를 input으로 받아 가중치를 계산하는 layer
⇒ convolution layer



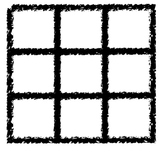
Convolution layer filter(kernel)

```
from keras.models import Sequential

model = Sequential()
model.add(Conv2D(96, (3, 3), activation='relu', padding='same', kernel_regularizer=l2(0.01)))
```

filter

filter size

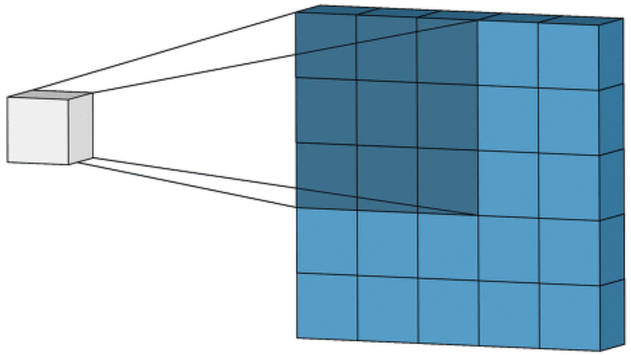


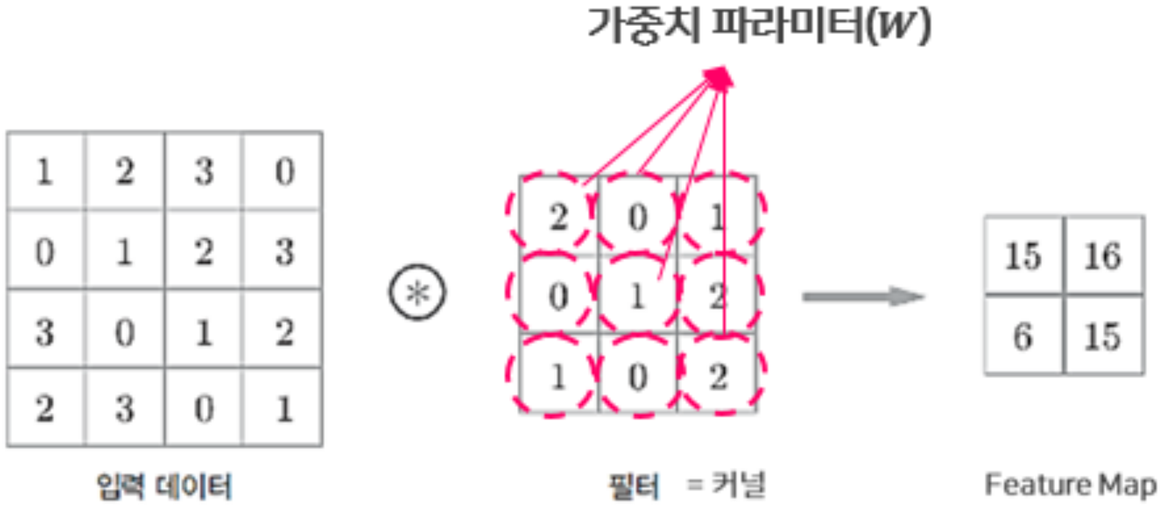
Ridge 정규화

✳ 추출하고자 하는 feature의 내용을 담고 있음

✳ 데이터에서 특징을 추출하는 기능

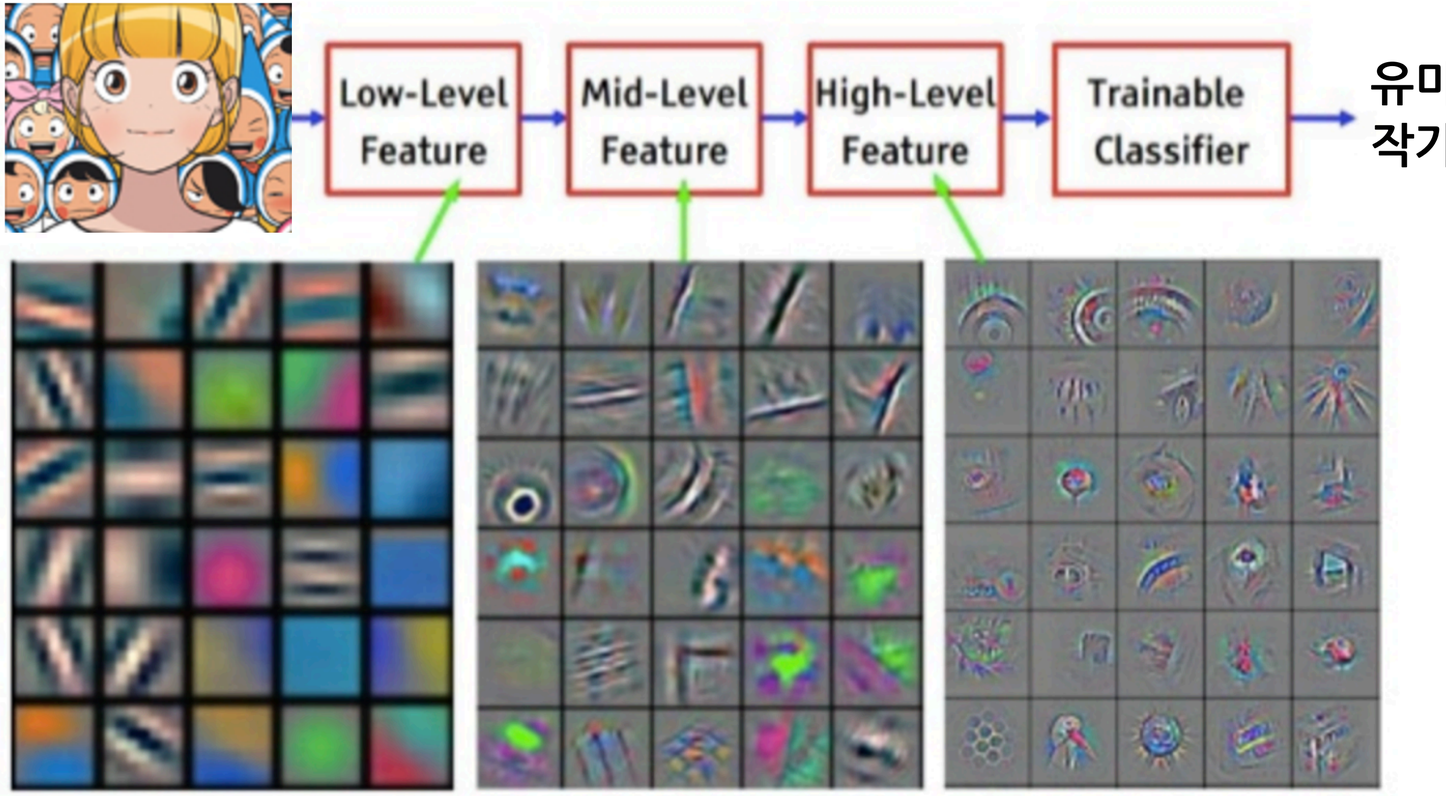
✳ 합성곱 사용





Convolution layer filter(kernel)

✳ 입력 데이터로부터 특징(feature)을 추출하는 역할



유미의 세포들
작가이름 : 이동건

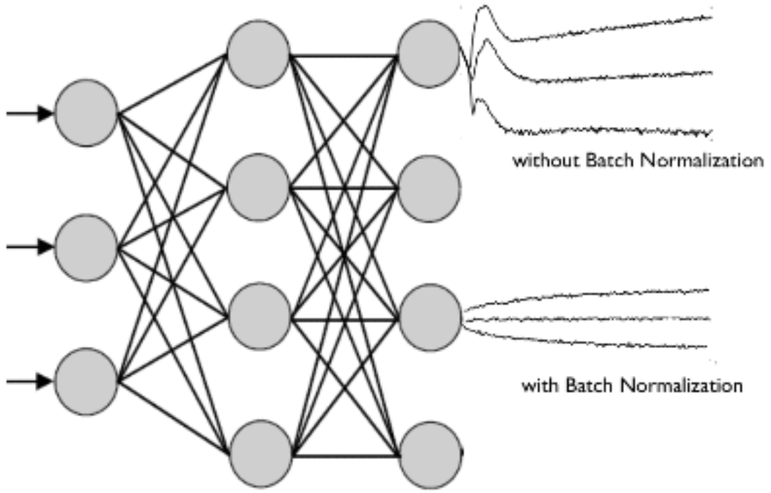
✓ 점점 더 구체적인 특징들을 잡아냄

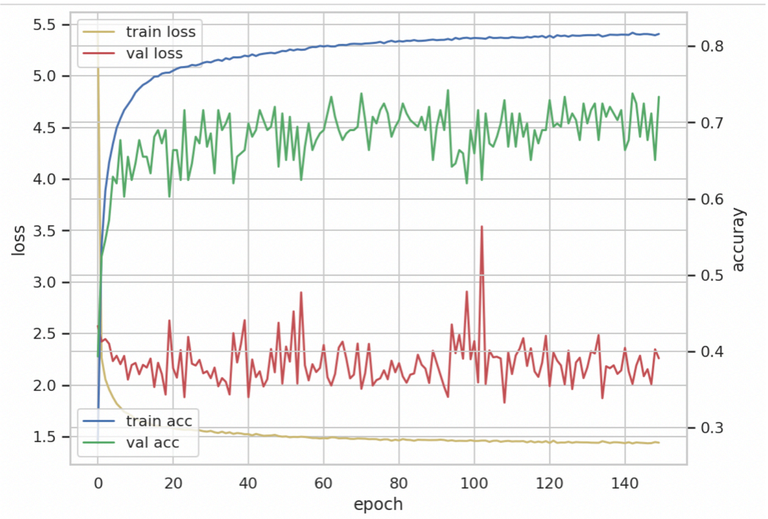
이렇게 추출된 특징을 기반으로, 뉴럴 네트워크를 이용하여 분류한다

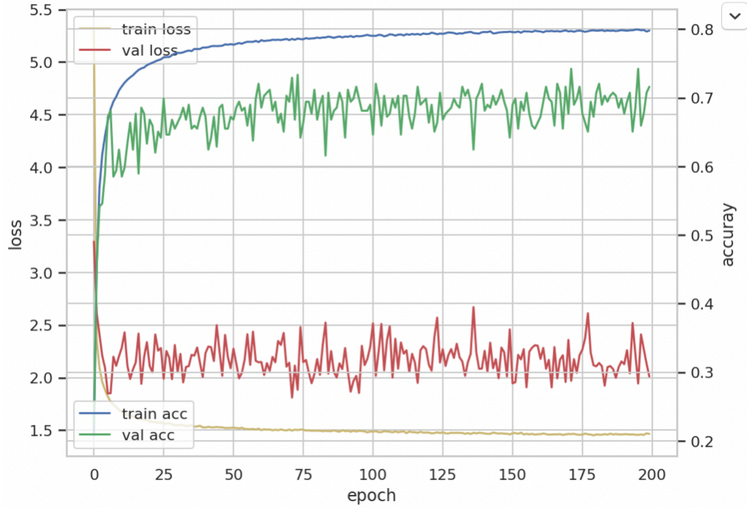
Batch Normalization layer

```
model.add(BatchNormalization())
```

✳ 학습 시 배치 사이즈만큼의 데이터 각각에 대한 평균과 분산을 구해서 정규화 시키는 것
✳ 각 층의 활성화 함수 출력값 분포가 골고루 분포되도록 하는 것
✳ 분포의 평균이 0, 분산이 1이 되도록 정규화 하는것

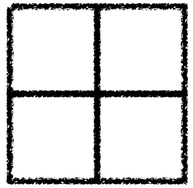






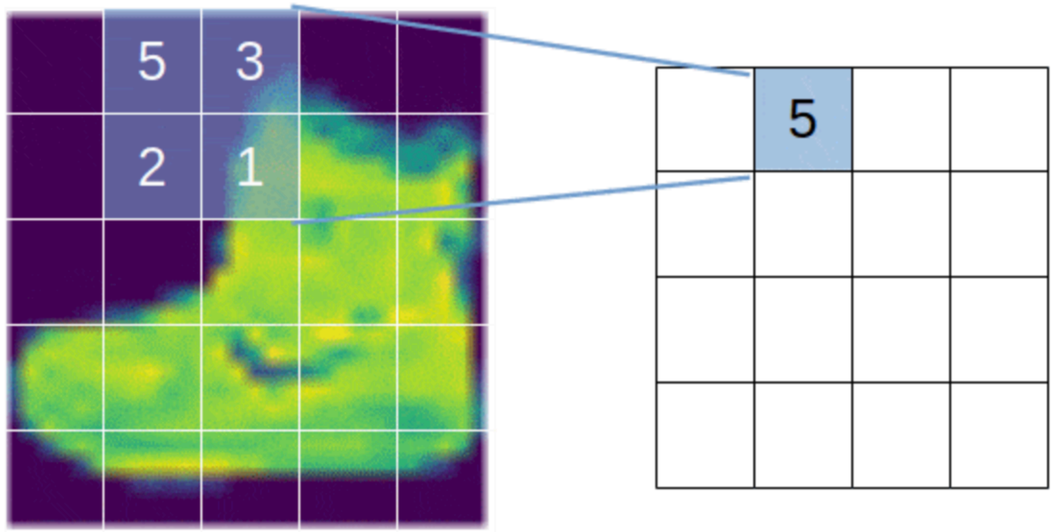
Maxpooling layer

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```



✳ 컨볼루션 레이어의 입력 데이터의 크기를 줄여 특정 부분만 강조하는 용도

✳ 의미가 있는 픽셀만 남겨서 연산량을 줄인다 ⇒ 학습시간 감소



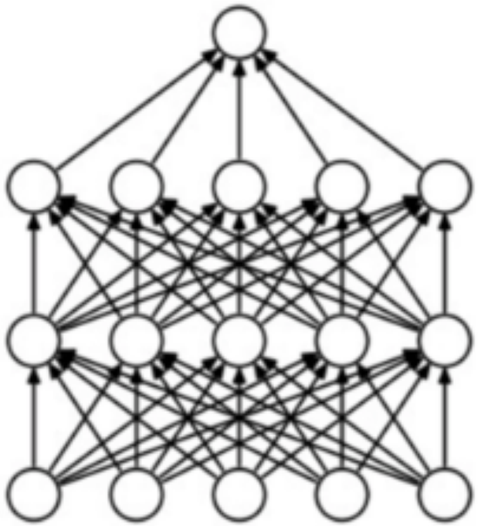
Dropout layer

```
model.add(Dropout(0.2))
```

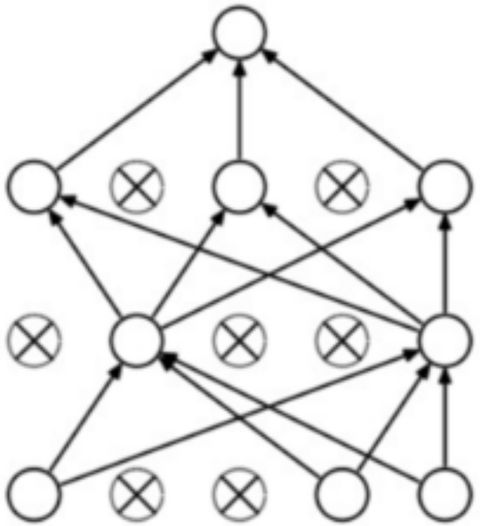
p

✳ p의 확률로 무작위로 뉴런을 선택하여 제외하고 학습시키는 방법

✳ 과적합 방지



(a) Standard Neural Net



(b) After applying dropout.

Fully Connected layer(Dense layer)

```
model.add(Flatten())
model.add(Dense(512, activation='relu', kernel_regularizer=l2(0.01)))
model.add(Dropout(0.5))
model.add(Dense(num_cat, activation='softmax')) # num_cat : 클래스 개수
```

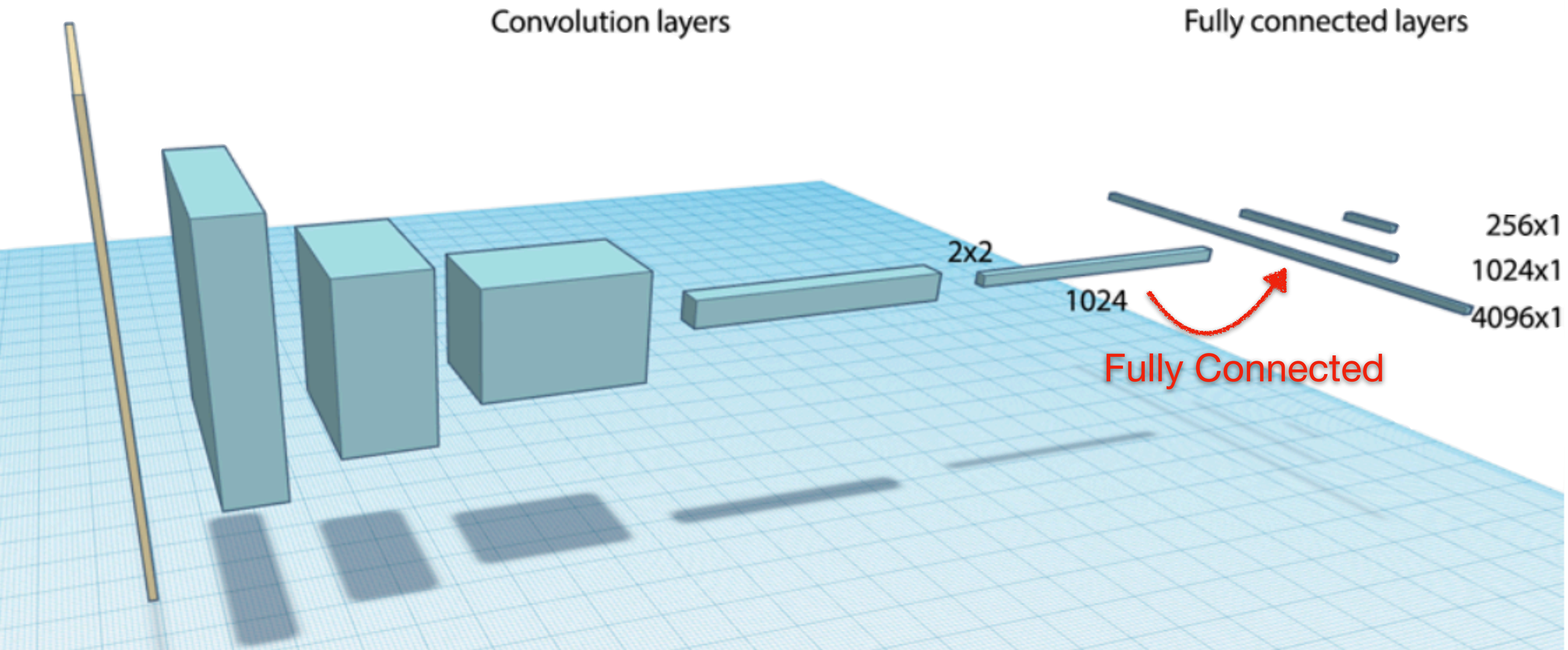


Image 분류 = CNN

Modeling

Modeling & Training 환경

Colab Pro

$9.99/월

더 빠른 GPU

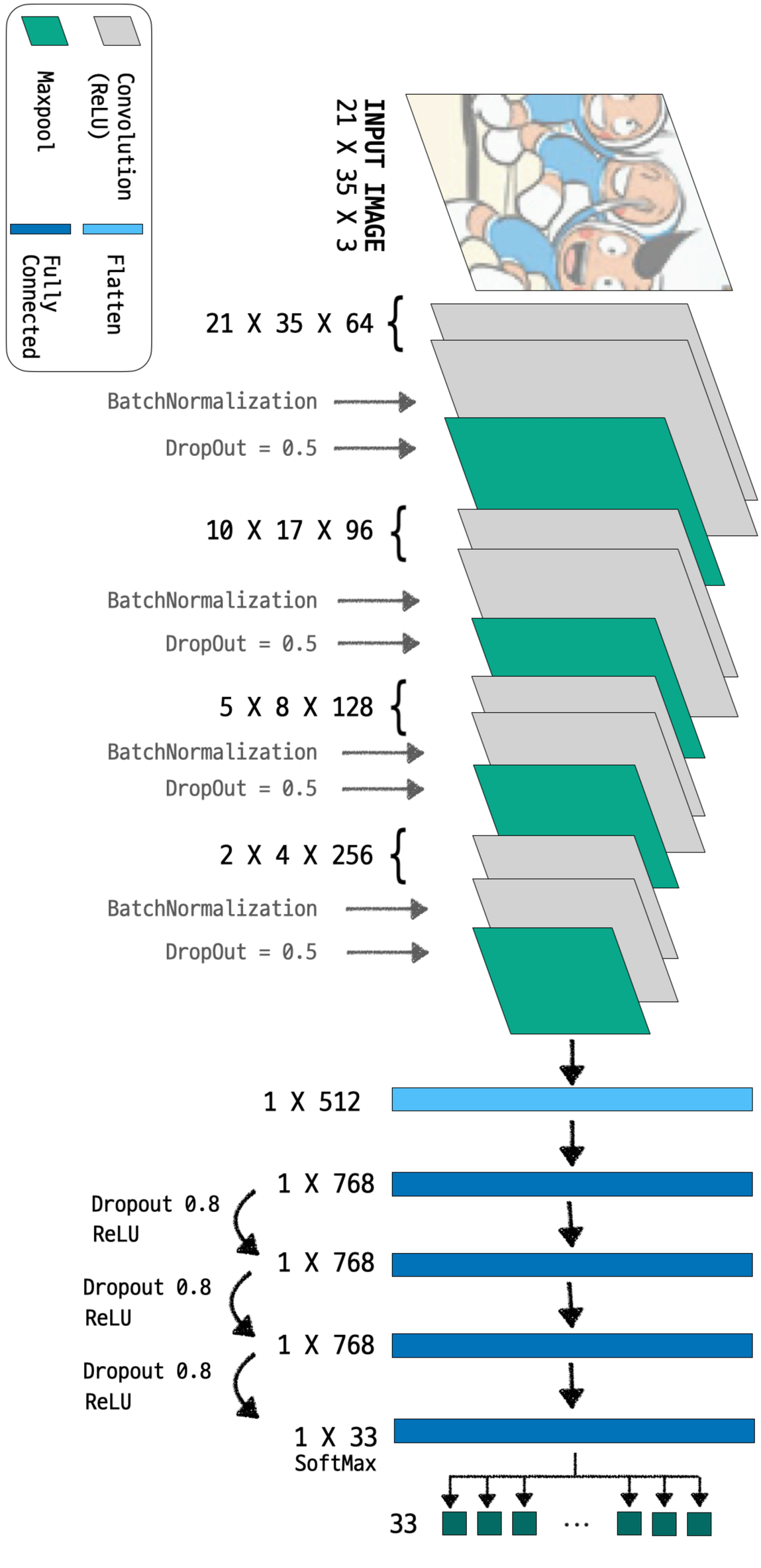
더 긴 런타임

추가 메모리

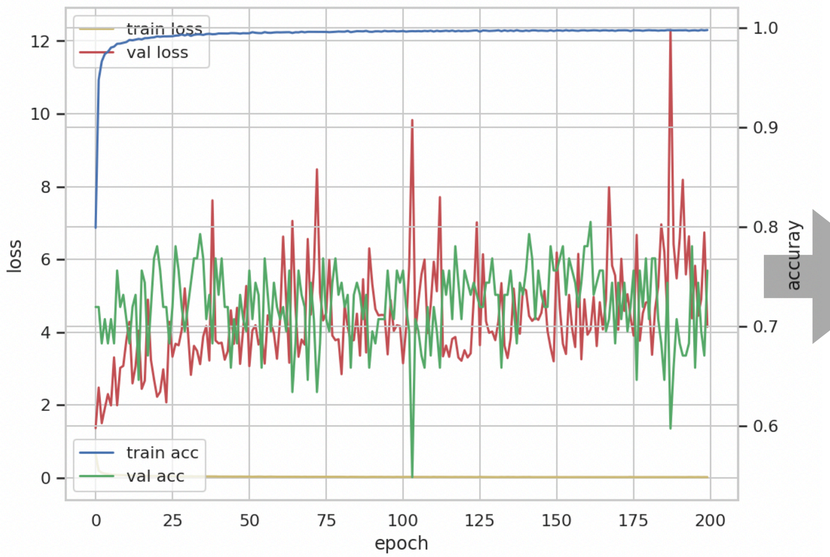
'Python 3 Google Compute Engine 백엔드 (GPU)에 연결됨

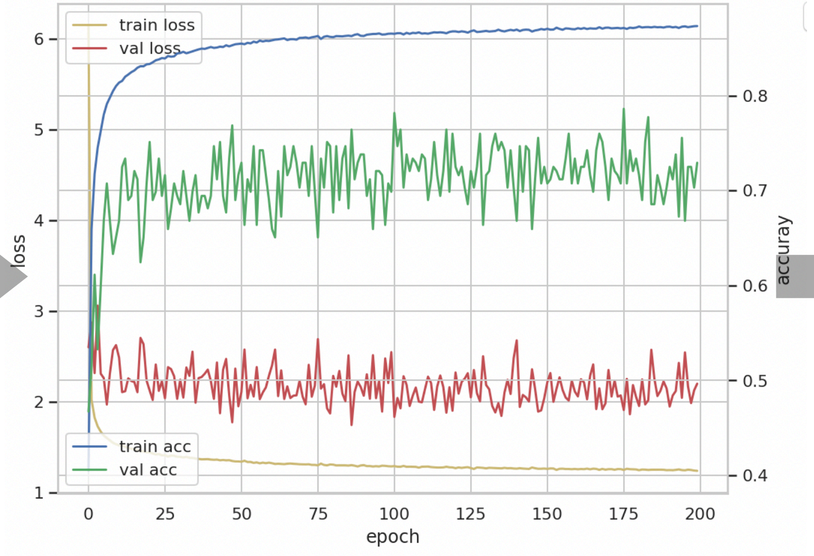
RAM: 1.02 GB/25.51 GB 디스크: 31.43 GB/68.40 GB

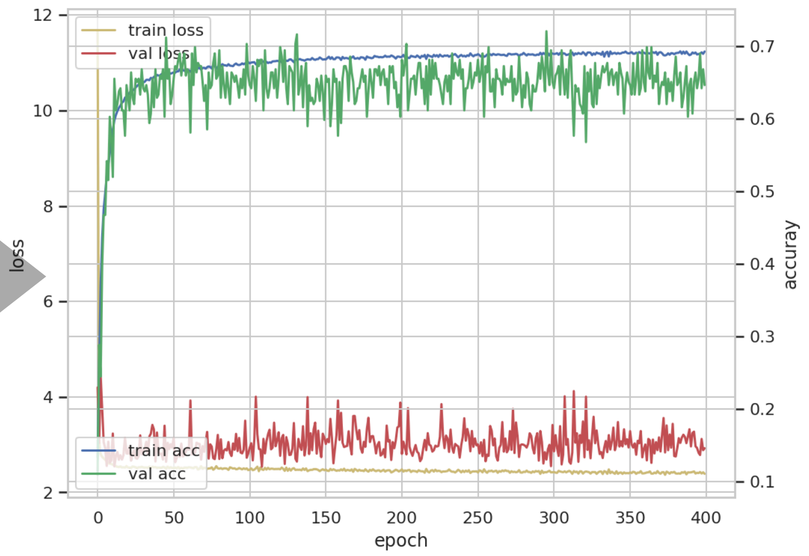
model


Convolution (ReLU)
Maxpool
Flatten
Fully Connected
INPUT IMAGE
21 X 35 X 3
21 X 35 X 64
BatchNormalization
DropOut = 0.5
10 X 17 X 96
BatchNormalization
DropOut = 0.5
5 X 8 X 128
BatchNormalization
DropOut = 0.5
2 X 4 X 256
BatchNormalization
DropOut = 0.5
1 X 512
1 X 768
Dropout 0.8
ReLU
1 X 768
Dropout 0.8
ReLU
1 X 768
Dropout 0.8
ReLU
1 X 33
SoftMax
33

CNN modeling 결과 그래프





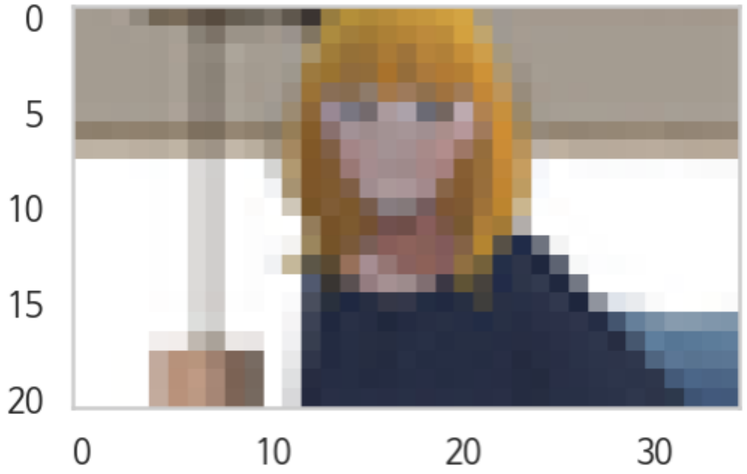


Classification Result

인풋데이터 : 유미의 세포들

작가명 : 이동건(라벨있는 데이터의 최신 화)

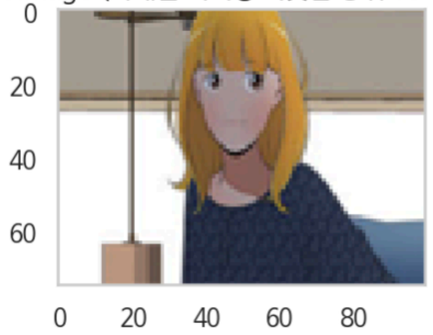
테스트로 사용한 이미지 (35*21로 조정된 이미지)




```
['유미의 세포들컷툰, 이동건',
 '신의 탑, SIU',
 '신의 언어, 장래혁',
 '열렙전사, 김세훈',
 '외모지상주의, 박태준',
 '패밀리 사이즈, 남지은&김인호',
 '에이머, 구동인',
 '호랑이형님, 이상규',
 '윈드브레이커, 조용석']
```

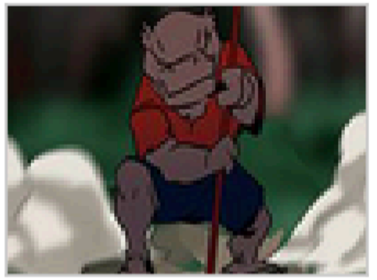
	id	proba
20	이동건	0.93854
1	SIU	0.0136306
23	장래혁	0.0119081
7	김세훈	0.0112274
12	박태준	0.00707228
8	김인호	0.00650305
4	구동인	0.00319602
21	이상규	0.0015323
27	조용석	0.0014997

input image (아래는 이거랑 비슷한 상위 10개 이미지)

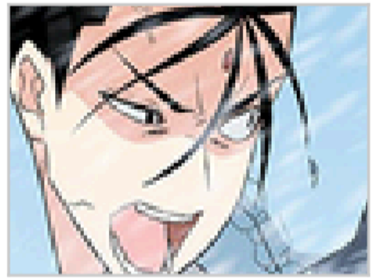


1 & 0.9385395050048828

2 & 0.013630582951009274

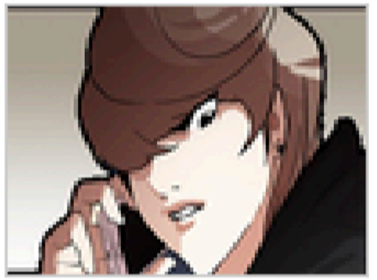


3 & 0.011908140964806080



4 & 0.011227411217987537

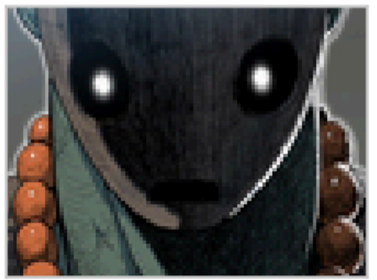
5 & 0.007072276435792446



6 & 0.006503047421574593

7 & 0.003196024103090167

8 & 0.0015322951367124915



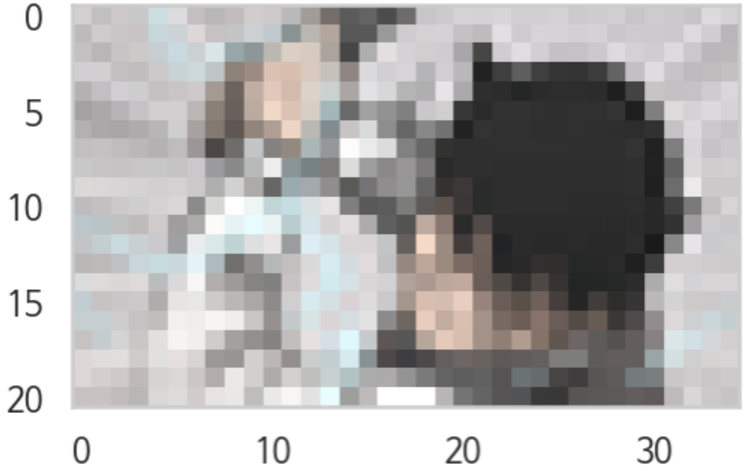
9 & 0.001499698031693697

Classification Result

인풋데이터 : 문래빗

작가명 : 이난(라벨에 없는 데이터)

테스트로 사용한 이미지 (35*21로 조정된 이미지)

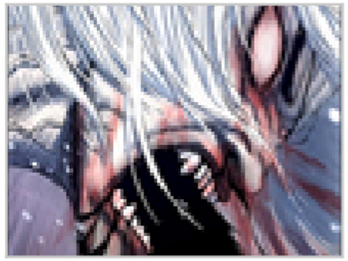



```
['호랑이형님, 이상규',
 '신의 탑, SIU',
 '윈드브레이커, 조용석',
 '다이스(DICE), 윤현석',
 '열렙전사, 김세훈',
 '유미의 세포들컷툰, 이동건',
 '특수 영능력 수사반, 사다함',
 '더 게이머, 성상영&상아',
 '트럼프, 이채은',
```

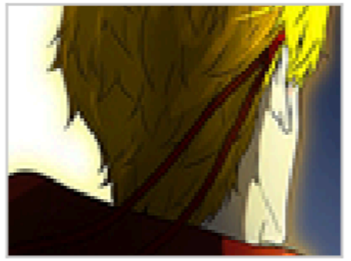
	id	proba
21	이상규	0.623774
1	SIU	0.0746681
27	조용석	0.0551247
19	윤현석	0.0392863
7	김세훈	0.0293384
20	이동건	0.0274684
14	사다함	0.0261647
16	상아	0.0245491
22	이채은	0.0173333

input image (아래는 이거랑 비슷한 상위 10개 이미지)

1 & 0.6237735748291016

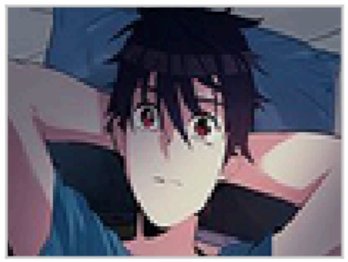


2 & 0.07466805726289749



3 & 0.055124688893556595

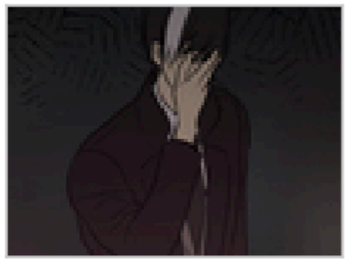
4 & 0.03928627073764801



5 & 0.029338376596570015

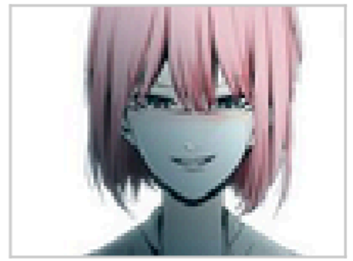
6 & 0.027468366548419

7 & 0.026164650917053223



8 & 0.024549106135964394

9 & 0.017333269119262695



프로젝트 회고

잘한 점	모두 합심하여 CNN 하이퍼파라미터 튜닝 열심히 시도함		
	Trello 활용	빠른 주제 재설정	💸 구글 코랩 사용 💸
아쉬운 점	댓글 활용하지 못한 것	accuracy를 더 높이지 못한 점	
배운 점	CNN 모델		
궁금증	CNN 앙상블 기법	전이학습	이미지의 개수가 같아야하는지
향후 계획	댓글 자연어 처리	추천시스템	

참고문헌

- ✓ https://www.slideshare.net/leeseungeun/cnn-vgg-72164295
- ✓ https://nittaku.tistory.com/264
- ✓ https://excelsior-cjh.tistory.com/180
- ✓ https://tykimos.github.io/2017/03/25/Fit_Talk/
- ✓ https://excelsior-cjh.tistory.com/m/79?category=1013831
- ✓ https://pythonkim.tistory.com/52
- ✓ http://taewan.kim/post/cnn/
- ✓ https://blog.naver.com/samsjang/220908155111
- ✓ https://tykimos.github.io/2017/01/27/CNN_Layer_Talk/
- ✓ https://kevinthegrey.tistory.com/134?category=793117
- ✓ https://machinelearningmastery.com/improve-deep-learning-performance/
- ✓ https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9
- ✓ https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/
- ✓ https://www.slideshare.net/leeseungeun/cnn-vgg-72164295
- ✓ https://sonofgodcom.wordpress.com/2018/12/31/cnn을-이해해보자-fully-connected-layer는-뭔가/
- ✓ https://stanford.edu/~shervine/l/ko/teaching/cs-230/cheatsheet-convolutional-neural-networks
- ✓ https://excelsior-cjh.tistory.com/180
- ✓ https://sacko.tistory.com/44, https://m.blog.naver.com/laonple/220808903260
- ✓ https://bcho.tistory.com/1156
- ✓ https://brunch.co.kr/@taeboklee/31

감사합니다 (๑˃◡˂๑)