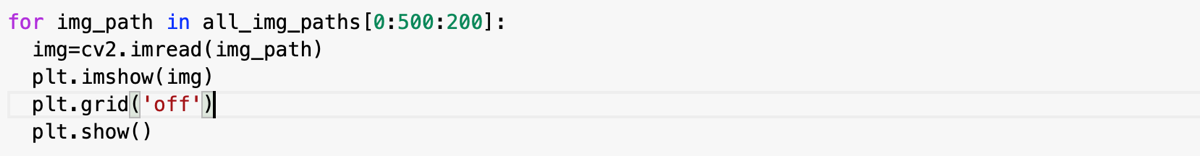
**Term Project**

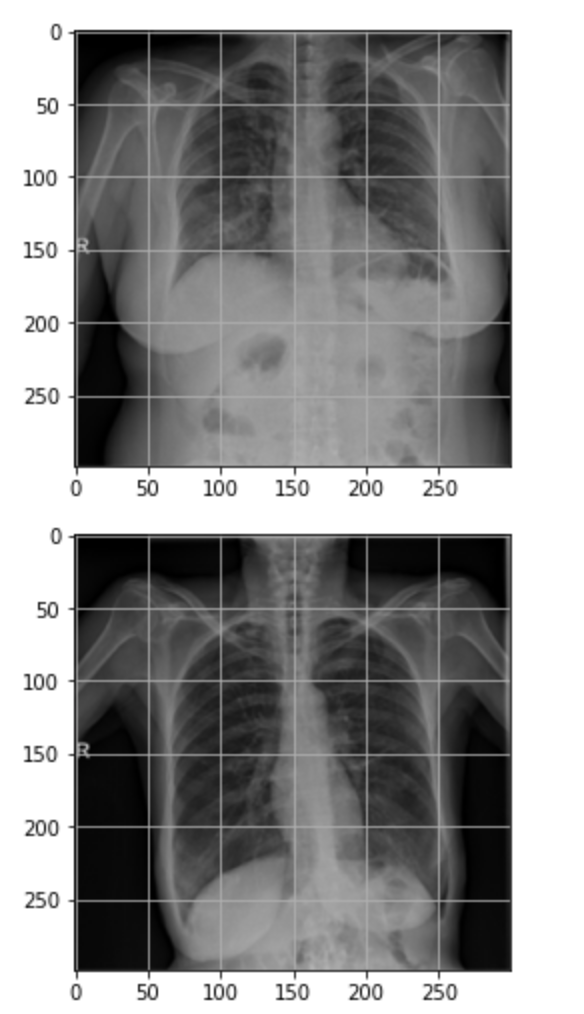
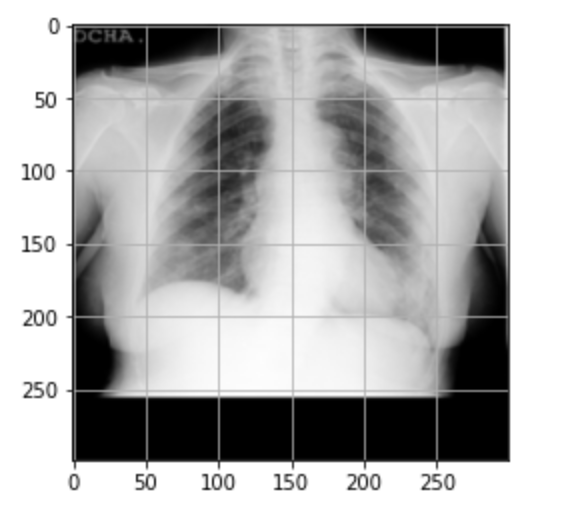
2019150445 신백록

**1. 서론**

1-1. Description of the dataset

앞으로 사용할 데이터는 Kaggle에서 받은 dataset으로 chest x-ray 이미지 데이터이다. 코로나 19 환자의 chest x-ray 이미지 데이터는 약 4000장, Lung Opacity는 6000장, 일반인은 약 1만장, 바이러스성 폐렴 환자의 이미지는 약 1000장으로 도합 2만5천여 장의 이미지가 있다. 각 사진은 300\*300의 pixel을 가진 png파일로 저장 되어있고, 아래는 예시 사진이다.



1-2. Related Works

Kaggle에서 다른 사람들의 code들을 살펴보았다. 코로나 19환자와 일반인을 구분하는 모델, Lung Opacity, 바이러스성 폐렴을 포함한 4개의 class를 구분하는 모델을 만든 약 100여개의 code들이 있었다. 그 중 most vote를 받은 코드들과 dataset이 시간이 지날수록 update 되었기 때문에 최근 코드를 중점으로 보았다. InceptionV3와 vgg16을 활용하여 모델을 define한 code들이었다.

코로나19와 일반인을 구분하는 모델을 만든 코드에서는 InceptionV3를 활용하여 test set에 대하여 96퍼센트의 정확도를 얻은 모델을 찾아볼 수 있었고, 4개의 class를 구분하는 모델에서는 Vgg16과 fine tuning을 활용하여 test set의 accuracy가 91퍼센트 정도 나온 것을 볼 수 있었다.(<https://www.kaggle.com/matheushms/covid-19-chest-x-ray-classificator>)

1-3. 연구 주제와 목적

코로나 19 환자와 일반인을 구분하는 모델을 먼저 만들고, 코로나 19 환자와 일반인, Lung Opacity, 그리고 바이러스성 폐렴 환자를 구분하는 모델을 만들고자 한다. 데이터는 2만5천여 장의 이미지로 그리 많지 않은 데이터이다. 이것만으로 significant한 결과가 나온다면 데이터를 더 추가해 모델의 도입을 고려해보아도 괜찮을 것이다. 본 연구에서는 significant한 모델이 나와 의사의 진단을 도울 수 있는 것을 목표로 한다.

또한 Related work들보다 accuracy를 더 높이는 데에 초점을 맞출 것이다. 앞에서 정의된 모델들과 다른 많은 모델을 define할 것이고, 그 중 가장 accuracy가 높은 모델을 선택할 것이다.

**2. 본론**

2-1. Data preprocessing

코로나19 환자와 일반인의 data만 이용해 각각의 폴더에 넣은 후, random.shuffle 함수를 통해 폴더 안에 있는 데이터들을 random하게 섞었다. 각각의 directory를 저장하고 다음 코드를 통해 test set, validation set을 각각(층화 추출) 0.15씩 분리했다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

그 후 ImageDataGenerator의 flow\_from\_directory function을 통해 size가 (200,200,3)인 train\_generator, test\_generator, validation\_generator을 정의하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

2-2. 코로나19 환자와 일반인 구분

#Simple Model

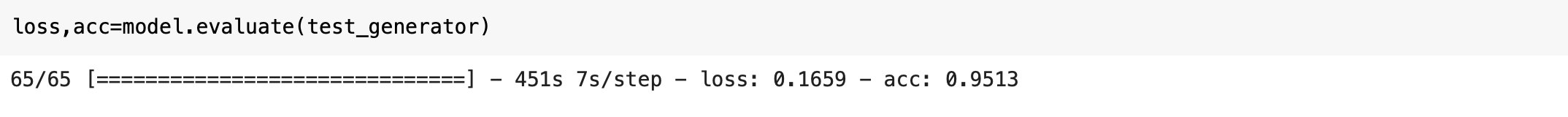
테이블이(가) 표시된 사진

자동 생성된 설명

다음과 같이 Convolution2D와 MaxPooling, GlobalAveragePooling을 사용해 마지막에 Dense(1)로 코로나19 환자와 일반인을 구분하는 간단한 모델을 만들었다.

간단한 모델임에도 test set의 acc는 95퍼센트로 매우 좋게 나왔고, overfitting도 발생하지 않았다.

텍스트이(가) 표시된 사진

자동 생성된 설명

마지막에 Flatten 대신 GlobalAveragePooling을 사용해 related work들의 모델보다 모수가 훨씬 줄었고, Overfitting이 발생하지 않았다. 여기서는 오히려 train acc보다 test acc가 더 높게 나온 걸 보아 제대로 된 feature들이 잘 안 뽑힌 것 같아 층이 더 깊은 모델을 만들어야 할 필요성을 느꼈다. 따라서 vgg16을 이용한 이전학습과 Data augmentation을 활용하기로 하였다.

# Transfer learning & Data augmentation

Train\_generator을 Data augmentation을 위해 다음과 같이 다시 정의하고,

텍스트이(가) 표시된 사진

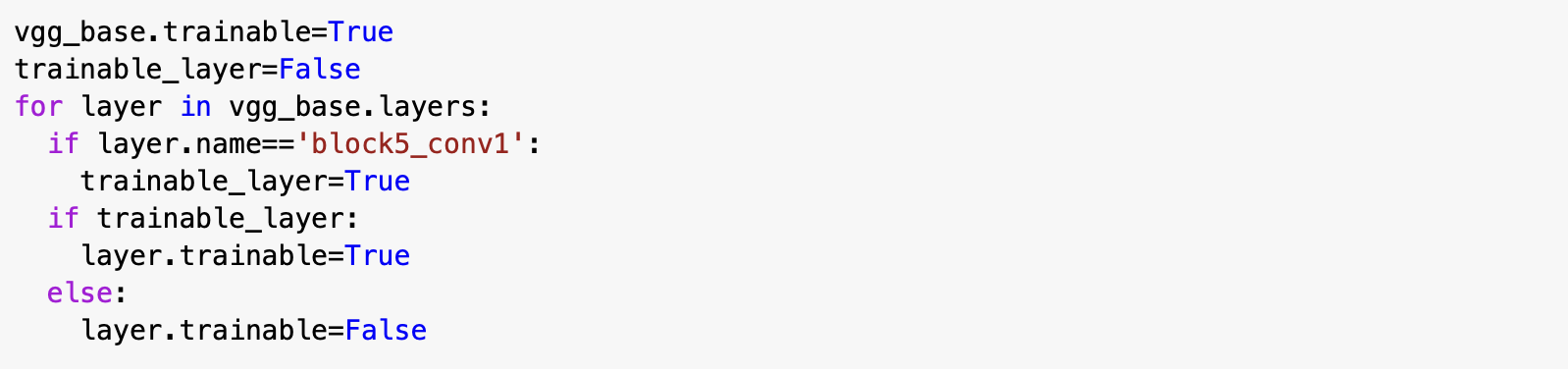
자동 생성된 설명

Vgg16을 불러와서 모델을 다음과 같이 정의하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그 후 fine tuning을 위해 vgg16의 4번째 block까지의 layer를 trainable=False로 설정한 후, 모델을 학습시켰다.



Simple 모델보다 validation accuracy가 훨씬 높게 나온 것을 알 수 있었고, 이를 test set에 evaluate시켰더니 대략 97퍼센트의 acc가 나왔고 simple model에 비해 대략 2퍼센트의 성능 향상이 있었다.

Epoch 13/15

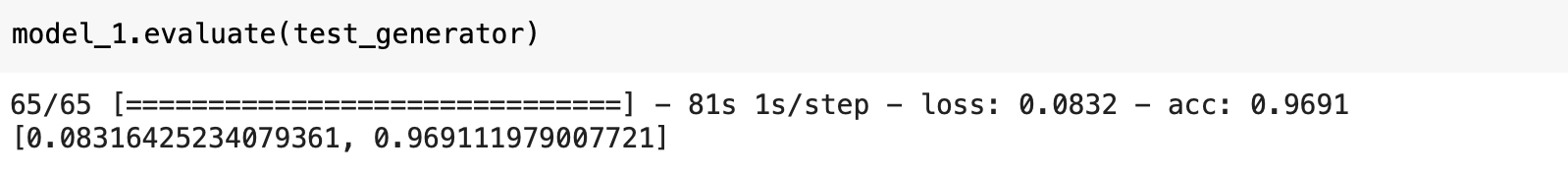
303/303 [==============================] - 570s 2s/step - loss: 0.1509 - acc: 0.9403 - val\_loss: 0.1064 - val\_acc: 0.9618

Epoch 14/15

303/303 [==============================] - 568s 2s/step - loss: 0.1550 - acc: 0.9406 - val\_loss: 0.1047 - val\_acc: 0.9548

Epoch 15/15

303/303 [==============================] - 564s 2s/step - loss: 0.1467 - acc: 0.9437 - val\_loss: 0.0775 - val\_acc: 0.9720



코로나19환자의 data는 3600장, 일반인의 chest data는 10000장으로 그리 큰 데이터 셋이 아님에도 불구하고 만족할만한 성능이 나왔고, 이는 related work의 accuracy와 조금 높거나 비슷한 성능을 보였다. 이제 다음 주제로 넘어가 Lung Opacity, 바이러스성 폐렴 환자를 더한 class 4개를 구분하는 모델을 만들어보았다.

2-3. 코로나 19 환자와 일반인, Lung Opacity, Viral Pneumonia구분

#Simple Model

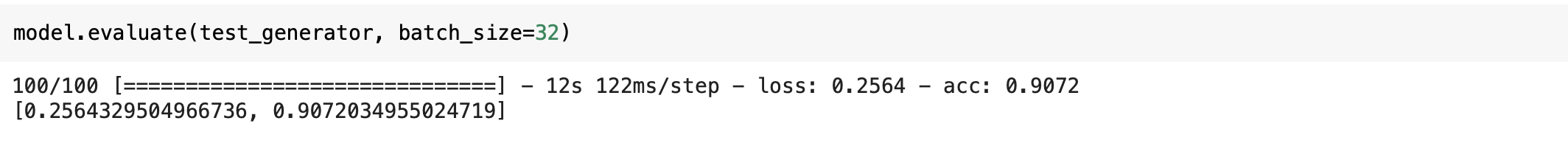
테이블이(가) 표시된 사진

자동 생성된 설명

전과 똑같이 추가된 class들에 대해 validation과 test set을 0.15씩 나누고, (200,200,3)의 size로 ImageDataGenerator을 통해 data를 불러왔다. 그 후 출력층을 Dense(4, activation=’softmax’)로 변경한 것만 빼고 1번과 똑같은 모델을 simple model로 설정하고, 학습시켰다.

텍스트이(가) 표시된 사진

자동 생성된 설명

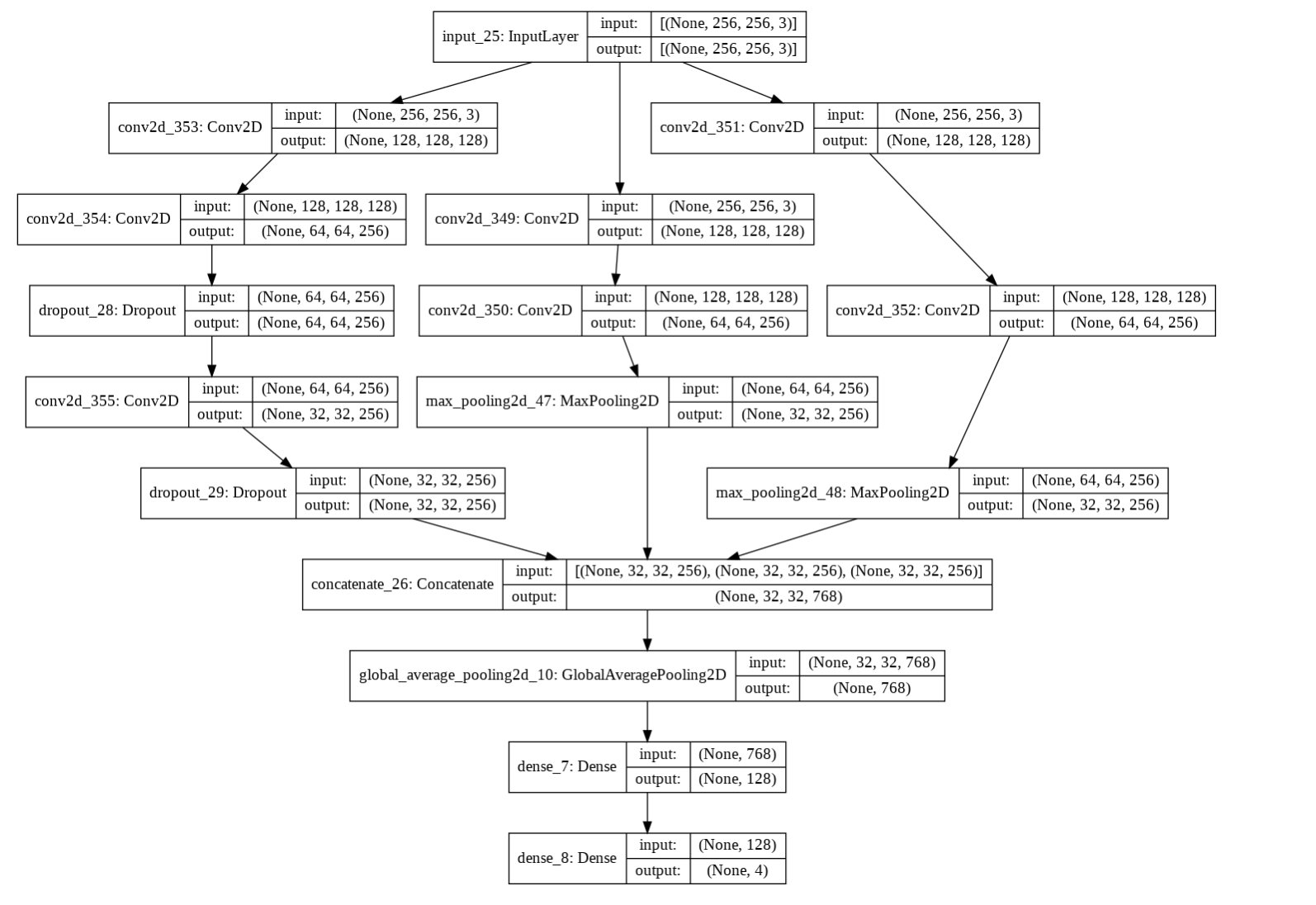


1번과 똑같이 overfitting이 발생하지는 않았고, train acc는 89퍼센트, validation과 test accuracy는 90퍼센트 정도 나왔다. 이제 overfitting을 방지하면서 accuracy를 더 높이는 방법으로 모델을 고안해야할 것이다.

#병렬형 CNN

Branch\_a는 (1,1)의 kernel로 channel의 정보를 뽑아내고자 하였고, branch\_b는 (3,3) & (1,1)의 kernel로 (3,3)로 공간정보의 feature을 뽑아낸 후 channel의 정보를 가져오고자 하였다. Branch\_c는 (5,5) & (3,3)의 kernel을 이용하였다. 그 후 concatenate를 통해 특성변수들을 합친 후, Dense(4, activation=’softmax’)를 출력층으로 하는 model을 define했다.

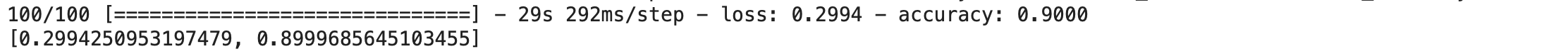




그 후 model을 학습시키고, test set에 evaluate해봤더니 train set의 acc는 92퍼센트인 반면 validation set의 acc는 88퍼센트, test set의 accuracy는 90퍼센트가 나와 약간의 overfitting이 발생했다고 볼 수 있다. 오히려 simple model보다 성능이 떨어진 것을 관찰할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명



이 data는 Chest-Xray 사진들로 모두 흑백 사진으로 이루어져 있다. 그러므로 애초에 굳이 3채널을 쓸 이유가 없었고, 채널 정보를 추출해 낼 이유도 없었던 것 같다. 따라서 성능의 향상 없이 단순히 모수의 개수만 늘어난 것으로 파악된다.

generator에서 data를 흑백(channel=1)로 설정한 후, input size의 channel만 변경하고 같은 병렬형 CNN 모델에 적용시켜보았다. 전의 모델, simple model보다 대략 1퍼센트의 성능 향상이 있었고, overfitting도 channel=3 모델보다는 나아진 모습을 볼 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

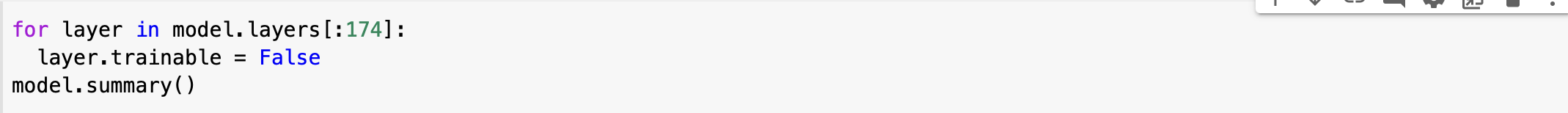
#resnet50을 이용한 이전학습과 residual connection

Resnet이 잔차연결을 이용해서 학습된 model이라는 것을 이용해 resnet50을 이용한 이전학습을 실시하였다. 이에 pixel이 반으로 줄어들 때마다 직접 잔차연결을 또 실시하였다. Overfitting을 방지하기 위해 곳곳에 Dropout을 넣고, kernel\_initializer을 설정하였다. 마지막엔 Flatten 대신 GlobalAveragePooling2D를 이용해 모수를 줄였다.

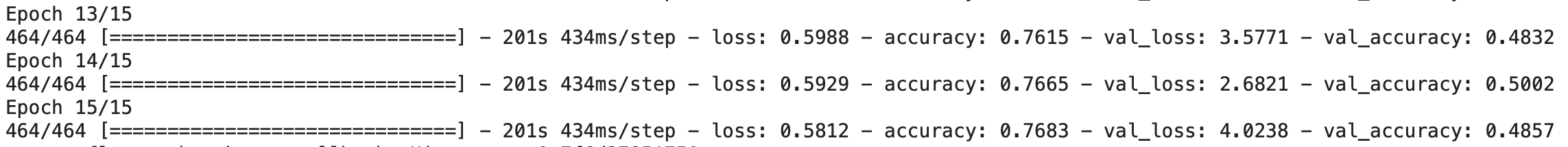
텍스트이(가) 표시된 사진

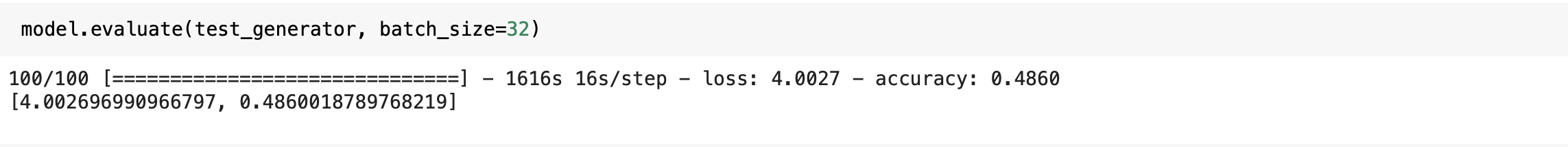
자동 생성된 설명

 그랬더니 Layer가 거의 200개에 달했고, resnet50의 weight을 이용하기 위해 resnet50까지 layer의 trainable=False를 설정하였다.



모델을 학습시키고, 결과를 확인했더니 train set의 acc도 76퍼센트로 낮아졌고, validation과 test set의 acc도 50퍼센트대로 전보다 훨씬 안 좋은 결과를 가져다 주었다.





Train set과 Test set 모두의 성능이 좋지 않은 이유를 해석해 보았다.

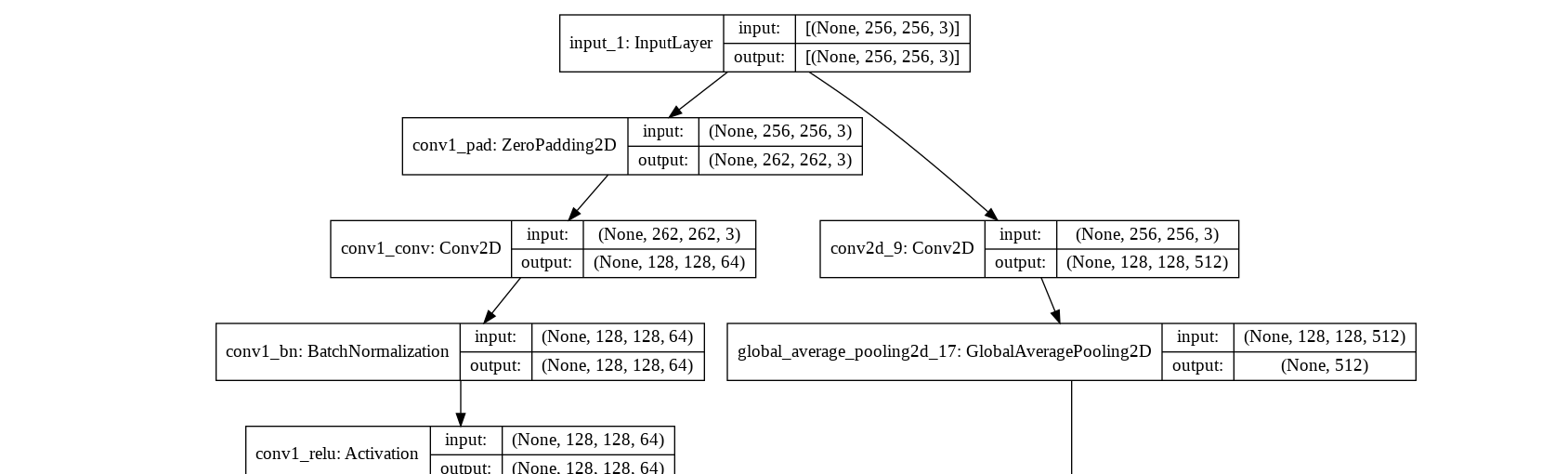
1번. Resnet50의 모수는 Non-trainable로 설정했음에도 추가한 residual connection때문에 Trainable한 모수가 천만개에 달한다. (data는 30000개)

2번. Simple model일 때에 비해 성능이 많이 떨어진 걸 보아 잔차연결이 계속 일어났음에도 층이 너무 깊어진 것 같다.

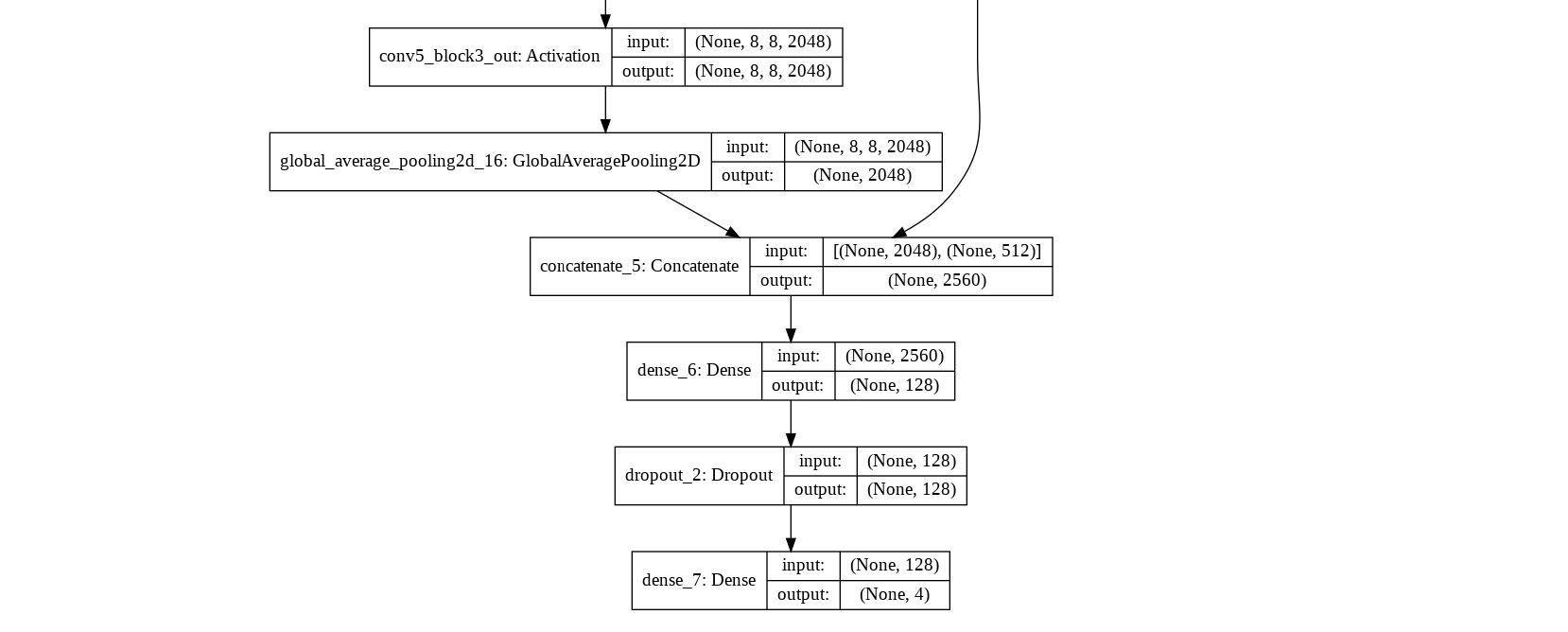
3번. Class 4개를 구분하는 모델에서 굳이 upsampling을 해서 pixel의 개수를 다시 키울 이유가 없다. 모수가 늘어나 오히려 성능이 나빠지는 결과를 낳은 것 같다.

이 모든 현상들을 해결하기 위해 다시 한 번 residual connection을 이용한 model을 만들었다.

#Final model

Resnet50을 그대로 이용하되, 잔차연결을 저렇게 많이 하지 않고 Input을 바로 출력층의 직전 층과 연결하는 model을 생각했다. 그렇다면 굳이 upsampling을 해서 잔차연결을 무리하게 할 필요도 없을 것이고, 저렇게 많은 모수를 다 추정해야 할 필요도 없을 것이라고 생각했다.

Resnet50의 층이 너무 깊어 모델의 사진을 잘랐다. 왼쪽이 resnet50, 오른쪽이 잔차연결을 위한 층이다. Input과 출력층 직전층과 바로 연결하고 싶었지만 size를 맞추기 위해 convolution층 하나와 pooling층 하나를 두었다. 저렇게 쭉 모델이 이어져서 마지막에 다음과 같이 concatenate를 하고, Dense(4)를 통해 class를 추정한다.



이 모델을 train 시켜보았더니 train acc는 97.5퍼센트, validation acc와 test accs는 94퍼센트 정도 나왔다. 약간의 오버피팅이 있었지만, 전 모델보다 확실히 성능 향상이 된 모습을 볼 수 있고, simple model 보다 대략 4퍼센트 정도의 성능 향상이 있었던 것을 알 수 있다. 또한 Related work의 accuracy인 91퍼센트보다 무려 3퍼센트의 성능 향상이 있었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**3. 결론**

3-1. Summary 및 첨언

코로나 19 환자와 일반인을 구분하는 final model에선 정확도 97퍼센트, 코로나 19 환자와 일반인, Lung Opacity, 바이러스성 폐렴을 구분하는 final model에선 정확도 94퍼센트가 나왔다. 이는 Related work의 모델들보다 좋은 성능으로, 괜찮은 모델을 define했다고 할 수 있다. 실제 코로나 19 검사에 쓰이는 PCR 중에서 가장 정확도가 높다는 비인두도말 PCR은 민감도 98퍼센트, 특이도 100퍼센트의 성능을 보인다는 것을 생각해봤을 때, 97퍼센트의 정확도를 가지는 이 model은 결코 낮은 성능인 건 아니다. 이 모델들은 실제 병원에서 충분히 쓰일 수 있을 정도의 성능을 가지고 있다고 생각한다. 물론 여기에 data가 훨씬 많이 추가되어야 하겠지만, AI의 성능이 의사의 진단 성능, 약품의 진단 성능보다 낮다고 할 정도는 아니라는 것은 보여준다. 지금도 이미 상용화 되긴 했지만, 의사가 인공지능 모델의 도움을 받아 더 좋은 진단을 할 수 있는 사회가 되길 바란다.