

과목명	IoT소프트웨어응용			
평가 내용	개인 피드백 보고서			
학생 정보	학번	201904020	이름	윤건용

1. 안개 감지 모델 생성중 1000에폭으로 돌렸지만 정확도가 낮음

-> 찾아보니 과다한 학습률은 과적합을 이르킴, 100으로 조정하니 90%의 정확도를 보여줌

2. 최적의 에폭을 찾기 위해 GPT검색

- 러닝 모델의 최적의 에폭 값을 찾기 위해서는 다음과 같은 기법들을 사용할 수 있습니다:
- Early Stopping: Early stopping은 검증 데이터에 대한 성능이 더 이상 개선되지 않을 때 학습을 중단하는 기법입니다. 즉, 모델이 과적합이 시작되는 시점을 자동으로 감지하여 학습을 멈춥니다. Keras에서는 EarlyStopping 콜백을 제공하여 이 기법을 쉽게 사용할 수 있습니다.

code.py

- `early_stopping = EarlyStopping(monitor='val_loss', patience=10)`
- `model.fit(X_train, Y_train, validation_data=(X_val, Y_val), epochs=2000, callbacks=[early_stopping])`
- 위 코드에서 monitor는 감시할 지표를, patience는 성능이 개선되지 않는 에폭을 얼마나 기다릴지를 나타냅니다.

- 답변으로 얻은 코드를 적용시켜 보아도 실행불가

3. 과다한 에폭수로 돌릴 때 이를 해결하기 위해서 학습하던 도중 checkpoint를 만들어 당시 모델을 저장할 수 있게 하는 방법을 찾음

```
model_checkpoint = ModelCheckpoint('best_model_epoch1000.h5', monitor='val_loss',
save_best_only=True)
history = model.fit(train_data, validation_data=valid_data, epochs=1000,
callbacks=[model_checkpoint])
```

- 괜찮은 방법 같음

4. 안개 유무 확인 모델을 만들기 위해서 안개 사진 데이터를 구할 필요가 있었음, keggel에서도 충분한 데이터가 있지 않아 문제가 있었음
- 데이터 부족 문제를 해결할수 있는 방법으로 전이학습이 있는데 전이학습이란, 학습 데이터가 부족한 분야의 모델 구축을 위해 데이터가 풍부한 분야에서 훈련된 모델을 재사용하는 머신러닝 학습 기법이다. 하지만 충분한 학습을 진행할 수 있는 환경이 없어 새로운 모델을 만드는 것은 불가능하였다.
 - 다른 여러 가지 방법 또한 많았지만 이러한 방법을 실행하기에는 어려움이 있어 사진을 찾아 데이터를 모아주어 해결하였다.

5. 실행환경 구축

- 현재 사용하고 있는 개인 컴퓨터의 사양은 CPU: AMD-5600x, GPU: rx6600xt, RAM: 32G로 사용하고 있다. 모델 학습을 위해 GPU 사용을 하려 찾아보니 NVIDIA가 아닌 다른 제조사의 그래픽카드는 쿠다를 지원 안하고 사용을 할 수 있는 방법은 있지만 그 방법 또한 좋은 방법이 아니라고 한다. 딥러닝을 공부하는 사람은 무조건 NVIDIA 그래픽카드를 사야 한다는 것을 깨달았다.

6. 좋은 기회를 얻어 리눅스 서버를 사용할 수 있게 되었다. 처음 외부 서버를 사용하는거여서 사용법에 대해 많은 공부가 필요하였다. 공부하는 과정속에서 GPU 할당법 원하는 일정 코드만 GPU를 사용할수 있게하는 코드 등 여러 가지 경험을 할수 있었다. 하지만 처음인 만큼 과정이 순탄하지 않았다. 새로운 가상환경을 만들고 그 가상환경에 내가 원하는 환경을 만들어가는 과정에 문제가 많았다. 파이썬의 버전마다 맞는 텐서플로우의 버전이 존재한다는 것을 알게 되었고, 또한 쿠다 버전에 맞는 텐서플로우가 있다는 것을 알게되었다. 환경 구축을 마치고 코드를 가동하니 이러한 오류가 발생하였다

```
2023-12-13      01:21:12.777253:      W      tensorflow/tsl/framework/bfc_allocator.cc:497]
*****
*****
Traceback (most recent call last):
  File "ADAS.py", line 59, in <module>
    X_train.append(load_image(x, True)) # 이미지를 불러와서 전처리한 후 X_train에 추가
  File "ADAS.py", line 36, in load_image
    image = tf.image.resize(image, image_size, method=tf.image.ResizeMethod.LANCZOS3)
F                                i                                l                                e
"/home/junyong/miniconda3/envs/GY/lib/python3.8/site-packages/tensorflow/python/util/traceback_util.py", line 153, in error_handler
    raise e.with_traceback(filtered_tb) from None
F                                i                                l                                e
"/home/junyong/miniconda3/envs/GY/lib/python3.8/site-packages/tensorflow/python/framework/ops.py", line 6656, in raise_from_not_ok_status
    raise core._status_to_exception(e) from None # pylint: disable=protected-access
tensorflow.python.framework.errors_impl.InternalError: Failed copying input tensor from
/job:localhost/replica:0/task:0/device:CPU:0 to /job:localhost/replica:0/task:0/device:GPU:0 in order to
run ExpandDims: Dst tensor is not initialized. [Op:ExpandDims] name:
```

이 오류에 대해 GPT에 물어보니 이러한 답변을 얻었다.

It seems like you're encountering the same issue as before when attempting to resize an image using TensorFlow, specifically with the LANCZOS3 method. The error indicates a problem copying a tensor from the CPU to the GPU for the ExpandDims operation due to an uninitialized destination tensor.

다음 사항을 다시 확인하면 문제를 해결하는 데 도움이 될 수 있습니다.

입력 데이터 처리: 크기를 조정하기 전에 입력 영상 데이터가 올바르게 로드되고 사전 처리되었는지 확인합니다. 데이터가 예상 형식과 모양으로 되어 있는지 확인합니다.

TensorFlow Session/GPU Handling: 해당되는 경우, TensorFlow 세션이 GPU를 활용할 수 있도록 적절하게 설정되어 있는지, GPU에서 작업에 사용할 수 있는 메모리가 충분한지 확인합니다.

TensorFlow 버전 및 호환성: TensorFlow 버전이 최신이고 사용 중인 작업과 호환되는지 확인합니다. 특정 TensorFlow 버전과 호환성 문제가 있을 수 있습니다.

코드 검토: 관련 코드를 확인합니다. ExpandDims 작동. 관련된 모든 텐서가 제대로 초기화되었는지, 모양이 호환되는지 확인합니다. 때때로 텐서를 잘못 재구성하거나 조작하면 이러한 오류가 발생할 수 있습니다.

크기 조정 방법: 문제가 지속되는지 확인하기 위해 다른 이미지 크기 조정 방법을 사용하거나 크기 조정 단계를 일시적으로 생략하는 것을 고려합니다. 메소드를 변경하면 문제가 LANCZOS3 메소드에만 해당하는지 식별하는 데 도움이 될 수 있습니다.

문제가 지속될 경우 문제가 발생한 영역 주변에 더 많은 컨텍스트나 코드 조각을 제공하면 문제에 대한 통찰력을 높일 수 있고 문제 해결에 도움이 될 수 있습니다.

처음에는 입력데이터가 잘못된줄 알아 다시 데이터셋을 받으며 해결해 보려 노력을 하였으나 같은 오류가 발생, 텐서플로우의 버전 또한 변경해 보았지만 이또한 실패 하였다. 코드의 문제는 CPU로 학습을 진행 하였을 때 정상적으로 가동이 되어 문제가 없다고 생각하였다. 결국 원인은 GPU의 메모리 부족 문제여서 서버사용이 어렵게 되었다.

7. 배치처리를 이용하여 모델의 학습속도와 정확도를 높이려 시도 하였음

```
history = model.fit(
    train_generator,
    steps_per_epoch=7000 // batch_size,
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=1000 // batch_size,
)
```

->
412/437 [=====>..] - ETA: 4:39 - loss: 0.4229 - accuracy: 0.8778
WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least steps_per_epoch * epochs batches (in this case, 437 batches). You may need to use the repeat() function when building your dataset.
WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or

generator can generate at least steps_per_epoch * epochs batches (in this case, 62 batches). You may need to use the repeat() function when building your dataset.
437/437 [=====] - 4763s 11s/step - loss: 0.4229 - accuracy: 0.8778
- val_loss: 0.5358 - val_accuracy: 0.8489

끝까지 학습이 안되는 문제 발생

```
train_generator = train_generator.repeat()
validation_generator = validation_generator.repeat()
```

```
history = model.fit(
    train_generator,
    steps_per_epoch=7000 // batch_size,
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=1000 // batch_size,
)
```

repeat() 함수를 사용하여 해결

8. U-net 모델 공부

- UNET은 ISBI cell tracking challenge 2015 대회에서 등장한 모델
- 첫번째, 인코더(Contract path)의 피쳐맵을 디코더 피쳐맵에 Concat하여 위치 정보전달
- 두번째, 데이터셋의 전처리, 변형(deformation)을 이용하여 데이터 수 증가
- 세번째, 테두리(border line)를 더 잘 분할하기 위해 Weight를 추가한 손실함수(Loss function)
- Contracting Path: 점진적으로 넓은 범위의 이미지 픽셀을 보며 의미정보(Context Information)을 추출
- Bottle Neck: 수축 경로에서 확장 경로로 전환되는 전환 구간
- Expanding Path: 의미정보를 픽셀 위치정보와 결합(Localization)하여 각 픽셀마다 어떤 객체에 속하는지를 구분
- <https://wikidocs.net/148870>