

# Microprocessor Lab Report

## Lab 02: NVIDIA nano Classification

소속: 컴퓨터정보공학부

학번: 2022202090

이름: 남재현



**MPLAB**  
Media Processing Laboratory

이번 실험은 엄지척(좋아요 손모양)과 엄지다운(싫어요 손모양)을 구분 하는 모델과 얼굴의 4 가지 표정을 구분하는 모델로 라즈베리파이에서 학습하게 만드는 목적으로 진행할 것이다.

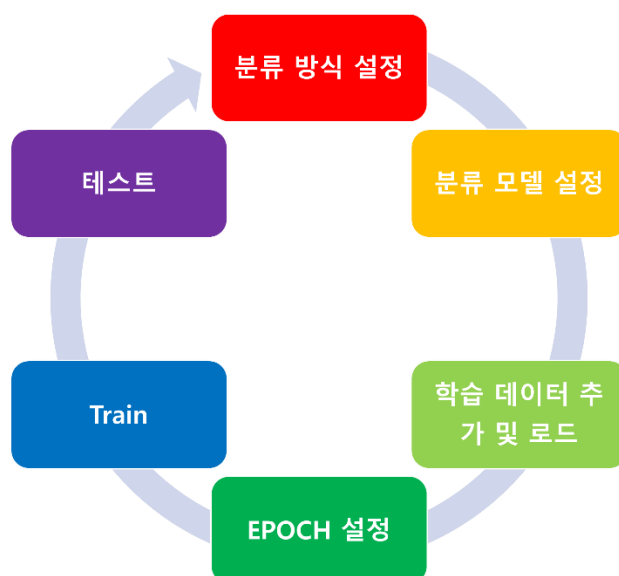
실험 준비물 : 실행환경이 준비된 라즈베리파이, 연결선(5pin to usb), 노트북, 카메라(연결선 포함), 라즈베리파이 dc 전원 연결선 1.5v

## 목차

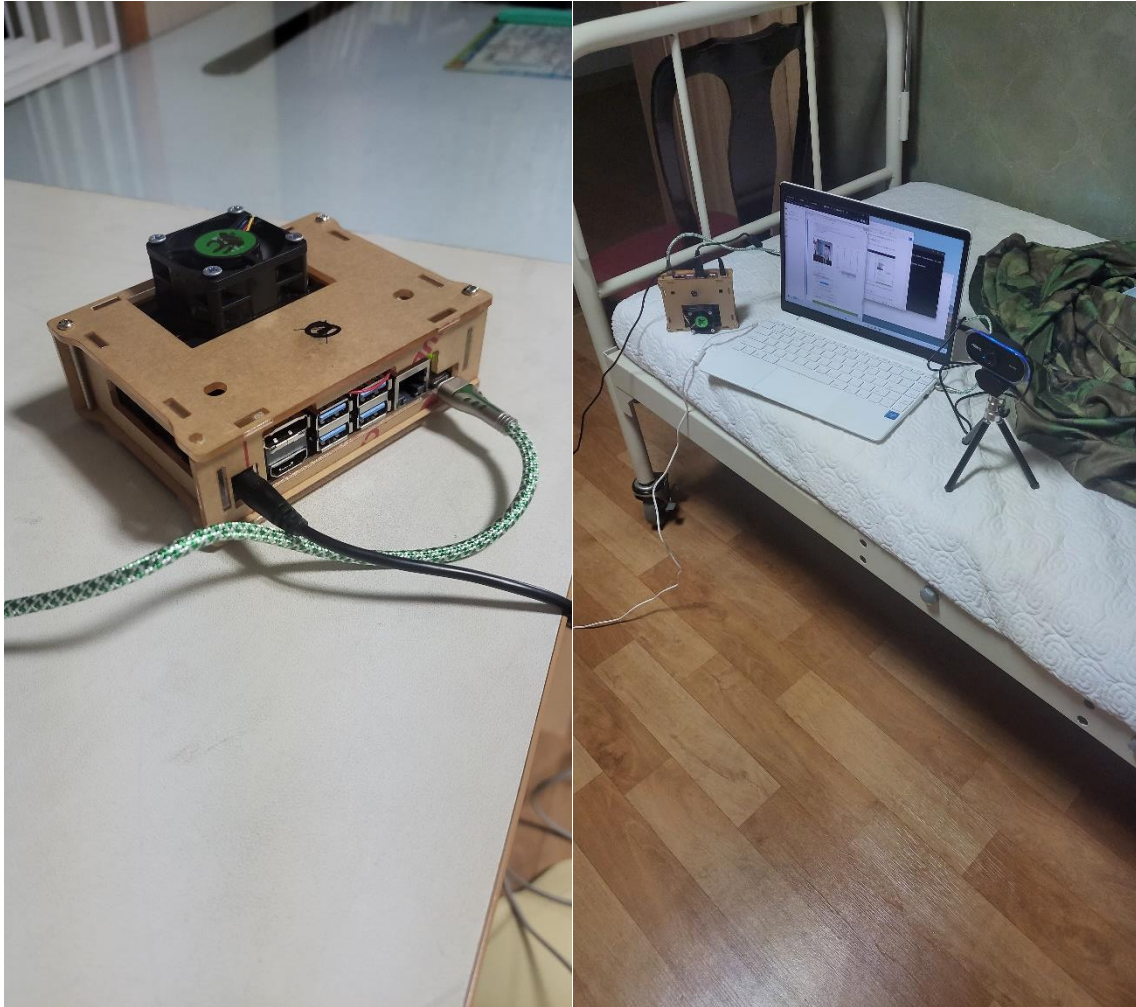


\* 마치며 느낀점과 퀴즈 요약 서술

### - 모델 훈련 순서



## 1. 실행환경 구성



라즈베리파이를 연결한 모습이다. (전원선과 카메라 선, usb 선을 연결하였다.)

```
root@nano:/nvdli-nano
PS C:\Users\nwlefe> ssh nano@192.168.55.1
ssh: connect to host 192.168.55.1 port 22: Connection timed out
PS C:\Users\nwlefe> ssh nano@192.168.55.1
nano@192.168.55.1's password:
Permission denied, please try again.
nano@192.168.55.1's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.9.201-tegra aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

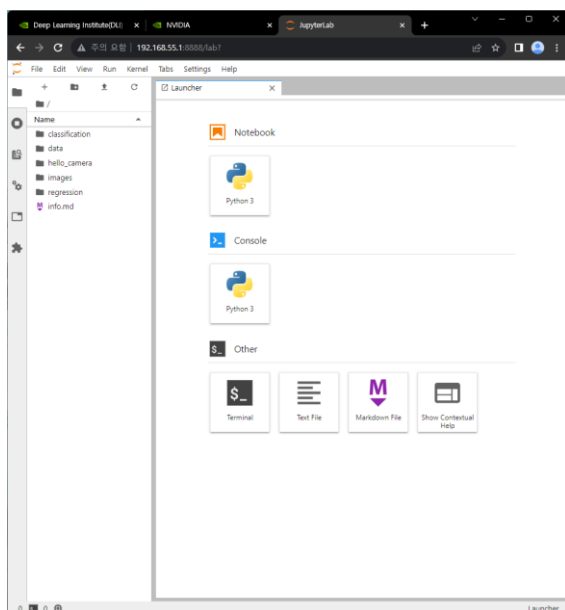
To restore this content, you can run the 'unminimize' command.

16 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Mon Aug 30 18:28:14 2021 from 192.168.55.100
nano@nano:~$ ls
CSI-Camera  docker_dli_run.sh  Downloads      jetson-fan-ctl  nvdli-data  Public  Videos
Desktop    Documents          examples.desktop  Music          Pictures    Templates
nano@nano:~$ ./docker_dli_run.sh
docker: Error response from daemon: error gathering device information while adding custom device "/dev/video0": no such
file or directory.
nano@nano:~$ ./docker_dli_run.sh
allow 10 sec for JupyterLab to start @ http://192.168.55.1:8888 (password dlinano)
JupyterLab logging location: /var/log/jupyter.log (inside the container)
root@nano:/nvdli-nano#
```

터미널(Windows PowerShell)을 이용하여 라즈베리파이에 접속한 후 도커를 실행한다.<sup>1</sup>

(ssh nano@192.168.55.1 && ./docker\_dli\_run.sh)



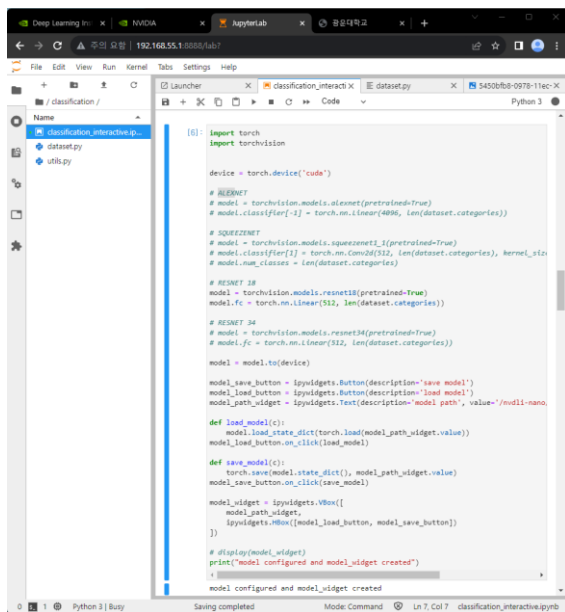
주피터 노트북에 접속하면서 실행환경 구성이 끝났다.

---

<sup>1</sup> Docker 출처 (새로운 Zet Nano의 AI 시작하기) : <https://courses.nvidia.com/courses/course-v1:DLI+S-RX-02+V2/about>

## 2. RESNET 18 엄지 방향 예측 모델

분류	thumbs_up, thumbs_down
모델	RESNET 18
epoch	100 회
훈련 데이터 개수	70 장 (thumbs_up : 35 장, thumbs_down: 35 장)
정확도 평균	100%
요약	학습이 잘 되었다.



```
[1]: import torch
import torchvision

device = torch.device('cuda')

# RESNET
# model = torchvision.models.resnet18(pretrained=True)
# model.classifier[1] = torch.nn.Linear(4096, len(dataset.categories))

# SQUEEZENET
# model = torchvision.models.squeezenet1_1(pretrained=True)
# model.classifier[1] = torch.nn.Conv2d(128, len(dataset.categories), kernel_size=1)
# model.num_classes = len(dataset.categories)

# RESNET 18
model = torchvision.models.resnet18(pretrained=True)
model.fc = torch.nn.Linear(512, len(dataset.categories))

# RESNET 34
# model = torchvision.models.resnet34(pretrained=True)
# model.fc = torch.nn.Linear(512, len(dataset.categories))

model = model.to(device)

model_save_button = ipynbwidgets.Button(description='save model')
model_load_button = ipynbwidgets.Button(description='load model')
model_path_widget = ipynbwidgets.Text(description='model path', value='./mdl1-mano')

def load_model(c):
    model.load_state_dict(torch.load(model_path_widget.value))
    model_load_button.on_click(load_model)

def save_model(c):
    torch.save(model.state_dict(), model_path_widget.value)
    model_save_button.on_click(save_model)

model_widget = ipynbwidgets.VBox([
    model_load_button,
    model_path_widget,
    model_save_button
])

# display(model_widget)
print("model configured and model_widget created")

model configured and model_widget created
```

RESNET 18 을 이용하여 모델 생성한 모습이다.

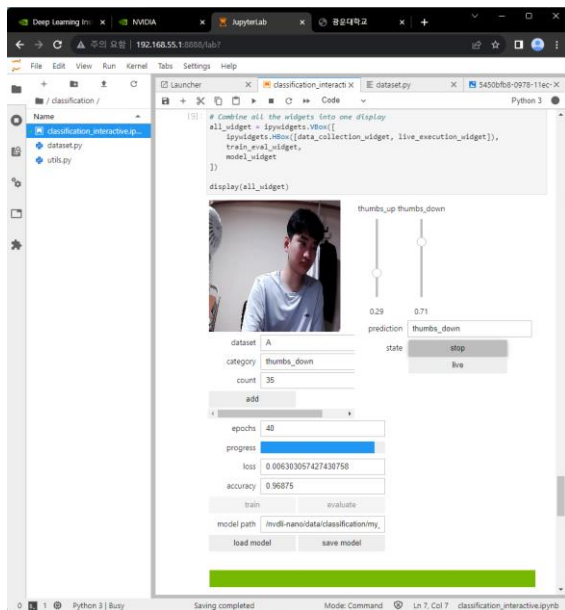
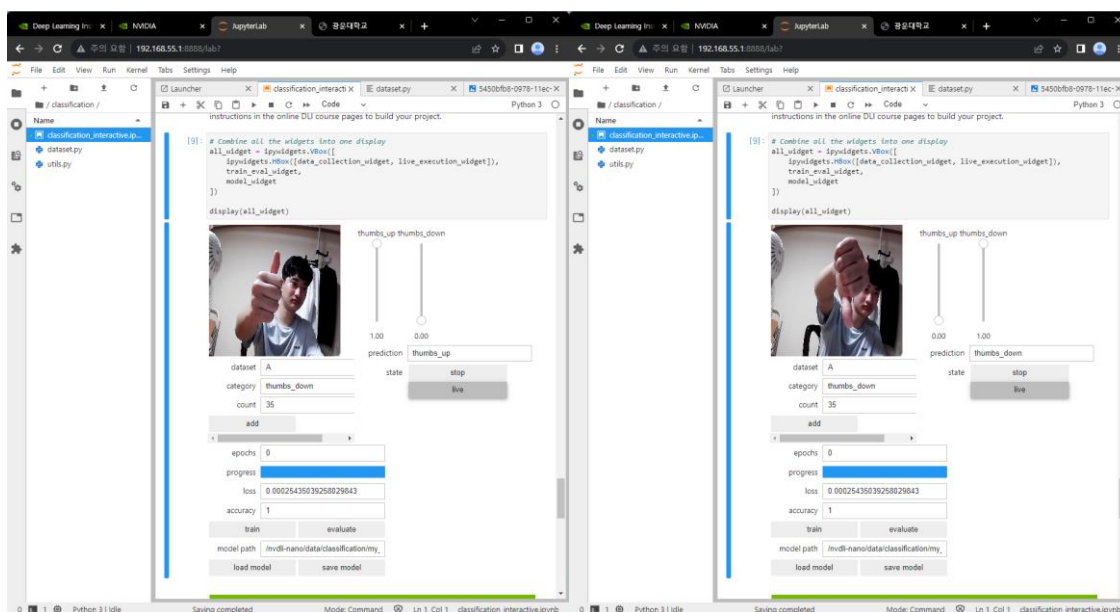


사진 총 70 장으로 epoch 100 으로 학습

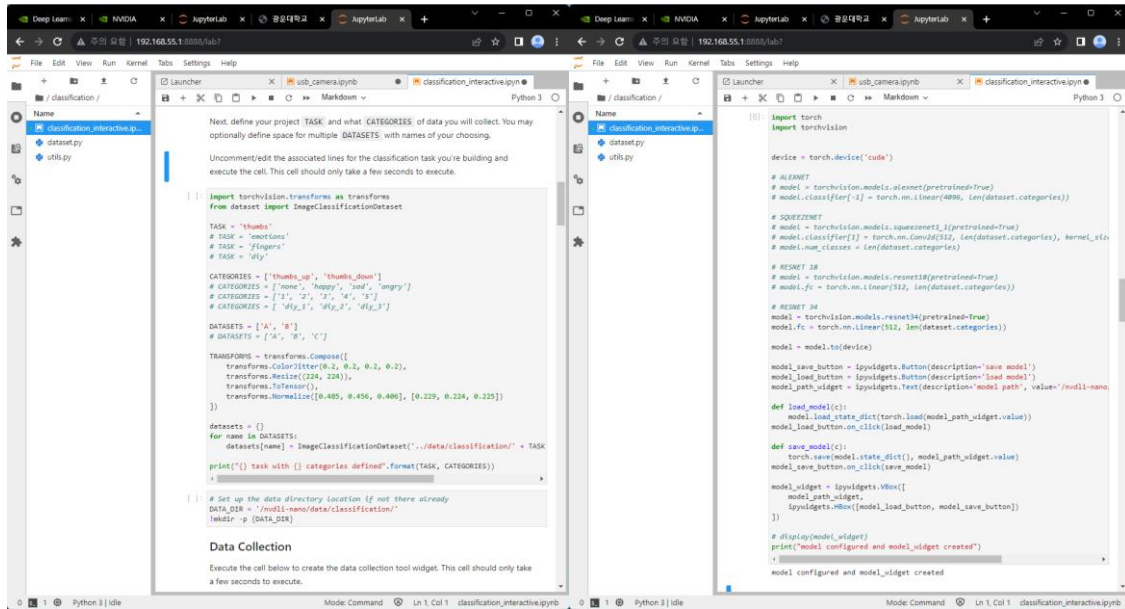


학습 후 정상적으로 분류되는 것을 확인할 수 있다.

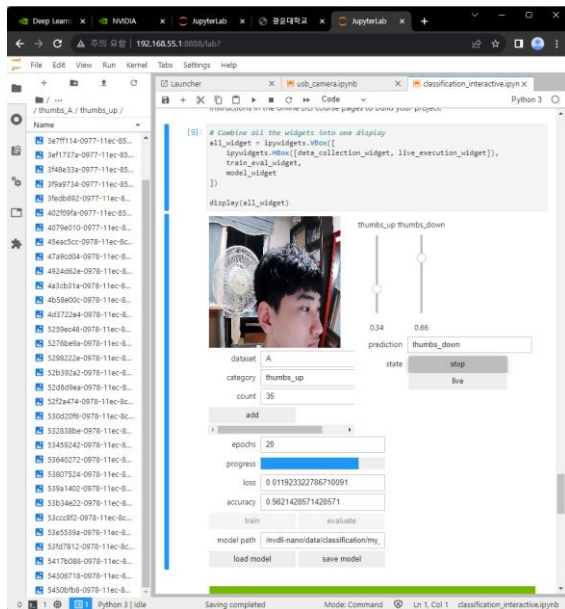
### 3. RESNET 34 엄지 방향 예측 모델

분류	thumbs_up, thumbs_down
모델	RESNET 34
epoch	20 회

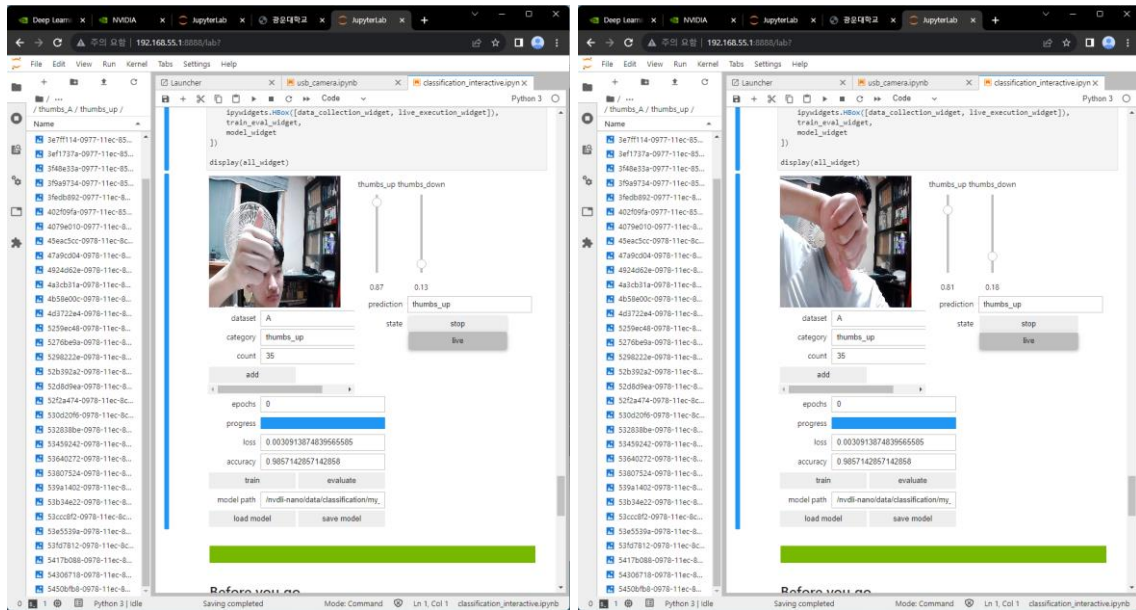
훈련 데이터 개수	70 장 (thumbs_up : 35 장, thumbs_down: 35 장)
정확도 평균	약 53%
요약	thumbs_down 을 잘 학습하지 못하였다.



엄지 방향을 분류하는 RESNET 34 모델을 사용하였다.



epochs 20 으로 학습하였다.

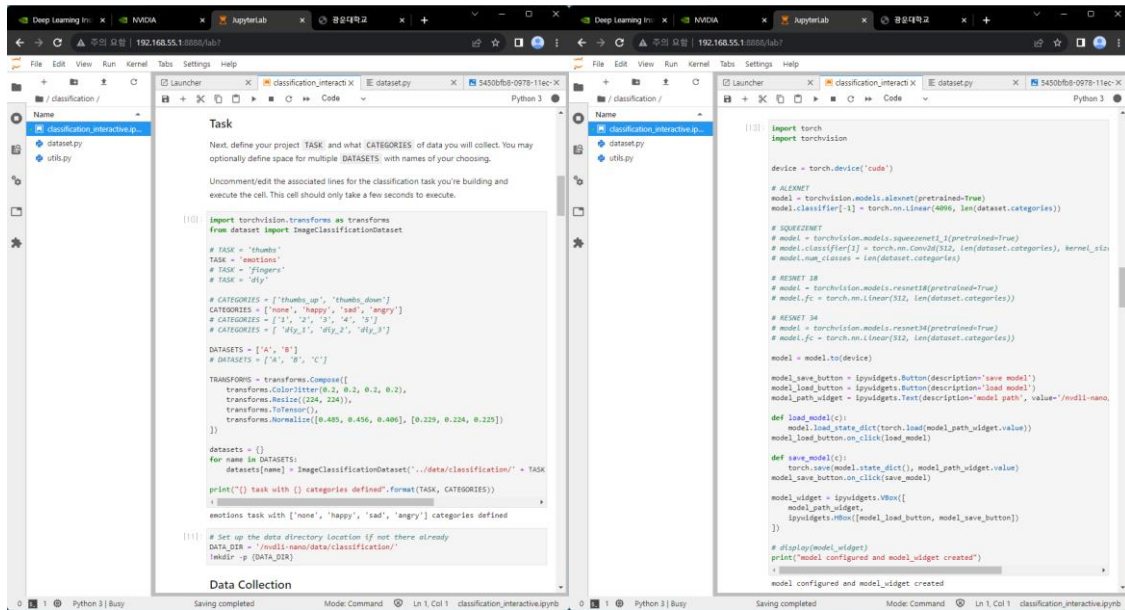


epochs 20 으로 진행한 결과, 인공지능이 내린 판단이 우유부단한 것 같다. 그리고 thumbs\_down 을 제대로 학습하지 못하였다.

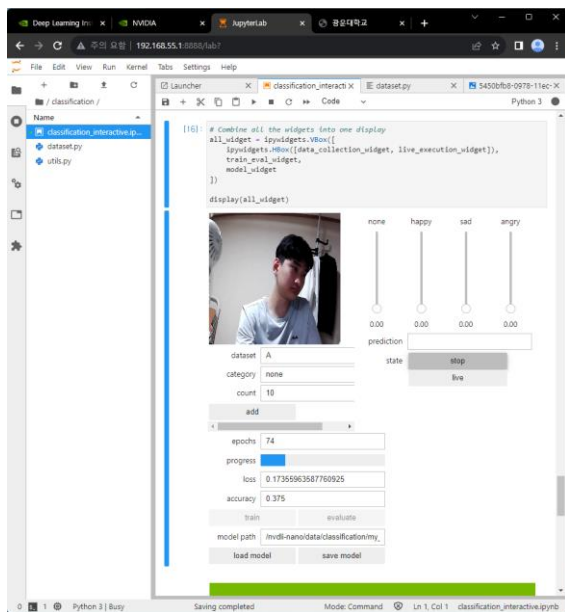
## 4. ALEXNET 얼굴 표정 예측 모델

분류	none, happy, sad, angry
모델	ALEXNET
epoch	100 회
훈련 데이터 개수	40 장 (none : 10 장, happy : 10 장, sad : 10 장, angry : 10 장)
정확도 평균	약 25%
요약	각 분류마다 가능성이 25%인 것으로 보아 표정의 특징을 찾지 못한 것 같다.

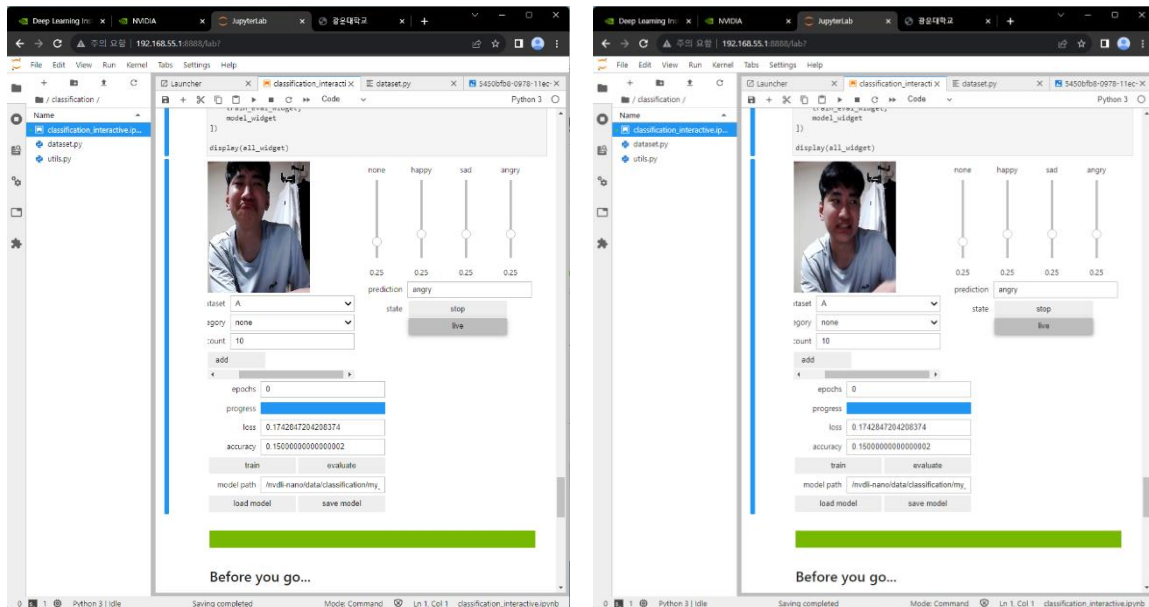
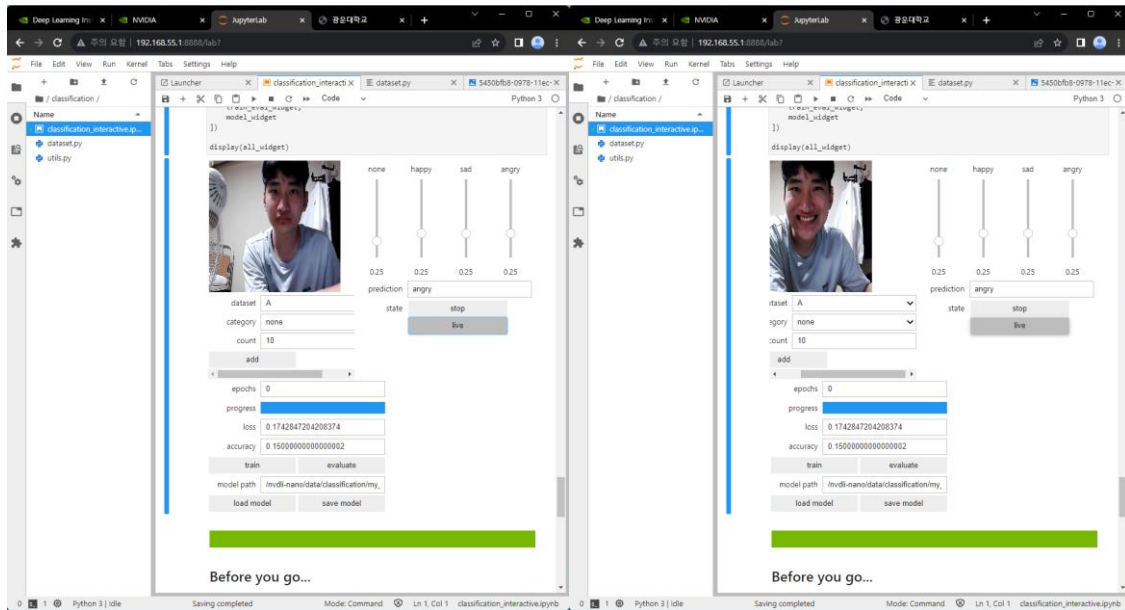




얼굴의 표정을 분류하는 ALEXNET 모델이다. 왜 ALEXNET 을 사용하기로 지정했냐면, 독단적이게 Conv2d 라는 함수를 호출하는 인공지능 모델이기 때문에 지정하였다.



총 40 장의 사진으로 epoch 100 으로 학습하였다.



ALEXNET 으로 학습시킨 결과 네 표정에서 모두 각 표정일 확률이 4 분의 1 로 고르게 나왔다. 학습 데이터가 적은것 같다.

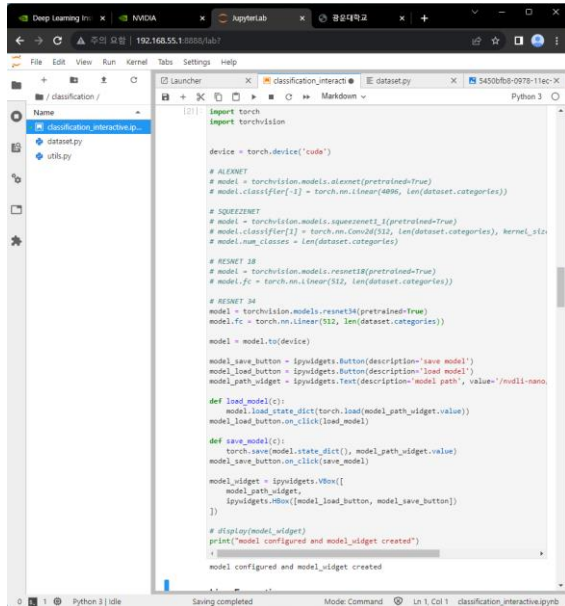
## 5. RESNET 34 얼굴 표정 예측 모델

분류	none, happy, sad, angry
모델	RESNET 34
epoch	40 회
훈련 데이터 개수	104 장 (none : 26 장, happy : 26 장, sad : 26 장, angry : 26 장)
정확도 평균	약 19.5%

## 요약

큰 노력에도 불구하고 인공지능이 표정을 잘 맞추지 못하였다.

마지막으로 RESNET 34 를 이용하여 학습 데이터를 더 모은 뒤 학습시키고 3 번째 모델로 다시 시도해보았다.



```
import torch
import torchvision

device = torch.device("cuda")

# ALEXNET
# model = torchvision.models.alexnet(pretrained=True)
# model.classifier[-1] = torch.nn.Linear(4096, len(dataset.categories))

# SQUEEZENET
# model = torchvision.models.squeezenet1_1(pretrained=True)
# model.classifier[1] = torch.nn.Conv2d(312, len(dataset.categories), kernel_size=1)
# model.num_classes = len(dataset.categories)

# RESNET 18
# model = torchvision.models.resnet18(pretrained=True)
# model.fc = torch.nn.Linear(512, len(dataset.categories))

# RESNET 34
model = torchvision.models.resnet34(pretrained=True)
model.fc = torch.nn.Linear(512, len(dataset.categories))

model = model.to(device)

model_save_button = ipynbwidgets.Button(description='save model')
model_load_button = ipynbwidgets.Button(description='load model')
model_path_widget = ipynbwidgets.Text(description='model path', value='/null:nono')

def load_model():
    model.load_state_dict(torch.load(model_path_widget.value))
    model_load_button.on_click(load_model)

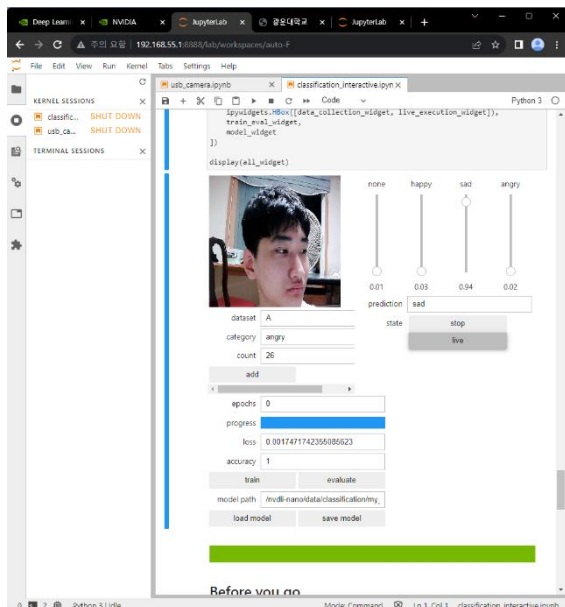
def save_model():
    torch.save(model.state_dict(), model_path_widget.value)
    model_save_button.on_click(save_model)

model_path_widget = ipynbwidgets.VBox([
    model_path_widget,
    ipynbwidgets.HBox([model_load_button, model_save_button])
])

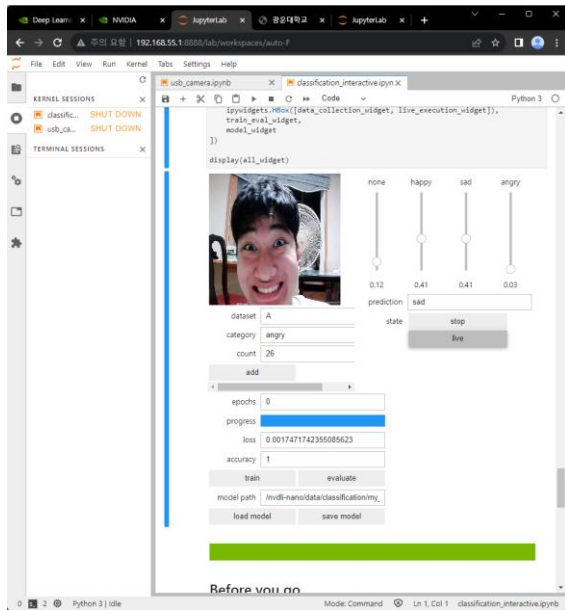
# display(model_widget)
print("model configured and model_widget created")

# =====
model configured and model_widget created
```

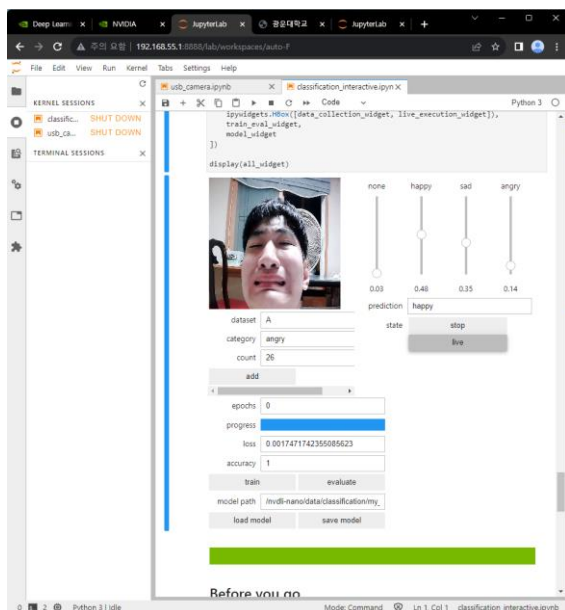
표정이 담긴 사진 26 장 X 4 = 총 104 장을 가지고 epochs 를 40 으로 맞추고 학습을 진행시켰다.



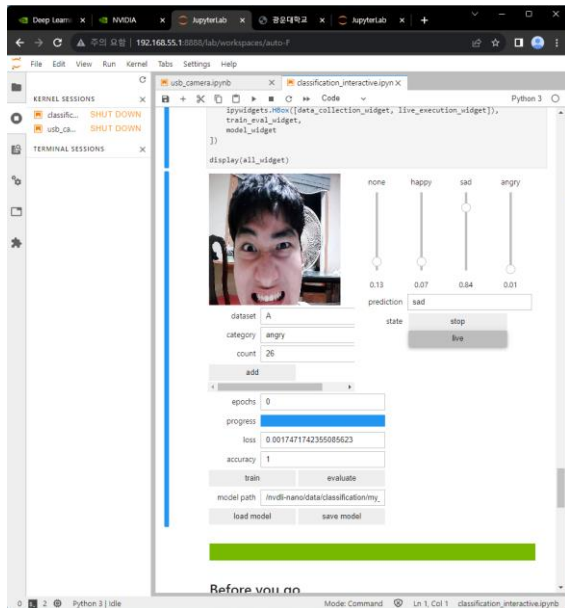
정답은 무표정이다. 인공지능은 슬픈 표정으로 예측하였다.



정답은 웃는 표정이다. 인공지능은 행복한 표정 또는 슬픈 표정 둘 중의 하나일 가능성이 높다고 예측하였다.



정답은 우는 표정이다. 인공지능은 행복한 표정일 가능성이 가장 높을 것으로 예측하고 있다.



정답은 화내는 표정이다. 인공지능은 슬픈 표정일 것으로 예측하고 있다.  
표정을 맞추는 인공지능의 성능은 별로 좋지 않았다.

## 느낀 점

힘들었던 점 : 계속 커널과 연결이 끊기고 주피터가 끊겨서 재시작하고를 반복해서 처음부터 다시하는 일이 자주 일어나 헛수고를 많이했다. 시간도 많이 들고 노력도 많이 들어서 힘들었다.

학습시키면서 느낀 점 : 처음 엄지 방향 모델을 학습시켜보고 다른 모델도 학습이 잘 될 것 같았지만, 생각보다 쉽지 않았다. 사진이 많을수록, epochs 가 많을수록 학습이 잘되는 것 같다. 학습 데이터를 늘리고 epoch 를 더욱 늘리고 학습 모델 4 가지를 모두 구현하는 개선의 여지가 있다.

## Nvidia 사이트 assessment 요약

 Bookmark this page

## Classification

1/1 point (graded)

What is "Classification" in Machine Learning?

☐ Mapping a group of inputs to discrete outputs

☐ Classifying an input to a given set of categories

☐ Predicting labels for input images

☒ All of the above



제출

“분류”란 결과값이 다른 입력들의 묶음을 맵핑하는 것, 주어진 항목들의 집합에 입력을 분류하는 것, 사진 입력들에 대한 라벨을 예측하는 것을 뜻한다.

## Data Collection

1/1 point (graded)

What are the key points to keep in mind when collecting image data?

☐ Try to provide different backgrounds so that model learns essential featur

☐ Vary the light settings, so that model learns color variance.

☐ Collect the data with minimum labeling errors so that the data set will ha

☒ All of the above



제출

✓ Correct (1/1 point)

이미지 데이터를 모을 때에는, 서로 다른 배경에서 제공되어야 하고, 다양한 색을 학습할 수 있도록 광량을 조절해야 하며, 라벨링 오류를 최소화해야한다.

## Classification Steps

1/1 point (graded)

What are the steps followed in the Classification Project?

☐ Train Model->Collect Data->Live Demonstration

☒ Collect Data->Train Model->Live Demonstration

☐ Live Demonstration->Train Data->Collect Data

☐ None of the above



제출

✓ Correct (1/1 point)

**분류 프로젝트 절차 : 데이터 수집 -> 모델 훈련 -> 실시간 테스트**

## Framework

1/1 point (graded)

Which Deep Learning Framework was used to train the classification model?

☐ TensorFlow

☐ TensorRT

☒ PyTorch

☐ MXNET

☐ None of the above



제출

✓ Correct (1/1 point)

**이 분류 모델은 PyTorch 라는 딥러닝 프레임워크가 사용되었다.**

## Transfer Learning

1/1 point (graded)

What is Transfer Learning?

- ☐ A. Use a pre-trained model so that we don't have to train at all
- ☒ B. Use a pre-trained model so that we only need to train on new datasets
- ☐ C. Do not use a pre-trained model, because that will reduce the accuracy
- ☐ D. Both options A and B



제출

✓ Correct (1/1 point)

**“Transfer Learning”이란, 새로운 데이터셋을 학습하기 위해 이미 학습된 모델을 사용하는 것이다.**

## Network Layers

1/1 point (graded)

For our projects, we modified the last layer of a pre-trained ResNet-18 model. V

- ☐ MaxPool layer
- ☐ Convolution Layer
- ☒ Fully Connected Layer
- ☐ Dropout Layer



제출

✓ Correct (1/1 point)

**RestNet-18 모델의 마지막 레이어는 Fully Connected Layer 이다.**



## Data Sets

1/1 point (graded)

We used a PyTorch provided pre-trained ResNet-18 architecture. What dataset

☐ COCO

☒ ImageNet

☐ MNIST

☐ CIFAR-100



제출

✓ Correct (1/1 point)

**이미 학습된 ResNet-18 모델은 ImageNet 이라는 데이터셋으로 훈련되었다.**

## CATEGORIES variable

1/1 point (graded)

There is a global variable named `CATEGORIES` in the notebook. If we add a new dimension of the neural network change for this classification task?

☐ The output dimension will stay the same

☒ The output dimension will increase by 1

☐ The output dimension will increase by 2

☐ The output dimension will increase by 4



제출

✓ Correct (1/1 point)

**CATEGORIES 로 이름 지어진 전역 변수에 새로운 분류 작업이 추가된다면 출력 차원은 1 이 증가한다.**

## Optimizer

1/1 point (graded)

What optimizer was used for training the model?

☐ Stochastic Gradient Descent

☐ Averaged Stochastic Gradient Descent

☒ Adam

☐ AdaGrad

**모델 훈련에 사용된 최적화 도구는 Adam 이다.**

## Numerical Input

1/1 point (graded)

How many classes did the Image Classification "Emotions Project" include?

4



4

제출

**Emotions Project 에 포함된 이미지 분류의 개수는 4 개이다.**