



# Assignment #8

## AATG013/ANT5013 Deep Learning and Computer Vision

**Assigned: Tuesday, 5/9, 2023**

**Due: Tuesday, 5/16, 2023 (by midnight)**

- Colab code 2 set을 사이버캠퍼스에 제출
  - assignment8-1.ipynb
  - assignment8-2.ipynb

code는 colab에서 수행되어 결과를 보여야 합니다.



## Problem [1] PyTorch code: Transfer Learning for multi-class classification problem

**Problem [1]** [30+30=60 points]

지난 과제에서 pre-trained VGG-16 모델을 활용하여 binary classification 문제를 해결하는 transfer learning을 구현하였다. 본 과제에서는 facial emotion recognition 알고리즘을 구현해본다. 이 facial emotion recognition 문제를 pretrained ResNet18 모델을 transfer learning 하여 multi-class classification 을 푸는 문제로 해결할 수 있다. 코드 구현은 지난 과제와 교재의 코드를 참고하여 완성하세요.

본 숙제에서는 Kaggle의 "facial emotion recognition"(<https://www.kaggle.com/datasets/chiragsoni/ferdata>)를 활용하여, 7 classes를 classification하는 문제를 풀어보겠다.

### Multi-class (7-classes) classification problem



Angry



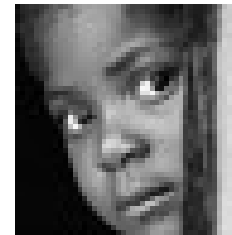
Disgust



fear



happy



neutral



sad



surprise



# Problem [1] PyTorch code: Transfer Learning for multi-class classification problem

**Problem [1]** [30+30=60 points]

Kaggle dataset은 train폴더와 test폴더로 구성되어 있다. Train폴더의 데이터는 학습에 사용하고, test폴더는 validation 용으로 사용하시기 바랍니다.

최종적으로 본과제에서는 구현한 코드와 사용한 train dataset과 validation dataset의 accuracy 비교 그래프를 포함하여 제출하시기 바랍니다.

## Kaggle dataset 가져오기

```
[2] !pip install -q kaggle
    from google.colab import files
    files.upload()
    !mkdir -p ~/.kaggle
    !cp kaggle.json ~/.kaggle/
    !ls ~/.kaggle
    !chmod 600 /root/.kaggle/kaggle.json
```

파일 선택 kaggle.json

- kaggle.json(application/json) - 64 bytes, last modified: 2023. 4. 26. - 100% done

Saving kaggle.json to kaggle.json  
kaggle.json

```
!kaggle datasets download -d chiragsoni/ferdata
!unzip ferdata.zip
```

```
!kaggle datasets download -d chiragsoni/ferdata
```

```
!unzip ferdata.zip
```

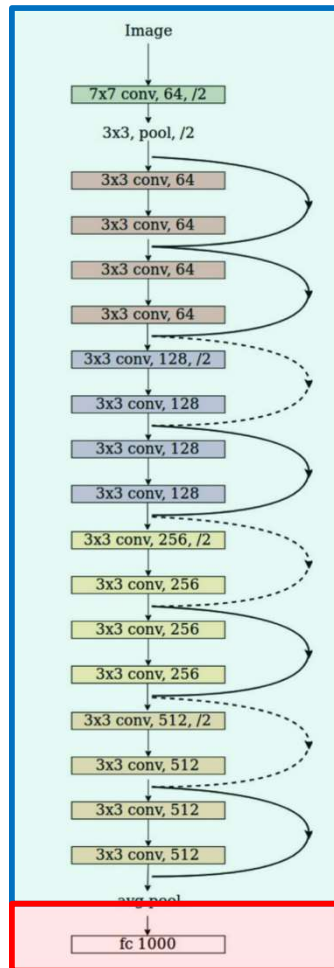


# Problem [1] PyTorch code: Transfer Learning for multi-class classification problem

## Problem [1]

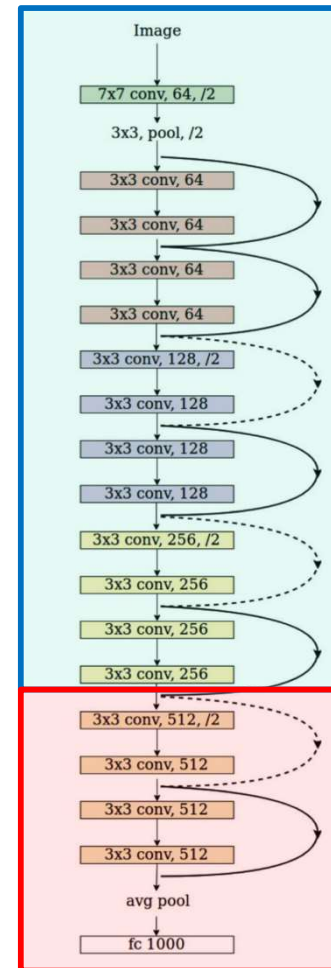
- Pretrained ResNet18 model을 transfer learning을 통해서 facial emotion classification을 수행한다. 본 과제에서는 아래 그림과 같이 2가지 경우에 대한 transfer learning 을 구현하고, train dataset과 validation dataset에 대하여 epoch의 증가에 따른 각 각의 performance를 accuracy graph 로 보이시오. 단 epoch의 수는 colab이 허용하는 한 최대로 한다.

Case 1.



Classification layers만  
재학습

Case 2.



Classification layers과  
마지막 resnet-block까지 재학습

- blue 영역:  
frozen layers
- red 영역:  
trainable layers



# 코드 구현 가이드

## Problem [1]

- 지난 과제부터는 이전 과제 들보다 상대적으로 많은 데이터와 큰 네트워크를 다루기 때문에 보다 큰 computing resource가 요구된다
- 따라서, 처음 코드를 구현시에는 전체 데이터를 다 load하고 큰 값의 epoch로 계산하기 보다는 전체 데이터의 일부( e.g., 전체의 1/10)와 적은 epoch ( e.g., 3~5epoch)를 가지고 코딩을 실행해보고 학습이 잘되는 것을 먼저 확인한다.
- 위의 과정으로 코드를 완성하고, 현재 본인의 구현환경(e.g, colab)을 고려해서 실행 가능한 최대 학습데이터와 epoch회수로 최종 결과를 제출하시기 바랍니다.
- 즉 과제를 수행함에 있어서 전체 데이터를 사용하지 않고 일부만 사용해도 되고, epoch 수도 적절한 결과를 얻는 선에서 본인이 선택하시면 됩니다.