Take-home midterm exam with Traffy Fondue dataset Report

Chapter 1 – Introduction

Objective:

- To apply all the knowledge and processes (e.g., EDA, modeling, etc.) learned in class to create an image classification task that can be performed to achieve a macro-averaged F1 score higher than 0.80 on test data in the private leaderboard and rank in the 11th to 30th percentile

Type of problem:

- Supervised learning with Multiclass image classification

Hypothesis:

- From the description of the dataset on Kaggle, the images (dataset) are taken and submitted to the application by problem reporters, and then the system will notify such problems to the responsible staff immediately. Since the images didn't pass any screening process, my hypothesis is that the data may be quite noisy and need to be cleaned thoroughly before being fed to the Neural Network model.

Framework

- Using Python language with Pytorch as the main library

Chapter 2 - Data preparation

1. Examining images to observe particular characteristics of images in each class.
2. Moving images that are in the wrong class (folder) to the correct class, getting rid of images with an unclear class (e.g., two problems illustrated in one image), and remove irrelevant images. Using a manual process to do this.
3. Augmented the train images (images in train folder)
    a. Resize the input images to (230, 230)
    b. Randomly rotate the input image by a degree between -30 and 30 degrees
    c. Random horizontal flip
    d. Convert images to tensor
    e. Normalize images with mean equals to [0.4303, 0.4301, 0.4139] and standard deviation equals to [0.2186, 0.2140, 0.2205]
4. Transform image in validation set and test set.
    a. Resize the input images to (224, 224)
    b. Convert images to tensor
    c. Normalize images with mean equals to [0.4303, 0.4301, 0.4139] and standard deviation equals to [0.2186, 0.2140, 0.2205]
5. Generate filename.csv that stores two columns, which are image filenames (.jpg) and their classes.
6. Use filename.csv to observe amount of data for each class (before split the data into train, validation, test).

Class	Number of images
Flooding	1341
Road	1331
electric	941
light	790
sidewalk	746
traffic	465
sanitary	450
canal	436
sewer	429
stray	346

7. Split the dataset into train set, validation set, and test set with ratio 80:10:10 respectively. Applied stratification and random state = 2022. Amount of data for each class are as follow.
    a. Train set

Class	Number of images
Flooding	1073
Road	1065
Electric	752
Light	632
Sidewalk	597
Traffic	372
Sanitary	360
Canal	349
Sewer	343
Stray	277

    b. Test set

Class	Number of images
Flooding	134
Road	133
Electric	94
Light	79
Sidewalk	74
Traffic	46
Sanitary	45
Canal	44
Sewer	43
Stray	35

c. Validation set

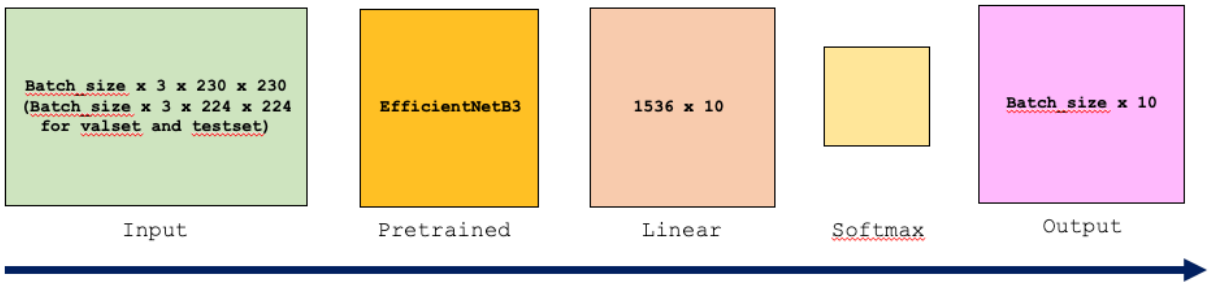
Class	Number of images
Flooding	134
Road	133
Electric	95
Light	79
Sidewalk	75
Traffic	47
Sanitary	45
Sewer	43
Canal	43
Stray	34

8. Create “TraffyFondueDataset” class that provide dataset for pytorch to feed to the model.
9. Prepare class weights to be applied while training the model to deal with imbalance class. This class weights are calculated by
   i. weights_for_samples $= \frac{1.0}{numpy.array(numpy.power(samples\_per\_cls, power))}$
   ii. norm_weights_for_samples $= \frac{\text{weights\_for\_samples} * no\_of\_classes}{\text{numpy.sum(weights\_for\_samples)}}$

   Finally, norm_weights_for_samples is an array of class weights
10. Prepare Dataloader for pytorch with batch size 32 which consists of 4 dataloaders
    a. Train dataloader
    b. Validate dataloader
    c. Test dataloader
    d. All dataloader (all data in train folder) – after monitoring and analyzing the model performance with train, validation, and test set, train model on All dataloader to submit result to Kaggle

Chapter 3 – Model

1. Using EfficientNet B3 model (ref. https://arxiv.org/pdf/1905.11946.pdf) with a linear transformation layer to get 10 output per sample and calculate Softmax to get the scores for each class. The model architecture is shown in the figure below.



2. Hyperparameters
    a. Monitoring and analyzing part (train, validation, test)
        i. Using CrossEntropyLoss as the loss function with class_weights applied
        ii. Using SDG optimizer with learning rate = 0.02 and using StepLR scheduler with step_size = 7 and gamma = 0.5
        iii. Training 20 epochs with batch size = 32
    b. Train with all data in train folder
        i. Using CrossEntropyLoss as the loss function with class_weights applied
        ii. Using SDG optimizer with learning rate = 0.02 and using StepLR scheduler with step_size = 7 and gamma = 0.5
        iii. Training 23 epochs with batch size = 32
3. Using wandb to monitor performance while training (https://api.wandb.ai/links/earth-akkharawat/15tnzdt2)

Chapter 4 – Result

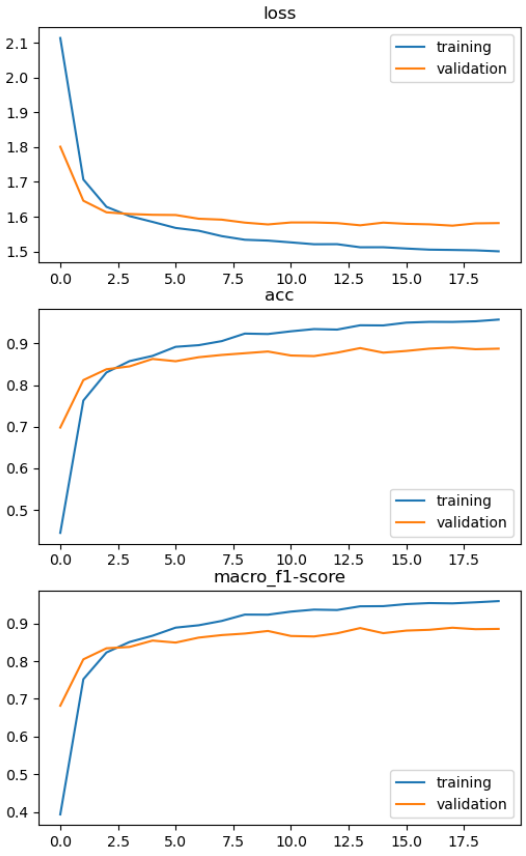
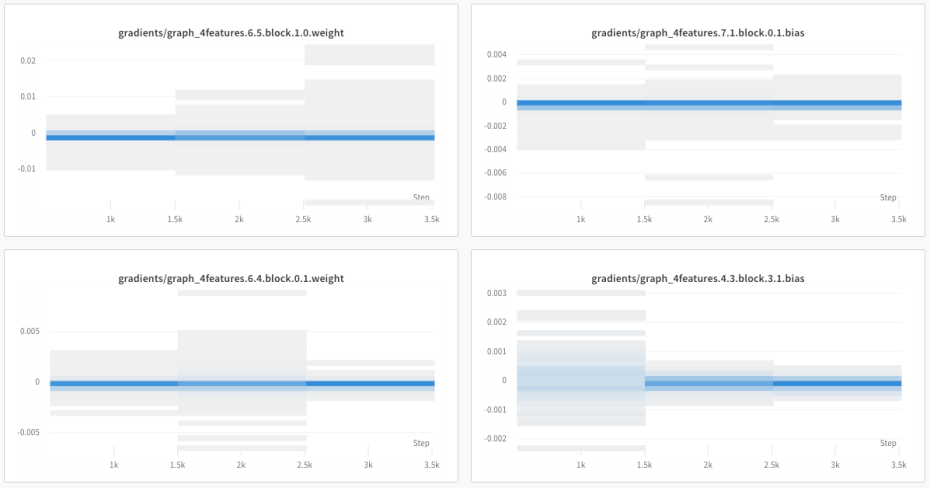


Figure 1.



Example of the gradients graph that shrinking and enlarging in some layers.

1. Train and validation set
    a. From wandb,
        i. Number of steps = 3639
        ii. Validation loss = 1.581
        iii. Validation macro-averaged F1 score = 0.8855
2. Test set
    a. Test loss = 1.577
    b. Test macro-averaged F1 score = 0.8827

testing loss: 1.577

	precision	recall	f1-score	support
0	0.7547	0.9091	0.8247	44
1	0.9149	0.9149	0.9149	94
2	0.9200	0.8582	0.8880	134
3	0.7931	0.8734	0.8313	79
4	0.9268	0.8571	0.8906	133
5	0.9048	0.8444	0.8736	45
6	0.8913	0.9535	0.9213	43
7	0.9041	0.8919	0.8980	74
8	0.9062	0.8286	0.8657	35
9	0.8654	0.9783	0.9184	46
accuracy			0.8845	727
macro avg	0.8781	0.8909	0.8827	727
weighted avg	0.8884	0.8845	0.8849	727

Figure 2.

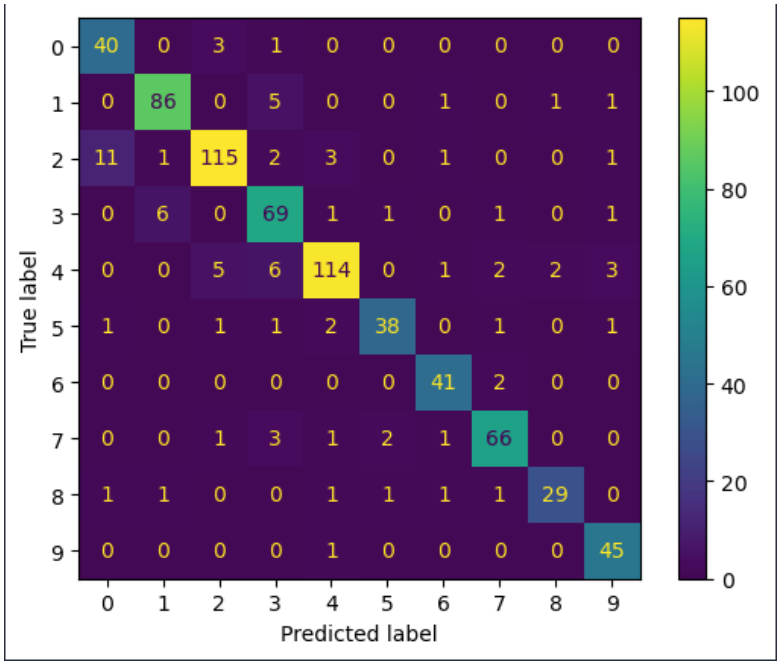


Figure 3.

3. Analyzing the results and then train model with all data in the train folder (I submitted result of this part to Kaggle)

Chapter 5 - Discussion, e.g., error analysis and how to improve it

Error 1

From the train and validation loss graph in figure 1 above, a little overfitting has occurred, so when I use all the data in the train folder to train the model, the overfitting behavior may occur like this. Thus, one of the reasons that my private score decreases from my public score

How to improve 1

1. Stop or decrease learning rate when overfit happens
2. Decrease number of epochs
3. Try using a model that is less complex than EfficientNet B3 (e.g., EfficientNet B0)

Error 2

From the classification report in figure 2, the lowest F1 score class is canal. From the confusion matrix in figure 3, the obvious error is the model predicted canal class while the true class is flooding 11 images.

How to improve 2

1. Removing image in canal and flooding class that tend to make the model confuse

Error 3

From the confusion matrix in figure 3, the model predicted light class while the true class is electric, and the model also predicted electric class while the true class is light.

How to improve 3

1. Removing image in light and electric class that tend to make the model confuse

Some technique that may improve the result.

1. Freeze the pretrained layers in the very first epochs and then unfreeze in the rest epochs
2. Using other schedulers such as get_linear_schedule_with_warmup, ReduceLROnPlateau
3. Using Cross-validation technique
4. Augment the images with ColorJitter

Chapter 6 – Conclusion

I conducted a multiclass image classification using EfficientNet B3 model to classify images into 10 different classes. The dataset was preprocessed by examining the images to ensure they were correctly labeled, augmented by resizing, rotating, and horizontal flipping the images, and then normalized. The dataset was split with stratification into a train set, validation set, and test set with an 80:10:10 ratio respectively. Class weights are applied to deal with the class imbalance issue. First, the model was trained for 20 epochs with a batch size of 32 and monitored the model's performance using wandb. The model achieved a validation macro-averaged F1 score of 0.8855 and a test macro-averaged F1 score of 0.8827. Then, the model was trained on all the data in the train folder for 23 epochs and submitted to Kaggle for evaluation which received a private macro-averaged F1 score of 0.7941. Discussion about errors in the model and ways to improve it have been stated.

6330585221 Akkharawat Burachokviwat