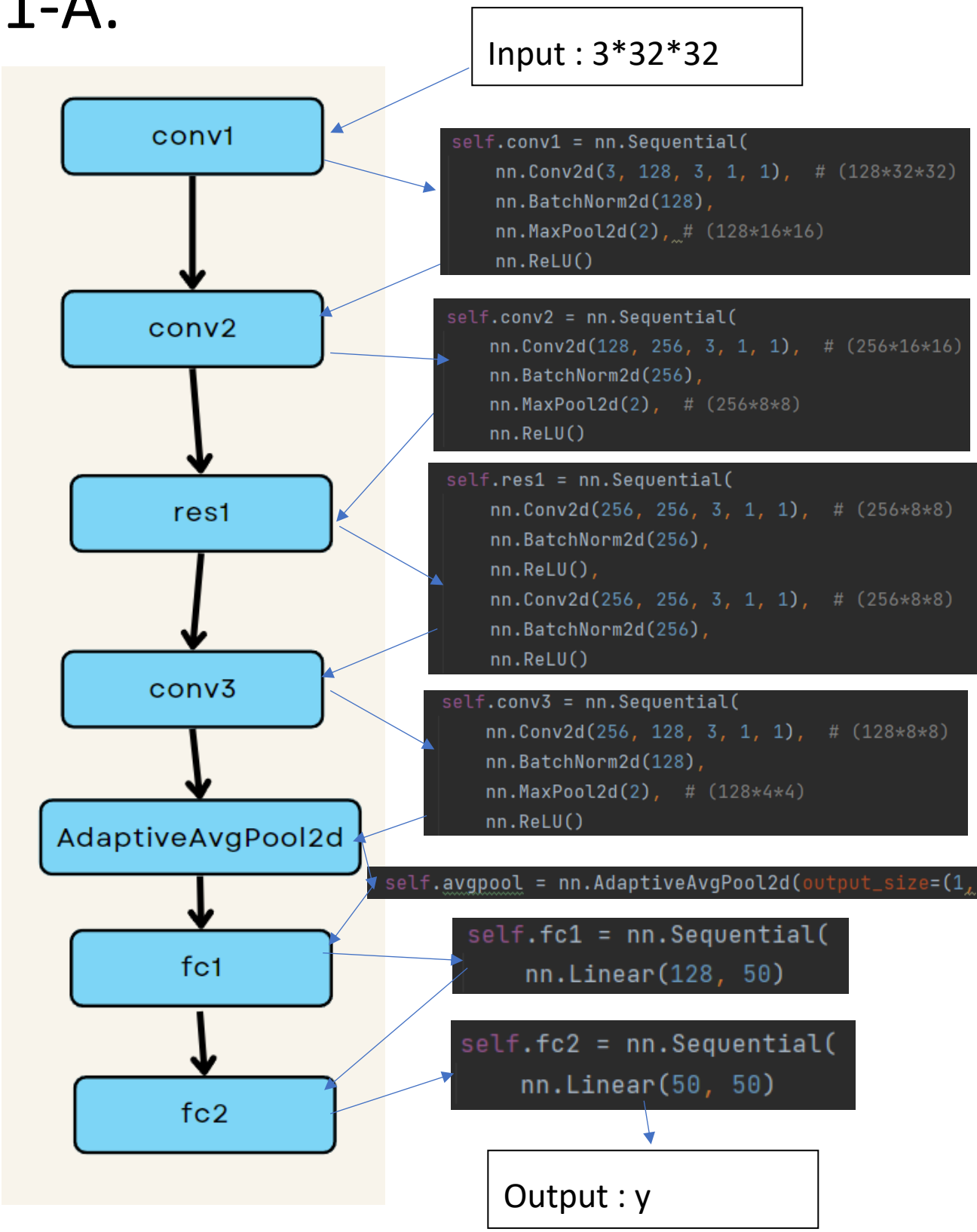
DLCV 2022 – HW1

Name : 周宇玄

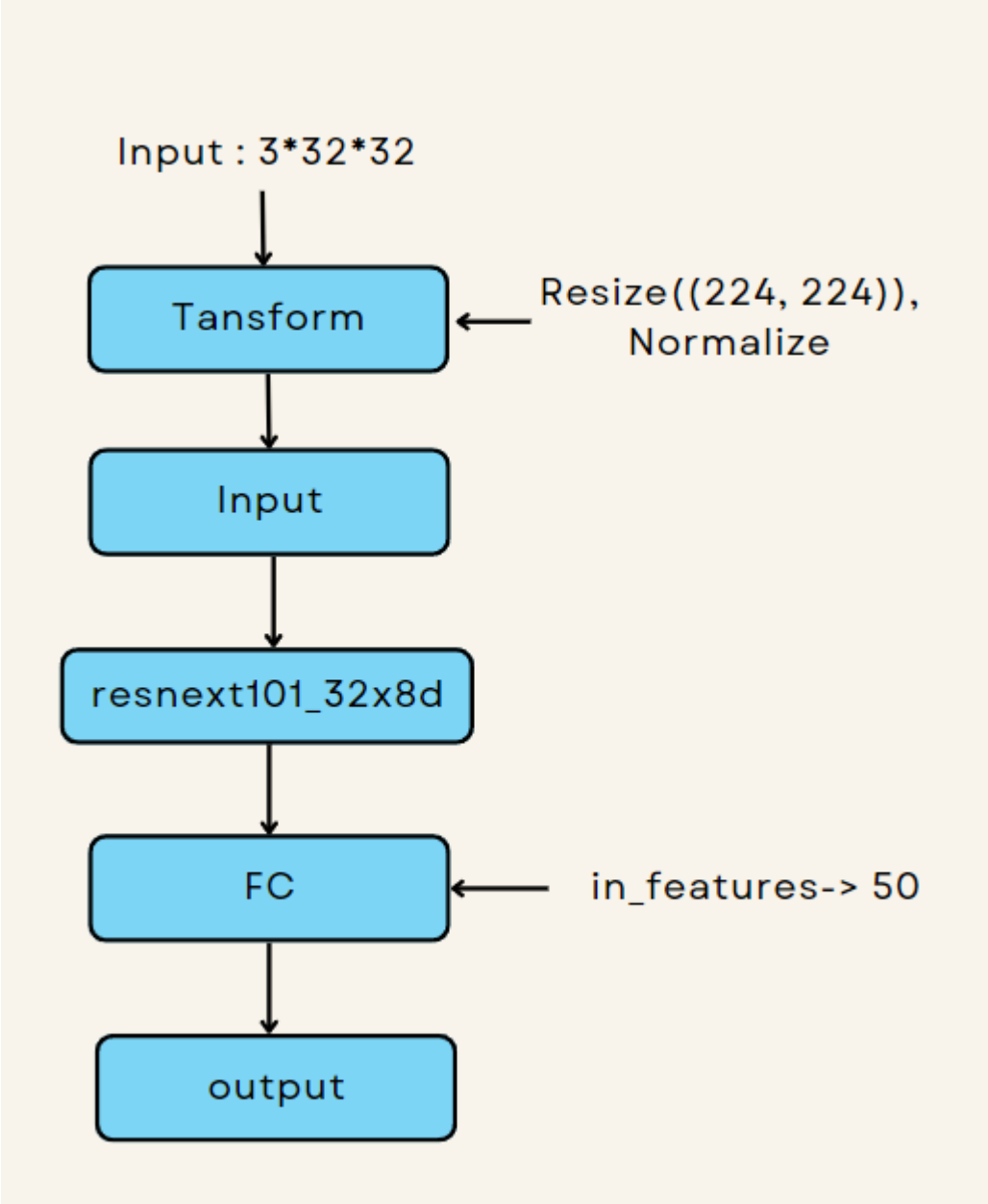
Student ID : R10525104

Problem 1.:

1-A.


Input : 3*32*32
conv1
self.conv1 = nn.Sequential(
nn.Conv2d(3, 128, 3, 1, 1), # (128*32*32)
nn.BatchNorm2d(128),
nn.MaxPool2d(2), # (128*16*16)
nn.ReLU()
conv2
self.conv2 = nn.Sequential(
nn.Conv2d(128, 256, 3, 1, 1), # (256*16*16)
nn.BatchNorm2d(256),
nn.MaxPool2d(2), # (256*8*8)
nn.ReLU()
res1
self.res1 = nn.Sequential(
nn.Conv2d(256, 256, 3, 1, 1), # (256*8*8)
nn.BatchNorm2d(256),
nn.ReLU(),
nn.Conv2d(256, 256, 3, 1, 1), # (256*8*8)
nn.BatchNorm2d(256),
nn.ReLU()
conv3
self.conv3 = nn.Sequential(
nn.Conv2d(256, 128, 3, 1, 1), # (128*8*8)
nn.BatchNorm2d(128),
nn.MaxPool2d(2), # (128*4*4)
nn.ReLU()
AdaptiveAvgPool2d
self.avgpool = nn.AdaptiveAvgPool2d(output_size=(1,1))
fc1
self.fc1 = nn.Sequential(
nn.Linear(128, 50)
fc2
self.fc2 = nn.Sequential(
nn.Linear(50, 50)
Output : y

1-B.


Input : 3*32*32
Tansform
Resize((224, 224)),
Normalize
Input
resnext101_32x8d
FC
in_features-> 50
output

2-A: 58%

2-B: 89%

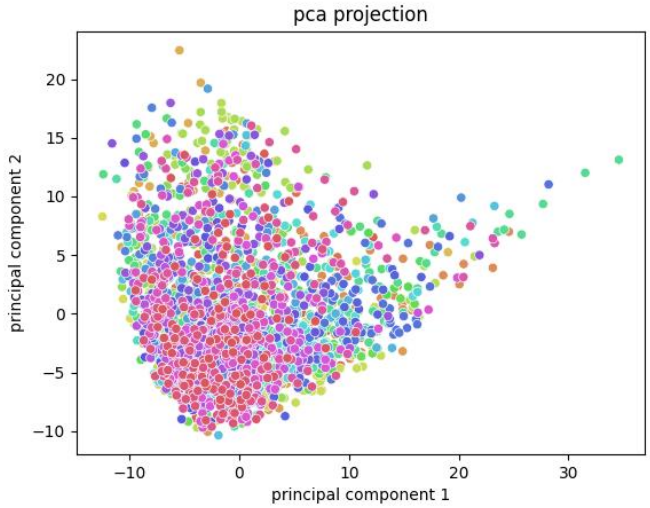
3: This model is using lr = 0.001, momentum = 0.9, optimizer = SGD, and loss function using CrossEntropy, the model architecture is as above, and cross validation by validation dataset, the key point that make this model get higher acc method is that the data augmentation, by add TandomHorizontalFilp, Affine, ColorJitter, and Normalize in training data set, this augmentation let acc become higher(around 50%->58%), and it's also the key point increase next model's acc..

4: This model is using pre-train model(Res101), lr = 0.0005, momentum = 0.9, weight_decay = 0.0003, optimizer = SAM, loss function is smoothcrossentropy, the smoothing set 0.1, we can see that the first different with model A is that this

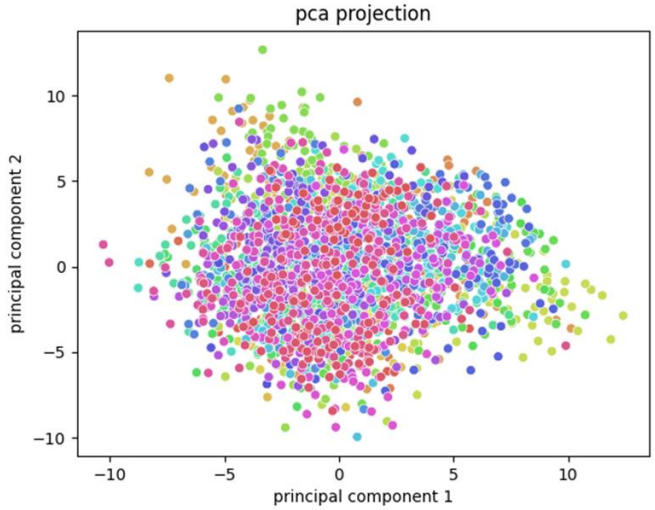
model was a pre-train model, then use different lr, optimizer, loss function, second different is that make total training data become double(45000), which is made by augmentation data + original data, by testing we know that double dataset is more good of training then single dataset with data augmentation, and the third different is augment we add resize(32 -> 224) make feature become more conspicuous, even this change will let training become heavy but it will significant improvement acc. of test on validation dataset.

5:

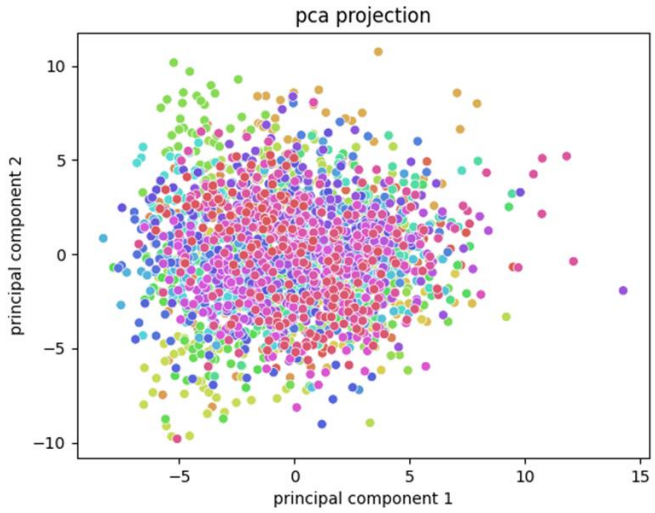
PCA Epoch 0 acc 15



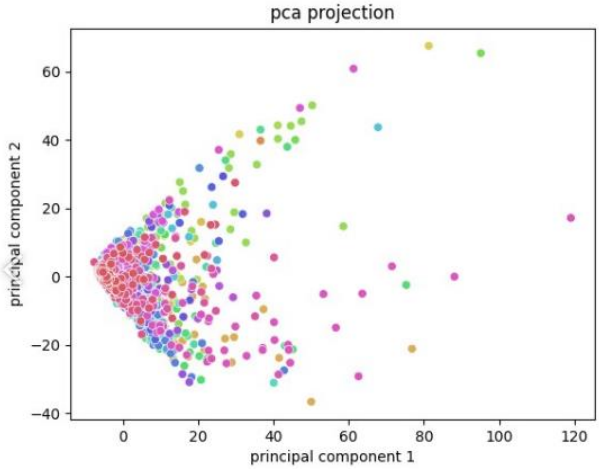
PCA Epoch 24 acc 53



PCA Epoch 49 acc 58

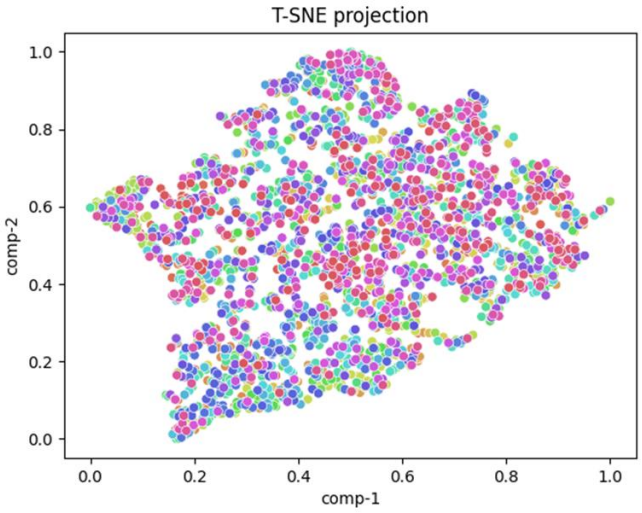


We can see that each point become closer and move to the left with higher acc., if acc. become high enough (as acc. 89%'s result on below) it will become crowded and cluster on the left, the reason is that PCA just simply use formula to location the position of each point and not consider each point's far and near.



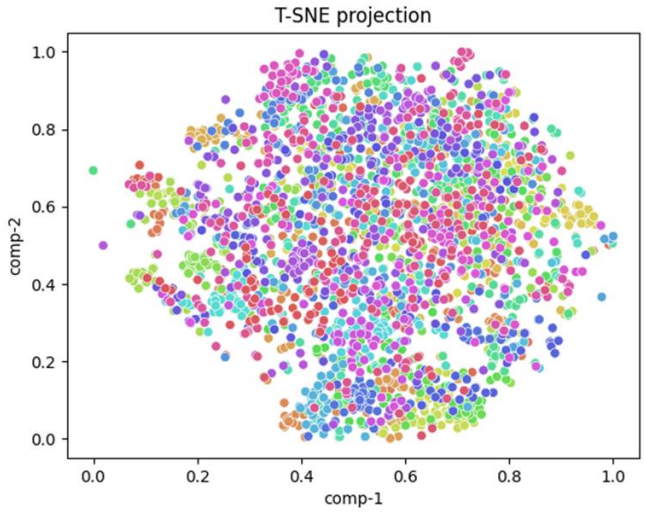
6:

t-SNE Epoch 0 acc 15

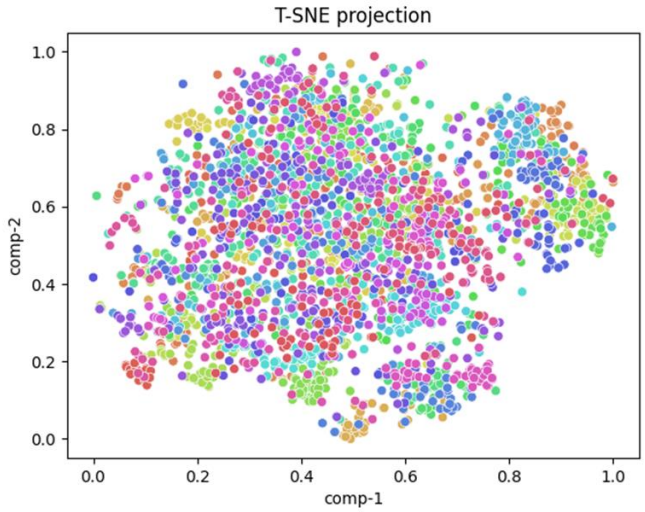


:

t-SNE Epoch 24 acc 53

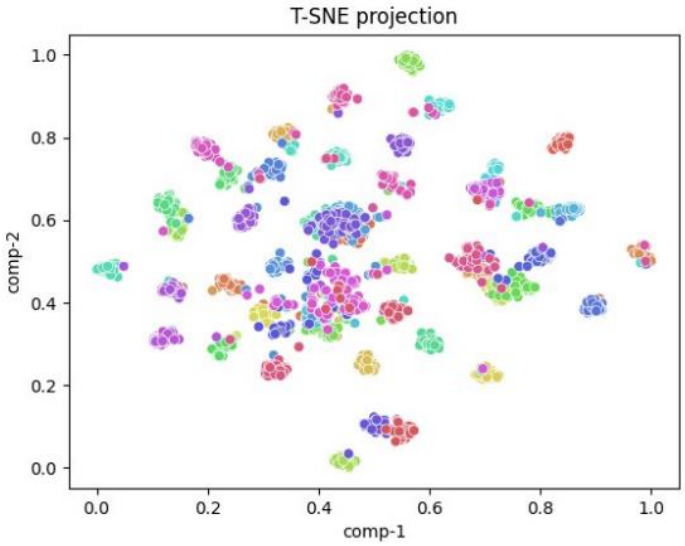


t-SNE Epoch 49 acc 58

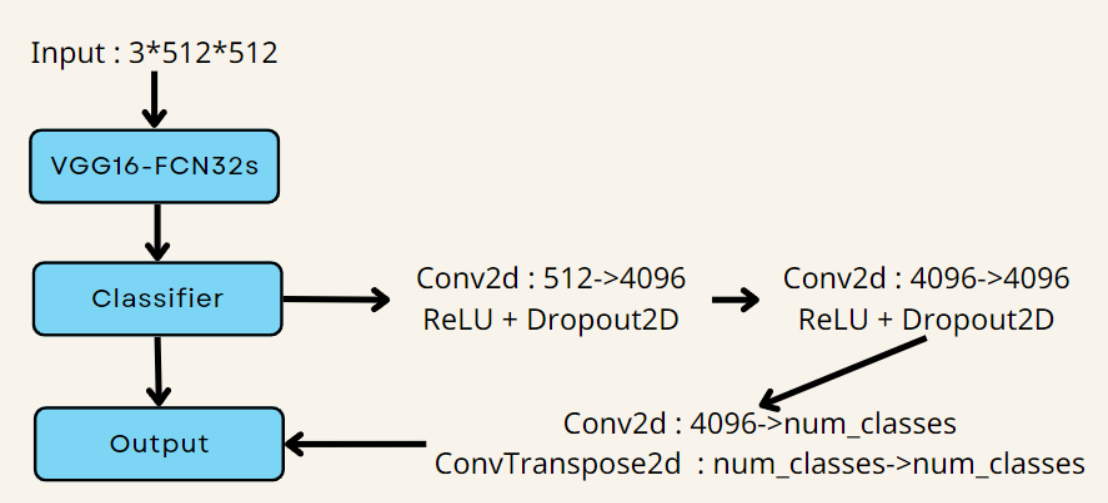


Just like PCA result, we can see that each point become closer with other point which have same colour, if acc become higher(like model-B 89% result shows on below), the t-SNE result will cluster same

colour more obvious too, but the difference with PCA result is that they(all point that have different colours) are not clustered together anymore, they will become further and further with other points which have different colours, this is because t-SNE will consider distance of each colours.

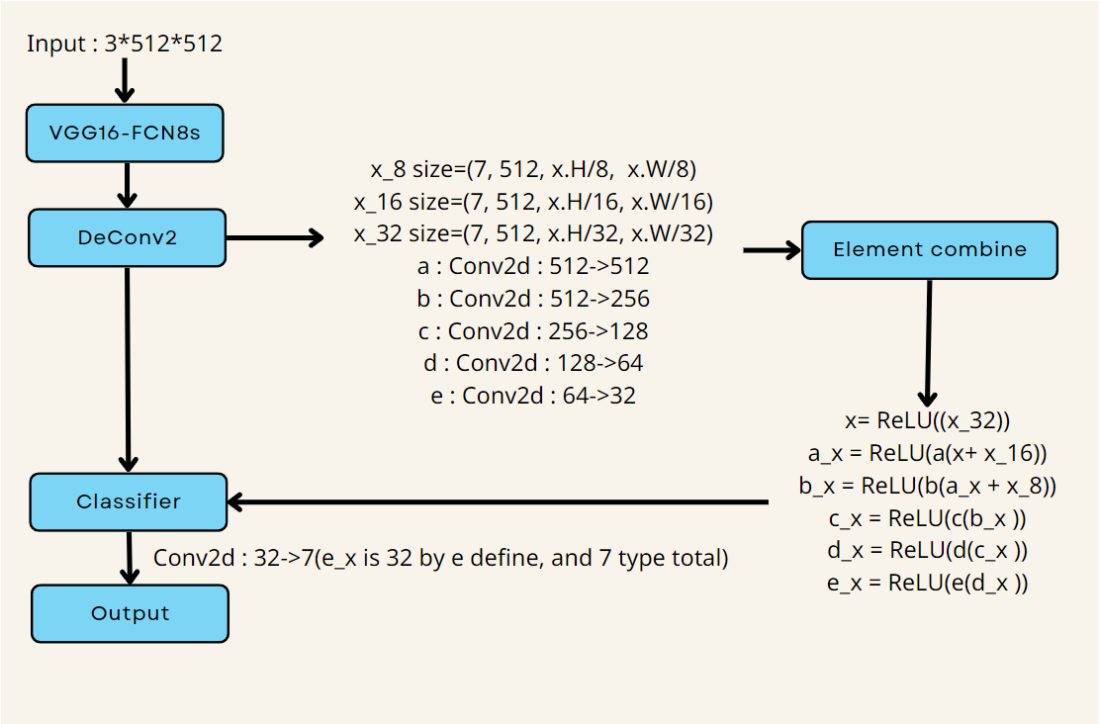


Problem 2.:

1. 

This model is using lr = 0.0005, momentum = 0.9, optimizer = Adam, and loss function using NLLLoss2d, the model architecture is as above, the point of this part(hw1_2) is that I use the data augmentation by flip image up-side-down, left-side-right, and do both of image, this augmentation make training data number become 4 times and make acc. increase about 2%.

2.



Using same lr, momentum, optimizer, data augmentation. The first differ is that use FCN8s to instead of FCN32s, which make acc. increase about 7%, and second differ is that make Conv2D has dimensionality reduction, it will make acc become higher(about 2~3%), it's because by dimensionality reduction feature become more obvious, third differ is change loss function to CrossEntropyLoss, it make acc.

increase 1% to achieve strong baseline.

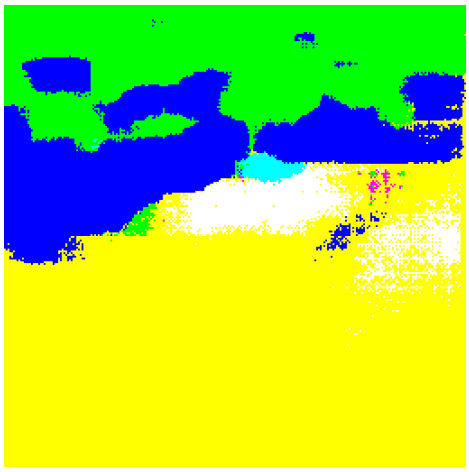
3-A: 62%

3-B: 73%

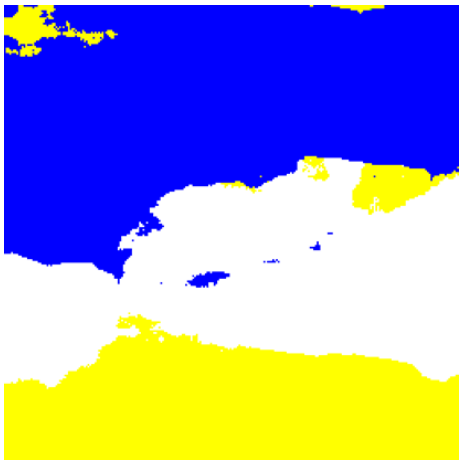
4.

predicted segmentation mask of “validation/0013_sat.jpg”

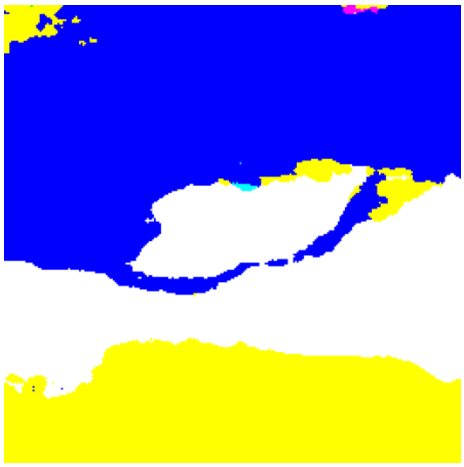
ep1:



ep100:

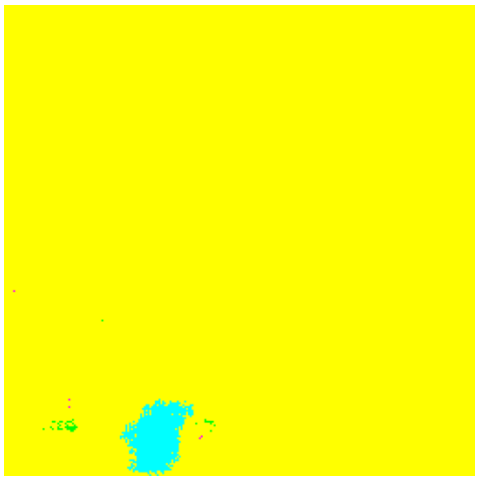


ep200:



predicted segmentation mask of “validation/0062_sat.jpg”

ep1:

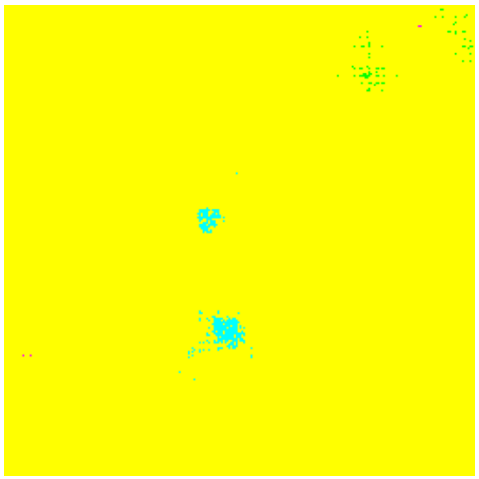


ep100:

ep200:

predicted segmentation mask of “validation/0104_sat.jpg”

ep1:



ep100:

ep200: