**The following is an introduction to the modified parts of the code in my project**

**Hyperparameter section:**

**First, set the number of categories to 5, because there are five artistic styles in total. Secondly, in order to verify and test the performance of the model, the data set is divided into a training set, a verification set,** and a test set.num\_classes = 5

val\_size = 0.2  
test\_size = 0.2

batch\_size = 150

learn\_rate = 0.001(Different values were tested)

**Preprocessing and data enhancement part:**

**Since the data enhancement in the original case was already relatively comprehensive, not many other operations were added.**

**Currently the following two have been added**

**# Flip the image vertically with 50% probability to increase the model’s robustness to up and down flips**transforms.RandomVerticalFlip(p=0.5),  
**# Perform a random rotation between -90 and 90 degrees**transforms.RandomRotation(degrees=(-90,90)),

**Dataset loading part:**

**In order to facilitate testing the model effect, the code for dividing the test set is added.**

**The following is the corresponding code**

**# First split for train and temporary split. Test size here is combined validation and test size**train\_indices, temp\_indices = train\_test\_split(indices, test\_size=val\_size + test\_size, random\_state=42)  
  
**# Second split for validation and test from the temporary split**val\_indices, test\_indices = train\_test\_split(temp\_indices, test\_size=test\_size/(val\_size + test\_size), random\_state=42)

**Visualize parts of the dataset:**

**Added code for visual test set**

**# Plot some test images**real\_batch = next(iter(test\_loader))  
plt.figure(figsize=(8,8))  
plt.axis("off")  
plt.title("Test Images")  
plt.imshow(np.transpose(vutils.make\_grid(real\_batch[0].to(device)[:64], padding=2, normalize=True).cpu(),(1,2,0)))

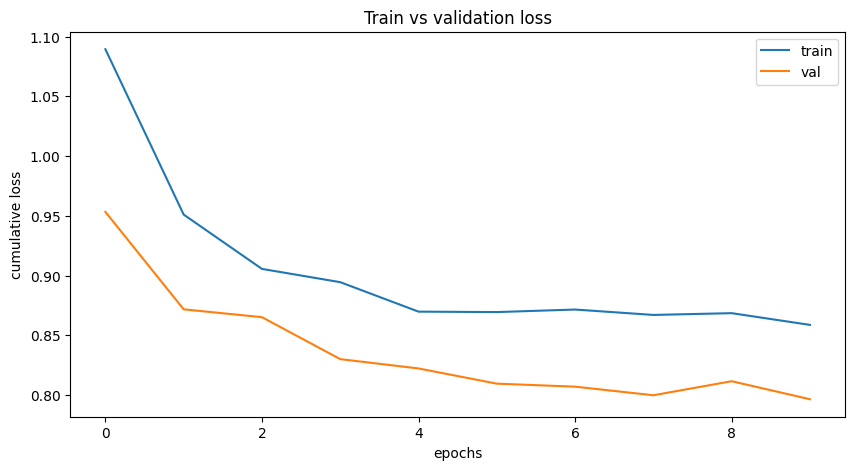
**Model part:**

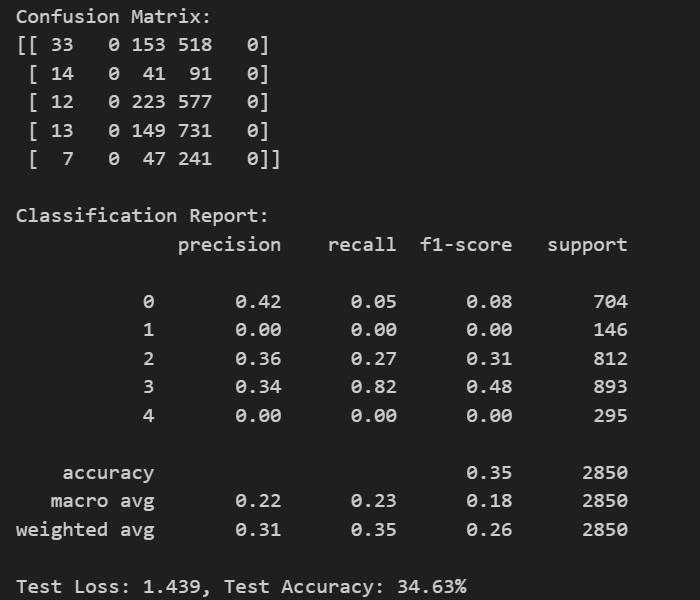
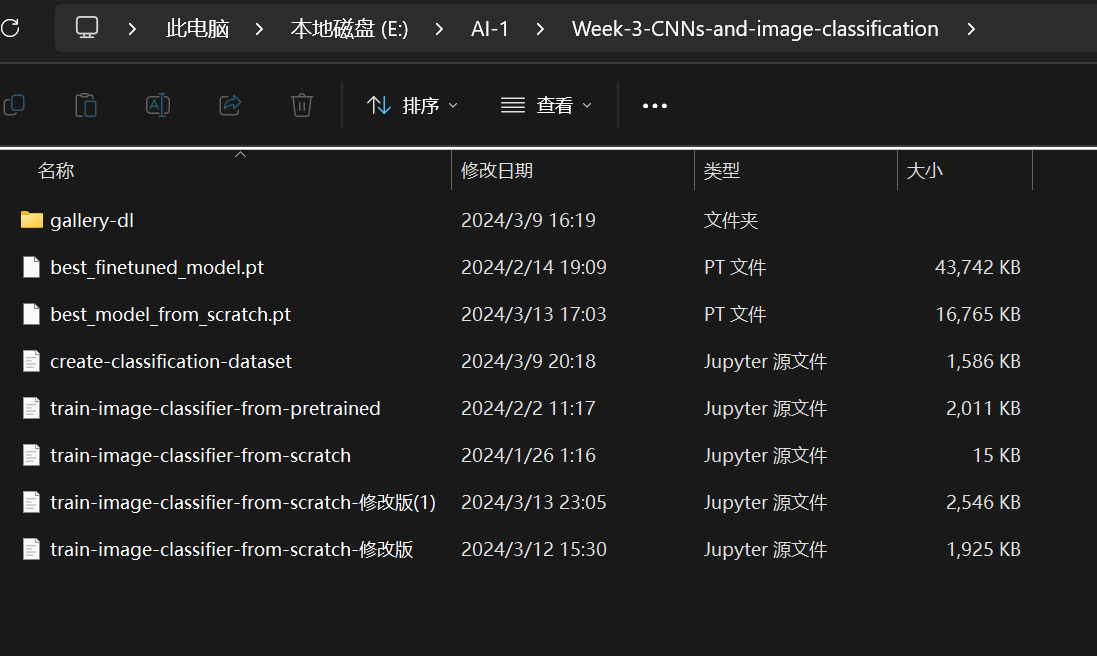
**I reduced the number of filters in the first two convolutional layers, which reduces the model size and computational requirements. At the same time, by adding padding in the convolution operation, it can be ensured that the size of the feature map is maintained after the convolution operation. A Dropout layer is added to prevent model overfitting. The number of neurons in the first fully connected layer is returned to 512 to reduce model complexity and increase training speed. If the above effect is not very good, I also tested the more complex model written below, but found that the speed would be slower.**  
class ClassificationNetwork(nn.Module):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
  
 self.conv1 = nn.Conv2d(3, 32, 3, padding=1)  
 self.conv2 = nn.Conv2d(32, 64, 3, padding=1)  
 self.conv3 = nn.Conv2d(64, 128, 3, padding=1)  
  
 self.pool = nn.MaxPool2d(2, 2)  
  
 self.dropout = nn.Dropout(0.25)  
  
 self.fc1 = nn.Linear(128 \* 8 \* 8, 512)  
 self.fc2 = nn.Linear(512, num\_classes)  
  
 def forward(self, x):  
 x = self.pool(F.relu(self.conv1(x)))  
 x = self.pool(F.relu(self.conv2(x)))  
 x = self.pool(F.relu(self.conv3(x)))  
  
 x = x.view(-1, 128 \* 8 \* 8)  
  
 x = self.dropout(x)  
  
 x = F.relu(self.fc1(x))  
 x = self.dropout(x)  
  
 x = self.fc2(x)  
 return x

**Test part:**

**In addition, after training, the test code is also added. The main indicators include Confusion Matrix, Classification Report, and Accuracy.**

**About data selectionThe source of the data set is crawled from the Pinterest website, and the data type is art pictures of different categories. I grabbed a total of 5 pictures of artistic styles and put them into classes 1-5 respectively. At the beginning, I found in the test that the selection styles of the data sets were relatively similar, but the current model could not classify types that were too similar, and the results were not ideal, so I increased the amount of data and chose a more distinctive artistic style for classification. The data Three of the groups were concentrated at 3000-4000 pictures, and the other two groups were 716 and 1476. I entered more data into the groups whose styles were more difficult to distinguish. In addition to data preprocessing, I also performed manual filtering.**

**Test table comparison 图表, 折线图

描述已自动生成**