1. CNN

```python
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(65, 128, kernel_size=3, stride=1, padding=1)
        self.conv2 = nn.Conv2d(128, 256, kernel_size=3, stride=1, padding=1)
        self.pool = nn.MaxPool2d(kernel_size=(2,1), stride=2, padding=0)
        self.fc1 = nn.Linear(1024 * 4, 3)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = self.pool(x)
        x = F.relu(self.conv2(x))
        x = self.pool(x)

        x = x.view(-1, 512 * 8)
        x = self.fc1(x)

        return x
```

두개의 합성곱층만 사용했음 (효과 가장 좋음)

트레인 F1 score 지표는 아래와 같음

```
Classification Report:
              precision    recall  f1-score   support

           0       0.29      0.80      0.42       125
           1       0.00      0.00      0.00       251
           2       0.27      0.35      0.31        99

    accuracy                           0.28       475
   macro avg       0.19      0.38      0.24       475
weighted avg       0.13      0.28      0.18       475


F1 Score: 0.28421052631578947
```

2. VIT

Vit는 좀 시행착오가 많았습니다 -> train살짝 수정해서 사용했음

```python
class VisionTransformer(nn.Module):
    def __init__(self, img_size:int, patch_size:int, in_channels:int, embed_dim:int, num_heads:int, num_layers:int, hidden_dim:int, num_classes:int):
        super().__init__()
        self.embedding = EmbeddingLayer(img_size, patch_size, in_channels, embed_dim)
        self.blocks = nn.ModuleList([Block(embed_dim, num_heads, hidden_dim) for _ in range(num_layers)])
        self.fc_out = nn.Linear(embed_dim, num_classes)

    def forward(self, x):
        x = self.embedding(x)
        for block in self.blocks:
            x = block(x)
        x = x.mean(dim=1)
        return self.fc_out(x)
```

임베드 레이어로 인스턴스 나누고 벡터로 변환하고 트랜스 포머 block에서 입력데이터 처리후 출력레이어에서 매핑하고 클라스 예측

```
Classification Report:
              precision    recall  f1-score   support

           0       0.26      1.00      0.42       125
           1       0.00      0.00      0.00       251
           2       0.00      0.00      0.00        99

    accuracy                           0.26       475
   macro avg       0.09      0.33      0.14       475
weighted avg       0.07      0.26      0.11       475


F1 Score: 0.2631578947368421
```

학습시간은 두배로 걸린 것 같은데 효과는 더 안좋게 등장