學號：b06901153 系級：電機三 姓名：林瑩昇
Collaborator：b07901052 劉展碩

1. 請說明你實作的 CNN 模型(best model)，其模型架構、訓練參數量和準確率為何？(1%)

```
"""# Model"""
class Classifier(nn.Module):
    def __init__(self):
        super(Classifier, self).__init__()
        #input dimension [3, 128, 128]
        self.cnn = nn.Sequential(
            nn.Conv2d(3, 64, 3, 1, 1), # [64, 128, 128]
            nn.BatchNorm2d(64),
            nn.RReLU(0.001,0.005),
            nn.Conv2d(64, 64, 3, 1, 1),
            nn.BatchNorm2d(64),
            nn.RReLU(0.001,0.005),
            nn.MaxPool2d(2, 2, 0), # [64, 64, 64]
            nn.Dropout(0.3),

            nn.Conv2d(64, 128, 3, 1, 1), # [128, 64, 64]
            nn.BatchNorm2d(128),
            nn.RReLU(0.001,0.005),
            nn.MaxPool2d(2, 2, 0), # [128, 32, 32]
            nn.Dropout(0.3),

            nn.Conv2d(128, 256, 3, 1, 1), # [256, 32, 32]
            nn.BatchNorm2d(256),
            nn.RReLU(0.001,0.005),
            nn.MaxPool2d(2, 2, 0), # [256, 16, 16]
            nn.Dropout(0.3),

            nn.Conv2d(256, 512, 3, 1, 1), # [512, 16, 16]
            nn.BatchNorm2d(512),
            nn.RReLU(0.001,0.005),
            nn.MaxPool2d(2, 2, 0), # [512, 8, 8]
            nn.Dropout(0.3),

            nn.Conv2d(512, 512, 3, 1, 1), # [512, 8, 8]
            nn.BatchNorm2d(512),
            nn.RReLU(0.001,0.005),
            nn.MaxPool2d(2, 2, 0), # [512, 4, 4]
            nn.Dropout(0.3),
        )
        self.fc = nn.Sequential(
            nn.Linear(512*4*4, 1024),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(1024, 512),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(512, 256),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(256, 11)
        )
    def forward(self, x):
        out = self.cnn(x)
        print(out.size())
        out = out.view(out.size()[0], -1)
        return self.fc(out)
```

model參數量：parameter total:12999371, trainable:12999371
Validation set acc：0.759 (150 epochs)

2. 請實作與第一題接近的參數量，但 CNN 深度（CNN 層數）減半的模型，並說明其模型架構、訓練參數量和準確率為何？(1%)

```python
"""# Model"""
class Classifier(nn.Module):
    def __init__(self):
        super(Classifier, self).__init__()
        #torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride, padding)
        #torch.nn.MaxPool2d(kernel_size, stride, padding)
        #input dimension [3, 128, 128]
        self.cnn = nn.Sequential(
            nn.Conv2d(3, 64, 3, 1, 1), # [64, 128, 128]
            nn.BatchNorm2d(64),
            nn.RReLU(0.001,0.005),
            nn.Conv2d(64, 64 , 3, 1, 1),
            nn.BatchNorm2d(64),
            nn.RReLU(0.001,0.005),
            nn.MaxPool2d(2, 2, 0), # [64, 64, 64]
            nn.Dropout(0.3),

            nn.Conv2d(64, 128, 3, 1, 1), # [128, 64, 64]
            nn.BatchNorm2d(128),
            nn.RReLU(0.001,0.005),
            nn.MaxPool2d(2, 2, 0), # [128, 32, 32]
            nn.Dropout(0.3),

            nn.Conv2d(128, 256, 3, 1, 1), # [256, 32, 32]
            nn.BatchNorm2d(256),
            nn.RReLU(0.001,0.005),
            nn.MaxPool2d(4, 4, 0), # [256, 8, 8]
            nn.Dropout(0.3),
        )
        self.fc = nn.Sequential(
            nn.Linear(256*8*8, 1024),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(1024, 512),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(512, 256),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(256, 11)
        )
    def forward(self, x):
        out = self.cnn(x)
        out = out.view(out.size()[0], -1)
        return self.fc(out)
```

原model參數量：parameter total:12999371, trainable:12999371
新model參數量：parameter total:17845963, trainable:17845963
Validation set acc：0.687 (150 epochs)

3. 請實作與第一題接近的參數量，簡單的 DNN 模型，同時也說明其模型架構、訓練參數和準確率為何？(1%)

我一開始先建了一個簡單的model，想說之後再慢慢逼近參數量，結果發現這個DNN model 不僅參數量是CNN model的4倍，train了50個epoch後，validation acc大概會在0.27左右震盪，結果非常差。

```
"""# Model"""
class Classifier(nn.Module):
    def __init__(self):
        super(Classifier, self).__init__()
        #input dimension [3, 128, 128]
        self.fc = nn.Sequential(
            nn.Linear(3*128*128, 1024),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(1024, 512),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(512, 256),
            nn.ReLU(),
            nn.Dropout(0.5),
            nn.Linear(256, 11)
        )

    def forward(self, x):
        x = x.view(x.size()[0], -1)
        return self.fc(x)
```

4. 請說明由 1 ~ 3 題的實驗中你觀察到了什麼？(1%)

由前三題的實作中可以發現，CNN在影像辨識方面會有比較好的表現，一來它網絡裡input的方式特別適合圖片，二來因為用convolution layer的關係，相同的參數量下，可以做得比一般DNN更深，正確率也就比較高

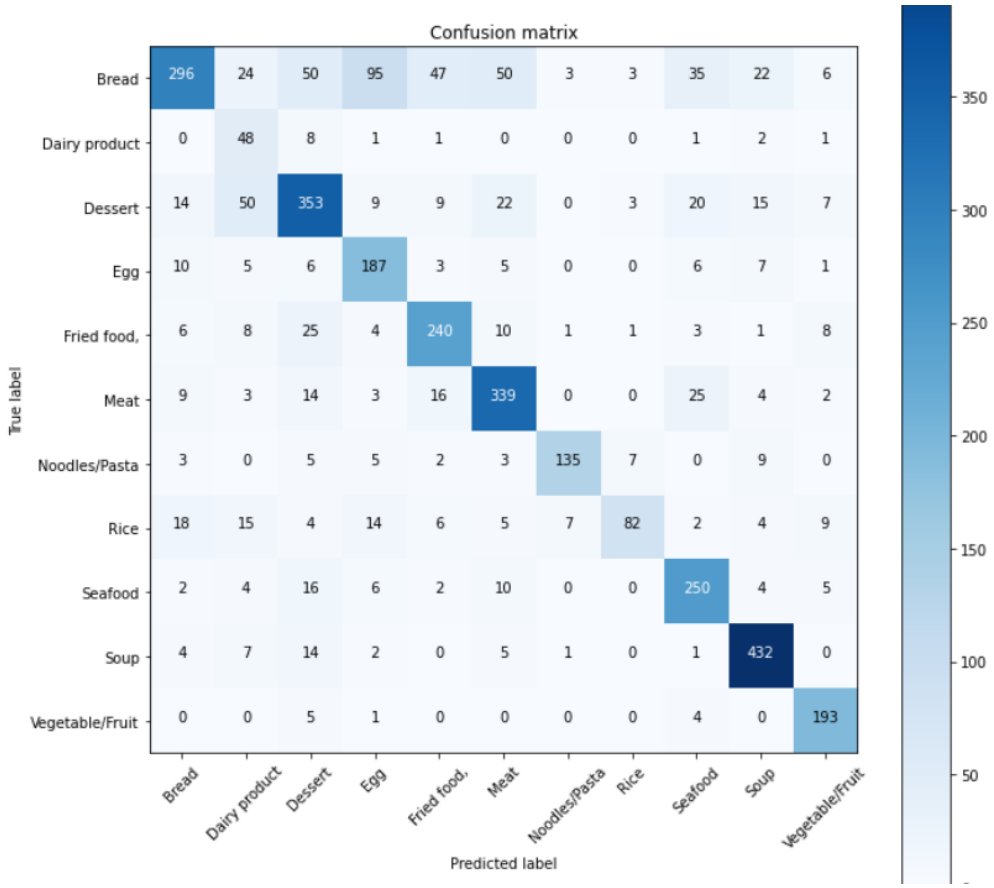
5. 請嘗試 data normalization 及 data augmentation，說明實作方法並且說明實行前後對準確率有什麼樣的影響？(1%)

   我augmentation有用到Flip、Rotate及ColorJitter，我把這些都拿掉後validation acc從原先0.759下降到0.711(150 epochs)

   我也有用過data normalization，但結果會稍差，我將data normalize至[-1, 1]後，validation acc = 0.727

6. 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析](1%)

   Confusion matrix



由上圖可以發現麵包很容易被誤判為其他的食物，尤其容易跟蛋搞混。
我認為有兩個原因：(1) 麵包與蛋的顏色很相近 (2) 麵包與蛋其實可能同時出現