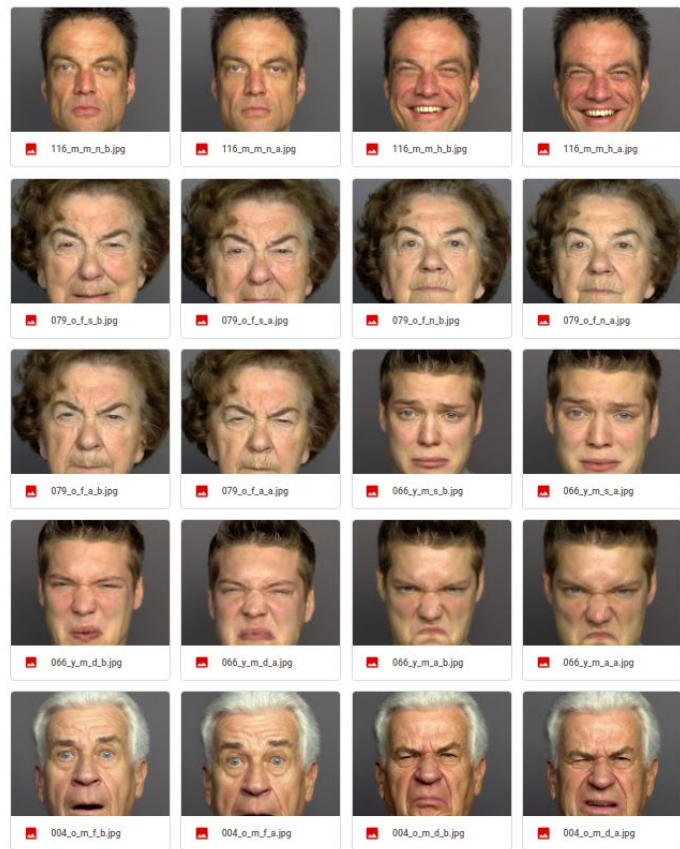
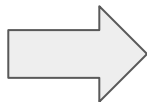
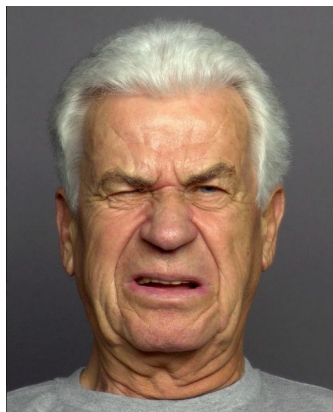


# Data preprocessing

- Cut the faces only
- Grayscale
- Jitter
- Resize



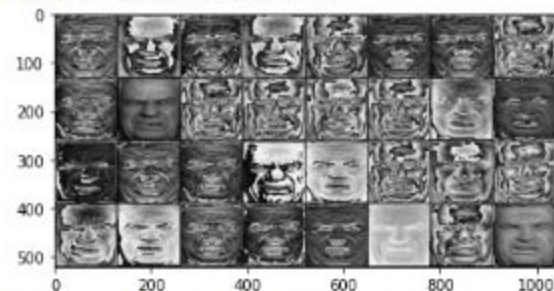
# Emotions

- Limited data set
  - 12 photos per each emotion

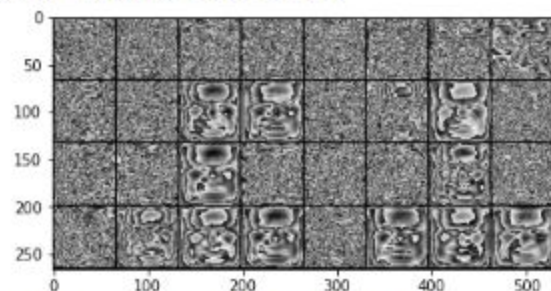
```
class CNN(nn.Module):
    def __init__(self, num_classes=2):
        super(CNN, self).__init__()
        self.layer1 = nn.Sequential(
            nn.Conv2d(3, 32, kernel_size=11, stride=2, padding=5),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.layer2 = nn.Sequential(
            nn.Conv2d(32, 32, kernel_size=7, stride=1, padding=3),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.layer3 = nn.Sequential(
            nn.Conv2d(32, 32, kernel_size=5, stride=1, padding=2),
            nn.Dropout(0.4),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.layer4 = nn.Sequential(
            nn.Conv2d(32, 32, kernel_size=5, stride=1, padding=2),
            nn.Dropout(0.4),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.layer5 = nn.Sequential(
            nn.Conv2d(32, 32, kernel_size=3, stride=1, padding=1),
            nn.Dropout(0.4),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.fc = nn.Sequential(
            nn.Linear(512, 64),
            nn.ReLU(),
            nn.Linear(64, 16),
            nn.ReLU(),
            nn.Linear(16, num_classes))

    def forward(self, x):
        out = self.layer1(x)
        out = self.layer2(out)
        out = self.layer3(out)
        out = self.layer4(out)
        out = self.layer5(out)
        out = out.reshape(out.size(0), -1)
        out = self.fc(out)
        return out
```

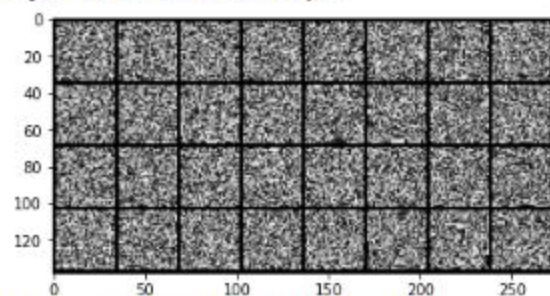
layer 1: Convolution output



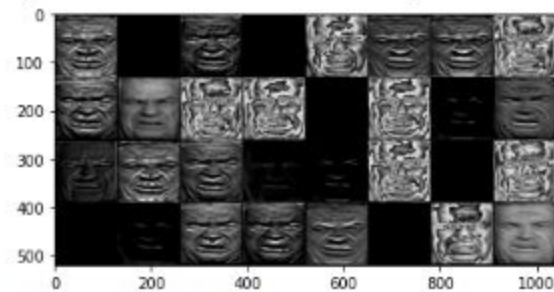
layer 2: Convolution output



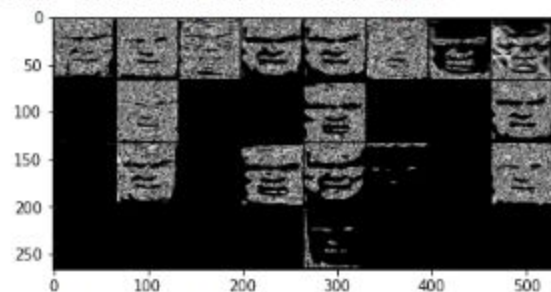
layer 3: Convolution output



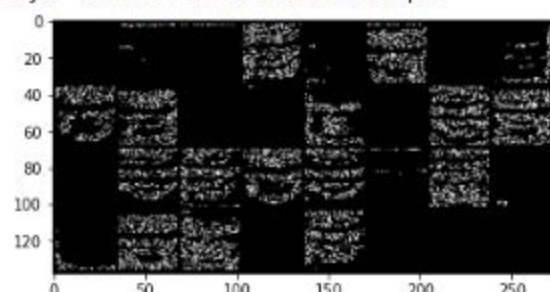
layer 1: Activation function output



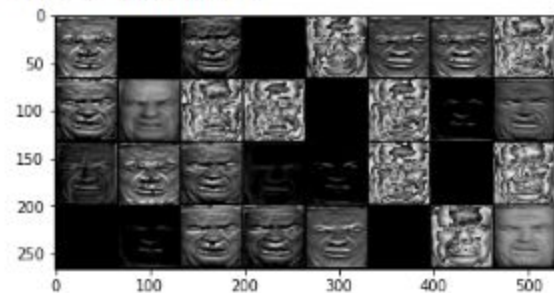
layer 2: Activation function output



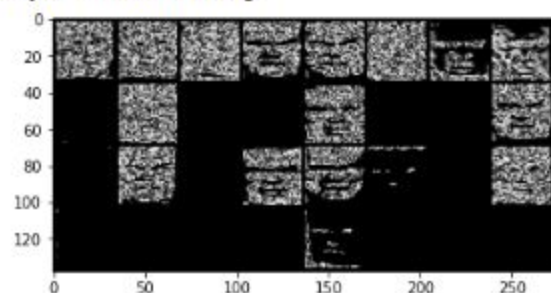
layer 3: Activation function output



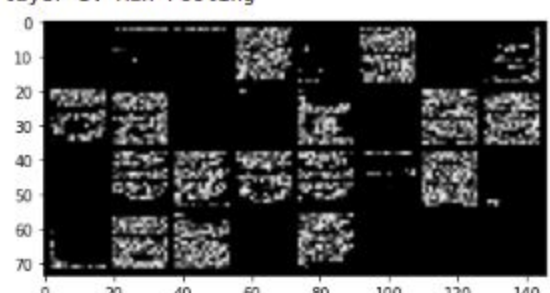
layer 1: Max-Pooling



layer 2: Max-Pooling

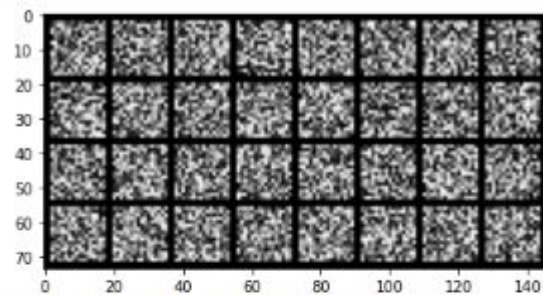


layer 3: Max-Pooling

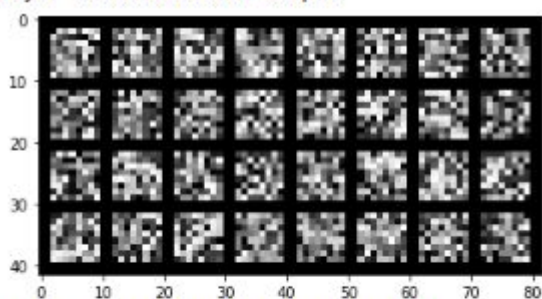




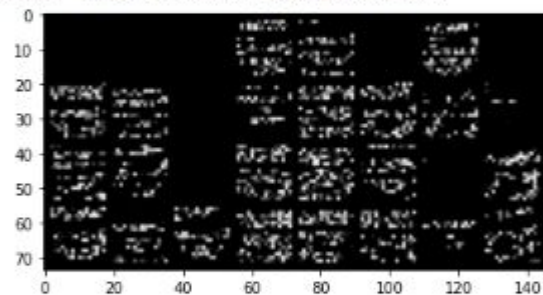
layer 4: Convolution output



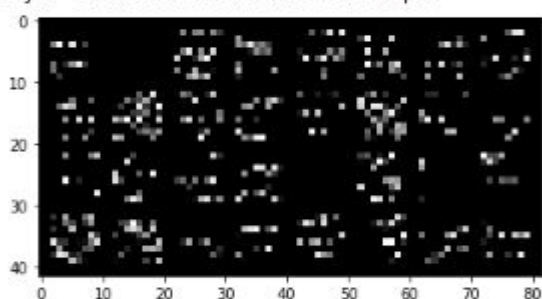
layer 5: Convolution output



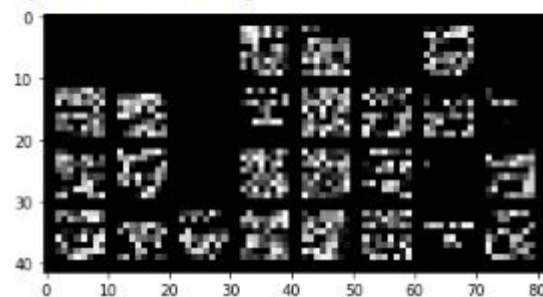
layer 4: Activation function output



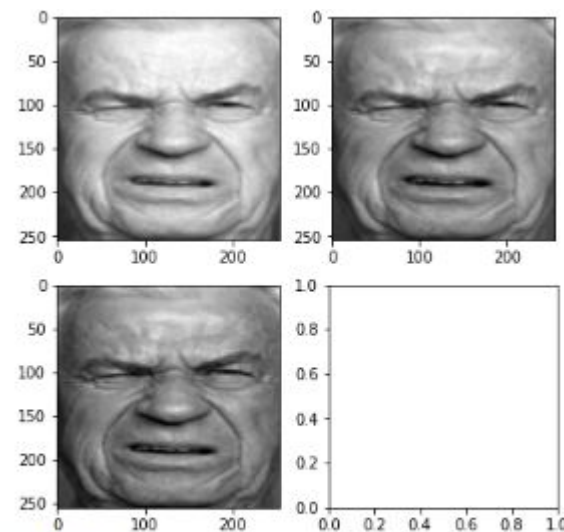
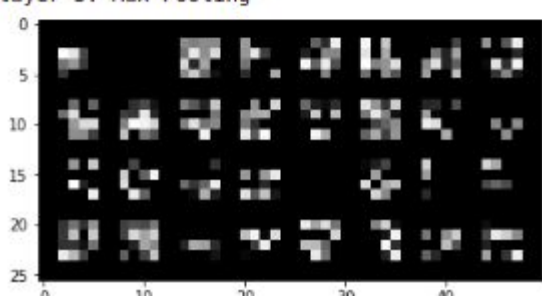
layer 5: Activation function output



layer 4: Max-Pooling

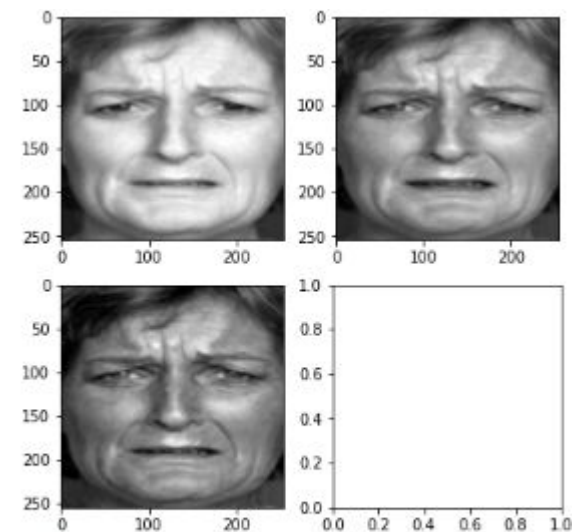


layer 5: Max-Pooling



Label: 1  
Label: disgust  
Prediction: angry .

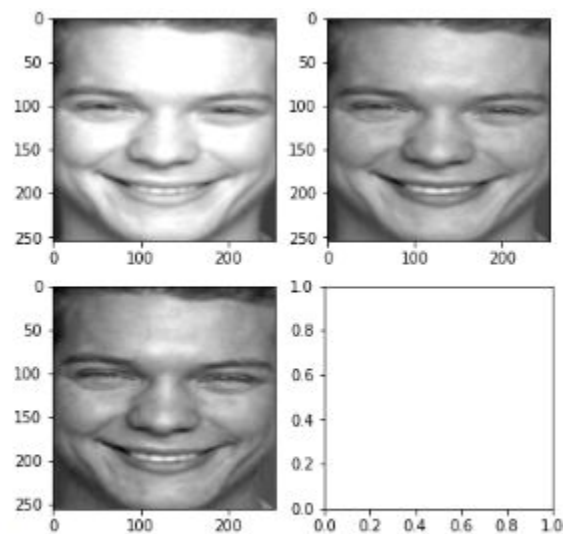
# Results



Label: 5  
Label: sad  
Prediction: sad .



Label: 1  
Label: disgust  
Prediction: disgust .

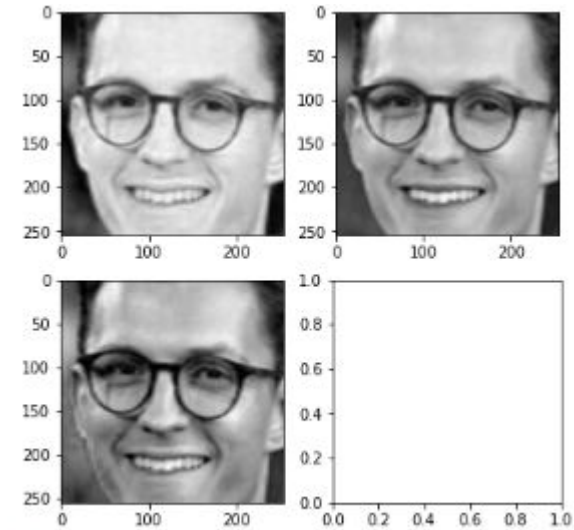


Label: 3  
Label: happy  
Prediction: sad .

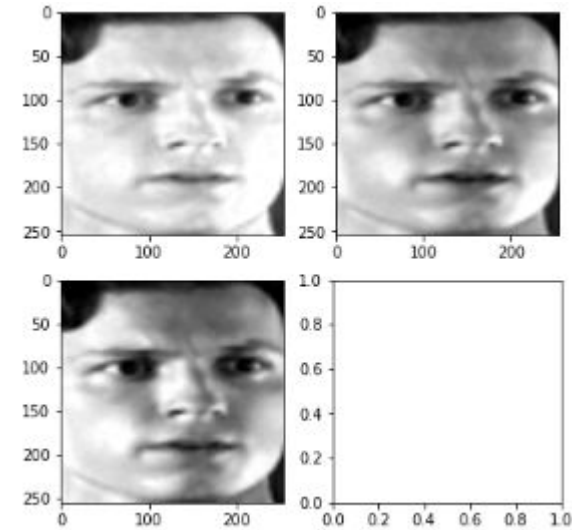
```
1 test(model)
```

Test Accuracy: 77.272727272727 %

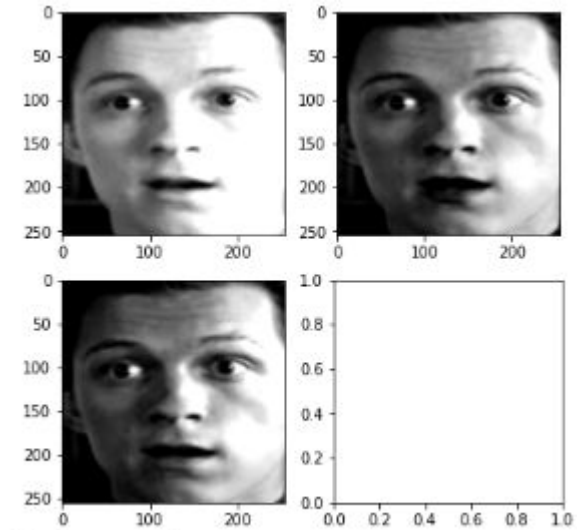
# Results with different data set test



Emotion: angry .



Emotion: disgust .



Emotion: sad .

- Not great result because the limit data set