

ML_Hw6

一、 想法

這次作業我是使用 CNN 的 Encoder、Decoder 來完成，我有嘗試用 Resnet、VAE 來實作，不過效果都挺差的，VAE 的部分，本身對於處理去雜訊的圖片上有較好的準確度，對於這次作業的任務就沒有那麼好的表現；Resnet 的部份讓我蠻匪夷所思的，Resnet 對於處理圖片上相較於傳統上簡易的 CNN 明顯優於其，不過也許是其解釋圖片的能力太過於詳細，以至於做人臉的 Anomaly Detection 效果就會較差。

二、 實作

本次作業我是用 CNN 做 Multi-Encoder 來完成準確度來到 98.7% 左右，模型如下圖：

```

177 class conv_autoencoder(nn.Module):
178     def __init__(self):
179         super(conv_autoencoder, self).__init__()
180         self.encoder1 = nn.Sequential(
181             nn.Conv2d(3, 12, 4, stride=2, padding=1),
182             nn.BatchNorm2d(12),
183             nn.ReLU(),
184             nn.Conv2d(12, 24, 4, stride=2, padding=1),
185             nn.BatchNorm2d(24),
186             nn.ReLU(),
187             nn.Conv2d(24, 48, 4, stride=2, padding=1),
188             nn.BatchNorm2d(48),
189             nn.ReLU(),
190             # nn.Tanh(),
191             nn.BatchNorm2d(48),
192         )
193         self.encoder2 = nn.Sequential(
194             nn.Conv2d(3, 8, 4, stride=2, padding=1),
195             nn.BatchNorm2d(8),
196             nn.ReLU(),
197             nn.Conv2d(8, 24, 4, stride=2, padding=1),
198             nn.BatchNorm2d(24),
199             nn.ReLU(),
200             nn.Conv2d(24, 48, 4, stride=2, padding=1),
201             nn.BatchNorm2d(48),
202             # nn.Tanh(),
203             nn.ReLU(),
204             nn.BatchNorm2d(48),
205         )
206         self.encoder3 = nn.Sequential(
207             nn.Conv2d(3, 8, 3, stride=2, padding=1),
208             nn.BatchNorm2d(8),
209             nn.ReLU(),
210             nn.Conv2d(8, 24, 3, stride=2, padding=1),
211             nn.BatchNorm2d(24),
212             nn.ReLU(),
213             nn.Conv2d(24, 48, 3, stride=2, padding=1),
214             nn.BatchNorm2d(48),
215             # nn.Tanh(),
216             nn.ReLU(),
217             nn.BatchNorm2d(48),
218         )
219         self.decoder = nn.Sequential(
220             nn.ConvTranspose2d(48, 24, 4, stride=2, padding=1),
221             nn.BatchNorm2d(24),
222             nn.ReLU(),
223             nn.ConvTranspose2d(24, 12, 4, stride=2, padding=1),
224             nn.BatchNorm2d(12),
225             nn.ReLU(),
226             nn.ConvTranspose2d(12, 3, 4, stride=2, padding=1),
227             nn.BatchNorm2d(3),
228             nn.ReLU(),
229             nn.BatchNorm2d(3),
230         )

```

```

256 def forward(self, x):
257     x = self.encoder1(x) + self.encoder2(x) + self.encoder3(x)
258     x = self.BN(x)
259     x = self.decoder(x)
260     return x

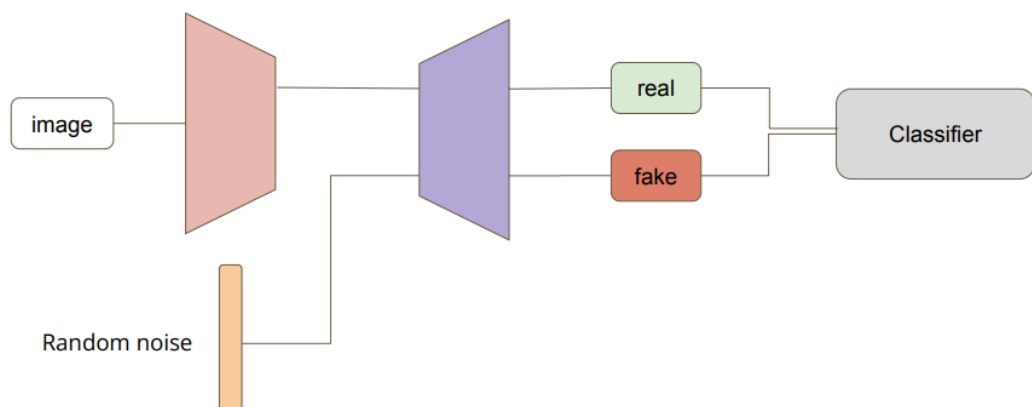
```

另外值得注意的是在 Encoder、Decoder 結束後，我都有做 Batch Normalization，不過我不知道為什麼這樣做會變好==，這是 trial and error 之後測出來的。

三、 額外實作

在 Hw06-Classifer 的檔案中，我有試著實作以下流程：

Add random noise and extra classifier



先訓練好一個 Encoder 和 Decoder 的模型，然後將訓練集，透過訓練好的模型產生圖片，此時的圖片為 Real，然後我們再將訓練集的資料經過 Encoder 後，先加入 Random noise 後再經過 Decoder，此時產生的圖片為 Fake，我們將這些資料再拿去訓練 1 個 CNN 分類器，不過這個效果沒有達到理

想，應該是 Fake 的 Domain 和測試集的 Anomaly 資料不同導致的，不過有人有成功做出來，我想我可能還有一些地方沒做好，如果有興趣的人可以試試看。