HW4 Report
謝忱 R06921088

1.Explain the structure of your networks and loss terms in detail (1.5%)

Model使用的是AC-GAN

Generator只使用反卷積層進行行upconv,中間會加上batchnorm來對張量進行歸一化, 最後輸出圖像爲128*128的fake img. Generator的架構如下圖所示

Discriminator使用相等層數的卷積層就行下採樣, 最後一層卷積輸出的張量會分別進到不同的nn針對不同class的label來進行分類.題目使用四種class,所以有四個不同的nn. 每個nn都有三層的FC層,最後一層輸出該class的數量,最後會再經過softmax進行計算.

Generator和Discriminator 有其相對應的loss term, 訓練時會先更新D的參數, D_loss主要分成兩部分, 第一部分是針對D提取出的feature, 此時會把真實的data和G生成的data進入D,分別出來的feature會對softlabel和fakelabel作BCEloss的計算並加總;第二部分針對每個class生成的label與真實的label也做BCEloss的計算並加總.

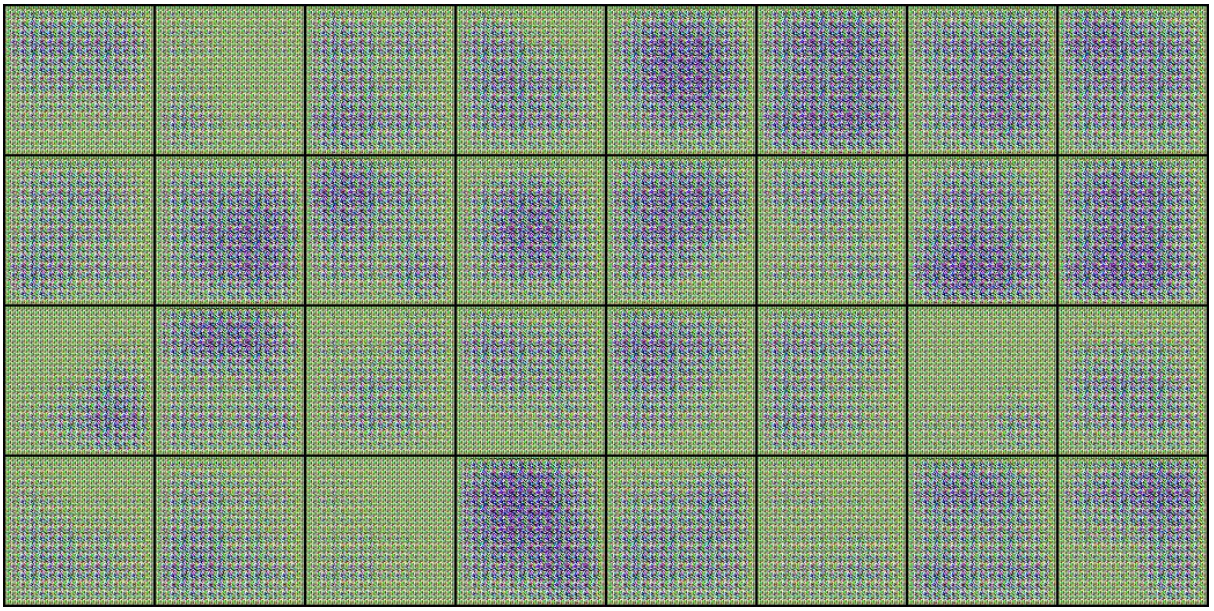
其次更新G的參數,G的loss計算只需要使用分類的label的loss乘上權重再加上D提取出的fake feature和fake label的loss即可.

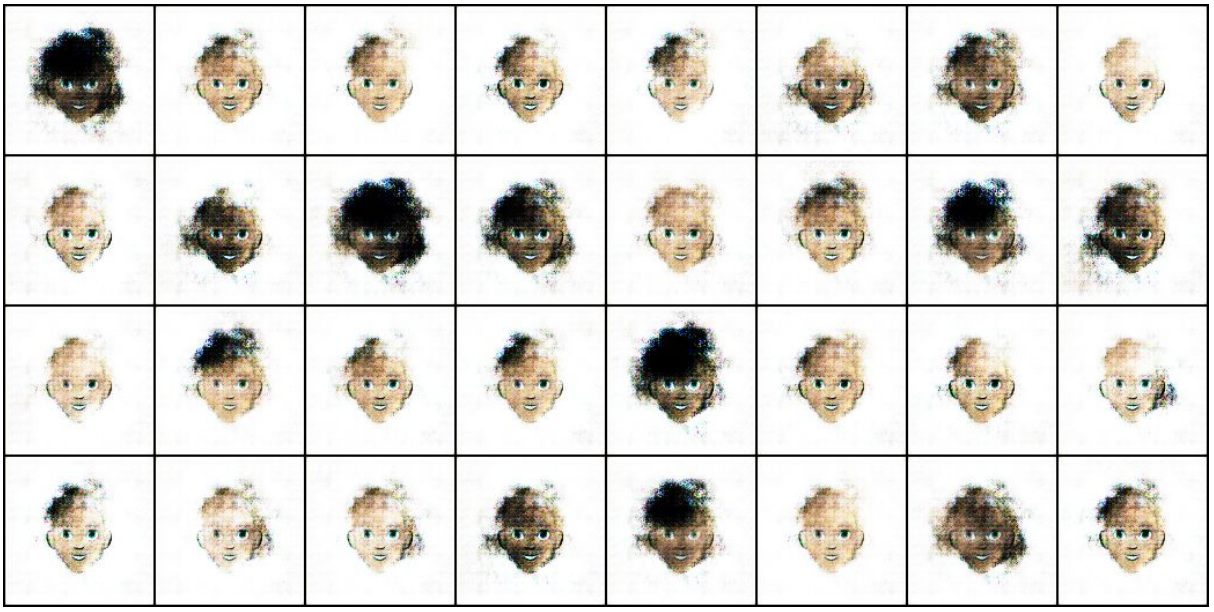
更新完兩個架構即是完成了一次更新完整的更新.

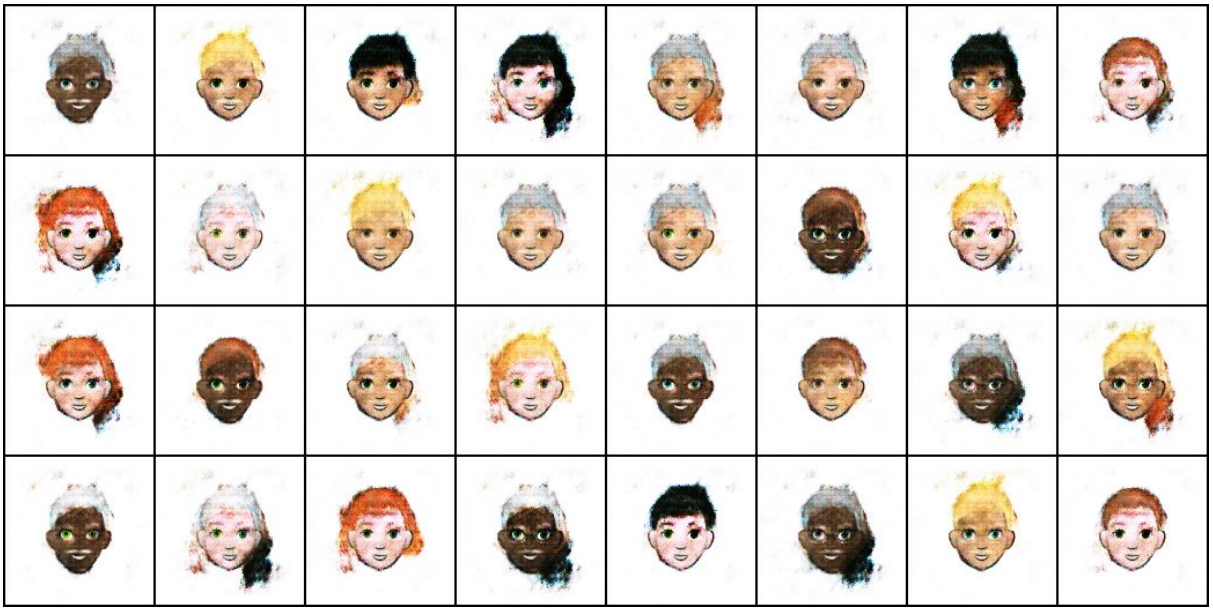
Layers	Details
ConvTranspose2d	out_channels=4096, kernel_size=4, stride=1,
BatchNorm2d	UNITS = 4096
ReLU	
ConvTranspose2d	in_channels=4096, out_channels=2048, kernel_size=4,

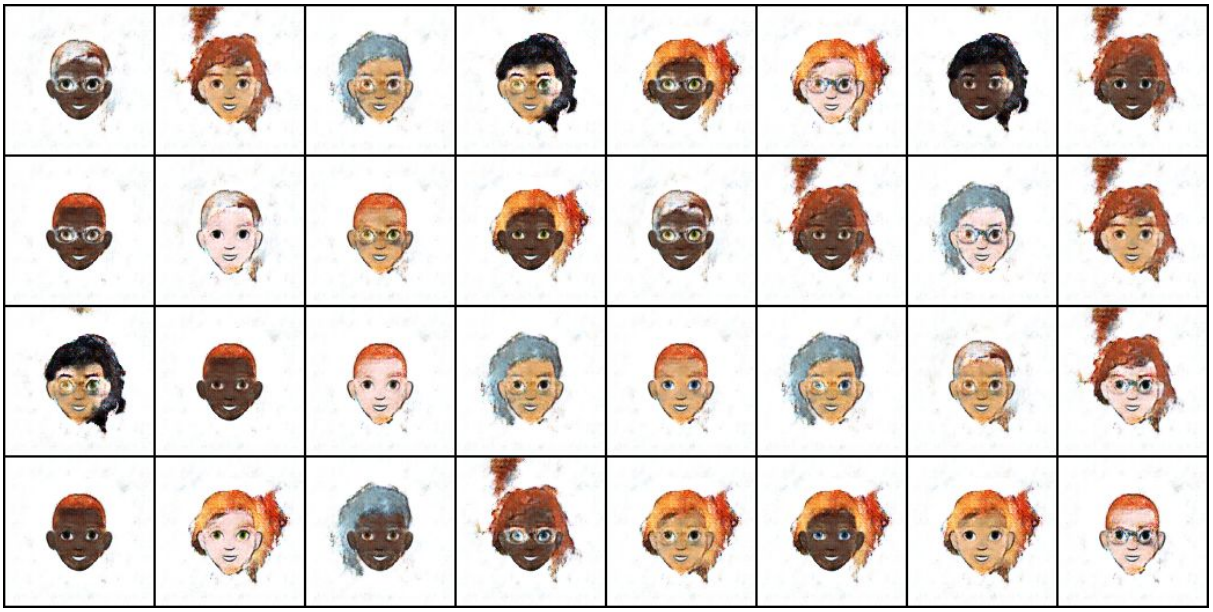
	stride=1,
BatchNorm2d	UNITS = 2048
ReLU	
ConvTranspose2d	in_channels=2048, out_channels=1024, kernel_size=4, stride=1,
BatchNorm2d	UNITS = 1024
ReLU	
ConvTranspose2d	in_channels=1024, out_channels=512, kernel_size=4, stride=1,
BatchNorm2d	UNITS = 512
ReLU	
ConvTranspose2d	in_channels=512, out_channels=256, kernel_size=4, stride=1,
BatchNorm2d	UNITS = 256
ReLU	
ConvTranspose2d	in_channels=256, out_channels=3, kernel_size=4, stride=1,
nn.Tanh()	

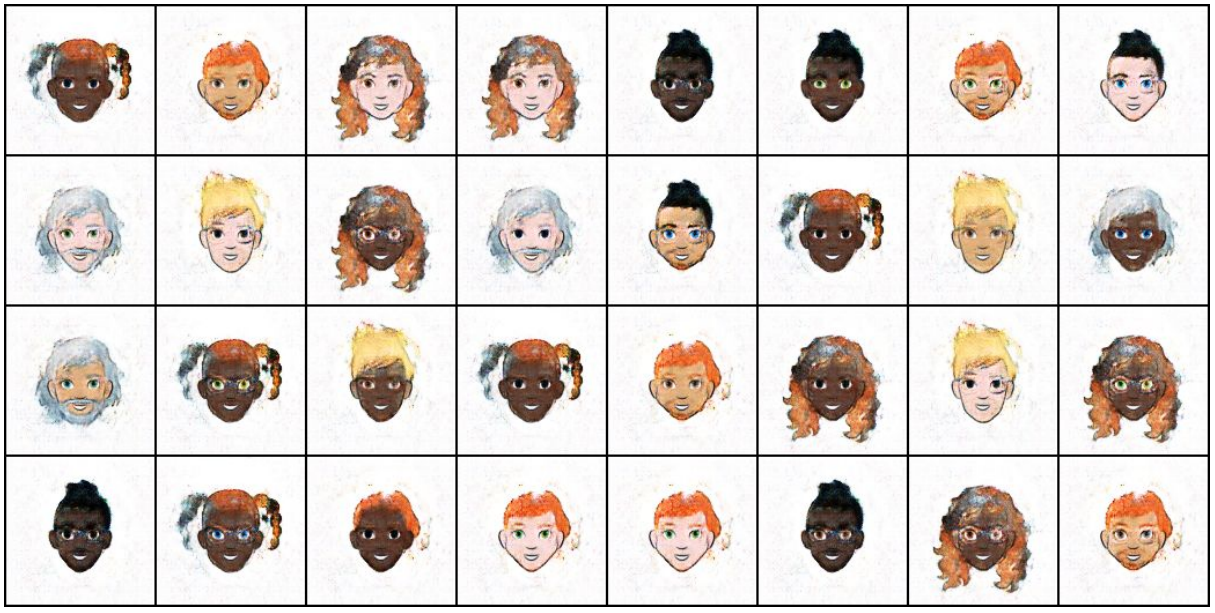
2.Plot your training progress (10 pics) (0.5%)Plot your training progress (10 pics) (0.5%)

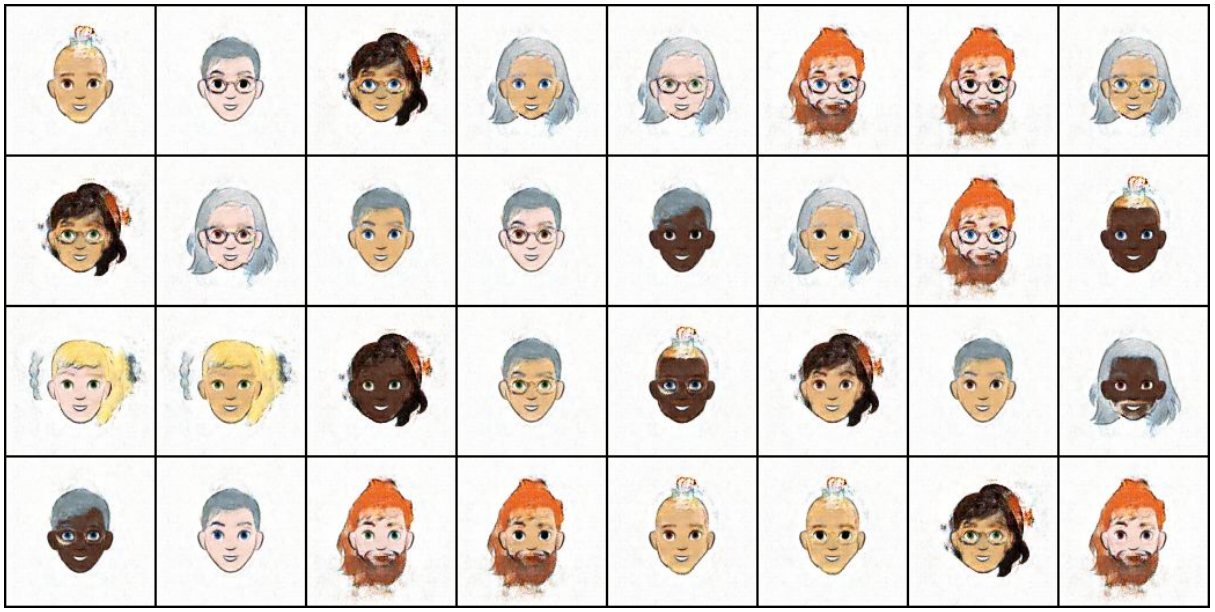


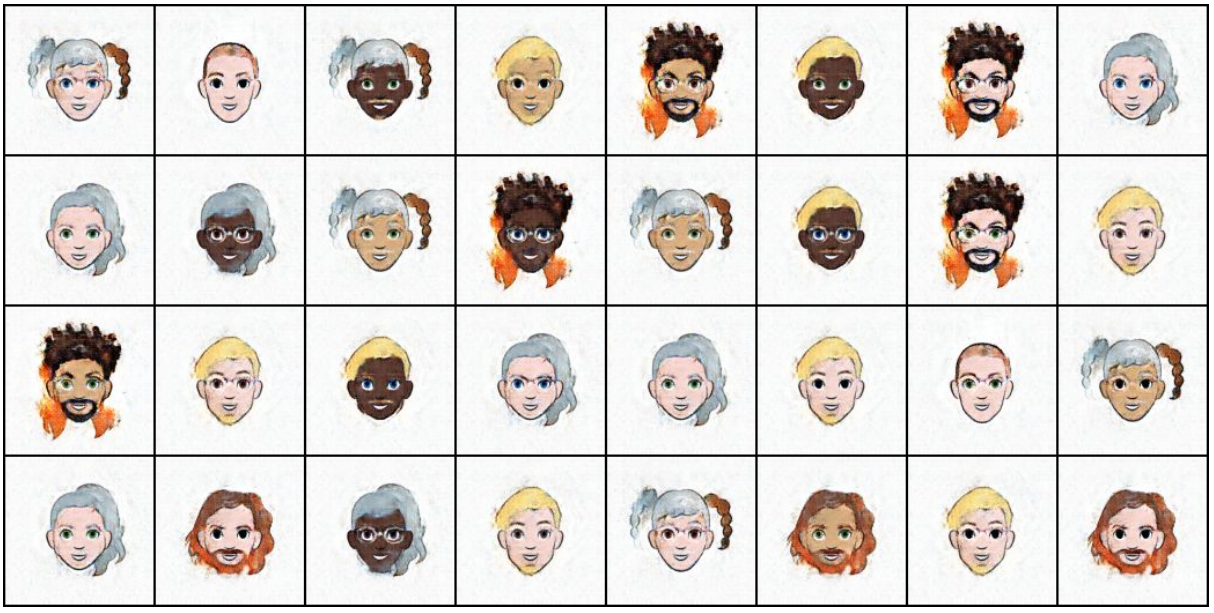


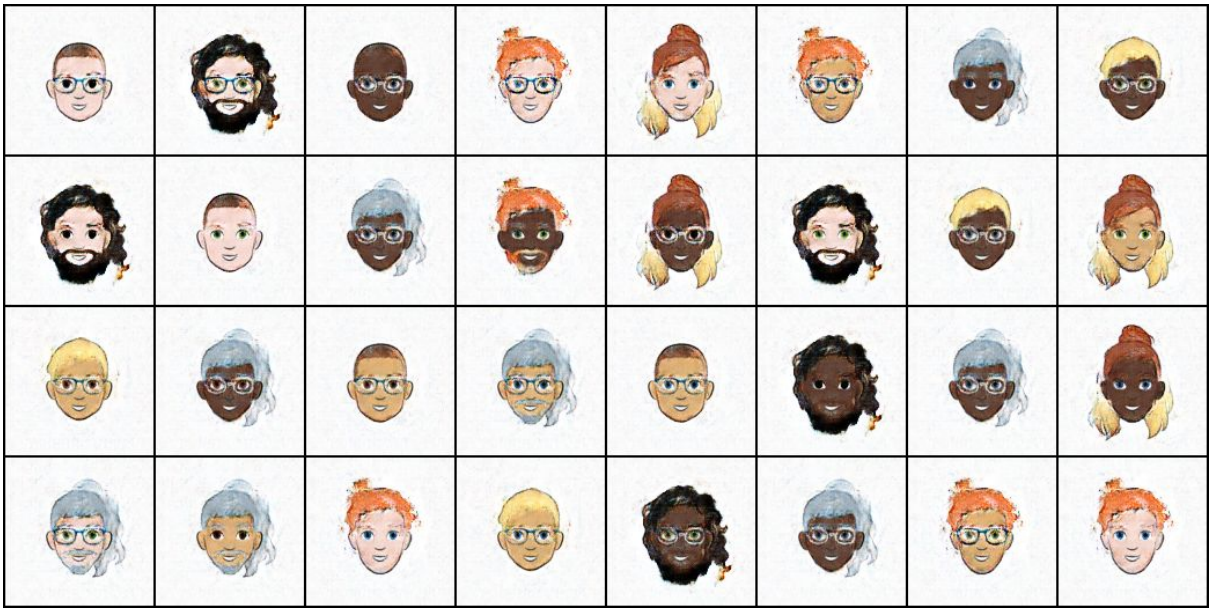


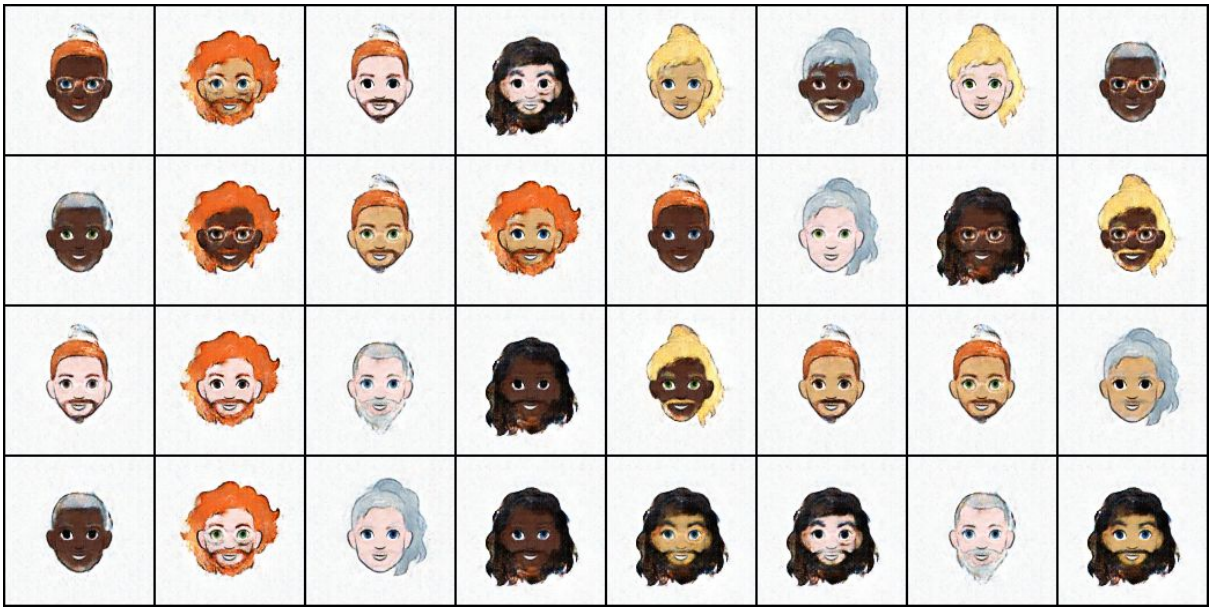


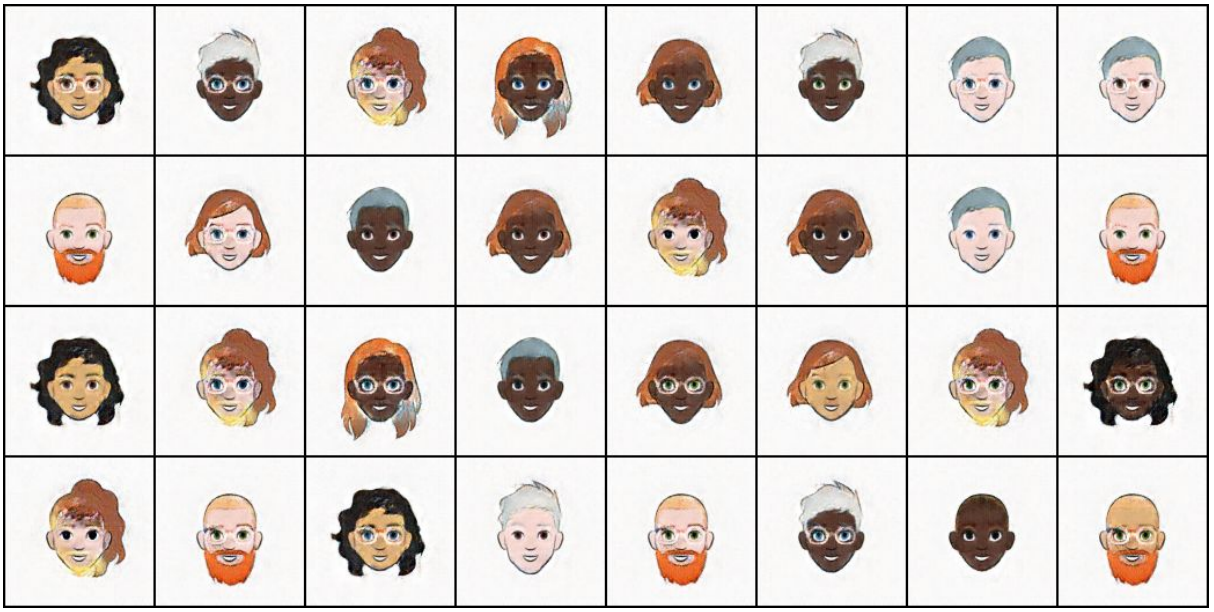












3.Design at least 3 different experiments. Describe your settings, making comparisons and report your observations. (4% + 0.5% bonus)

1.針對兩個不同的架構使用不同的optimizer, G使用Adam, D使用SGD. 並且訓練時進行動態learning rate的調整,初始時G和D的learning rate分別爲0.001和0.0001,batch爲32,當iteration上升至8000時將D的learning rate降至0.00005,G維持不變.這樣的做的原因是原本兩個架構的learing rate都保持0.0001時,G的loss會直接暴增, 導致生成的圖片爛掉. 分析是G的更新速度追上D所導致,因此想讓D降速.但是觀察到的結果最後還是爛掉,只是爛掉的iteration的次數往後推延了一些. 如果只是把optimizer 換成SGD和Adam則一直無法生出人臉.

2.降低D的更新頻率. 在降低D的learning rate無法達到預期效果後, 使用降低更新頻率的方法, 具體是在5000筆之前都採用每一次iter更新一次.5000筆之後則是每10iter更新一次,10000筆之後是20iter更新一次.觀察到的結果也只是爛掉的時間往後推延了一些.

3.更改架構嘗試使用INFO_GAN, 在使用AC-GAN一致無法通過strong之後選擇想使用INFO_GAN來train, 針對model label部分的loss進行更改.並開始訓練, INFO_GAN的訓練過程,G和D loss一直保持穩定且相對降低但是生成的圖片缺不夠AC_GAN生成來的精細如下圖所示,故放棄.

You may be deducted some points if TAs consider your experiments “太過敷衍”

EX: Normalizing the input image to [-1,1] is considered too simple and basic

幾個參考方向: GAN的不同objective、model condition on label 的方式、各種tricks ...

Bonus - Unsupervised Conditional Generation (3%)

Do **not** use the labels (**cartoon_attr.txt**) as input

You should describe your model architecture, loss terms, experiment settings and results in the **report**.

You should submit your code under the directory named ***Bonus/***

Unsupervised conditional model使用的InfoGan, model的G的架構如下表,D的架構同樣爲普通的CNN加上FC進行預測.

在model loss的計算方面,model先會對G的loss進行計算.首先把G產生出來的output丟進D, D提取出的feature會和全爲1的valid feature進行MSE的計算得到G的loss.

第二步對D的loss計算,分別讓D提取G產生的img和真實的img的feature, 再算這兩個feature分別和全爲1和全爲0的valid feature的MSE, 兩者相加求平均就可以更新D的loss.

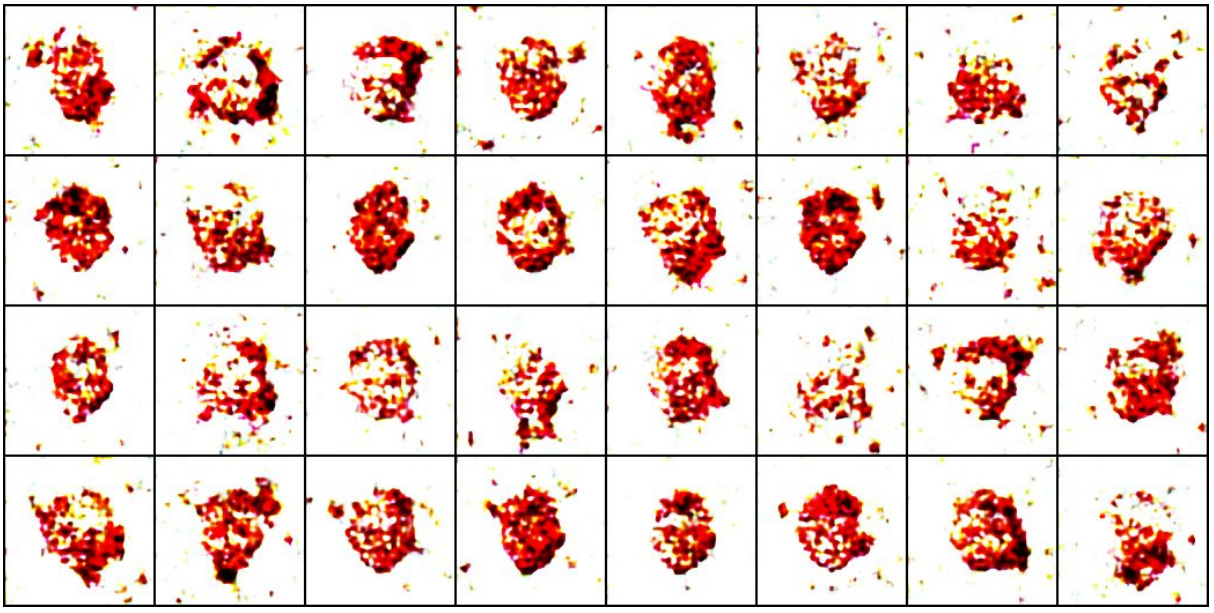
最後一步比較特別的是會對info的loss計算,這個loss分爲兩部分,第一是label的pred和ground truth的crossentorpy,第二是pred code和code input的MSE計算,兩者算出後分別乘上對應的權重即可得到info的loss.兩者其實都是由D產生的,code input是D比較特別的參數,這個input可以 自行定義維度,這train的過程中我將其定義爲兩維.

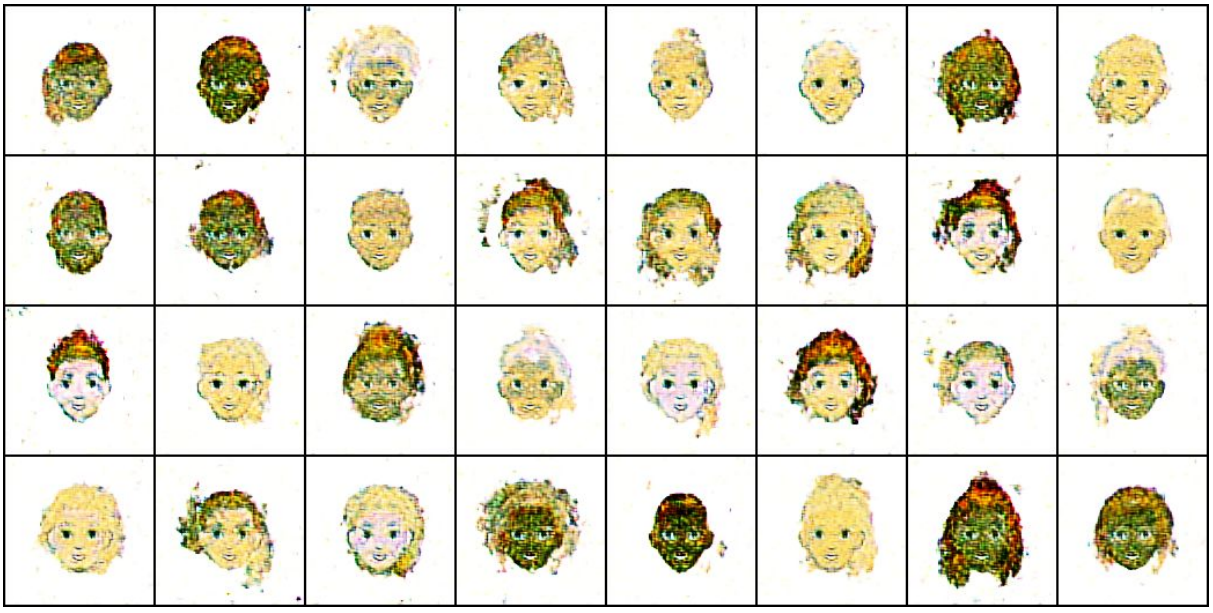
訓練時三個 optimizer都使用adam進行更新, 分別是D,G和Info,learning rate分別是0.0006, 0.0001, 0.0001. batch size爲32, 一共train十萬個interation.

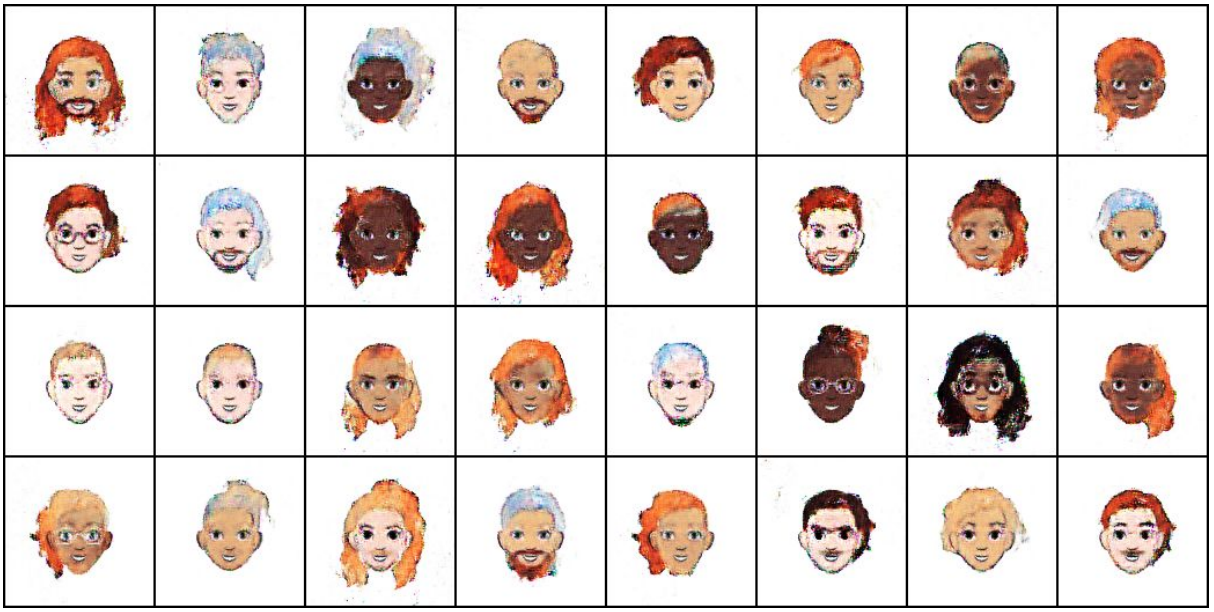
Layers	Details
BatchNorm2d	128
Upsample	scale_factor=2
Conv2d	128, 128, 3, stride=1, padding=1

BatchNorm2d	UNITS = 128
LeakyReLU	0.2
Upsample	scale_factor=2
Conv2d	128, 64, 3, stride=1, padding=1
BatchNorm2d	64
LeakyReLU	0.2
Conv2d	64, opt.channels, 3, stride=1, padding=1
Tanh	

產生的結果選取五張舉例.







The model checkpoint for bonus is **not required**

請大家上傳code是確保與report內容一致 (所以請不要在report裡憑空創造)

幾個參考方向: InfoGAN, StyleGAN, clustering, interpolation ...