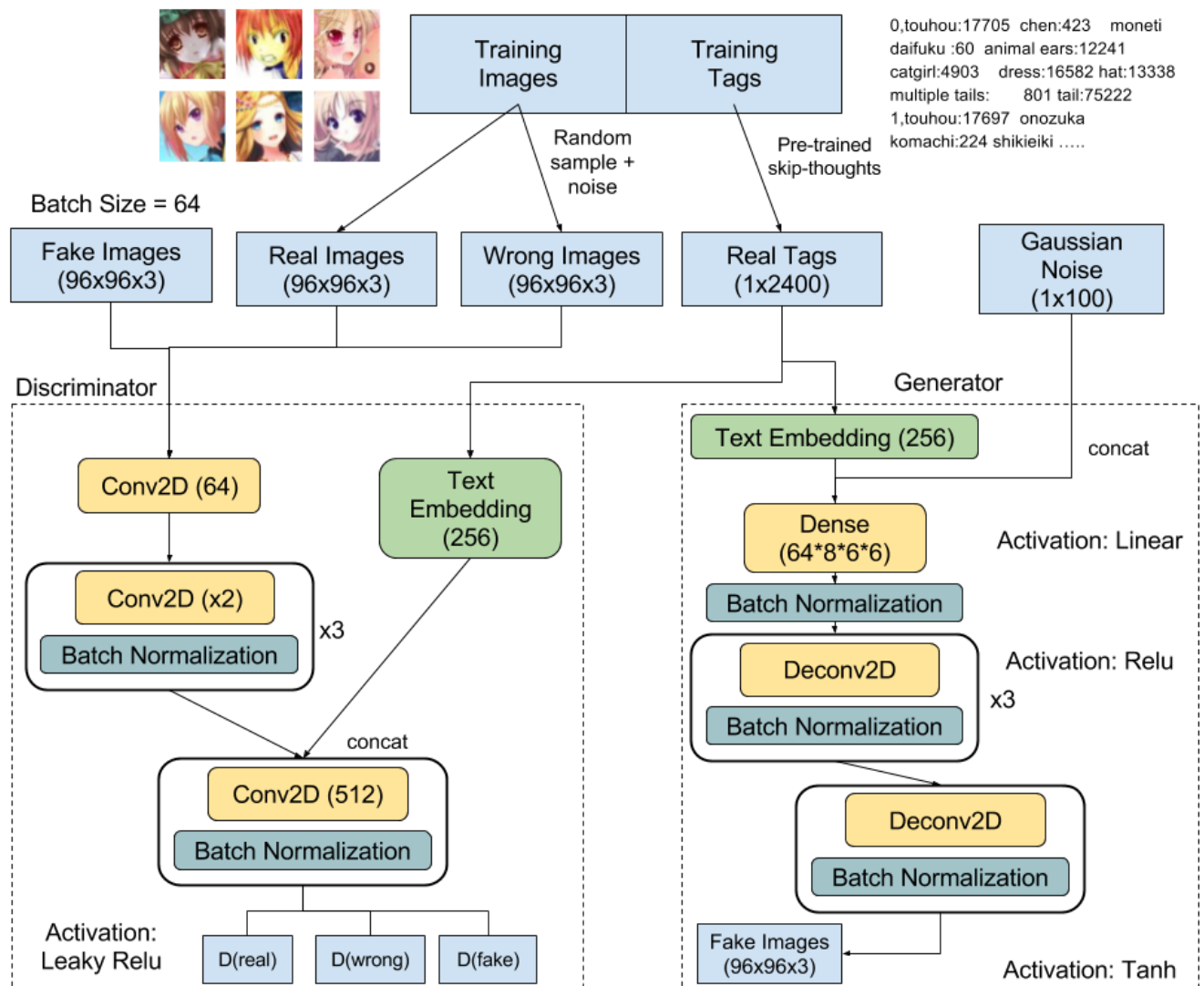


● Model description

- 模型架構：以下為GAN的模型架構圖和一些超參數。



訓練方式：

每個epoch中，前25個batch和第500xn個batch都讓discriminator更新參數25次，其餘更新5次。而generator則是每個batch都只更新一次。

更新參數方式：

使用RMSprop的梯度下降方式，learning rate設為0.00005。

目標函數：以下式子batch size設為n

■ Generator:

$$\frac{1}{n} \sum_{i=1}^n D(G(z_i)), \text{ where } z = \text{text} + \text{noise}, G(.) = \text{generator}, D(.) = \text{discriminator}$$

■ Discriminator:

$$\frac{1}{n} \sum_{i=1}^n D(\text{real}_i) - \frac{1}{n} \sum_{i=1}^n D(\text{wrong}_i) - \frac{1}{n} \sum_{i=1}^n D(G(z_i))$$

● How do you improve your performance

- 由於此為圖像的生成，因此我選擇實作Deep Convolutional GAN，並且選擇可以有較好的gradient數值的Wasserstein GAN(WGAN)。

■ Weight Clipping :

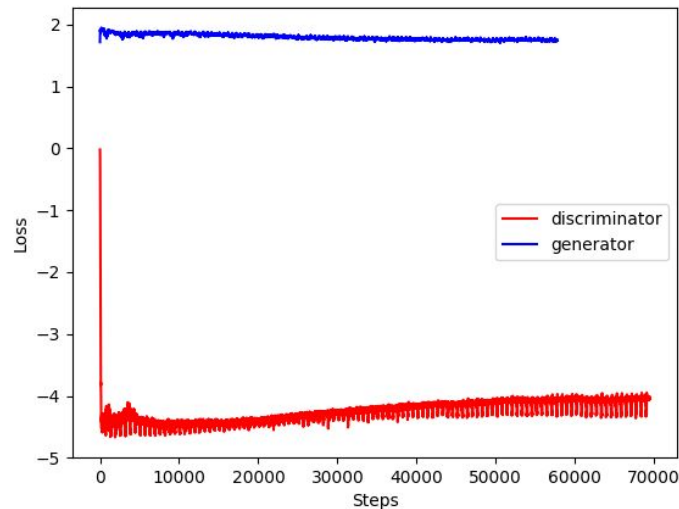
將參數修剪到-0.01~0.01之間的數值，根據WGAN的文獻，如此可以滿足k-lipschitz的限制，讓上述的objective function可以有比較好的能力描述現在generator的分佈和我的目標分佈的差距。

■ 96x96的訓練圖像：

輸入的訓練圖像為原始的96x96x3的圖片，並同除以255.0作scaling，讓數值變成0~1之間。相較64x64的圖片而言，此舉讓我保留較多的訓練圖片資訊。

- Experiment settings and observation

- 訓練學習曲線



由於discriminator的更新次數比generator還要多很多，因此我每五個step取一個loss的資料點，和generator的loss畫在同一張圖上。此圖y軸為100steps的running average，可以看到，雖然兩者的loss有靠近的趨勢，但是在訓練約110個epoch(每個epoch約500steps)之後，兩者loss接近的程度還有點距離。

- 生成圖像結果

Tags	Images
blue hair blue eyes blue hair green eyes blue hair red eyes green hair blue eyes	
touhou onozuka komachi shikieiki yamaxanadu	

討論：

所有輸入的tag都先經過skip-thought uni-directional的pre-trained模型，encode之後的vector接上不同gaussian noise的vector，通過generator之後所生成的圖片如上圖。不同的tags所生成的圖片都沒有什麼明顯差異，且看不出來眼睛等臉孔會有的特徵。但是仍有些微符合tag中“blue hair”和“green hair”的要求。

結果不甚理想的原因我認為可能有幾種：

第一是因為訓練的tag是沒有經過挑選的，我只有將tags中的冒號和數字去掉，將所有tag以空格接在一起，因此跟testing tag的長度和內容的多樣性相差很大，所以導致生成出來的結果不如預期。輸入training時的tag還能看出眼睛和些微嘴巴的地方，且髮色勉強有一些桃紅色。

第二則是可能為訓練次數還不足以讓generator產生可以接受的圖片，因為我的訓練更新方式為D:G=25:1或5:1，generator更新的次數比起discriminator少很多，從學習曲線來看，兩者的loss也還距離甚遠，因此還無法生成讓discriminator滿意的圖片。