

CEDL-HW3 Generative Models

姓名：李冠毅 學號：104064510

姓名：李季紘 學號：(交大)0556083

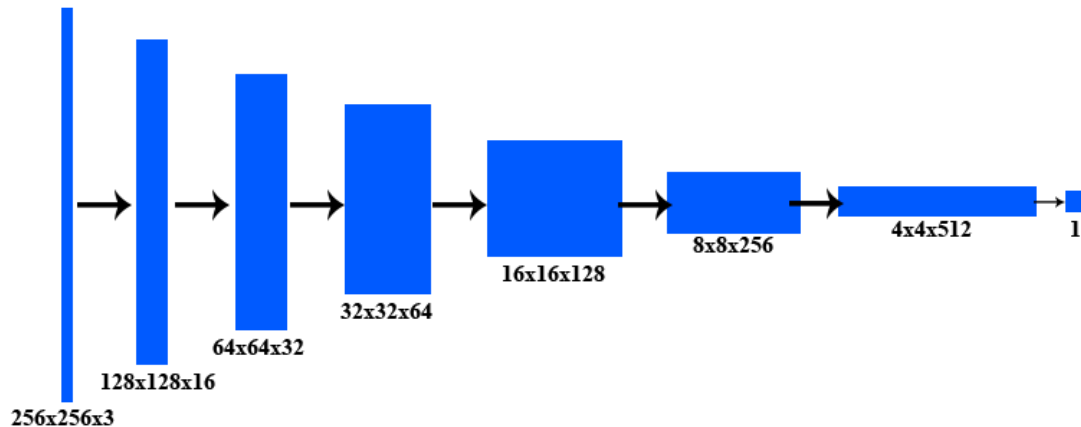
簡介

本次作業目標是基於 carpedm20 的 DCGAN[1]-tensorflow 程式碼[2]，修改內部架構使其能從 indoor scenes dataset 學習並能產生出 256x256 維度的 RGB 圖片，在整體 Network 訓練完畢後，再利用從各組所生成的圖片所組成的 Evaluation Dataset 去評估所訓練出來的 Discriminator 分辨真假圖片強度，本篇報告專注於在 Network training 階段的一些想法與心得。

架構變動

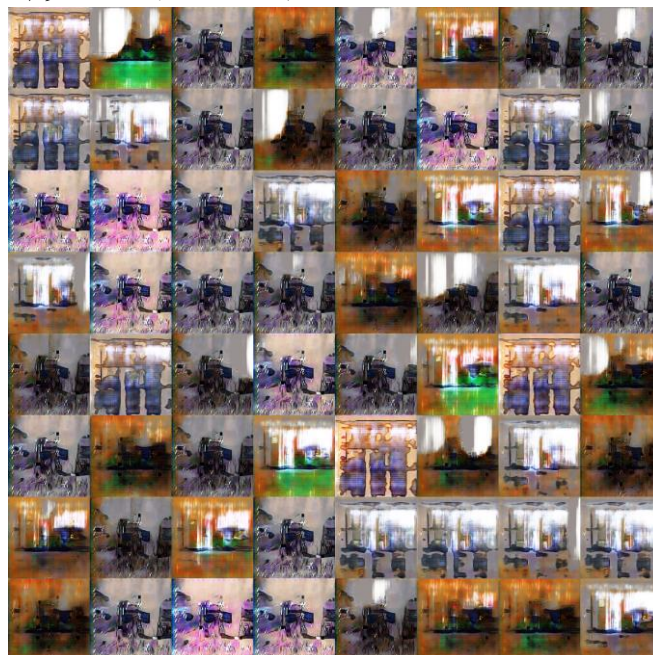
1. Discriminator 與 Generator 各加兩層 Conv 與 Deconv layer (初次實驗，非最終結果。)

這是我們最一開始直觀的想法，將兩者各加上 128x128x16 與 64x64x32 的兩個 feature layer 便可以達成 256x256x3 的 Input 以及 Output。



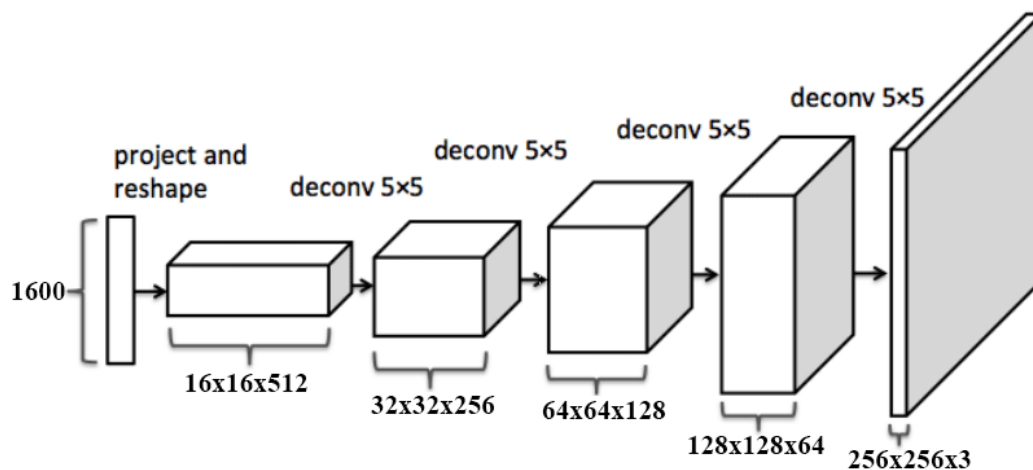
Discriminator 部分圖解

然而經實際試驗產生的結果圖如下圖所示，發現似乎帶有著色塊很多都糊在一塊，線條不太分明的一種情形，我們推測有可能是因為 Generator 部分由於經過太多次的 Upsampling，導致加入許多不必要的雜訊所致，故將 Generator 做調整。



2. Discriminator 加兩層, Generator 維持 4 層 (二次實驗)

與實驗一相比，Discriminator 並無變化，Generator 則維持如同 Code 最原始的四層架構，不過最初一層 Deconv 後的 Map Size 則會從 16x16 開始，這樣才能在後續 Upsample 成 256 維的圖片，此外由於原本的 $z=100$ 會投影到 4x4 的 Map size，然而現在最初的 Map Size 變成了 16x16，因顧慮有可能 z 值數量不夠投影，所以改成 $z=1600$ 。



Generator 部分圖解，原圖來自[3]

試驗後產生的結果圖如下，雖然多少使的圖片清晰一些些，但卻有同一 Batch 內的圖片色調很相近，但另一個 Batch 內的圖片色調卻截然不同的問題，目前尚無法解釋原因。

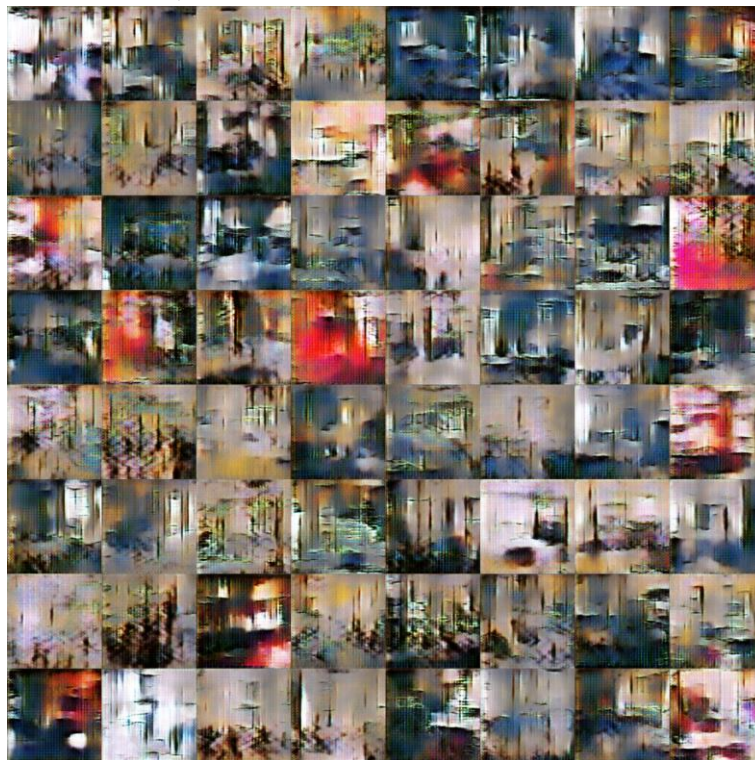


3. Discriminator 與 Generator 均維持 4 層 (最終使用)

我們最終所使用的 Network 架構便是與原始架構相同，維持 Discriminator 與 Generator 均為 4 層的模式，只是每一層的 Conv 和 Deconv layer 的 Map Size 均為原本 4x4 倍大小，此外後續發現 Generator 中的 z 值維度大小由於並沒有太大的差異，故將其轉換回原來的 100 維，並以 Batch Size = 40，訓練 40 次 Epoch，整體訓練時間約 1 天半左右。

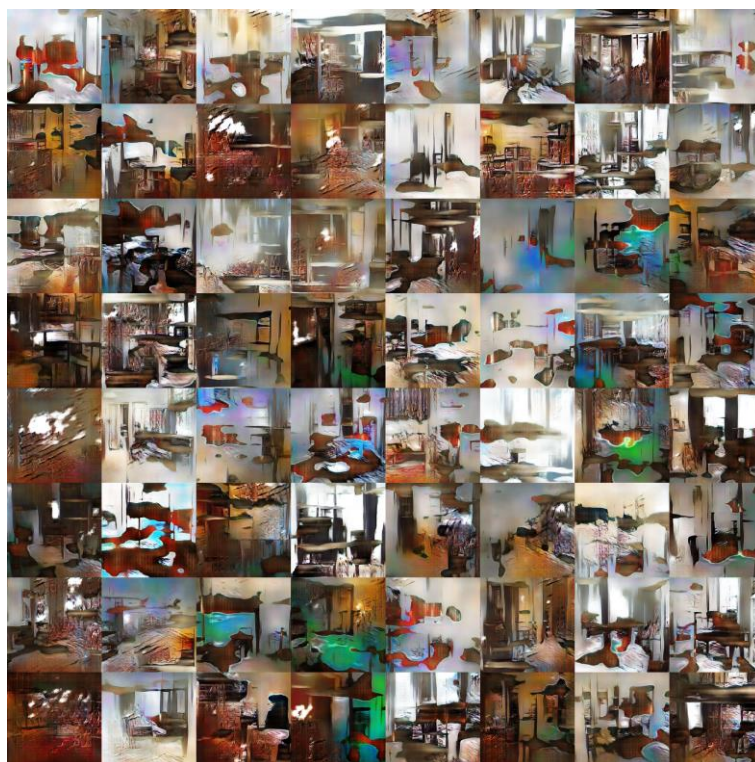
Checkerboard effect

在實驗的過程中由於多半側重於觀察所生成的圖片結果，調整 Deconv layer 的參數將會有最直接的影響，在調查資料的過程中發現有資料顯示[4]，當 Layer Size 為 Stride 值的整數被情況下很有可能幫助減緩 Checkerboard Artifacts 問題，因此我們便再在不調整 Stride 值的前提下調整 Layer Size 成 4 or 6 試驗，然而這卻產生了離生現象：Checkerboard Artifacts 現象反倒加劇，如下圖所示，不管 Size 大小為 4 還是 6 都會有這問題，雖然論文內並沒有詳述 Size=5 的選擇與理由，不過調整此值卻產生極大且不符預期的結果令我們相當意外。

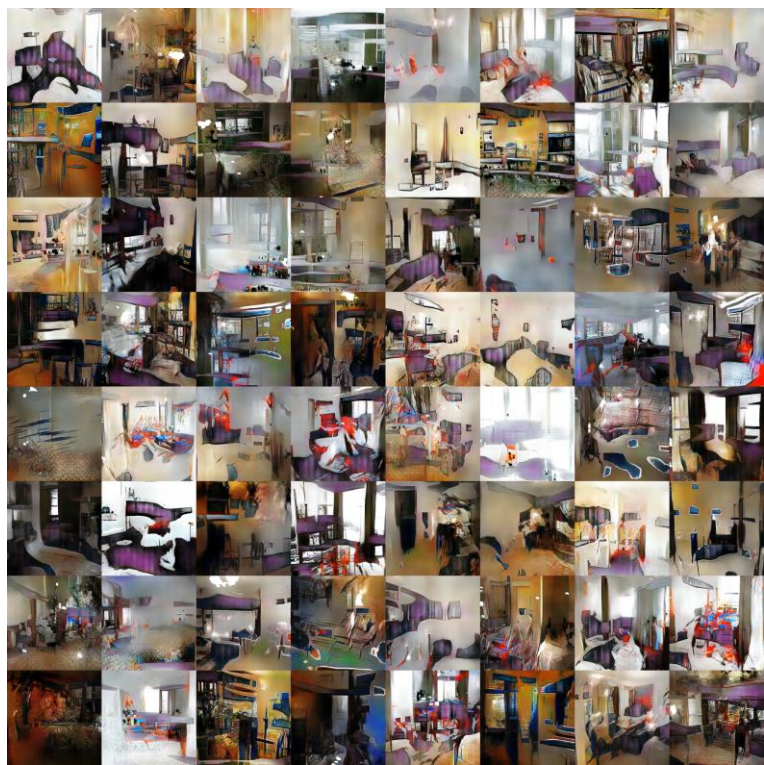


Generated Samples:

Epoch 10 :



Epoch 20 :



Epoch 30 :



Epoch 40 :



心得

在架構方其實多半都維持原 Code 本身的結構，連 Deconvolution layer 部分都沒有去作變動，僅只是讓 Output Size 改變而已，而這已是我們經過多次實驗後所得到較為不錯的結果了，雖然不太清楚論文撰寫者是經過多次實驗後才採取這些值還是有理論可循，不過從實驗結果足以感受的到這架構的心思縝密，也同時感受到寫出 Code 人的洗鍊程度。

References

- [1] Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In International Conference on Learning Representations (ICLR), 2016
- [2] <https://github.com/carpedm20/DCGAN-tensorflow>, “DCGAN-Tensorflow”, carpedm20, 2016
- [3] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do. Semantic Image Inpainting with Perceptual and Contextual Losses. arXiv preprint arXiv:1607.07539, 2016
- [4] <http://distill.pub/2016/deconv-checkerboard/>, Augustus Odena, Vincent Dumoulin, & Chris Olah. “Deconvolution and Checkerboard Artifacts”, Distill, 2016.
- [5] <http://bamos.github.io/2016/08/09/deep-completion/>, “Image Completion with Deep Learning in TensorFlow”, Brandon Amos, 2016