Decoupling-NeRF: Decompose the scene and renderer in NeRF

黃仁鴻
P76094169
NCKU CSIE

1. Introduction

於 2020 年提出的神經輻射場(Neural Radiance Field, NeRF) [1]利用簡單的類神經網路結構來擬合 Volume Rendering 的 3D 模型。但 NeRF 的設計如 Fig. 1，相當於是將 Renderer 與 Scene 嵌入於同一個類神經網路中，導致兩者具有高度耦合性而無法拆分。因此每當需要更換場景時，NeRF 就需要重新進行訓練。與一般我們在深度學習中訓練完成後，即可套用於不同場景中的方法有所差別。

然而，在一般 3D 場景的儲存與展示都是將 Scene 及 Renderer 拆分開來，並將 Scene 作爲輸入以取得對應視角的照片。這樣一來，Renderer 的部分就能重複利用於不同的 3D 場景上。對應於原本 NeRF 中，訓練所使用的照片便相當於嵌入在 NeRF 內的場景，若可以將照片改用於模型的輸入，便可將 Scene 與 Renderer 解耦合。

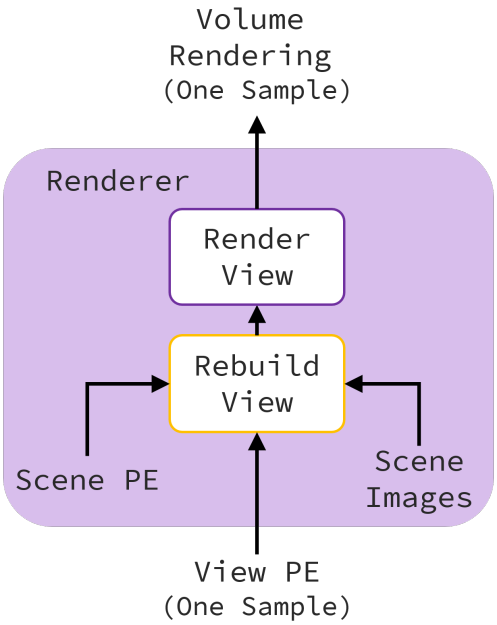

Figure 1. NeRF 的設計使 Scene 與 Renderer 具有高度耦合性，使 Scene 內嵌於模型當中無法分離。

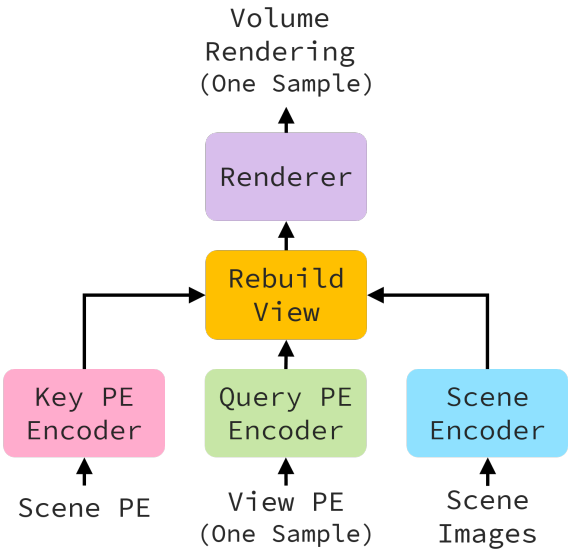

Figure 2. 本次專題所提出的 Decoupling-NeRF 就是希望將 Scene 資訊與 Rebuild View 獨立出來。當要替換繪製場景時就不需要再經過耗時的訓練，只需要變更輸入的 Scene Images 跟 Scene Position Encoding 即可。

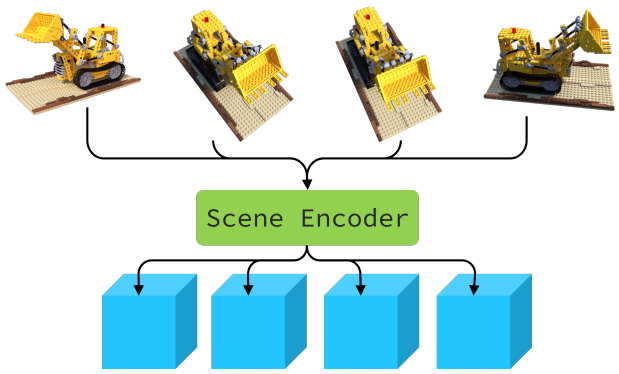

Figure 3. 將場景照片透過 Encoder 編碼成 Scene Embedding。

因此，本次專題研究目標便是提出 Decoupling-NeRF 這個架構，使其可以快速應用在各種場景而不

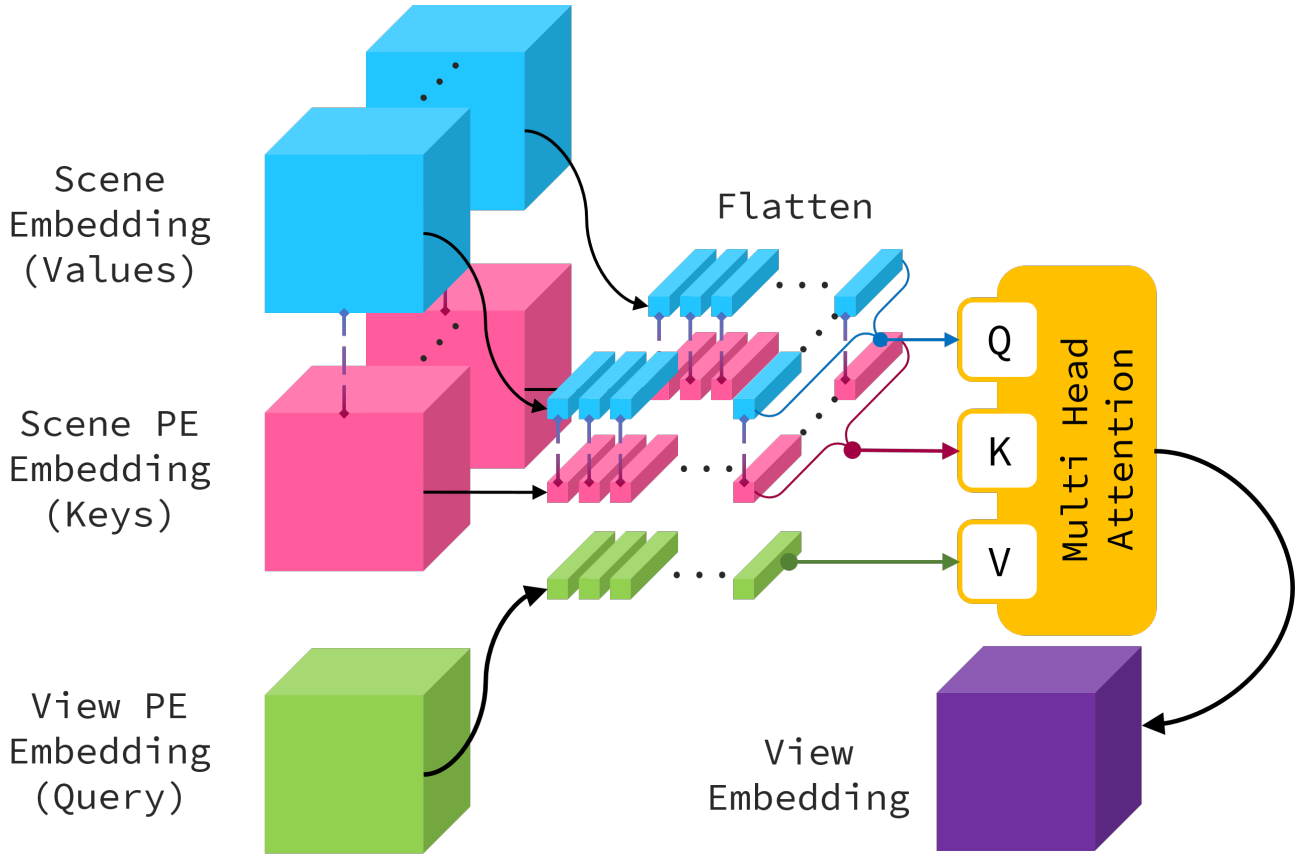

Figure 4. 將目標視角的位置編碼作爲 Querys，與 Scene Embedding 及其對應的位置編碼計算 Multi Head Attention 來取得 View Embedding。

需要重新擬合。Fig. 2 是本次專題的架構，將隱含於 NeRF 中的 Renderer 與 Rebuild 分開，並利用 Scene Encoder 對場景照片進行編碼，在 Rebuild View 使用 Multi Head Attention [2] 將其重新組建爲 View Embedding，最後透過單一 Neural Renderer 生成場景照片。

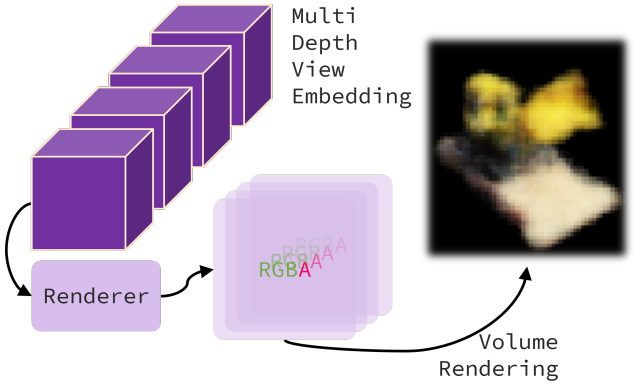

Figure 5. 將 Fig. 4 得到的 View Embedding 交由 Renderer 進行 Volume Rendering。

2. Method

在原版的 NeRF 中，會使用目標視角的位置編碼(Position Encoding, PE)作爲輸入，以生成該視角會拍攝到場景照片，而 Position Encoding 是將相機的位置資訊 Eq. 1 轉換所得:

$$PE(p) = [sin(2^0\pi p),\ cos(2^0\pi p),\ ...,\\ sin(2^{L-1}\pi p),\ cos(2^{L-1}\pi p)] \tag{1}$$

本專題架構如 Fig. 2 所示，可以分成 Scene Encoder、Rebuild View 與 Renderer 三個區塊。在進行照片生成時，先將多張場景照片編碼成 Scene Embedding(Fig. 3)，Scene Image 對應的 Scene PE 及生成目標的 View PE 也會通過各自的 Encoder 轉換爲 Values、Keys 及 Querys，將這三者以 Fig. 4 的形式建立出 View Embedding，最後再交由 Renderer 產生視圖(Fig. 5)。

在 Fig. 4 中，利用了 Vaswani et al. [2] 所提出的 Multi Head Attention 來將 Scene Embedding 重新組織成需要

的 View Embedding。其計算方法如 Eq. 2:

$$
\begin{aligned}
MHA(Q,K,V) &= Concat(head_1,...,head_h)W^O \\
head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \\
Attention(Q,K,V) &= softmax(\frac{QK^T}{\sqrt{d_k}})V
\end{aligned}
\tag{2}
$$

在訓練時使用 SSIM[3] 作爲損失函數來計算生成結果與實際影像的誤差。

3. Experiments

本專題會使用 NeRF 提供的 nerf_synthetic 資料集[1]進行訓練與測試，nerf_synthetic 提供了 8 個物件的多張照片及拍攝時的姿態矩陣，將其中的 chair, drums, ficus, hotdog, lego, materials 訓練物件，mic, ship 則當成測試物件。爲了加快訓練速度並減少記憶體使用量，將 800x800 pixels 的原圖縮小爲 64x64 pixels。

訓練使用的優化器是 Adamax，learning rate 是預設的 0.002，batch size 爲 4，每個 batch 會隨機挑選 4 16 張照片轉換爲場景編碼，Volume Rendering 所用到的深度採樣數則是 16 64，會在累計 8 個 batch 的梯度後才進行權重的更新。Positional Encoding 的 $L_{embed} = 12$，Multihead Attention 有 32 個 head，每個 head 維度爲 32d，在 Encoder 與 Renderer 分別具有 8 倍的 downsample/upsample。

Fig. 6 與 Fig. 7 分別是訓練物件與測試物件的重構結果。可以發現，在訓練時沒看過的物件是無法正確還原的，其生成結果都會帶有訓練物件的特徵。

4. Conclusion

目前只有在訓練時看過的物件才能成功重建，而沒有看過的物件在重建後會被扭曲成看過的物件。推測原因有可能是在 Multihead Attention 計算時，類神經網路將位置資訊混進輸出中。導致模型退化成 NeRF + Object Condition，使其缺乏泛化能力。

爲了解決這個問題，未來會研究在中間層的特徵編碼進行約束，使得 Multihead Attention 的輸出結果與目標影像通過 AutoEncoder 所產生的編碼相近，這部分的約束可以參考 Contrastive Learning 的相關研究，但是若要達成此作法，就必須在特徵編髮的階段就進行類似 Volume Rendering 的計算，好將不同深度的特徵編碼進行合成。

References

[1] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000—6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[3] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[1] Link https://drive.google.com/drive/folders/128yBriW1IG_3NJ5Rp7APSTZsJqdJdfc1

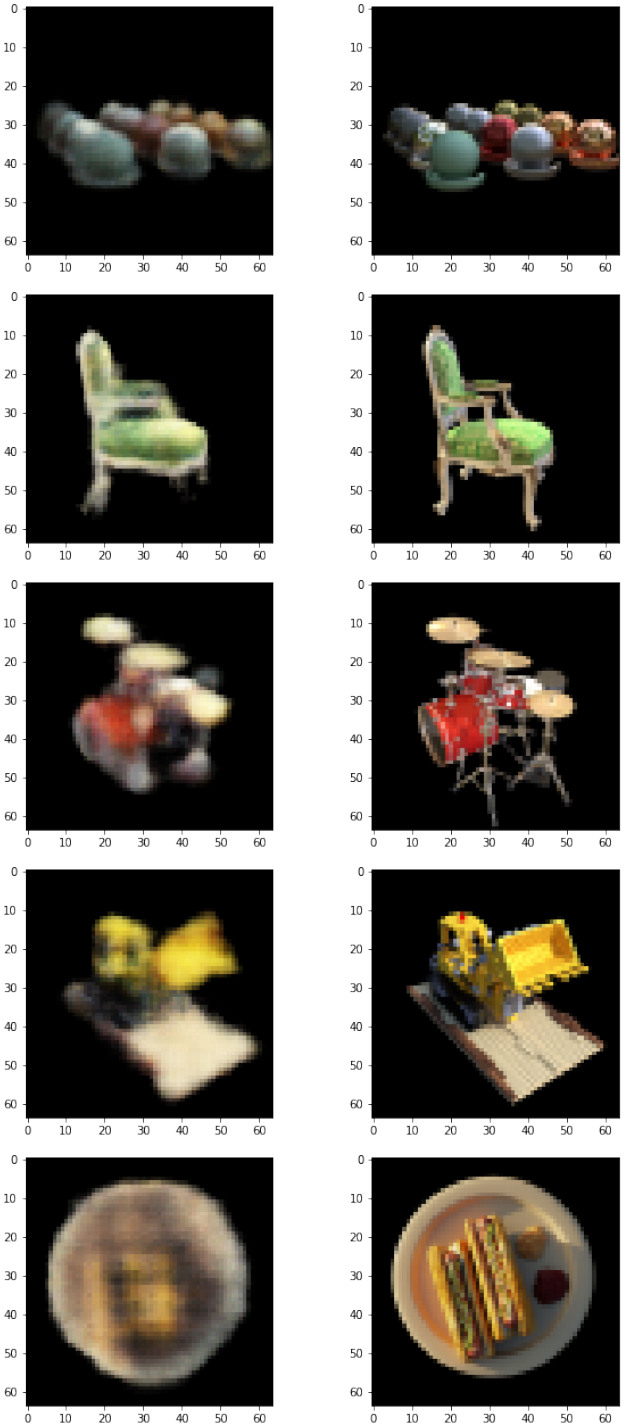

Figure 6. 左邊是訓練物件的還原結果，右邊是眞實影像。

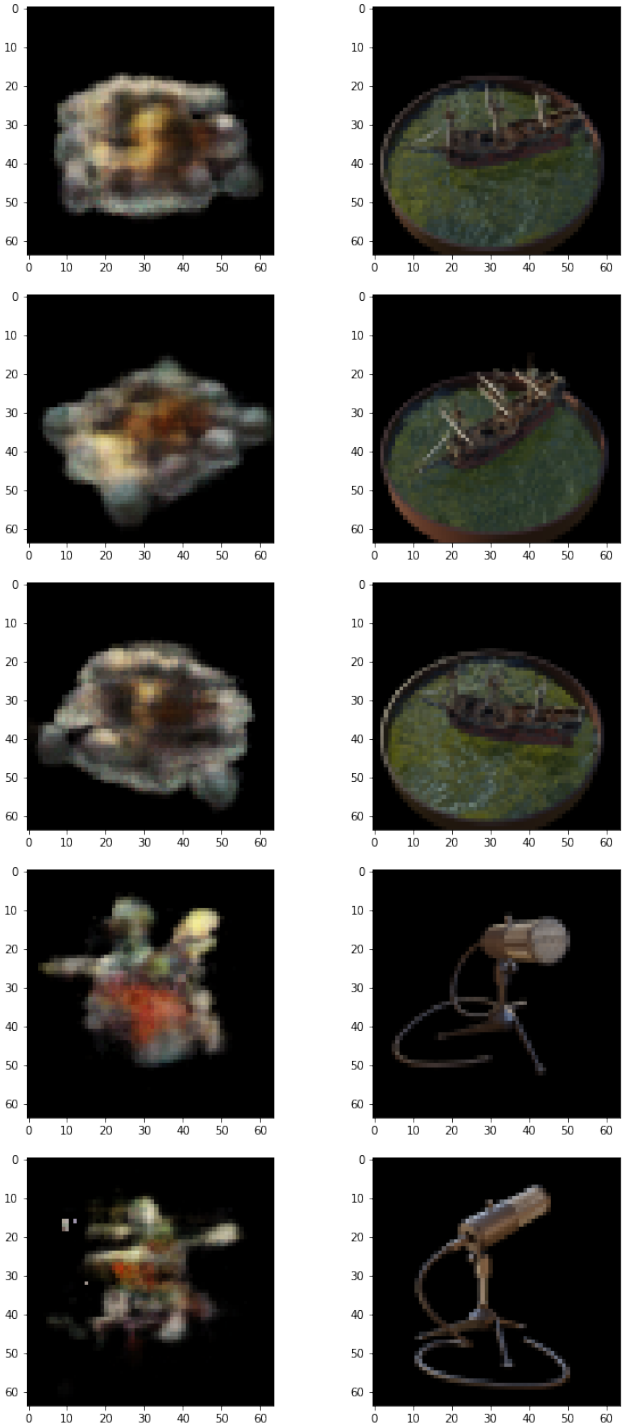

Figure 7. 左邊是測試物件的還原結果，右邊是眞實影像。