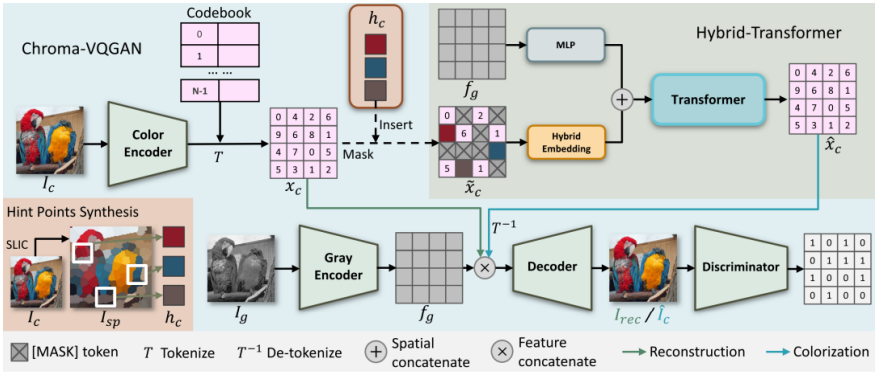
# Term Project 書面報告

Group 19

109550135 范恩宇/109612041 吳伯諺

## Method :

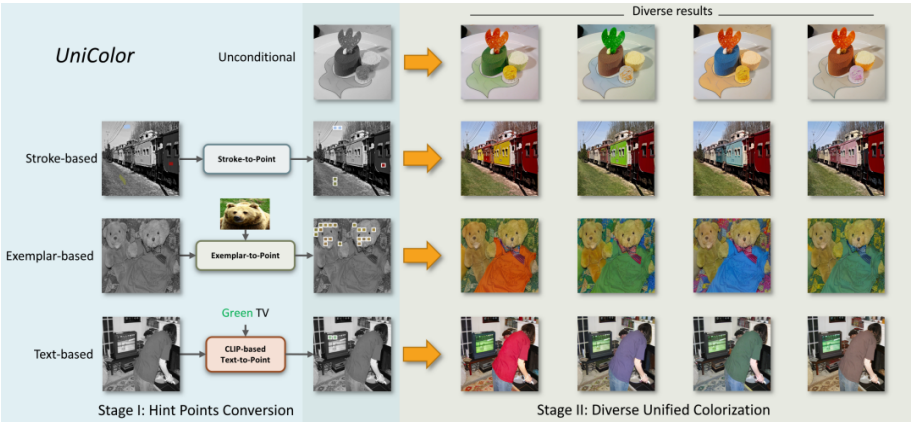



我們這次使用的論文是:" UniColor: A Unified Framework for Multi-Modal Colorization with Transformer " , by ZHITONG HUANG, NANXUAN ZHAO, JING LIAO。其中的研究提出了統一框架 UniColor 以支持多種模式的著色，包含了無條件與有條件的模式(如:筆畫、示例、文本，以上的混合等)。Unicolor的特別之處在於，它沒有為每種類型的條件學習單獨的模型，而是引入一個兩階段著色框架，用於將使用者給出的條件結合進模型中。

第一階段先進行多模態條件的轉換，將其轉換為提示點的通用表示。作者們提出一種基於CLIP的新方法，將文本轉換為提示點。利用了CLIP模型的能力，將文本描述與圖像特徵連結，進而生成相應的提示點。

接著第二階段，則是基於Transformer的網絡，由Chroma-VQGAN和Hybrid-Transformer組成，用於生成以提示點為條件的多樣化和高品質的著色結果。Chroma-VQGAN模型用於生成初始的彩色圖像，Hybrid-Transformer模型則進一步處理前述圖像以產生多樣性的著色結果。

最後產生的UniColor能根據不同的提示點生成多種的著色結果。此外，作者們也設計了提供多種功能的互動介面，包括 Unconditional、Stroke-Based、Exemplar-Based , Text-Based 等，讓使用者能根據自己的需求著色。




## Improvements :

這次做下來有個很大的感觸：每張圖片要用的處理方式真的都不太一樣。在經過 Unicolor 的 code 產出第一階段的上色後，我們都會依據產出的效果加上不同的後處理。即便是同樣的處理（例如：降燥），我們每張的個別參數也不盡相同。

我們主要增加過的包含以下幾種：

- 白平衡：

  不同於人眼在不同色溫下都能準確判斷出白色，相機所產生的圖像會因周遭不同色溫的光源而出現偏色，從而呈現出與「人眼看到的顏色」不同的結果。我們為了讓產出的圖像看起來更為自然有活力，多數照片在用 Unicolor 初步上色後都會再進行白平衡處理。做法如下：

  1. 計算圖片每個 color channel 的平均值
  2. 計算平均 gray value 來得出每個 channel 對應的 scaling factor
  3. 用2.得到的 scaling factor 對每個 channel 進行 scaling
  4. 將影像裁至應有的範圍 [0, 255]

- 增加色彩飽和度：

  Unicolor 產生的初始彩圖對我們而言都有點色彩過淡，有些甚至看起來像沒上一樣，因此選擇增加其中色彩的飽和度，期望達到接近實物景象的效果。做法如下：

  1. 將圖像的色彩空間由 BGR 轉至 HSV

2. 將圖像拆為 H,S,V 三 channel 並調整S
3. 將調整過的三 channel 合併回 HSV 影像
4. 影像轉回BGR

- 圖像物體銳化:

部分照片的邊緣實在是破碎不全，讓人看起來相當不舒服，甚至可能因為本身noise過多，在降燥時導致影像更為模糊不清。因此對某些照片降噪前，我們會先對它使用 unsharp masking 來銳化其中物體的邊緣，做法如下:

1. 先對照片使用 Gaussian blur，並計算產出結果與原圖的差異，從而得到mask
2. Mask 乘上指定的 strength factor（多用1.5）
3. 將 Mask 用於原圖並得到銳化後的照片
4. 將影像裁至應有的範圍 [0, 255]

- 圖像降噪:

提供的照片大部分都有明顯的 noise，我們起初選擇在 Unicolor 上色前降噪，卻發現產出結果會因物體輪廓更不明確而變得不理想，最終選擇在上色後降噪。

降噪採用的是非局部平均（Non-local means）演算法，它相較於局部（local）的演算法（例如：Gaussian blur）只使用各個目標像素附近的點來將影像平滑化來去除雜訊，Non-local means 演算法會在影像中各個目標像素周圍區域平滑取均值，可以說它使用影像中的所有像素。並且根據「目標像素區域」和「該像素周圍區塊的區域」的相似度進行加權平均。濾波後圖像清晰度高，而且損失較少的細節。

做法其實就是套用 Opencv 的 fastNlMeansDenoisingColored( )函數，濾波器強度控制在3~10（依據產出結果），templateWindowSize 和 searchWindowSize 分別為 7 和 21。

## Results :

以下是整輪實作後我們較滿意的幾張:

1. 圖片 1 -

第一階段上色使用 Text-Based 模式，給出" blue sky, white cloud, red tiles, gray building and bell tower, green tree "的提示，就能得到初步結果。

接著由於白平衡在這張圖上效果不佳，且其中場景的邊緣明顯而不須銳化，我們僅先提升色彩飽和度，再使用非局部平均演算法來降噪。

最後結果相對於第一階段產出的圖看起來較像近代相機拍下來的照片，而沒有原本那種類似油畫的斑駁感覺。

↓↓↓

2. 圖片 8 -

第一階段上色同樣使用 Text-Based 模式，提示用" earthy yellow and gray castle "得到初步結果。

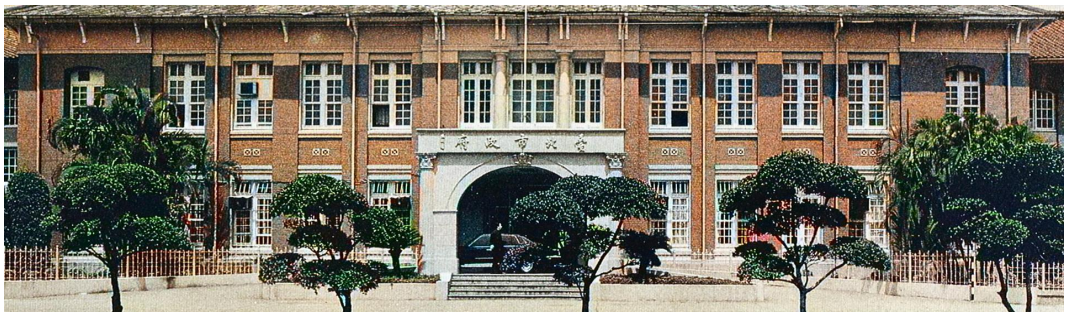
這是我們最滿意的其中一張。初步上色版看起來有點像陰天，為了讓它看起來更有活力，我們先調整白平衡來讓它變得比較像晴天。後面跟圖片 1 一樣先增加色彩飽和度後降噪，不過這張的降噪係數調比較低，從而避免地磚的一部分被磨平。

↓↓↓

3. 圖片 10 -

第一階段上色這次使用 Stroke-Based 模式得到初步結果。

過程和圖片 8 一樣先調整白平衡，後面增加色彩飽和度再非局部平均降噪。這次的降噪係數則介於圖片 1 和 8 間，因為我們希望在盡可能減少黑點的同時，保留柱子的輪廓。



↓↓↓

4. 圖片 11 -

第一階段上色換成使用 Exemplar-Based 模式得到初步結果。

過程大部分跟前兩張的處理方式一樣，只是在做色彩飽和度增加前我們多進行了 unsharp masking 來增強照片中物體的邊緣並凸顯立體感。跟其他圖片相比，這張加了 unsharp masking 後隨之產生的 noise 影響確實不多。

↓↓↓

5. 圖片 30 -

第一階段上色一樣使用 Exemplar-Based 模式得到初步結果。

這是我們最滿意的第 2 張，處理方式和另一張圖片 8 其實很類似：先調整白平衡來讓它變得像晴天，接著增加色彩飽和度，最後用非局部平均降噪來去除多的噪點。


↓↓↓

## Conclusion & Feedback :

經過這次project, 確實親身體會到開始做前老師說的 " It's not that easy " 。我們翻了不少論文與 Github 上他人的作品, 發現做出來效果較好的都需要設定與安裝不少額外套件, 搞動整個東西的運作方式與程式邏輯也花費了比預期中更多時間。

可惜適逢期末以致實作時間不夠, 不然還希望能在上色階段前加上影像修復來完成更高品質的結果。總結來說, 實作圖片上色實在相當有趣, 希望以後能實作產出結果更為優秀的方法。