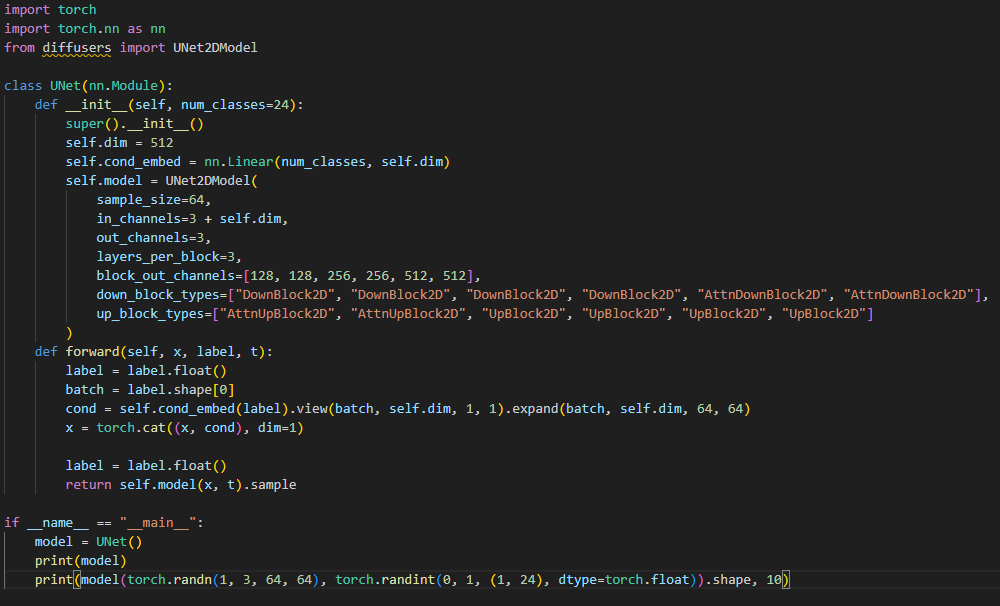
1. Introduction

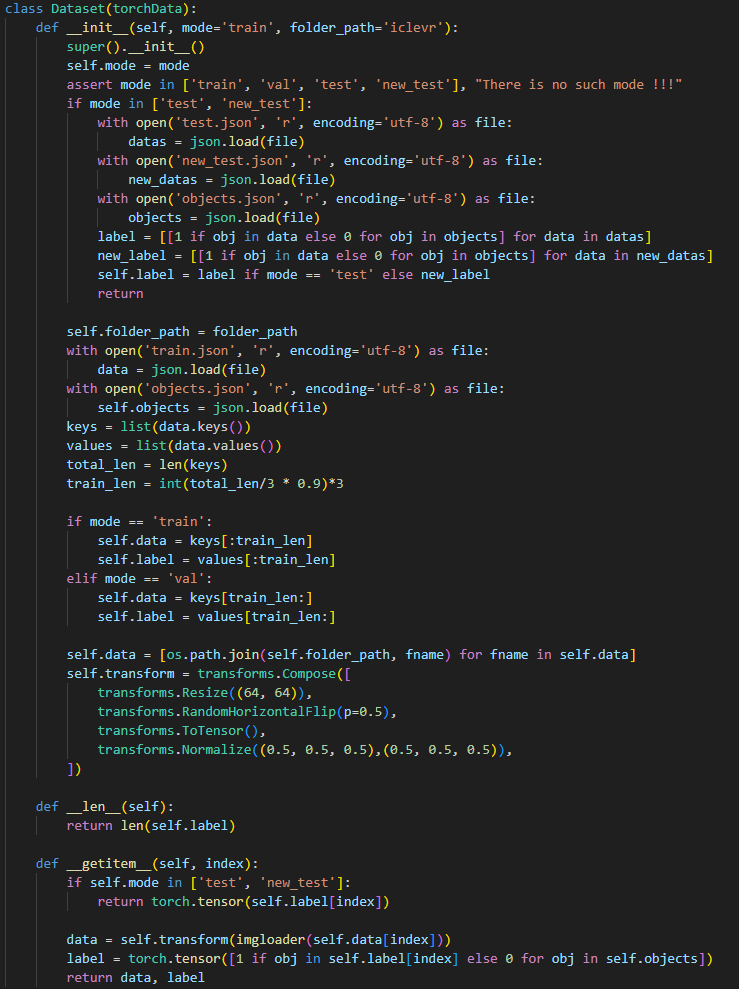
在本實驗中聚焦在DDPM模型，以及如何將人類提供的標籤融入DDPM中，並利用這些標籤進行去噪和模型訓練。最後以pretrain model的正確率作為模型效能的指標之一。

1. Implementation details
   1. Model



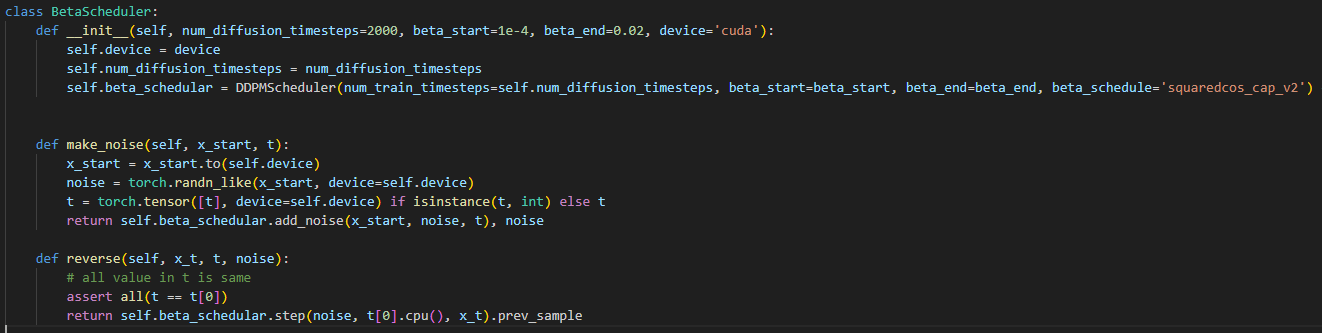
程式碼實現條件化 U-Net 模型。先將label的為度用Linear轉到512，在將其作為模型的輸入放進Unet。模型結構基於Unet2DModel，設置樣本大小為 64x64，輸入通道為 3 加上嵌入維度，輸出通道為 3，而在Unet模型當中，每一個block都是由ResNet所構建的，除此之外，整體架構上Unet的最底下2層採用的是Attention block，讓模型在底層能夠考慮所有訊息的相互關係。Forward的時候會先將一個multi-hot的label透過Linear轉到512為度，並且將其接在input x之後和t一起丟入模型。

* 1. DataSet, DataLoader



DataSet紀錄該資料集的所有圖片檔名和標籤，並將train、validation的資料9:1拆分，而在做隨機存取的時候，會先讀取照片並做隨機翻轉以提升資料多樣性，標籤的表示方式為multi-hot，代表對於所有24個類別，資料有包含該類別時候該index裡的值為1反之為0。

* 1. BetaSchedular

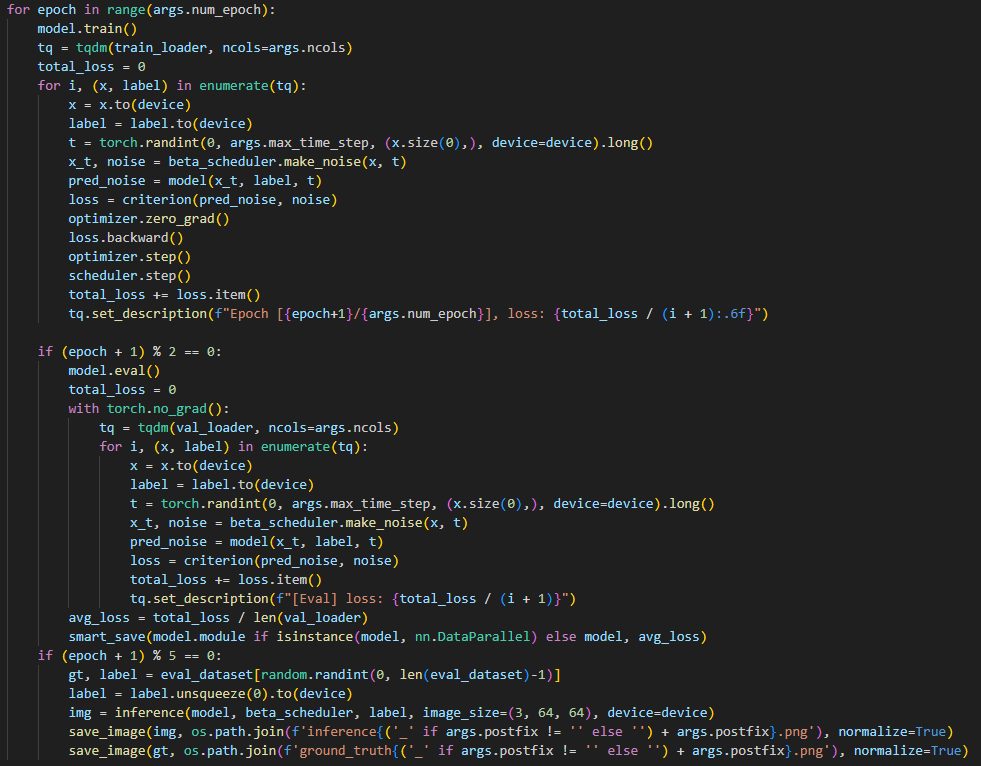


BetaScheduler類別用於管理diffusion模型中的雜訊參數，負責產生雜訊並去噪。採用DDPMScheduler框架。make\_noise吃一個資料x\_start(x\_0)和時間t，產生與隨機雜訊noise，然後透過DDPMScheduler內建函式add\_noise將雜訊加入x\_start，傳回新增雜訊後的資料和原始雜訊，其add\_noise遵循著以下公式產生雜訊圖片，reverse也同樣採用其內建函式實作。

* 1. Training Strategy

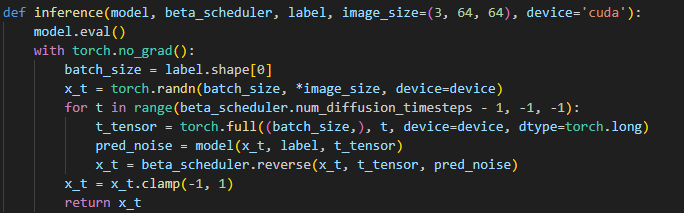


本次實驗在正式training之前，會先將資料平均分布在所有的GPU上面以加快訓練速度，其他細項如loss採用MSE，schedular採用diffusers的內建cos loss。



而在training的過程中使用tqdm，首先會先隨機產生時間t，並且透過BetaSchedular中的make\_noise得到加上雜訊的圖片以及其雜訊。接著將雜訊圖片、標籤、時間t丟進模型當中預測雜訊noise，最後透過MSE計算loss。而每2個epoch則做一次validation，並且每次儲存validation loss最小的那個模型。每5個epoch則做一次inference可視化，將validation data的資料隨機抽取一個做inference得出照片並記錄起來。

* 1. Inference



Inference函數作為生成圖片的函式，其假設圖片加上許多次的雜訊後，其樣態接近常態分佈，則一開始設x\_t為一個常態分佈的圖像，透過模型預測的雜訊丟給BetaSchedular做reverse，隨著t\_tensor的值越來越小，x\_t會越來越接近label標籤所表示的圖片，而在最終將圖片數值固定在-1~1之間並回傳生成後的圖片。

* 1. Test



Test將test.json和new\_test.json中的標籤透過inference生成圖片並儲存在images資料夾中。

1. Results and discussion