CAVEDU教育團隊部落格

[ 技術教學文-格式 ]

|  |  |  |  |
| --- | --- | --- | --- |
| **標題** | 嘗試讓電腦玩玩填色遊戲 | | |
|  | | | |
| **分類** |  | **標籤** |  |
| **撰寫/攝影** |  | | |
| **前情提要** | 文字＋文章連結........ | | |
| **時間** |  | **材料表** |  |
| **成本** |  |
| **難度** |  |
| **前言** | | | |
| 上次已經教大家cDCGAN的建構方法並利用cDCGAN來生成特定的手寫數字，而接下來我們將繼續在生成對抗的汪洋大海中游離，充分感受一下這片大海的艱辛 (誤 ) 。之前提到GAN最著名的有兩種應用：圖像生成、風格轉換，既然我們已經做完圖像生成接著就可以來做做看風格轉換拉！這次的目標是要讓電腦自己玩填色遊戲～ | | | |
| **內文** | | | |
| |  |  | | --- | --- | | **作者** | **張嘉鈞** | | **難度** | 中偏難 | | **材料表** | 1. Jetson Nano 2. PC or Notebook |  Pix2pix 說到風格轉換第一個想到的就是 pix2pix，他算是很早期透過 cGAN 的方式去完成圖像與圖像之間的翻譯，是風格轉換的經典作品之一，下圖擷取自該論文，風格轉換有很多種有趣的應用，像是從語意分析圖轉換成實體圖片，又或者將衛星地圖轉換成簡化地圖，也有給輪廓去填顏色，這些都算是風格轉換也算是 pix2pix 提供範例的範疇。   風格轉換 Style Transform 風格轉換的概念其實很簡單，只要將輸入跟輸出改變一下就可以完成風格轉換了。原本圖片生成的部分生成器的輸入是一組雜訊、輸出是圖片，就像下圖一樣：    現在只要把他改成輸入是一個風格的圖片、輸出是另一種風格的圖片，那我們就完成風格轉換了～不過在這邊我們的數據必須是對應的，如下圖來說Ａ風格的數字５跟Ｂ風格的數字５就是對應的：   Pix2pix重點技術簡介  1. 運用cGAN的架構 ( 以訓練鑑別器為例 )   下圖可以看到輸入是A風格而經過生成器後會產生B風格的圖片，這時候A風格跟B風格都要丟進鑑別器，主要用意是在輪廓都是A的狀況下，鑑別器要去區分是真實顏色還是由生成器所生成出來的。     1. 生成器採用 U-Net結構   論文中有提到生成器的架構有兩種，一種是AutoEncoder的架構；另一種是U-Net的架構，作者提出來的論點主要在輸入與輸出的差別只有在「表面外觀」不同，實際的「基礎結構」是相同的，如果常看捲積神經網路的讀者應該知道，捲積神經網路的淺層主要都在色塊而深層特徵比較多是在輪廓等結構上，所以使用U-Net結構可以讓結構的訊息被共享，白話一點就是可以讓輪廓這類型的特徵在神經網路中更突出。     1. 鑑別器使用 PatchGAN的技術   鑑別器運用了PatchGAN的技術，一般的GAN都是在輸出的時候給予一個數值( 0或1 )，但是PatchGAN的技術是給予N\*N的矩陣 (每一個位置也是 0或1)，每一個數值都代表一個Patch，可以想像把圖片切成很多個區塊去判斷成像是否真實，因為判斷的區域較小可以顧及到的細節更多，而相對的整體上細節也就會更好，特別是Patch數量越多的時候。     1. 損失函數加入了L1正規化   從下圖就可以看到cGAN的結果應該還不錯了，但是加上L1約束後細節會更加明顯，主要是L1會讓數值區間更窄，表示讓圖像均值化，當RGB都相近的時候會越接近灰階，而 cGAN則是目標要讓圖片有更多顏色。   實現 pix2pix 畢竟是經典之作，現在github上有提供很多PyTorch去實作pix2pix的程式，我看了幾個覺得這個整合度比較高，他將pix2pix跟CycleGAN整合在一起，此外也寫了PyTorch跟Tensorflow的版本，所以我們就用這個 github來體驗一下pix2pix的威能吧！   1. 下載github   如果沒安裝過git則需先安裝，下載完後檔案結構如下圖：   |  | | --- | | pip install git  git clone <https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix> |      1. 安裝需求套件   接著在檔案目錄中可以找到　requirements.txt，我們只需要執行下列程式進行套件安裝：   |  | | --- | | pip install -r requirements.txt |   套件如下，其中torch跟torchvision就是必備套件，dominate跟visdom是用於可是化的套件，跟tensorboard一樣需要在開啟server才能進行觀察。     1. 下載數據集   根據論文提供的數據集，總共有五種可以下載！今天我們想要讓電腦嘗試去填顏色，所以我們可以選擇 edges2shoes來玩玩看～下載後解壓縮到pytorch-CycleGAN-and-pix2pix/datasets當中：   |  | | --- | | 數據集連結：<https://people.eecs.berkeley.edu/~tinghuiz/projects/pix2pix/datasets/> | |  |   Edges2shoes中有兩個資料夾 ( train, val)，每張圖片大小為 ( 256 , 512 ) ，代表是將輪廓圖與原始圖合併在一起，左邊為輪廓右邊為原始圖，所以如果之後我們要預測自己的圖也必須符合這個形式：     1. 下載欲訓練模型   接下來就要下載欲訓練模型了，一樣提供5種，請找到跟數據集對應的下載：   |  | | --- | | 欲訓練模型載點：<http://efrosgans.eecs.berkeley.edu/pix2pix/models-pytorch/> | |  |   下載完後放置到pytorch-CycleGAN-and-pix2pix\checkpoints\edges2shoes\_pretrained底下，並且更名為 latest\_net\_G.pth，通常需要自己新增資料夾checkpoints、edges2shoes\_pretrained。  這邊可以注意到.pth檔是PyTorch的其中一種模型檔，它包含了神經網路模型以及權重，儲存跟讀取都很方便，缺點就是自由度不高，官方也比較傾向只儲存權重的方式，不過這部分就不是今天探討的範圍了。   1. 進行預測   首先，需要先將數據集中的val 更名為 test，接著執行程式：   |  | | --- | | python test.py --dataroot ./datasets/edges2shoes --name edges2shoes\_pretrained --model pix2pix --direction BtoA |   --dataroot 就是數據集的位置  --name 是checkpoint的資料夾，也就是模型、權重的資料夾名稱  --model 有 cycleGAN與pix2pix可選擇  --direction 是風格轉換方向；要將輪廓填滿還是將填滿的轉成輪廓   1. 執行結果   最後結果將會輸出在pytorch-CycleGAN-and-pix2pix的results當中，有個別的圖檔也有作者整理在網頁上的比較圖，下圖擷取部分html上的結果。可以看到效果還蠻有趣的，大致的輪廓其實掌握得很好，但顏色都會稍微有一點色偏。   玩轉 pix2pix 我在找pix2pix的時候找到一個很厲害的大神，他自己做了貓咪數據集並且放在互動式網頁上<https://affinelayer.com/pixsrv/>，我就在思考自己或許也能做一個陽春版的。    既然有了決斷就只差執行了！我就直接拿預訓練好的edges2shoes來嘗試，要做到這個首先需要做一個手寫繪圖版，這次我使用的是　pyqt5 ，它是 Python 用來撰寫 GUI的套件之一，可以取代內建的TKinter，我個人蠻喜歡它是它有一個Qt Designer可以像 Visual Studio 拉視窗程式那樣處理，相對來說方便許多。  Qt Designer screenshot on Windows  不過今天我們要製作極簡手繪版就不需要這個Qt Designer了，一切從簡～  首先記得沒安裝pyqt5的要先安裝一下   |  | | --- | | pip install PyQt5 |   先在那個github中建立一個新的 Jupyter Notebook，我們可以透過終端機開啟 Notebook，輸入：   |  | | --- | | jupyter notebook |   開啟後，在右上角new的地方新增 python3 檔案即可，這邊我命名為 drawpad。    首先要先定義視窗程式，設定標題、視窗顯示的位置 ( 起始座標x，起始座標y，寬，高)   |  | | --- | | self.setWindowTitle("Paint with PyQt5")  merge = 40  size = canvas\_size - merge  self.setGeometry(merge, merge, size, size) |   接著建立一個全白的圖片用於繪圖   |  | | --- | | self.image = QImage(self.size(), QImage.Format\_RGB32)  self.image.fill(Qt.white) |   繪圖的相關設定，狀態變數、筆刷大小顏色以及滑鼠最後的位置   |  | | --- | | self.drawing = False  self.brushSize = brush\_size  self.brushColor = Qt.black  self.lastPoint = QPoint() |   定義菜單以及對應功能，鑒於觸碰螢幕的菜單有點難按所以我沒增加到菜單中，不過有設定快捷鍵所以使用上還是很方便的，總共有三個功能，儲存、清除、離開：   |  | | --- | | mainMenu = self.menuBar()  fileMenu = mainMenu.addMenu("File")  saveAction = QAction("Save", self)  saveAction.setShortcut(QKeySequence('Ctrl+s'))  fileMenu.addAction(saveAction)  saveAction.triggered.connect(self.save)  clearAction = QAction("Clear", self)  clearAction.setShortcut(QKeySequence('Ctrl+c'))  fileMenu.addAction(clearAction)  clearAction.triggered.connect(self.clear)  exitAction = QAction("Exit", self)  exitAction.setShortcut(QKeySequence('Ctrl+q'))  fileMenu.addAction(exitAction)  exitAction.triggered.connect(self.exitapp) |   接下來就是繪圖的關鍵，pyqt5在觸碰上面好像有另外的寫法，不過這邊我就直接採用滑鼠點擊的方式來寫，所以要先定義滑鼠按下、拖曳、放開三個動作事件。首先是按下的時候，我們要先將狀態設定成True並且紀錄按下的位置：   |  | | --- | | def mousePressEvent(self, event):  if event.button() == Qt.LeftButton:  self.drawing = True  self.lastPoint = event.pos() |   在滑鼠拖曳的時候，要先實例化畫布功能，實現在image上並且設定筆刷參數，接著劃一條線從上一個位置到現在位置，最後更新位置資訊並且刷新畫布：   |  | | --- | | def mouseMoveEvent(self, event):    if (event.buttons() & Qt.LeftButton) & self.drawing:    painter = QPainter(self.image)  painter.setPen(QPen(self.brushColor, self.brushSize,  Qt.SolidLine, Qt.RoundCap, Qt.RoundJoin))    painter.drawLine(self.lastPoint, event.pos())    self.lastPoint = event.pos()  self.update() |   最後放開左鍵的時候就將狀態設為False，也就不會畫線跟刷新了：   |  | | --- | | def mouseReleaseEvent(self, event):    if event.button() == Qt.LeftButton:  self.drawing = False |   各種功能設定，第一個是畫圖的事件，需要先將畫圖功能打開，對象是主視窗，利用drawImage將剛剛的self.image繪製上去；第二跟第三個是為了綁定菜單跟快捷鍵而設計的副函式：   |  | | --- | | # 建立 painter 後還須建立畫圖的事件  def paintEvent(self, event):  canvasPainter = QPainter(self)  canvasPainter.drawImage(self.rect(), self.image, self.image.rect())    # 清除畫布  def clear(self):    self.image.fill(Qt.white)  self.update()    # 離開視窗  def exitapp(self):  self.close() |   接下來是儲存圖片的部分，儲存圖片的大小我是預設為畫布大小所以會隨著你的螢幕大小而改變，但是訓練、測試資料維度大小為 (256, 256) 所以必須先重新朔形，此外還需增加Ground Truth的部分，不過因為是自己畫得所以Ground Truth為空白，上述說的全都再initImage 這個副函式中執行，接著再儲存的時候先進行儲存 (self.image.save) 再進行前處理 ( iniImage)：   |  | | --- | | def initImage(self, filepath):  trg\_size = 256  new\_img = np.zeros([256,512,3], dtype=np.uint8)  new\_img.fill(255)  org\_img = cv2.imread('{}'.format(filepath))  src\_img = cv2.resize(org\_img, (trg\_size, trg\_size) )  print('org\_img {} > src\_img {}'.format(org\_img.shape, src\_img.shape))  for h in range(trg\_size):  for w in range(trg\_size\*2):  if w< 256:  new\_img[h][w] = src\_img[h][w]  else:  break  cv2.imwrite(filepath, new\_img)  # 儲存畫布  def save(self):  filePath = {your path}  self.image.save(filePath)  self.initImage(filePath)  print('save image: ', filePath)  self.close() |   以上宣告完，就是主要執行的階段了：   |  | | --- | | # 實例化 app 這支程式  if not QtWidgets.QApplication.instance():  app = QtWidgets.QApplication(sys.argv)  else:  app = QtWidgets.QApplication.instance()  # 獲得螢幕可使用範圍  screen = app.primaryScreen()  rect = screen.availableGeometry()  print('Available: %d x %d' % (rect.width(), rect.height()))  # 宣告畫布大小以及筆刷大小  canvas\_size = rect.width() if rect.width() < rect.height() else rect.height()  brush\_size = 10  # 實例化視窗  window = Window(canvas\_size, brush\_size)    # 顯示視窗  window.show()    # 執行 app  sys.exit(app.exec()) |   接著下一個block的目標就是執行pix2pix並且查看成果，這邊直接寫了指令 (有點懶得整合)，執行完成後再透過opencv來開啟成果圖，按任意鍵即可退出。   |  | | --- | | !python test.py --dataroot datasets\edges2shoes --model pix2pix --name edges2shoes\_pretrained    ImPath = {your path} + ‘custom\_edge\_fake\_B.png’  im = cv2.imread(ImPath)  cv2.imshow('result', im)  cv2.waitKey(0)  cv2.destroyAllWindows() |  玩轉pix2pix的執行成果 我有嘗試用PC以及Jetson Nano都可以成功運行，當程式開始執行會跳出一個手繪板，  畫完之後按 ctrl + s 就會自動儲存到 test，接下來可以執行下一個 block的程式來利用 pix2pix 生成圖片。    這邊也提供了 PC 、Jetson Nano的DEMO影片：     結語 這次帶大家認識了pix2pix，是不是很好玩？GAN很多應用都是相當有趣的～不過就是訓練起來要人命，所以可以先嘗試別人做好的預訓練模型，來看能不能完成自己想要的結果。像是今天這個案例，如果我要做一個狗狗自動填色的其實就是在datasets中換成自己的數據就可以了。 下一次將帶大家認識另個經典之作 CycleGAN，它與pix2pix都是屬於風格轉換，不過它的數據是可以不用成對的，下一篇會再仔細說明～ | | | |
| **相關文章** | | | |
| AI修圖！pix2pix網路介紹　 <https://www.itread01.com/content/1544921642.html>  圖像翻譯——pix2pix模型　 <https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix>  CycleGAN and pix2pix in PyTorch　 <https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%AA%9E%E8%A8%80/662259/>  Python GUI How to Create Paint Application in PyQt5　<https://codeloop.org/python-gui-how-to-create-paint-application-in-pyqt5/> | | | |