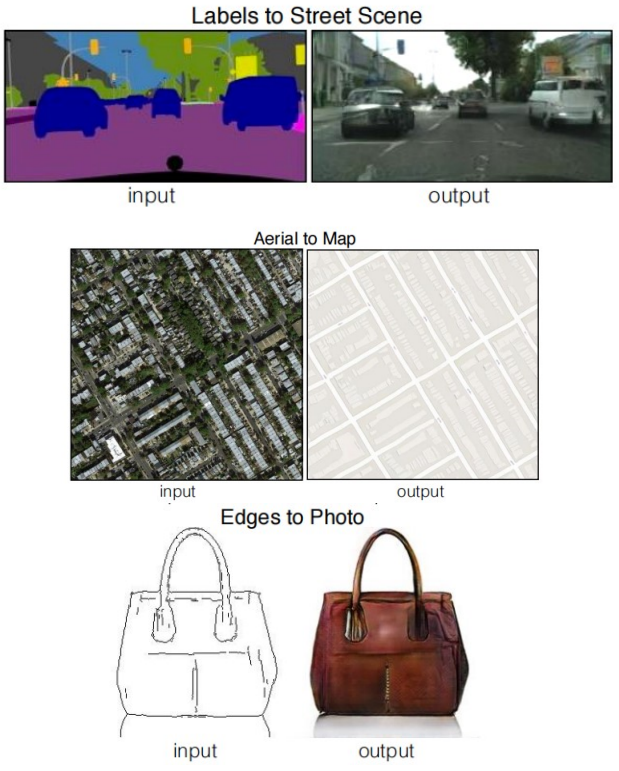
DỊCH HÌNH ẢNH QUA HÌNH ẢNH VỚI CGAN

1st **Thái Thị Hiền - 19527801**
Lớp: DHKHDL15A
Trường đại học Công Nghiệp thành phố Hồ Chí Minh
hienthai1234thptdh@gmail.com

2nd **Trịnh Ngọc Đức - 19469091**
Lớp DHKHDL15A
Trường đại học Công Nghiệp thành phố Hồ Chí Minh
trinhngocduc2000@gmail.com

I. Abstract

***Tóm tắt nội dung*—Chúng em tìm hiểu các mạng GAN có điều kiện như một giải pháp có mục đích chung cho các vấn đề dịch từ hình ảnh sang hình ảnh. Các mạng này không chỉ học cách ánh xạ từ ảnh đầu vào sang hình ảnh đầu ra mà còn học một hàm mất mát để huấn luyện ánh xạ này. Điều này giúp có thể áp dụng cùng một cách tiếp cận chung cho các vấn đề mà theo truyền thống sẽ yêu cầu các công thức tổn thất rất khác nhau. Để đạt được điều này, chúng em sử dụng một bộ phân biệt Markovian, sửa đổi hàm mất mát và một quá trình huấn luyện điển hình hơn của một GAN có điều kiện (cGAN). Bộ dữ liệu được sử dụng trong bài của nhóm em là bộ dữ liệu pix2pix-dataset được lấy từ Kaggle. Trong bộ dữ liệu này nhóm em đã sử dụng 3 bộ dữ liệu nhỏ gồm: Cityscapes, Edges2shoes, Maps để đào tạo mô hình. Qua bài toán này nhóm chúng em thu được kết quả đào tạo mô hình là 99%**



Hình 1: Nhiều vấn đề trong xử lý hình ảnh, đồ họa và tầm nhìn liên quan đến việc dịch một hình ảnh đầu vào thành một hình ảnh đầu ra tương ứng.

II. Giới thiệu

Nhiều vấn đề trong xử lý hình ảnh, đồ họa máy tính và thị giác máy tính có thể được coi là "chuyển đổi" một hình ảnh đầu vào thành một hình ảnh đầu ra tương ứng. Giống như một khái niệm có thể được thể hiện bằng tiếng Anh hoặc tiếng Pháp, một cảnh có thể được hiển thị dưới dạng hình ảnh RGB, trường gradient, bản đồ cạnh, bản đồ nhãn ngữ nghĩa, ... Tương tự như dịch ngôn ngữ tự động, chúng em định nghĩa hình ảnh tự động dịch sang hình ảnh là nhiệm vụ dịch một biểu diễn có thể có của một cảnh sang một cảnh khác, được cung cấp đủ dữ liệu huấn luyện (xem Hình 1). Theo cách truyền thống, mỗi nhiệm vụ này đã được giải quyết bằng các máy chuyên dụng, riêng biệt (ví dụ: [16, 25, 20, 9, 11, 53, 33, 39, 18, 58, 62]), mặc dù thực tế là cài đặt luôn giống nhau: dự đoán pixel từ pixel. Mục tiêu của chúng em trong bài báo này là phát triển một khuôn khổ chung cho tất cả những vấn đề này.

Nhiều người đã thực hiện các bước quan trọng theo hướng này, với các mạng thần kinh tích chập (CNN) trở thành công cụ phổ biến đằng sau nhiều vấn đề dự đoán hình ảnh. CNN học cách giảm thiểu hàm mất mát – một mục tiêu đánh giá chất lượng của kết quả – và mặc dù quá trình học là tự động, vẫn còn rất nhiều nỗ lực thủ công đi vào thiết kế hiệu quả tổn thất. Nói cách khác, vẫn phải cải tiến CNN những gì muốn giảm thiểu. Nhưng, giống như King Midas, chúng ta phải cẩn thận với những gì mình muốn! Nếu chúng ta thực hiện một cách tiếp cận không tốt và yêu cầu CNN giảm thiểu khoảng cách Euclide giữa các điểm ảnh được dự đoán và sự thật cơ bản, nó sẽ có xu hướng tạo ra kết quả mờ nhạt [43, 62]. Điều này là do khoảng cách Euclide được giảm thiểu bằng cách lấy trung bình tất cả các đầu ra hợp lý, điều này gây ra hiện tượng mờ. Tìm ra các hàm mất mát buộc CNN phải làm những gì chúng ta thực sự muốn – ví dụ: xuất ra hình ảnh chân thực, sắc nét – là một vấn đề mở và thường đòi hỏi kiến thức chuyên môn

Thay vào đó, sẽ rất đáng mong đợi nếu chúng ta có thể chỉ xác định một mục tiêu cấp cao, chẳng hạn như "làm cho đầu ra không thể phân biệt được với thực tế", và sau đó tự động học một hàm mất mát phù hợp để đáp ứng mục tiêu này. May mắn thay, đây chính xác là những gì được thực hiện bởi Mạng GAN được đề xuất gần đây [24, 13, 44, 52, 63]. GAN học một tổn thất cố gắng phân loại xem hình ảnh đầu ra là thật hay giả, đồng thời đào tạo một mô hình tổng quát để giảm thiểu tổn thất này. Hình ảnh mờ sẽ không được chấp nhận vì chúng

trông rõ ràng là giả. Bởi vì GAN học một mất mát thích ứng với dữ liệu, chúng có thể được áp dụng cho vô số tác vụ mà theo truyền thống sẽ yêu cầu các loại hàm mất mát rất khác nhau.

Trong bài báo này, chúng em khám phá các GAN có điều kiện. Giống như GAN học một mô hình tổng quát của dữ liệu, các GAN có điều kiện (cGAN) học một mô hình tổng quát có điều kiện [24]. Điều này làm cho các cGAN phù hợp với các tác vụ dịch từ hình ảnh sang hình ảnh, trong đó chúng em dựa vào hình ảnh đầu vào và tạo ra hình ảnh đầu ra tương ứng.

GAN đã được nghiên cứu mạnh mẽ trong mấy năm qua và nhiều kỹ thuật chúng em khám phá trong bài báo này đã được đề xuất trước đó. Tuy nhiên, các bài báo trước đây đã tập trung vào các ứng dụng cụ thể và vẫn chưa rõ các GAN có điều kiện hình ảnh hiệu quả như thế nào có thể là một giải pháp đa năng để dịch từ hình ảnh sang hình ảnh. Đóng góp chính của chúng em là chứng minh rằng trong nhiều vấn đề khác nhau, GAN có điều kiện tạo ra kết quả hợp lý. Đóng góp thứ hai của chúng em là trình bày một khung đơn giản đủ để đạt được kết quả tốt, và để phân tích ảnh hưởng của một số lựa chọn kiến trúc quan trọng.

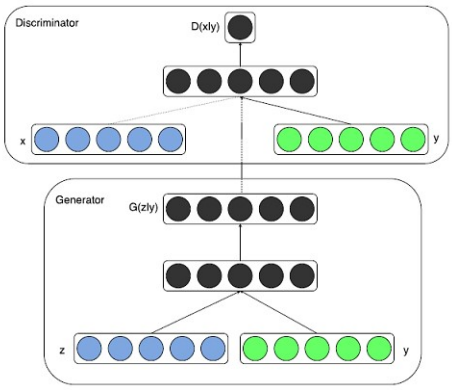
III. Các nghiên cứu liên quan

A. Tổn thất có cấu trúc cho mô hình hóa hình ảnh

Các vấn đề dịch từ hình ảnh sang hình ảnh thường được xây dựng dưới dạng hồi quy hoặc phân loại theo pixel (ví dụ: [39, 58, 28, 35, 62]). Các công thức này coi không gian đầu ra là "không có cấu trúc" theo nghĩa là mỗi pixel đầu ra được coi là có điều kiện phụ thuộc vào tất cả các pixel khác cho hình ảnh đầu vào. Thay vào đó, các GAN có điều kiện học một sự mất mát có cấu trúc. Nhiều tài liệu đã xem xét các tổn thất thuộc loại này, với các phương pháp bao gồm các trường ngẫu nhiên có điều kiện [10], số liệu SSIM [56], đối sánh đặc trưng [15], tổn thất không theo tham số [37], giả tích chập trước [57], và tổn thất dựa trên thống kê hiệp phương sai phù hợp [30]. GAN có điều kiện khác ở chỗ tổn thất được học và về lý thuyết có thể xử lý bất kỳ cấu trúc khả thi nào khác nhau giữa đầu ra và mục tiêu.

B. GAN có điều kiện

Chúng em không phải là người đầu tiên áp dụng GAN có điều kiện. Các tác phẩm trước đó và đồng thời có các GAN có điều kiện trên các nhãn riêng biệt [41, 23, 13], văn bản [46] và thực tế là hình ảnh. Các mô hình có điều kiện về hình ảnh đã xử lý dự đoán hình ảnh từ bản đồ thông thường [55], dự đoán khung hình trong tương lai [40], tạo ảnh sản phẩm [59] và tạo hình ảnh từ các chú thích thưa thớt [31, 48] (ở [47] cho tự hồi quy cách tiếp cận vấn đề tương tự). Một số bài báo khác cũng đã sử dụng GAN để ánh xạ hình ảnh tới hình ảnh, nhưng chỉ áp dụng GAN một cách vô điều kiện, dựa trên các thuật ngữ khác (chẳng hạn như hồi quy L2) để buộc đầu ra phải được điều chỉnh dựa trên đầu vào. Những bài báo này đã đạt được kết quả ấn tượng [43], dự đoán trạng thái tương lai [64], thao tác hình ảnh được hướng dẫn bởi các ràng buộc của người dùng [65], chuyển kiểu [38] và siêu phân giải [36]. Mỗi phương pháp được điều chỉnh cho một ứng dụng cụ thể. Khung của chúng em khác ở chỗ không có ứng dụng cụ thể nào. Điều này làm cho thiết lập của chúng em đơn giản hơn đáng kể so với hầu hết những người khác.

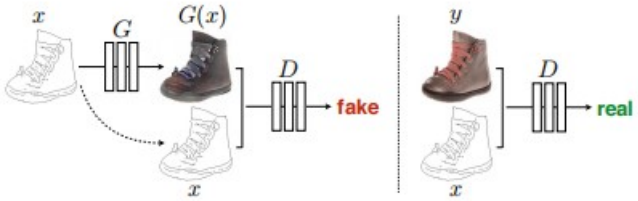


Hình 2: Kiến trúc mạng cGan

Phương pháp của chúng em cũng khác với các bài báo trước đó trong một số lựa chọn kiến trúc cho trình tạo và bộ phân biệt. Không giống như công việc trước đây, đối với trình tạo của chúng em, chúng em sử dụng kiến trúc dựa trên "U-Net" [50] và đối với bộ phân biệt, chúng em sử dụng trình phân loại "PatchGAN" phức tạp, chỉ xử lý cấu trúc ở quy mô các bản vá hình ảnh. Một kiến trúc PatchGAN tương tự đã được đề xuất trước đây trong [38] để thu thập số liệu thống kê kiểu cục bộ. Ở đây chúng em chỉ ra rằng phương pháp này có hiệu quả đối với nhiều vấn đề hơn và chúng em tìm hiểu tác động của việc thay đổi kích thước bản vá.

IV. Phương pháp

GAN là các mô hình tổng quát học cách ánh xạ từ vectơ nhiễu ngẫu nhiên z đến hình ảnh đầu ra y, $G : z \hookrightarrow y$ [24]. Ngược lại, các GAN có điều kiện học cách ánh xạ từ hình ảnh được quan sát x và vectơ nhiễu ngẫu nhiên z, tới y, G : x, z $\rightarrow$ y. Trình tạo G được đào tạo để tạo ra các đầu ra không thể phân biệt được với hình ảnh "thực" bởi bộ phân biệt được đào tạo đối nghịch, D, được đào tạo để làm tốt nhất có thể trong việc phát hiện "hàng giả" của trình tạo. Quy trình đào tạo này được sơ đồ hóa trong Hình 3.



Hình 3: Huấn luyện GAN có điều kiện để ánh xạ các cạnh $\rightarrow$ ảnh.

A. Phương pháp khách quan

Hàm mục tiêu của Gan:

$$L_{GAN}(G, D) = E_x[logD(x)] + E_z[log(1 - D(G(x))$$

Trình tạo sẽ cố gắng giảm thiểu hàm trên trong khi bộ phân biệt cố gắng tối đa hoá hàm trên:

Hàm mục tiêu của GAN có điều kiện có thể được biểu thị bằng:

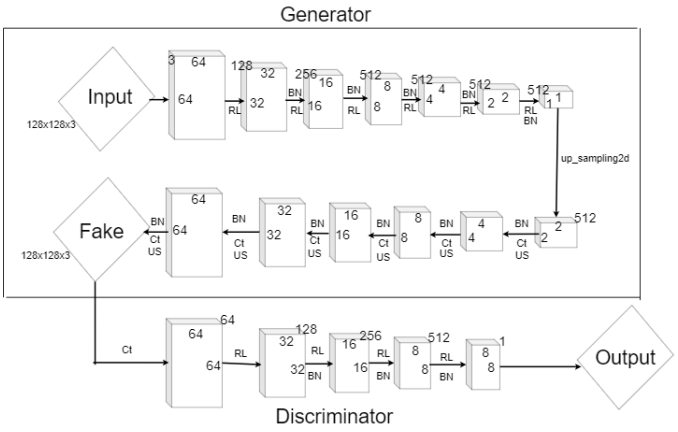
$$L_{cGAN}(G, D) = E_{x,y}[logD(x, y)] + E_{x,z}[log(1 - D(G(x, z))$$

trong đó G cố gắng giảm thiểu mục tiêu này, với D ngược lại cố gắng tối đa hóa nó, tức là:

$$G^* = argmin_G max_D L_{cGAN}(G, D)$$

B. Kiến trúc mạng

Chúng em điều chỉnh kiến trúc trình tạo và bộ phân biệt từ những kiến trúc trong [44]. Cả trình tạo và bộ phân biệt đều sử dụng các mô-đun có dạng convolution-BatchNorm-ReLu [29].



1) Trình tạo (Generator): Một đặc điểm xác định của các bài toán dịch từ hình ảnh sang hình ảnh là chúng ánh xạ lưới đầu vào có độ phân giải cao sang lưới đầu ra có độ phân giải cao. Ngoài ra, đối với các vấn đề chúng em xem xét, đầu vào và đầu ra khác nhau về hình thức bên ngoài, nhưng cả hai đều là kết xuất của cùng một cấu trúc cơ bản. Do đó, cấu trúc trong đầu vào gần giống với cấu trúc trong đầu ra. Chúng em thiết kế kiến trúc trình tạo xung quanh những cân nhắc này cụ thể chúng em đã sử dụng mạng U-net

Chúng em đã sử dụng cùng một kiến trúc cho trình tạo nhưng với kích thước bộ lọc là 4 , không sử dụng max-pooling và 7 lớp upsampling và 7 lớp downsampling.

- Trình tạo bỏ qua

Một đặc điểm xác định của các bài toán dịch từ hình ảnh sang hình ảnh là chúng ánh xạ lưới đầu vào có độ phân giải cao sang lưới đầu ra có độ phân giải cao. Ngoài ra, đối với các vấn đề chúng em xem xét, đầu vào và đầu ra khác nhau về hình thức bên ngoài, nhưng cả hai đều là kết xuất của cùng một cấu trúc cơ bản. Do đó, cấu trúc trong đầu vào gần giống với cấu trúc trong đầu ra.

Nhiều giải pháp trước đây [43, 55, 30, 64, 59] cho các vấn đề trong lĩnh vực này đã sử dụng mạng encoder - decoder [26]. Trong một mạng như vậy, đầu vào được chuyển qua một loạt các lớp dần dần lấy mẫu xuống, cho đến khi một lớp bottleneck, tại đó quá trình được đảo ngược. Một mạng như vậy yêu cầu tất cả luồng thông tin đi qua tất cả các lớp, bao gồm cả nút bottleneck. Đối với nhiều vấn đề về dịch thuật hình ảnh, có rất nhiều thông tin cấp thấp được chia sẻ giữa đầu vào và đầu ra, và mong muốn đưa thông tin này trực tiếp qua mạng. Ví dụ, trong trường hợp tô màu hình ảnh, đầu vào và đầu ra chia sẻ vị trí của các cạnh nổi bật.

Để cung cấp cho trình tạo một phương tiện để phá vỡ nút thắt bottleneck đối với thông tin như thế này, chúng em thêm các kết nối bỏ qua, theo hình dạng chung của "U-Net" [50]. Cụ thể, chúng em thêm các kết nối bỏ qua giữa mỗi lớp i và lớp $n - i$, trong đó n là tổng số lớp. Mỗi kết nối bỏ qua chỉ đơn giản là nối tất cả các kênh ở lớp i với các kênh ở lớp $n - i$.

2) Bộ phân biệt (Discriminator): Đối với bộ phân biệt, chúng em đã sử dụng bộ phân loại PatchGAN tích chập, chỉ xử phạt cấu trúc ở quy mô của các bản vá hình ảnh. Bộ phân biệt này cố gắng phân loại xem mỗi bản vá N X N trong một hình ảnh là thật hay giả. Bộ phân biệt này chạy phức tạp trên hình ảnh, lấy trung bình tất cả phản hồi để cung cấp đầu ra cuối cùng của D.

- Bộ phân biệt Markovian(PatchGAN)

Ai cũng biết rằng sự mất mát L2 – và L1, xem Hình 4 – tạo ra kết quả mờ đối với các vấn đề về tạo ảnh [34]. Mặc dù những mất mát này không khuyến khích tần số sắc nét, trong nhiều trường hợp chúng vẫn nắm bắt chính xác các tần số thấp. Đối với các sự cố trong trường hợp này, chúng em không cần một khung hoàn toàn mới để thực thi tính chính xác ở tần số thấp. L1 sẽ làm rồi.

Điều này thúc đẩy việc hạn chế bộ phân biệt GAN chỉ lập mô hình cấu trúc tần số cao, dựa vào thuật ngữ L1 để buộc tính chính xác của tần số thấp. Để lập mô hình tần số cao, chỉ cần giới hạn sự chú ý của chúng ta vào cấu trúc trong các mảng hình ảnh cục bộ là đủ. Do đó, chúng em thiết kế một kiến trúc phân biệt – mà chúng em gọi là PatchGAN – chỉ xử lý cấu trúc ở quy mô các bản vá. Bộ phân biệt này cố gắng phân loại xem mỗi bản vá NN trong một hình ảnh là thật hay giả. Chúng em chạy đồng minh tích chập của bộ phân biệt này trên hình ảnh, tính trung bình tất cả các phản hồi để cung cấp đầu ra cuối cùng của D.

Trong Phần 4.4, chúng em chứng minh rằng N có thể nhỏ hơn nhiều so với kích thước đầy đủ của hình ảnh mà vẫn tạo ra kết quả chất lượng cao. Điều này thuận lợi vì PatchGAN

nhỏ hơn có ít tham số hơn, chạy nhanh hơn và có thể được áp dụng cho các hình ảnh lớn tùy ý.

Một bộ phân biệt như vậy mô hình hóa hình ảnh một cách hiệu quả dưới dạng trường ngẫu nhiên Markov, giả sử tính độc lập giữa các điểm ảnh cách nhau hơn một đường kính miếng vá. Mối liên hệ này trước đây đã được khám phá trong [38], và cũng là giả định phổ biến trong các mô hình kết cấu [17, 21] và phong cách [16, 25, 22, 37]. Do đó, PatchGAN của chúng em có thể được coi là một dạng mất kết cấu/kiểu dáng.

C. Tối ưu hóa và suy luận

Để em ưu hóa mạng của mình, chúng em tuân theo phương pháp tiếp cận tiêu chuẩn từ [24]: chúng em xen kẽ giữa một bước gradient descent trên D, sau đó một bước trên G. Như đã đề xuất trong bài báo GAN gốc, thay vì đào tạo G để giảm thiểu

$$\log(1 - D(x, G(x, z))$$

chúng em đào tạo để tối đa hóa

$$\log(D(x, G(x, z))$$

. Ngoài ra, chúng em chia mục tiêu cho 2 trong khi tối ưu hóa D, điều này làm chậm tốc độ học của D so với G. Chúng em sử dụng minibatch SGD và áp dụng bộ giải Adam [32], với learning rate là 0,0002 và tham số $\beta 1$ = 0,5, $\beta 2$ = 0,999

Tại thời điểm suy luận, chúng em chạy trình tạo theo cách giống hệt như trong giai đoạn huấn luyện. Điều này khác với giao thức thông thường ở chỗ chúng em áp dụng loại bỏ tại thời điểm kiểm tra và chúng em sử dụng chuẩn hóa hàng loạt [29] bằng cách sử dụng số liệu thống kê của đợt kiểm tra, thay vì số liệu thống kê tổng hợp của đợt huấn luyện. Cách tiếp cận chuẩn hóa batch này, khi batch sizes được đặt thành 1, được gọi là "chuẩn hóa cá thể" và đã được chứng minh là có hiệu quả đối với các tác vụ tạo tuổi ảnh [54]. Trong các thử nghiệm của mình, chúng em sử dụng batch sizes từ 1 đến 10 tùy thuộc vào thử nghiệm.

V. Thực nghiệm

Để khám phá tính tổng quát của GAN có điều kiện, chúng em thử nghiệm phương pháp này trên nhiều tác vụ và bộ dữ liệu khác nhau, bao gồm cả tác vụ đồ họa, như tạo ảnh và tác vụ hình ảnh, như phân đoạn ngữ nghĩa:

Dữ liệu phân vùng ảnh [12].⇔ ảnh, được đào tạo từ bộ dữ liệu Cityscapes

Với tập train là: 2975 tập validate là: 500

Bản đồ ⇔ ảnh chụp từ trên không, được đào tạo từ bộ dữ liệu Map.

Với tập train là: 45825 tập validate là: 200

Ảnh phác thảo → ảnh, được đào tạo từ bộ dữ liệu edges2shoes

Với tập train là: 1096 tập validate là: 1098

Yêu cầu về dữ liệu và tốc độ thì chúng em cảm thấy rằng thường có thể thu được kết quả tốt ngay cả trên các tập dữ liệu nhỏ.

A. Tham số mô hình cgan

epoch = 100

batchsize = 32

learningrate = 0.0002

beta1 = 0.5

beta2 = 0.999

B. Số liệu đánh giá

Đối với mạng Generator chúng em đã sử dụng 2 độ đo MSE và MAE cho hàm loss

Về mạng Discriminator chúng em sử dụng độ do MSE cho hàm loss

MSE được gọi nôm na là giá trị sai số bình phương trung bình hoặc là lỗi bình phương trung bình, nó tính trung bình của bình phương sai số giữa giá trị thực tế và giá trị dự đoán

Nếu $\hat{Y}$ là một vector của n trị dự báo, và Y là vector các trị quan sát được, tương ứng với ngõ vào của hàm số phát ra dự báo, thì MSE của phép dự báo có thể ước lượng theo công thức:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MAE (Mean Absolute Error) là 1 metric đánh giá mô hình bằng cách tính trung bình giá trị tuyệt đối sai số giữa giá trị thực tế và giá trị dự đoán. Công thức MAE được định nghĩa như sau:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i^{'}|$$

Khi xây dựng mô hình phân loại chúng ta sẽ muốn biết một cách khái quát tỷ lệ các trường hợp được dự báo đúng trên tổng số các trường hợp là bao nhiêu. Tỷ lệ đó được gọi là độ chính xác. Độ chính xác giúp ta đánh giá hiệu quả dự báo của mô hình trên một bộ dữ liệu. Độ chính xác càng cao thì mô hình của chúng ta càng chuẩn xác:

$$\frac{TP + TN}{Tong}$$

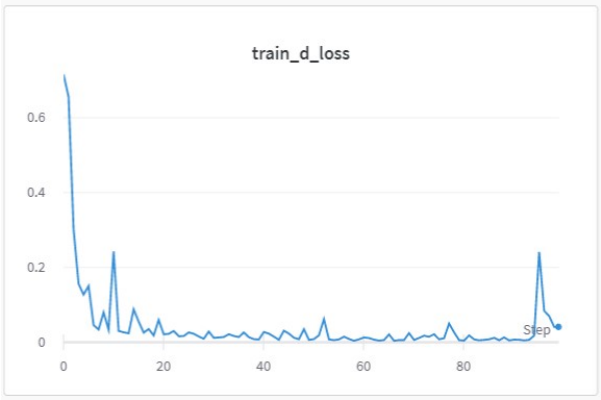
VI. KẾT LUẬN

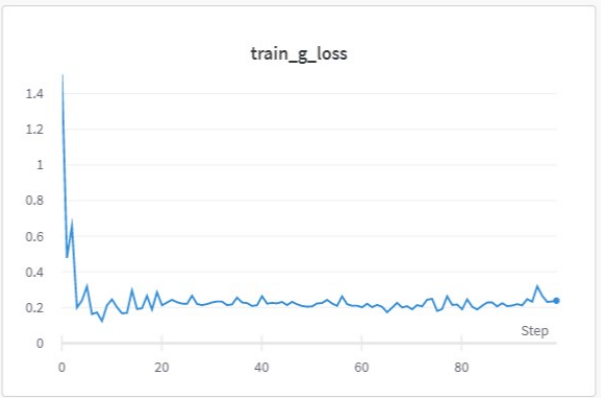
A. Bảng số liệu

Data	D_loss	G_loss	Accuracy
Cityscapes	0.04109	0.23862	98% - 99%
Edges2shoes	0.01404	0.13925	98% - 99%
Maps	0.001749	0.111907	98% - 99%

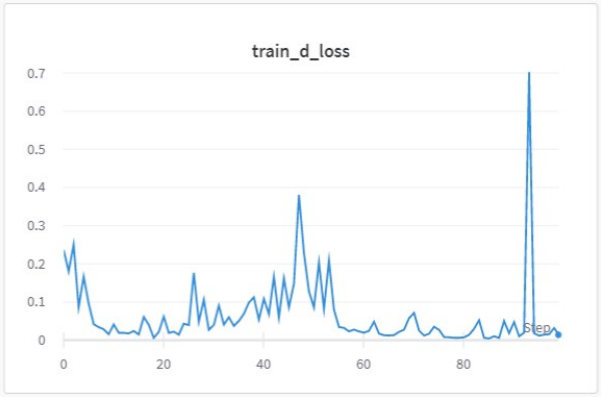
B. Biểu đồ loss

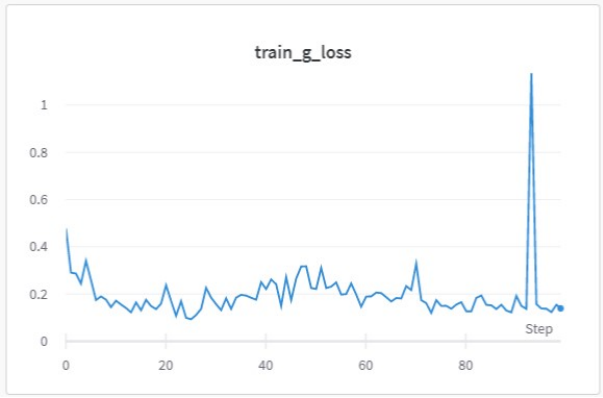
Hình ảnh loss của tập data : Cityscapes



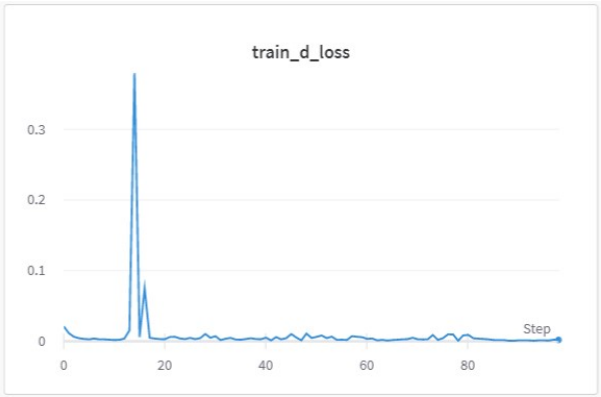


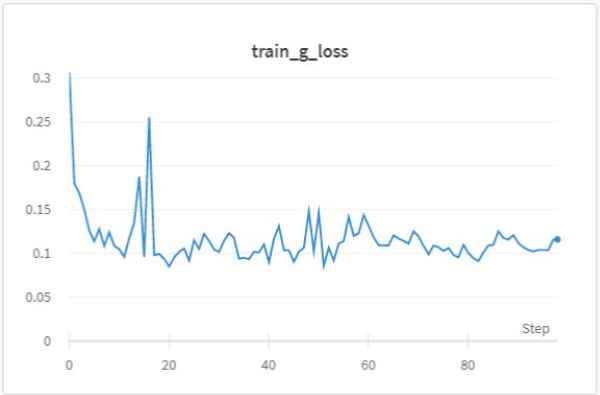
Hình ảnh loss của tập data : Edges2shoes





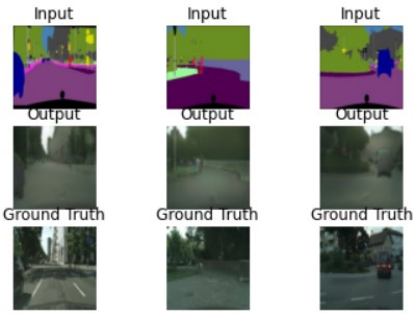
Hình ảnh loss của tập data : Maps

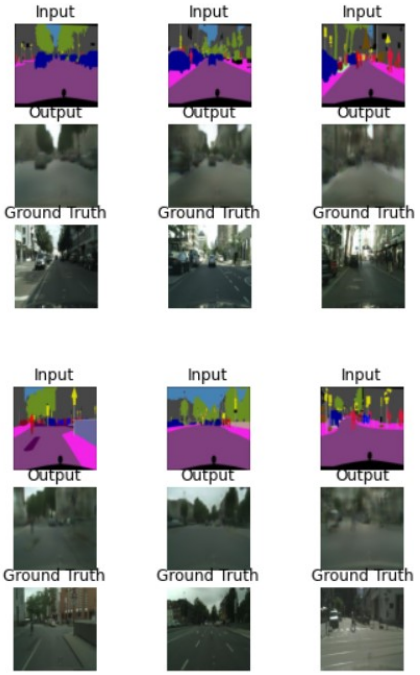




C. Kết quả hình ảnh

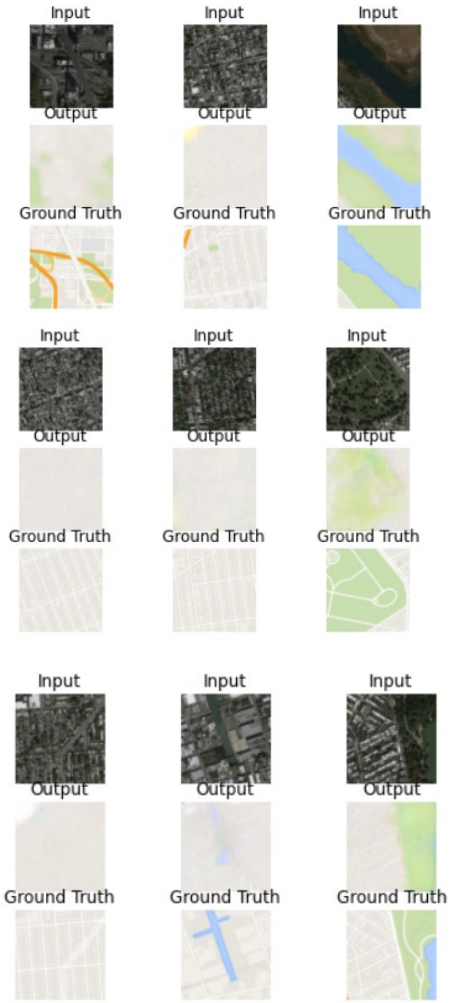
Hình ảnh dự đoán của tập data : Cityscapes





Hình ảnh dự đoán của tập data : Edges2shoes

Hình ảnh dự đoán của tập data : Maps



VII. Hướng phát triển

Với đề tài này nhóm chúng em đã tiến hành tìm hiểu, triển khai mô hình và tìm những biện pháp cải tiến. Bên cạnh đó kết quả đem lại tương đối cao nhưng nhóm chúng em có định hướng sẽ tiếp tục tìm hiểu rộng và sâu hơn về đề tài từ đó có thể đưa ra các mô hình đào tạo khác có kết quả tốt hơn mô hình hiện tại.

VIII. Lời cảm ơn

Đây là bài báo cáo đồ án kết thúc môn học Thị giác máy tính của nhóm chúng em và lời đầu tiên nhóm chúng em muốn cảm ơn các thầy giảng dạy bộ môn thị giác máy tính đã chỉ dạy, hướng dẫn, theo sát nhóm chúng em rất tận tình trong quá trình thực hiện đồ án.Nhóm chúng em muốn gửi lời cảm ơn đến các thầy và nhóm sẽ cố gắng tiếp tục tìm hiểu và phát triển đề tài!

IX. Tài liệu tham khảo

[1] Bertrand gondouin. https://twitter.com/ bgondouin/status/818571935529377792. Accessed, 2017-04-21. 9

[2] Brannon dorsey. https://twitter.com/ brannondorsey/status/806283494041223168. Accessed, 2017-04-21. 9

[3] Christopher hesse. https://affinelayer.com/ pixsrv/. Accessed: 2017-04-21. 9

[4] Gerda bosman, tom kenter, rolf jagerman, and daan gosman. https://dekennisvannu.nl/site/artikel/ Help-ons-kunstmatige-intelligentie-testen/ 9163. Accessed: 2017-08-31. 9

[5] Jack qiao. http://colormind.io/blog/. Accessed: 2017-04-21. 9

[6] Kaihu chen. http://www.terraai.org/ imageops/index.html. Accessed, 2017-04-21. 9

[7] Mario klingemann. https://twitter.com/ quasimondo/status/826065030944870400. Accessed, 2017-04-21. 9

[8] Memo akten. https://vimeo.com/260612034. Accessed, 2018-11-07. 9

[9] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In CVPR, 2005. 1

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In ICLR, 2015. 2

[11] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: internet image montage. ACM Transactions on Graphics (TOG), 28(5):124, 2009. 1

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In CVPR, 2016. 4, 16

[13] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In NIPS, 2015. 2

[14] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros. What makes paris look like paris? ACM Transactions on Graphics, 31(4), 2012. 4, 13, 17

[15] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In NIPS, 2016. 2

[16] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In SIGGRAPH, 2001. 1, 4

[17] A. A. Efros and T. K. Leung. Texture synthesis by nonparametric sampling. In ICCV, 1999. 4

[18] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In ICCV, 2015. 1

[19] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? In SIGGRAPH, 2012. 4, 12

[20] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. ACM Transactions on Graphics (TOG), 25(3):787–794, 2006. 1

[21] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In NIPS, 2015. 4

[22] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. CVPR, 2016. 4

[23] J. Gauthier. Conditional generative adversarial nets for convolutional face generation. Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester, (5):2, 2014. 2

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, 2014. 2, 4, 6, 7

[25] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In SIGGRAPH, 2001. 1, 4

[26] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. Science, 313(5786):504–507, 2006. 3

[27] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In CVPR, 2015. 4, 13, 16

[28] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. ACM Transactions on Graphics (TOG), 35(4), 2016. 2

[29] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015. 3, 4

[30] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In ECCV, 2016. 2, 3

[31] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. arXiv preprint arXiv:1612.00215, 2016. 2

[32] D. Kingma and J. Ba. Adam: A method for stochastic optimization. ICLR, 2015. 4

[33] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. ACM Transactions on Graphics (TOG), 33(4):149, 2014. 1, 4, 16

[34] A. B. L. Larsen, S. K. Sønderby, and O. Winther. Autoencoding beyond pixels using a learned similarity metric.

In ICML, 2016. 3

[35] G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. ECCV, 2016. 2, 8, 16

[36] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In CVPR, 2017. 2

[37] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. CVPR, 2016. 2, 4

[38] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. ECCV, 2016. 2, 4

[39] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015. 1, 2, 5