Tô màu ảnh xám một cách sống động

Richard Zhang, Phillip Isola, Alexei A. Efros

ABSTRACT

- Với 1 bức ảnh trắng đen làm đầu vào, bài báo này sẽ giải quyết vấn đề tưởng tượng ra bản màu hợp lý của bức đó. Đây

là 1 rõ ràng là thiếu thông tin, vì vậy các phương trước đây thường dựa vào các tương tác của người dùng hoặc cho ra

các kết quả màu bị nhạt. Chúng tôi để xuất pp tự động hoàn toàn, có thể tạo ra các màu sống động và chân thực. Sử

dụng kỹ thuật cân bằng lại các lớp trong quá trình huấn luyện để tăng sự đa dạng về màu sắc trong kết quả. Hệ thống

của chúng tôi được triển khai như một quá trình lan truyền xuôi (feed-forward) trong mạng nơ-ron tích chập (CNN) ở

giai đoạn kiểm tra, và được huấn luyện trên hơn 1 triệu ảnh màu. Chúng tôi đánh giá thuật toán bằng cách sử dụng một

=>" bài kiểm tra Turing tô màu"<=. Yêu cầu người tham gia chọn giữa bức ảnh được tạo ra và bức ảnh gốc. Phương pháp

của chúng tôi đánh lừa con người trong 32% số lần thử, cao hơn so với các phương pháp trước. Hơn nữa, Chúng tôi chứng

minh rằng bài toán tô màu có thể trở thành 1 tác vụ =>"tự giám sát" <= mạnh mẽ trong học đặc trưng (feature learning)

hoạt động như một bộ mã hóa giữa các kênh ảnh.

INTRODUCTION

===================> Dán ảnh 1 <=====================

- Hãy xem xét các bức ảnh đen trắng trong Hình 1

- Thoạt nhìn, việc "tưởng tượng" ra màu sắc của chúng tôi có vẻ khăn, vì phần lớn thông tin đã bị mất (2 trong 3 kênh)

- Tuy nhiên, nếu nhìn kỹ hơn, ta nhận ra trong nhiều TH, ngữ nghĩa của cảnh vật và kết cấu bề mặt vẫn cung cấp nhiều

manh mối cho việc tô màu:

+ Như là: Có thường có màu xanh lá, Bầu trời thường xanh dương, Bọ rùa thì có màu đỏ

- Nhứng hiểu biết này dựa trên kiến thức sẵn có về thế giới không phải lúc nào cũng đúng

+ Ví dụ như Cái áo không phải lúc nào cũng màu đỏ vàng tím như ta đoán

- Bài báo này có mục tiêu là không phải là khôi phục chính xác màu gốc thực tế, mà là tạo ra một phiên bản màu hợp

lý, có thể khiến con người tin nó là thật.

- Trong không gian màu CIE Lab, hệ thống chúng tôi sử dụng kênh L ( độ sáng) làm đầu vào, và dự đoán 2 kênh màu a và

b tương ứng để tái tạo màu

- Chúng tôi sử dụng dữ liệu quy mô lớn. Việc dự đoán màu sắc có một đặc điểm thuận lợi là dữ liệu huấn luyền gần như

là miễn phí: bất kỳ ảnh nào cũng có thể huấn luyện, đơn giản bằng cách lấy kênh L của ảnh làm đầu vào và các kênh a

và b làm tín hiệu giám sát. Các nc trước đây cho các kết quả có xu hướng trong bị nhạt nhòa màu.

- [3] Đã được chỉ ra rằng dự đoán màu là một bài toán vốn mang tính đa mô thức - nhiều đối tượng có thể có nhiều cách

tô màu hợp lý khác nhau. Ví dụ, 1 quả táo có thể đỏ vàng xanh, nhưng hiếm khi là màu blue hoặc cam. Để mô hình hóa

lý tính đa mô thức của bài toán, chúng tôi dự đoán một phối các màu khả thi cho mỗi điểm ảnh. Cuối cùng, chúng tôi

tạo ra ảnh màu bằng cách lấy giá trị trung bình lam nguội của phân phối. Kết quả cuối cùng là các ảnh được tô màu

sống động và chân thực hơn so với pp trước đó.

- Chúng tôi đã thiết lập 1 bài kiểm tra Turing tô màu trong đó người tham gia đc đưa cho 2 ảnh gốc và ảnh tô màu bằng

nhân tạo và đc xác định ảnh giả

- Trong thiết lập này, chúng tôi có thể đánh lừa người tham gia trong các trường hợp là 32% trong khi nếu có 2 ảnh

thật thì là 50%. Cao hơn so với các phương pháp trước đây[2]

- Đóng góp của chúng tôi nằm ở 2 lĩnh vực

+ Chúng tôi cải thiện bài toán đồ họa tô màu ảnh tự động bằng cách:

+ Thiết kế 1 hàm mục tiêu phù hợp có thể xử lý sự không chắc chắn đa mô thức và nắm bắt được

sự đa dạng màu sắc rộng hơn

+ giới thiệu 1 khung kiểm thử mới cho các thuật toán tô màu, có thể áp dụng cho các bài toán tổng hợp

khác

+ thiết lập một mốc hiệu suất mới bằng cách huấn luyện trên một triệu ảnh màu.

+ Chúng tôi giới thiệu bài toán tô màu như một phương pháp học biểu diễn tự giám sát đơn giản và cạnh tranh,

+ đạt kết quả hàng đầu trên nhiều bộ đo chuẩn.

- Các nghiên cứu trước về tô màu ảnh

+ các thuật toán tô màu chủ yếu khác nhau ở cách chúng xử lý và khai thác dữ liệu để xây dụng mô hình tương

quan giữa ảnh xám và ảnh màu

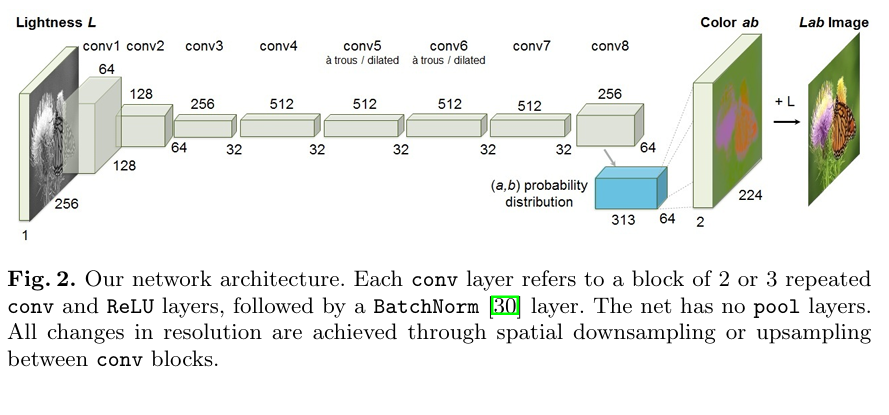
- Phương pháp của chúng tôi học cách phân loại màu, nhưng thực hiện điều đó với một mô hình lớn hơn, được huấn luyện

trên lượng dữ liệu lớn hơn và có nhiều cải tiến hàm mất mát cũng như cách chuyển sang đầu ra liên tục cuối cùng.

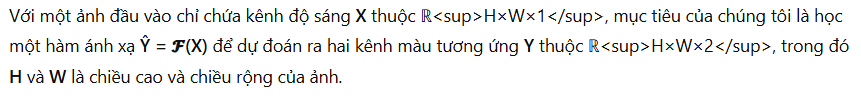
APPROACH PHƯƠNG PHÁP TIẾP CẬN

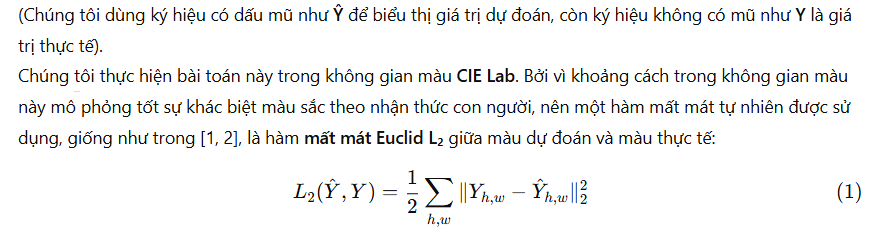
- Chúng tôi huấn luyện mạng nơ ron tích chập để ánh xạ từ ảnh đầu vào dạng thang xám sang một phân phối các giá trị

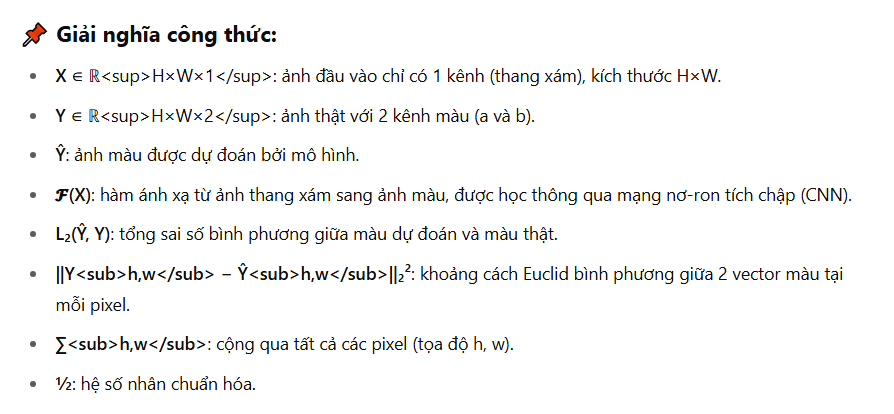
màu đã được lượng tử hóa, sử dụng kiến trúc được minh họa trong



OBJECTIVE FUNCTION HÀM MỤC TIÊU





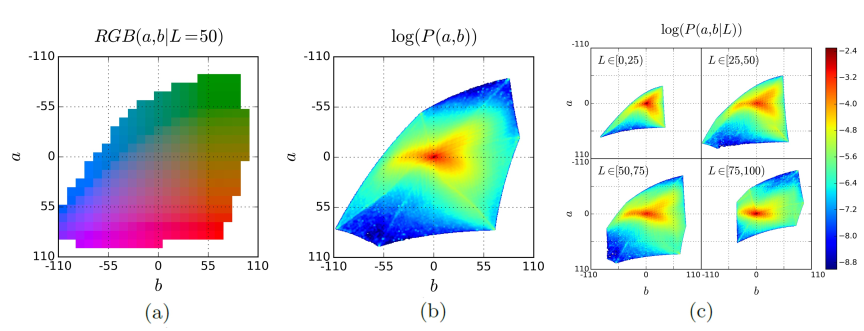


====🡺 Có làm ví dụ bên file ví dụ

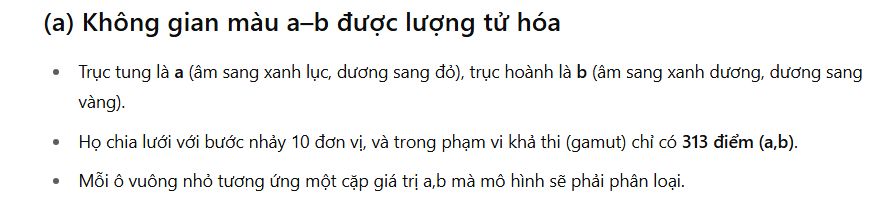
Tuy nhiên, hàm mất mát (loss) này không đủ mạnh để xử lý tính mơ hồ vốn có và bản chất nhiều khả năng (multimodal) của bài toán tô màu. Nếu một vật thể có thể mang nhiều giá trị màu **a** và **b** khác nhau, thì nghiệm tối ưu theo hàm mất mát Euclidean sẽ là **trung bình cộng** của các giá trị đó.

Trong bài toán dự đoán màu, hiệu ứng trung bình này khiến cho kết quả có xu hướng **xám xịt**, thiếu màu sắc sống động.

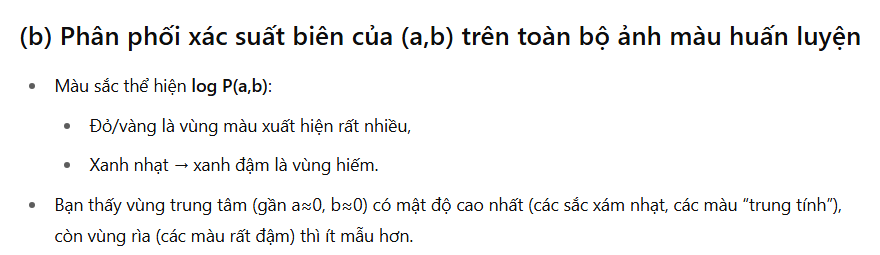
Thêm vào đó, nếu tập hợp các màu khả dĩ là một tập **không lồi** (non-convex), thì nghiệm trung bình còn có thể **không nằm trong tập hợp đó**, dẫn đến kết quả tô màu **không hợp lý**.



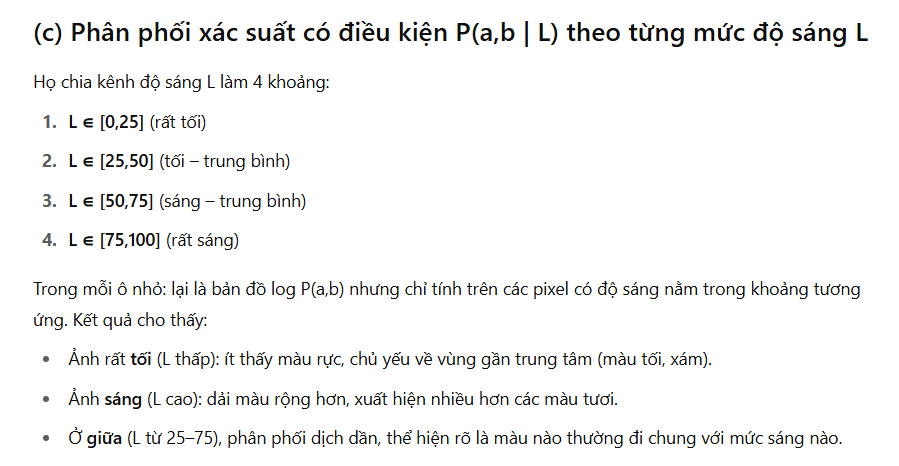
minh họa cách tác giả “phân vùng” và khảo sát phân phối màu trong không gian a–b của CIE Lab:



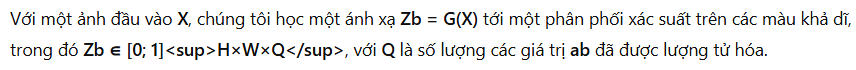
A là biểu diễn tọa độ của ảnh a-b khi sang RGB và giả sử độ sáng cố định là L=50



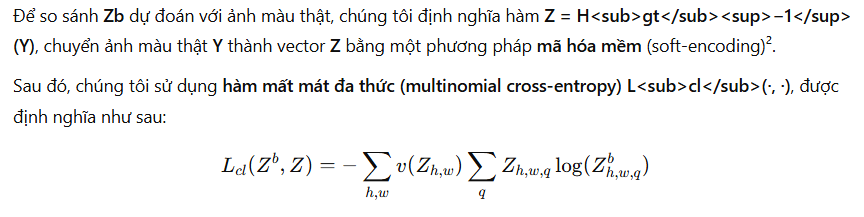
B là thống kế số màu xuất hiện nhiều và hiếm trong thực tế với đỏ vàng là nhứng màu xuất hiện rất nhiều trong tập huấn luyện( như da người, cỏ cây, trời). Xanh đậm là nhứng màu hiếm gặp, ít xuất hiện

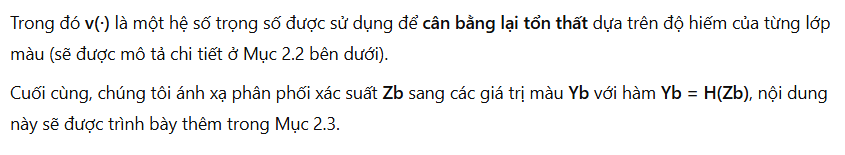


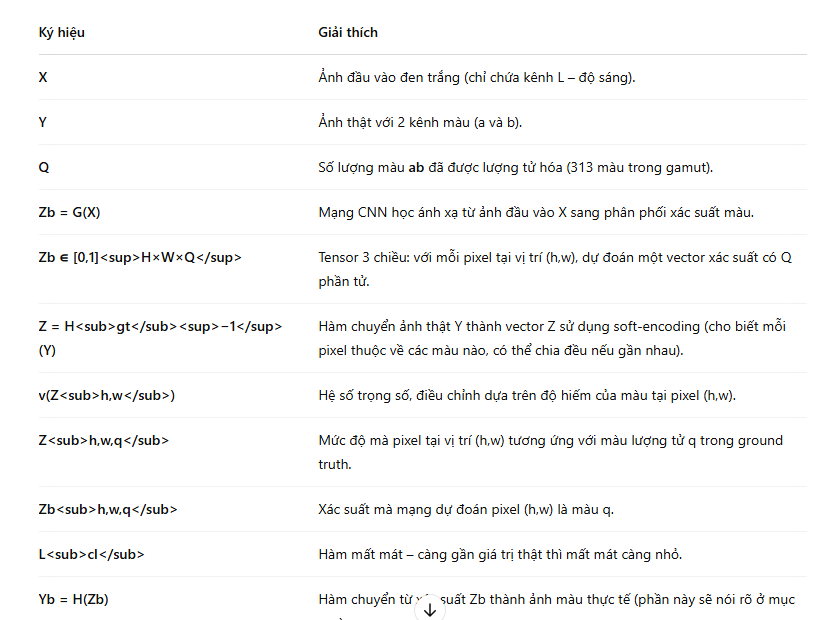
Ở 3C này thì chi biết độ sáng L thuốc [x,y] thì màu nào xuất hiện nhiều. sau khi lượng tử hóa thành 313 màu

* Thay vì vậy, chúng tôi xử lý bài toán như là một bài toán phân loại đa thức.Chúng tôi lượng tử hóa không gian đầu ra **ab** thành các ô vuông (bin) với kích thước lưới là 10 và giữ lại **Q = 313** giá trị nằm trong **gamut màu** (phạm vi màu hợp lệ), như thể hiện trong Hình 3(a).
* 

\*Zb nghĩa là Z có mũ ngang

-

- 



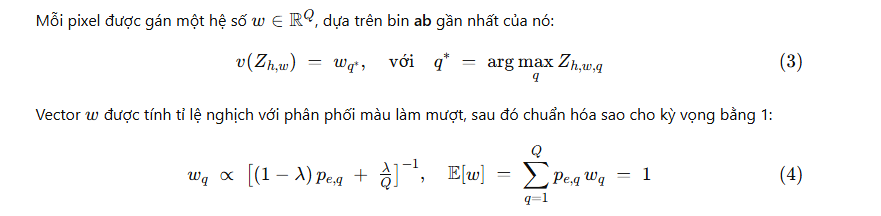
==🡺 Có ví dụ tính toán

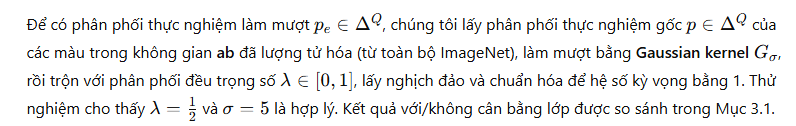
2.2 Cân bằng lại tần suất các lớp (Class rebalancing)

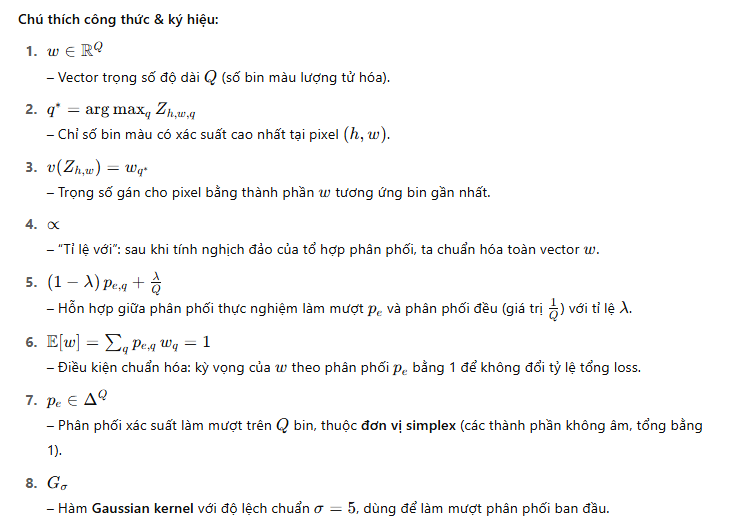
- Phân phối các giá trị **ab** trong ảnh tự nhiên có xu hướng bị lệch mạnh về các giá trị **ab thấp**, là do sự xuất hiện phổ biến của các nền như mây, vỉa hè, đất, và tường. Hình 3(b) cho thấy phân phối thực nghiệm của các pixel trong không gian **ab**, thu thập từ **1.3 triệu** ảnh huấn luyện trong ImageNet [28]. Quan sát cho thấy số lượng pixel ở các giá trị **desaturated** (gần trung tính , màu trung tính là màu nhạt nhòa hơn với màu thực tế như đỏ thiên về 1 chút trắngs) lớn hơn hàng chục đến hàng trăm lần so với các giá trị đậm màu. Nếu không tính đến điều này, các màu nhạt sẽ chiếm ưu thế trong quá trình huấn luyện, khiến kết quả cuối cùng bị nhạt.

**Chú thích 2.** Mỗi giá trị ground truth Yh,wY\_{h,w}Yh,w​ có thể được mã hóa thành vector 1‑hot Zh,wZ\_{h,w}Zh,w​ bằng cách tìm bin ab lượng tử hóa gần nhất. Tuy nhiên, chúng tôi thấy rằng **soft‑encoding** hoạt động rất tốt cho huấn luyện, và cho phép mạng nhanh chóng học được mối liên hệ giữa các phần tử trong không gian đầu ra [31]. Cụ thể, ta tìm **5 láng giềng gần nhất** của Yh,wY\_{h,w}Yh,w​ trong không gian ab lượng tử hóa, và gán trọng số cho mỗi nhãn tỉ lệ thuận với hàm Gaussian của khoảng cách đến ground truth, với độ lệch chuẩn σ=5\sigma = 5σ=5.

Hàm mất mát bị chi phối mạnh bởi các giá trị **ab** nhạt (gần trung tính). Chúng tôi giải quyết vấn đề mất cân bằng lớp bằng cách **tái trọng số** loss của mỗi pixel trong quá trình huấn luyện, dựa trên độ hiếm của màu tại pixel đó. Về mặt tiệm cận, phương pháp này tương đương với cách **resampling** bộ dữ liệu huấn luyện thông thường [32].







\*\* Cái này dùng để giải quyết vấn đề mất bằng dữ liệu màu sắc do các màu trung tính gây ra hay nói cách khác do ab thấp khiến màu nhạt và mất câng bằng

=🡺 Báo cáo không nói