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

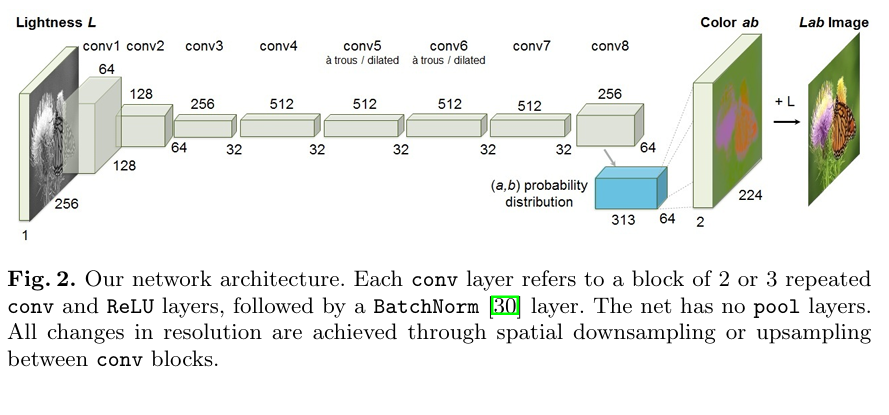
- Phương pháp của chúng tôi học cách phân loại màu, nhưng thực hiện điều đó với một mô hình lớn hơn, được huấn luyện

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.

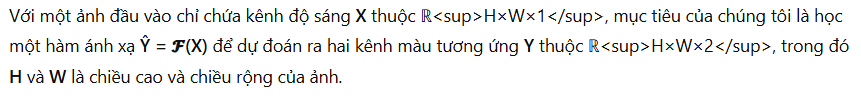
APPROACH PHƯƠNG PHÁP TIẾP CẬN

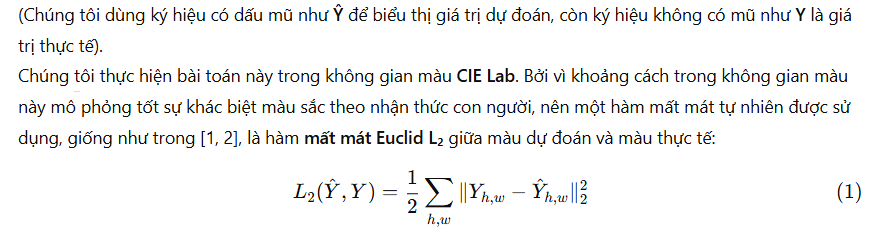
- Chúng tôi huấn luyện mạng nơ ron tích chập để ánh xạ từ ảnh đầu vào dạng thang xám sang một phân phối các giá trị

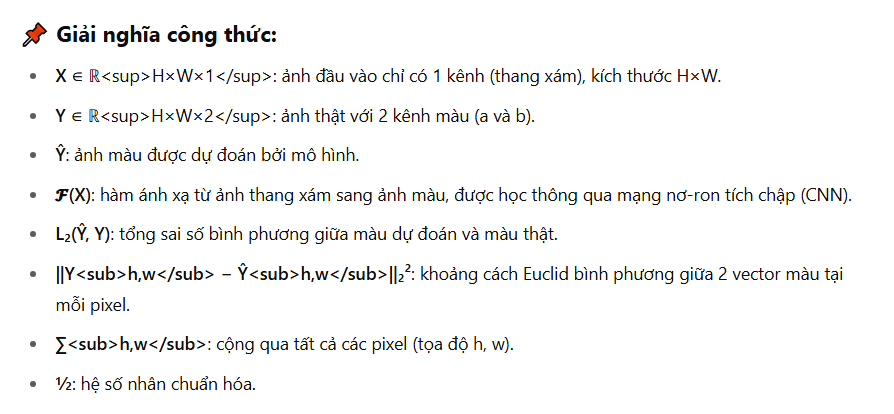
màu đã được lượng tử hóa, sử dụng kiến trúc được minh họa trong



OBJECTIVE FUNCTION HÀM MỤC TIÊU





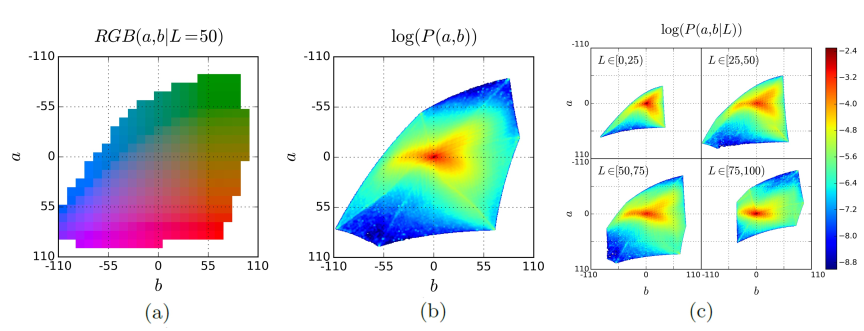


====🡺 Có làm ví dụ bên file ví dụ

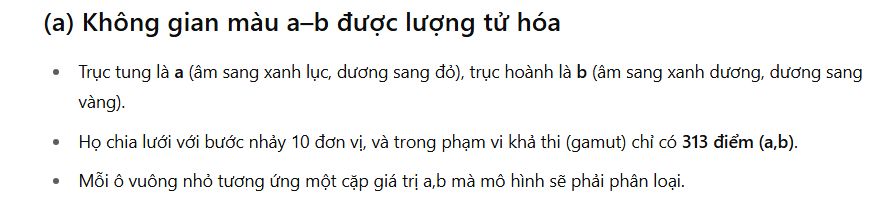
Tuy nhiên, hàm mất mát (loss) này không đủ mạnh để xử lý tính mơ hồ vốn có và bản chất nhiều khả năng (multimodal) của bài toán tô màu. Nếu một vật thể có thể mang nhiều giá trị màu **a** và **b** khác nhau, thì nghiệm tối ưu theo hàm mất mát Euclidean sẽ là **trung bình cộng** của các giá trị đó.

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.

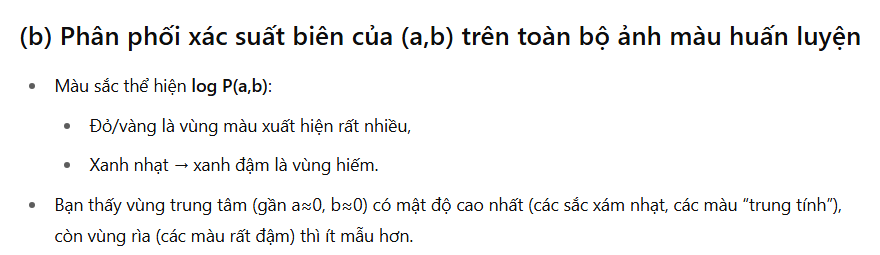
Thêm vào đó, nếu tập hợp các màu khả dĩ là một tập **không lồi** (non-convex), thì nghiệm trung bình còn có thể **không nằm trong tập hợp đó**, dẫn đến kết quả tô màu **không hợp lý**.



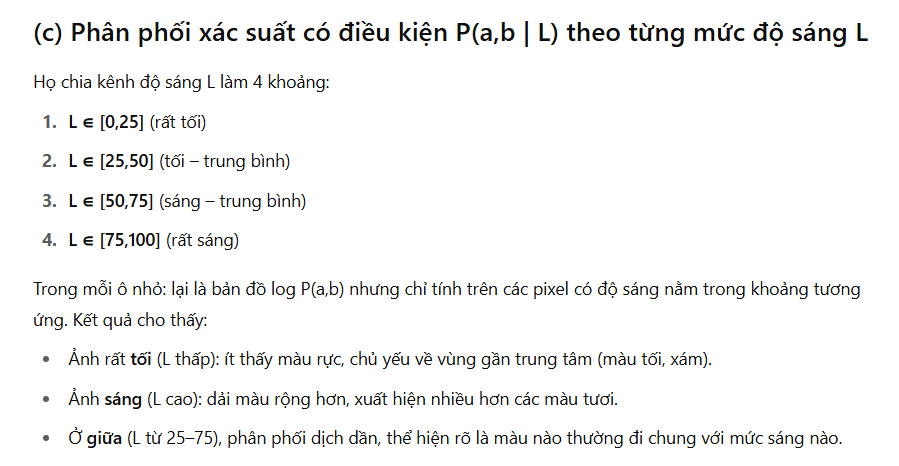
minh họa cách tác giả “phân vùng” và khảo sát phân phối màu trong không gian a–b của CIE Lab:



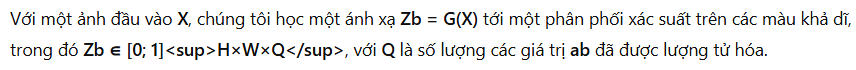
A là biểu diễn tọa độ của ảnh a-b khi sang RGB và giả sử độ sáng cố định là L=50



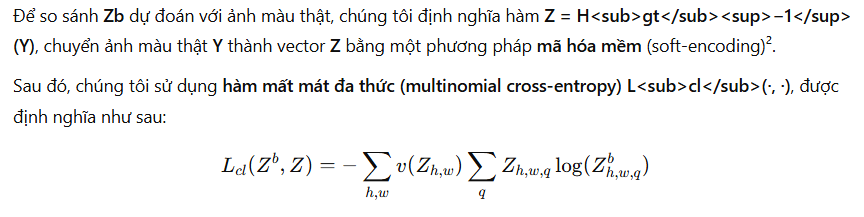
B là thống kế số màu xuất hiện nhiều và hiếm trong thực tế với đỏ vàng là nhứng màu xuất hiện rất nhiều trong tập huấn luyện( như da người, cỏ cây, trời). Xanh đậm là nhứng màu hiếm gặp, ít xuất hiện

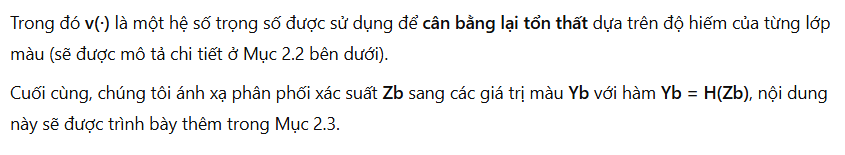


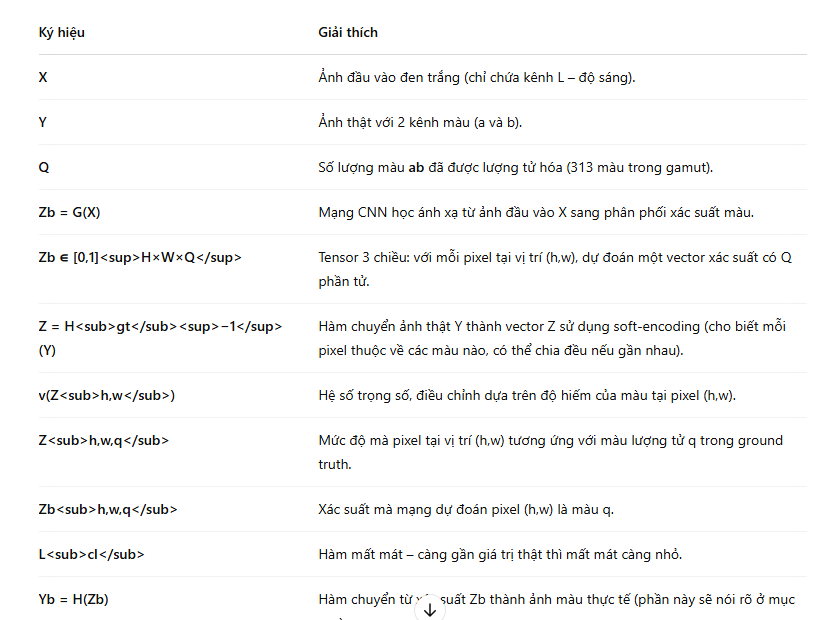
Ở 3C này thì chi biết độ sáng L thuốc [x,y] thì màu nào xuất hiện nhiều. sau khi lượng tử hóa thành 313 màu

* 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

-

- 



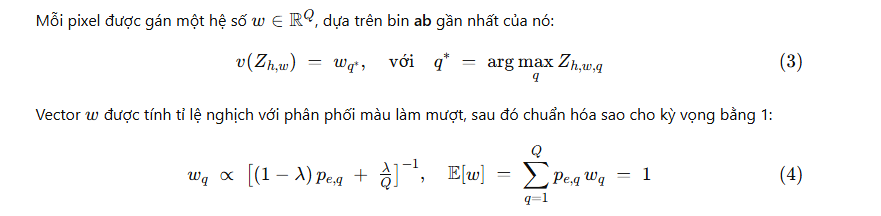
==🡺 Có ví dụ tính toán

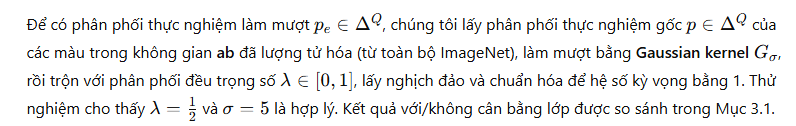
2.2 Cân bằng lại tần suất các lớp (Class rebalancing)

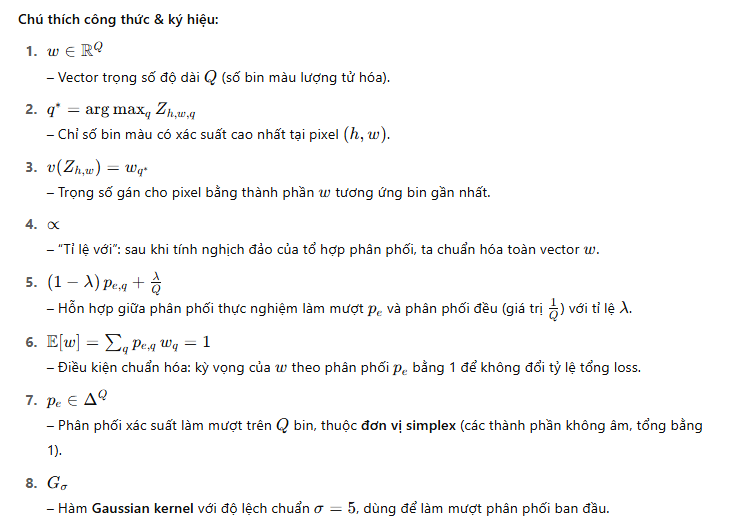
- Phân phối các giá trị **ab** trong ảnh tự nhiên có xu hướng bị lệch mạnh về các giá trị **ab thấp**, là do sự xuất hiện phổ biến của các nền như mây, vỉa hè, đất, và tường. Hình 3(b) cho thấy phân phối thực nghiệm của các pixel trong không gian **ab**, thu thập từ **1.3 triệu** ảnh huấn luyện trong 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].



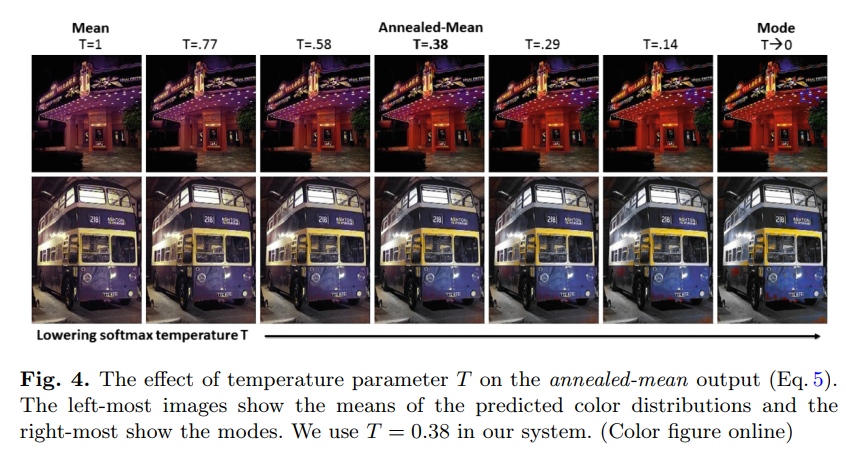




\*\* Cái này dùng để giải quyết vấn đề mất bằng dữ liệu màu sắc do các màu trung tính gây ra hay nói cách khác do ab thấp khiến màu nhạt và mất câng bằng

=🡺 Báo cáo không nói

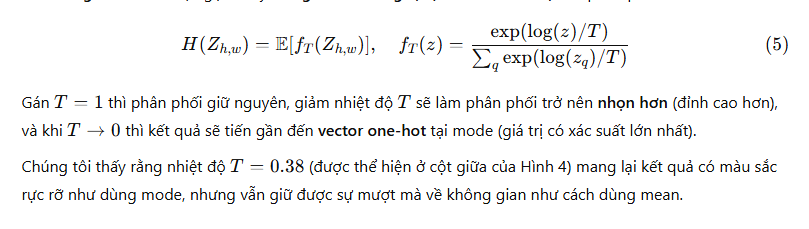
2.3 Từ phân phối xác suất lớp màu đến giá trị màu cụ thể



- Cuối cùng, chúng tôi định nghĩa hàm **H**, dùng để chuyển phân phối xác suất dự đoán **Zb** thành một giá trị điểm (point estimate) **Yb** trong không gian màu **ab**.

- Một cách đơn giản là chọn **mode** (giá trị có xác suất cao nhất) của phân phối dự đoán cho mỗi pixel, như minh họa ở cột ngoài cùng bên phải trong Hình 4 cho hai ảnh ví dụ. Cách này cho kết quả màu **rực rỡ**, nhưng đôi khi lại **không đồng nhất không gian**, ví dụ như có các đốm đỏ không hợp lý trên xe buýt.

- Ngược lại, nếu chọn **mean** (trung bình) của phân phối dự đoán, kết quả màu sẽ đồng nhất hơn về mặt không gian, nhưng màu bị **bạc (desaturated)**, giống như ảnh bị ám nâu (sepia) — đây là điều dễ hiểu vì việc lấy trung bình sau phân loại vẫn gặp vấn đề tương tự như tối ưu hóa với hàm mất mát Euclidean trong mô hình hồi quy.





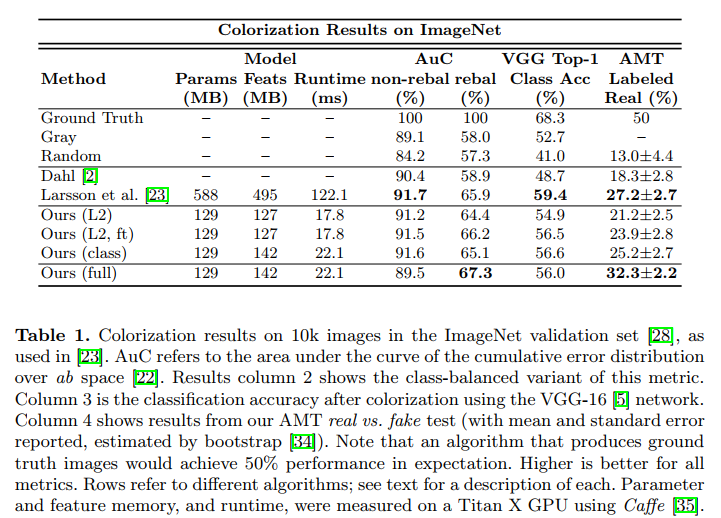
3 THỰC NGHIÊM

Trong **Mục 3.1**, chúng tôi đánh giá **khía cạnh đồ họa** của thuật toán, bằng cách kiểm tra **mức độ chân thực theo cảm nhận (perceptual realism)** của các ảnh tô màu mà mô hình tạo ra, cùng với một số **chỉ số chính xác khác**.

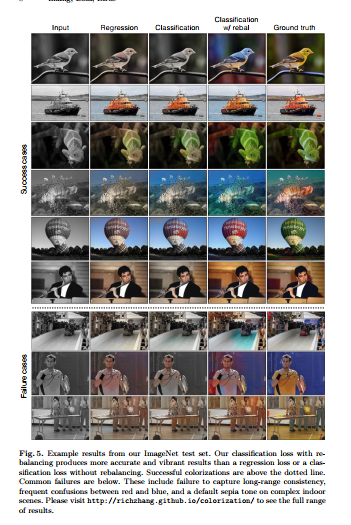
Trong **Mục 3.2**, chúng tôi kiểm tra việc **sử dụng bài toán tô màu ảnh như một phương pháp học biểu diễn tự giám sát (self-supervised representation learning)**.

Cuối cùng, trong **Mục 10.1**, chúng tôi trình bày **các ví dụ định tính** với những **bức ảnh đen trắng cổ**.

3.1 ĐÁNH GIÁ CHẤT LƯỢNG TÔ MÀU



* Chúng tôi huấn luyện mạng nơ-ron của mình trên **1.3 triệu hình ảnh** từ **tập huấn luyện ImageNet** [28], xác thực trên **10.000 ảnh đầu tiên** trong **tập xác thực ImageNet**, và **kiểm tra** trên **10.000 ảnh khác** trong cùng tập xác thực, giống như cách làm trong [23].
* Chúng tôi trình bày **kết quả định lượng** trong **Bảng 1**, dựa trên **ba chỉ số**. Một số **so sánh định tính** (bằng hình ảnh) giữa các trường hợp thành công và thất bại được thể hiện trong **Hình 5**. Để xem **so sánh với tập ảnh chọn ngẫu nhiên đầy đủ**, vui lòng truy cập **trang web dự án của chúng tôi**.
* Để kiểm tra rõ ràng ảnh hưởng của **các hàm mất mát (loss function)** khác nhau, chúng tôi huấn luyện mạng CNN với **nhiều loại hàm mất mát** khác nhau. Chúng tôi cũng so sánh với **phương pháp trước đó** [2] và **phương pháp đồng thời** [23], cả hai đều dùng **CNN huấn luyện trên ImageNet**, cùng với một số **phương pháp cơ bản đơn giản (naive baselines)**:
  + **Ours (full)**: Phương pháp đầy đủ của chúng tôi, sử dụng **hàm mất mát phân loại** được định nghĩa trong **Công thức (2)**, **kết hợp cân bằng lớp màu** như đã mô tả ở **Mục 2.2**. Mạng được huấn luyện **từ đầu** với **khởi tạo bằng k-means** [36], sử dụng thuật toán tối ưu **ADAM**, với khoảng **450.000 vòng lặp**³.
  + **Ours (class)**: Mạng nơ-ron sử dụng **hàm mất mát phân loại**, **nhưng không áp dụng cân bằng lớp màu** (λ = 1 trong **Công thức (4)**).



**Hình 5.**  
Ví dụ kết quả từ **tập kiểm tra ImageNet** của chúng tôi.  
**Hàm mất mát phân loại kết hợp với tái cân bằng lớp màu** (classification loss with rebalancing) tạo ra **kết quả chính xác và sống động hơn** so với việc sử dụng **hàm mất mát hồi quy** (regression loss) hoặc **phân loại mà không tái cân bằng**.

Các **kết quả tô màu thành công** được thể hiện ở **phía trên đường chấm chấm**.  
Các **thất bại phổ biến** được thể hiện **phía dưới**. Những lỗi phổ biến này bao gồm:

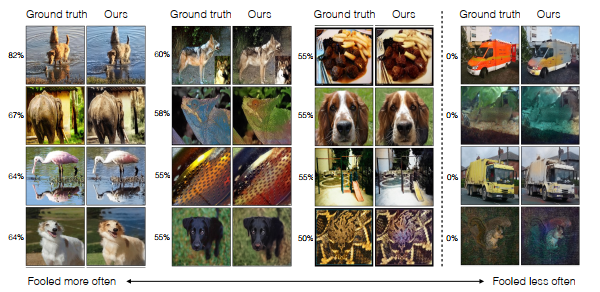
* không nắm bắt được tính nhất quán màu sắc trên phạm vi rộng (long-range consistency),
* thường xuyên nhầm lẫn giữa màu **đỏ và xanh lam**,
* hoặc tạo ra màu **sepia (nâu xám)** mặc định cho các cảnh trong nhà phức tạp.

 **Ours (full)**: Phương pháp đầy đủ của chúng tôi, sử dụng **hàm mất mát phân loại (classification loss)** như được định nghĩa trong Phương trình 2, và **tái cân bằng lớp màu** như mô tả trong Mục 2.2. Mạng nơ-ron được huấn luyện từ đầu bằng cách **khởi tạo k-means** [36], sử dụng **trình tối ưu ADAM** trong khoảng **450 nghìn lượt lặp (iterations)**.

* **Ours (class)**: Mạng nơ-ron của chúng tôi sử dụng **hàm mất mát phân loại**, nhưng **không có tái cân bằng lớp màu** (tức là **λ = 1** trong Phương trình 4).
* **Ours (L2)**: Mạng nơ-ron của chúng tôi được huấn luyện từ đầu với **hàm mất mát hồi quy L2** như mô tả trong Phương trình 1, theo cùng một quy trình huấn luyện.
* **Larsson et al. [23]**: Một phương pháp sử dụng **CNN** (mạng nơ-ron tích chập) cũng được trình bày trong hội nghị này.
* **Dahl [2]**: Một mô hình trước đó sử dụng **kim tự tháp Laplace (Laplacian pyramid)** trên các đặc trưng từ mạng VGG, được huấn luyện bằng **hàm mất mát hồi quy L2**.
* **Gray**: Tô màu mọi pixel bằng **màu xám**, với giá trị (a, b) = 0.
* **Random**: Sao chép màu sắc từ **một ảnh ngẫu nhiên** trong tập huấn luyện.

Việc đánh giá chất lượng của ảnh tô màu tổng hợp là một **nhiệm vụ khó**, vì các **chỉ số định lượng đơn giản**, như lỗi RMS trên giá trị pixel, thường **không thể hiện đúng độ chân thực về mặt thị giác**.  
Để khắc phục điểm yếu của từng phương pháp đánh giá riêng lẻ, chúng tôi kiểm tra **ba phương pháp** khác nhau để đo lường các **khía cạnh khác nhau về chất lượng**, được trình bày trong **Bảng 1**.

1. Phương pháp 1: Độ chân thực về cảm nhận (AMT):
   1. Đối với nhiều ứng dụng, chẳng hạn như trong lĩnh vực đồ họa, tiêu chí cuối cùng để đánh giá chất lượng tô màu là màu sắc có thuyết phục đối với người quan sát hay không.  
      Để kiểm tra điều này, chúng tôi đã thực hiện một thí nghiệm so sánh “ảnh thật hay giả” (real vs. fake) theo hình thức lựa chọn bắt buộc giữa hai lựa chọn (two-alternative forced choice) trên nền tảng Amazon Mechanical Turk (AMT).
   2. Người tham gia trong thí nghiệm được hiển thị một loạt cặp ảnh. Mỗi cặp bao gồm một ảnh màu thật bên cạnh một ảnh được tô màu lại, do thuật toán của chúng tôi hoặc một phương pháp so sánh khác tạo ra. Người tham gia được yêu cầu nhấp vào ảnh mà họ tin là ảnh có màu giả do máy tính tạo ra.
   3. Mỗi ảnh trong cặp có độ phân giải **256×256** và được hiển thị trong **1 giây**. Sau khi xem một cặp ảnh, người tham gia có **thời gian không giới hạn** để đưa ra câu trả lời.  
      Mỗi phiên thử nghiệm bao gồm **10 lượt luyện tập** ban đầu



Các con số là tỉ lệ người dùng chọn nhầm ảnh do mô hình tạo ra ảnh tô

1. Phương pháp 2 Tính diễn giải ngữ nghĩa (Semantic interpretability - phân loại bằng VGG):
   1. Phương pháp của chúng tôi có tạo ra các ảnh tô màu đủ **chân thực để một mô hình phân loại đối tượng có sẵn** có thể hiểu được không?
   2. Chúng tôi kiểm tra điều này bằng cách **đưa ảnh tô màu giả** (do thuật toán của chúng tôi tạo ra) vào một mạng VGG [5] của tác giả được huấn luyện để **dự đoán các lớp trong ImageNet** từ **ảnh màu thật**. Nếu mô hình phân loại hoạt động tốt, điều đó có nghĩa là ảnh tô màu đủ chính xác để **truyền đạt thông tin về loại đối tượng**.
   3. Việc sử dụng mô hình phân loại sẵn có để đánh giá độ chân thực của dữ liệu được tạo ra từng được đề xuất bởi [12].

**Kết quả** được thể hiện ở **cột thứ hai từ phải sang** trong Bảng 1:

* Khi **loại bỏ màu khỏi ảnh gốc**, độ chính xác của mô hình phân loại **giảm từ 68.3% xuống còn 52.7%**.
* Sau khi ảnh được tô lại bằng **phương pháp đầy đủ của chúng tôi**, độ chính xác **tăng lên 56.0%** (một vài biến thể khác của phương pháp cũng đạt kết quả cao hơn chút).
* Phương pháp của **Larsson et al. [23]** đạt hiệu suất cao nhất với **59.4%**.
* Là tài liệu tham khảo, một mạng phân loại VGG được **tinh chỉnh trên ảnh xám** đạt hiệu suất **63.5%**.

1. Phương pháp 3 **. Độ chính xác thô (Raw accuracy – AuC)**:

Là một phép thử ở mức thấp, chúng tôi tính **tỷ lệ phần trăm các màu pixel được dự đoán nằm trong một ngưỡng L2 nhất định** so với ảnh thật trong không gian màu **ab**. Sau đó, chúng tôi **quét qua các ngưỡng từ 0 đến 150** để tạo ra **hàm tích lũy (cumulative mass function)** như được giới thiệu trong [22], **tính diện tích dưới đường cong (Area under the Curve – AuC)** và **chuẩn hóa** kết quả.

Lưu ý rằng **chỉ số AuC này đo độ chính xác thuần túy**, trong khi phương pháp của chúng tôi **nhắm đến sự hợp lý và tự nhiên** của màu sắc hơn là độ chính xác tuyệt đối.

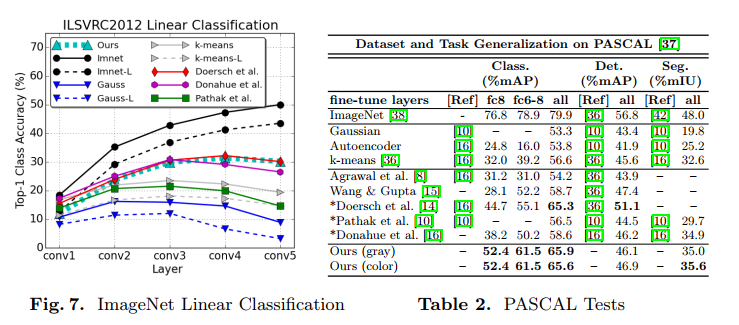
* Mạng của chúng tôi, \*\*được huấn luyện với loss phân loại nhưng **không có class-rebalancing**, cho kết quả tốt hơn mạng dùng L2 loss (nếu huấn luyện từ đầu).
* Khi mạng L2 được **fine-tune (tinh chỉnh)** từ mạng đã học phân loại màu, thì nó cho hiệu suất tương đương mạng phân loại, cho thấy **L2 có thể cho kết quả chính xác nếu khởi đầu tốt**, nhưng khó tối ưu nếu huấn luyện từ đầu.

Phương pháp của **Larsson et al. [23]** đạt **độ chính xác cao hơn một chút**.

Lưu ý: chỉ số này **bị chi phối bởi các pixel có độ bão hòa thấp**, vì ảnh tự nhiên thường chứa nhiều vùng màu xám hoặc ít màu sắc (như trình bày trong Hình 3(b)). Vì vậy, **ngay cả khi dự đoán tất cả pixel là màu xám (gray)**, kết quả vẫn **khá tốt**, và **phương pháp đầy đủ của chúng tôi với class rebalancing cũng chỉ đạt mức tương đương**.

Tuy nhiên, những vùng hình ảnh hấp dẫn về mặt thị giác **thường có độ bão hòa màu cao hơn**. Do đó, chúng tôi còn tính thêm **một biến thể cân bằng theo lớp (class-balanced AuC)** bằng cách **tái trọng số các pixel theo xác suất lớp màu (Equation 4, với λ = 0)**.

→ **Theo chỉ số này, phương pháp đầy đủ của chúng tôi vượt trội so với tất cả biến thể và phương pháp so sánh**, cho thấy việc **dùng class-rebalancing trong quá trình huấn luyện đã đạt được hiệu quả như mong đợi**.



Hình 7. Tổng quát hóa nhiệm vụ trên ImageNet  
Chúng tôi **giữ nguyên các mạng đã huấn luyện trước**, rồi học các bộ phân loại tuyến tính (linear classifiers) trên các tầng bên trong của mạng đối với nhiệm vụ phân loại ImageNet.  
Các đặc trưng được lấy trung bình (average-pooled), với **kích thước nhân và bước trượt bằng nhau**, cho đến khi số chiều đặc trưng (feature dimensionality) giảm xuống dưới 10k.

Các phương pháp được so sánh gồm:

* ImageNet [38]
* k-means [36]
* Gaussian
* và phương pháp của nhóm Zhang (Ours) với ảnh màu (solid line) và ảnh xám (dotted line)

Các phương pháp học đặc trưng khác như:

* Autoencoder
* Pathak et al.
* Doersch et al.
* Donahue et al.
* Agrawal et al.
* Wang & Gupta

Bảng 2. Tổng quát hóa nhiệm vụ và tập dữ liệu trên PASCAL

**Chúng tôi đánh giá khả năng tổng quát hóa** của các đặc trưng học được bằng cách thử nghiệm trên 2 tác vụ khác nhau trong tập dữ liệu **PASCAL VOC 2007**:

 **Phân loại (Class.):** dùng chỉ số **%mAP** (mean Average Precision)

 **Phát hiện vật thể (Det.):** dùng %mAP

 **Phân đoạn ảnh (Seg.):** dùng chỉ số **%mIU** (mean Intersection over Union)

Các phương pháp được huấn luyện (fine-tune) trên ảnh màu hoặc ảnh xám, sau đó được kiểm tra trên từng tác vụ.

* Cột “fine-tune layers” chỉ rõ các lớp nào được fine-tune (fc6, fc6-8 hoặc toàn bộ mạng).
* Những phương pháp có dấu \* (như Pathak, Doersch) chỉ được huấn luyện trên một phần nhỏ của AlexNet.
* Các phương pháp còn lại được khởi tạo bằng cách của [36] (k-means).

→ **Phương pháp của nhóm Zhang (Ours)** với ảnh **màu** cho kết quả tốt hơn so với ảnh **xám**, và gần bằng hoặc cao hơn so với nhiều phương pháp khác — cho thấy khả năng học được đặc trưng màu sắc hữu ích và tổng quát cho các nhiệm vụ khác.

3.2 Mã hóa giữa các kênh (Cross-Channel Encoding) như một phương pháp học đặc trưng tự giám sát

Bên cạnh việc đạt được tiến bộ trong nhiệm vụ đồ họa là tô màu ảnh, chúng tôi còn đánh giá cách mà việc tô màu có thể đóng vai trò như một **nhiệm vụ tiền đề (pretext task)** cho **học biểu diễn (representation learning)**.

Mô hình của chúng tôi giống như một mạng autoencoder, ngoại trừ việc **đầu vào và đầu ra là các kênh ảnh khác nhau**, nên chúng tôi gọi đây là **bộ mã hóa giữa các kênh (cross-channel encoder)**.

Để đánh giá biểu diễn đặc trưng mà mô hình học được qua quá trình mã hóa giữa các kênh này, chúng tôi thực hiện hai nhóm kiểm tra trên mạng của mình:

1. **Kiểm tra khả năng khái quát hóa tác vụ** bằng cách **giữ nguyên biểu diễn đã học** và chỉ huấn luyện các bộ phân loại tuyến tính để thực hiện phân loại đối tượng trên tập dữ liệu đã biết (được minh họa trong Hình 7).
2. **Tinh chỉnh lại mạng (fine-tune)** trên tập dữ liệu **PASCAL [37]** cho các tác vụ phân loại (classification), phát hiện (detection) và phân đoạn ảnh (segmentation). Ngoài việc kiểm tra trên các tác vụ mới, nhóm thí nghiệm này còn đánh giá **khả năng khái quát hóa theo tập dữ liệu**.

Để so sánh công bằng với các thuật toán học đặc trưng trước đó, chúng tôi huấn luyện lại một mạng **AlexNet [38]** trên nhiệm vụ tô màu, sử dụng phương pháp đầy đủ của chúng tôi, trong **450.000 vòng lặp**. Kết quả cho thấy biểu diễn đặc trưng mà mạng học được đạt **hiệu suất cao hơn** trên các tác vụ phân loại và phân đoạn đối tượng **so với các phương pháp trước đó** (xem Bảng 2).

**Phân loại ImageNet**

Mạng được huấn luyện trước để tô màu ảnh từ tập dữ liệu **ImageNet**, **không sử dụng thông tin nhãn phân loại (semantic labels)**. Chúng tôi kiểm tra xem các đặc trưng học được có biểu diễn tốt ngữ nghĩa cấp đối tượng hay không.  
Để làm điều đó, chúng tôi **đóng băng (freeze) trọng số của mạng**, cung cấp nhãn phân loại ngữ nghĩa, và **huấn luyện các bộ phân loại tuyến tính trên từng lớp tích chập (conv layer)**. Kết quả được trình bày ở **Hình 7**.

* **AlexNet** được huấn luyện trực tiếp trên nhiệm vụ phân loại ImageNet đạt hiệu suất cao nhất và được xem là "mức trần" (ceiling) cho bài kiểm tra này.
* Các khởi tạo ngẫu nhiên bằng **Gaussian** hoặc **k-means** [36] đạt hiệu suất cao nhất ở các lớp giữa (middle layers).
* Do mô hình của chúng tôi học trên ảnh **grayscale**, nên nó bị "thiệt thòi" vì thiếu thông tin màu ở đầu vào. Để đo ảnh hưởng của việc mất thông tin này, chúng tôi tinh chỉnh AlexNet trên ảnh grayscale và cũng chạy các phương pháp khởi tạo ngẫu nhiên trên ảnh grayscale.

Thú vị là, ở cả ba phương pháp, có một **khoảng cách hiệu suất khoảng 6%** giữa ảnh màu và ảnh xám, và khoảng cách này giữ gần như không đổi trong suốt mạng.

Chúng tôi cũng so sánh mô hình với các phương pháp **self-supervised learning** gần đây trên ImageNet [14,10,16].

* Ở lớp **conv1**, mô hình của chúng tôi có hiệu suất phân loại tuyến tính thấp hơn các phương pháp cạnh tranh như [14,16], nhưng tương đương với các phương pháp sử dụng đầu vào grayscale.
* Tuy nhiên, **ngay từ lớp conv2 trở đi**, mô hình của chúng tôi đạt hiệu suất **tương đương hoặc tốt hơn** các phương pháp khác. Điều này cho thấy rằng, mặc dù bị thiếu thông tin ở đầu vào, việc giải bài toán tô màu **khuyến khích mạng học ra các biểu diễn có khả năng phân tách ngữ nghĩa tốt**.

**Phân loại, phát hiện và phân đoạn ảnh trên PASCAL**

Chúng tôi kiểm tra mô hình trên các benchmark self-supervised phổ biến, bao gồm:

 **Phân loại (classification)**

 **Phát hiện đối tượng (detection)**

 **Phân đoạn ngữ nghĩa (segmentation)**

 trên tập dữ liệu **PASCAL VOC**, như giới thiệu trong [14,36,10]. Kết quả được trình bày ở **Bảng 2**.

 Mạng của chúng tôi đạt hiệu suất mạnh ở cả ba nhiệm vụ và đạt **kết quả tốt nhất (state-of-the-art)** trong **phân loại và phân đoạn**.

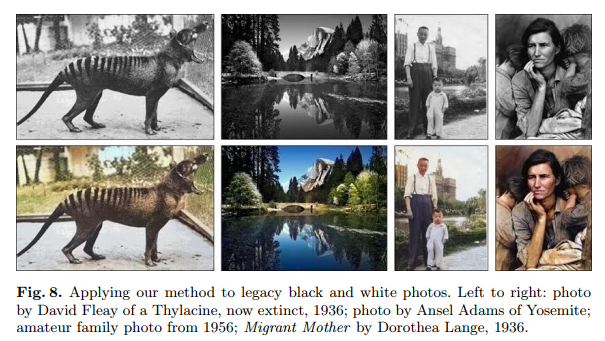
 Chúng tôi sử dụng kỹ thuật từ [36], **chuẩn hóa tốc độ học của các lớp** để tất cả học cùng tốc độ.

Chúng tôi kiểm tra mạng theo hai chế độ:

1. **Ours (gray)**: giữ đầu vào là ảnh grayscale, bỏ qua thông tin màu.
2. **Ours (color)**: chỉnh lớp conv1 để nhận đầu vào ảnh Lab đầy đủ 3 kênh, khởi tạo trọng số của hai kênh ab bằng 0.

**Kết quả chi tiết:**

* **Phân loại (PASCAL VOC 2007):** Theo quy trình từ [16], kết quả cao nhất đạt được khi conv1 bị "freeze", nghĩa là mạng chỉ thấy ảnh grayscale. Mô hình của chúng tôi đạt **độ chính xác cao nhất trong tất cả các phương pháp**.
* **Phát hiện đối tượng:** Sử dụng **Fast R-CNN** [41], theo quy trình trong [36].
  + **Doersch et al.** [14] đạt 51.1%
  + Chúng tôi đạt 46.9% (gray) và 47.9% (color)
  + Vượt qua baseline mạnh k-means [36] (45.6%)
  + Nhưng vẫn **chưa bằng mô hình huấn luyện có giám sát với ImageNet (56.8%)**
* **Phân đoạn ngữ nghĩa (PASCAL VOC 2012):** Dùng kiến trúc **FCN** [42] theo [10].
  + **Ours (gray):** đạt 35.0%
  + **Ours (color):** tăng lên 35.6%
  + Cả hai kết quả đều cao hơn các phương pháp so sánh khác.



Hình ảnh **Fig. 8** đang trình bày **kết quả của phương pháp tô màu ảnh đen trắng tự động (automatic colorization)** do nhóm tác giả nghiên cứu. Cụ thể:

 **Hàng trên**: Các bức ảnh **đen trắng gốc** (legacy black and white photos) từ nhiều thời kỳ lịch sử khác nhau.

 **Hàng dưới**: Các bức ảnh **sau khi được tô màu** bằng thuật toán của nhóm tác giả.

3.2 Ảnh Đen Trắng Di Sản

Vì mô hình của chúng tôi được huấn luyện bằng cách sử dụng các ảnh xám "giả" (tức là được tạo ra bằng cách loại bỏ các kênh màu *ab* từ ảnh màu gốc), nên chúng tôi cũng đã thử áp dụng phương pháp này trên các bức ảnh đen trắng **thật sự** từ quá khứ — như thể hiện trong **Hình 8** (các kết quả bổ sung có thể được xem trên trang web dự án của chúng tôi).

Có thể thấy rằng **mô hình vẫn có thể tạo ra các bản tô màu chất lượng tốt**, mặc dù các đặc trưng thống kê cấp thấp trong ảnh di sản **khá khác biệt** so với các bức ảnh hiện đại mà mô hình đã được huấn luyện.

4 Kết luận

Mặc dù tô màu ảnh là một nhiệm vụ thuộc lĩnh vực đồ họa máy tính mang tính chuyên biệt, nhưng nó cũng là một ví dụ của **bài toán dự đoán điểm ảnh khó** trong thị giác máy tính. Trong nghiên cứu này, chúng tôi đã chỉ ra rằng việc tô màu bằng mạng nơ-ron tích chập sâu (CNN) cùng với một **hàm mục tiêu được lựa chọn kỹ lưỡng** có thể tạo ra kết quả gần như không phân biệt được với ảnh màu thật.

Phương pháp của chúng tôi không chỉ cung cấp một đầu ra hữu ích trong đồ họa, mà còn có thể được xem như một **nhiệm vụ tiền huấn luyện (pretext task)** cho **học biểu diễn (representation learning)**.

Phương pháp của chúng tôi không chỉ cung cấp một đầu ra hữu ích trong đồ họa, mà còn có thể được xem như một **nhiệm vụ tiền huấn luyện (pretext task)** cho **học biểu diễn (representation learning)**. Mặc dù chỉ được huấn luyện để tô màu, mạng nơ-ron của chúng tôi **đã học được biểu diễn có ích đáng ngạc nhiên** cho các nhiệm vụ nhận dạng đối tượng, phát hiện và phân đoạn ảnh — và đạt hiệu suất tốt so với các phương pháp tự giám sát khác.