

# LOSSY TỐI MAGE CỨNG TUYỂN VỚI C NHÂN VIÊN AUTOENCODERS

Lucas Theis, Wenzhe Shi, Andrew Cunningham & Ferenc Huszár

Twitter

L

{ltheisy, wshi, acunningham, fhuszar} @ twitter.com

## ABSTRACT

Chúng tôi đề xuất một cách tiếp cận mới cho vấn đề tối ưu hóa mã tự động để nén ảnh bị mất. Các định dạng phương tiện mới, công nghệ phần cứng thay đổi, cũng như các yêu cầu và loại nội dung đa dạng tạo ra nhu cầu về các thuật toán nén có giá trị cao hơn các codec hiện có. Các bộ mã tự động có đủ khả năng để giải quyết nhu cầu này, nhưng khó tối ưu hóa trực tiếp do tính ổn định vốn có của việc mất nén. Ở đây, chúng tôi cho thấy rằng những thay đổi tối thiểu đối với tần số là phù hợp để đào tạo các bộ mã tự động sâu cạnh tranh với JPEG 2000 và làm tốt hơn các phương pháp được đề xuất gần đây dựa trên RNN. Hơn nữa, mạng của chúng tôi còn phù hợp với mặt tính toán nhờ vào kiến trúc điểm ảnh phụ, làm cho nó phù hợp với các hình ảnh có độ phân giải cao. Điều này trái ngược với công việc trước đây về bộ mã tự động để nén bằng cách sử dụng các phép gần đúng hơn,

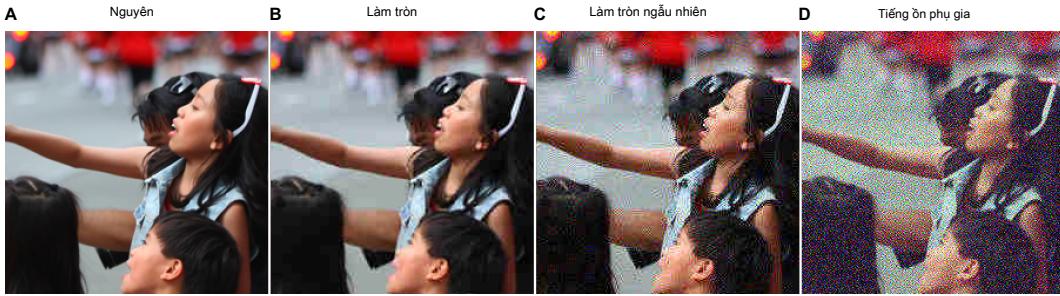
## 1 TÔI GIỚI THIỆU

Những tiến bộ trong việc đào tạo mạng nơ-ron đã giúp cải thiện hiệu suất trong một số nguồn cung cấp, nhưng mạng nơ-ron vẫn chưa vượt qua các codec hiện có trong việc nén hình ảnh mất dữ liệu. Các kết quả đầu tiên đã đạt được gần đây bằng cách sử dụng tự động mã (Ballé và cộng sự, 2016; Toderici và cộng sự, 2016b) - đặc biệt là trên các hình ảnh nhỏ (Toderici và cộng sự, 2016a; Gregor và cộng sự, 2016; van den Oord et al., 2016b) - và mạng nơ-ron đã đạt được kết quả tiên tiến nhất trong việc nén hình ảnh không mất dữ liệu (Theis & Bethge, 2015; van den Oord et al., 2016a).

Autoencoders có tiềm năng giải quyết nhu cầu ngày càng tăng về các thuật toán nén mất mát có thể thực hiện được. Tùy thuộc vào tình huống, các bộ mã hóa và giải mã có độ phức tạp tính toán khác nhau được yêu cầu. Khi gửi dữ liệu từ máy chủ đến thiết bị di động, có thể mong muốn ghép nối một bộ mã hóa mạnh mẽ với một bộ giải mã ít phức tạp hơn, nhưng các yêu cầu sẽ bị đảo ngược khi gửi dữ liệu theo hướng khác. Lượng sức mạnh tính toán và băng thông có sẵn cũng thay đổi theo thời gian khi có công nghệ mới. Đối với mục đích lưu trữ, thời gian mã hóa và giải mã ít quan trọng hơn so với các ứng dụng phát trực tuyến. Cuối cùng, các thuật toán nén hiện tại có thể không tối ưu cho các định dạng phương tiện mới như hình ảnh trường ánh sáng, video 360 hoặc nội dung VR. Trong khi việc phát triển một codec mới có thể mất nhiều năm,

Thật không may, nén mất dữ liệu là một vấn đề vốn dĩ không thể phân biệt được. Đặc biệt, lượng tử hóa là một phần không thể thiếu của đường ống nén nhưng không thể phân biệt được. Điều này làm cho việc đào tạo mạng nơ-ron cho nhiệm vụ này trở nên khó khăn. Các phép biến đổi hiện tại thường được chọn theo cách thủ công (ví dụ: phép chuyển đổi DCT được sử dụng trong JPEG) hoặc đã được tối ưu hóa cho một tác vụ khác với việc nén mất dữ liệu (ví dụ: Testa & Rossi, 2016, đã sử dụng việc khử mã tự động để nén). Ngược lại với hầu hết các công việc trước đó, nhưng phù hợp với Ballé et al. (2016), chúng tôi ở đây nhằm mục đích trực tiếp tối ưu hóa sự cân bằng biến dạng tỷ lệ được tạo ra bởi một bộ mã tự động. Chúng tôi đề xuất một cách tiếp cận đơn giản nhưng hiệu quả để đối phó với tính không phân biệt của lượng tử hóa dựa trên làm tròn,

Sử dụng phương pháp này, chúng tôi đạt được hiệu suất tương tự hoặc tốt hơn JPEG 2000 khi được đánh giá về chất lượng cảm nhận. Tuy nhiên, không giống như JPEG 2000, khung của chúng tôi có thể được tối ưu hóa cho nội dung cụ thể (ví dụ: hình thu nhỏ hoặc hình ảnh không tự nhiên), các chỉ số tùy ý và có thể dễ dàng tổng quát hóa cho các



Hình 1: Ảnh hưởng của việc làm tròn và các lựa chọn thay thế có thể phân biệt được khi được sử dụng làm thay thế trong nén JPEG. A: Cắt ảnh trước khi nén (GoToVan, 2014). B: Việc chặn các đồ tạo tác trong JPEG là do làm tròn hệ số DCT thành số nguyên gần nhất. Vì việc làm tròn được sử dụng tại thời điểm thử nghiệm, một giá trị gần đúng tốt sẽ tạo ra các đồ tạo tác tương tự. C: Làm tròn ngẫu nhiên đến số nguyên gần nhất tương tự như cách làm tròn của Toderici et al. (2016a). D: Tiếng ồn phụ gia đồng nhất (Ballé và cộng sự, 2016).

các hình thức của phương tiện truyền thông. Đáng chú ý, chúng tôi đạt được hiệu suất này bằng cách sử dụng kiến trúc mạng nơ-ron phù hợp cho phép giải mã hình ảnh lớn theo thời gian thực ngay cả trên các thiết bị tiêu dùng có công suất thấp.

## 2 CÔNG TY CỔ PHẦN TỰ ĐỘNG HÓA CÓ NHÂN VIÊN

Chúng tôi thiết kế một bộ mã tự động nén (CAE) để có ba thành phần: một bộ mã hóa  $f$ , một người giải mã  $g$ , và một mô hình xác suất  $Q$ ,

$$f: \mathbb{R}^N \rightarrow \mathbb{R}^M, g: \mathbb{R}^M \rightarrow \mathbb{R}^N, Q: \mathbb{Z}^M \rightarrow [0, 1]. \quad (1)$$

Phân phối xác suất rời rạc được xác định bởi  $Q$  được sử dụng để gán một số bit để biểu diễn lại dựa trên tần số của chúng, nghĩa là *mã hóa entropy*. Cả ba thành phần đều có thể có các tham số và mục tiêu của chúng tôi là tối ưu hóa sự cân bằng giữa việc sử dụng một số lượng bit nhỏ và có độ méo nhò,

$$\frac{\text{khối lượng}}{\text{Số lượng bit}} = Q\left(\frac{f(x)}{\beta \cdot d}\right) + \beta \cdot d \frac{g(f(x))}{\beta \cdot d}. \quad (2)$$

Đây,  $\beta$  kiểm soát sự cân bằng, dấu ngoặc vuông cho biết lượng tử hóa thông qua việc làm tròn đến số nguyên gần nhất và  $d$  đo độ méo được giới thiệu bằng mã hóa và giải mã. Đầu ra lượng tử hóa của bộ mã hóa là mã được sử dụng để đại diện cho một hình ảnh và được lưu trữ không mất mát. Nguồn thất thoát thông tin chính là lượng tử hóa (Phụ lục A.3). Thông tin bổ sung có thể bị bộ mã hóa loại bỏ và bộ giải mã có thể không giải mã hoàn hảo thông tin có sẵn, làm tăng độ méo.

Rất tiếc, chúng tôi không thể tối ưu hóa trực tiếp Phương trình 2 bằng cách sử dụng các kỹ thuật dựa trên gradient, như  $Q$  và  $[y]$  không thể phân biệt được. Hai phần sau đây đề xuất một giải pháp để đối phó với vấn đề này.

### 2.1 Q TỐI UỐNG HÓA VÀ CÁC THUẬT TOÁN KHÁC BIỆT

Đạo hàm của hàm làm tròn bằng 0 ở mọi nơi ngoại trừ các số nguyên, ở đó nó là unde fi ned. Chúng tôi đề xuất thay thế dẫn xuất của nó trong quá trình truyền ngược của nhân giống ngược (Rumelhart và cộng sự, 1986) với đạo hàm của một phép gần đúng tròn tru,  $r$ , nghĩa là, một cách hiệu quả để xác định đạo hàm trở thành

$$\frac{d}{dy} [y] = \frac{d}{dy} r(y). \quad (3)$$

Quan trọng là, chúng ta không thể thay thế hoàn toàn hàm làm tròn bằng một phép gần đúng tròn tru mà chỉ thay thế đạo hàm của nó, có nghĩa là quá trình lượng tử hóa vẫn được thực hiện như bình thường trong chuỗi tiếp. Nếu chúng tôi thay thế làm tròn hoàn toàn bằng một phép gần đúng tròn tru, bộ giải mã có thể học cách đảo ngược

xấp xỉ tròn tru, do đó loại bỏ cổ chai thông tin buộc mạng phải nén thông tin.

Theo kinh nghiệm, chúng tôi đã tìm thấy danh tính,  $r(y) = y$ , để làm việc cũng như lựa chọn tinh vi hơn. Điều này làm cho thao tác này dễ thực hiện, vì chúng ta chỉ cần chuyển các gradient mà không cần modi cation từ bộ giải mã sang bộ mã hóa.

Lưu ý rằng gradient liên quan đến các tham số của bộ giải mã có thể được tính toán mà không cần dùng đến các phép gần đúng, giả sử  $d$  có thể phân biệt được. Ngược lại với các cách tiếp cận liên quan, cách tiếp cận của chúng tôi có ưu điểm là nó không thay đổi độ dốc của bộ giải mã, vì chuyển tiếp được giữ nguyên.

Sau đây, chúng tôi thảo luận về các cách tiếp cận thay thế do các tác giả khác đề xuất. Được thúc đẩy bởi các liên kết đặc biệt để phối màu, Ballé et al. (2016) đề xuất thay thế lượng tử hóa bằng phụ gia nhiễu đồng nhất,

$$[f(x)] \approx f(x) + u. \quad (4)$$

Toderici và cộng sự. (2016a), mặt khác, sử dụng một dạng ngẫu nhiên hóa ngẫu nhiên (Williams, 1992). Tổng quát hóa ý tưởng này thành các số nguyên, chúng ta thực hiện phép toán làm tròn ngẫu nhiên sau:

$$\{y\} \approx b y c + \varepsilon, \varepsilon \in \{0, 1\}, P(\varepsilon = 1) = y - b y c, \quad (5)$$

Ở đây  $b \cdot c$  là toán tửoor. Trong phép quay ngược, đạo hàm được thay thế bằng đạo hàm của kỳ vọng,

$$\frac{d\{y\}}{dy} = \frac{dE[\{y\}]}{dy} = \frac{d}{dy} y = 1. \quad (6)$$

Hình 1 cho thấy hiệu quả của việc sử dụng hai lựa chọn thay thế này như một phần của JPEG, mà bộ mã hóa và bộ giải mã của chúng dựa trên sự biến đổi DCT khôn ngoan theo khối (Pennebaker & Mitchell, 1993). Lưu ý rằng đầu ra rõ ràng khác với đầu ra được tạo ra với lượng tử hóa thông thường bằng cách làm tròn và tín hiệu lỗi được gửi đến bộ mã tự động phụ thuộc vào những hình ảnh này. Trong khi trong Hình 1B, tín hiệu lỗi mà bộ giải mã nhận được sẽ là để loại bỏ các đồ tạo tác chặn, thì tín hiệu trong Hình 1D sẽ là để loại bỏ nhiễu tần số cao. Chúng tôi hy vọng sự khác biệt này sẽ ít gây ra vấn đề hơn với các số liệu đơn giản như sai số trung bình và có tác động lớn hơn khi sử dụng các phép đo độ méo có ý nghĩa nhận thức hơn.

Một giải pháp thay thế sẽ là chỉ sử dụng các giá trị gần đúng sau cho gradient của bộ mã hóa nhưng không sử dụng cho các gradient của bộ giải mã. Mặc dù điều này là có thể, nhưng nó phải trả giá bằng việc tăng độ phức tạp so sánh và triển khai, vì chúng ta sẽ phải thực hiện chuyển tiếp và chuyển tiếp qua bộ giải mã hai lần: một lần sử dụng làm tròn, một lần sử dụng xấp xỉ. Với cách tiếp cận của chúng tôi, gradient của bộ giải mã là chính xác ngay cả đối với một lần chuyển tiếp và chuyển tiếp.

## 2.2 Ế DỰ TOÁN TÝ LỆ NTROPY

Từ  $Q$  là một hàm rời rạc, chúng ta không thể phân biệt nó theo đối số của nó, điều này ngăn chúng ta tính toán gradient cho bộ mã hóa. Để giải quyết vấn đề này, chúng tôi sử dụng liên tục, sai phân xấp xỉ table. Chúng tôi giới hạn trên  $\int$  số lượng bit không phân biệt được bằng cách đầu tiên biểu thị phân phối của mô hình  $Q$  xét về mật độ xác suất  $q$ ,

$$Q(z) = \int_{[-5, 5]^M} q(z+u) du. \quad (7)$$

Giới hạn trên được đưa ra bởi:

$$-\text{khúc gõ}_2 Q(z) = -\text{khúc gõ}_2 \int_{[-5, 5]^M} q(z+u) du \leq \int_{[-5, 5]^M} -\text{khúc gõ}_2 q(z+u) du, \quad (\text{số 8})$$

trong đó bước thứ hai tiếp theo từ bất đẳng thức của Jensen (xem thêm Theis và cộng sự, 2016). Ước tính không chêch của giới hạn trên thu được bằng cách lấy mẫu  $u$  từ khối lập phương đơn vị  $[-5, 5]^M$ . Nếu chúng tôi sử dụng mật độ có thể phân biệt, ước tính này sẽ có thể phân biệt được trong  $z$  và do đó có thể được sử dụng để đào tạo bộ mã hóa.

## 2,3 V TỶ GIÁ BIT CÓ THỂ SẮP XẾP

Trong thực tế, chúng ta thường muốn kiểm soát được số lượng bit được sử dụng. Một cách để đạt được điều này là đào tạo một bộ mã tự động cho các cân bằng biến dạng tỷ lệ khác nhau. Nhưng điều này đòi hỏi chúng tôi phải đào tạo và lưu trữ một số lượng lớn các mô hình tiềm năng. Để giảm những chi phí này, chúng tôi cung cấp một bộ mã tự động được đào tạo trước cho các tỷ lệ khác nhau bằng cách giới thiệu các thông số tỷ lệ  $\lambda \in \mathbb{R}^M$ .

$$\text{- khúc gõ } q(f(x) \circ \lambda) + u + \beta \cdot d(x, g(f(x) \circ \lambda / \lambda)). \quad (9)$$

Đây,  $\circ$  cho biết phép nhân và phép chia theo điểm cũng được thực hiện theo điểm. Để giảm số lượng các thang đo có thể huấn luyện, chúng cũng có thể được chia sẻ trên các thứ nguyên. Ở đâu  $f$  và  $g$  là phức hợp, ví dụ: chúng tôi chia sẻ các thông số tỷ lệ trên các kích thước không gian nhưng không chia sẻ trên các kênh.

Một ví dụ về các tham số tỷ lệ đã học được thể hiện trong Hình 3A. Để kiểm soát chi tiết hơn đối với tốc độ bit, các thang đo được tối ưu hóa có thể được nội suy.

## 2,4 R CÔNG VIỆC ĐÃ ĐẠT ĐƯỢC

Có lẽ liên quan chặt chẽ nhất đến công việc của chúng tôi là công trình của Ballé et al. (2016). Sự khác biệt chính nằm ở cách chúng ta xử lý lượng tử hóa (xem Phần 2.1) và ước tính tốc độ entropy. Các biến đổi được sử dụng bởi Ballé et al. (2016) bao gồm một lớp tuyến tính duy nhất kết hợp với một dạng điều khiển độ lợi tương phản, trong khi khuôn khổ của chúng tôi dựa trên các mạng nơ-ron tích hợp sâu tiêu chuẩn hơn.

Toderici và cộng sự. (2016a) đề xuất sử dụng mạng nơ-ron tuần hoàn (RNN) để nén. Thay vì mã hóa entropy như trong công việc của chúng tôi, mạng cố gắng giảm thiểu sự biến dạng cho một số bit nhất định. Hình ảnh được mã hóa theo cách lặp đi lặp lại và việc giải mã được thực hiện theo từng bước để có thể tinh đến phần còn lại ở lần lặp tiếp theo. Một ưu điểm của thiết kế này là nó cho phép mã hóa hình ảnh tiến bộ. Một điểm bất lợi là quá trình nén tiêu tốn nhiều thời gian hơn so với cách tiếp cận của chúng tôi, vì chúng tôi sử dụng các mạng nơ-ron tích hợp phù hợp và không nhất thiết yêu cầu giải mã ở giai đoạn mã hóa.

Gregor và cộng sự. (2016) đã khám phá bằng cách sử dụng bộ mã tự động biến thể với bộ mã hóa và bộ giải mã lặp lại về khả năng xảy ra nhặt ký hoặc để nén các hình ảnh nhỏ. Loại [autoencoder này được đào tạo để tối đa hóa giới hạn dưới tương đương để giảm thiểu

$$\text{- E}_{p(y|x)} \text{khúc gõ } \frac{q(y) q(x|y)}{p(y|x)} \quad , \quad (10)$$

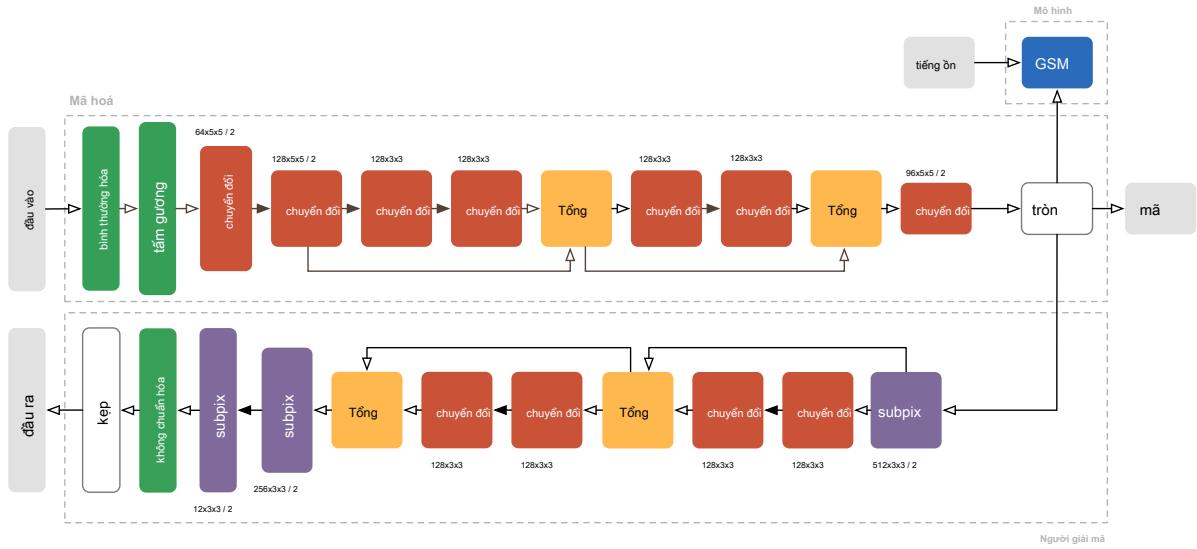
Ở đây  $p(y|x)$  đóng vai trò của bộ mã hóa và  $q(x|y)$  đóng vai trò của bộ giải mã. Trong khi Gregor và cộng sự. (2016) đã sử dụng phân phối Gaussian cho bộ mã hóa, chúng ta có thể liên kết cách tiếp cận của họ với công trình của Ballé et al. (2016) bằng cách giả định nó là đồng nhất,  $p(y|x) = f(x) + u$ . Nếu chúng ta cũng giả định một khả năng Gaussian w [phương sai thứ i fixe,  $q(x|y) = N(x|g(y), \sigma_2^2)$ ], hàm mục tiêu có thể được viết

$$\text{E}_{\text{khúc gõ }} q(f(x) + u) + \frac{1}{2} \frac{\|x - g(f(x) + u)\|^2 + C}{\sigma_2^2} \quad , \quad (11)$$

Đây,  $C$  là một hằng số bao gồm entropy âm của bộ mã hóa và hằng số chuẩn hóa của khả năng Gaussian. Lưu ý rằng phương trình này giống với sự đánh đổi giữa sự biến dạng tỷ giá với  $\beta = \sigma_2 / 2$  và lượng tử hóa được thay thế bằng nhiều đồng nhất phụ gia. Tuy nhiên, không phải tất cả các biến dạng đều có công thức tương đương như một bộ mã tự động biến thể (Kingma & Welling, 2014). Cái này chỉ

hoạt động nếu  $e \cdot d(x, y)$  có thể bình thường hóa trong  $x$  và hằng số chuẩn hóa không phụ thuộc vào  $y$ , hay nói cách khác  $C$  sẽ không phải là hằng số. So sánh thực nghiệm trực tiếp giữa cách tiếp cận của chúng tôi với các bộ mã tự động biến thể được cung cấp trong Phụ lục A.5.

Ollivier (2015) thảo luận về các bộ mã tự động biến thể để nén không mất dữ liệu cũng như các kết nối để khử mã tự động.



Hình 2: Minh họa kiến trúc bộ mã tự động nén được sử dụng trong bài báo này. Lấy cảm hứng từ công việc của Shi et al. (2016), hầu hết các chập được thực hiện trong không gian được lấy mẫu xuống để tăng tốc độ tính toán và việc lấy mẫu ngược được thực hiện bằng cách sử dụng các chập điểm ảnh phụ (các biến chập theo sau là định hình lại / chỉnh sửa lại các điểm nhận dạng). Để giảm bớt sự lỏn xộn, chỉ hiển thị hai khối còn lại của bộ mã hóa và bộ giải mã. Các biến đổi theo sau bởi các biến trở bị rò rỉ được biểu thị bằng các mũi tên liền mạch, trong khi các mũi tên trong suốt cho biết không có thêm các điểm phi tuyến tính. Là một mô hình cho sự phân bố của các coef lượng tử hóa, chúng tôi sử dụng các hỗn hợp thang đo Gauss. Ký hiệu  $C \times K \times K$

đề cập đến  $K \times K$  co giật với  $C$  fi mõ. Số theo sau dấu gạch chéo biểu thị bước tiến trong trường hợp chập và các hệ số lấy mẫu trong trường hợp chập điểm ảnh phụ.

### 3 E XÁC NHẬN

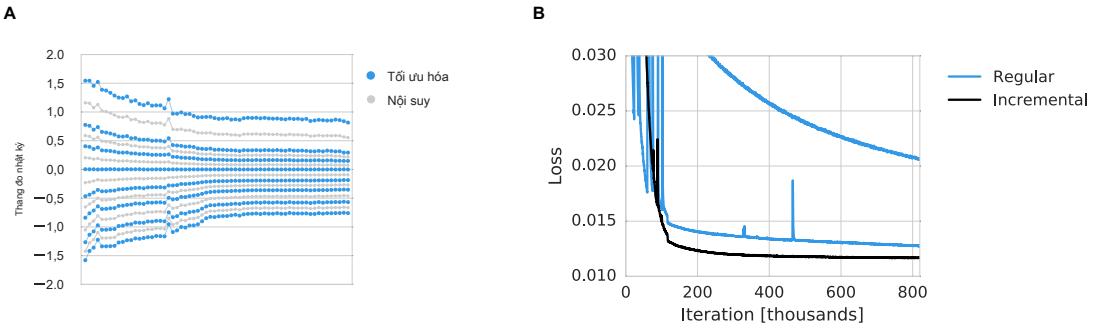
#### 3.1 E MÔ HÌNH NCODER, DECODER VÀ ENTROPY

Chúng tôi sử dụng mạng nơ-ron tích tụ phổ biến (LeCun và cộng sự, 1998) cho bộ mã hóa và bộ giải mã của bộ mã tự động nén. Kiến trúc của chúng tôi được lấy cảm hứng từ công trình của Shi et al. (2016), người đã chứng minh rằng siêu độ phân giải có thể đạt được hiệu quả hơn nhiều bằng cách hoạt động trong không gian có độ phân giải thấp, tức là bằng cách xoay các hình ảnh và sau đó lấy mẫu lên thay vì lấy mẫu lần đầu rồi sau đó xoay một hình ảnh.

Hai lớp đầu tiên của bộ mã hóa thực hiện tiền xử lý, cụ thể là lớp đệm phản chiếu và chuẩn hóa pixel-không ngoan. Phần đệm gương được chọn sao cho đầu ra của bộ mã hóa có cùng phạm vi không gian với hình ảnh được lấy mẫu xuống 8 lần. Chuẩn hóa tập trung phân phối các giá trị của mỗi kênh và đảm bảo kênh có phương sai đơn vị xác xỉ. Sau đó, hình ảnh được xoay vòng và thu nhỏ không gian trong khi đồng thời tăng số lượng kênh lên 128. Tiếp theo là ba khối dư (He et al., 2015), trong đó mỗi khối bao gồm thêm hai lớp chập với 128 fi lót mỗi. Một lớp tích chập cuối cùng được áp dụng và hệ số tích lũy được lấy mẫu lại một lần nữa trước khi lượng tử hóa thông qua việc làm tròn đến số nguyên gần nhất.

Bộ giải mã phản ánh kiến trúc của bộ mã hóa (Hình 9). Thay vì đệm gương và *có hiệu lực* chập, chúng tôi sử dụng chập không đệm. Việc lấy mẫu ngược được thực hiện thông qua tích chập được tạo ra bằng cách tổ chức lại các coef fi cient. Việc tổ chức lại này biến một tensor có nhiều kênh thành tensor có cùng chiều nhưng ít kênh hơn và phạm vi không gian lớn hơn (về phần đuôi, xem Shi và cộng sự, 2016). Một sự tích hợp và tổ chức lại các coef cùng nhau tạo thành một *lớp chập điểm ảnh phụ*. Sau ba khối còn lại, hai lớp chập điểm ảnh phụ nâng cao hình ảnh lên độ phân giải của đầu vào. Cuối cùng, sau khi không chuẩn hóa, các giá trị pixel được cắt thành

<sup>†</sup> Để đảm bảo tính tích cực, chúng tôi sử dụng một tham số hóa khác và tối ưu hóa thang đo nhật ký thay vì thang đo.



Hình 3: A: Thông số tỷ lệ thu được bằng cách tạo mã tự động nén (màu xanh lam). Có thể đạt được kiểm soát chi tiết hơn đối với tốc độ bit bằng cách nội suy thang đo (màu xám). Mỗi dấu chấm tương ứng với tham số tỷ lệ của một hệ số cho một sự cân bằng độ méo tốc độ cụ thể. Các coef được sắp xếp theo quy trình đào tạo gia tăng. B: So sánh giữa đào tạo gia tăng và đào tạo không tăng dần. Tỷ lệ học tập đã giảm sau 116.000 lần lặp lại (hai dòng dưới cùng). Huấn luyện không gia tăng ban đầu kém ổn định hơn và cho thấy hiệu suất kém hơn ở những lần tập luyện sau đó. Sử dụng một tốc độ học tập nhỏ ngay từ đầu sẽ ổn định quá trình đào tạo không gia tăng nhưng chậm hơn đáng kể (dòng trên cùng).

phạm vi từ 0 đến 255. Tương tự như cách chúng tôi xử lý các gradient của hàm làm tròn, chúng tôi xác định lại gradient của hàm clipping thành 1 bên ngoài phạm vi đã cắt. Điều này đảm bảo rằng tín hiệu huấn luyện khác 0 ngay cả khi các pixel được giải mã nằm ngoài phạm vi này (Phụ lục A.1).

Để mô hình hóa sự phân bố của coefficient  $\sum_s$  và ước tính tốc độ bit, chúng tôi sử dụng thang đo Gaussian độc lập hỗn hợp (GSM),

$$\text{khúc gõ}_2 q(z+u) = \sum_{t(i,j,k)} \frac{\pi_{ks}}{S} N(z^{kj} + u^{kj}; 0, \sigma^2_{ks}), \quad (12)$$

Ở đâu  $T(i, j, k)$  là tất cả các vị trí không gian và  $k$  lặp qua các kênh của vùng coefficient cho một hình ảnh duy nhất  $z$ . GSM được thiết lập tốt như những khối xây dựng hữu ích để mô hình hóa các phản ứng của ảnh tự nhiên (ví dụ, Portilla và cộng sự, 2003). Chúng tôi sử dụng 6 thang đo trong mỗi GSM. Thay vì sử dụng cách tham số phổ biến hơn ở trên, chúng tôi đã tham số hóa GSM để nó có thể dễ dàng được sử dụng với các phương pháp dựa trên gradient, tối ưu hóa trọng số log và độ chính xác log thay vì trọng số và phương sai. Chúng tôi lưu ý rằng bản chất từ vựng của GSM (Andrews & Mallows, 1974) có nghĩa là thuật ngữ tỷ lệ khuyến khích sự thua thót của người nhận.

Tất cả các mạng được triển khai bằng Python bằng Theano (2016) và Lasagne (Dieleman và cộng sự, 2015). Để mã hóa entropy của các coefficient được lượng tử hóa, trước tiên chúng tôi tạo ra các ước tính biểu đồ làm mịn Laplace của các phân phối coefficient trên một tập huấn luyện. Các xác suất ước tính sau đó được sử dụng với việc triển khai BSD được cấp phép công khai của một bộ mã hóa phạm vi<sup>2</sup>.

### 3.2 TỐI ĐÀO TẠO NOCREMENTAL

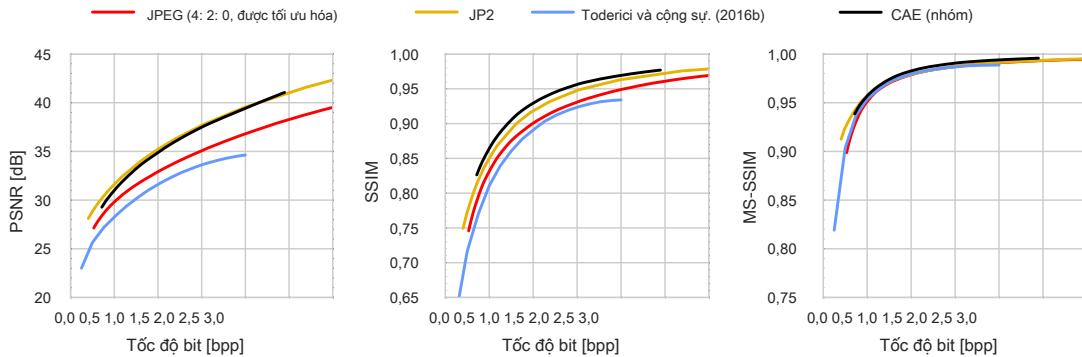
Tất cả các mô hình đã được đào tạo bằng cách sử dụng Adam (Kingma & Ba, 2015) áp dụng cho lô 32 hình ảnh  $128 \times 128$  kích thước pixel. Chúng tôi nhận thấy rất hữu ích khi tối ưu hóa lượng coefficient theo cách tăng dần (Hình 3B). Điều này được thực hiện bằng cách giới thiệu một mặt nạ nhị phân bổ sung  $m$ ,

$$- \text{khúc gõ}_2 q([f(x)] \circ m + u) + \beta \cdot d(x, g([f(x)] \circ m)). \quad (13)$$

Ban đầu, tất cả trừ 2 mục nhập của mặt nạ được đặt thành 0. Mạng được đào tạo cho đến khi các hiệu suất đạt dưới ngưỡng và sau đó một hệ số khác được kích hoạt bằng cách đặt một mục nhập của mặt nạ nhị phân thành 1. Sau khi tất cả các giá trị hệ số đã được bật, tốc độ học tập sẽ giảm so với giá trị ban đầu của  $10^{-4}$  đến  $10^{-5}$ . Đào tạo đã được thực hiện cho đến  $10^6$  cập nhật nhưng thường đạt hiệu suất tốt sớm hơn nhiều.

Sau khi một mô hình đã được đào tạo để đánh đổi sự cân bằng tỷ lệ biến dạng ( $\beta$ ), chúng tôi giới thiệu và fix các tham số tỷ lệ ne-diều chỉnh (Phương trình 9) cho các giá trị khác của  $\beta$  trong khi vẫn giữ tất cả các tham số khác fix. Đây

<sup>2</sup> <https://github.com/kazuho/rangecoder/>



Hình 4: So sánh các thuật toán nén khác nhau đối với PSNR, SSIM và MS-SSIM trên tập dữ liệu ảnh Kodak PhotoCD.

Chúng tôi lưu ý rằng đường màu xanh lam đề cập đến kết quả của Toderici et al. (2016b) đạt được *không có mã hóa entropy*.

chúng tôi đã sử dụng tỷ lệ học ban đầu là  $10^{-3}$  và liên tục giảm nó theo một hệ số  $\tau\kappa / (\tau + t)\kappa$ .

Ở đâu  $t$  là số lượng cập nhật hiện tại được thực hiện,  $\kappa =$  số 8, và  $\tau = 1000$ . Cân được tối ưu hóa cho 10.000 lần lặp. Để kiểm soát chi tiết hơn nữa đối với tốc độ bit, chúng tôi đã nêu suy giữa các thang đo được tối ưu hóa cho các cân bằng biến dạng tốc độ gần đó.

### 3,3 N HÌNH ẢNH THIÊN NHIÊN

Chúng tôi đã đào tạo những người tạo mã tự động nén trên 434 hình ảnh chất lượng cao được cấp phép theo giấy phép sáng tạo và được lấy từ [flickr.com](http://flickr.com). Các hình ảnh đã được lấy mẫu xuống bên dưới  $1536 \times 1536$  pixel và được lưu trữ dưới dạng PNG không mất dữ liệu để tránh các đố tạo tác nén. Từ những hình ảnh này, chúng tôi đã trích xuất  $128 \times$

128 cây trồng để đào tạo mạng lưới. Sai số trung bình bình phương được sử dụng làm thước đo độ méo trong quá trình luyện tập. Các siêu tham số ảnh hưởng đến kiến trúc mạng và đào tạo được đánh giá trên một tập hợp nhỏ các hình ảnh Flickr bị loại bỏ. Để thử nghiệm, chúng tôi sử dụng tập dữ liệu Kodak PhotoCD thường được sử dụng gồm  $24\ 768 \times 512$  hình ảnh pixel<sup>3</sup>.

Chúng tôi đã so sánh phương pháp của mình với JPEG (Wallace, 1991), JPEG 2000 (Skodras và cộng sự, 2001) và phương pháp dựa trên RNN của (Toderici và cộng sự, 2016b)<sup>4</sup>. Các bit cho thông tin tiêu đề không được tính vào tốc độ bit của JPEG và JPEG 2000. Trong số các biến thể khác nhau của JPEG, chúng tôi nhận thấy rằng JPEG được tối ưu hóa với lấy mẫu phụ sắc độ 4:2:0 thường hoạt động tốt nhất (Phụ lục A.2).

Mặc dù việc không điều chỉnh một bộ mã tự động nén duy nhất cho nhiều tốc độ bit hoạt động tốt, việc tối ưu hóa tất cả các thông số của mạng để cân bằng độ méo tốc độ cụ thể vẫn hoạt động tốt hơn. Ở đây, chúng tôi đã chọn sự thỏa hiệp của việc kết hợp các bộ mã tự động được đào tạo cho tốc độ bit thấp, trung bình hoặc cao (xem Phụ lục A.4 để biết thêm chi tiết).

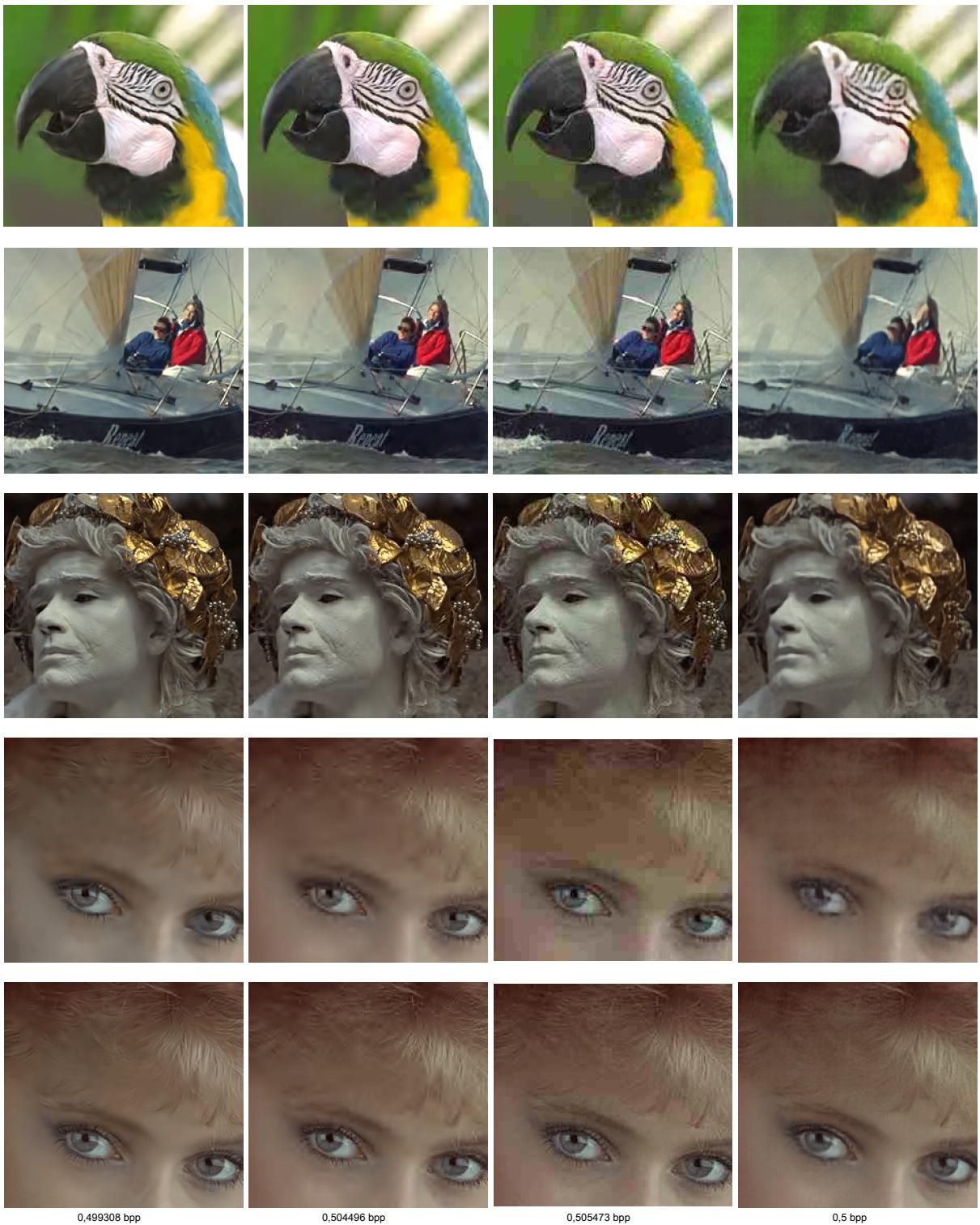
Đối với mỗi hình ảnh và tốc độ bit, chúng tôi chọn bộ mã tự động tạo ra sự biến dạng nhỏ nhất. Điều này làm tăng thời gian cần thiết để nén một hình ảnh, vì một hình ảnh phải được mã hóa và giải mã nhiều lần. Tuy nhiên, việc giải mã một hình ảnh vẫn nhanh như vậy, vì nó chỉ yêu cầu chọn và chạy một mạng bộ giải mã. Một giải pháp hiệu quả hơn nhưng có khả năng kém hiệu quả hơn sẽ là luôn chọn cùng một bộ mã tự động cho một sự cân bằng về độ méo tốc độ nhất định. Chúng tôi đã thêm 1 byte vào chi phí mã hóa để mã hóa bộ mã tự động nào của một nhóm được sử dụng.

Các đường cong tốc độ-điển dạng được tính trung bình trên tất cả các hình ảnh thử nghiệm được thể hiện trong Hình 4. Chúng tôi đã đánh giá các phương pháp khác nhau về PSNR, SSIM (Wang và cộng sự, 2004a) và SSIM đa cấp độ (MS-SSIM; Wang và cộng sự, 2004b). Chúng tôi đã sử dụng việc triển khai van der Walt et al. (2014) cho SSIM và triển khai của Toderici et al. (2016b) cho MS-SSIM. Chúng tôi nhận thấy rằng về mặt PSNR, phương pháp của chúng tôi thực hiện

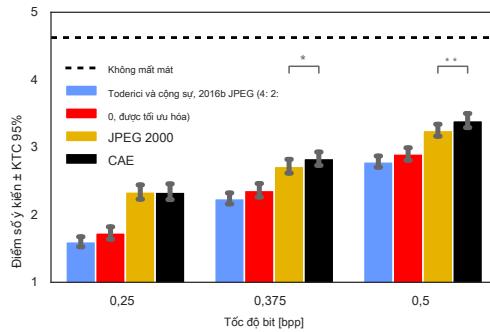
<sup>3</sup> <http://r0k.us/graphics/kodak/>

<sup>4</sup> Chúng tôi đã sử dụng mã được cung cấp trên <https://github.com/tensorflow/models/>

cây / 2390974a / nén. Chúng tôi lưu ý rằng tại thời điểm viết bài này, việc triển khai này không bao gồm mã hóa entropy như trong bài báo của Toderici et al. (2016b).



Hình 5: Ảnh chụp cận cảnh được tạo ra bởi các thuật toán nén khác nhau ở tốc độ bit tương đối thấp. Hàng thứ hai cho thấy một ví dụ trong đó phương pháp của chúng tôi hoạt động tốt, tạo ra các đường nét sắc nét hơn và ít đồ tạo tác hơn các phương pháp khác. Hàng thứ tư cho thấy một ví dụ mà phương pháp của chúng tôi gặp khó khăn, tạo ra các đồ tạo tác đáng chú ý trên tóc và làm đổi màu da. Ở tốc độ bit cao hơn, những vấn đề này biến mất và việc tái tạo CAE xuất hiện sắc nét hơn so với JPEG 2000 (hàng thứ f). Hình ảnh đầy đủ được cung cấp trong Phụ lục A.6.



Hình 6: Kết quả của một bài kiểm tra điểm ý kiến trung bình.

tương tự như JPEG 2000 mặc dù kém hơn một chút ở tốc độ bit thấp và trung bình và tốt hơn một chút ở tốc độ bit cao. Về mặt SSIM, phương pháp của chúng tôi vượt trội hơn tất cả các phương pháp đã được thử nghiệm khác. MS-SSIM tạo ra điểm rất giống nhau cho tất cả các phương pháp, ngoại trừ ở tốc độ bit rất thấp. Tuy nhiên, chúng tôi cũng cho rằng những kết quả này phụ thuộc nhiều vào hình ảnh. Kết quả cho từng hình ảnh được cung cấp dưới dạng tài liệu bổ sung 5.

Trong Hình 5, chúng tôi hiển thị các vùng ảnh được nén xuống tốc độ bit thấp. Phù hợp với kết quả định lượng, chúng tôi nhận thấy rằng các bản dựng lại JPEG 2000 trông giống với các bản dựng lại CAE hơn so với các bản dựng lại bằng các phương pháp khác. Tuy nhiên, đồ tạo tác do JPEG 2000 tạo ra có vẻ ồn hơn so với của CAE, mượt mà hơn và đôi khi có vẻ giống Gábor hơn.

Để định lượng chất lượng chủ quan của hình ảnh nén, chúng tôi đã chạy *Điểm số ý kiến* (MOS) thử nghiệm. Trong khi các bài kiểm tra MOS có những hạn chế của chúng, chúng là một tiêu chuẩn được sử dụng rộng rãi để đánh giá chất lượng cảm nhận (Streijl và cộng sự, 2014). Bộ thử nghiệm MOS của chúng tôi bao gồm 24 bản gốc không nén có độ phân giải đầy đủ từ tập dữ liệu Kodak, cũng như các hình ảnh giống nhau được nén bằng cách sử dụng từng thuật toán trong số bốn thuật toán ở hoặc gần ba tốc độ bit khác nhau: 0,25, 0,372 và 0,5 bit trên mỗi pixel. Chỉ có CAE tốc độ bit thấp mới được đưa vào thử nghiệm này.

Đối với mỗi hình ảnh, chúng tôi chọn cài đặt CAE tạo ra tốc độ bit cao nhất nhưng không vượt quá tốc độ bit mục tiêu. Tốc độ bit trung bình của hình ảnh nén CAE là 0,24479, 0,36446, và 0,48596, tương ứng. Sau đó, chúng tôi chọn hệ số chất lượng nhỏ nhất cho JPEG và JPEG 2000 mà tốc độ bit vượt quá tốc độ của CAE. Tốc độ bit trung bình cho JPEG là 0,25221, 0,37339 và 0,49534, cho JPEG 2000 0,24631, 0,36748 và 0,49373. Đối với một số hình ảnh, tốc độ bit của CAE ở cài đặt thấp nhất vẫn cao hơn tốc độ bit mục tiêu. Những hình ảnh này bị loại khỏi kết quả cuối cùng, chỉ còn lại 15, 21 và 23 hình ảnh, tương ứng.

Chất lượng cảm nhận của kết quả 273 hình ảnh được đánh giá bởi  $n = 24$  người đánh giá không phải chuyên gia. Một người đánh giá đã không hoàn thành thử nghiệm, vì vậy dữ liệu của cô ấy đã bị loại bỏ. Các hình ảnh được trình bày cho từng cá nhân theo thứ tự ngẫu nhiên. Những người đánh giá đã đưa ra một điểm số ý kiến riêng biệt cho mỗi hình ảnh từ thang điểm giữa 1 (xấu) đến 5 (thông minh). Trước khi đánh giá bắt đầu, các đối tượng được trình bày một hình ảnh hiệu chuẩn không nén có cùng kích thước với các hình ảnh thử nghiệm (nhưng không phải từ tập dữ liệu Kodak). Sau đó, họ được cho xem bốn phiên bản của hình ảnh hiệu chuẩn sử dụng cài đặt chất lượng kém nhất trong cả bốn phương pháp nén và đưa ra hướng dẫn “Đây là những ví dụ về hình ảnh nén. Đây là một số ví dụ về chất lượng tồi tệ nhất”.

Hình 6 cho thấy kết quả MOS trung bình cho mỗi thuật toán ở mỗi tốc độ bit. 95% khoảng thời gian được tính toán thông qua khởi động. Chúng tôi nhận thấy rằng CAE và JPEG 2000 đạt được MOS cao hơn JPEG hoặc phương pháp của Toderici et al. (2016b) ở tất cả các tốc độ bit mà chúng tôi đã thử nghiệm. Chúng tôi cũng nhận thấy rằng CAE không thể hoạt động tốt hơn JPEG 2000 ở 0,375 bpp ( $p < 0,05$ ) và 0,5 bpp ( $p < 0,001$ ).

<sup>s</sup> [https://figshare.com/articles/supplementary\\_zip/4210152](https://figshare.com/articles/supplementary_zip/4210152)

## 4 D BẢN LUẬN

Chúng tôi đã giới thiệu một cách đơn giản nhưng hiệu quả để giải quyết vấn đề không phân biệt được trong việc đào tạo autoencoders để nén mất dữ liệu. Cùng với chiến lược đào tạo gia tăng, điều này cho phép chúng tôi đạt được hiệu suất tốt hơn JPEG 2000 về độ tương tự như SSIM và MOS. Đáng chú ý, hiệu quả này đã đạt được bằng cách sử dụng một kiến trúc tích chập phù hợp, kết hợp với lượng tử hóa dựa trên lõi tròn đơn giản và một sơ đồ mã hóa entropy đơn giản. Các codec hiện tại thường được hưởng lợi từ sự hỗ trợ của thiết bị cứng, cho phép chúng chạy với chi phí năng lượng thấp. Tuy nhiên, các chip phần cứng được tối ưu hóa cho mạng nơ-ron phức hợp có thể sẽ sớm được phổ biến rộng rãi, do các mạng này hiện là chìa khóa để đạt được hiệu suất tốt trong rất nhiều ứng dụng.

Trong khi các thuật toán được đào tạo khác đã được chứng minh là cung cấp kết quả tương tự như JPEG 2000 (ví dụ: van den Oord & Schrauwen, 2014), theo hiểu biết của chúng tôi, đây là lần đầu tiên một kiến trúc được đào tạo từ đầu đến cuối được chứng minh để đạt được mức hiệu suất trên hình ảnh có độ phân giải cao. Một *end-to-end* autoencoder được đào tạo có lợi thế là nó có thể được tối ưu hóa cho các số liệu tùy ý. Thật không may, nghiên cứu về các số liệu phù hợp về mặt tri giác phù hợp với tối ưu hóa vẫn còn đang được ưa chuộng (ví dụ: Dosovitskiy & Brox, 2016; Ballé và cộng sự, 2016). Mặc dù tồn tại các thước đo tri giác có tương quan tốt với nhận thức của con người về một số dạng sai lệch nhất định (ví dụ: Wang và cộng sự, 2004a; La- parra và cộng sự, 2016), việc phát triển một thước đo tri giác có thể được tối ưu hóa là một nhiệm vụ khó khăn hơn, vì điều này đòi hỏi chỉ số phải hoạt động tốt đối với nhiều cặp hình ảnh và biến dạng lớn hơn nhiều.

Trong công việc trong tương lai, chúng tôi muốn khám phá việc tối ưu hóa các mã tự động nén cho các số liệu khác nhau. Một hướng dẫn hứa hẹn đã được trình bày bởi Bruna et al. (2016), người đã đạt được kết quả siêu phân giải thú vị bằng cách sử dụng các số liệu dựa trên mạng nơ-ron được đào tạo cho phân loại hình ảnh. Gatys và cộng sự. (2016) đã sử dụng các cách trình bày tương tự để đạt được bước đột phá trong việc chuyển giao phong cách có ý nghĩa về mặt tri giác. Một giải pháp thay thế cho các chỉ số cảm nhận có thể là sử dụng các mạng đối thủ chung (GAN; Goodfellow et al., 2014). Xây dựng dựa trên công trình của Bruna et al. (2016) và Dosovitskiy & Brox (2016), Ledig et al. (2016) gần đây đã chứng minh kết quả siêu phân giải ấn tượng bằng cách kết hợp GAN với các chỉ số dựa trên tính năng.

### A LỜI CẢM ƠN

Chúng tôi muốn cảm ơn Zehan Wang, Aly Tejani, Clément Farabet và Luke Alonso vì những phản hồi hữu ích về bản thảo.

### R TIỀN TÊ

DF Andrews và CL Mallows. Quy mô hỗn hợp của các phân phối chuẩn. *Tạp chí Hoàng gia Hiệp hội thống kê, Dòng B*, 36 (1): 99–102, 1974.

J. Ballé, V. Laparra và EP Simoncelli. Tối ưu hóa đầu cuối của mã biến đổi phi tuyến cho chất lượng cảm nhận. Trong *Hội nghị chuyên đề về mã hóa hình ảnh*, 2016.

J. Bruna, P. Sprechmann và Y. LeCun. Độ phân giải siêu cao với thống kê tích hợp sâu. Trong *Hội nghị quốc tế về đại diện học tập*, 2016.

S. Dieleman, J. Schluter, C. Raffel, E. Olson, SK Sonderby, D. Nouri, D. Maturana, M. Thoma, E. Battenberg, J. Kelly, J. De Fauw, M. Heilman, D. Moitinho de Almeida, B. McFee, H. Weide- man, G. Takacs, P. de Rivaz, J. Crall, G. Sanders, K. Rasul, C. Liu, G. French, và J. Degrave. Lasagne: Bản phát hành đầu tiên, năm 2015. URL <http://dx.doi.org/10.5281/zenodo.27878>.

A. Dosovitskiy và T. Brox. Tạo hình ảnh với các chỉ số tương đồng về tri giác dựa trên mạng sâu, 2016. arXiv: 1602.02644.

LA Gatys, AS Ecker và M. Bethge. Chuyển kiểu hình ảnh bằng cách sử dụng mạng nơ-ron tích tụ. Trong *Kỷ yếu của Hội nghị IEEE về Thị giác Máy tính và Nhận dạng Mẫu*, 2016.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, và Y. Bengio. Lưới đối thủ tạo ra. Trong *Những tiến bộ trong hệ thống xử lý thông tin thần kinh* 27, 2014.

- GoToVan. Lễ diễu hành ngày Canada, 2014. URL <https://www.flickr.com/photos/gotovan/14579921203>.
- K. Gregor, I. Danihelka, A. Graves và D. Wierstra. Hướng tới nén khái niệm, 2016. arXiv: 1601.06759.
- K. He, X. Zhang, S. Ren, và J. Sun. Học phần dư sâu để nhận dạng hình ảnh, 2015. arXiv: 1512.03385.
- D. Kingma và M. Welling. Tự động mã hóa bayes biến thể. Trong *Hội nghị quốc tế về Trình bày học tập*, 2014.
- DP Kingma và J. Ba. Adam: Một Phương pháp Tối ưu hóa Stochastic. Trong *Quốc tế Hội nghị về đại diện học tập*, 2015.
- V. Laparra, J. Ballé và EP Simoncelli. Đánh giá chất lượng hình ảnh cảm quan bằng cách sử dụng kim tự tháp Laplacian chuẩn hóa. Trong *SPIE Conf. về Thị giác con người và Hình ảnh Điện tử XXI*, 2016.
- Y. LeCun, L. Bottou, Y. Bengio và P. Haffner. Học tập dựa trên Gradient được áp dụng để nhận dạng tài liệu. 86 (11), 1998.
- C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Aitken, A. Tejani, J. Totz, Z. Wang và W. Shi. Ảnh thực tế đơn ảnh siêu phân giải siêu phân giải sử dụng mạng đối phương chung, 2016. arXiv: 1609.04802.
- Y. Ollivier. Bộ mã hóa tự động: tái tạo so với nén, 2015. 1403.7752.
- WB Pennebaker và JL Mitchell. *Chuẩn nén dữ liệu ảnh tĩnh JPEG*. Springer, thứ 3 ấn bản, 1993.
- J. Portilla, V. Strela, MJ Wainwright và EP Simoncelli. Làm giảm hình ảnh bằng cách sử dụng hỗn hợp tỷ lệ của người gaussia trong miền wavelet. *IIE Trans. Quy trình Hình ảnh.*, 12 (11): 1338–1351, 2003.
- DE Rumelhart, GE Hinton và RJ Williams. Học các biểu diễn bằng lõi lan truyền ngược. *Thiên nhiên*, 323 (6088): 533–536, 1986.
- W. Shi, J. Caballero, F. Huszar, J. Totz, A. Aitken, R. Bishop, D. Rueckert và Z. Wang. Độ phân giải siêu phân giải video và hình ảnh đơn theo thời gian thực bằng cách sử dụng mạng nơ-ron phức hợp pixel phụ nhanh chóng. Trong *Hội nghị IEEE về Thị giác Máy tính và Nhận dạng Mẫu*, 2016.
- A. Skodras, C. Christopoulos và T. Ebrahimi. Chuẩn nén ảnh tĩnh JPEG 2000. *Tạp chí Xử lý Tín hiệu*, 18 (5): 36–58, 2001.
- RC Streijl, S. Winkler và DS Hands. Điểm ý kiến trung bình (MOS) đã được xem xét lại: các phương pháp và ứng dụng, các hạn chế và các lựa chọn thay thế. *Hệ thống đa phương tiện*, trang 1–15, 2014.
- D. Del Testa và M. Rossi. Nén tổn thất nhẹ của các mẫu sinh trắc học thông qua Bộ mã hóa tự động khử nhiễu. *Thu xử lý tín hiệu IEEE*, 22 (12), năm 2016.
- Nhóm phát triển Theano. Theano: Một khuôn khổ Python để tính toán toán học nhanh chóng biến thức, 2016. arXiv: 1605.02688.
- L. Theis và M. Bethge. Tạo mô hình ảnh bằng cách sử dụng LSTM không gian. Trong *Những tiến bộ trong thần kinh Hệ thống xử lý thông tin* 28, 2015.
- L. Theis, A. van den Oord, và M. Bethge. Một lưu ý về việc đánh giá các mô hình tổng quát. Trong *Các Hội nghị quốc tế về đại diện học tập*, 2016.
- G. Toderici, SM O'Malley, SJ Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, và R. Sukthankar. Nén hình ảnh tỷ lệ thay đổi với mạng nơ-ron tuần hoàn. Trong *Hội nghị quốc tế về đại diện học tập*, 2016a.
- G. Toderici, D. Vincent, N. Johnston, SJ Hwang, D. Minnen, J. Shor, và M. Covell. Nén hình ảnh độ phân giải đầy đủ với mạng nơ-ron tái phát, 2016b. arXiv: 1608.05148v1.

- A. van den Oord và B. Schrauwen. Hỗn hợp student-t như một bản vá hình ảnh tự nhiên trước khi ứng dụng để nén hình ảnh. *Tạp chí Nghiên cứu Máy học*, 15 (1): 2061–2086, 2014.
- A. van den Oord, N. Kalchbrenner và K. Kavukcuoglu. Mạng nơ-ron lặp lại pixel. Trong *Ký yếu của Hội nghị Quốc tế lần thứ 33 về Học máy*, 2016a.
- A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves và K. Kavukcuoglu. Tạo hình ảnh tùy chọn với Bộ giải mã PixelCNN, 2016b. arXiv: 1606.05328v2.
- S. van der Walt, JL Schönberger, J. Nunez-Iglesias, F. Boulogne, J.D Warner, N. Yager, E. Gouil-lart, và T. Yu. scikit-image: xử lý hình ảnh bằng Python. *PeerJ*, 2 năm 2014.
- GK Wallace. Chuẩn nén ảnh tĩnh JPEG. *Thông tin liên lạc của ACM*, 34 (4): 30–44, 1991.
- Z. Wang, AC Bovik, HR Sheikh và EP Simoncelli. Đánh giá chất lượng hình ảnh: từ khả năng hiển thị lỗi đến độ tương đồng về cấu trúc. *Giao dịch IEEE về Xử lý hình ảnh*, 13 (4): 600–612, 2004a.
- Z. Wang, EP Simoncelli và AC Bovik. Sự giống nhau về cấu trúc đa tỷ lệ để đánh giá chất lượng hình ảnh. Trong *Biên bản Hội nghị của Hội nghị Asilomar lần thứ 31 về Tin hiệu, Hệ thống và Máy tính*, tập 2, trang 1398–1402, 2004b.
- RJ Williams. Các thuật toán theo sau gradient thống kê đơn giản để học tập cùng cố kết nối. *Học máy*, 8 (3-4): 229–256, 1992.

## AA PHỤ LỤC

### A.1 GRADIENT OF CLIPPING

Chúng tôi xác định lại độ dốc của thao tác cắt là không đổi,

$$\frac{d \text{kẹp}_{0.255}(\hat{x})}{d\hat{x}} = 1. \quad (14)$$

Hãy xem xét điều này ảnh hưởng như thế nào đến các gradient của tổn thất bình phương,

$$\frac{d}{d\hat{x}} (\text{kẹp}_{0.255}(\hat{x}) - x)^2 = 2 (\text{kẹp}_{0.255}(\hat{x}) - x) \frac{d \text{kẹp}_{0.255}(\hat{x})}{d\hat{x}}. \quad (15)$$

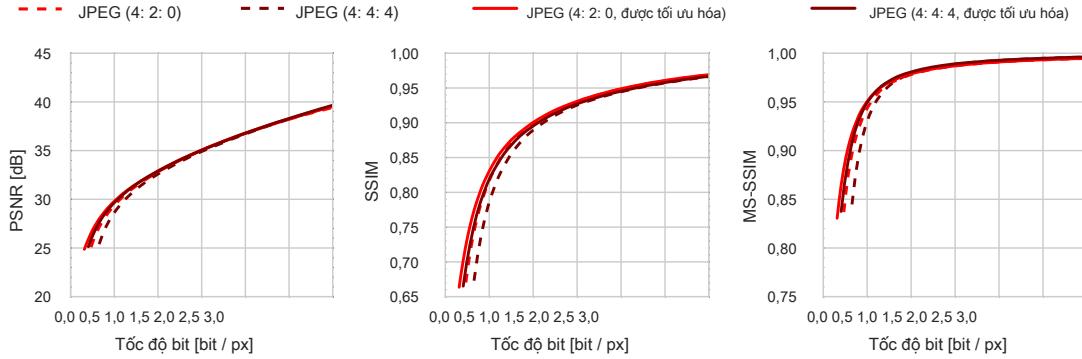
Giả định rằng  $\hat{x}$  lớn hơn 255. Nếu không xác định lại đạo hàm, tín hiệu lỗi sẽ bằng 0 và không hữu ích. Nếu không có bất kỳ đoạn cắt nào, tín hiệu lỗi sẽ phụ thuộc vào giá trị của  $\hat{x}$ , mặc dù bất kỳ giá trị nào trên 255 sẽ có cùng tác động đến sự mất mát tại thời điểm thử nghiệm. Mặt khác, việc sử dụng clipping nhưng một tín hiệu khác trong đường chuyển ngược là trực quan, vì nó tạo ra một tín hiệu lỗi tỷ lệ với lỗi cũng sẽ phát sinh tại thời gian thử nghiệm.

### A.2 D CÁC CHẾ ĐỘ KHÁC NHAU CỦA JPEG

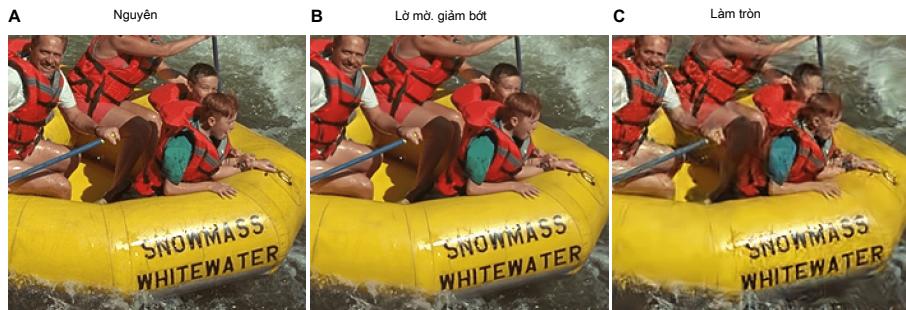
Chúng tôi so sánh JPEG được tối ưu hóa và không được tối ưu hóa với (4: 2: 0) và không có (4: 4: 4) lấy mẫu phụ sắc độ. JPEG được tối ưu hóa tính toán đặc điểm của bảng Huffman vào một hình ảnh nhất định, trong khi JPEG chưa được tối ưu hóa sử dụng bảng Huffman chuẩn bị trước. Chúng tôi không đếm các bit được phân bổ cho tiêu đề của định dạng file, nhưng đối với JPEG được tối ưu hóa, chúng tôi đã đếm các bit cần thiết để lưu trữ bảng Huffman. Chúng tôi nhận thấy rằng trung bình, JPEG được lấy mẫu theo màu sắc và được tối ưu hóa hoạt động tốt hơn trên tập dữ liệu Kodak (Hình 7).

### A.3 C KHẢ NĂNG LÀM VIỆC VÀ GIẢM KÍCH THƯỚC

Vì một số thực duy nhất có thể mang nhiều thông tin như một thực thể nhiều chiều, nên việc giảm thiểu số chiều không ảnh hưởng đến việc nén. Tuy nhiên, nếu chúng tôi hạn chế kiến trúc của bộ mã hóa, nó có thể buộc phải loại bỏ một số thông tin nhất định. Để hiểu rõ hơn về cách thông tin bị mất do giảm kích thước và lượng thông tin bị mất do lượng tử hóa, Hình 8 cho thấy các bản tái tạo được tạo ra bởi một bộ mã tự động nén có và không có lượng tử hóa. Hiệu quả của việc giảm kích thước là nhỏ nhất so với ảnh hưởng của quá trình lượng tử hóa.

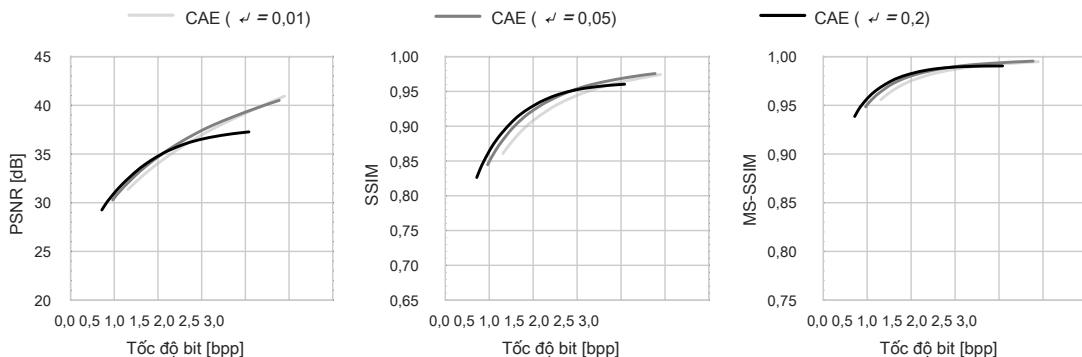


Hình 7: So sánh các chế độ JPEG khác nhau trên tập dữ liệu ảnh Kodak PhotoCD. Các bảng Huffman được tối ưu hóa hoạt động tốt hơn các bảng Huffman mặc định.



Hình 8: Để gỡ rối các ảnh hưởng của lượng tử hóa và giảm kích thước, chúng tôi đã dựng lại các hình ảnh bị vô hiệu hóa lượng tử hóa. A: Hình ảnh ban đầu không nén. B: Một bản tái tạo được tạo ra bởi một bộ mã tự động nén, nhưng với thao tác làm tròn đã bị loại bỏ. Kích thước của đầu ra của bộ mã hóa là  $3 \times$  nhỏ hơn đầu vào. C: Một bản tái tạo được tạo bởi cùng một bộ mã tự động nén. Trong khi các tác động của việc giảm kích thước hầu như không thể nhận thấy, lượng tử hóa giới thiệu các đồ tạo tác có thể nhìn thấy được.

#### A.4 ENSEMBLE



Hình 9: So sánh các CAE được tối ưu hóa cho tốc độ bit thấp, trung bình hoặc cao.

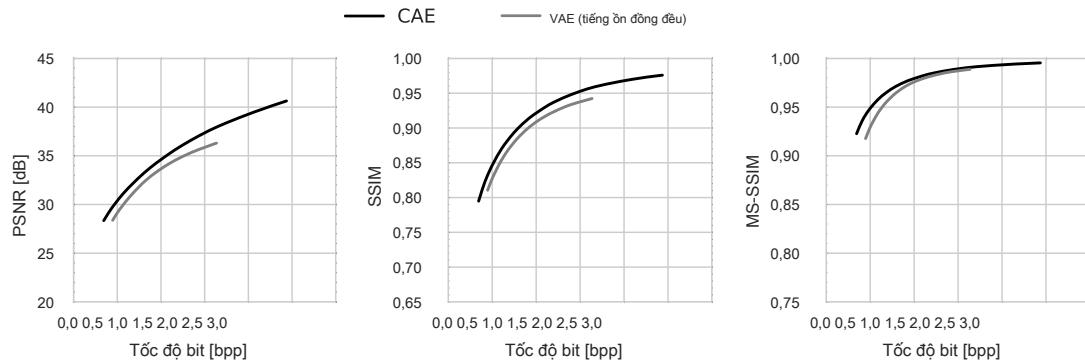
Để đưa thông số kiểm soát sự cân bằng giữa biến dạng tỷ lệ vào một phạm vi trực quan hơn, chúng tôi đã thay đổi tỷ lệ của thuật ngữ biến dạng và thể hiện mục tiêu như sau:

$$-\frac{\alpha}{N} \ln q([f(x) \circ \lambda] + u) + \frac{1 - \alpha \cdot \|x - g([f(x) \circ \lambda]/\lambda)\|_2^2}{1000 \cdot M} \quad (16)$$

Đây,  $N$  là số lượng coef được tạo ra bởi bộ mã hóa và  $M$  là chiều của  $x$  (tức là gấp 3 lần số pixel).

CAE tốc độ bit cao đã được đào tạo với  $\alpha = 0,01$  và 96 kênh đầu ra, CAE tốc độ bit trung bình đã được đào tạo với  $\alpha = 0,05$  và 96 kênh đầu ra, và CAE tốc độ bit thấp đã được đào tạo với  $\alpha = 0,2$  và 64 kênh đầu ra.

#### A.5 COMPARISON VỚI VAE



Hình 10: Một giải pháp thay thế cho cách tiếp cận của chúng tôi là thay thế hàm làm tròn bằng nhiều đơn dạng cộng thêm trong quá trình huấn luyện (Ballé et al., 2016). Sử dụng sai số trung bình để đo độ méo, tối ưu hóa tốc độ biến dạng theo cách này tương đương với việc đào tạo một bộ mã tự động biến thiên (Kingma & Welling, 2014) với khả năng Gaussian và bộ mã hóa đồng nhất (Phần 2.4). Sử dụng cùng một kiến trúc bộ mã tự động bao mật đào tạo cho cả hai phương pháp (ở đây được đào tạo cho tốc độ bit cao), chúng tôi nhận thấy rằng nhiều công hoạt động kém hơn các dẫn xuất red ning như trong cách tiếp cận của chúng tôi. Lượng tử hóa dựa trên làm tròn được sử dụng lại thời điểm thử nghiệm theo cả hai cách tiếp cận.

#### A.6 CẢM ỨNG CỦA NHÂN VIÊN

Dưới đây chúng tôi hiển thị các hình ảnh hoàn chỉnh tương ứng với các loại cây trồng trong Hình 5. Đối với mỗi hình ảnh, chúng tôi hiển thị hình ảnh gốc (trên cùng bên trái), tái tạo bằng CAE (trên cùng bên phải), tái tạo bằng JPEG 2000 (dưới cùng bên trái) và tái tạo bằng phương pháp (Toderici và cộng sự, 2016b) (dưới cùng bên phải). Hình ảnh được xem tốt nhất trên màn hình điều khiển.