

## **Lời cảm ơn**

Lời đầu tiên, em xin bày tỏ sự cảm ơn chân thành đối với Cô giáo, TS. Nguyễn Thị Mỹ Bình – giáo viên hướng dẫn trực tiếp em.

Em cũng xin gửi lời cảm ơn tới các thầy cô trong khoa Công nghệ thông tin, trường Đại học Công Nghiệp Hà Nội đã hướng dẫn, chỉ bảo và tạo điều kiện cho em học tập cũng như nghiên cứu trong thời gian qua.

Cảm ơn Câu lạc bộ HIT, Đội Olympic Tin học khoa Công nghệ thông tin đã đồng hành cùng em trong suốt quãng thời gian học tập, làm việc tại trường.

Mặc dù đã cố gắng hoàn thành báo cáo đồ án tốt nghiệp này nhưng chắc chắn sẽ không tránh khỏi những sai sót, em kính mong nhận được sự thông cảm và chỉ bảo của các thầy cô và các bạn.

---

## Mục lục

<b>MỞ ĐẦU</b> .....	4
0.1 Lý do chọn đề tài .....	4
0.2 Mục đích của đề tài .....	4
0.3 Đối tượng và phạm vi nghiên cứu của đề tài .....	4
0.4 Bảng các ký hiệu .....	7
<b>1 Nghiên cứu tổng quan</b> .....	8
1.1 Các phương pháp nghiên cứu .....	8
1.2 Ưu nhược điểm của các phương pháp .....	8
<b>2 Cơ sở lý thuyết</b> .....	11
2.1 Tổng quan về các kĩ thuật nén mất mát thông tin .....	11
2.2 Các kĩ thuật nén codec thường được sử dụng .....	11
2.3 Bộ mã hóa tự động (Autoencoder) .....	11
<b>3 Triển khai chương trình và đánh giá hiệu suất</b> .....	18
3.1 Phân tích .....	18
3.2 Xây dựng .....	19

---

<b>4</b>	<b>Kết luận và hướng phát triển . . . . .</b>	<b>27</b>
4.1	Kết luận . . . . .	27
4.2	Hướng phát triển . . . . .	27
	<b>Tài liệu tham khảo . . . . .</b>	<b>29</b>

---

# MỞ ĐẦU

## 0.1. Lý do chọn đề tài

Với sự phát triển không ngừng của khoa học và công nghệ, đặc biệt là 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 phương pháp nén hiện tại (codec). Chúng em lựa chọn đề tài này nhằm tìm kiếm, xây dựng, triển khai các phương pháp nén mới tận dụng sự mạnh mẽ của phần cứng ngày nay.

## 0.2. Mục đích của đề tài

- Phân tích các kĩ thuật nén cũ (codec)
- Xây dựng, tìm kiếm các kĩ thuật, phương pháp mới sử dụng trong nén dữ liệu đa phương tiện hiệu quả
- Tận dụng sự mạnh mẽ của phần cứng.

## 0.3. Đối tượng và phạm vi nghiên cứu của đề tài

### 0.3.1. Đối tượng

- Các kĩ thuật nén thường được sử dụng (DCT, Huffman, kmean, ... )
  - Các phương pháp nén được phát triển và thể hiện tính hiệu quả trong thời gian gần đây
  - Bộ mã hóa tự động (Autoencoder)
-

- Các phương pháp phương pháp đánh giá hiệu năng nén


### **0.3.2. Phạm vi nghiên cứu**

- Tập trung sử dụng bộ dữ liệu tự tạo của các sinh viên trong trường đại học Công nghiệp Hà Nội
  - Nghiên cứu tập trung chủ yếu các kĩ thuật nén có mất mát dữ liệu.
  - Tìm kiếm các phương pháp, kĩ thuật nén liên quan đến học máy, học sâu để tận dụng khả năng của phần cứng.
-

## 0.4. Bảng các ký hiệu

Các ký hiệu sử dụng trong sách được liệt kê trong Bảng 0.1.

Bảng 0.1: Các quy ước ký hiệu và tên gọi được sử dụng trong báo cáo

Ký hiệu	Ý nghĩa
$x, y, N, k$	in nghiêng, thường hoặc hoa, là các số vô hướng
$\mathbf{x}, \mathbf{y}$	in đậm, chữ thường, là các vector
$\mathbf{X}, \mathbf{Y}$	in đậm, chữ hoa, là các ma trận
$\mathbb{R}$	tập hợp các số thực
$\mathbb{N}$	tập hợp các số tự nhiên
$\mathbb{C}$	tập hợp các số phức
$\mathbb{R}^m$	tập hợp các vector thực có $m$ phần tử
$\mathbb{R}^{m \times n}$	tập hợp các ma trận thực có $m$ hàng, $n$ cột
$\mathbb{S}^n$	tập hợp các ma trận vuông đối xứng bậc $n$
$\mathbb{S}_+^n$	tập hợp các ma trận nửa xác định dương bậc $n$
$\mathbb{S}_{++}^n$	tập hợp các ma trận xác định dương bậc $n$
$\in$	phần tử thuộc tập hợp
$\exists$	tồn tại
$\forall$	mọi
$\triangleq$	ký hiệu là/bởi. Ví dụ $a \triangleq f(x)$ nghĩa là “ký hiệu $f(x)$ bởi $a$ ”.
$x_i$	phần tử thứ $i$ (tính từ 1) của vector $\mathbf{x}$
$\text{sgn}(x)$	hàm xác định dấu. Bằng 1 nếu $x \geq 0$ , bằng -1 nếu $x < 0$ .
$\exp(x)$	$e^x$
$\log(x)$	logarit <i>tự nhiên</i> của số thực dương $x$
$\underset{x}{\operatorname{argmin}} f(x)$	giá trị của $x$ để hàm $f(x)$ đạt giá trị nhỏ nhất
$\underset{x}{\operatorname{argmax}} f(x)$	giá trị của $x$ để hàm $f(x)$ đạt giá trị lớn nhất
o.w	<i>otherwise</i> – trong các trường hợp còn lại
$\frac{\partial f}{\partial x}$	đạo hàm của hàm số $f$ theo $x \in \mathbb{R}$
$\nabla_{\mathbf{x}} f$	gradient của hàm số $f$ theo $\mathbf{x}$ ( $\mathbf{x}$ là vector hoặc ma trận)
$\nabla_{\mathbf{x}}^2 f$	gradient bậc hai của hàm số $f$ theo $\mathbf{x}$ , còn được gọi là <i>Hesse</i>
$\odot$	Hadamard product (elementwise product). Phép nhân từng phần tử của hai vector hoặc ma trận cùng kích thước.
$\propto$	tỉ lệ với
	đường nét liền
	đường nét đứt
	đường nét chấm (đường chấm chấm)
	đường chấm gạch
	nền chấm
	nền sọc chéo

---

# Nghiên cứu tổng quan

---

## 1.1. Các phương pháp nghiên cứu

Hiện nay nén dữ liệu được chia thành 2 loại:

- Nén không mất mát thông tin
- Nén có mất mát thông tin

## 1.2. Ưu nhược điểm của các phương pháp

### 1.2.1. Nén không mất mát thông tin:

Các thuật toán nén không mất dữ liệu thường dựa trên giả thuyết dư thừa trong dữ liệu và thể hiện dữ liệu chính xác hơn mà không mất các thông tin. Nén mà không làm mất dữ liệu là khả thi vì tất cả các dữ liệu thực tế đều có dư thừa. Ví dụ một hình ảnh có thể có các vùng màu sắc không thay đổi trong nhiều pixel. Thay vì ghi nhận từng pixel như đỏ, đỏ, đỏ... dữ liệu có thể được ghi là 279 điểm ảnh đỏ liên tiếp. Đây là một ví dụ về run-length encoding; ngoài ra còn có rất nhiều giải thuật khác.

Dựa theo mức áp dụng thuật toán nén người ta chia nén thành các dạng sau:

- Nén tệp tin: Đây là dạng thức nén truyền thống và thuật toán nén được áp dụng cho từng tệp tin riêng lẻ. Tuy vậy nếu 2 tệp tin giống nhau thì vẫn được nén 2 lần và được ghi 2 lần. Chỉ các byte trùng lặp trong 1 file được loại trừ
-



để giảm kích thước. Tùy dữ liệu nhưng thông thường khả năng giảm sau khi nén chỉ từ 2-3 lần.

- Loại trừ trùng lặp file: Đây là dạng thức nén mà thuật toán nén được áp dụng cho nhiều tập tin. Các file giống hệt nhau sẽ chỉ được lưu một lần. Ví dụ một thư điện tử có tệp tin đính kèm được gửi cho 1000 người. Chỉ có một bản đính kèm được lưu và vì vậy có thể giảm khá nhiều. Thông thường có thể giảm từ 5-10 lần so với dữ liệu gốc. - Loại trừ trùng lặp ở mức sub-file: Đây là một dạng thức kết hợp cả nén tệp tin và loại trừ trùng lặp

### **1.2.2. Nén có mất mát dữ liệu:**

Nén mất dữ liệu giảm số lượng bit bằng cách xác định các thông tin không cần thiết và loại bỏ chúng.

Chuẩn nén tín hiệu số gồm có các chuẩn sau:

- Chuẩn MJPEG: Đây là một trong những chuẩn cổ nhất mà hiện nay vẫn sử dụng. MJPEG (Morgan JPEG). Chuẩn này hiện chỉ sử dụng trong các thiết bị DVR rẻ tiền, chất lượng thấp. Không những chất lượng hình ảnh kém, tốn tài nguyên xử lý, cần nhiều dung lượng ổ chứa, và còn hay làm lỗi đường truyền.
  - Chuẩn MPEG2: Chuẩn MPEG là một chuẩn thông dụng. Đã được sử dụng rộng rãi trong hơn một thập kỷ qua. Tuy nhiên, kích thước file lớn so với những chuẩn mới xuất hiện gần đây, và có thể gây khó khăn cho việc truyền dữ liệu.
  - Chuẩn MPEG-4: Mpeg-4 là chuẩn cho các ứng dụng MultiMedia. Mpeg-4 trở thành một tiêu chuẩn cho nén ảnh kỹ thuật truyền hình số, các ứng dụng về đồ họa và Video tương tác hai chiều (Games, Videoconferencing) và các ứng dụng Multimedia tương tác hai chiều (World Wide Web hoặc các ứng dụng nhằm phân phát dữ liệu Video như truyền hình cáp, Internet Video...). Mpeg-4 đã trở thành một tiêu chuẩn công nghệ trong quá trình sản xuất, phân phối và truy cập vào các hệ thống Video. Nó đã góp phần giải quyết vấn đề về dung lượng cho các thiết bị lưu trữ, giải quyết vấn đề về băng thông của đường truyền tín hiệu Video hoặc kết hợp cả hai vấn đề trên.
-

### 1.2.3. Kết luận

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 bộ mã hóa tự động - đặc biệt là trên các hình ảnh nhỏ - và mạng nơ-ron đã đạt được kết quả tiên tiến nhất trong việc nén hình ảnh không mất dữ liệu.

Bộ mã hóa tự động (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.

Vậy nên chúng em đã quyết định chọn bộ mã hóa tự động là đối tượng nghiên cứu chính cho chủ đề này

---

# Cơ sở lý thuyết

---

## 2.1. Tổng quan về các kỹ thuật nén mất mát thông tin

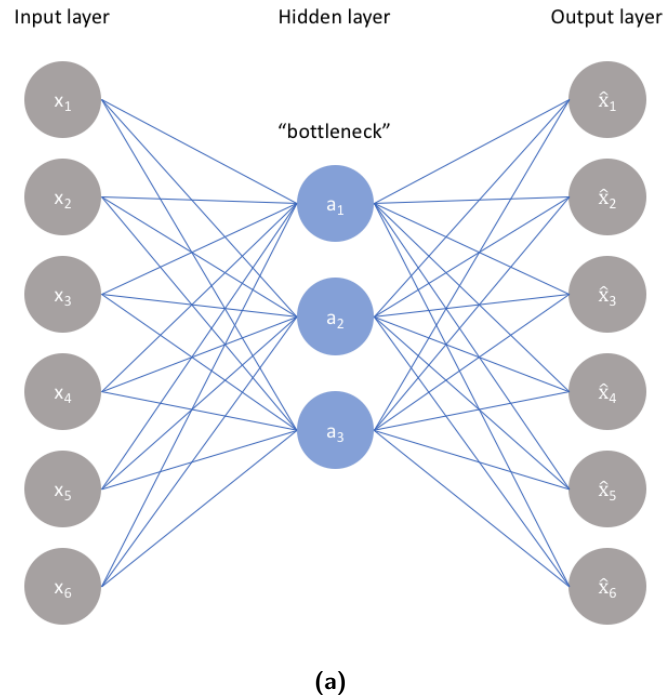
## 2.2. Các kỹ thuật nén codec thường được sử dụng

## 2.3. Bộ mã hóa tự động (Autoencoder)

Bộ mã hóa tự động là một kỹ thuật học tập không có giám sát, trong đó chúng em tận dụng mạng nơ-ron cho nhiệm vụ của học để biểu diễn các tập giá trị dưới dạng nén, học cách để giải mã dữ liệu từ dạng nén.

Cụ thể, chúng em sẽ thiết kế một kiến trúc mạng nơ-ron nhân tạo sau đó áp đặt một nút thắt cổ chai trong mạng - điều này đại diện cho sự nén lại một cách tự động. Mạng này sẽ phải biểu diễn tri thức đầu vào dưới dạng các biểu diễn trong ít chiều không gian hơn, đây chính là biểu diễn nén của đầu vào.

---



**Hình 2.1.** Bộ mã hóa tự động.

Nếu các tính năng đầu vào từng độc lập của nhau, việc nén này và tái tạo sau đó sẽ là một nhiệm vụ rất khó khăn. Tuy nhiên, nếu một số loại cấu trúc tồn tại trong dữ liệu (ví dụ như: mối tương quan giữa các tính năng đầu vào), cấu trúc này có thể được học và do đó được tận dụng khi buộc đầu vào thông qua nút thắt cổ chai của mạng.

Bộ mã hóa tự động lý tưởng cân bằng những điều sau đây:

- Nhạy cảm với các yếu tố đầu vào đủ để xây dựng lại một cách chính xác.
- Đủ nhạy cảm với các đầu vào mà mô hình không chỉ đơn giản là ghi nhớ hoặc trang bị quá nhiều dữ liệu đào tạo.

Sự đánh đổi này buộc mô hình chỉ duy trì các biến thể trong dữ liệu cần thiết để cấu trúc lại đầu vào mà không giữ lại các phần dư thừa trong đầu vào. Đối với hầu hết các trường hợp, điều này liên quan đến việc xây dựng một hàm mất mát trong đó phải thỏa mãn mô hình của chúng ta nhạy cảm với các yếu tố đầu vào (ví dụ: xây dựng lại 1 hàm mất mát  $\mathcal{L}(x, \hat{x})$  và thêm một chính quy hóa)

$$\mathcal{L}(x, \hat{x}) + \text{regularizer} \quad (2.1)$$

Thông thường, sẽ có thêm một tham số tỷ lệ trước thuật ngữ chính quy để chúng ta có thể điều chỉnh sự cân bằng giữa hai mục tiêu.

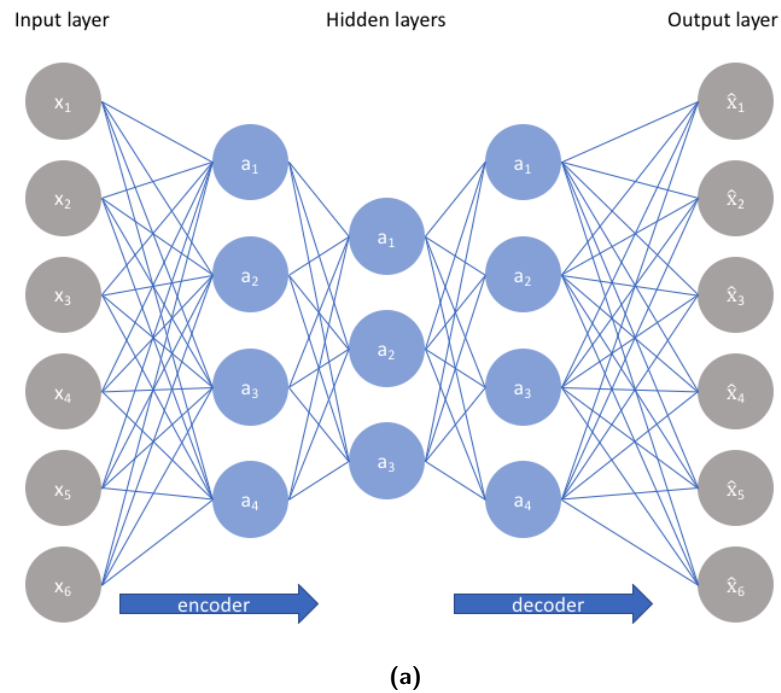
Dưới đây chúng em sẽ trình bày về một số kiến trúc của bộ mã hóa tự động tiêu chuẩn để áp đặt 2 ràng buộc này và điều chỉnh sự cân bằng.

### 2.3.1. Bộ mã hóa tự động chưa hoàn chỉnh

Kiến trúc đơn giản nhất để xây dựng bộ mã tự động là hạn chế số lượng nút hiện diện trong (các) lớp ẩn của mạng, hạn chế lượng thông tin có thể truyền qua mạng. Bằng cách sử dụng các hình phạt mạng theo lỗi xây dựng lại, mô hình của chúng tôi có thể tìm hiểu các thuộc tính quan trọng nhất của dữ liệu đầu vào và cách tái tạo tốt nhất dữ liệu đầu vào ban đầu từ trạng thái "được mã hóa". Lý tưởng nhất là bảng mã này sẽ tìm hiểu và mô tả các thuộc tính tiềm ẩn của dữ liệu đầu vào.

Bởi vì mạng nơ-ron có khả năng học các mối quan hệ phi tuyến, điều này có thể được coi là một sự tổng quát hóa (phi tuyến) mạnh mẽ hơn của PCA (kỹ thuật giảm chiều dữ liệu tuyến tính)

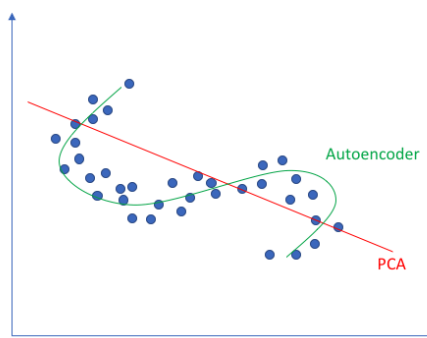
---



**Hình 2.2.** Mô tả mô hình bộ mã hóa tự động chưa hoàn chỉnh.

Trong khi PCA cố gắng khám phá một siêu phẳng có chiều thấp hơn mô tả dữ liệu ban đầu, thì các bộ mã tự động có khả năng học các đa tạp phi tuyến (đa tạp được định nghĩa theo thuật ngữ đơn giản là liên tục, không giao nhau bề mặt). Sự khác biệt giữa hai cách tiếp cận này được hình dung bên dưới.

Linear vs nonlinear dimensionality reduction



**Hình 2.3.** Mô tả mô hình bộ mã hóa tự động chưa hoàn chỉnh.

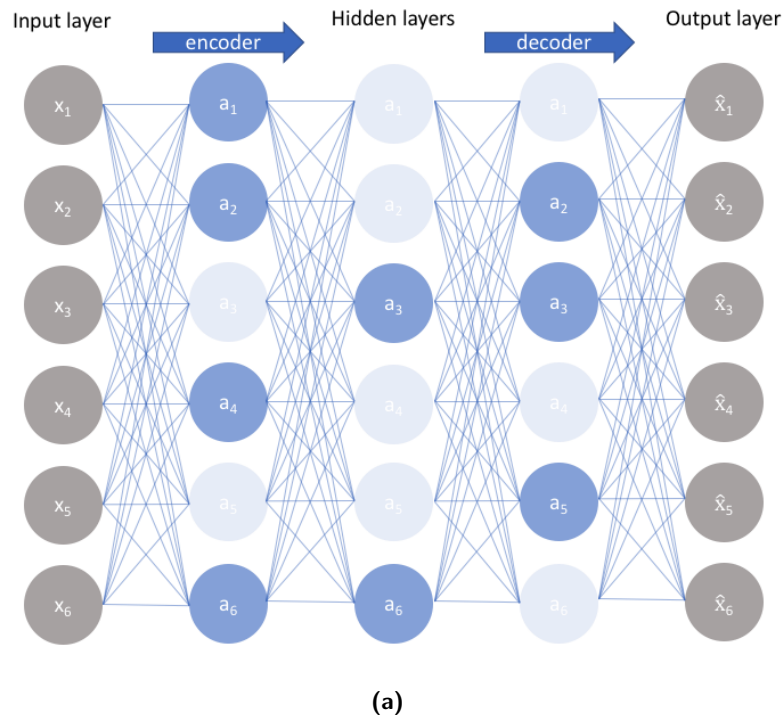
Một bộ mã tự động chưa hoàn chỉnh không có thuật ngữ chính quy rõ ràng - chúng tôi chỉ đào tạo mô hình của mình theo sự mất mát khi xây dựng lại. Do đó, cách duy

nhất của chúng tôi để đảm bảo rằng mô hình không ghi nhớ dữ liệu đầu vào là đảm bảo rằng chúng tôi đã hạn chế đủ số lượng các nút trong (các) lớp ẩn.

Đối với các bộ mã tự động sâu, chúng ta cũng phải lưu ý về dung lượng của bộ mã hóa và các kiểu máy giải mã của chúng ta. Ngay cả khi "lớp nút cổ chai" chỉ là một nút ẩn, mô hình của chúng tôi vẫn có thể ghi nhớ dữ liệu huấn luyện với điều kiện là các mô hình bộ mã hóa và giải mã có đủ khả năng để học một số chức năng tùy ý có thể ánh xạ dữ liệu thành một chỉ mục.

### 2.3.2. Bộ mã tự động thưa thớt

Các bộ mã tự động thưa thớt cung cấp cho chúng ta một phương pháp thay thế để giới thiệu một nút thắt cổ chai thông tin mà không yêu cầu giảm số lượng nút ở các lớp ẩn của chúng ta. Thay vào đó, chúng tôi sẽ xây dựng hàm mất mát của chúng tôi để chúng tôi xử phạt các hàm kích hoạt trong một lớp. Đối với bất kỳ quan sát nhất định nào, chúng ta sẽ khuyến khích mạng của mình học cách mã hóa và giải mã chỉ dựa vào việc kích hoạt một số lượng nhỏ nơ-ron.



**Hình 2.4.** Mô tả mô hình bộ mã hóa tự động chưa hoàn chỉnh.

Một bộ mã tự động thừa thớt chung được hiển thị bên trên nơi độ mờ của một nút tương ứng với mức độ kích hoạt. Điều quan trọng cần lưu ý là các nút riêng lẻ của một mô hình được đào tạo kích hoạt phụ thuộc vào dữ liệu, các đầu vào khác nhau sẽ dẫn đến việc kích hoạt các nút khác nhau thông qua mạng.

Một kết quả của thực tế này là chúng tôi cho phép mạng của mình nhạy cảm với các nút lớp ẩn riêng lẻ đối với các thuộc tính cụ thể của dữ liệu đầu vào. Trong khi một bộ mã tự động chưa hoàn chỉnh sẽ sử dụng toàn bộ mạng cho mỗi lần quan sát, một bộ mã tự động thừa thớt sẽ buộc phải kích hoạt có chọn lọc các vùng của mạng tùy thuộc vào dữ liệu đầu vào. Do đó, chúng tôi đã giới hạn khả năng ghi nhớ dữ liệu đầu vào của mạng mà không giới hạn khả năng mạng trích xuất các tính năng từ dữ liệu.

Điều này cho phép chúng tôi xem xét biểu diễn trạng thái tiềm ẩn và quy định của mạng một cách riêng biệt, do đó chúng tôi có thể chọn biểu diễn trạng thái tiềm ẩn (tức là kích thước mã hóa) phù hợp với những gì có ý nghĩa với ngữ cảnh của dữ liệu trong khi áp đặt chính quy bởi ràng buộc thừa thớt.

Có hai cách chính mà chúng ta có thể áp đặt hạn chế thừa thớt này; cả hai đều liên quan đến việc đo lường các kích hoạt lớp ẩn cho mỗi lô đào tạo và thêm một số thuật ngữ vào hàm mất mát để xử phạt các kích hoạt quá mức. Các cách này là:

- **L1 Regularization:** Chúng ta có thể thêm một thuật ngữ vào hàm mất mát để phạt giá trị tuyệt đối của vector kích hoạt  $a$  trong lớp  $h$  với quan sát  $i$ , được chia tỉ lệ bằng 1 tham số điều chỉnh  $\lambda$

$$\mathcal{L}(x, \hat{x}) + \lambda \sum_i |a_i^{(h)}| \quad (2.2)$$

- **KL-phân kỳ** Về bản chất, KL-phân kỳ là thước đo sự khác biệt giữa hai phân phối xác suất. Chúng ta có thể xác định tham số thừa thớt  $p$  biểu thị kích hoạt trung bình của một nơ-ron trên một tập hợp các mẫu. Kỳ vọng này có thể được tính là :

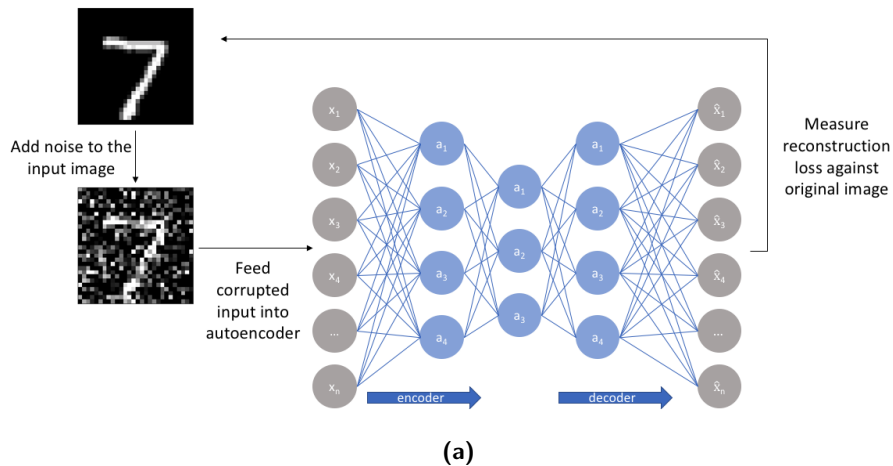
$$\hat{\rho}_j = \frac{1}{m} \sum_i [a_i^{(h)}(x)] \quad (2.3)$$

trong đó chỉ số con  $i$  biểu thị nơ-ron cụ thể trong lớp  $h$ , tính tổng các kích hoạt cho các quan sát huấn luyện  $m$  được ký hiệu riêng lẻ là  $x$ .



### 2.3.3. Bộ mã hóa tự động giảm nhiễu

Như ở phần giới thiệu, bộ mã hóa tự động chính là một mạng nơ-ron được đào tạo trong đó đầu vào giống hệt đầu ra và mô hình có nhiệm vụ tái tạo đầu vào càng chặt chẽ càng tốt khi chuyển qua thông tin ở lớp thất cổ chai. Một cách tiếp cận khác hướng tới việc phát triển một mô hình tổng quát hóa là làm nhiễu một chút dữ liệu đầu vào nhưng vẫn duy trì dữ liệu không bị gián đoạn làm đầu ra mục tiêu của mô hình.



**Hình 2.5.** Mô tả mô hình bộ mã hóa tự động chưa hoàn chỉnh.

Với cách tiếp cận này, mô hình không thể đơn giản phát triển một ánh xạ ghi nhớ dữ liệu đào tạo vì đầu vào và đầu ra mục tiêu không còn giống nhau. Thay vào đó, mô hình học một trường vectơ để ánh xạ dữ liệu đầu vào tới một đa tạp có chiều thấp hơn (nhớ lại từ hình ảnh trước đây của tôi rằng một đa tạp mô tả vùng mật độ cao nơi dữ liệu đầu vào tập trung).

# Triển khai chương trình và đánh giá hiệu suất

---

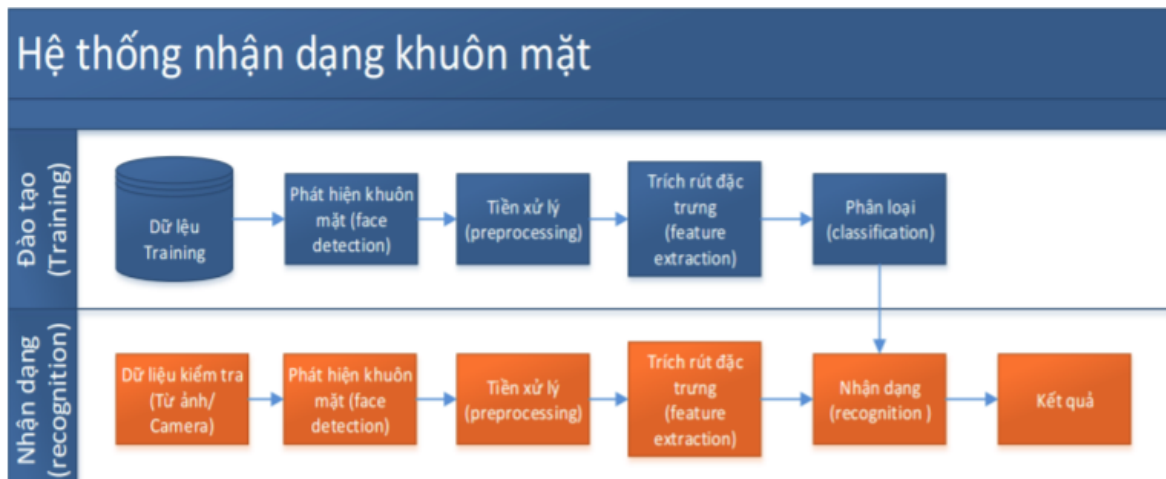
## 3.1. Phân tích

Về cơ bản một hệ thống điểm danh bằng khuôn mặt gồm các bước sau:

- Thu thập dữ liệu khuôn mặt
- Phát hiện khuôn mặt dựa trên ảnh đầu vào và gán nhãn dữ liệu
- Làm giàu dữ liệu
- Trích xuất các đặc trưng (sử dụng học sâu)
- Đưa các đặc trưng đã được gán nhãn vào thuật toán phân loại
- Lưu trữ các thông tin và kết quả phân loại đã được học
- Nhận dạng khuôn mặt và tiến hành điểm danh

Hệ thống được chia thành 2 quy trình:

- Người dùng đăng ký khuôn mặt của mình
  - Hệ thống tiến hành nhận dạng và điểm danh
-



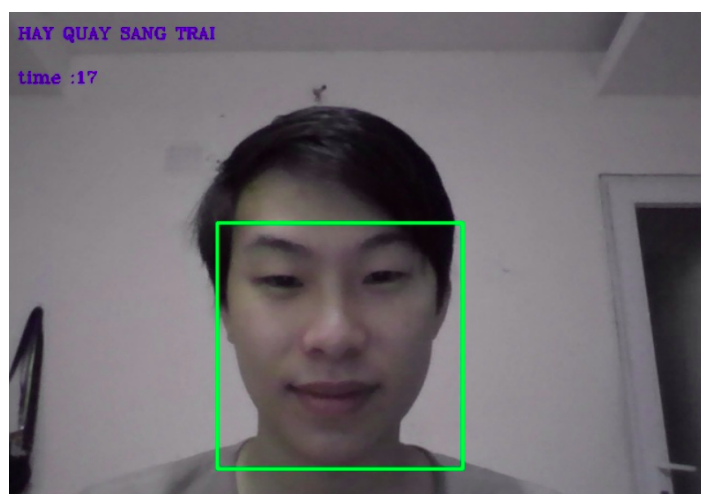
**Hình 3.1.** Tổng quan hệ thống

## 3.2. Xây dựng

### 3.2.1. Thu thập dữ liệu khuôn mặt

Hệ thống thu thập hình ảnh dữ liệu khuôn mặt bằng cách sử dụng chính webcam của máy tính, hoặc có thể là hình ảnh từ nhiều nguồn khác. Các ảnh được thu thập cần đảm bảo các yếu tố như điều kiện ánh sáng, các góc độ khác nhau của khuôn mặt, tuổi tác,... Và khuôn mặt không nên có các vật cản như kính.

Hệ thống có thể gợi ý người dùng quay mặt sang trái hay phải để có thể thu thập được nhiều góc mặt nhất có thể



**Hình 3.2.** Thu thập dữ liệu khuôn mặt người dùng

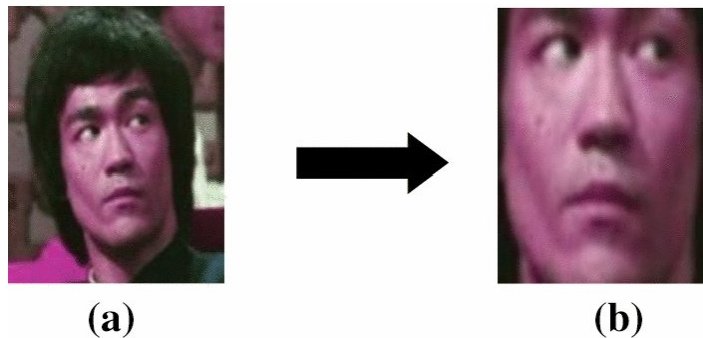
Ngoài ra, để đảm bảo độ chính xác cho hệ thống, đối với mỗi người dùng cần thu thập một số lượng ảnh nhất định không quá ít, và mỗi bức ảnh chỉ chứa duy nhất một khuôn mặt.

Bộ dữ liệu tôi sử dụng trong dự án này gồm 400 ảnh của 10 sinh viên. Với số lượng ảnh của mỗi sinh viên là khác nhau dao động từ 30 đến 40 ảnh cho mỗi sinh viên.

### 3.2.2. Phát hiện khuôn mặt và gán nhãn dữ liệu

Để trích chọn đặc trưng cho mỗi khuôn mặt, trước tiên ta cần tìm ra vị trí khuôn mặt trong bức hình. Vì bộ dữ liệu sẽ bao gồm nhiều ảnh có điều kiện ánh sáng cũng như các góc độ của khuôn mặt khác nhau, chính vì vậy việc lựa chọn máy dò khuôn mặt cũng rất quan trọng để đảm bảo hiệu quả cao nhất cho hệ thống.

Tôi sử dụng MTCNN thực hiện công việc này và tiến hành gán nhãn dữ liệu, yêu cầu người dùng nhập tên.



**Hình 3.3.** MTCNN lấy khuôn mặt của người dùng

### 3.2.3. Làm giàu dữ liệu

Dữ liệu là nguồn sống của học máy và học sâu, nhận thấy sự quan trọng của dữ liệu trong các mô hình học sâu nên các phương pháp làm giàu dữ liệu đã ra đời và đóng góp rất nhiều vào sự thành công của các mô hình học sâu.

Ở trong dự án này tôi cũng sử dụng một số phương pháp làm giàu dữ liệu phổ biến sử dụng các thuật toán xử lý ảnh:

- Chuẩn hóa theo phân phối chuẩn các pixels của ảnh

- Tạo ảnh với các góc nghiêng 20 độ
- Dịch chuyển ảnh theo chiều dài và chiều rộng
- Lật ảnh theo chiều ngang



**Hình 3.4.** Dữ liệu đã được làm giàu

#### 3.2.4. Trích chọn các đặc trưng ảnh khuôn mặt

Trong hệ thống này tôi sử dụng 1 mô hình có sẵn với mạng cơ sở là InceptionResnetV1 được huấn luyện trong tập dữ liệu với hàng triệu ảnh khuôn mặt khác nhau trong đó có cả người Việt Nam.

$$f\left(\begin{array}{c} \text{[Image of a man]} \end{array}\right) = \begin{pmatrix} 0.112 \\ 0.067 \\ 0.091 \\ 0.129 \\ 0.002 \\ 0.012 \\ 0.175 \\ \vdots \\ 0.023 \end{pmatrix}$$

**Hình 3.5.** Mô tả nhúng ảnh khuôn mặt thành các vector đặc trưng

Bộ dữ liệu khuôn mặt sẽ được chia theo từng thư mục tương ứng với hình ảnh của từng đối tượng (sinh viên). Hệ thống sẽ tiến hành quét qua toàn bộ ảnh trong các thư

mục. máy dò khuôn mặt sẽ tìm kiếm khuôn mặt có trong ảnh (mặc định mỗi ảnh sẽ chỉ chứa một khuôn mặt), cắt lấy khuôn mặt và đưa kích thước về 160x160 pixel.

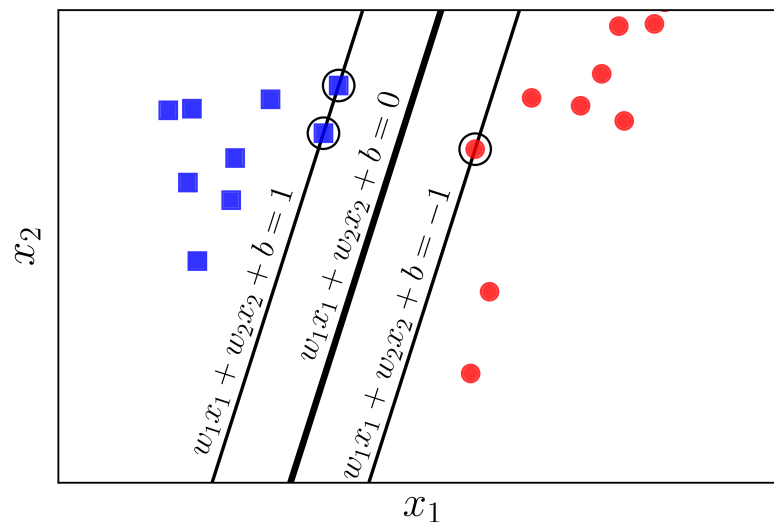
Sau đó FaceNet sẽ tiến hành trích rút đặc trưng của từng khuôn mặt, áp dụng mô hình học với thuật toán hàm đánh giá bộ ba và gán nhãn cho từng khuôn mặt (nhãn sẽ được lấy theo tên thư mục chứa ảnh).

### 3.2.5. Đưa các vector đặc trưng vào mô hình phân loại

Sau khi đã có các vector đặc trưng của các khuôn mặt, tôi sẽ đưa các vector này vào một mô hình để thuật toán có thể học được cách phân loại các đối tượng đã đăng ký.

Mô hình thuật toán phân loại mà tôi sử dụng là thuật toán SVM (Support vector machine)

Sau khi thuật toán SVM đã được huấn luyện để phân loại xong thì tiến hành lưu kết quả huấn luyện cùng với các vector và nhãn đã đào tạo lại, nhằm mục đích huấn luyện lại mô hình khi có thêm khuôn mặt mới đăng ký.



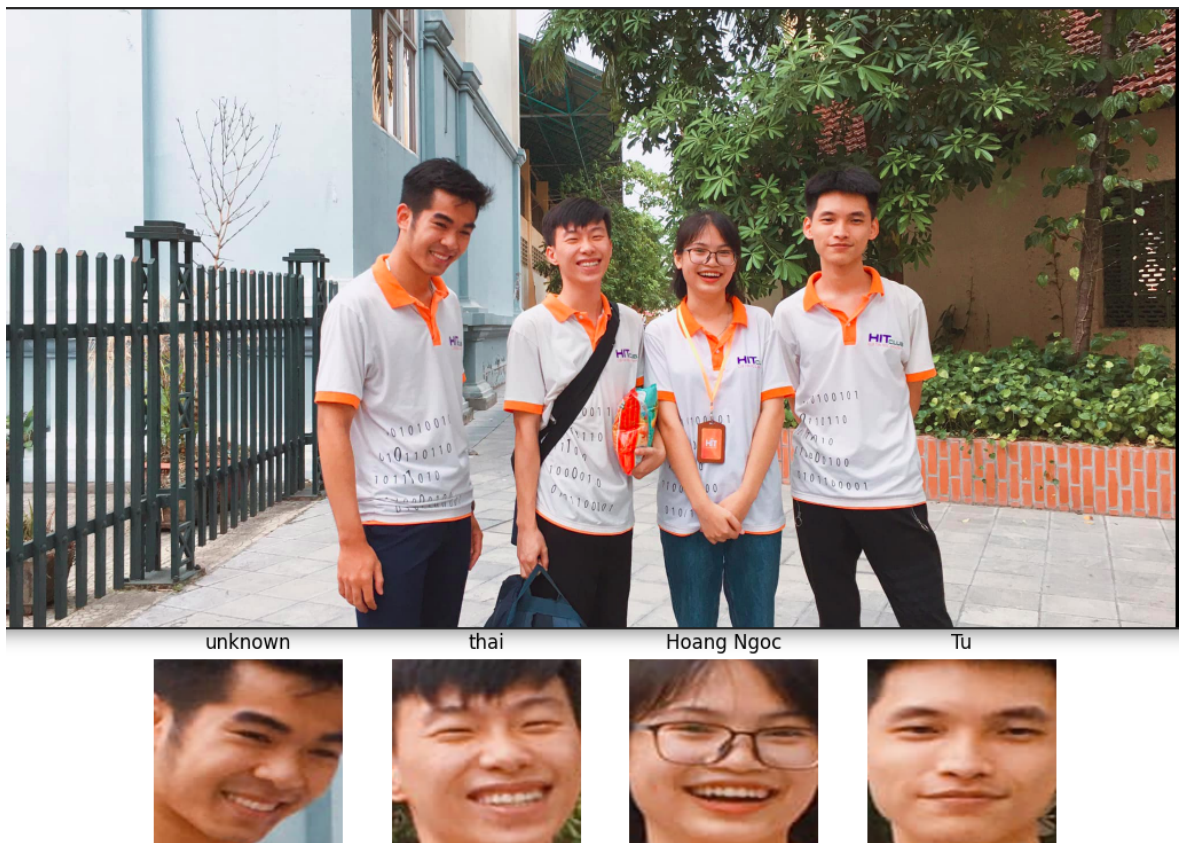
**Hình 3.6.** Mô tả cơ chế phân loại của SVM

### 3.2.6. Nhận diện khuôn mặt và tiến hành điểm danh

Khi hệ thống đã thực hiện huấn luyện xong các mô hình học sâu, tôi tiến hành thử nghiệm với một số ảnh có các khuôn mặt đã đăng ký và chưa đăng ký.

Hệ thống sẽ dò tìm các khuôn mặt trong ảnh, sau đó thực hiện việc mã hóa các khuôn mặt này thành các vector đặc trưng rồi đưa vào các mô hình phân loại đã được huấn luyện.

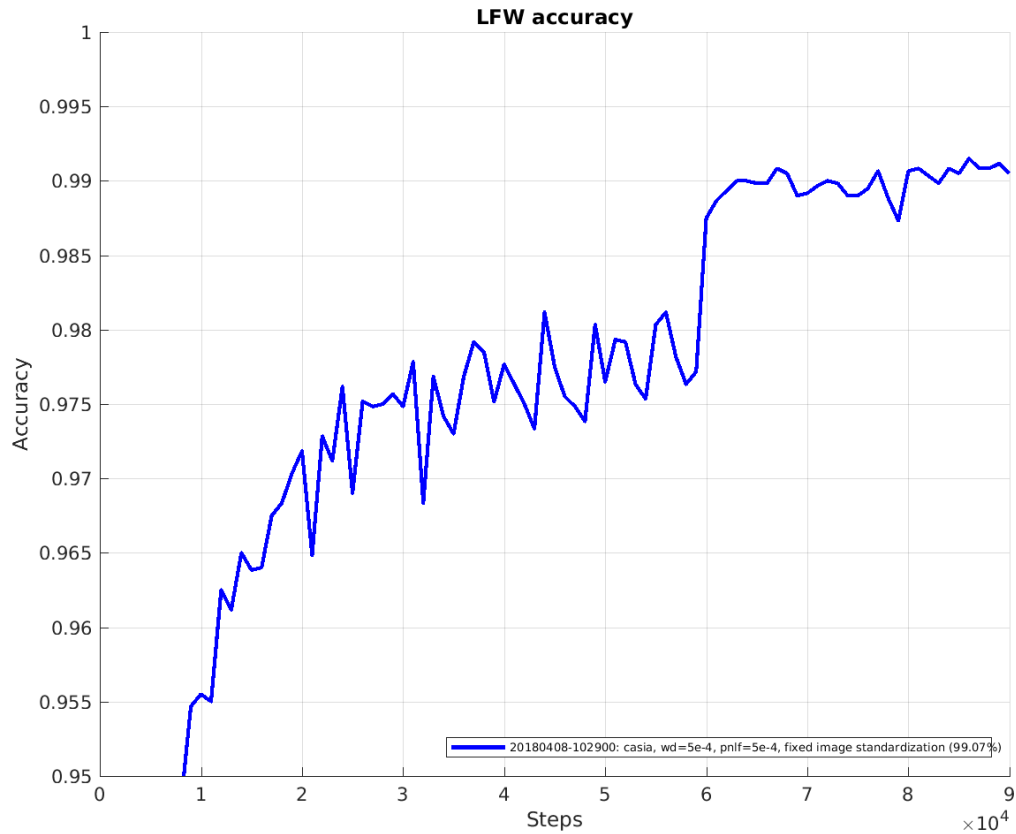
Kết quả cuối cùng hệ thống sẽ đưa ra hình ảnh các khuôn mặt và kèm theo các tên của khuôn mặt đó nếu khuôn mặt này đã được đăng ký, ngược lại hệ thống sẽ đưa ra "unknown face" nếu khuôn mặt này chưa xuất hiện trong tập dữ liệu đã đăng ký



**Hình 3.7.** Hệ thống tiến hành điểm danh

### 3.2.7. Kết quả thử nghiệm

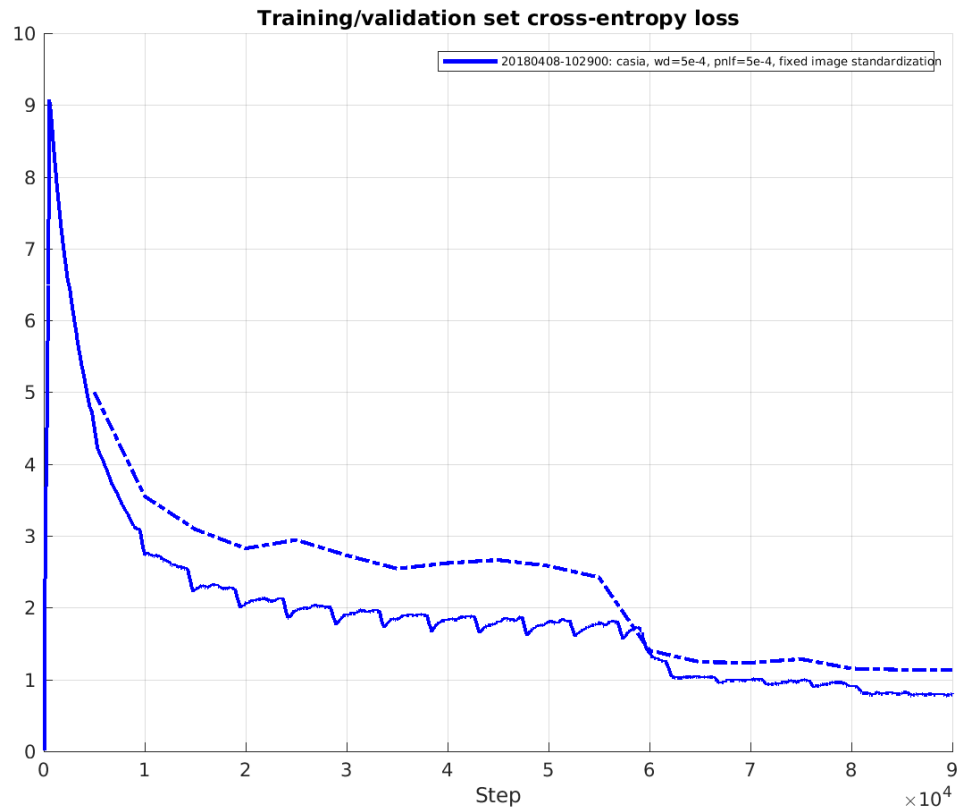
#### *Độ chính xác của mô hình facenet*



**Hình 3.8.** Độ chính xác trên tập dữ liệu LFW

Ở đây có thể thấy rằng ngay cả khi độ chính xác trong lần đánh giá cuối cùng là 0,9965 thì độ chính xác trung bình cho 10 lần đánh giá cuối cùng thấp hơn một chút (0,9950), có lẽ gần với những gì người ta có thể mong đợi khi tái tạo kết quả. Mức trung bình có lẽ là một số liệu tốt hơn để sử dụng khi so sánh các cài đặt siêu tham số khác nhau.





**Hình 3.9.** Đánh giá mất mát bằng Entropy chéo

Hình này cho thấy sự mất mát entropy chéo trong quá trình huấn luyện (đường liền nét) và xác nhận (đường đứt nét). Bộ xác nhận bao gồm khoảng 20000 hình ảnh và đánh giá được thực hiện sau mỗi 5 kỷ nguyên. Entropy chéo trong quá trình đào tạo được ghi lại ở mỗi bước đào tạo nhưng đã được lọc bằng bộ lọc trung bình trượt trên 500 bước. Từ điều này, rõ ràng là mô hình có lợi hơn một chút vì vậy có thể tốt nếu thử nghiệm thêm một chút với xác suất phân rã trọng lượng L2 và xác suất bỏ học.

***Hiệu suất của hệ thống***

Khi sử dụng các phương pháp đo độ chính xác trên dữ liệu thức tế mà tôi tự thu thập từ những người bạn, sinh viên trong khoa, trường. Thì độ chính xác đo được là 96.4%



**Hình 3.10.** Dữ liệu tự thu thập

---

# Kết luận và hướng phát triển

---

## 4.1. Kết luận

Trên cơ sở tìm hiểu về bài toán nhận diện mặt người trong ảnh, sử dụng pre-trained model FaceNet được huấn luyện trước trên mạng cơ sở là InceptionResnetV1 do tiến sĩ khoa học máy tính David Sandberg cung cấp, tôi đã xây dựng thành công hệ thống điểm danh thông qua hình ảnh khuôn mặt.

Về khả năng phát hiện khuôn mặt, kết quả phát hiện khá tốt hầu hết các trường hợp, kể cả trong điều kiện thiếu sáng, góc nghiêng, hay có vật che khuất như kính mắt,...

Về khả năng nhận dạng, hệ thống đạt kết quả từ 96-98% đối với các khuôn mặt thẳng và điều kiện ánh sáng thích hợp, đạt 92-95% đối với các khuôn mặt nghiêng hoặc thiếu sáng.

Về khả năng loại trừ các khuôn mặt “unknown face”, kết quả đạt khoảng 85-90% khuôn mặt lạ được phát hiện trong quá trình thử nghiệm. Hệ thống điểm danh hoạt động ổn định và mượt mà nhờ máy chủ viết bằng Python. Giao diện được xây dựng trên nền Web là một lợi thế vì tính đơn giản và tiện lợi.

## 4.2. Hướng phát triển

Dựa trên những cơ sở sẵn có này hệ thống có thể được cải tiến trong tương lai bằng những phương pháp sau:

---

- Cải thiện thời gian chạy của hệ thống, nâng cấp lên có thể chạy trong thời gian thực
- Để cải thiện độ chính xác cho hệ thống, đầu tiên ta cần cải thiện bộ dữ liệu dựa trên các tiêu chí như tư thế chụp, góc chụp, hạn chế sự che khuất các bộ phận trên mặt, biểu cảm khuôn mặt, điều kiện ánh sáng, tuổi tác...
- Thử nghiệm với nhiều mô hình được huấn luyện trước và thuật toán huấn luyện khác nhau cho bộ dữ liệu của hệ thống.
- Thay thế phương pháp loại bỏ khuôn mặt lạ, thử nghiệm và chọn ra ngưỡng cho phép phù hợp hơn.

Không chỉ dừng lại ở việc điểm danh, của hệ thống nhận dạng khuôn mặt có thể được sử dụng trong các hệ thống mở khóa, thanh toán, hay truy tìm tội phạm,...

---

## Tài liệu tham khảo

---