

# CamoPatch: An Evolutionary Strategy for Generating Camouflaged Adversarial Patches

Le Van Hoang   Nguyen Hoang Hiep   Huynh Anh Dung  
Ha Huy Hoang   Nguyen Duy Hoang

University of Information Technology - VNUHCM  
CS410 - Neural Networks and Genetic Algorithm  
GVHD: Luong Ngoc Hoang

6th June 2025

# Sumary

- ① Introduce
- ② Adversarial Example
- ③ Adversarial Patch
- ④ CamoPatch
- ⑤ Real World

Trong những năm gần đây, trí tuệ nhân tạo (AI) đã đạt được những bước tiến ấn tượng, đặc biệt trong các lĩnh vực như phân loại hình ảnh, nhận diện giọng nói và xử lý ngôn ngữ tự nhiên.

Những tiến bộ này không chỉ thúc đẩy sự phát triển mạnh mẽ của khoa học công nghệ mà còn tạo ra những thay đổi rõ rệt trong cách con người sống, làm việc và tương tác với thế giới xung quanh.

## AI Application

- **Y tế:** AI hỗ trợ chẩn đoán hình ảnh y khoa với độ chính xác vượt trội, giúp phát hiện bệnh tật sớm hơn.
- **Giao tiếp:** Công nghệ nhận diện giọng nói giúp nâng cao chất lượng dịch vụ chăm sóc khách hàng và tự động hóa các tác vụ hành chính.
- **Ngôn ngữ:** Các mô hình AI như ChatGPT đã đạt được khả năng hiểu và tạo ra ngôn ngữ tự nhiên gần giống con người, mở ra nhiều ứng dụng mới.

# AI achievement

Một số hệ thống AI đã vượt qua con người trong các nhiệm vụ cụ thể. Chẳng hạn, vào năm 2016, AlphaGo của DeepMind đã đánh bại Lee Sedol, một trong những kỳ thủ cờ vây hàng đầu, với tỷ số 4-1.

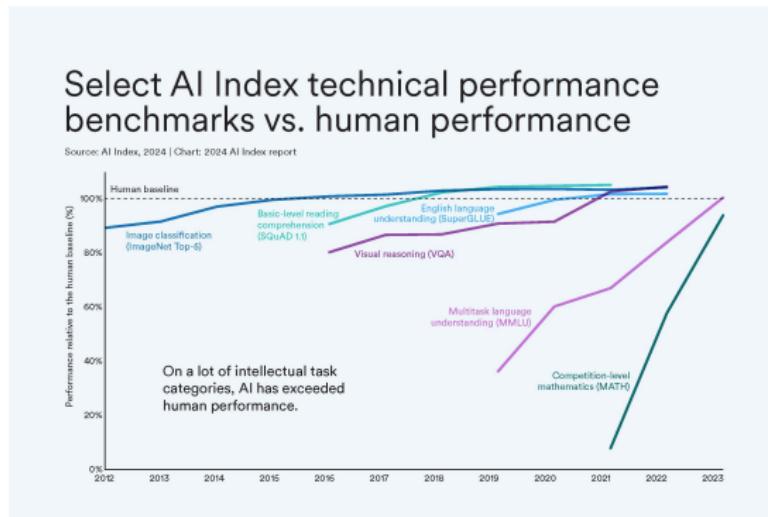


Figure 1: AI outperforms Human on ImageNet

# AI Challenge

Tuy nhiên, liệu AI có thể thay thế hoàn toàn con người?

Dù AI hiện đã được ứng dụng rộng rãi trong nhiều lĩnh vực, nhưng việc triển khai các mô hình AI trong những ngành đòi hỏi mức độ an toàn và độ tin cậy cao như xe tự lái, chăm sóc sức khỏe hay an ninh quốc phòng vẫn gặp nhiều thách thức.

## Những thách thức trong việc áp dụng AI

- AI không thể chịu trách nhiệm pháp lý nếu xảy ra sự cố nghiêm trọng.
- Mối đe dọa từ các *Adversarial Examples*.

# Adversarial Example

## Dịnh nghĩa

Trong *computer vision*, **adversarial examples** là những mẫu dữ liệu đầu vào được tinh chỉnh có chủ đích. Những thay đổi này thường rất nhỏ và khó nhận biết bằng mắt thường, nhưng lại khiến mô hình dự đoán sai một cách tự tin.

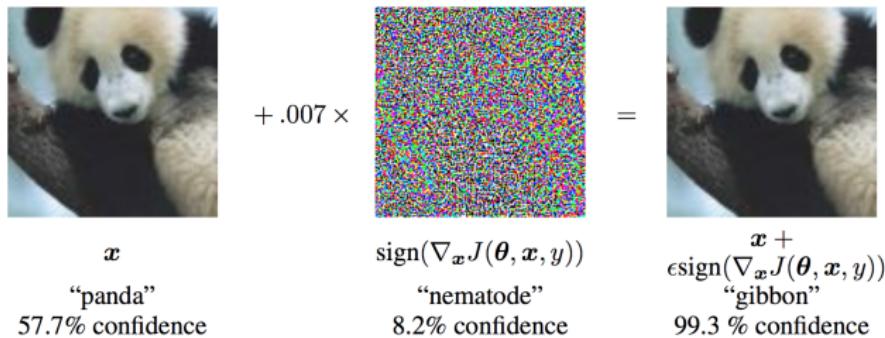


Figure 2: Ví dụ về Adversarial Example trong Computer Vision.

# Adversarial Example

Các cuộc tấn công adversarial có thể được phân loại dựa trên mức độ truy cập vào mô hình nạn nhân:

## White-box Attack

Kẻ tấn công có toàn quyền truy cập vào mô hình:

- Biết kiến trúc mô hình.
- Biết các trọng số và tham số bên trong.
- Có thông tin về gradient của mô hình.

## Black-box Attack

Kẻ tấn công không biết gì về cấu trúc bên trong mô hình:

- Chỉ có thể gửi đầu vào và quan sát đầu ra.
- Mô hình được xem như một "hộp đen".

# Adversarial Patch

Trong khi những nghiên cứu ban đầu tạo mẫu đối nghịch (adversarial example) bằng cách thay đổi pixel trên toàn bộ tấm ảnh trong một giới hạn norm ( $l_2$  or  $l_{inf}$ ) thì một vài nghiên cứu gần đây chuyển sang chỉ thay đổi một phần nhỏ của hình ảnh.

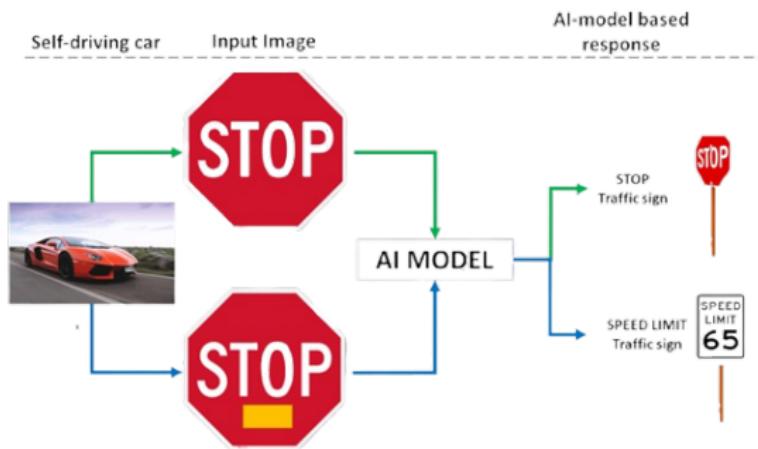


Figure 3: Kĩ thuật adversarial patch

# Adversarial Patch

Kĩ thuật này tỏ ra đặc biệt nguy hiểm trong real world khi chúng có thể được in ra để dán lên các vật thể.



Figure 4: Ví dụ về Adversarial Patch trên biển báo giao thông

# Visual Limitations of Existing Patches

- Hầu hết các miếng vá đổi kháng có hình dạng và màu sắc không tự nhiên, dễ bị con người phát hiện
- Việc thiếu khả năng ngụy trang làm giảm tính ứng dụng thực tế và gây đánh giá sai về độ bền mô hình

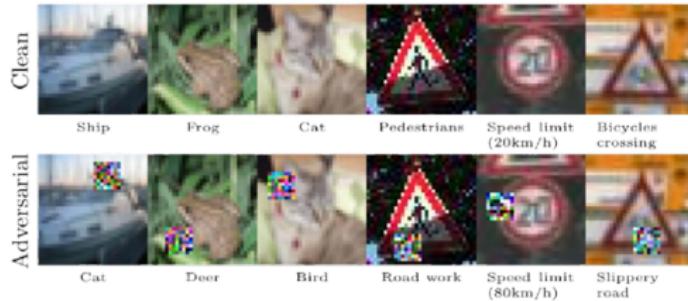


Figure 5: Adversarial Patch khá dễ phát hiện

# CamoPatch

CamoPatch là một kỹ thuật tấn công đối nghịch dạng Black-box, ứng dụng chiến lược tiến hóa để tạo ra các patch có khả năng đánh lừa mô hình một cách hiệu quả, đồng thời duy trì mức độ ngụy trang cao nhằm tránh bị phát hiện.<sup>1</sup>

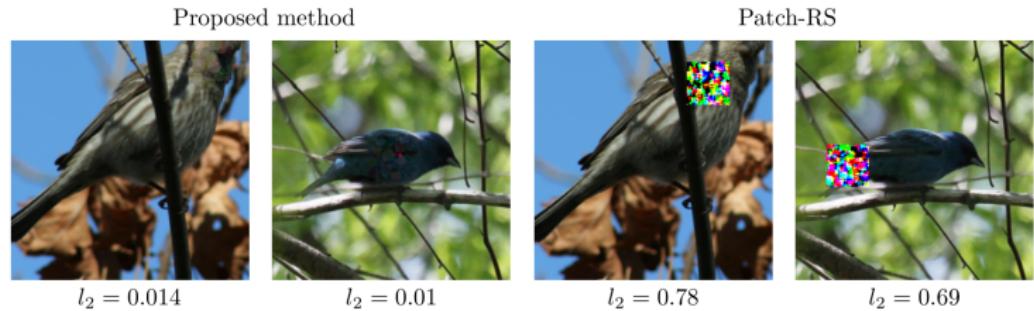


Figure 6: Ví dụ về CamoPatch (trái) và Patch-RS (phải)

---

<sup>1</sup>CamoPatch: An Evolutionary Strategy for Generating Camouflaged Adversarial Patches (NeurIPS 2023)

## Bài toán tối ưu trong CamoPatch

CamoPatch giải quyết một bài toán tối ưu đa mục tiêu nhằm tìm ra bản vá (*patch*) thoả mãn hai điều kiện sau:

- Đánh lừa được mô hình phân loại
- Giữ cho độ khác biệt ( $l_2$  distance) với vùng ảnh gốc bị thay thế là nhỏ nhất có thể

# Low-Poly Art

Low-Poly Art là một phong cách nghệ thuật số đặc trưng bởi việc sử dụng các hình đa giác đơn giản (thường có độ trong suốt bán phần) để xây dựng những tác phẩm hình ảnh độc đáo. Các hình khối này được xếp chồng lên nhau để tạo ra hiệu ứng hình ảnh mong muốn.



**Tham khảo thêm:** Primitive - GitHub Repository

# Low-Poly Art in CamoPatch

Áp dụng ý tưởng low-poly art, chúng tôi tạo ra adversarial patch bằng cách xếp chồng các hình tròn RGB bán trong suốt.



(a) 100 queries



(b) 1000 queries



(c) 5000 queries



(d) 10000 queries

Figure 7: Ví dụ CamoPatch khi cố định vị trí

# Solution Definition

## Biểu diễn một cá thể

Một cá thể (giải pháp) được biểu diễn dưới dạng một ma trận có kích thước  $(N, 7)$ , trong đó  $N$  là số lượng hình tròn.

## Giá trị mỗi hàng

Mỗi hàng trong ma trận chứa 7 giá trị tương ứng với 7 thuộc tính của một hình tròn, tất cả đều đã được chuẩn hóa về khoảng  $[0, 1]$ :

- **Tọa độ tâm:** 2 giá trị
- **Bán kính:** 1 giá trị
- **Màu sắc (RGB):** 3 giá trị
- **Độ trong suốt (Alpha):** 1 giá trị

=> Dùng các thuật toán search để tìm ra cá thể thỏa mãn: **Chiến lược tiến hóa (ES)**

# Evolutionary Strategy (ES)

- Evolutionary Strategy (chiến lược tiến hóa) là một phương pháp trong trí tuệ nhân tạo và tối ưu hóa, bắt chước cách loài sinh vật tiến hóa trong tự nhiên.
- Tạo ra cá thể (giải pháp) mới bằng cách đột biến cá thể hiện tại.

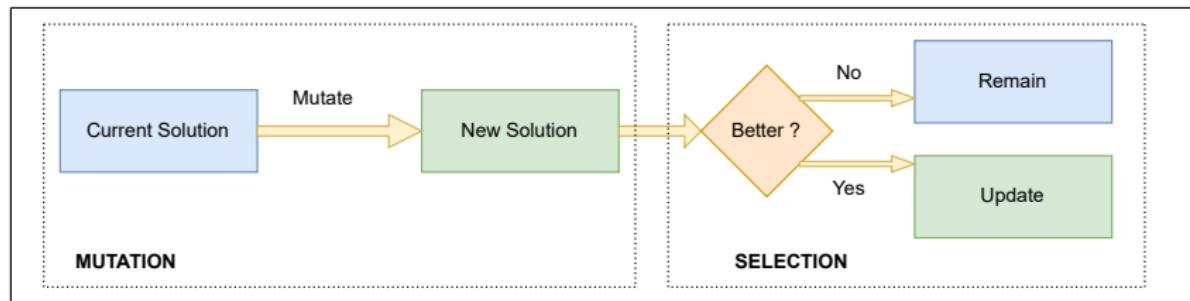


Figure 8: Ví dụ về ES 1+1 trong CamoPatch

## Quy trình đột biến

- Tham số đột biến: **mut = 0.3**
- Ngẫu nhiên chọn một số thuộc tính cần được đột biến
- Ngẫu nhiên chọn một hình tròn/hàng để thực hiện đột biến
- Với các thuộc tính được chọn:
  - Với xác suất **0.3** (tức là bằng giá trị `mut`), gán lại giá trị hoàn toàn ngẫu nhiên
  - Với xác suất **0.7**, cộng thêm một giá trị ngẫu nhiên nhỏ (đã được scale) vào các thuộc tính đã chọn.

# Patch Update (Patch Selection)

Dịnh nghĩa Loss:

$$L(f, x + \delta) = f_{y_t}(x + \delta) - f_{y_q}(x + \delta)$$

Trong đó:

$$y_q = \arg \max_{y \neq y_t} f_y(x + \delta)$$

Khi nào được cập nhật?

- Nếu ảnh dùng(render) cur\_sol và ảnh dùng new\_sol đều đánh lừa được mô hình, và:

$$\text{new\_l2} < \text{cur\_l2} \Rightarrow \text{cập nhật}$$

- Hoặc nếu:

$$\text{new\_L} < \text{cur\_L} \Rightarrow \text{cập nhật} \quad \left\{ \begin{array}{l} \text{new\_L} < 0 \& \text{cur\_L} > 0 \\ \text{new\_L} < \text{cur\_L} (\text{both} \geq 0) \end{array} \right.$$

# Location Mutation và Location Update

## Location Mutation

- Chọn ngẫu nhiên một vị trí mới cho `cur_sol`.

## Location Update

- Cập nhật tương tự patch.
- Áp dụng **Simulated Annealing** để tồn tại xác xuất chấp nhận vị trí mới kể cả khi  $L$  cao hơn.

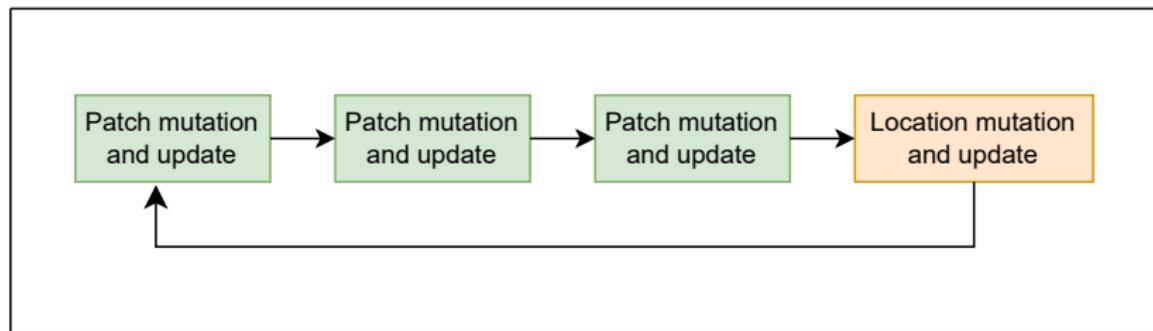


Figure 9: Minh họa về vòng lặp với tham số  $li = 4$

# Real World: Problem

Hàm mất mát trong tấn công:

$$L(f, x + \delta) = f_{y_t}(x + \delta) - f_{y_q}(x + \delta)$$

$$\text{với } y_q = \arg \max_{y \neq y_t} f_y(x + \delta)$$

Thách thức trong thực tế

Hầu hết các mô hình chỉ trả về:

- Nhãn dự đoán cuối cùng ( $\hat{y}$ )
- Cùng với độ tin cậy (confidence) hoặc giá trị *logits* tương ứng với  $\hat{y}$

**Không cung cấp đầy đủ logits của tất cả các lớp**  $\Rightarrow$  Không thể tính trực tiếp  $y_q$  như công thức trên.

# Solution for Real World Problem

Dịnh nghĩa Loss:

$$L(f, x + \delta) = f_{y_q}(x + \delta)$$

Trong đó:

$$y_q = \arg \max f_y(x + \delta)$$

Khi nào được cập nhật?

- Nếu ảnh dùng(render) `cur_sol` và ảnh dùng `new_sol` đều đánh lừa được mô hình, và:

$$\text{new\_l2} < \text{cur\_l2} \Rightarrow \text{cập nhật}$$

- Hoặc nếu ảnh dùng `new_sol` đánh lừa được mô hình (nghĩa là `cur_sol` không đánh lừa được)  $\Rightarrow$  cập nhật
- Nếu cả 2 đều không đánh lừa được mô hình, nghĩa là mô hình đang dự đoán  $y_t$  thì xét:  $\text{new\_L} < \text{cur\_L} \Rightarrow$  cập nhật

# CamoPatch on Sign Classifier

Tiến hành huấn luyện một mạng học sâu trên bộ dữ liệu biển báo giao thông, bao gồm 9 lớp:

- Speed Limit 5 km/h
- Speed Limit 15 km/h
- Speed Limit 30 km/h
- Speed Limit 40 km/h
- Speed Limit 50 km/h
- Speed Limit 60 km/h
- Speed Limit 70 km/h
- Speed Limit 80 km/h
- No Car Allowed

Tổng cộng có 1162 ảnh huấn luyện và 424 ảnh kiểm tra.

# CamoPatch on Sign Classifier

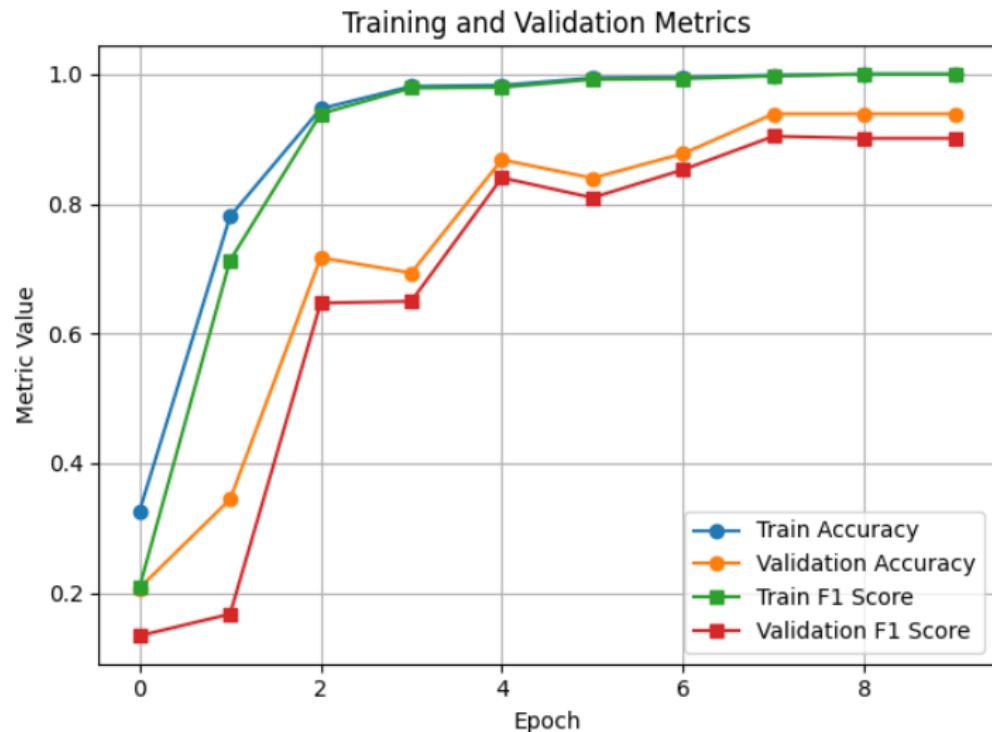


Figure 10: Quá trình huấn luyện bộ phân loại biển báo giao thông

# CamoPatch on Sign Classifier

Sử dụng CamoPatch để tấn công mô hình nhằm kiểm tra tính bền vững của mô hình



Figure 11: Ví dụ về ảnh sau khi bị tấn công

# Experiment

Success rate: Tỉ lệ tấn công thành công (làm mô hình dự đoán sai)

L2 distance: Khoảng cách L2 giữa ảnh gốc và ảnh bị tấn công

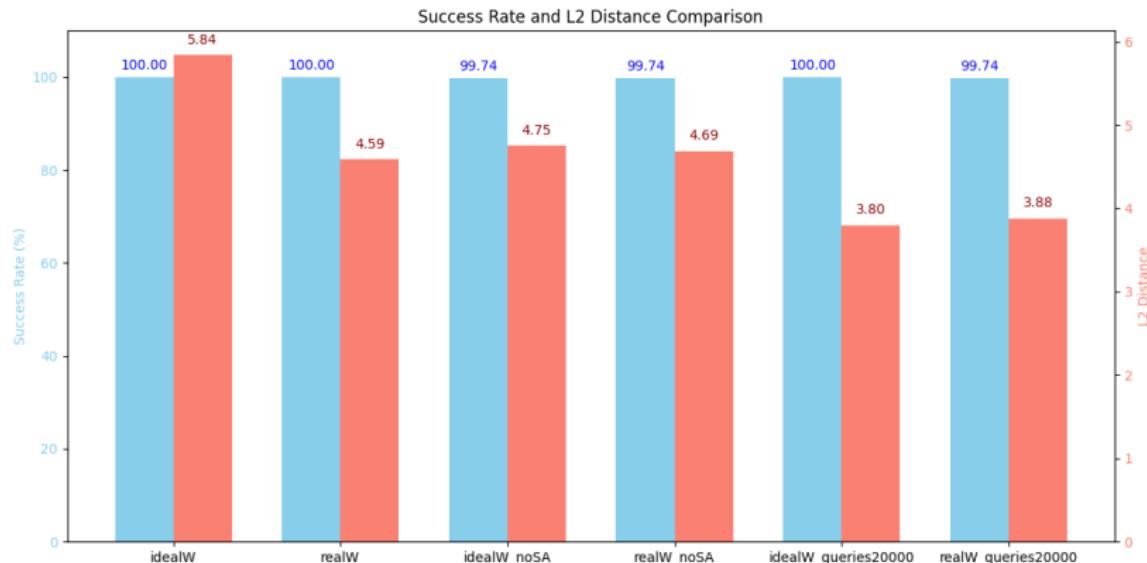


Figure 12: Compare between two settings