

**MÔ HÌNH NHẬN DIỆN 15 LOÀI CÁ CẢNH BẰNG MẠNG NORON TÍCH CHẬP**

***Hà Tấn Phát***

*Trường Đại học Sư phạm Kỹ thuật thành phố Hồ Chí Minh, 1 đường Võ Văn Ngân, Linh Chiểu, thành phố Thủ Đức, thành phố Hồ Chí Minh.*

*Mọi thắc mắc vui lòng liên hệ:* [*19146231@student.hcmute.edu.vn*](mailto:19146231@student.hcmute.edu.vn)

**TÓM TẮT**

*Convolution Neural Network (CNNs – Mạng noron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho người lập trình xây dựng được những hệ thống thông minh với độ chính xác cao. CNN được sử dụng nhiều trong các hệ thống xử lý ảnh. Trong bài báo này sẽ giới thiệu cách để tạo ra một mô hình mạng noron tích chập dùng để nhận diện 15 loài cá cảnh đồng đánh giá chất lượng mô hình đạt được bao gồm các bước như tạo tập dữ liệu, tiền xử lý dữ liệu, xây dựng các lớp tích chập, kiểm chứng và thiết kế website cho người dùng. Mô phỏng được thực hiện trên môi trường lập trình Google Colab dùng để xử lý dữ liệu, tạo lớp tích chập cũng như chạy máy chủ của website.*

**I. Giới thiệu**

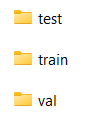
Nhận thấy nhu cầu nuôi thú cưng, vật nuôi trong nhà ngày càng phổ biến với người Việt Nam, mục đích của việc này không chỉ thỏa mãn đam mê của mỗi người mà còn làm giảm bớt áp lực do xã hội, bộn bề công việc. Mỗi loại vậy nuôi không chỉ có hình dáng khác mà còn có những màu sác vô cùng đa dạng tùy theo sở thích của mỗi người mà lựa chọn loại vật nuôi thích hợp với bản thân càng trở nên phong phú. Tuy vậy, việc nhớ tên các loại vật nuôi mà mình thích thích không phải việc đơn giản, tùy theo hình dáng, màu sắc mà chúng được đặt tên khác nhau theo chi, theo loài, theo bộ, ... đặc biệt là các loại cá cảnh, loại vật nuôi phổ biến bậc nhất tại Việt Nam. Nhận thấy khó khăn trên, em đã lên ý tưởng và thiết kế một mô hình mạng nơ ron dùng để nhận diện 15 loại cá cảnh.

Ứng dụng thực tiễn của mô hình là xử lý các hình ảnh cá bằng cách, sau khi người dùng chụp một bức ảnh liên quan đến cá đã được huấn luyện sẵn trước đó, khi hình ảnh trên được đưa vào mô hình sẽ được đi qua các bước làm mờ, giảm độ phân giải, làm nổi bật các đường nét đặc trưng của loài cá đó và được đem đi so sánh với từng loài cá trong mô hình có sẵn, tiếp đến mô hình sẽ đưa ra tỷ lệ phần trăm hình ảnh cá vừa đưa vào mô hình xem coi gần giống với loài cá nào có sẵn trong mô hình nhất, mô hình sẽ chọn phần trăm cao nhất và báo cho người dùng biết là hình ảnh cá vừa chụp gần giống với loài cá này nhất.

15 loài cá sẽ được dùng trong mô hình này lần lượt sẽ là: cá ali đầu bò, cá ba đuôi, cá bắp nẻ xanh, cá beta trắng, cá chim sâu, cá đĩa cá đuôi gai vàng, cá gamma hoàng gia, cá hề, cá la hán, cá lau kính cá mó tím cá ngựa, cá thù lù, cá trạng nguyên.

**II. Tiền xử lý dữ liệu**

Các hình ảnh sau khi được lấy từ các nguồn và lưu về máy và chia thành 3 thư mục khác nhau: train, test, validation, khi huấn luyện sẽ sử dụng tập train, sau mỗi bước huấn luyện sẽ sử dụng tập test để đánh giá lại mô hình ngay trong lúc huấn luyện và tập test để kiểm tra mô hình sau khi hoàn tất quá trình huấn luyện.



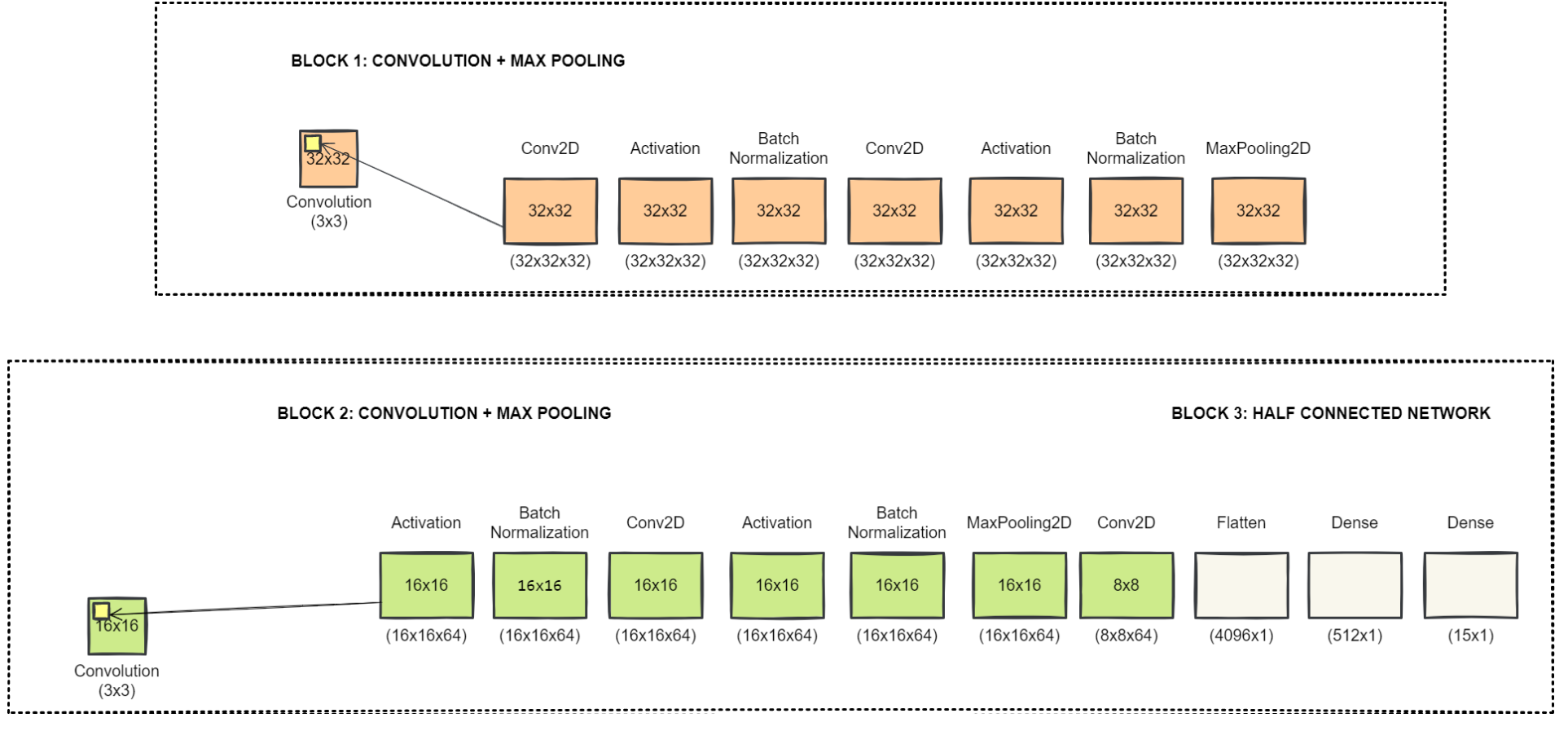
Hình 1: Các thư mục chứa ảnh 15 loài cá

Sau đó các ảnh sẽ được resize, cân bằng màu sắc, xoay ảnh, zoom ảnh, cắt ảnh. Nhằm để giảm nhiễu, tăng độ phức tạp cũng như tăng tốc độ quá trình huấn luyện.

**III. Xây dựng mô hình & hàm tối ưu**

**Mô hình**

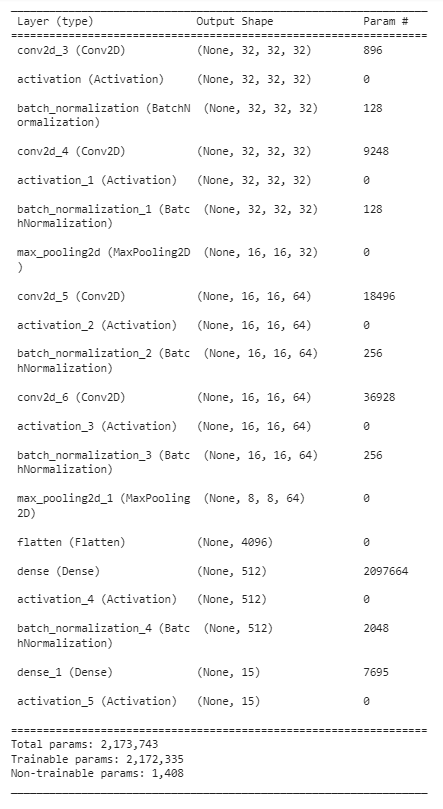
Dựa vào mô hình bên dưới ta có thể thấy ảnh đầu vào là 32x32x3 và ta sẽ sử dụng 32 kernel với kích thước 3x3 (màu vàng) để trượt qua từng ảnh gốc để làm nổi bật lên các đặc trưng, đồng thời sử dụng thêm padding = same để giữ lại kích thước gốc ban đầu của ảnh, lúc này kích thước ảnh trở thành 32x32 với 32 lớp màu 32x32x32. Sau đó ta sẽ thêm một lớp Activision (relu) phi tuyến để có thể sấp sỉ được các phân bố phức tạp và hàm này sẽ không làm thay đổi kích thước ban đầu của ảnh đầu vào. Tiếp theo ta sẽ thêm vào 1 lớp Batch Normization để cho phân bố của data đồng nhất với input ban đầu vì qua các lớp như Convolution hay Activation thì phân bố của data so với đầu vào là rất lớn. Tiếp tục thêm các lớp tương tự như trên để tăng độ phức tạp cho mô hình. Tiếp theo ta sẽ thêm 1 lớp MaxPooling2D, lớp này dùng để chọn ra những đặc trưng nổi bật nhất làm tăng độ chính xác đồng thời cũng làm giảm kích thước ảnh.



Hình 2: Kiến trúc mạng CNN sẽ xây dựng

Với Block 2 ta sẽ làm tương tự như Block 1 tuy nhiên ta sẽ giảm kích thước ảnh xuống 16x16 nhưng số lượng lớp màu sẽ tăng lên thành 64.

Ở Block 3 để có thể thực hiện phân loại thì đầu vào của lớp đó phải là vecto chứ không thể là 1 ma trận nữa. Để có thể chuyển từ ma trận sang vecto ta sẽ cho dữ liệu qua 1 lớp đó là Flatten. Tiếp tục thu nhỏ số lớp sao cho phù hợp với số lượng cần dự đoán, trong trường hợp này là 15 lớp tương ứng với 15 loài cá.



Hình 3: Mô hình sau khi được xây dựng

**Hàm tối ưu**

Learning\_rate = 0.001: Do dữ liệu không quá lớn khoảng 4000 hình ảnh nên em chọn tốc độ học tương đối lâu để mô hình chính xác nhất có thể.

Batch\_size = 128: Số lượng dữ liệu trong 1 lần lấy ra để học là 128.

Epochs = 10: Số lần huấn luyện là 10 lần.

Momentum = 0.9: Tạo một quán tính cho mô hình dùng để làm giảm sai số.

Thuật toán tối ưu SGD.

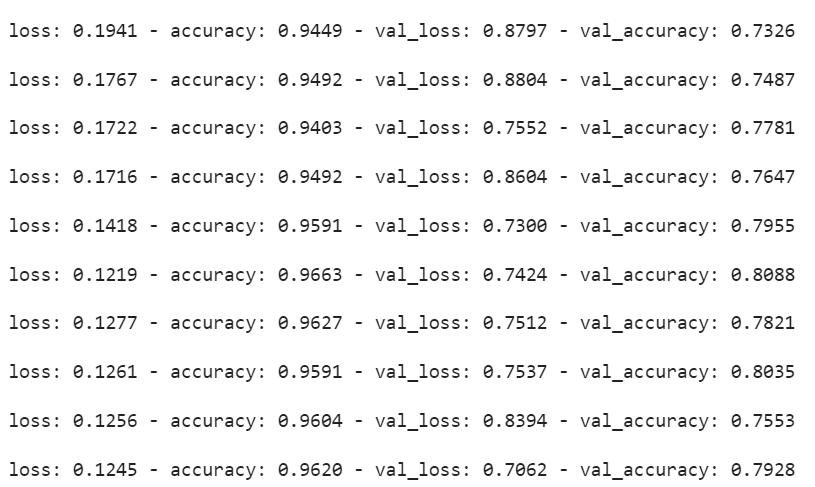
Loss = categorical\_crossentrophy: hàm mất mát này được sử dụng phổ biến trong các bài toán phân lớp.

Metrics = accuracy: Thông số accuracy dùng để đánh giá mô hình.



Hình 4: Các hàm tối ưu

**IV. Kết quả và đánh giá**

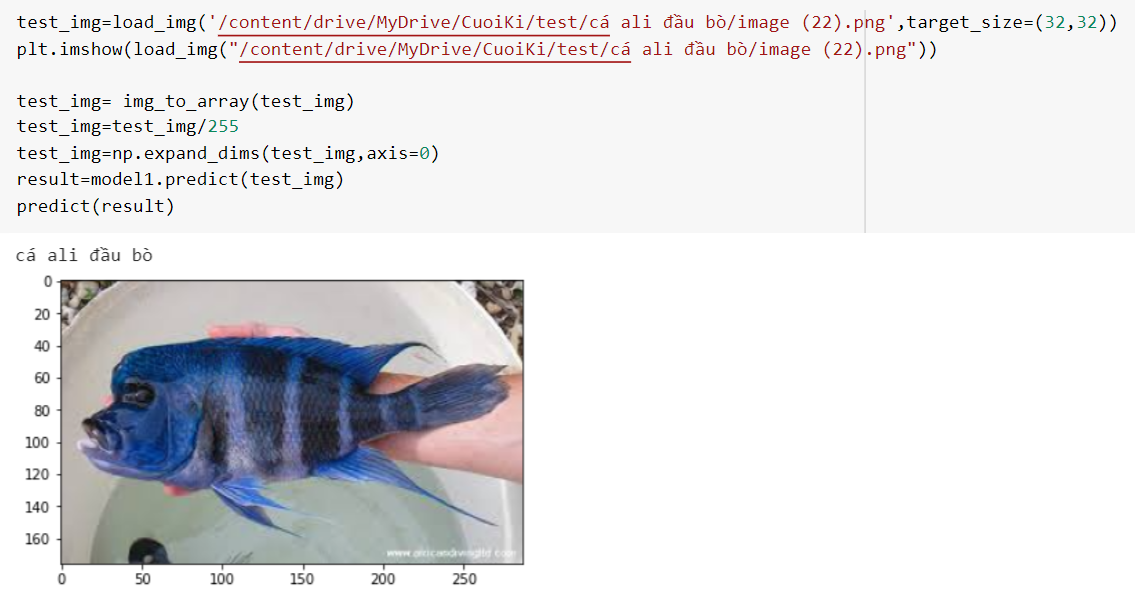


Hình 5: Các thông số sau khi huấn luyện mô hình

***Nhận xét:*** Sai số của tập dữ liệu huấn luyện là 0.125 với độ chính xác là 0.9620, sai số của tập dữ liệu kiểm chứng là 0.7062 với độ chính xác là 0.7928.

***Đánh giá:*** Chất lượng của mô hình phân loại là tương đối tốt với sai số < 1.

Dự đoán một số ảnh thực tế trong tập test.



Hình 6: Dự đoán cá ali đầu bò

***Nhận xét:*** Hình ảnh dự đoán là chính xác.



Hình 7: Dự đoán cá ngựa

***Nhận xét:*** hình ảnh dự đoán là chính xác.



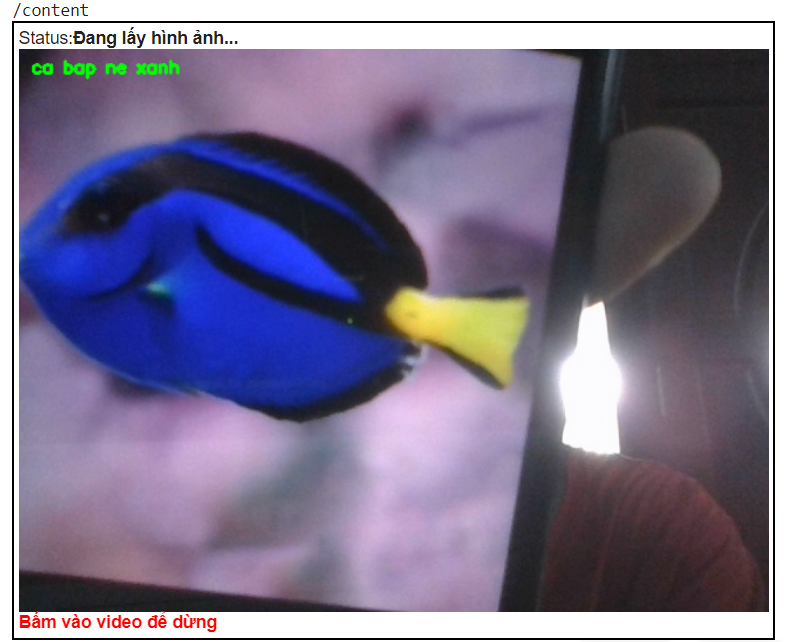
Hình 7: Dự đoán cá beta trắng

***Nhận xét:*** Hình ảnh dự đoán là sai cá beta trắng thành cá ngựa do dáng bơi thẳng đứng của cá beta trắng trong tấm hình này gần giống với dáng bơi của cá ngựa.

***Đánh giá:*** Tỷ lệ phân loại đúng của mô hình với tập dữ liệu test là khá chính xác lên đến 70-80% trừ trường hợp hình dạng của các loài cá gần giống nhau hoặc một số cá thể có màu sắc đặc biệt khác với các cá thể cùng loài.

**V. Nhận diện thời gian thực**

Chương trình sử dụng mã javascript để yêu cầu truy cập webcam của máy tính. Sau đó mỗi giây sẽ chụp một tấm hình với kích thước 640x480 và lưu hình ảnh dưới dạng jpg, sau đó tiến hành giảm độ phân giải xuống thành 32x32 và đưa vào mô hình.



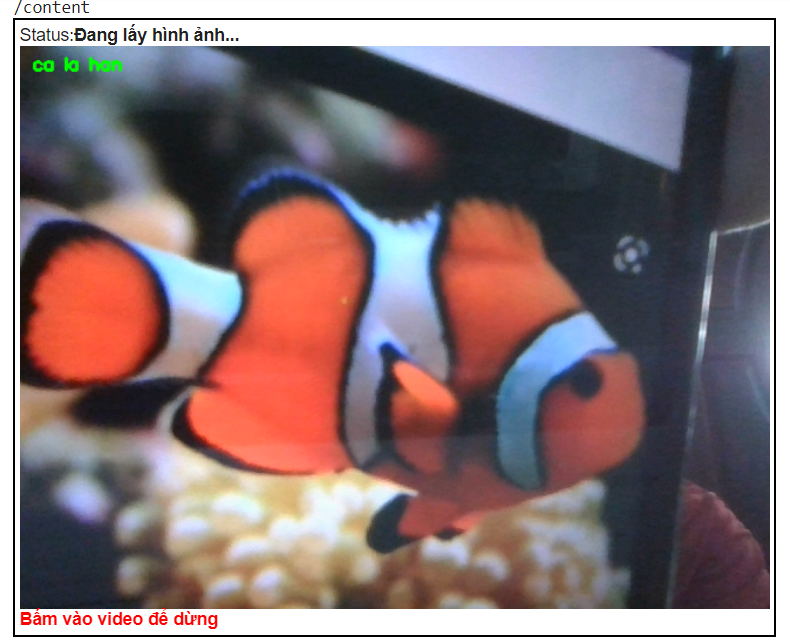
Hình 8: Dự đoán cá bắp nẻ xanh thời gian thực

***Nhận xét:*** hình ảnh dự đoán là chính xác



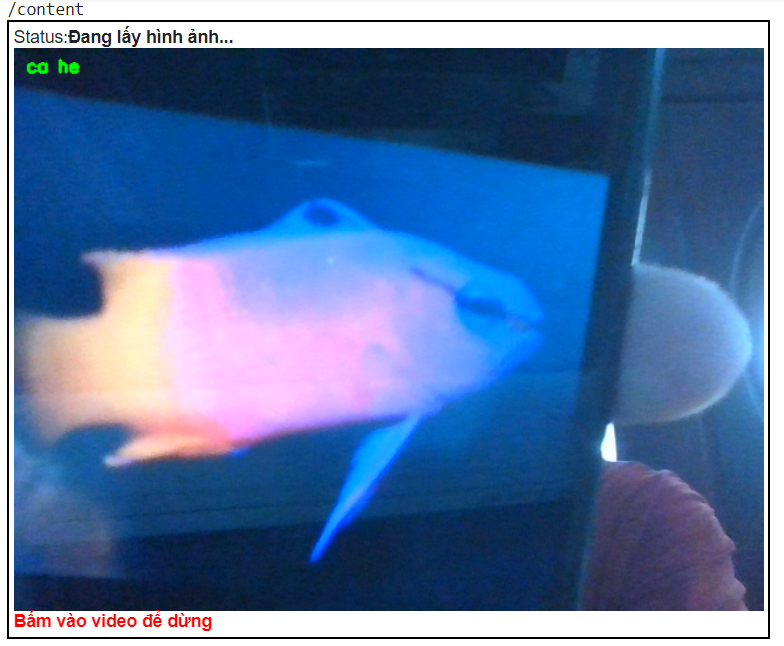
Hình 9: Dự đoán cá la hán thời gian thực

***Nhận xét:*** hình ảnh dự đoán là chính xác



Hình 10: Dự đoán cá hề thời gian thực

***Nhận xét:*** Hình ảnh dự đoán sai cá hề thành cá la hán



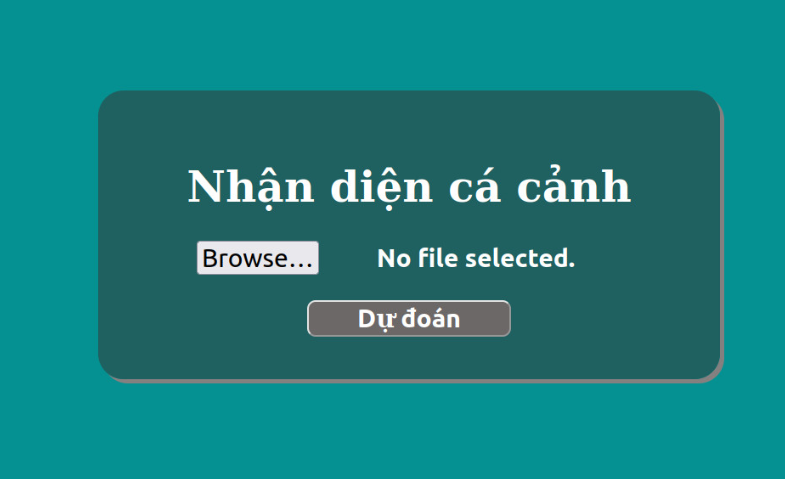
Hình 11: Dự đoán cá gmma hoàng gia thời gian thực

***Nhận xét:*** Hình ảnh dự đoán sai cá gamma hoàng gia thành cá hề

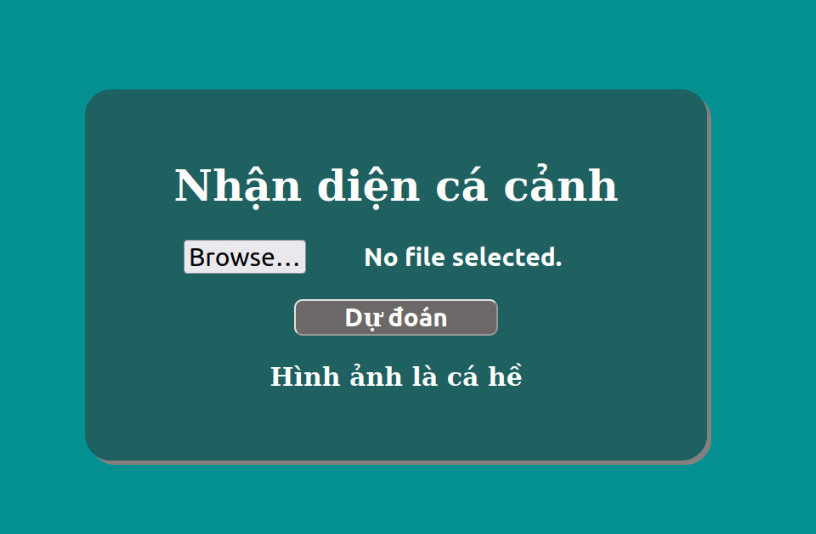
***Đánh giá:*** Mô hình dự đoán khá tệ khi ở thời gian thực do nhiều yếu tố như nhiễu ánh sáng, độ phân giải của camera máy tính, ... tỷ lệ dự đoán chính xác nằm ở khoảng 30-35%.

**VI. Thiết kế giao diện web**

Giao diện web đơn giản rất dễ sử dụng chỉ cần ấn nút “Browse”, sau đó chọn tệp hình ảnh có liên quan đến 15 loài cá đã được huấn luyện, sau đó ấn nút “dự đoán” sẽ có một dòng thông báo xuất hiện.



Hình 12: Giao diện web trước khi dự đoán



Hình 13: Giao diện web sau khi dự đoán

**KẾT LỜI**

Em xin trân trọng cảm ơn giảng viên Nguyễn Trường Thịnh và các trợ giảng đã cho em cơ hội cũng như cho em một đề tài vô cùng thú vị, giúp em có thêm nhiều kiến thức không chỉ liên quan đến môn học mà còn về tính tự học, cách giải quyết vấn đề. Em mong quý thầy cô và các đơn vị chấm bài sẽ cho em các nhận xét mang tính xây dựng và phê bình để em có thể cải thiện và phát triển hơn đề tài nghiên cứu hiện tại.

**TÀI LIỆU THAM KHẢO**

[1] Vũ Hữu Tiệp: Machine Learning cơ bản

[2] Chris Albon. 2018, Python Machine Learning Cookbook

[3] Wes McKinney. 2017, Python for Data Analysis

[4] Quing Kong, Timmy Siauw, Alexandre M. Bayen. 2020, Python Programming And Numerical Medthods

[5] Website Mì AI

[6] Website ProtonX