



NHẬN DIỆN 20 LOÀI THÚ CUNG SỬ DỤNG REALTIME

REAL TIME 20 PETS DETECTION

SINH VIÊN THỰC HIỆN: NGUYỄN THANH PHÁP

Khoa Đào Tạo Chất Lượng Cao, Trường Đại học Sư phạm kỹ thuật TP.HCM

Email liên hệ: 19146230@student.hcmute.edu.vn

Tóm tắt

Trong lĩnh vực giáo dục, các bài giảng của Thầy Cô giáo cần lặp đi lặp lại nhiều lần tên những loài động vật nói chung cũng như các loài thú cưng thường nuôi nói riêng rất nhiều lần để giúp học sinh nhớ lâu và không quên, đặc biệt là trẻ từ 3-6 tuổi. Với ứng dụng nhận diện thú cưng này, sẽ khắc phục được tình trạng đó, giảm bớt gánh nặng, hoạt động cho giáo viên tạo niềm cảm hứng học tập mới cho trẻ nhỏ, phù hợp với quá trình học tập của trẻ nhỏ cũng như các bậc phụ huynh kèm cặp con trẻ tại nhà. Vì vậy đề tài “Nhận diện 20 loài thú cưng sử dụng realtime” được ra đời để giải quyết vấn đề này. Với mô hình deep learning được viết trên ngôn ngữ Python, CSDL sử dụng mạng nơ-ron tích chập (CNN) cùng với sử dụng một số thư viện như Matplotlib, Numpy, Tensorflow hay Keras sẽ giúp chúng ta làm điều đó một cách hiệu quả và chính xác nhất.

Từ khóa: CSDL, Matplotlib, Numpy, mạng nơ-ron tích chập, CNN, Python, Tensorflow, Keras.

Abstract

In the field of education, teachers' lectures need to repeat many times the names of animals in general as well as pets in particular many times to help students remember for a long time and not forget, especially children 3-6 years old. With this pet recognition application, this situation will be overcome, reducing the burden, activities for teachers to create new learning inspirations for young children, suitable for the learning process of young children as well as children. Parents accompany their children at home. Therefore, the topic "Identifying 20 pet species using real time" was born to solve this problem. With a deep learning model written in Python language, a database using convolutional neural networks (CNN) along with the use of some libraries such as Matplotlib, Numpy, Tensorflow or Keras will help us do it efficiently and the most exactly.

Keywords: CSDL, Matplotlib, Numpy, convolutional neural network, CNN, Python, Tensorflow, Keras.

1 Đặt vấn đề

Trong thời gian gần đây, nhờ có sự phát triển mạnh mẽ về khả năng tính toán của các thế hệ máy tính hiện đại cũng như sự bùng nổ dữ liệu thông qua mạng lưới Internet trải rộng, ta đã chứng kiến nhiều sự đột phá trong lĩnh vực Trí tuệ nhân tạo, Machine learning cũng như là deep learning. Sự quay lại và phát triển vượt bậc của phương pháp học sâu bằng thuật toán CNN đã có được những thành tựu to lớn trong lĩnh vực nhận diện ảnh, trong đó có bài toán nhận diện động vật. Đề tài nghiên cứu “Nhận diện 20 loài thú cưng sử dụng realtime” đã được đưa ra với hy vọng có thể ứng dụng thành công các mô hình học sâu hiện đại để xây dựng các hệ thống nhận diện động vật một cách tự động, đặc biệt là đối với các loại thú cưng thường nuôi ở nước ta.

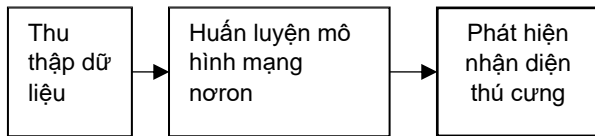
Để đưa ra đánh giá tổng quát và so sánh độ chính xác tương đối giữa các phương pháp học máy truyền thống với phương pháp học sâu, đề tài thực hiện cài đặt một mạng huấn luyện nơ-ron nhân tạo truyền thống và một mạng huấn luyện nơ-ron tích chập trong học sâu, sau khi thực hiện huấn luyện trên cùng bộ cơ sở dữ liệu ảnh và so sánh kết quả.

Đối với phương pháp Học máy truyền thống: nghiên cứu, tìm hiểu các phương pháp đã được trình bày trong các bài báo, công trình khoa học và thống kê ra các đặc trưng thường được sử dụng và cho kết quả huấn luyện tốt nhất. Các đặc trưng này thể hiện thông tin của các loài vật nuôi, hình dạng và kết cấu và đưa ra bộ tính toán, trích chọn đặc trưng của mạng nơ-ron nhân tạo.

Đối với mạng nơ-ron tích chập thuộc nhóm Học sâu: tìm hiểu và chọn một trong các mô hình huấn luyện phổ biến trong lĩnh vực Nhận diện ảnh trên thế giới để thực hiện cài đặt và so sánh kết quả với bộ nhận diện truyền thống.

Trong phạm vi nghiên cứu của bài báo, em sẽ tiến hành phát hiện nhận diện các loài vật nuôi dựa trên mạng nơ-ron tích chập, và đưa trực tiếp kết quả cho người dùng hiển thị tên con vật lên màn hình, đồng thời đọc tên con vật giúp trẻ nhỏ nhận biết nhanh chóng.

Cấu trúc chương trình gồm 3 bước được thể hiện trên hình dưới đây:



Bước 1: Thu thập dữ liệu: Sử dụng data tự tạo được tải từng bức ảnh trên google, lưu trữ tổng số ảnh cho mỗi loài động vật là 200 ảnh, tổng số ảnh trong cả bộ dữ liệu là 4000 ảnh, chia tập dữ liệu ra làm 3 phần Train, Validation và Test, mỗi phần đều có hình ảnh của 20 loài, tương ứng với tỉ lệ chiếm 70%, 20% và 10%.

Bước 2: Sử dụng nguồn dữ liệu đã thu thập được ở bước 1 để huấn luyện mô hình dựa trên mô hình mạng nơron tích chập CNN.

Bước 3: Phát hiện nhận diện các loài vật nuôi thú cưng, hiển thị tên của con vật trên màn hình, đồng thời đọc tên của con vật được nhận diện. Có sử dụng realtime nhận biết trực tiếp bằng Webcam.

2 Thuật toán của chương trình

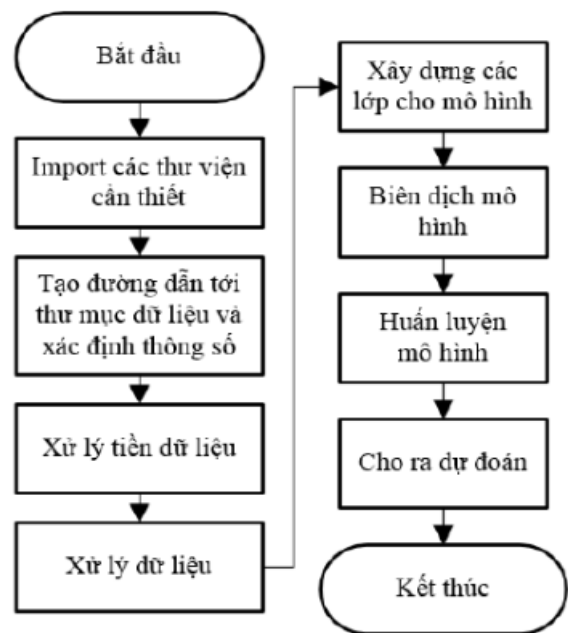
2.1 Thuật toán thu thập dữ liệu

Để thu thập dữ liệu, thay vì lấy nguồn dữ liệu trên mạng, đã thực hiện thu thập bằng chương trình được viết trên ngôn ngữ python, sử dụng thuật toán CNN để nhận diện, và Webcam để nhận diện trực tiếp. Dữ liệu sau khi thu thập dưới dạng file ảnh .jpg sẽ được lưu trữ ở 20 file riêng biệt: Mỗi file có 200 ảnh của từng loài vật nuôi bao gồm Cá Betta, Chuột Hamster, Gà tre cảnh, Lợn Cảnh, Nhện Tarantula, Nhím Kiêng, Rùa, Tắc Kè, Thỏ, Vẹt, Chó, Mèo, Cá Hải Tượng, Đại Bàng, Công, Rắn Ngộ,...

Mỗi file dữ liệu trên sẽ được chia làm hai phần chính nhờ hàm split trong python, bao gồm:

- Tập dữ liệu phục vụ quá trình thử nghiệm cuối cùng - Test (chiếm 10%);
- Tập dữ liệu phục vụ quá trình huấn luyện (chiếm 90%), trong đó tập dữ liệu huấn luyện này sẽ tiếp tục được chia thành 2 thành phần Train (chiếm 70% của tập huấn luyện) và Validation (chiếm 20% của tập huấn luyện) để kiểm thử độ chính xác của mô hình trong quá trình huấn luyện. Trong quá trình thu thập dữ liệu, điều kiện về ánh sáng của bức ảnh cũng làm ảnh hưởng tới chất lượng tạo mô hình sau này.

2.2 Thuật toán huấn luyện mô hình



Mô hình huấn luyện được xây dựng dựa trên mô hình mạng nơron tích chập CNN với thuật toán trên.

2.2.1 Import các thư viện cần thiết

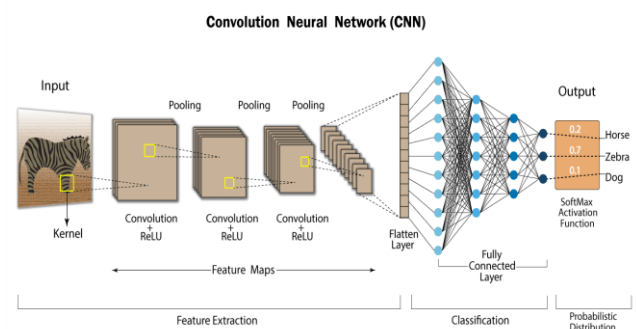
Nhập các thư viện cần thiết như OpenCV, Keras, Numpy (để xử lý dữ liệu), Os (để quản lý file), Matplotlib (trực quang hóa dữ liệu), Tensorflow (tính toán machinery với quy mô lớn) ...

2.2.2. Xử lý tiền dữ liệu và xử lý dữ liệu

Việc tiền xử lý dữ liệu nhằm đưa tất cả các ảnh về cùng kích thước, sau đó các ảnh này sẽ được chuyển đổi để phục vụ cho quá trình xử lý ảnh ở bước sau.

2.2.3. Xây dựng các lớp cho mô hình

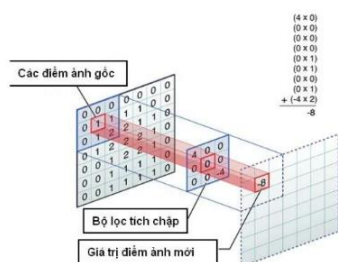
Các lớp được xây dựng dựa trên mô hình mạng nơron tích chập CNN (mạng nơron tích chập được thiết kế đặc biệt để xử lý dữ liệu đầu vào). Kiến trúc của CNN có các phần chính như sau:



Hình 1 Xây dựng các lớp cho mô hình

- Lớp Convolutional (tích chập): Đây là thành phần quan trọng nhất trong mạng CNN, cũng là nơi thể hiện tư tưởng xây dựng sự liên kết cục bộ thay vì kết

nổi toàn bộ các điểm ảnh. Các liên kết cục bộ này được tính toán bằng phép tích chập giữa các giá trị điểm ảnh trong một vùng ảnh cục bộ với các bộ lọc – filters – có kích thước nhỏ.



Hình 2 Phép tích chập giữa các giá trị điểm ảnh

Trong ví dụ, ta thấy bộ lọc được sử dụng là một ma trận có kích thước 3x3. Bộ lọc này được dịch chuyển lần lượt qua từng vùng ảnh đến khi hoàn thành quét toàn bộ bức ảnh, tạo ra một bức ảnh mới có kích thước nhỏ hơn hoặc bằng với kích thước ảnh đầu vào. Như vậy, sau khi đưa một bức ảnh đầu vào cho lớp Tích chập ta nhận được kết quả đầu ra là một loạt ảnh tương ứng với các bộ lọc đã được sử dụng để thực hiện phép tích chập. Các trọng số của các bộ lọc này được khởi tạo ngẫu nhiên trong lần đầu tiên và sẽ được cải thiện dần xuyên suốt quá trình huấn luyện.

- Lớp ReLU: Lớp này được xây dựng với ý nghĩa đảm bảo tính phi tuyến của mô hình huấn luyện sau khi đã thực hiện một loạt các phép tính toán tuyến tính qua các lớp Tích chập. Trong số các hàm kích hoạt phổ biến nhất như tanh, sigmoid..., hàm ReLU được chọn do cài đặt đơn giản, tốc độ xử lý nhanh mà vẫn đảm bảo được tính toán hiệu quả. Cụ thể, phép tính toán của hàm ReLU chỉ đơn giản là chuyển tất cả các giá trị âm thành giá trị 0.

Thông thường, lớp ReLU được áp dụng ngay phía sau lớp Tích chập, với đầu ra là một ảnh mới có kích thước giống với ảnh đầu vào, các giá trị điểm ảnh cũng hoàn toàn tương tự trừ các giá trị âm đã bị loại bỏ.

- Lớp Pooling: Một thành phần tính toán chính khác trong mạng CNN là pooling, thường được đặt sau lớp Tích chập và lớp ReLU để làm giảm kích thước kích thước ảnh đầu ra trong khi vẫn giữ được các thông tin quan trọng của ảnh đầu vào. Có hai phương pháp lấy mẫu phổ biến hiện nay là lấy mẫu giá trị lớn nhất (Max Pooling) và lấy mẫu giá trị trung bình (Average Pooling).

- Lớp Fully-connected: Lớp Fully-connected này hoàn toàn tương tự như trong mạng nơ-ron truyền thống, tức là tất cả các điểm ảnh được kết nối đầy đủ với node trong lớp tiếp theo.

Model: "sequential_1"

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 150, 150, 32)	896
conv2d_9 (Conv2D)	(None, 150, 150, 32)	9248
max_pooling2d_4 (MaxPooling 2D)	(None, 75, 75, 32)	0
conv2d_10 (Conv2D)	(None, 75, 75, 64)	18496
conv2d_11 (Conv2D)	(None, 75, 75, 64)	36928
max_pooling2d_5 (MaxPooling 2D)	(None, 37, 37, 64)	0
conv2d_12 (Conv2D)	(None, 37, 37, 128)	73856
conv2d_13 (Conv2D)	(None, 37, 37, 128)	147584
max_pooling2d_6 (MaxPooling 2D)	(None, 18, 18, 128)	0
conv2d_14 (Conv2D)	(None, 18, 18, 256)	295168
conv2d_15 (Conv2D)	(None, 18, 18, 256)	590080
max_pooling2d_7 (MaxPooling 2D)	(None, 9, 9, 256)	0
flatten_1 (Flatten)	(None, 20736)	0
dense_3 (Dense)	(None, 512)	10617344
dropout_2 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 256)	131328
dropout_3 (Dropout)	(None, 256)	0
dense_5 (Dense)	(None, 20)	5140
Total params: 11,926,068		
Trainable params: 11,926,068		
Non-trainable params: 0		

Bảng 1 Các lớp layer trong mô hình

2.2.4 Các bước chạy mô hình huấn luyện trên Google Colab

Bước 1: Khai báo các thư viện cần sử dụng.

Bước 2: Tiền xử lý dữ liệu.

Bước 3: Đi đến địa chỉ của các lớp.

Bước 4: Thiết lập, xây dựng model.

Bước 5: Biên dịch.

Bước 6: Kiểm tra kết quả.

Bước 7: Đánh giá độ chính xác của mô hình và vẽ đồ thị.

Bước 8: Chạy realtime nhận diện vật nuôi.

3. Kết quả thử nghiệm đánh giá, nhận xét và kết luận

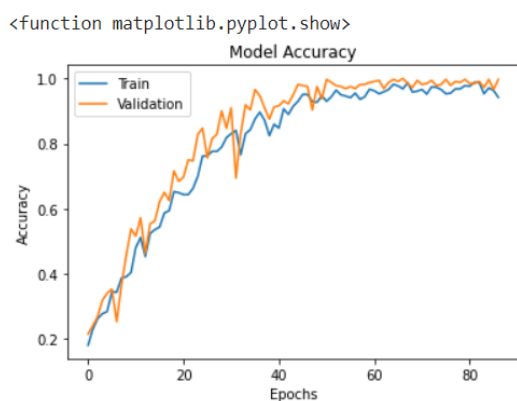
3.1 Kết quả thực nghiệm

Với phương pháp deep learning: Thống kê độ chính xác với tỉ lệ bộ Train/Validation/Test là 70/25/5. Độ chính xác đạt được là rất cao, ~98.3%, vượt trội so

với độ chính xác của mô hình huấn luyện sử dụng phương pháp Học máy truyền thống. Nhận diện đúng 20/20 loài thú cưng, gần như là độ chính xác tuyệt đối ở phần nhận diện bằng hình ảnh. Còn nhận diện bằng web cam khoảng 82%.

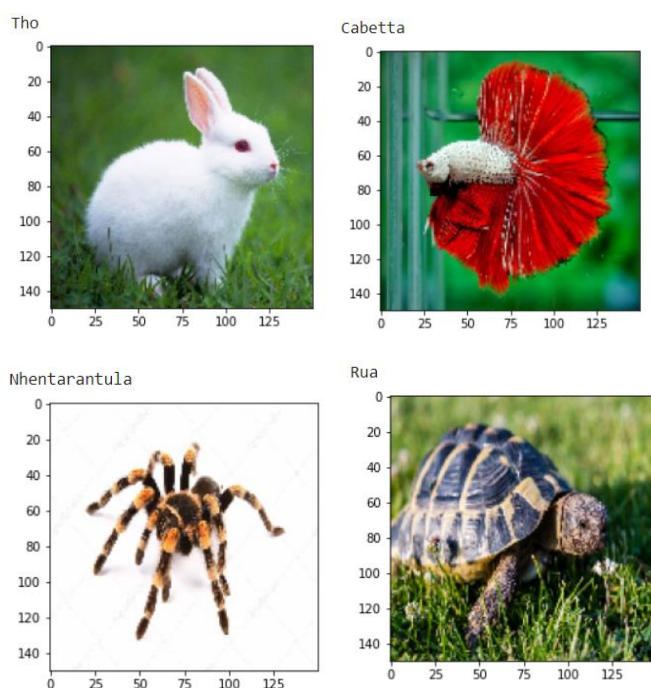
Train Loss 0.06689542531967163
Train Accuracy 0.9829145669937134

Hình 3 Độ chính xác của mô hình

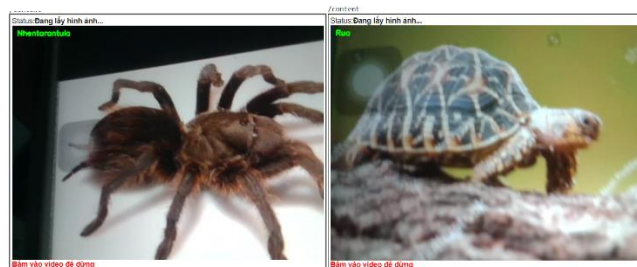
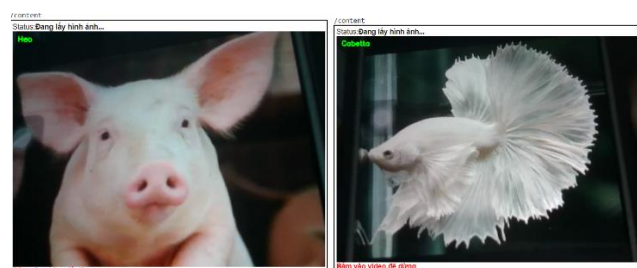


Hình 4 Đồ thị thể hiện độ chính xác của mô hình

Train bằng hình ảnh:



Train realtime bằng webcam:



3.2 Đánh giá kết quả

Với các bài toán nhận dạng và phân loại đối tượng nói chung, trong đó rất khó có thể chọn được các đặc trưng hiệu quả, thì Học sâu là phương pháp có ưu thế vượt trội so với các phương pháp Học máy truyền thống. Học sâu giúp đơn giản hóa quá trình huấn luyện mô hình nhận dạng khi không yêu cầu sự tham gia của người huấn luyện trong quá trình trích chọn đặc trưng, đồng thời cho phép tái sử dụng các mô hình đã huấn luyện trước để giảm thời gian cài đặt giải pháp cho các bài toán nhận dạng mới.

Sự cải thiện rõ rệt trong độ chính xác của mô hình nhận dạng sau khi tăng cường CSDL ảnh huấn luyện đã cho thấy hiệu quả thực tế của các phép sinh ảnh tự động sử dụng các phương pháp xử lý ảnh cơ bản. Chất lượng nhận dạng của ứng dụng trong thực tế cũng được tăng lên do các ảnh được sinh tự động giúp mô phỏng quá trình chụp ảnh trong đời thực, như các góc chụp khác nhau, các nhiễu sinh ra do môi trường, chất lượng máy ảnh... cũng như sự đa dạng của nền mà người dùng sử dụng để chụp ảnh. Việc tăng cường CSDL ảnh cũng là một giải pháp cho trường hợp khó thu thập ảnh để huấn luyện mô hình, tuy nhiên cũng cần phải chú ý đến mặt trái của việc lạm dụng phương pháp tăng cường ảnh này, đó là nguy cơ gây ra trạng thái “overfit” dữ liệu

(mô hình nhận dạng quá khớp với dữ liệu huấn luyện mà bị sai lệch với dữ liệu thực tế).

3.3 Kết luận

Đề tài project cuối kì môn Trí Tuệ Nhân Tạo đã nghiên cứu, tìm hiểu bài toán tự động nhận diện các loài vật nuôi, thú cưng trong ảnh, và thực hiện phát triển, cài đặt phương án giải quyết cho bài toán dựa trên sự tìm kiếm, thống kê các hướng tiếp cận đã được công bố qua rất nhiều bài báo, công trình khoa học trên thế giới. Các kết quả chính mà bài viết đã đạt được, tương ứng với các mục tiêu đề ra ban đầu như sau:

- Hoàn thiện xây dựng bộ cơ sở dữ liệu ảnh phục vụ huấn luyện nhận dạng cho 20 loài vật nuôi
 - phổ biến ở nước ta, với số lượng ảnh gốc trung bình cho mỗi loại quả là từ 150-200 ảnh.
 - Thống kê các đặc trưng thường được sử dụng để huấn luyện bộ nhận diện vật nuôi trong các phương pháp Học máy truyền thống, bao gồm các đặc trưng về màu sắc, hình dạng và kết cấu. Từ đó làm cơ sở xây dựng một mạng nơ-ron nhân tạo truyền thống và so sánh kết quả với một mạng nơ-ron tích chập thuộc nhóm phương pháp Học sâu.
 - Cài đặt và tinh chỉnh một mạng nơ-ron tích chập đã được huấn luyện trước, ứng dụng vào bài toán nhận diện các loài vật nuôi.. Thực nghiệm với bộ dữ liệu test và trong thực tế đã cho kết quả khá tốt, nguyên nhân chính là do phạm vi số lượng vật nuôi để nhận dạng đã được hạn chế chỉ còn 20 loài
 - Một con số rất khiêm tốn so với số lượng thú cưng ở Việt Nam nói riêng và cả thế giới nói chung. Hệ thống tự động nhận dạng thú cưng còn cần rất nhiều cải thiện, đặc biệt là về khả năng mở rộng phạm vi loài vật cũng như kích thước, chất lượng của bộ CSDL ảnh huấn luyện.

Link github:

<https://github.com/thanhphap059/Nhandienthucung>

Mã QR:



TÀI LIỆU THAM KHẢO

- [1] Liu Y, Sun P, Highsmith M R, et al. Performance comparison of deep learning techniques for recognizing birds in aerial images, 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). IEEE, pp.317-324, 2018
- [2] Wenxuan Han; Zitong Huang; Alifu. kuerban; Meng Yan; Haitang Fu, A Mask Detection Method for Shoppers Under the Threat of COVID-19 Coronavirus, 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), 2020.
- [3] Mohammad Marufur Rahman, Md. Motaleb Hossen Manik, An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network, 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2020.
- [4] Cezanne Camacho, 2018, Convolutional Neural Networks.
- [5] <https://www.slideshare.net/trongthuy3/luan-van-nhan-dang-va-phan-loai-hoa-qua-trong-anh-mau-hay>