Báo cáo công việc Đồ án tuần 4/3/2019.

Sinh viên: Nguyễn Tuấn Dương.

Em xin báo cáo kết quả những việc được giao trong tuần qua như sau:

1. Tìm các dataset cho bài toán phân vùng:

* Em không tìm thấy dataset nào là các ảnh chụp người mà có sẵn output đã phân vùng giữa người và background. Vì vậy phần output của dataset chắc chắn phải tự xử lý.
* Về dataset có ảnh mặt người đây là 2 dataset phù hợp nhất cho bài toán của mình mà có sẵn:
  + Selfie-dataset: <https://www.crcv.ucf.edu/research/data-sets/selfie/>

Có sẵn rất nhiều ảnh (46k ảnh), đa dạng tuy nhiên tất cả đều ở độ phân giải thấp (306x306), hầu hết các ảnh đều chụp góc hẹp vào khuôn mặt người, phần background rất ít trong khi bài toán của mình là xóa background, ảnh selfie góc rộng. Rất nhiều ảnh màu sắc bị biến đổi do filter, chỉnh sửa...

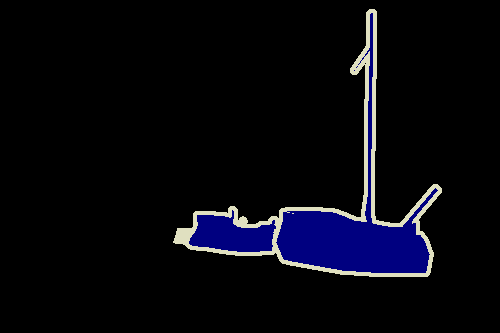
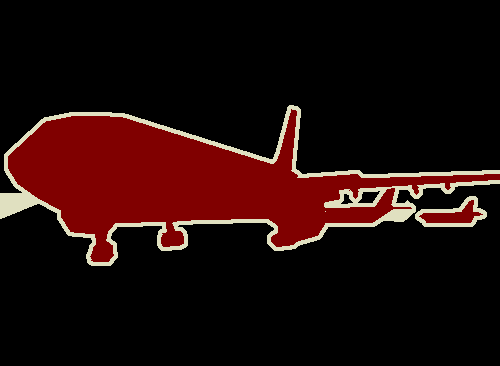
* + Selfiecity: <http://selfiecity.net/selfiexploratory/>

Có 3840 ảnh selfie với chất lượng tốt hơn nhưng cũng gặp vấn đề về bố cục như dataset trên. Thêm nữa em chưa tìm cách download được toàn bộ dataset, phải tải xuống từng ảnh một trên web.

* + Mình có thể dùng 2 dataset trên hoặc tải ảnh trên facebook về. Chất lượng tốt hơn với bố cục mong muốn. Tuy nhiên sẽ mất thời gian khá lâu (vì hiện giờ bọn em vẫn đang học fulltime trên công ty đến hết 25/3, chỉ có thời gian buổi tối). Về phần này em xin ý kiến anh nên dùng dataset nào ạ?

1. Về định dạng ảnh đầu ra của bộ dữ liệu PASCAL VOC 2012:

* Ảnh PNG kích cỡ 500x366. Vật thể được khoanh vùng bằng màu còn background thì trong suốt:



1. Về tiền xử lý ảnh:

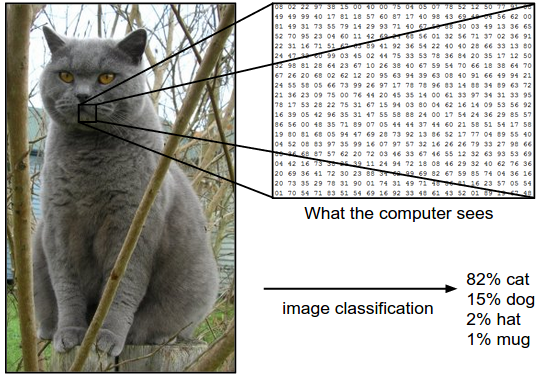
* Với python em sử dụng hàm có sẵn của Tensorflow để resize, crop, flip. Còn Tensorflow trên java không có các hàm đấy nên em sử dụng thư viện có sẵn của java.
* Code demo đã hoàn thành, em để trong thư mục preprocessing.

1. Về các kỹ thuật Machine Learning, Deep Learning em đã tìm hiểu tháng trước:

* Machine Learning: Huấn luyện cho máy học làm 1 công việc nào đó mà không cần phải lập trình 1 cách rõ ràng.

Ví dụ: Bài toán gán nhãn ảnh:

Nhiệm vụ là đưa ra nhãn (trong tập các nhãn) của bức ảnh được cho trước. Mỗi bức ảnh đầu vào là 1 mảng 3 chiều kích cỡ Width x Height x 3. Mỗi phần tử là số nguyên từ 0 – 255.



Sử dụng 1 tập các ảnh đã được gán nhãn trước (dataset) để huấn luyện cho mô hình. Sau khi huấn luyện máy sẽ có khả năng gán nhãn cho các bức ảnh đầu vào.

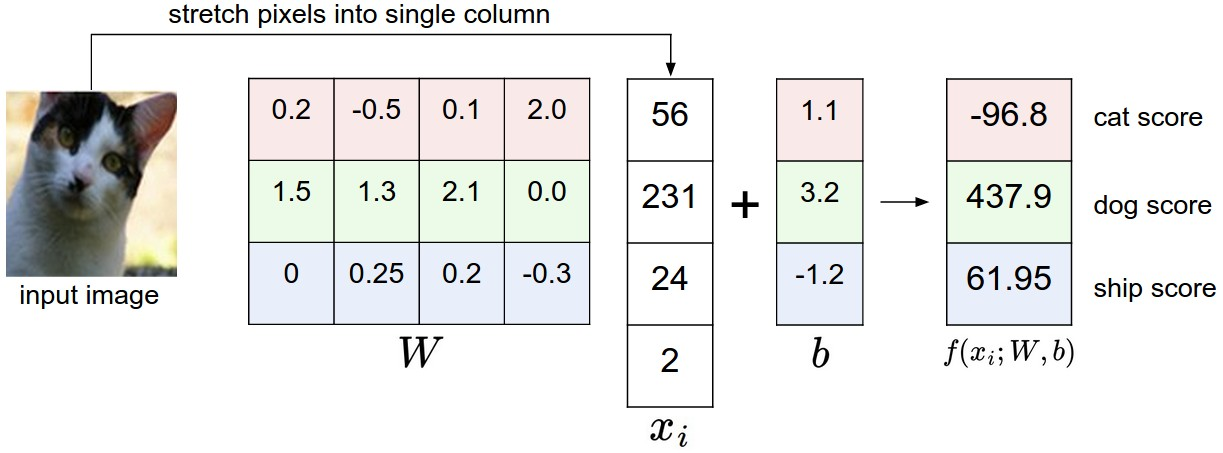
Linear Classification:

yi = f(xi,W,b)=Wxi+b

Trong công thức này: xi là ma trận của bức ảnh đã trải thành 1 cột. Ví dụ ảnh 32x32 sẽ có D = 32x32x3 = 3072 số. Trải thành ma trận [3072x1].

W là ma trận trọng số kích thước [K x D] (K là số các nhãn)

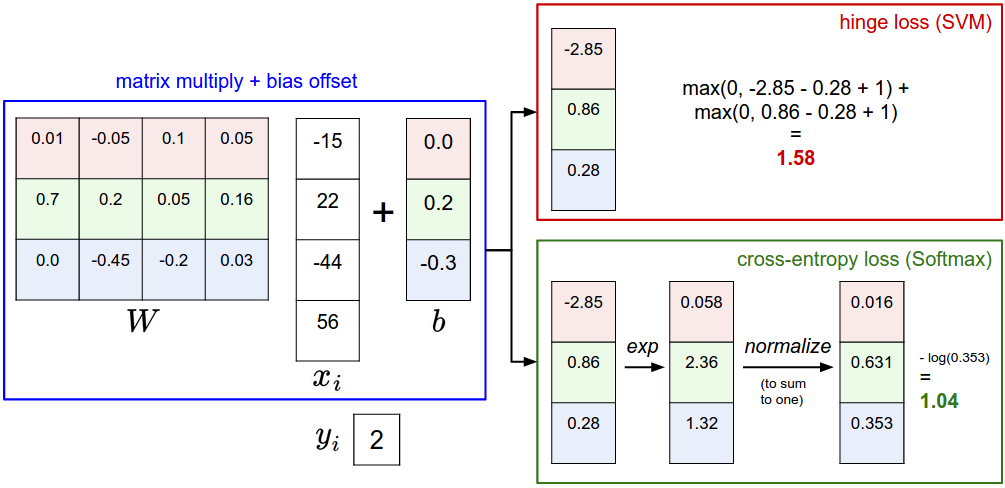
Với dataset đã gán nhãn thì (xi, yi) được fixed. Việc cần làm là sử dụng nó để huấn luyện các tham số W, b.



Tiền xử lý ảnh: Trong ví dụ trên mỗi số nguyên nằm trong khoảng [0..255]. Tuy nhiên thực tế cần phải đưa dữ liệu về trung tâm (khoảng [-127..127]) sau đó scale lại về [-1..1].

Loss function: Định lượng sự sai khác của dự báo trong tập training set. Từ đó tối ưu các tham số của mô hình W, b bằng cách giảm thiểu tối đa Loss function. Mỗi mô hình có cách tính Loss function khác nhau.

Ví dụ cách tính Loss function của SVM vs Softmax:



Optimization: Loss function giúp đánh giá chất lượng của tập trọng số W. Tuy nhiên, ma trận W có rất nhiều chiều (ví dụ trên là [10x3072]) nên để tối ưu hóa W, người ta sử dụng phương pháp Gradient Descent.

Ta sẽ tính toán độ dốc của hàm tại x loss dựa trên đạo hàm tại điểm đó. Sau đó điều chỉnh tham số đi theo hướng loss function giảm.

Backpropagation (truyền ngược): Một cách tính biểu thức gradient sử dụng đệ quy và quy tắc chuỗi. Được áp dụng trong Neural Network. Ví dụ tính đạo hàm của f(x,y,z) = (x+y)\*z tại (x,y,z)=(-2,5,-4):

