# Object Detection in an Image

Bruno de Almeida Silveira

October 2019

## Abstract

Here comes an abstract...

# 1 Definition

## 1.1 Project Overview

The main challenge in this project is to create a model that identifies many objects in an image. This challenge involves two main tasks. The former identifies in an image where it is the position of an object that it could be classified, and around it, we are going to draw a bounding box. The latter classifies those bounding boxes correctly, labeling a chair as a chair, a table as a table, and a human hand as a human hand.

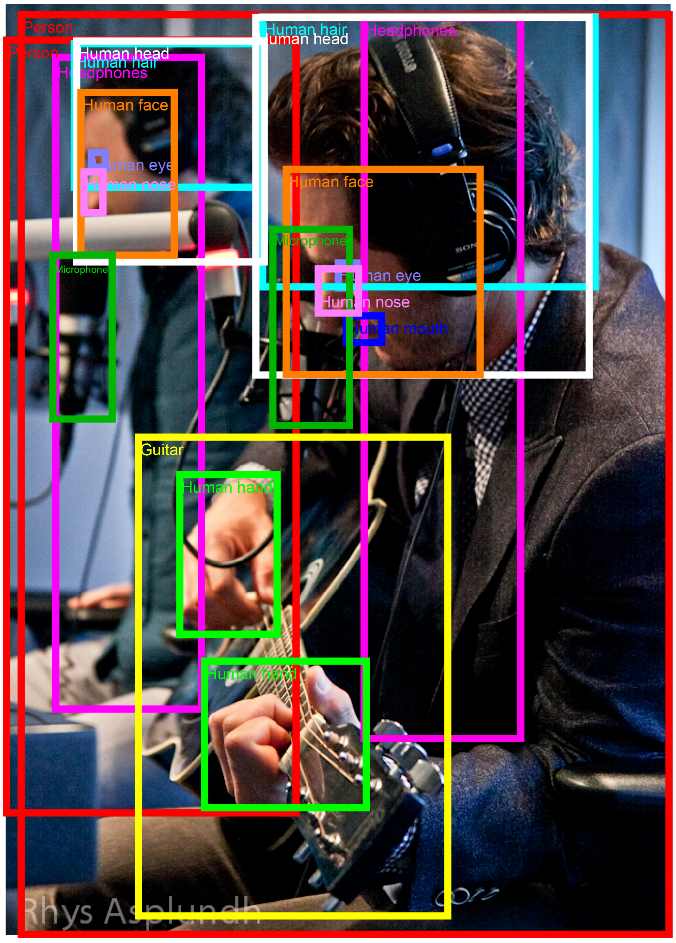


Figure 1: Mark Paul Gosselaar plays the guitar by Rhys A. [1]

The Kaggle challenge created by Google called Open Images 2019 - Object Detection [2] motivates this project. This Kaggle was created using the recent data set announced, the Open Images Dataset v5 [1]. This project proposes to show some strategies to solve the problem, given a deep dive into some deep neural network architectures.

## 1.2 Problem Statement

As described, the problem is going to be divided into two, the problem to create the correct bounding box and the problem to classify it. The idea is to create a simple pipeline with two models concatenated, and each model is going to optimize each technique.

The first model, called Bounding Box Identifier, is responsible for identifying all objects that could be classified by the Image Classifier Model. The loss function of this model is going to be the IoU (Intersection over Union) - described better in the next subsection.

On the other side, for the Image Classifier model, it is going to use the Categorial Crossentropy as a loss function.

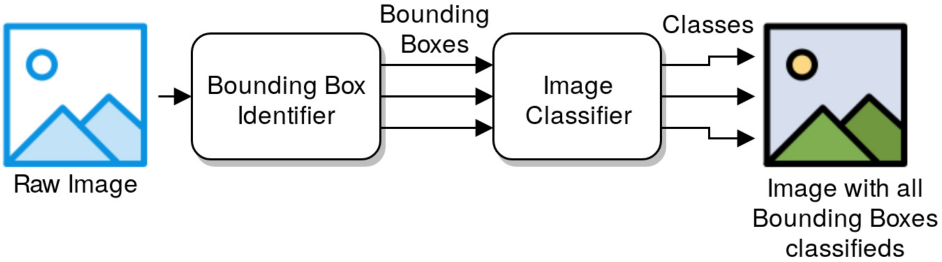


Figure 2: High level Model Pipeline

It is import to explaining more about the loss functions.

The Categorical Crossentropy used in the Classifier model is straightforward since it is a multiclass classifier. The idea is to compare for each class each distribution of the predictions made with the labeled distribution.

The IoU metric used as loss function arrives because mAP derives directly from the IoU. Since the mAP is going to be used as a metric to compare all models, it sounds reasonable to use IoU as a loss function to be optimized in the model responsible for creating the Bounding Boxes.

In the first step, the project starts to using the Transfer Learning technique in many famous networks - such as ResNet-N layers [3], Inception-V4 [4], Xception [5], and Inception-ResNets [4] - for both models, Object Identification, and Classifier. With Transfer Learning, we expect to achieve some reasonable results.

For the second and final step of the project, this work proposes to create, at minimum, one new architecture that could face other architectures seen above.

## 1.3 Metrics

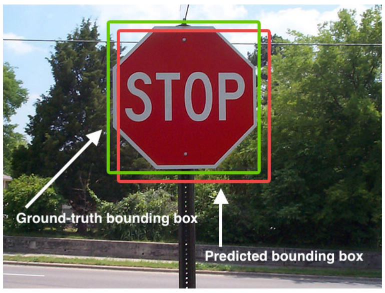
The metric proposed by Google in the competition is the mean Average Precision (mAP) [6], a very didactic explanation about the metric could be found in here [7] and [8].

The mAP metric could be defined as

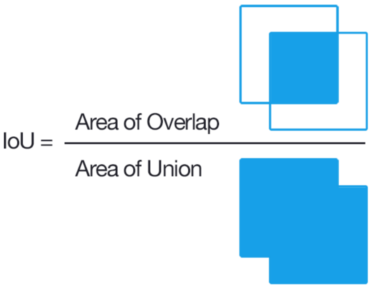
$$mAP = \frac{\sum\limits_{c=1}^{C} AP_c}{C}$$

where $C$ value is the number of all categories (classes).

To understand $AP_c$, it must comprehend first what is $IoU$. $IoU$ is the Intersection over Union, and it is equal to the ratio of the area of intersection and area of the union of the predicted bounding box (created by the model) and Ground-truth bounding box (previously annotated).



(a) Difference between a predict bounding box with a Ground-truth bouding box [9]



(b) IoU visual represented [9]

Here, it is going to define:

$$\begin{aligned} TruePositive &\doteq IoU > 0.5 \\ FalsePositive &\doteq IoU < 0.5 \\ &\quad or\ DuplicatedPredictBoundingBox \\ FalseNegative &\doteq IoU > 0.5 \\ &\quad and\ WrongClassification \end{aligned}$$

With the concept of True Positive (TP), True Negatives (TN), and False Positives (FN) defined, it is possible to create a Precision-Recall Curve, which defines a function that gives a precision based on the recall.

So by definition, $AP_c$ (Average Precision of some category c), is defined as an Area Under the Curve ($AUC$) of the Precision-Recall curve.

$$AP_c = \int_0^1 p(r)dr$$

where $p(r)$ is the precision defined in function of recall.

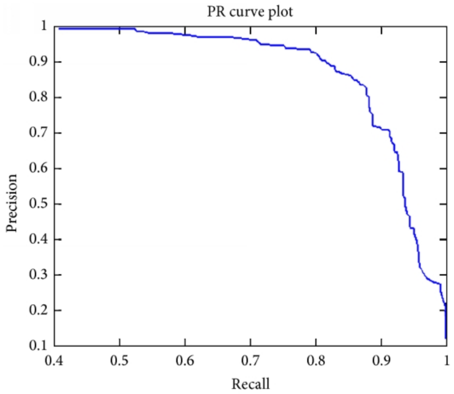


Figure 3: Precision-Recall curve [7]

It is essential to retain that in this project it is going to be used $AP_{50}$ (which uses a threshold of 0.5 in $IoU$ to define $TP$), but other average precisions metrics could be used, like, $AP_{75}$ (with $IoU$ threshold of 0.75) or $AP_{90}$ (with $IoU$ threshold of 0.90).

# 2 Analysis

## 2.1 Data Exploration

## 2.2 Exploratory Visualization

## 2.3 Algorithms and Techniques

## 2.4 Benchmark

# 3 Methodology

## 3.1 Data Preprocessing

## 3.2 Implementation

## 3.3 Refinement

# 4 Results

## 4.1 Model Evaluation and Validation

## 4.2 Justification

# 5 Conclusion

## 5.1 Free-Form Visualization

## 5.2 Reflection

## 5.3 Improvement

# References

[1] Google AI Blog. Introducing the open images dataset. https://ai.googleblog.com/2016/09/introducing-open-images-dataset.html, 2016.

[2] Google Research. Open images 2019 - object detection. https://www.kaggle.com/c/open-images-2019-object-detection/overview, jun 2019.

[3] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. Dec 2015. https://arxiv.org/abs/1512.03385.

[4] Vincent Vanhoucke Alex Alemi Christian Szegedy, Sergey Ioffe. Inception-v4, inception-resnet and the impact of residual connections on learning. Feb 2016. https://arxiv.org/abs/1602.07261.

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. Oct 2016. https://arxiv.org/abs/1610.02357.

[6] wikipedia. Mean average precision. https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval)#Mean_average_precision.

[7] Jonathan Hui. map (mean average precision) for object detection. https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173.

[8] Ren Jie Tan. Breaking down mean average precision (map). https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52#d439.

[9] Adrian Rosebrock. Intersection over union (iou) for object detection. https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.