

Large Scale Object-Detection using Focal-Loss Objective

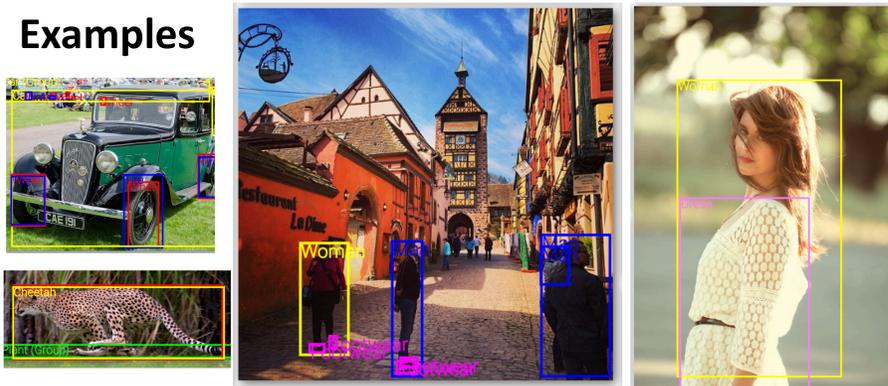
Abstract

Our approach to Google's open images challenge on object detection track involves using a **RetinaNet** architecture with its **focal-loss** objective function. The images then were fed through a pre-trained **resnet-152 backbone** layer initialized with ImageNet weights and then subsequently to a FPN network, and Bounding box and Class subnets to generate the predictions. The network performed well in terms of both mAP achieved and time taken per image during training and testing with $1/10^{\text{th}}$ of given training data used for training. Furthermore, our experiments predict that this particular architecture has much **potential for improvement** if all of the training data is used.

Data

The training data of open images dataset consists of **1,743,042 images** that has zero or more bounding-box annotations describing the type and location of 601 categories of objects [1]. The data for the challenge is a subset of the data with **500 categories** with **12.2 million bounding boxes**. The given images are resized so that their largest size is no larger than 1024 pixels. The entirety of open images dataset takes up approximately **560 GB of storage space**.

Examples



Ramana Anandakumar (1)

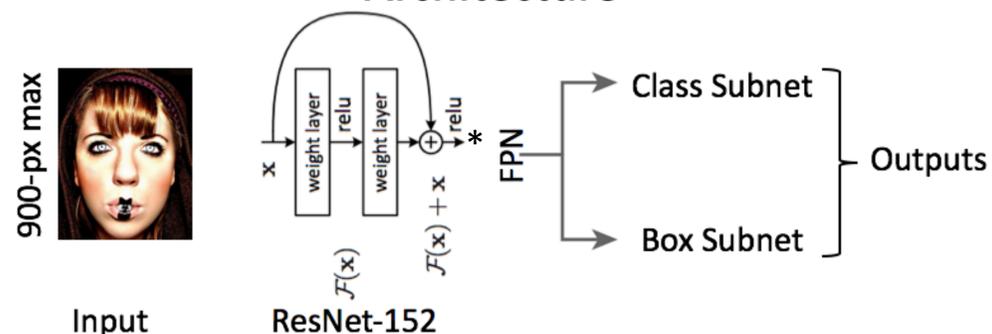
Keywords: Object Detection, Focal Loss, Single Shot Detection, Open Images

Focal-Loss Objective Function

The Focal Loss function aims to address scenarios where there is a large imbalance between foreground and background classes [2] and is defined as: $FL(p_t) = -(1 - p_t)^\gamma \log(p_t)$.

where, $p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases}$
 where, p is the model's predicted probability and $y = \pm 1$ is the label ground truth

Architecture



Implementation Details

All computations were performed on a custom-built workstation with:

- ❖ Intel i7 6-core processor
- ❖ **Nvidia 1080 Ti 11GB GPU**
- ❖ 32 GB RAM
- ❖ 256 GB NVME storage
- ❖ 4TB SATA storage

The input images were resized so that the largest side is no more than 900 pixels long. A **mini batch size of 1** was used with an **Adam optimizer** having a **learning rate of $1e-5$** . The ResNet-152 backbone layer with ImageNet weights were not frozen to allow for fine tuning of weights for this domain. The model was trained for **~29 hours** on **110,000 images**.

Results

Metrics

Training losses after training on 110,000 images:

loss: 1.3831
 regression_loss: 0.9273
 classification_loss: 0.4558

Local validation performed on a random sample of 1000 images from the training set mAP: 0.3179

Competition Leaderboard: mAP .11494

Training and Testing Speed

During training our model achieved a processing speed of **1.322 Images/second**, during testing it achieved **4.37 Images/second**.

Future Work

During training the model gained roughly about .02 in local validation mAP per 5000 images at a near linear rate. The training was stopped at 110,000 images. However, the rate of mAP growth and the failure to observe any asymptotic behavior of the metric at the time of stoppage suggests that the model has much room for improvement if trained on all of the data. Further improvements could be achieved by using the input images in its original resolution.

References

- [1] <https://github.com/cvdfoundation/open-images-dataset#download-images-with-bounding-boxes-annotations>
- [2] <https://arxiv.org/pdf/1708.02002.pdf>
- [3] <https://arxiv.org/pdf/1512.03385.pdf>
- [4] <https://storage.googleapis.com/openimages/web/visualizer/index.html?>