Final Project: INFO 7374 Cognitive Computing & Deep Neural Networks

Object Detection Using Mask R-CNN

04.09.2018

Group 5

Dhruv Patel

Fiona Lobo

Overview

In several applications of computer vision and image processing, the inception of the processing starts with object detection and subsequently tracking it if the need arises. In recent years, there has been extensive research in the field of object detection and tracking. Many remarkable algorithms have been developed for object detection and tracking, including color segmentation, edge tracking and many more. However, all these algorithms faced the limited success in their implementation in the real world and were also bounded by the constraints such as white/plain background. To overcome this limitation, a new method which is currently the best for object detection and instance segmentation was introduced called Mask RCNN (Regional – Convolution Neural Network).

Goals

- Detecting and segmenting objects in an image based on their outline by checking at pixel boundary level.
- Providing the accuracy of the detected objects along with the pixel boundary
- Implementing the model in TensorFlow

Future Goals

- Providing object detection in a real-time environment
- Implementing the model using PyTorch, OpenCV

Approach

Object Detection

Iterating over the problem of localization plus classification we end up with the need for detecting and classifying multiple objects at the same time. Object detection is the problem of finding and classifying a variable number of objects on an image. The important difference is the “variable” part. In contrast with problems like classification, the output of object detection is variable in length, since the number of objects detected may change from image to image. The main aim of object detection task is to detect the object (or a set of objects) from a predefined set of classes as well as detect the minimal sufficient bounding box around each object instance.

Instance & Image Segmentation

Moving ahead from object detection we would want to not only find objects inside an image, but find a pixel by pixel mask of each of the detected objects. We refer to this problem as instance or object segmentation. It does this by using the predefined set of classes or a probability distribution, which measures how a given pixel is probable to belong to the given class. Usually, when any other class is not assigned to the given pixel, an extra background class is added.

Mask R-CNN over Fast R-CNN

The main purpose of object instance segmentation with masking is to find a binary mask for every instance of objects from a predefined set of classes. The Mask R-CNN approach efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. The method, called Mask R-CNN, extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding boxes and a full object mask. Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps. Besides, Mask R-CNN is easy to generalize to other tasks, e.g., allowing us to estimate human poses in the same framework.

Use Cases

Face detection

Detection of users on Facebook and other social media. Eg. Detection of face on uploading a photo or others uploading your photo.

People / Object Counting

Object detection can be also used for counting people or objects in a store. It can be used to count the number of people at a festival or the number of articles sold from a store (Amazon Go).

Vehicle detection

Detecting if an object is a vehicle such as a bicycle or car. This is useful for self-driving cars or automatic vehicles. This system for detecting ships based on their type and structure are currently in development in some European countries.

Security

Detecting harmful objects such as a knife or gun can be useful to prevent outbursts of violence in a crowd. It can be used to identify harmful objects by feeding a video footage.

Data

Coco Dataset: http://cocodataset.org/#home

COCO is an image dataset designed to spur object detection research with a focus on detecting objects in context. The annotations include pixel-level segmentation of object belonging to 80 categories, key point annotations for person instances, stuff segmentations for 91 categories and five image captions per image.

We will work with a subset of the data provided by Microsoft as the dataset is huge and would require a high-end CPU.

Process Outline

1. Data Preprocessing
   - Reducing and later defining the size of training, testing and validation datasets
2. Anchor Sorting and Filtering
   - Visualizes every step of the first stage Region Proposal Network and displays positive and negative anchors along with anchor box refinement.
3. Bound Box Refinement
   - Final detection boxes (dotted lines) and the refinement applied to them (solid lines)
4. Mask Generation
   - Pixel wise mask generation around the detected object which get scaled and placed on the image in the right location.
5. Deploy the Model on Azure/AWS or Google Cloud Computing Platform
6. Taking observations and noting results
7. Preparing the demo and final submission

Milestones

Timeframe	Delivery
Week 1	Developing the project proposal Studying about Mask RCNN
Week 2	Familiarizing ourselves with the data, data preprocessing and loading Building the model and developing the code
Week 3	Training the model, generation of bound boxes and masks
Week 4	Training the model for better accuracy Providing Results & Demo Try to implement future goals

Personas:

1) Video Surveillance:

- To check for harmful objects in a crime scene investigation
- Analyze video for objects

2) Traffic Detection:

- For vehicles to detect other vehicles and objects

Deployment Details:

1) Language: Python
2) Deep Learning Framework: TensorFlow, Keras, OpenCV
3) Container: github
4) Cloud Tools: AWS EC2
5) Other Cloud Platforms: Microsoft Azure Machine Learning Studio, Google Cloud Platform, IBM Data Science Experience

Reference and Sources:

1. Paper Link: Paper: https://research.fb.com/wp-content/uploads/2017/08/maskrcnn.pdf
2. Dataset: MS COCO (Common Objects in Context): http://cocodataset.org/#download
3. Code written in Caffee2: https://github.com/facebookresearch/Detectron