1/21/2022

# *Image Processing*

2021-2022 Fall Semester
Project II – Final Report

**Group Members**

| | |
|---|---|
| 05170000809 | Morteza Yosefy |
| 05170000814 | Kibru Joba Kuture |
| 05170000815 | Michael Derece Kebede |

## Table of Contents

## 1. Section I

### The difference between image classification, object-detection & image segmentation?

One of the most common doubts which most of us have is what is the difference between image classification, object detection and image segmentation. Let's break down these terminologies which will help us to understand the difference between each of them.

*Figure 1 A cat.*

Image classification refers to a type of labelling where an image is assigned certain concepts, with the goal of answering the question, "**What is in this image?**"

You will have instantly recognized it. It's a cat. Take a step back and analyse how you come to this conclusion. You were shown an image and you classified to the class it belonged to. And that, in a nutshell, is what image classification is all about.

**Object detection** is a computer vision technique that deals with distinguishing between objects in an image. While it's related to classification, it is more specific in what it identifies, applying classification to distinct objects in an image, and using bounding boxes to tells us where each object is in an image. **Face detection** is one form of object detection. This technique is useful if you need to identify objects in a scene, like **scanner apps**, which detect the document on a table instantly, or detecting the **license plate** of the cars in an auto-park or in highways.

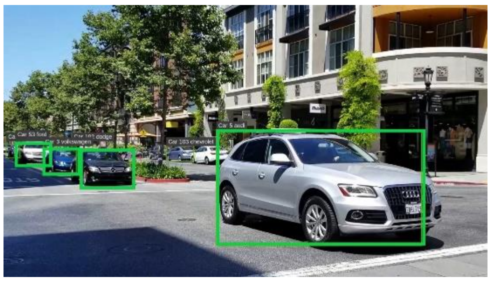


*Figure 2 Real-Time License Plate Detection and Recognition*

Before detecting the objects and even before classifying the image, we need to understand what the image consists of. This is where **Image Segmentation** can be helpful. We can divide or partition the image into various parts called **segments**. It's not a great approach to process the entire image at the same time as there will be areas which do not contain any information. By dividing the image into segments, we can make use of the important segments for processing the image. That, in a nutshell, is how Image Segmentation works.

An image is a collection or set of different pixels. **We group together the pixels that have similar attributes using image segmentation.** In other words, it's a type of labelling where each pixel in an image labelled with given concepts.

**In conclusion**, image Classification helps us to classify what is contained in an image. Image Localization will specify the location of single object in an image whereas Object Detection specifies the location of multiple objects in the image. Finally, Image Segmentation will create a pixel wise mask of each object in the images. We will be able to identify the shapes of different objects in the image using Image Segmentation.

# 2. Section II

## Deep Learning

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. ... In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound.

## Image Processing with Deep Learning

Imagine how much more valuable your data would be to your business if your document-intake solution could extract data from images as seamlessly as it does from the text.

Thanks to deep learning, intelligent document processing (IDP) can combine various AI technologies to not only automatically classify photos, but also describe the various elements in pictures and write short sentences describing each segment with proper English grammar. IDP leverages a deep learning network known as CNN (Convolutional Neural Networks) to learn patterns that naturally occur in photos.

To turn those documents into data, the Convolutional Neural Networks are trained using GPU-accelerated deep learning frameworks such as Caffe2, Chainer, Microsoft Cognitive Toolkit, MXNet, PaddlePaddle, PyTorch, TensorFlow, and inference optimizers such as TensorRT.

Neural networks were first used in 2009 for speech recognition and were only implemented by Google in 2012. Deep learning, also called neural networks, is a subset of machine learning that uses a model of computing that's very much inspired by the structure of the brain.

"Deep learning is already working in Google search and in image search; it allows you to image-search a term like 'hug.' It's used to getting you Smart Replies to your Gmail. It's in speech and vision. It will soon be used in machine translation, I believe." said Geoffrey Hinton, considered the Godfather of neural networks.

Deep Learning models, with their multi-level structures, as shown above, are very helpful in extracting complicated information from input images. Convolutional neural networks are also able to drastically reduce computation time by taking advantage of GPU for computation, which many networks fail to utilize.

## Convolutional Neural Network

The name "convolutional neural network" indicates that the network employs a mathematical operation called convolution. Convolutional networks are a specialized type of neural networks that use convolution in place of general matrix multiplication in at least one of their layers. In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of artificial neural network, most applied to analyse visual imagery.

A convolutional neural network consists of an input layer, hidden layers, and an output layer. In any feed-forward neural network, any middle layers are called hidden because their inputs and outputs are masked by the activation function and final convolution. In a convolutional neural network, the hidden layers include layers that perform convolutions.

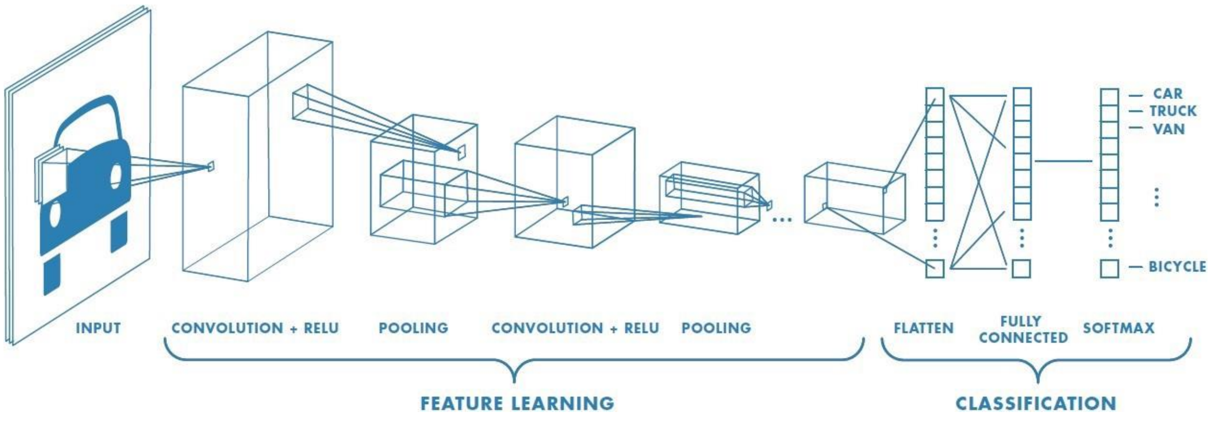


*Figure 3 Convolutional Neural Networks*

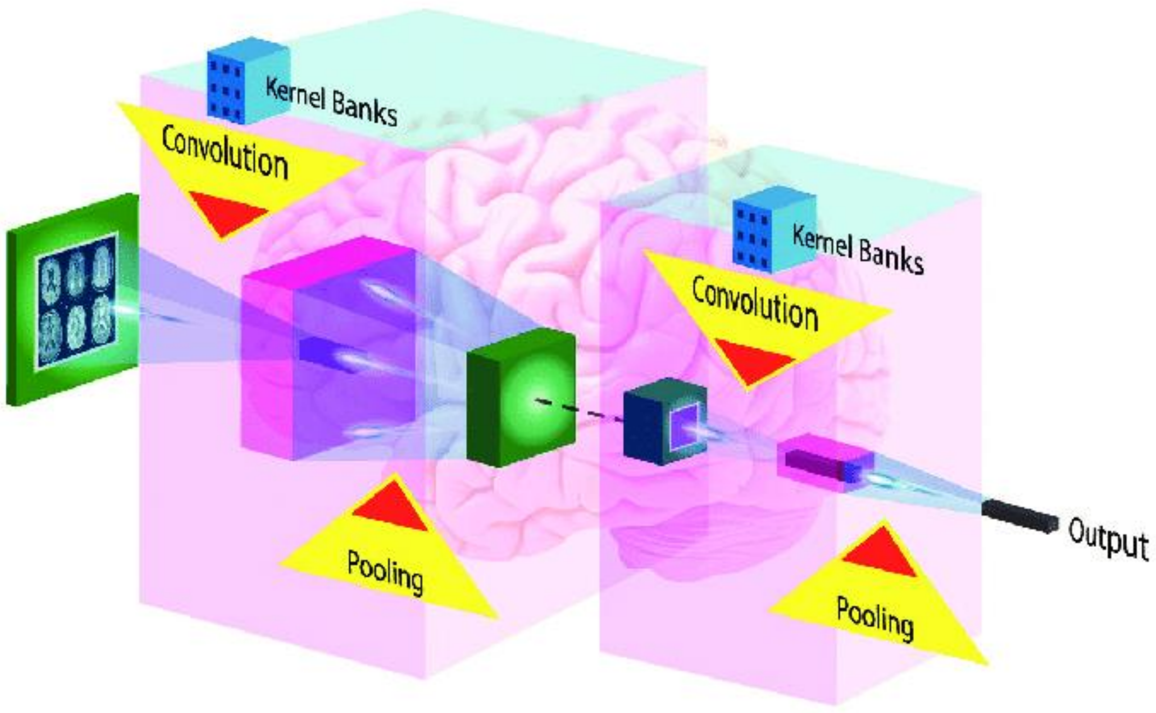


*Figure 4 Architecture of typical Convolutional Neural Networks (CNN) for image processing.*

## USING R-CNN

With the usage of region-based Convolutional Neural Network (aka RCNN), locations of objects in an image can be detected with ease. Within just 3 years the RCNN has moved from Fast RCNN, Faster RCNN to Mask RCNN, making tremendous progress towards human-level cognition of images.

Below is an example of the final output of the image recognition model where it was trained by deep learning CNN to identify categories and products in images.

## Applications of Deep Learning for Image Processing

Shelf auditing based on image classification using semi-supervised deep learning to increase on-shelf availability in grocery stores. [6]

I can look at the following computer vision problems where DL has been used.

Image Classification, Image Classification with Localization, Object Detection, Object Segmentation, Image Style Transfer, Image Colorization, Image Reconstruction, Image Super-Resolution, Image Synthesis, and other problems.

Link: [https://machinelearningmastery.com/applications-of-deep-learning-for-computer-vision/](https://machinelearningmastery.com/applications-of-deep-learning-for-computer-vision/)

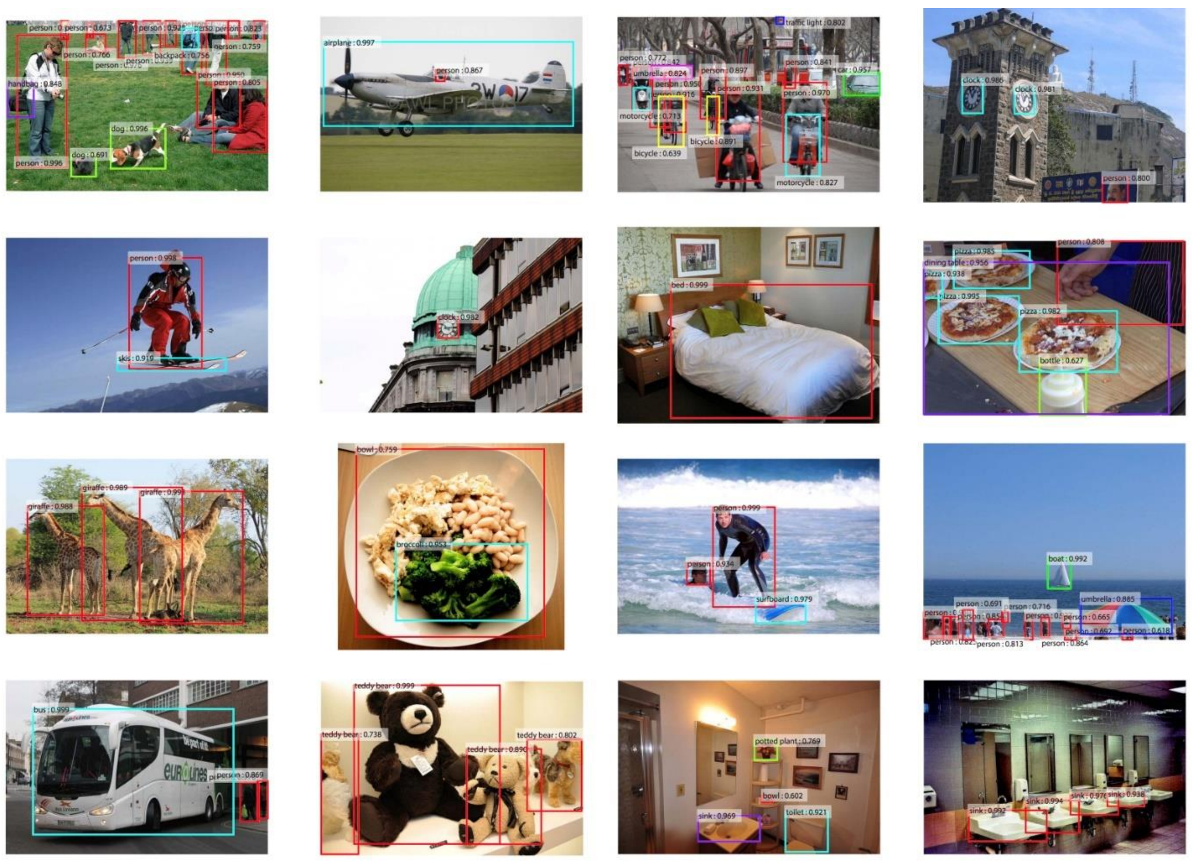


*Figure 5 Example of Object Detection with Faster R-CNN on the MS COCO Dataset*

## Conclusion

A lot of the computer vision techniques invented in the past 20 years have become irrelevant in recent years, all because of deep learning. However, knowledge is never obsolete and there is always something worth learning from each generation of innovation. Knowing only DL for CV will dramatically limit the kind of solutions in a CV engineer's arsenal.

## References for this section

1. https://www.infrrd.ai/blog/image-processing-with-deep-learning-a-quick-start-guide
2. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
3. https://en.wikipedia.org/wiki/Convolutional_neural_network
4. https://medium.com/s-a-a-s/dl-basic-concept-of-cnn-2ef4fc9b039b
5. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e
6. https://pdfs.semanticscholar.org/85e2/8202353cfe854e8975149fe9f6c486e6f900.pdf
7. https://machinelearningmastery.com/applications-of-deep-learning-for-computer-vision/
8. https://arxiv.org/pdf/1312.6229.pdf
9. https://arxiv.org/pdf/1506.01497.pdf
10. https://arxiv.org/ftp/arxiv/papers/1910/1910.13796.pdf

# 3. Section III

## 3.1. Topic Identification & Problem Definition

Our project topic is pistol object detector. An object detector localizes and identify multiple objects in a single image. Our object detector identifies/detects pistol(s) in a given image.

Object detection tells us not only what is in an image but also where the object is as well. Object detection as opposed to image classification can detect multiple classes in an image and draws rectangular boxes around them.

**Therefore, object detection algorithms allow us to:**

- Input one image
- Obtain multiple bounding boxes and class labels as output

At the very core, any object detection algorithm (regardless of traditional computer vision or state-of-the-art deep learning), follows the same pattern:

1. Input: An image that we wish to apply object detection to
2. Output: Three values, including:
   1. A list of bounding boxes, or the (x, y)-coordinates for each object in an image
   2. The class label associated with each of the bounding boxes
   3. The probability/confidence score associated with each bounding box and class label

https://www.pyimagesearch.com/2020/06/22/turning-any-cnn-image-classifier-into-an-object-detector-with-keras-tensorflow-and-opencv/

## 3.2. About The Dataset

### Weapon detection based on object detection

The datasets included in this section have been designed for the object detection task based on Deep Learning architectures with a CNN backbone. The selected images contain weapons and objects but also consider an enriched context of different background objects as well as the way objects are handled.

After the training stage on these datasets, the detection models must locate and distinguish between weapons and different common objects present in the background or handled similarly.

The datasets also attached the annotation files in Pascal VOC format with the region of the target objects in xml files.

The weapon data sets that are provided in this section focus specifically on the development of intelligent video surveillance automatic systems.

## The Pistol Object Detection

The Pistol detection dataset contains 3000 images of short guns with rich context in the background. The images selected from the internet contain one or more handguns in diverse situations including video surveillance contexts.

This dataset is designed in the related publication giving additional information about the image dataset and experiment results.

Link to a description about weapon detection:
https://www.dasci.es/transferencia/open-data/24705/

The following link brings you the pistol images:
https://github.com/ari-dasci/OD-WeaponDetection/tree/master/Pistol%20detection

## 3.3. Implementation

### 3.3.1. Environment, Libraries, and their Versions

- python3.7
- tensorflow==1.15.3
- keras==2.2.4
- numpy
- scipy
- Pillow
- cython
- matplotlib
- scikit-image
- opencv-python
- h5py
- imgaug
- IPython[all]

### 3.3.2. Detection Application with Deep Learning

The Region-Based Convolutional Neural Network, or R-CNN, is a family of convolutional neural network models designed for object detection, developed by Ross Girshick, et al. There are perhaps four main variations of the approach, resulting in the current pinnacle called Mask R-CNN. The Mask R-CNN introduced in the 2018 paper titled “Mask R-CNN” is the most recent variation of the family of models and supports both object detection and object segmentation.

Mask R-CNN is a sophisticated model to implement, especially as compared to a simple or even state-of-the-art deep convolutional neural network model. Instead of developing an implementation of the R-CNN or Mask R-CNN model from scratch, we can use a reliable third-party implementation built on top of the Keras deep learning framework.

https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/

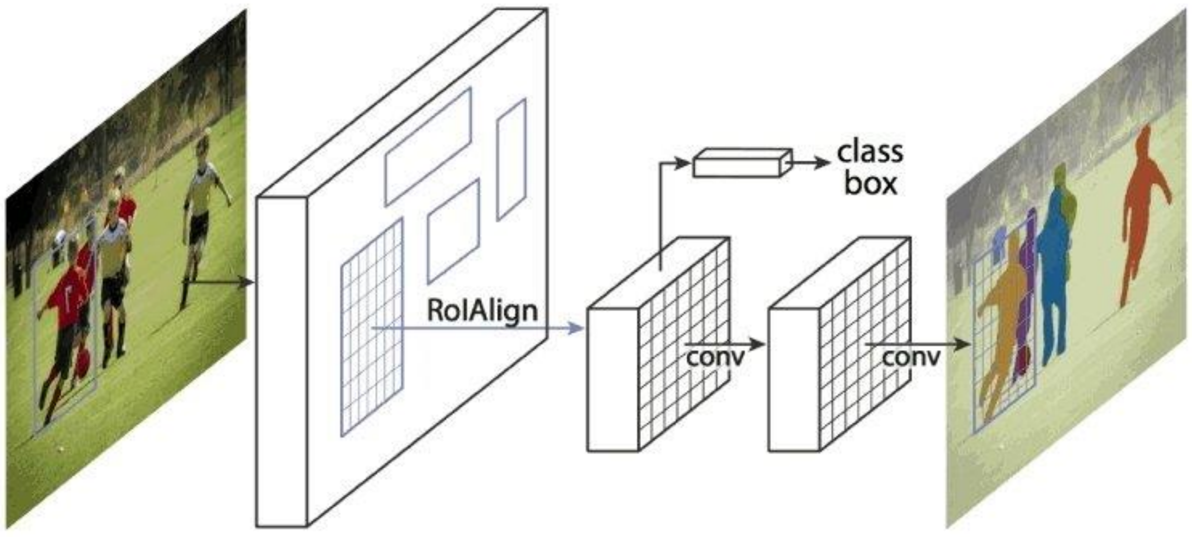
**How does Mask R-CNN work?**

Mask R-CNN was built using Faster R-CNN. While Faster R-CNN has 2 outputs for each candidate object, a class label and a bounding-box offset, Mask R-CNN is the addition of a third branch that outputs the object mask. The additional mask output is distinct from the class and box outputs, requiring the extraction of a much finer spatial layout of an object.

Mask R-CNN is an extension of Faster R-CNN and works by adding a branch for predicting an object mask (Region of Interest) in parallel with the existing branch for bounding box recognition.

**Advantages of Mask R-CNN**

- **Simplicity:** Mask R-CNN is simple to train.
- **Performance:** Mask R-CNN outperforms all existing, single-model entries on every task.
- **Efficiency:** The method is very efficient and adds only a small overhead to Faster R-CNN.
- **Flexibility:** Mask R-CNN is easy to generalize to other tasks. For example, it is possible to use Mask R-CNN for human pose estimation in the same framework



The Mask R-CNN model pre-fit on the MS COCO object detection dataset can be used as a starting point and then tailored to the specific dataset, in this case, the pistol dataset.

The first step is to download the model file (architecture and weights) for the pre-fit Mask R-CNN model. The weights are available from the GitHub project and the file is about 250 megabytes.

https://github.com/matterport/Mask_RCNN/releases/download/v2.0/mask_rcnn_coco.h5

https://machinelearningmastery.com/how-to-train-an-object-detection-model-with-keras/

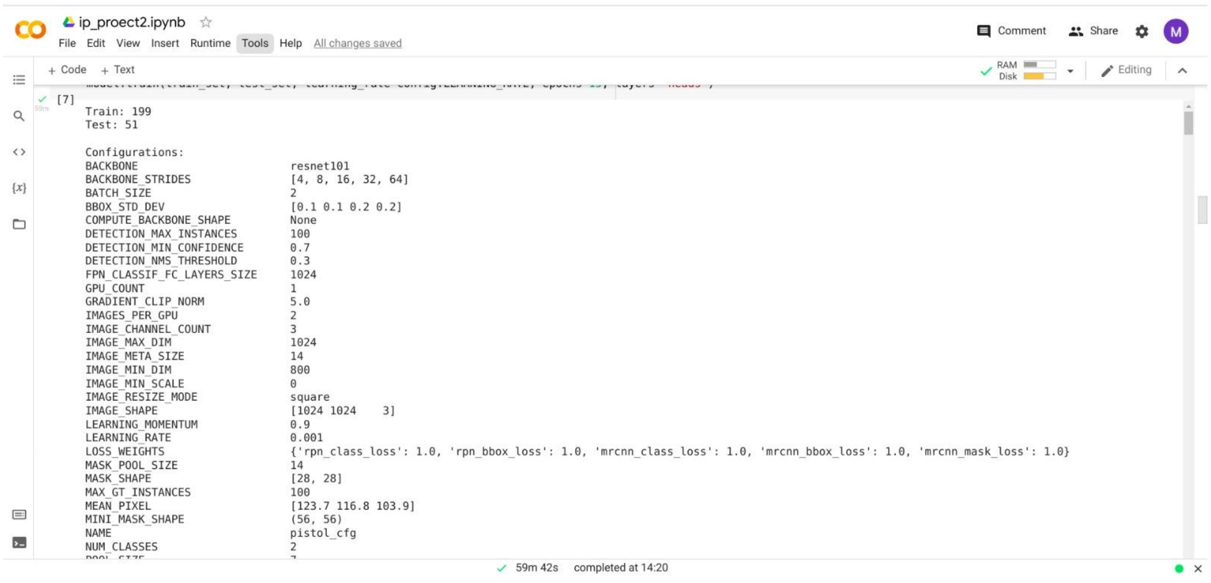
### 3.3.3. Which source have been used? What differences have been made?

The major difference between our project and the tutorial which we mentioned below, is the dataset that we used. We've used the pistol dataset, which we've already discussed about it before.

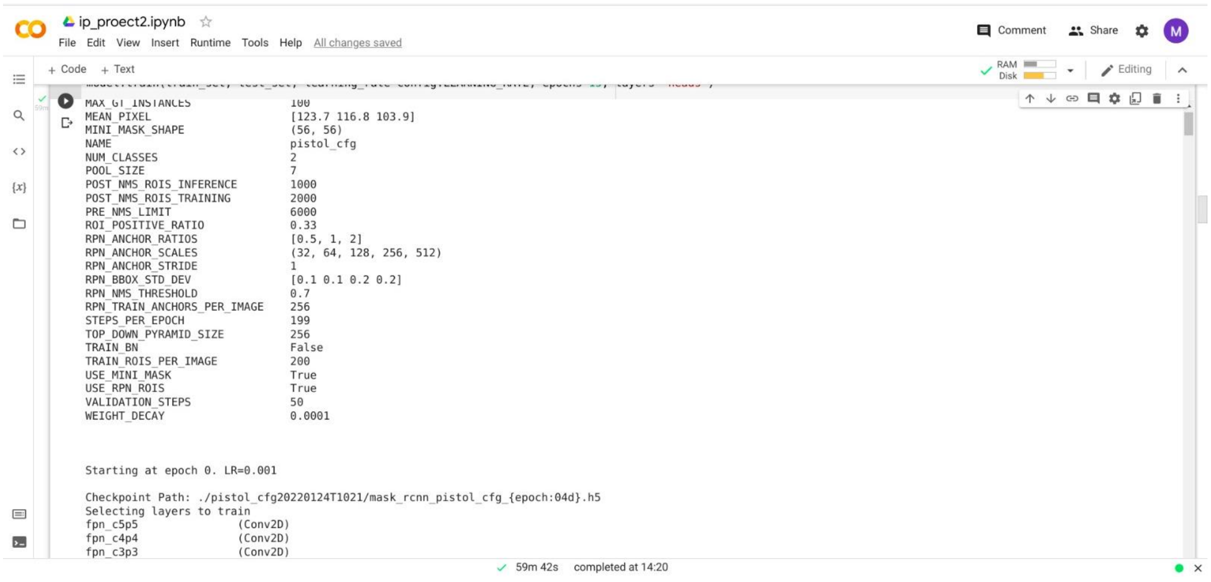
https://machinelearningmastery.com/how-to-train-an-object-detection-model-with-keras/

## 4.

## 5. Section IV - Experiments


ip_proect2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[7]
59m 42s completed at 14:20

```
Train: 199
Test: 51

Configurations:
BACKBONE                       resnet101
BACKBONE_STRIDES               [4, 8, 16, 32, 64]
BATCH_SIZE                     2
BBOX_STD_DEV                   [0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE         None
DETECTION_MAX_INSTANCES        100
DETECTION_MIN_CONFIDENCE       0.7
DETECTION_NMS_THRESHOLD        0.3
FPN_CLASSIF_FC_LAYERS_SIZE     1024
GPU_COUNT                      1
GRADIENT_CLIP_NORM             5.0
IMAGES_PER_GPU                 2
IMAGE_CHANNEL_COUNT            3
IMAGE_MAX_DIM                  1024
IMAGE_META_SIZE                14
IMAGE_MIN_DIM                  800
IMAGE_MIN_SCALE                0
IMAGE_RESIZE_MODE              square
IMAGE_SHAPE                    [1024 1024    3]
LEARNING_MOMENTUM              0.9
LEARNING_RATE                  0.001
LOSS_WEIGHTS                   {'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0, 'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}
MASK_POOL_SIZE                 14
MASK_SHAPE                     [28, 28]
MAX_GT_INSTANCES               100
MEAN_PIXEL                     [123.7 116.8 103.9]
MINI_MASK_SHAPE                (56, 56)
NAME                           pistol_cfg
NUM_CLASSES                    2
```

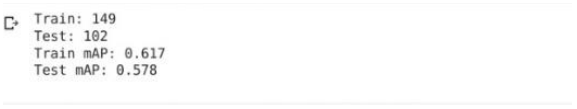

ip_proect2.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
59m 42s completed at 14:20

```
MAX_GT_INSTANCES               100
MEAN_PIXEL                     [123.7 116.8 103.9]
MINI_MASK_SHAPE                (56, 56)
NAME                           pistol_cfg
NUM_CLASSES                    2
POOL_SIZE                      7
POST_NMS_ROIS_INFERENCE        1000
POST_NMS_ROIS_TRAINING         2000
PRE_NMS_LIMIT                  6000
ROI_POSITIVE_RATIO             0.33
RPN_ANCHOR_RATIOS              [0.5, 1, 2]
RPN_ANCHOR_SCALES              (32, 64, 128, 256, 512)
RPN_ANCHOR_STRIDE              1
RPN_BBOX_STD_DEV               [0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD              0.7
RPN_TRAIN_ANCHORS_PER_IMAGE    256
STEPS_PER_EPOCH                199
TOP_DOWN_PYRAMID_SIZE          256
TRAIN_BN                       False
TRAIN_ROIS_PER_IMAGE           200
USE_MINI_MASK                  True
USE_RPN_ROIS                   True
VALIDATION_STEPS               50
WEIGHT_DECAY                   0.0001


Starting at epoch 0. LR=0.001

Checkpoint Path: ./pistol_cfg20220124T1021/mask_rcnn_pistol_cfg_{epoch:04d}.h5
Selecting layers to train
fpn_c5p5               (Conv2D)
fpn_c4p4               (Conv2D)
fpn_c3p3               (Conv2D)
```

Here we have almost 200 images for training the model, 50 pictures to test, two classes on for the background and the other for the pistol class. Also, number of epochs is 15.

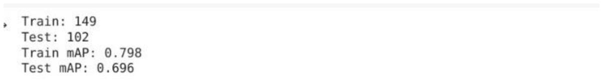
The performance of a model for an object recognition task is often evaluated using the mean absolute precision or mAP. The average or mean of the average precision (AP) across all the images in a dataset is called the mean average precision.

To increase our success rate we change the epoch number from 2 to 5 to 15.



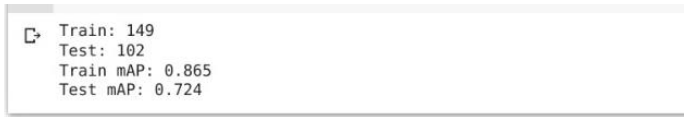
```
Train: 149
Test: 102
Train mAP: 0.617
Test mAP: 0.578
```

*Figure 6 Epoch 2*



```
Train: 149
Test: 102
Train mAP: 0.798
Test mAP: 0.696
```

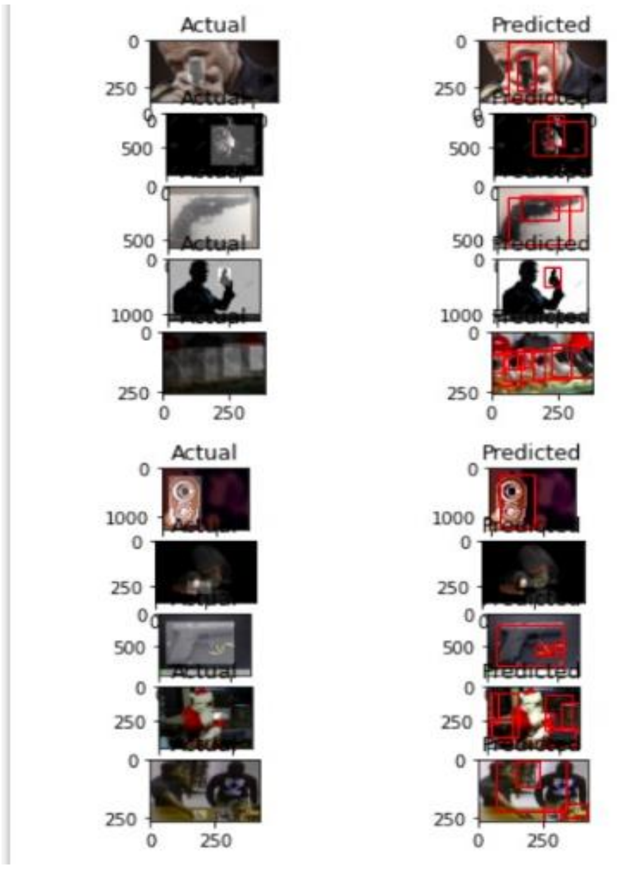
*Figure 7 Epoch 5*



```
Train: 149
Test: 102
Train mAP: 0.865
Test mAP: 0.724
```

*Figure 8 Epoch 15*

Actual vs. Predicted



In each photo at the top, the model has detected the pistol(s). We can see that in the case of the second last photo that a minor mistake was made. Specifically, the same pistol was detected multiple times.

No doubt these differences can be ironed out with more training, perhaps with a larger dataset and/or data augmentation, to encourage the model to detect people as background and to detect a given pistol once only.

## References

- https://www.infrrd.ai/blog/image-processing-with-deep-learning-a-quick-start-guide
- https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- https://en.wikipedia.org/wiki/Convolutional_neural_network
- https://medium.com/s-a-a-s/dl-basic-concept-of-cnn-2ef4fc9b039b
- https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e
- https://pdfs.semanticscholar.org/85e2/8202353cfe854e8975149fe9f6c486e6f900.pdf
- https://machinelearningmastery.com/applications-of-deep-learning-for-computer-vision/
- https://arxiv.org/pdf/1312.6229.pdf
- https://arxiv.org/pdf/1506.01497.pdf
- https://arxiv.org/ftp/arxiv/papers/1910/1910.13796.pdf
- https://developer.ibm.com/exchanges/models/all/max-object-detector/
- https://www.pyimagesearch.com/2020/06/22/turning-any-cnn-image-classifier-into-an-object-detector-with-keras-tensorflow-and-opencv/

## Self-Assessment Table

| | Requirements | Done | Pt | Explanation | Score |
|---|---|---|---|---|---|
| | **Cover Page** | ☒ | 5 | The required elements are included in cover page. | 5 |
| 1 | **Section I - Image Classification vs Object Detection vs Image Segmentation** | ☒ | 5 | We've gone over these concepts; using slides & google. | 5 |
| 2 | **Section II -Applications of DL for IP (Research)** | ☒ | 5 | We've explained this topic in detail. Some relevant concepts and examples are mentioned. | 5 |
| 3 | **Section III** | ☒ | | | |
| 3.1 | **Definition** | ☒ | 5 | | 5 |
| 3.2 | **About The Dataset** | ☒ | 5 | | 5 |
| 3.3 | **Model Implementation** | ☒ | 25 | | 24 |
| 4 | **Section IV - Experiments** | ☒ | 25 | | 23 |
| 5 | **Section V - Self-Assessment Table** | ☒ | 10 | | 10 |
| | **References** | ☒ | 5 | | 5 |
| | **Report's Layout** | ☒ | 10 | | 10 |
| **Total points out of 100:** | | | | | **97** |

## Labour Division

| Group members | Tasks (specific) | Hours |
|---|---|---|
| **Michael** | All the sections.<br>Focused on training the model | 40 h |
| **Kibru** | All the sections.<br>Focused on doing research | 16 h |
| **Morteza** | All the sections.<br>Prep report n documentation | 18 h |
| Note: All of us had equal contribution and spent some efforts on each part. | | |