Computer Vision 2023 Project

Sealife Classification and Detection

By

T4

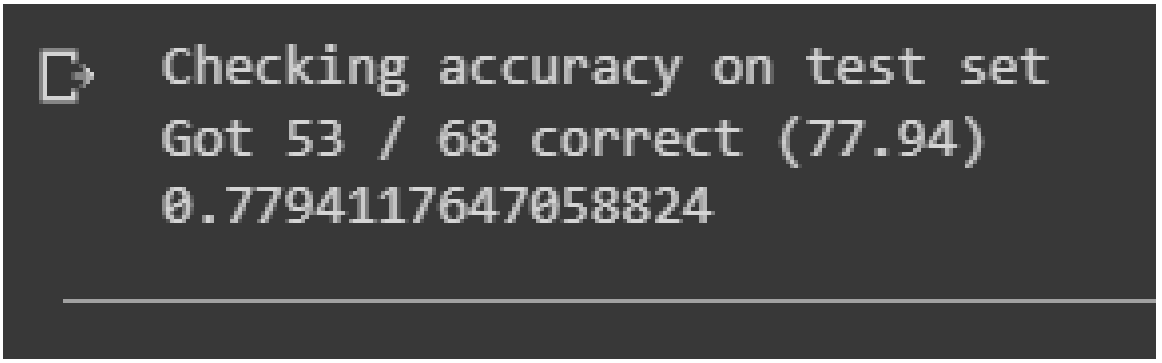
No	Name	ID
1	يوسف عزت احمد عبدالمجيد	2018170476
2	احمد مصدق ابراهيم	2018170057
3	يوسف محمد صديق يوسف	2017170530
4	احمد خالد صابر عبدالهادي	2018170512
5	احمد شعبان رجب عبدالفتاح	2018170036

Data preparation:

1. Read image path using by using OpenCV
2. Resize the images to (224,224)
3. Different data augmentation techniques
4. Apply normalization.

Models:

We used ResNet18 and VGG-16 in classification and the best accuracy was 77.94%



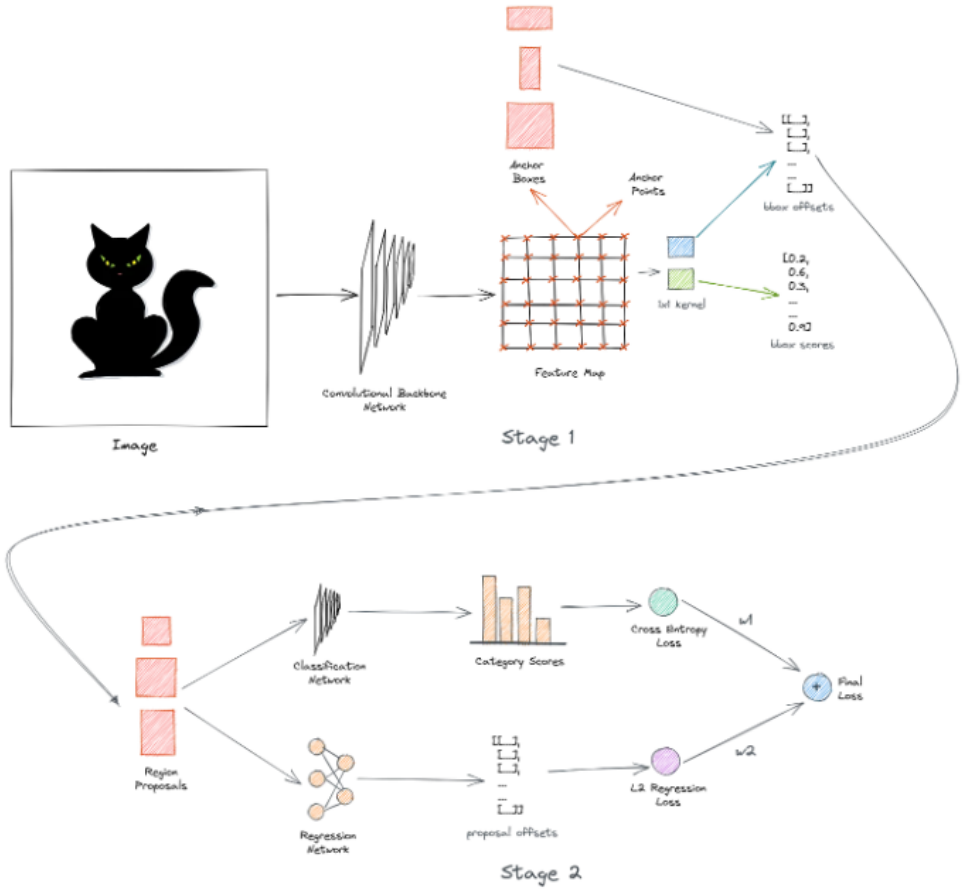
- Image size (224 x 224)
- 50 epochs
- Random Horizontal Flip
- Random Vertical Flip
- Random Rotation Flip
- Colour jitter

Segmentation:

We used k-means and mean shift clustering from OpenCV

Original	k-means	mean shift
		

Object Detection - Faster R-CNN



Faster R-CNN Overall Architecture

object detection we need to build a model and teach it to learn to both *recognize* and *localize* objects in the image.

Approach: Two Stages

Stage 1

- The Image first passes through the backbone network to get an output feature map, and the ground truth bounding boxes of the image get projected onto the feature map. The backbone network is usually a dense convolutional network like ResNet or VGG16. The output feature map is a spatially dense Tensor that represents the learned features of the image.
- Next, we treat each point on this feature map as an anchor. For each anchor, we generate multiple boxes of different sizes and shapes. The purpose of these anchor boxes is to capture objects in the image.
- We use a 1x1 convolutional network to predict the category and the offsets of all the anchor boxes.
- During training, we sample the anchor boxes that overlap the most with the projected ground truth boxes. These are called positive or activated anchor boxes. We also sample negative anchor boxes which have little to no overlap with the ground truth boxes. The positive anchor boxes are assigned the category object, while the negative boxes are assigned “background”.
- The network learns to classify anchor boxes using binary cross-entropy loss.
- Now, the positive anchor boxes may not exactly align with the projected ground truth boxes. So we train a similar 1x1 convolutional network to learn to predict offsets from ground truth boxes. These offsets when applied to the anchor boxes bring them closer to the ground truth boxes.

- We use L2 regression loss to learn the offsets. The anchor boxes are transformed using the predicted offsets and are called region proposals, and the network described above is called the region proposal network.
- This is stage 1 of the detector. Faster-RCNN is a two-stage detector. There is another stage.

Stage 2

- The input to stage 2 is the region proposals generated from stage 1. In stage 2, we learn to predict the category of the object in the region proposal using a simple convolutional network.
- Now, the raw region proposals are of different sizes, so we use a technique called ROI pooling to resize them before passing through the network.
- This network learns to predict multiple categories using cross-entropy loss.
- We use another network to predict offsets of region proposals from ground truth boxes. This network further tries to align region proposals with ground truth boxes.

- This uses L2 regression loss. Finally we take a weighed combination of both the losses to compute the final loss. In stage 2, we learn to predict both category and offsets. This is called multi-task learning.
- All of this happens during training. During inference, we pass the image through the backbone network and generate anchor boxes — same as before. However, this time we select only the top ~300 boxes that get a high classification score in stage 1 and qualify them for stage 2. In stage 2, we predict the final categories and offsets. In addition, we perform an extra post-processing step to remove duplicate bounding boxes using a technique called *non-max suppression*. If everything functions as expected, the detector recognizes and paints boxes over objects in the image as shown below: