Implementing YOLOV6 on PyTorch

Overview:

- Anchor free method, uses anchor points instead, No use of anchors in yolov6 version 1 but later introduced anchor aided training.
- The model predicts output having `num of classes+4 (or 5)` channels where there will be every channel for every class and 4 additional channel to directly predict `top, left, right, bottom` points. These points are scaled on the stride.
- In the ground truth assignment, ground truths are assigned based on the FPN, and not all boxes are assigned at all levels instead they are divided in range like 13x13 will be responsible for predicting boxes greater than size 256.
- Good idea is to use exponential in the regression prediction to avoid predicting negative bounding boxes.
- The architecture contains RepVGG blocks which are in inference converted into plain 3x3 blocks.
- The current implementation have inefficient Dataloader and also RepVGG block doesn't support conversion to plain 3x3 convolution and inference. It can be added by making few changes in RepVGG block.

Architecture Implementation

- The architecture of yolov6 small was implemented.
- Uses decoupled head.
- The architectural configuration:
- backbone_cfg =` {
  "width_mul": 0.5,
  "depth_mul": 0.33,
  "num_repeats": [1, 6, 12, 18, 6],
  "out_channels": [64, 128, 256, 512, 1024],
  }

head_cfg = {"channels": [512, 256, 128], "width_mul": 0.5}
neck_cfg = { "channels": [64, 128, 256, 512, 1024, 256, 128, 128, 256, 256, 512], # incudes backbone cfg upto 4th index
"num_repeats": [12, 12, 12, 12],
"width_mul": 0.5,
"depth_mul": 0.33,
} `

Implementation Details

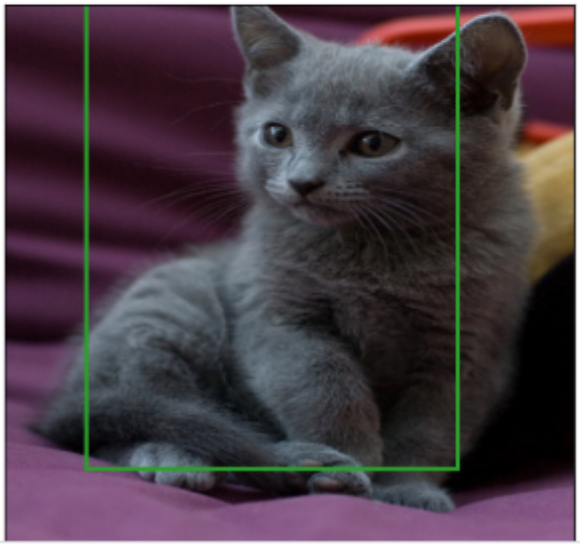
- SIoU loss and Varifocal loss were implemented as loss function. Additionally, CIoU loss was also tested. Additionally Centerness loss is also added.
- The Dataloader implementation currently uses two for loops to encode the ground truth bounding boxes, which is inefficient, All the conversion to tensor is done here.
- The head list of three outputs each with shape `BxSxSx(nc+5)`, where `nc` is number of classes.
-

Results

The model is overfitted on subset of 64 images to test the correct implementation and working. The results were then visualized on the same images.

- No of epochs : 70
- Learning rate: 0.0001
- Optimizer: Adam






Summary of the Work

- Implemented Model architecture, But conversion to normal 3x3 conv during the inference is not supported.
- Dataloader encodes ground truth bounding boxes into list of three tensors for three different levels.
- The loss functions are working and are well tested.
- Model overfits on the few images, so on full training model can learn.

References

- https://learnopencv.com/fcos-anchor-free-object-detection-explained/#Architecture-of-The-FCOS-Model
- https://github.com/meituan/YOLOv6

- https://arxiv.org/pdf/2209.02976