**Research Log**

**Sources:**

1. <https://www.mathworks.com/help/vision/ug/object-detection-using-yolo-v3-deep-learning.html>
2. <https://www.mathworks.com/help/vision/ug/estimate-anchor-boxes-from-training-data.html>
3. <https://www.geeksforgeeks.org/how-to-extract-frames-from-a-video-in-matlab/>
4. <https://www.mathworks.com/help/vision/ug/get-started-with-the-image-labeler.html>
5. <https://files.materovcompetition.org/2022/OER-2022%20ML%20_%20Computer%20Coding%20Challenge_FINAL.3.pdf>
6. <https://oceanexplorer.noaa.gov/okeanos/animal_guide/animal_guide.html>
7. <https://arxiv.org/pdf/2110.14711.pdf> (A Survey of Self-Supervised and Few-Shot Object Detection)
8. <https://arxiv.org/pdf/1804.02767.pdf> (YOLOv3: An Incremental Improvement)
9. <https://www.mathworks.com/help/vision/ug/get-started-with-the-image-labeler.html>

**Notes:**

* Downloaded images that were stored at <https://www.ncei.noaa.gov/waf/okeanos-animal-guide/images/> and extracted frames from videos provides in the challenge (5)
* Since all labeled data was classification data, decided to hand label some images for training and testing purposes
* Labeled ~30 frames from video frames in challenge using MATLAB labeler (9)
* Followed object detection with yolov3 model tutorial (1)
  + Note: To make sure that the code works, had to add all the supporting functions down at the bottom of the tutorial
  + Since in tutorial they download image and bounding box data instead of using image labeling gTruth object, had to slightly modify start of tutorial code
    - Images are stored at gTruth.DataSource.Source
    - Bounding Boxes are stored at gTruth.LabelData
  + Class names were manually inserted as cell array instead of at trainingDataTbl.Properties.VariableNames
* Initial training with ~30 video frame data led to less than desirable results as bounding boxes were inserted almost everywhere across the image
* Decided to take a combination of video frames and images from NOAA animal guide (6) to create a new data set of ~300 images
* Trained model for 80 epochs with a mini-batch size of 16 and a training testing split of 90% to 10%
* Model gave much most consistent bounding box predictions around creatures, however sometimes the model would lose a creature or place multiple bounding boxes around a single creature
* Model also started to get more stutter and less reliable once plants started appearing
* Decided to hand label 1000 images from animal guide (6) and train model longer to hopefully get better results
* Trained new model on 1000 images for 120 epochs with a mini batch size of 32 and a training testing split of 90% to 10%
* Model seems to be more confident with creatures that are in the center of the camera and has more trouble with creatures on border of image
* Model also has a rough time when it comes to coral and other sea plants
* After hearing presentation about self-supervised learning, decided to eventually look into to see if more training data can be obtained
* Tried removing class loss from the overall loss function, however this model became much less reliable than original 2 trained models
* Tried training model for shorter period to make sure model wasn’t overfitting with first 2, however this led to the model barely making predictions at all
* Tried to train model for 300 epochs overnight, however computer shut down sometime between falling asleep and waking up leading to a loss of training

**Models Trained:**

* Model 1
  + numEpochs = 80
  + miniBatchSize = 16
  + learningRate = 0.001
  + warmupPeriod = 1000
  + l2Regularization = 0.0005
  + penaltyThreshlold = 0.5
  + imageCount = 327
  + trainingSplit = 0.9
* Model 2
  + numEpochs = 120
  + miniBatchSize = 32
  + learningRate = 0.001
  + warmupPeriod = 1000
  + l2Regularization = 0.0005
  + penaltyThreshold = 0.5
  + imageCount = 1000
  + trainingSplit = 0.9
* Model 3
  + numEpochs = 50
  + miniBatchSize = 16
  + learningRate = 0.001
  + warmupPeriod = 1000
  + l2Regularization = 0.005
  + penaltyThreshold = 0.5
  + imageCount = 1000
  + trainingSplit = 0.9
  + NOTE: This model was training with the clsLoss constantly set to 0
* Model 4
  + numEpochs = 20
  + miniBatchSize = 32
  + learningRate = 0.001
  + warmupPeriod = 1000
  + l2Regularization = 0.0005
  + penaltyThreshold = 0.5
  + imageCount = 1000
  + trainingSplit = 0.9
  + NOTE: This model was to ensure that the stutters were not caused from the model overfitting to the training data
* Model 5
  + numEpochs = 160
  + miniBatchSize = 32
  + learningRate = 0.01
  + warmupPeriod = 1000
  + l2Regularization = 0.0005
  + penaltyThreshold = 0.5
  + imageCount = 1000
  + trainingSplit = 0.9

GITHUB Username: mboutell