Baseline implementation changelog

Dacia Sandero

Gabriel-Claudiu Stelea

W18015823

## Dataset

* Check dataset provided by the Solution Design mission
  + <https://data.gov.uk/dataset/bfc23a6d-8879-4072-95ed-125b091f908a/marine-habitats-and-species-open-data?fbclid=IwAR0XmKg1nkWhk0qaMcJHexquuJmO0a7G5LJdJfutVB5D2nIn8TRwKH-ouio>
  + Conclude that the dataset is unsuitable, as it only contains geographical data and not videos or images
* Search for dataset used in Academic Journal used in the Solution Design mission
  + <https://ieeexplore.ieee.org/document/8947884>
  + Did not find the dataset publicly available
* Search for alternative dataset on Kaggle
  + Found <https://www.kaggle.com/aalborguniversity/brackish-dataset>
* Separate video files into frames and add them together to a single directory
  + Error finding files in folder – solved by rewriting paths
  + Error setting up ffmpeg package
    - Tried manually installing in C drive and manually adding to system paths, did not work
    - Tried installing with Git, still not finding ffmpeg
    - Tried installing ffmpeg alternative library found at <https://github.com/kkroening/ffmpeg-python> ; did not work
    - Tried installing library via Anaconda, succeeded
  + Manually moving frames from 80 directories to a single directory

The dataset phase took roughly 11 days due to problems with the initial datasets and with the ffmpeg encoder. The choice of not generating alternative formats for the annotations was made due to some file formatting problems and due to the latest training algorithm already working with the COCO format.

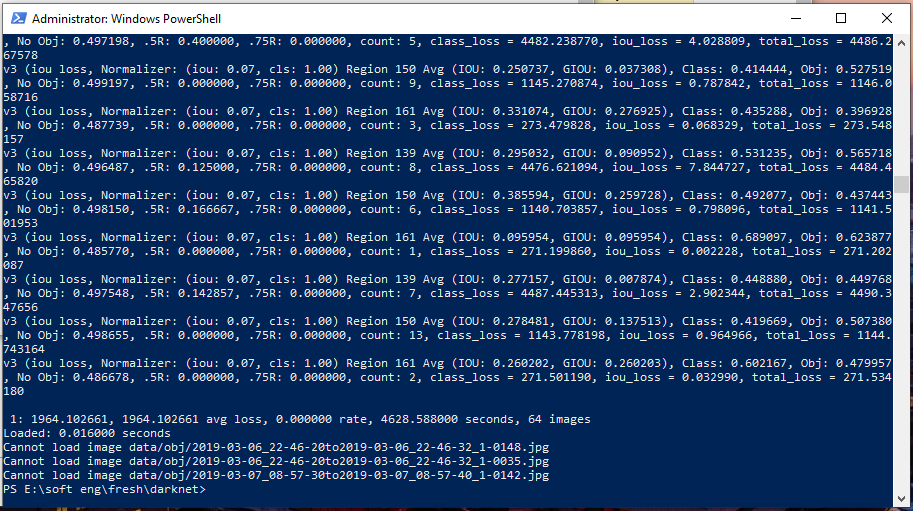
## Detection file

* Started looking for existing repos for yolo implementations
* Found Darknet implementation – most popular recommendation
  + <https://pjreddie.com/darknet/yolo/>
  + Multiple repos found linked to it
* First try, using <https://github.com/pjreddie/darknet>
  + Lots of confusion, multiple errors appearing
  + Annotations not working, need to convert from COCO to XML 14k files
  + Compiling errors, reported on the repo
* Second try, using <https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/>
  + Indentation errors
* Third try, using <https://github.com/experiencor/keras-yolo3>
  + See Training section
* Fourth try, using <https://towardsdatascience.com/object-detection-using-yolov3-using-keras-80bf35e61ce1>
  + Working for Detection, using a similar version from a lost repo
  + Slight changes in arguments
* Fifth try, using <https://github.com/AlexeyAB/darknet>
  + See Training section

The detection phase took roughly 2 weeks, part of which went at the same time with the training one. While not contributing much code-wise due to setup problems taking up too much time, it was needed as a provisional tool for the other missions to proceed, such as testing and API-GUI.

## Training

* Looking for models training in a similar format to the detection file
* Found <https://github.com/pjreddie/darknet>
  + Confusion, not sure how to properly tweak dataset
  + Dropped option
* Found <https://www.learnopencv.com/training-yolov3-deep-learning-based-custom-object-detector/>
  + Used tips from page to setup future configuration files
* Found <https://github.com/experiencor/keras-yolo3>
  + Problem with annotations from brackish dataset, need to convert COCO to vox
  + File explorer not finding files in the directory
  + Tried numerous ways of setting up the path, failed to run training
  + Used some elements from the detection file to tweak the one found for one file detection
  + Dropped development
* Found <https://github.com/AlexeyAB/darknet>
  + Error for paths to images, modified the train.txt file to contain correct path
  + Tried to make executable file using Cmake
    - Could not find Visual Studio, reinstalled and manually set windows path to installation folder
    - Could not find C++ development kit, reinstalled C++ packages in Visual Studio
    - Could not find Windows SDK kit, uninstalled from Virtual Studio, manually downloaded from the Microsoft website in a different directory and copied it to C:\ , added path to system
    - Could not run CMake in Windows Powershell due to restricted permissions, manually overridden user permissions and ran as administrator
    - Could not make executable with CMake due to several files missing, dropped approach
  + Tried to make executable by downloading repo using vcpkg
    - Error installing vcpkg, had to download Git and download it using command line
    - Error installing Darknet prerequisites due to OpenGL problems, had to reinstall C++ packages, video drivers
    - Error in detection of CUDA devices, Graphics card is an Nvidia GT710M with a Cuda grade of 2.0 and the lowest grade acceptable for learning is 3.4, dropped GPU training
    - Training keeps crashing and detecting missing files in the images folder, disabled antivirus after 10 hours of troubleshooting
  + Average training time 1.5 hours, average loss 1964 (when max value should be 3)
    - Added folder with image samples from training
    - Contacted lecturers for help in configuration
    - Tried setting resolution to native one, error
    - Tried using alternative configurations such as CRNN, Alexnet, tiny-yolo and older darknet versions, no boxes appearing around objects, program crashing
    - Dropped development



Screenshot from a successful run of training, 1.5 hours on 64 images with huge loss

* Tried to port Yolo V4 darknet to Google colab
  + Error while uploading dataset to google drive, too many files, solved
  + Error using CMake due to different notations on google colab to the linux ones
  + Missing files in CMake
  + Dropped development

The training phase took roughly 1 month and, while teaching a lot about the process and how complicated the project will end up being, has proven unsuccessful. Did not include environment setup steps in the readme file due to failure of training and incredibly confusing and complicated process.

For the future, planning to investigate on using

<https://github.com/EdjeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10>

and

<https://github.com/KerasKorea/KerasObjectDetector>

or migrating to a Linux device for ease of setup.