Assignment 1

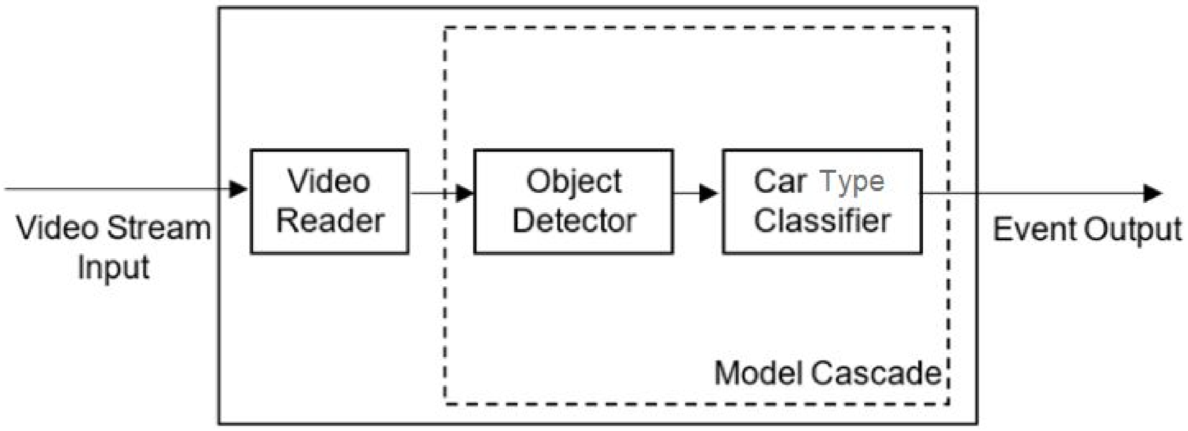
Real-time Object Detection and Classification

Due Date: 18/3/2021 17:00 (Strict)

Demo Date: 19/3/2021 (08:00-11:00 and 13:00-15:00)

Submission via Blackboard

Design and implement a computer vision model pipeline and perform matching to identify events from a given video stream. Figure 1 shows the high-level block diagram architecture of the pipeline. The pipeline consists of a video reader and a model cascade. The task is to detect object events (car) with a specific attribute (car type). The pipeline can be developed in Python (highly recommended) or Java [1].



Video: The given video clip (duration 30 seconds) is of a road cross-section.

Pipeline Design: Implement a pipeline that processes the video frames using a DNN model(s).

Pipeline Input: Stream the video stream into the system by using a video reader (fig. 1) at a rate of 30 frames per second. This requires reading the video file and simulating a stream of image frames as individual data items.

Model Cascade: The model cascade will be a 2-stage computer vision pipeline. The following steps need to be performed to build the model cascade:

[1] For java users: https://deeplearning4j.org/ library consists of TinyYolo and Mobilenet model.

- **Stage 1: Object Detector:** Deploy a state-of-the-art object detector model (TinyYolo), pre-trained on PASCAL VOC or MSCOCO dataset. Tiny YOLO model and weights can be downloaded here: https://github.com/qqwweee/keras-yolo3/ https://pjreddie.com/darknet/yolo/
- **Stage2- Attribute Classifier (Car Type):** Train an existing pretrained object classifier (Mobilenet from Keras library: https://keras.io/applications/ or DeepLearning 4j for Java) using a transfer learning approach for two car classes:
  - SUV
  - Sedan

The input to this attribute classifier will be the Region of Interest obtained from Stage-1.

- **Dataset Preparation for Stage 2:** Students need to create a small dataset of images to train the classifier. The more images, the better the classifier will be trained. It's recommended to download at least 400 images (SUV and Sedan) (scrape it from google using any image downloading api) for each class and divide the dataset into 80:20 ratio] for training and testing. Split the training set into 80:20 ratio for training and validation set.

Queries: Two queries need to execute after deploying the model pipeline. The queries are lined with their increasing complexities.

- **Q1 [Object Detection Task]:** Identify and count the number of cars in each frame.
- **Q2 [2-Stage Model Cascade- (Object Detection Attribute Classifier (Car Type))]:** Identify and count the cars and their types in each frame.

Optimisation: Implement a processing flow optimization of the pipeline and the model cascade for Q1 and Q2. One option is to use a producer consumer, where the producer thread continuously streams video frames and sends it to an internal queue from where the consumer (DNN models) read the frames from the queue and process them.

Output and Results Evaluation: Record your results in the format given below and compare your results to the ground truth.

For Q1 and Q2 do the following:

- Compare the count with ground truth and report the accuracy (F1 Score).
- Report the training accuracy of the car type classifier.
- Report the throughput of Q1 and Q2.For the evaluation, the experiment section of VidCEP paper can be useful to consult.
- Report on the effects of your flow optimisation for Q1 and Q2.

Code (50 Marks)

- Pipeline Design
- Dataset Preparation (download 500 images of each class)
- Preprocessing
- Deploying the pre-trained SOTA Object Detector [TinyYolo]
- Car Type Classifiers

- Model Cascade Deployment
- Pipeline Optimisation
- Comments to explain your source code. Insufficient comments will lead to mark deductions.

Report (50 Marks)
You are required to submit a short report (approximately 4-5 pages including references) specifying:

- **Pipeline Design:** A description of your approach and design decisions. This should include details on the flow of the pipeline, details of video reader, object detector and classifier. Justify your design decisions. Where appropriate include appropriate diagrams and references to research papers to support your arguments.
- **Pipeline Output:** Detail the output of your pipeline with all the graph results (event extraction, throughput etc).
- **Model Training Evaluation:** Description about how the classifier model is trained which includes dataset preparation to transfer learning approach and training process. Report the accuracy of the trained model.
- **Pipeline Optimisation:** A description of how you have optimized the execution of the pipeline and the Model Cascade. Report the improvement of the optimised pipeline. Justify your design decisions.
- **Design Strengths and Weaknesses:** Detail the Strengthens and Weaknesses of your approach Justify your design decisions. Where appropriate include references to research papers to support your arguments. (minimum 1 Page)
- **Download Link for Submission Files:** Provide a download link for the ZIP archive containing your demo video, code and the CSV file with complete results.

Groups and Individual

The assignment may be completed as a group. Each group can have a maximum of 2 students. Groups are self-selected. Please email your group (students name) at jaleed.khan@insight-centre.org by the end of day Friday 19th Feb. If you cannot form a group or have any other doubt, please connect with Jaleed at the above mail.

There is also the option to complete the assignment as an individual. Please email jaleed.khan@insight-centre.org to confirm you will complete it as an individual by the end of day Friday 19th Feb. For students who have not contacted Jaleed, we assume you will complete it as an individual.

Grading will be appropriate for individual and group submissions.

Submission Instructions

- **Report Submission:** Please put your report into a single .pdf file with name(s) "Name(s)_Assignment1_report.pdf", submit via Blackboard.

- Code: Submission: Put all parts of your submission into a single .ZIP archive with the name “Name(s)_Asignment1.zip”, upload the .ZIP file to your university OneDrive account and provide a download link for the ZIP file in your assignment report.
    - The ZIP file should contain a demo video, complete code and the CSV file having your complete results.
    - The demo video should show the final result of your pipeline for query Q2. i.e. you have to display the car count in each frame and annotate each car with type. You can use the built-in OpenCV functions for drawing bounding boxes and adding text to the frames. Name your video as “Name(s)_Assignment1_video”.
    - Include any file or component which is needed to re-run the assignment such as scripts and instructions. Please do not include any database internal files.
    - Include all source code files (that is, files with name ending .py or .java etc) required to compile and run your code.
    - Include the resulting dataset of the output of your pipeline.
- Please note that all submissions (both code and report) will be checked for plagiarism. Plagiarism (from another student, group, or any other sources) are not allowed and will be treated seriously.
- Please verify the submitted files. Mistakes cannot be corrected once the deadline has passed.

---

[1] For java users: https://deeplearning4j.org/ library consists of TinyYolo and Mobilenet model.

Common Questions on Assignment
If you have questions on the assignment please email

Q: Can you provide more details on the Producer Consumer?
A: Please the following tutorials for more details on this pattern. Tut1 and Tut2

Q: Is the internal communication of the pipeline must happen via ActiveMQ or I can pass the data as parameters in the method calls.
A: Treat it like a producer consumer problem, where the producer thread continuously streams video frames and sends it to an internal queue from where the consumer (DNN models) read the frames from the queue and process them. (See this and this for reference) There is **no requirement** for any third party messaging services like ActiveMQ.

Q: If a frame doesn't have a car, should it be considered while calculating f1-score of the system?
A: If the frame doesn’t have a car, it should be considered for calculating f1-score for Q1 only and not for Q2.

Q: Where can I download the video and groundtruth files?
A: Follow these links to download the files: Video, Groundtruth