SIT789 - Applications of Computer Vision and Speech Processing

High Distinction Task 5.2: Minor research project

Objectives

The objectives of this lab include:

- Reading and understanding a research paper in Computer Vision/ Speech Processing
- Performing literature review
- Programming upskilling
- Getting some experience on technical writing

Tasks

You are given a list of research papers with references for implementation (see the table below). This is individual task and thus each student will need to select a paper from the list. However, you are always welcome to recommend other papers/topics. In case that you have your own papers and would like to use them for this task, you should consult with the Lecturer/Unit Chair to make sure your recommended papers align with the Unit.

After choosing a paper, you need to read and understand the paper. In order to understand a paper, you may need to read other related papers and/or do further research. This is an opportunity for you to expand your knowledge and get updated with the literature of the chosen topic.

You also need to run the code implementing your selected paper. Note that all the papers in the list are accompanied with code. However, you are free to search and use other implementations. If you want to use your own paper, make sure that the paper's code is available or you can reimplement it.

Your tasks include,

1) Writing a report to summarise your selected paper (**7 marks**). Your report should satisfy the following criteria.
    a. The report should reasonably summarise the paper/topic chosen (**4 marks**)
    - The report should be comprehensive and provide enough details – 2 marks
    - Technical information is presented accurately – 2 marks
    b. The report should have a good presentation (**2 marks**)
    - The report should have a reasonable layout (students are free to choose any layout and do not necessarily need to follow the layout of the paper chosen) – 1 mark
    - The report should have a reasonable length (from 2,000 – 2,500 words depending on the paper/topic selected) – 1 mark
    c. References (**1 mark**)
    - The paper should provide a list of references. Students can choose any style for the references (e.g., ACM, IEEE, etc.). However, all the references should be presented consistently in one style (e.g., the references should not mix-up both ACM and IEEE).
    - The references may not be exactly same as those of the paper chosen but should cover enough important work related to the paper/topic chosen.

2) Running code (**3 marks**).
    a. To demonstrate the work in your chosen paper, you should collect a small dataset (e.g., just a few images are enough), and then run the code on your collected data. You could use sample data provided in the original paper. However, you should test the code on this sample data rather just simply copy results shown in the paper.
    b. You need to report several qualitative results (i.e., visual effects, e.g., images, not statistical numbers). You should also describe the results and draw conclusions based on your own observations.
    c. There is always room for improving/enhancing methods proposed in the paper (e.g., improving the accuracy, enhancing the speed). Therefore, it would be great if you could identify limitations of the work presented in the paper and suggest improvements. Note that, your proposed improvements should be specific and clear enough to be implemented though you are not required to implement them. Your proposed solutions should also be relevant to the Unit, e.g. improvements made by using more powerful hardware are not relevant to the Unit.

Submission instructions

1. Submit your report and results to OnTrack.
2. Submit your code to OnTrack.

Papers	Topics	Source code	Notes
Computer Vision			
ImageNet classification with deep convolutional neural networks	Image recognition	C++ (Cuda, Cafe, Tiny) Python	There are various implementations of this work. I just list here a few and you are also free to search and use any other implementations. For simplicity, I recommend Tiny and Python version
Deep residual learning for image recognition	Image recognition	Python	There are various implementations of this work. You are also free to search and use any other implementations.
YOLO9000	Object detection	Python, C++	Further information can be found at https://pjreddie.com/darknet/yolo
Fast R-CNN	Object detection	Python, C++	
Faster R-CNN	Object detection	Python, Matlab	
Realtime multi-person 2D pose estimation using part affinity fields	Human pose estimation	Python, C++, Matlab	
Mask R-CNN	Object/instance segmentation	Python	Another version can be found here https://github.com/matterport/Mask_RCNN
SegNet	Semantic segmentation	Python, C++, Matlab	
JSIS3D: joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields	3D scene understanding	Python, C++	
Speech Processing			
DeepSpeech: scaling up end-to-end speech recognition	Speech recognition	Python	
SpecAugment: a simple data augmentation method for automatic speech recognition	Speech recognition	Python	
Wav2vec: unsupervised pre-training for speech recognition	Speech recognition	Python	
Jasper: an end-to-end convolutional neural acoustic model	Speech recognition	Python	
3-D convolutional recurrent neural networks with attention model for speech emotion recognition	Speech emotion recognition	Python	
An interaction-aware attention network for speech emotion recognition in spoken dialogs	Speech emotion recognition	Python	
Speaker recognition from raw waveform with sincnet	Speaker recognition	Python	
Voice filter: targeted voice separation by speaker-conditioned spectrogram masking	Speaker recognition	Python	