**CTI One Corporation**

3679 Enochs St, Santa Clara, CA 95051

**AI Tech Camp 2019 Syllabus**

|  |  |  |
| --- | --- | --- |
| 2019-9-06 | Create this document | HL, PR, MO, YL |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Description | Speaker | Note |
| 1 | **1.1 Using TF Keras Tools for AI Deep Learning Applications**  Installation of TF Keras and running your first program for hand-written digits recognition | PR |  |
| 2 | **2.1 OpenCV Tools and Techniques for Deep Learning (Part 1)**  a. Installation of OpenCV.  b. OpenCV C++/Python:  b(1). Read and display images.  b(2). Read and save live videos.  b(3). Read and save video files.  c. Image augmentation techniques to prepare deep learning dataset.  d. Image online harvesting techniques. | PR |  |
| **2.2 OpenCV Tools and Techniques for Deep Learning (Part 2)**  a. OpenCV C++/Python canny edge detection on given:  a(1). Image  a(2). Live video  a(3). Video file | PR |  |
| **2.3 Computer Vision technique for ROI manipulations**  Objectives:  2.3.1. Digital Image definition and coordinate system of Image.  2.3.2. Use OpenCV to define rectangle ROI;  (1) Define ROIs  (2) Read image pixels from the ROIs  (3) Write image pixels to ROIs  (4) Write captions (text) on ROIs  2.3.3 Use OpenCV to define polygonal shape ROIs;  (1) Define the polygonal shape ROI;  (2) Read image pixels from the ROIs  (3) Write image pixels to ROIs  (4) Write captions (text) on ROIs  2.3.4. Use OpenCV to define rectangle ROI;  2.3.5. Use OpenCV to define polygon shape based on contour analysis;  (1) Write a program to display the processed result (display on top of the original image)  (4) Write captions (text) on the image contour . Another reference: <https://github.com/hualili/CMPE297/blob/master/2018S-15-Contour-Inference-final-2018-4-12.pdf> | MO |  |
| **2.4 Feature Extraction Based on Canny**  Objective:  2.4.1. OpenCV C++/Python  (1) read test image pattern then Canny edge detection, then display both original image and the edge map (Canny result); Test image: https://github.com/hualili/CMPE297/blob/master/2018S-23-test1.jpg  https://github.com/hualili/CMPE297/blob/master/2018S-23-4contourTesting.jpg  (2) read test live video then canny edge detection, then display both original live video and live edge map (Canny result);  (3) read video file then canny edge detection, then display both original live video file and edge map of the video file (Canny result);  2.4.2. OpenCV C++/Python  (1) save canny image result, save (2) video Canny result; | MO |  |
| **2.5 Feature Extraction Based on Contours Analysis**  Objectives:  2.5.1. OpenCV to compute contours  (1) Use the reference below, write OpenCV program to compute contours for the test image listed above. Reference:  https://github.com/hualili/CMPE297/blob/master/2019S/2019S-25-contourAnalysis-2018-9-13.pdf  (2) Compute each type of 4 different types of contours, and understand the difference between them;  (3) Read contours data, and display partial contours based on contours' size;  (4) Primitive contour properties. | MO |  |
| **2.6 Contour based ROI localization for Object Detection.**  a. Case study on handwritten digit recognition. | PR |  |
| **2.7 Use ROI localization for Better Handwritten Recognition**  a. Combine OpenCV based techniques with handwritten recognition. | PR |  |
| **2.8 Deep Learning based ROI localization for Object Detection.**  a. Utilization of Pedestrian Detection for ROI localization. | PR |  |
| **2.9 Case study**  a. Use Keras to test your handwritten digits. | PR |  |
| **2.10 Deep Learning based technique for ROI localization.**  a. Use pedestrian-based ROI localization and YOLO based ROI localization. | PR |  |
| 3 | **3.1 Case Study: Cell phone detection.** | LY |  |
| **3.2 Case Study: garbage detection**  3.2.1 Computer Vision algorithm to detect garbage developed by CTIOne Corp.  3.2.2 Show a demo on garbage detection. | MO |  |
| 4 | **4.1 Fine Tuning Techniques:**  4.1.1 Read information from the CNN basic building blocks (C2N for example);  4..1.2 Visualization of the information from the basic building blocks (C2N); | LY |  |
| 5 | **5.1 Deployment techniques:**  5.1.1 Save training result;  5.1.2 Loading trained result  5.1.3 Selective training for limited layers by UN-freazing techniques | LY |  |
| 6 | **6.1 Hardware Aspect for the Deployment**  Objectives:  6.1.1 Introduce the hardware platforms for Deep Learning program.  6.1.2 Pros and Cons for selection of right platform. | PR, MO |  |

(end)