ECE 449/ECE595 Machine Learning

Term Project

## ECE 449: Option 1: Vehicle image generation

- The vehicle images on Brightspace are manually segmented from images/videos acquired by the security cameras in a parking lot (http://web.inf.ufpr.br/vri/databases/parking-lot-database/).
- Each image in the dataset contains one vehicle.
- Rescale the images to the same matrix size if needed. Vehicles parking closer to a camera look bigger.
- Dataset filename: Parking_Lot_Vehicle_Images_UFPR04.zip

### Tasks

1. Reference the implementation of vanilla GAN for handwriting 0-9 digit generation using MNIST dataset. For example:
    - https://github.com/SudharshanShanmugasundaram/Vanilla-GAN/blob/master/GAN.ipynb
    - https://github.com/safwankdb/Vanilla-GAN/blob/master/Vanilla_GAN.ipynb

    Note: You need to add citations in the References section in your report or Jupyter Notebook file.

2. Implement your own generative adversarial network.
    Note: If you want to merge or modify code from other people's work, you need to read the license first and strictly follow the license. Clearly credit other people's work in your project.

3. Prepare training data and train your network.

4. Manually evaluate the performance of the trained GAN generator.
    The simplest way for performance evaluation is to mix a certain number of real and generated images and ask one or more people to identify real or generated images. Use accuracy or confusion matrix to evaluate the performance.

    Note: If you are interested in more objective GAN evaluation measures, read the following articles:

    Tim Salimans, et. al. "Improved Techniques for Training GANs"
    (Link: https://arxiv.org/pdf/1606.03498)

    Shane Barratt, et al. "A notes on the inception score"
    (Link: https://arxiv.org/pdf/1801.01973)

The implementation of inception score (IS) described in Salimans and Barratt's papers can be found at https://github.com/openai/improved-gan

For more GAN evaluation measures, see Ali Borji, "Pros and Cons of GAN Evaluation Measures"
(Link: https://arxiv.org/abs/1802.03446)

5. The Jupyter Notebook or your report file should contain a brief description of the project, the dataset, the network structure, training detail, intermediate output, fully trained generator output, and evaluation method and result. Add discussions and conclusions if needed.

6. The vehicle image dataset is suitable for training on a mainstream computer without GPU acceleration. It is encouraged to use an image dataset with higher resolutions if GPU resources are accessible for your group. However, no extra credit for a project implemented and trained on GPUs. (Free GPU resources: Colab. https://colab.research.google.com/).

7. Submit your jupyter notebook files or your Matlab file and report to BrightSpace

**ECE 595: Option 1: Find Swimming Pools in Fort Wayne**

This project is to use the Google Maps API to retrieve satellite images of Fort Wayne, and apply YOLO (You Only Look Once) object detection to identify swimming pools in the area.

**Tasks:**

1. Set up and generate Google Map API Key.
   (Reference: https://developers.google.com/maps/documentation/embed/get-api-key)
   **Important: Google Maps API has a usage limit. You can setup a usage quota in the Cloud Console to avoid exceeding the limit. The Department of ECE may not be able to reimburse the cost associated with Google Maps API usage.**

2. Write utility functions of retrieving satellite images using Google Maps API.
3. Select a YOLO for object detection (higher YOLO version and pretrained YOLO models are preferred).
4. Fine-tuning the pretrained model is required. A small dataset of satellite images with pool annotations should be prepared and used for improving object detection performance.
5. You program should
   a. overlay a bounding box on each swimming pool found by your algorithm,
   b. create a list of swimming pool locations and sizes,
6. Create a test dataset. Compare detected pools against actual pool locations to evaluate your model performance.

7. Submit your Jupyter Notebook files or your Matlab file and report to BrightSpace. The Jupyter Notebook or your report file should contain a brief description of the project, the dataset, the network structure, training detail, testing and evaluation methods, intermediate and fully trained output, and results. Add discussions and conclusions if needed.

**ECE 449 and ECE 595: Option 2: Choose your own project**

- You need to prepare a short (within one page) project description, and make an appointment with me to discuss goals, data and methods.
- The project description should briefly explain the project objectives, how to get data, and network design and evaluation.

**Tasks**

1. Prepare your data
2. Implement your project
3. Train, validate and test your implementation.
4. Evaluate the performance
5. The Jupyter Notebook or your report file should contain a brief description of the project, the dataset, the network structure, training detail, testing and evaluation methods, intermediate and fully trained output, and results. Add discussions and conclusions if needed.
6. Submit your Jupyter Notebook files or your Matlab file and report to BrightSpace