DEEP LEARNING FOR COMPUTER GRAPHICS

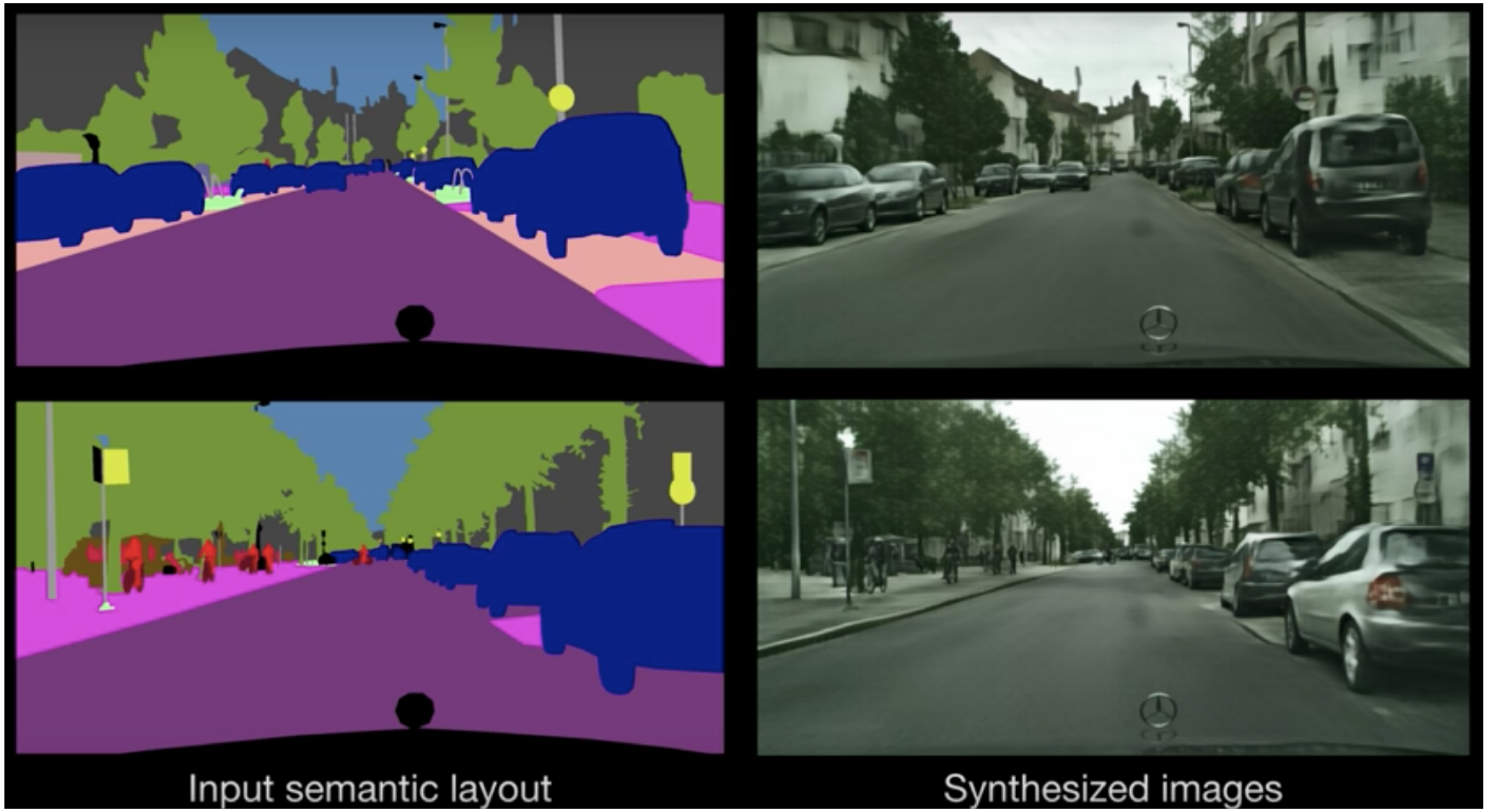
Final Project

Presented By: Saurabh Kumar

Objective

- To produce an image with photographic appearance that conforms to the input layout
- Input:
    - The input layout is given in the form of a semantic label map
    - Semantic classes are coded by color

Expected Input – Output Example



Proposal

- Paper:
  - Photographic Image Synthesis with Cascaded Refinement Networks
- Authors:
  - Qifeng Chen, Vladlen Koltun

Approach Presented

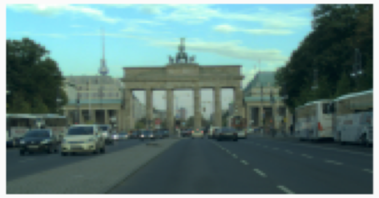
- Synthesizing photographic images by a direct supervised learning of single feedforward convolutional network trying to minimize regression loss
- Works seamlessly for high image resolutions (2 megapixels)

Training Dataset

- Inverse semantic segmentation
- Cityscapes dataset
    - https://www.cityscapes-dataset.com/
    - 2048X1024

gtFine_trainvaltest.zip (241MB) [md5]

fine annotations for train and val sets (3475 annotated images) and dummy annotations (ignore regions) for the test set (1525 images)



leftImg8bit_trainvaltest.zip (11GB) [md5]

left 8-bit images - train, val, and test sets (5000 images)

Architecture

- Consists of Cascaded Refinement Network (CRN)
- Each module M_i operates at a given resolution
- Resolution is doubled between successive modules

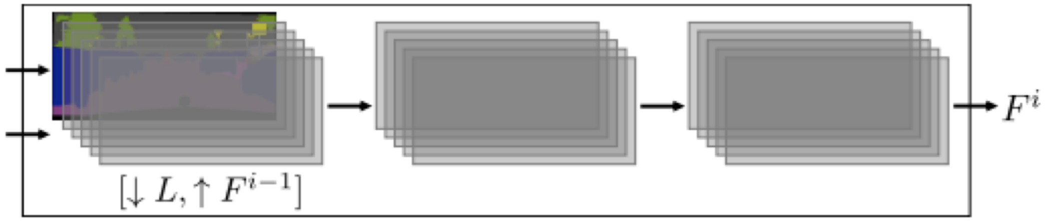


Figure 3. A single refinement module.

- Each module has 3 layers:
  - Input layer: w_i x h_i x $(d_{i-1} + c)$
  - Intermediate layer: w_i x h_i x d_i
  - Output layer: w_i x h_i x d_i

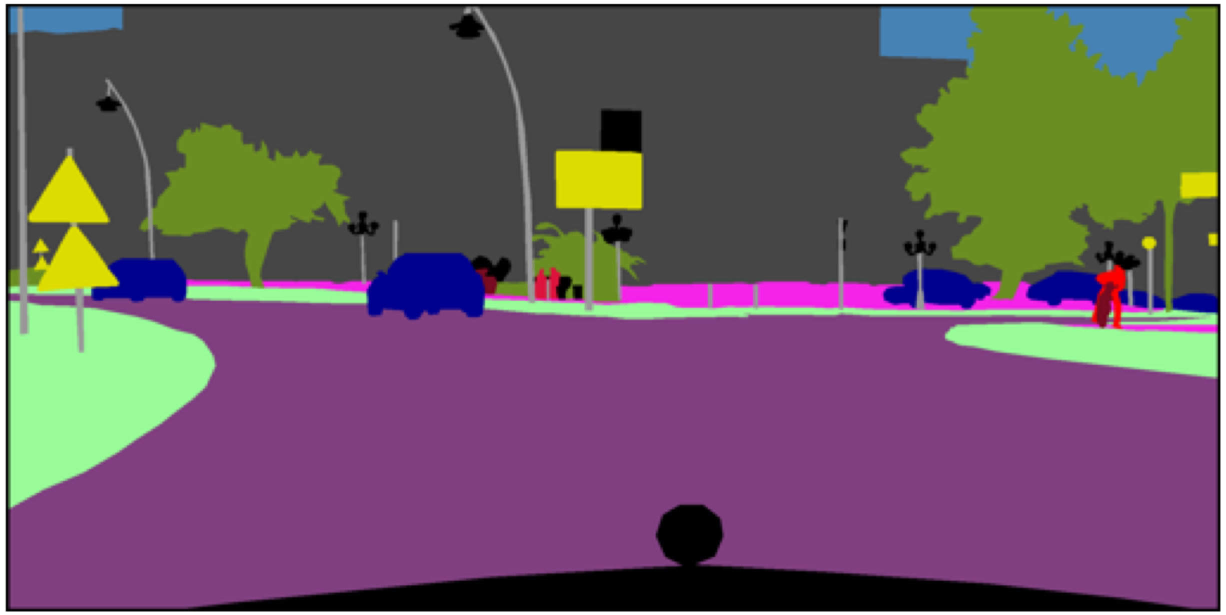
Architecture

- Each layer is followed by 3x3 convolutions, layer normalization & LReLU non-linearity
- The output layer of final module is not followed by above layers. A linear projection is applied to map F^i to generate output color image (w_i x h_i x 3)
- Total number of refinement modules depends on the output resolution
- Number of refinement modules for 8x4 to 512x256 is 7

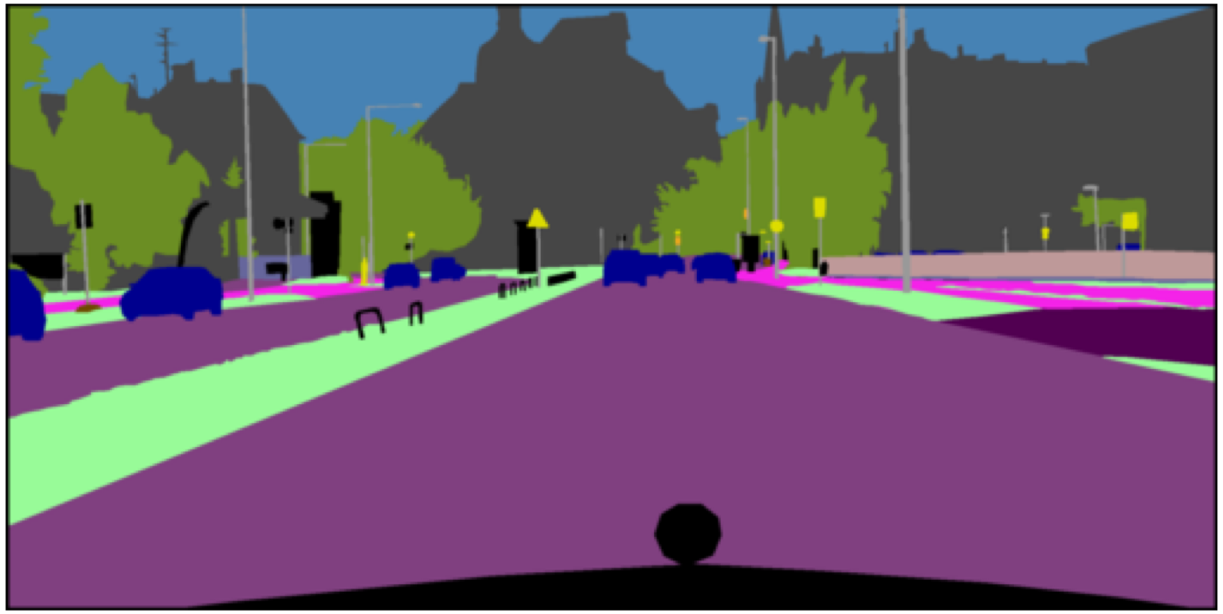
Challenges

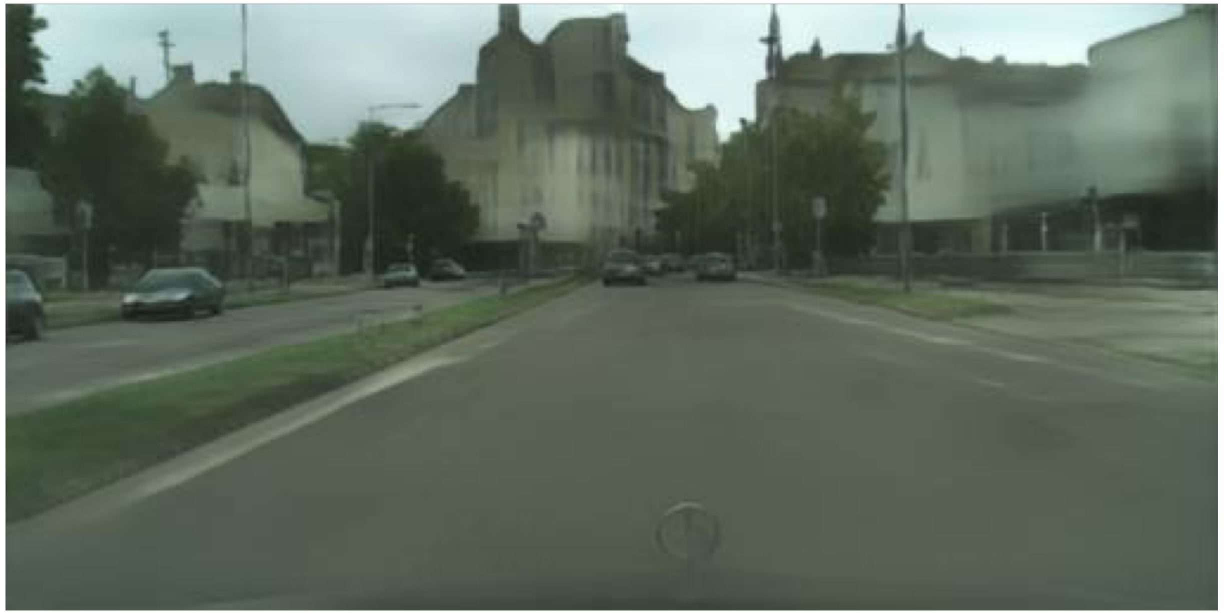
- Coding Challenges
- Execution Challenges
    - AWS
    - Google Colab
    - TAMU HPRC
    - Google Cloud Platform

Results



Results





Thank You