Auto Segmentation

Will LeVine & Cole Morgan
Department of Computer Science, Rice University

Background

- used U.S. car market generates over $100 billion annually and is quickly growing
- used car matching process is inefficient because selective online searches require car attributes to be manually labeled (e.g. color)

Goal

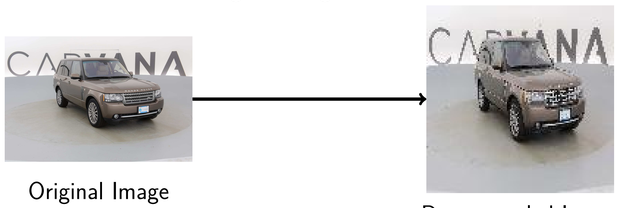
- reduce inefficiency present in the process of searching in the online used car market, improve user experience
- develop method to isolate an image of a car from its surrounding background
- train model to cluster groups of cars based off certain features (e.g. color) to expedite the used car searching process

Issues and Where We'd Like To Go

- would purchase extra compute power so that we could maintain image quality of the cars
  - ▷ couldn't do that for this project because of limited computing power and financial resources
  - ▷ instead downsampled from the images (more information below)
- would extend the clustering algorithm to instead be a recommendation system
  - ▷ couldn't do that for this project due to the inavailability of user-preference data

Data Downsampling

- we were limited by memory and computing power
- to remedy these issues, we downsampled all the input images and labels from 1918x1280 to 128x128 using Pillow's default image resizing



Original Image

Downsampled Image

UNet

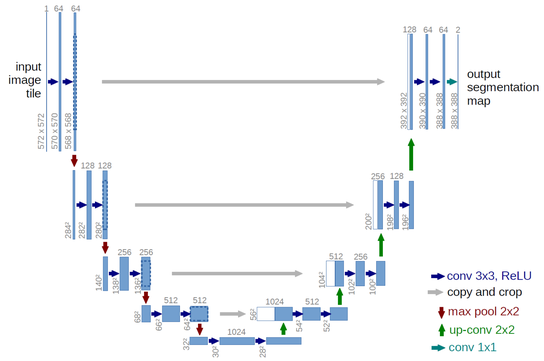
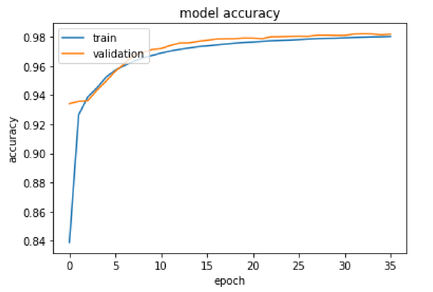
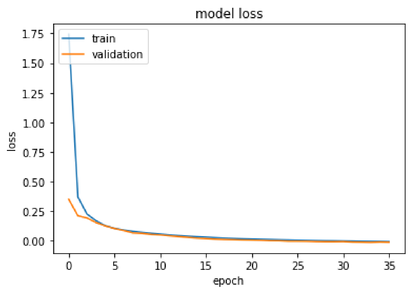


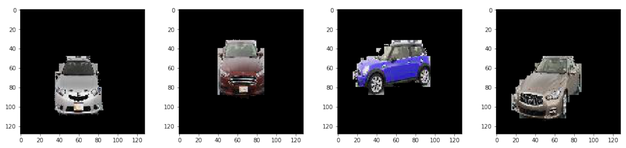
Figure: UNet Architecture



(a) UNet Train/Validation Accuracy History



(b) UNet Train/Validation Loss History



(c) Example 1 (d) Example 2 (e) Example 3 (f) Example 4

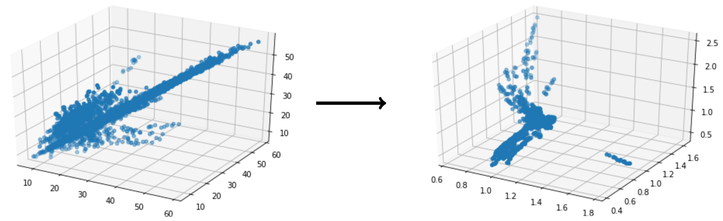
Figure: UNet Output Examples

UNet Modifications

- added Early Stopping callback with patience of 3
- added 9 dropout layers with hyperparameter of .5 (per suggestion of Geoff Hinton) to reduce overfitting
  - ▷ Train F1: 0.966
  - ▷ Test F1: 0.966
- added batch normalization to all 2D convolution blocks to unify feature scales
  - ▷ training F1
    - without batch normalization: 0.876
    - with batch noramlization: 0.966
  - ▷ testing F1
    - without batch normalization: 0.831
    - with batch noramlization: 0.966

Example Clustering Method: Automobile Color

- Goal: cluster automobiles based on RGB values. Used car sites provide search narrowing through color selection, so we would like to provide a serivce that autonomously identifies the color of the car
- Issue: RGB values are distributed in lines radiating from the origin
- Solution: Cluster based off of RGB ratios

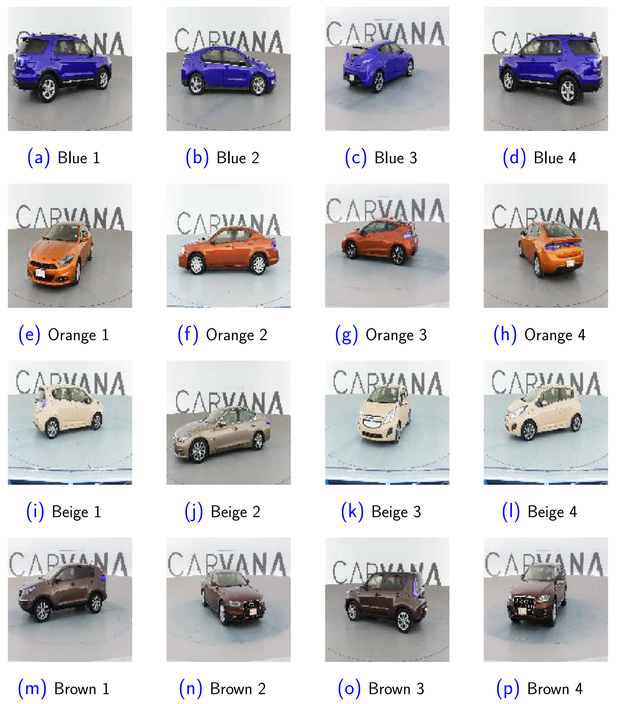


RGB Plot RGB Ratio Plot

Clustering Approach

- we used DBSCAN on the RGB ratios
  - ▷ we wanted the algorithm to be able to select the number of colors because we were unsure of how many colors there were in the dataset
  - ▷ we wanted to the algorithm to be able to identify when a data point is noise to alert the user that they must label the color of the car because the algorithm is unsure
- Number of Clusters: 12
- Number of Noise Points: 699 out of 5088

Clustering Examples



(a) Blue 1 (b) Blue 2 (c) Blue 3 (d) Blue 4

(e) Orange 1 (f) Orange 2 (g) Orange 3 (h) Orange 4

(i) Beige 1 (j) Beige 2 (k) Beige 3 (l) Beige 4

(m) Brown 1 (n) Brown 2 (o) Brown 3 (p) Brown 4

Figure: Example Color Clusters. Each row corresponds to a cluster

wvl1@rice.edu,cmm16@rice.edu