Capstone Project Proposal

Ananthesh J Shet

Machine Learning Engineer Nanodegree

31-Oct-2017

Object Classification Using Convolutional Neural Network For Reverse Parking

1. Overview

Reverse parking accidents are very common and involve damaging other vehicles and injuring/killing people. According to National Highway Traffic Safety Administration(NHTSA)[1], over a period of 3 years(2012-2014), an average of 1,898 people were killed in non-traffic crashes. Nonoccupants(such as pedestrians and bicyclists) accounted for 34% of these people – 42% of whom were killed by vehicles moving forward and 35% by vehicles backing up. An average of 92,000 people were injured in non-traffic crashes. Nonoccupants accounted for 33% of these people – 49% of whom were injured by vehicles moving forward and 40% by vehicles backing up.

Today's vehicle are fitted with rear-view cameras to assist the driver with reverse parking. There are backup aid systems too which provide visual and audible warning whenever the system detects an object behind the vehicle. These information are only as useful as the attentiveness of the driver as it is ultimately the driver who as to take corrective measures such as braking.

The system can be further improved by letting the system to automatically take corrective measures if the driver doesn't respond within a set time interval from the time an object has been detected. This will greatly help in reducing the number of accidents during backing up.

2. Problem Statement

For the system to take the right corrective measures at the right time, the system has to correctly identify/classify the objects. If the system thinks that there is an object when in reality there is not, then it is termed as False Positive(FP). If the system thinks there is no object when in reality there is, then it is termed as False Negative(FN). System with high FP leads to unnecessary braking causing discomfort to the driver/passengers. System with high FN leads to more number of accidents. This project aims at designing a classification system with low FP and FN.

---

[1] https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812311

3. Dataset

For this project, I will use the publicly available dataset from CIFAR-100[2]. The CIFAR-100 dataset consists of 60000 32x32 color images in 100 classes with 600 images per class. There are 500 training images and 100 testing images per class. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. Each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the super class to which it belongs).

Here is the list of classes in the CIFAR-100:

#	Superclass	Classes
0	aquatic mammals	beaver, dolphin, otter, seal, whale
1	fish	aquarium fish, flatfish, ray, shark, trout
2	flowers	orchids, poppies, roses, sunflowers, tulips
3	food containers	bottles, bowls, cans, cups, plates
4	fruit and vegetables	apples, mushrooms, oranges, pears, sweet peppers
5	household electrical devices	clock, computer keyboard, lamp, telephone, television
6	household furniture	bed, chair, couch, table, wardrobe
7	insects	bee, beetle, butterfly, caterpillar, cockroach
8	large carnivores	bear, leopard, lion, tiger, wolf
9	large man-made outdoor things	bridge, castle, house, road, skyscraper
10	large natural outdoor scenes	cloud, forest, mountain, plain, sea
11	large omnivores and herbivores	camel, cattle, chimpanzee, elephant, kangaroo
12	medium-sized mammals	fox, porcupine, possum, raccoon, skunk
13	non-insect invertebrates	crab, lobster, snail, spider, worm
14	people	baby, boy, girl, man, woman
15	reptiles	crocodile, dinosaur, lizard, snake, turtle
16	small mammals	hamster, mouse, rabbit, shrew, squirrel
17	trees	maple, oak, palm, pine, willow
18	vehicles 1	bicycle, bus, motorcycle, pickup truck, train
19	vehicles 2	lawn-mower, rocket, streetcar, tank, tractor

The superclasses and classes are numbered in alphabetical order as shown below:

```
***********************CIFAR-100 Superclasses***********************
 0 aquatic_mammals                  1 fish
 2 flowers                          3 food_containers
 4 fruit_and_vegetables             5 household_electrical_devices
 6 household_furniture              7 insects
 8 large_carnivores                 9 large_man-made_outdoor_things
10 large_natural_outdoor_scenes    11 large_omnivores_and_herbivores
12 medium_mammals                  13 non-insect_invertebrates
14 people                          15 reptiles
16 small_mammals                   17 trees
18 vehicles_1                      19 vehicles_2
```

[2] https://www.cs.toronto.edu/~kriz/cifar.html

```
************************************CIFAR-100 Classes************************************
 0 apple             1 aquarium_fish    2 baby             3 bear             4 beaver
 5 bed               6 bee              7 beetle           8 bicycle          9 bottle
10 bowl             11 boy             12 bridge          13 bus             14 butterfly
15 camel            16 can             17 castle          18 caterpillar     19 cattle
20 chair            21 chimpanzee      22 clock           23 cloud           24 cockroach
25 couch            26 crab            27 crocodile       28 cup             29 dinosaur
30 dolphin          31 elephant        32 flatfish        33 forest          34 fox
35 girl             36 hamster         37 house           38 kangaroo        39 keyboard
40 lamp             41 lawn_mower      42 leopard         43 lion            44 lizard
45 lobster          46 man             47 maple_tree      48 motorcycle      49 mountain
50 mouse            51 mushroom        52 oak_tree        53 orange          54 orchid
55 otter            56 palm_tree       57 pear            58 pickup_truck    59 pine_tree
60 plain            61 plate           62 poppy           63 porcupine       64 possum
65 rabbit           66 raccoon         67 ray             68 road            69 rocket
70 rose             71 sea             72 seal            73 shark           74 shrew
75 skunk            76 skyscraper      77 snail           78 snake           79 spider
80 squirrel         81 streetcar       82 sunflower       83 sweet_pepper    84 table
85 tank             86 telephone       87 television      88 tiger           89 tractor
90 train            91 trout           92 tulip           93 turtle          94 wardrobe
95 whale            96 willow_tree     97 wolf            98 woman           99 worm
```

For my project I will not require all the super classes. The most probable super classes to be found in parking lots/roads and relevant for my project are 3, 14, 18 and 19. I created a sub dataset, namely CIFAR-20 with the selected super classes.

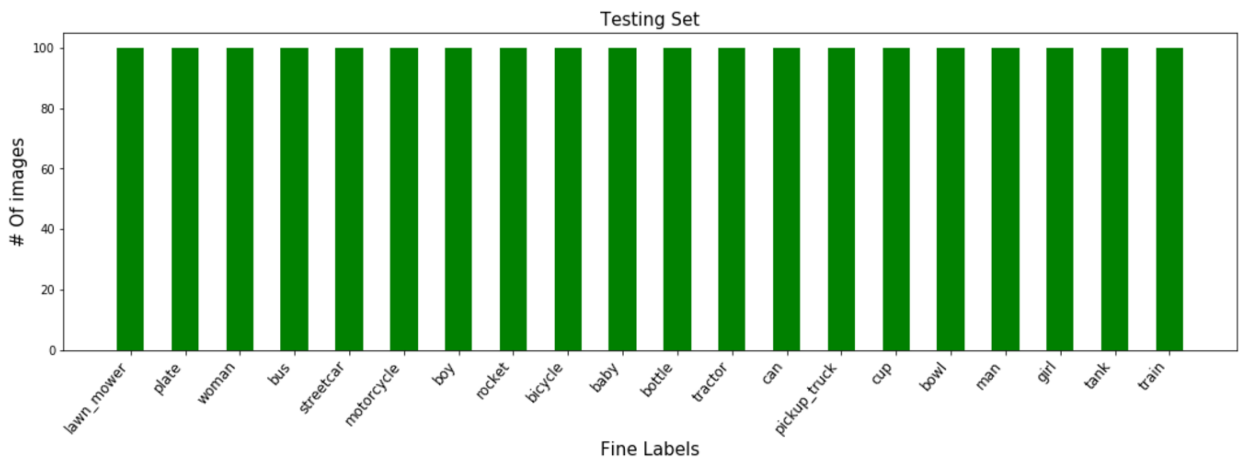
The superclasses and classes in CIFAR-20 are as shown below:

```
**CIFAR-20 Superclasses**
 0 food_containers
 1 people
 2 vehicles_1
 3 vehicles_2
```

```
*********CIFAR-20 Classes*********
 0 baby             1 bicycle
 2 bottle           3 bowl
 4 boy              5 bus
 6 can              7 cup
 8 girl             9 lawn_mower
10 man             11 motorcycle
12 pickup_truck    13 plate
14 rocket          15 streetcar
16 tank            17 tractor
18 train           19 woman
```

3.1. Data Exploration

Below is the distribution of each fine label in the CIFAR-20 dataset



It can be seen that only the fine labels of the selected super classes are present in the CIFAR-20 dataset. There are 500 images of each class in the training set and 100 images of each class in testing set.

3.2. Data Visualization

Below are few images from the CIFAR-20 dataset

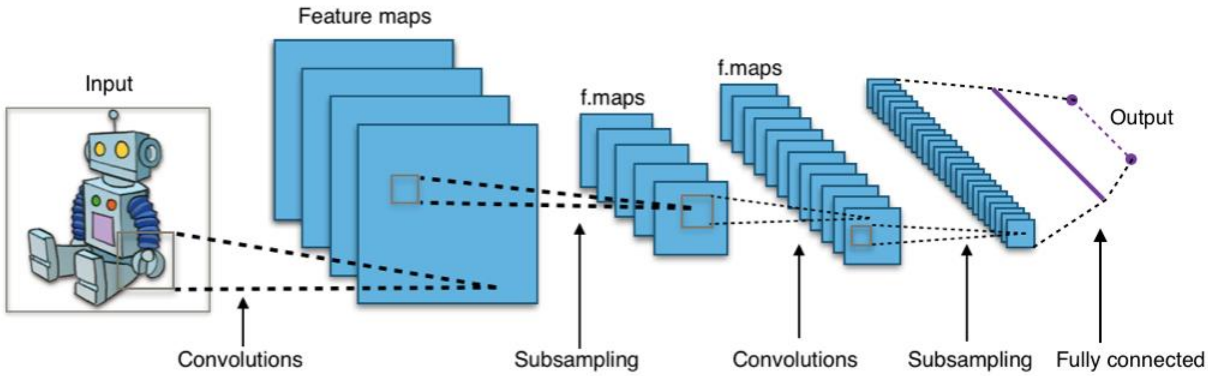
4. Solution Statement

The goal of the project is to design a Convolutional Neural Network(CNN) to classify objects. Convolutional networks were inspired by biological processes in which the connectivity pattern between neurons is inspired by the organization of the animal visual cortex[3]. CNNs require very little pre-processing compared to other image classification algorithms. This implies that the network learns the filters which in traditional algorithms have to be hand-designed.

---

[3] https://en.wikipedia.org/wiki/Convolutional_neural_network

Below is an image of a typical CNN:



A typical CNN consists of an input, an output and multiple hidden layers. The hidden layers are either convolutional, pooling/subsampling or fully connected. Convolutional layers apply a convolution operation to the input, passing the result to the next layer. Pooling layers combine the outputs of neuron clusters at one layer into a single neuron in the next layer. Fully connected layers connect every neuron in one layer to every neuron in another layer.

The project aims at developing a simple CNN model involving mostly of convolutional layers. Simple model implies less memory and less training time and faster execution.

5. Benchmark Model

I will use the All-CNN[4] model, with modification to last layer, as benchmark. All-CNN is convolutional neural network with focus on simple architecture using only convolutions and subsampling. It claims to match or even slightly outperform the state of the art on CIFAR-10 and CIFAR-100.
The network, with modification in the last layer to suite my CIFAR-20 dataset, is as below:

Input 32x32 RGB image
3x3 conv. 96 ReLU
3x3 conv. 96 ReLU
3x3 conv. 96 ReLU with stride r =2
3x3 conv. 192 ReLU
3x3 conv. 192 ReLU
3x3 conv. 192 ReLU with stride r = 2
3x3 conv. 192 ReLU
1x1 conv. 192 ReLU
1x1 conv. 10 ReLU
global averaging over 6 × 6 spatial dimensions
20-way softmax

[4] https://arxiv.org/pdf/1412.6806.pdf

6. Evaluation Metric

As shown in the Dataset section, every image has got a class number, e.g. class "boy" has got class number 4. The class numbers will be one-hot encoded[5] to get a vector of length 20, namely y_true. Given an image as input, the CNN outputs a "prediction" class, namely y_pred, predicting the class the image belongs to. As the dataset is uniformly distributed across all the 20 classes, I will use "categorical_accuracy" as my evaluation metric.

categorical_accuracy = mean(equal(argmax(y_true), argmax(y_pred))

argmax() returns the index of vector element having maximum value
equal() returns a boolean vector
mean() return the mean of a vector

[5] http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html