Vehicle Type Classification Using Convolutional Neural Network

Junhan Ma
College of Engineering,
Northeastern University

Bo Han
College of Engineering,
Northeastern University

Abstract**:** In this paper, a method to solve vehicle type classification using Convolutional Neural Network(CNN) has been proposed. CNN has been proved an outstanding performance on image classification. This task is to recognize the vehicle type in any given image which taken based on different conditions: illumination, scale, surface color of vehicles. In order to get better prediction, we implement baseline CNN model, Mimic AlexNet model and Transfer Learning on this dataset which includes vehicle images taken from the frontal view and pre-annotated type and vehicle location information. Softmax classifier is served as the output layer in each model. For a given vehicle image, each CNN model can present the probability of the type to which the vehicle belongs. Experimental results based on different models and related comparison show the best prediction presented by the Mimic AlexNet method with accuracy over 90%.

Introduction

Nowadays, vehicle types classification attracts more attention and research due to its increasing significance in multiple areas, such as traffic control and surveillance, autonomous navigation, intelligence parking management and communication guidance. Until now, numerous image-based analysis and models have been proposed, and they mainly can be divided into two areas: model-based methods and appearance-based methods. Model-based methods compute the vehicle's 3D parameters such as length, width, and height to recover the 3D model of the vehicle. Appearance-based methods extract appearance features[1]. Surveillance cameras mainly focus on frontal view of vehicles, therefore focusing on such type of vehicle image classification is essential.

In machine learning area, a convolutional neural network (CNN) is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery. Unlike traditional methods by using hand-crafted features, the convolutional neural network is able to automatically learn multiple stages of invariant features for the specific task[2].

In this paper, three CNN models: swallow CNN, Mimic AlexNet and pre-trained Inception V3 have been implemented. These models are either from scratch or from pre-trained. The convolutional neural network takes an original vehicle image as the input and outputs the probability of each vehicle type to which the vehicle belongs. The convolutional layer computes the convolutions between the input and a set of filters, provides a nonlinear representation of the input signal by using a point-wise nonlinear function. The average pooling layer and subsampling layer reduce the spatial resolution of the representation to achieve the robustness in both geometric distortions and small shifts.

The rest of the paper is organized and detailed with related information as follows: In Methods part, we demonstrate this dataset information, the architecture of convolutional neural network and its implementation with three different learning models, and also describe related network parameters. In Results part, experimental results and analysis are addressed. In Discussion section, conclusion about this paper is given, including the analysis in general and next step to be done.

Methods

There have many breakthroughs in image classification area these years. Images could either be trained from scratch or from pre-trained model. In this part, the same dataset has been processed by three different training models, related technical skills and specific implementations are addressed.

Dataset Information

The BIT-Vehicle dataset[3] contains 9,850 vehicle images that taken based on different conditions: illumination, scale, surface color of vehicles. Each image includes one vehicle and has either 1620x1200 or 1920x1080 resolution. All vehicles can be sourced into six categories: Sedan, Bus, Microbus, SUV, Minivan and Truck, where related number of vehicles based on each type are: 5,922, 558, 883, 1,392, 476 and 822 respectively.

Also one .mat file with the pre-annotated information about the vehicle type and the vehicle location in each image is included in this dataset.

Base Model

In the beginning, a simple VGG-like CNN[4] model is implemented with four convolutional layers and two fully connected layers and also add drop out layers to prevent over-fitting. Using this simple model is to get a general view of this dataset and consider it as the benchmark of future work.

Mimic AlexNet

AlexNet[5] model is proposed in ILSVRC2012. This network acquired a top-5 error of 15.3%, more than 10.8 percentage points ahead of the runner up in ImageNet Large Scale Visual Recognition Challenge in 2012. It contains only eight layers: the first five are convolutional layers, and the last three are fully connected layers.

Due to relatively simple implementation in base model, a more sophisticated CNN model should be conducted.

Within this model, in order to better fit this dataset, modifying some parameters of convolutional layers of original AlexNet model, using max-pooling layers to down sampling and dropout layers to prevent over-fitting.

Transfer Learning

Pre-trained model for one task can also be reused at the very first beginning of another task. Besides implementing those deep CNNs above, InceptionV3 model with pre-trained weights on ImageNet dataset without the top layers is introduced. Keras makes it easier to use this pre-trained mode.

InceptionV3 achieves 5.64% top-5 error while an ensemble of four of these models achieves 3.58% top-5 error on the validation set of the ImageNet whole image ILSVRC 2012 classification task[6].

First, training InceptionV3 model on this dataset once, and saving its bottleneck features from the layer before the fully-connected layers. Then we train a small fully-connected model based on these features.

Because running a whole InceptionV3 model is expensive, so saving the bottleneck features rather than adding our fully-connected model directly on top of a frozen convolutional base and running the whole thing[7].

Results

In general, the base model has been trained using local computer CPU due to its relatively simple and small data. Concerning the advanced model and transfer learning, Amazon Web Services EC2 instance is the suitable place for training sophisticated model using GPU implementation. And all work has been implemented based on Keras API.

Data Pre-processing

As mentioned before, all images in this dataset are taken based on frontal view and a Matlab file containing the information of vehicle's location in the image. We use OpenCV to reshape images. Therefore, cutting out the vehicle image from original one and convert them to the same size are needed. Figure 1 shows the results after cropping the vehicles.

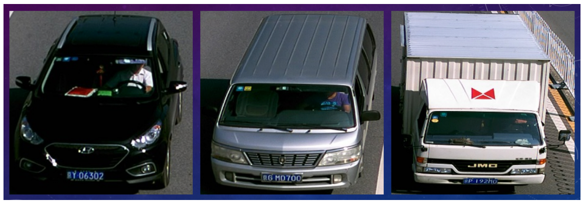

Figure 1. Different vehicle types after cropping.
Left: SUV Middle: Minivan Right: Truck

Baseline Model

This model with images resized to 32x32 is trained on local computer CPU, with the accuracy: 88.21% and the test accuracy: 88.63% after 60 epochs. This is a good result as a baseline model but needs further improvement.

Mimic AlexNet Model

In order to adapt this dataset, the first convolutional layer filter size is changed to 32 and kernel size is adjusted to 3x3. First we use images with 32x32 resolution on this model and the validation accuracy is 88.43%. Then, trying to change image resolution to 64x64. After training 21 epochs, this model reaches to plateau with the accuracy: 98% and the validation accuracy: 91%, which is better than VGG-like baseline model.

Transfer Learning

We resize images to 299x299, and use data augmentation to rescale data to 0-1, and extract bottleneck features using image data generators from pre-trained InceptionV3 which is a quite complex CNN model to train a small fully connected model. This model trains fast on AWS GPU instance because of the small size of features data. The accuracy is 58% and validation accuracy is 60%.

All the results based on three models are displayed in Table 1.

	Image Size	***Accuracy***	***Validation Accuracy***
Baseline Model	32x32	88.21%	88.63%
Mimic AlexNet	32x32	99%	88.43%
	64x64	98%	91%
Inception V3	299x299	58%	60%

Table 1. Models Evaluation

Discussions

Based on the experimental statistics: The mimic AlexNet model gains the best performance with validation accuracy higher than 0.9.

Referring to the low performance done by Inception V3 model with pre-trained weights on ImageNet, it could be the problem of matching, this dataset does not share too much common with ImageNet and also we doubt the performance when such a sophisticated CNN model performed on this small dataset.

Therefore, next step is to use more aggressive data augmentation, not only rescale, but also rotation, tuning the brightness, etc. and modify dropout layer parameters to train this model again.

All the code within this design can refer to this GitHub repository[8].

Reference

[1] Zhen Dong, Yuwei Wu, Mingtao Pei, and Yunde, "Vehicle type classification Using a Semisupervised Convolutional Neural Network from visual-based dimension estimation," in IEEE Transactions On Intelligent Transportation Systems, Vol. 16, NO.4, August 2015, pp. 2247–2256.

[2] Baidaa Al-Bander, Waleed Al-Nuaimy, Bryan M. Williams, Yalin Zheng, Multiscale sequential convolutional neural networks for simultaneous detection of fovea and optic disc, Biomedical Signal Processing and Control, Volume 40, 2018, Pages 91-101.

[3] Data Source: http://iitlab.bit.edu.cn/mcislab/vehicledb/

[4] Simonyan Karen, Zisserman Andrew, "Very Deep Convolutional Networks for Large-Scale Image Recognition", eprint arXiv:1409.1556, 09/2014.

[5] https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.

[6] Szegedy Christian, Vanhoucke, Vincent, Ioffe Sergey, Shlens Jonathon, Wojna Zbigniew, Rethinking the Inception Architecture for Computer Vision, eprint arXiv:1512.00567, 12/2015.

[7] https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

[8] Project Code with Documentation: https://github.com/stoneloe/Vehicle-Type-Classification-Using-CNN