Applying Pretrained Convolutional Neural Networks to Vehicle Classification Problem

Viktoria Plemakova
Institute of Computer Science
University of Tartu
viktoria.plemakova@ut.ee

Abstract—Artificial neural networks (NN) have gained a lot of popularity in the recent years. Although basic NNs can be applied to computer vision related tasks, convolutional neural networks (CNN) have shown better suitability and results. This report gives a general overview of NNs and CNNs as well as specific CNN models. Besides, the pretrained CNN models are tested on vehicle images as classifiers.

I. INTRODUCTION

Convolutional neural networks have been used in several computer vision related tasks such as image classification, object detection and localization but also in natural language processing. The popularity of CNNs is especially noticeable among the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [1] submissions. Vehicle classification and detection are some of the tasks in intelligent transportation systems. Since CNNs are designed to extract and learn specific features from images then this method can be applied to vehicle images as well.

Section II gives an overview of artificial neural networks that is followed by an explanation on convolutional neural networks in section III. Section IV describes some of the known convolutional neural networks that also have pretrained models available. In section V the pretrained models are tested as classifiers on vehicle images. Conclusions are made in section VI.

II. NEURAL NETWORKS

A neural network (NN) is a machine learning technique that was inspired by the human brain. The NNs are composed of an input and output layer and one or many hidden layers in the middle. The layers, in turn, consist of neurons (units) that perform computations. Mostly the layers are fully-connected meaning that neurons are connected to all neurons from the previous layer (Figure 1).

The inputs are passed to the hidden layers since no computations are done in the input layer. A single hidden neuron takes $n$ inputs $x_1, \ldots, x_n$ from the previous layer and assigned weights $w_1, \ldots, w_n$ for each connection. The equation 1 shows how the output of the $i$th neuron in the $l$th hidden layer is computed [2].

$$o_l^i = f(z_l^i) \text{ where } z_l^i = \sum_{j=1}^{n_l} w_{lj}^i x_j^{l-1}$$  (1)

![Neural network architecture](image)

The function $f$ in the equation 1 is an activation function that is applied to each neuron’s output between the hidden layers. Activation functions introduce non-linearity to the network. Different activation functions can be used depending on the suitability for a certain N. Some of the typical activation functions are [3]:

a) Sigmoid function produces a value between 0 and 1.

$$h(x) = \frac{1}{1 + e^{-x}}$$  (2)

b) Hyperbolic tangent outputs a value between -1 and 1.

$$h(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$  (3)

c) Rectified Linear Unit (ReLU) takes a value and outputs 0 if it is negative or the value itself if it is positive. ReLU is used predominantly with convolutional neural networks.

$$h(x) = \max(0, x)$$  (4)

III. CONVOLUTIONAL NEURAL NETWORKS

A CNN is a deep neural network that can be applied to images. CNN architectures can be different, but they all consists of convolutional layers, pooling layers and fully-connected layers as illustrated in Figure 2. This section describes the specifics of these layer types.

A. Convolutional layer

Convolution is performed on input images to extract features, e.g object edges. Every colour image is just a $n \times m \times 3$ array of pixel values from 0 to 255. Since images contain
three colour channels the convolution is done separately for each channel [3].

For that a kernel of size $n \times n$ is applied to each image resulting in a feature map. Usually more than one kernel is applied which results in multiple feature maps (Figure 3). The kernel is moved across the image with stride $s$ and at each region (local receptive field [2]) the kernel values are multiplied with the local receptive field values and then the products are summed up. The result corresponds to a single pixel value in the feature map as illustrated in Figure 3.

The CNNs use shared weights. Instead of trying to learn different features in every image location it is more useful to detect a single feature in the whole image [3], [5] using the same kernel.

B. Pooling layer

Pooling is used after the convolution to downsample each feature map. It helps to retain only the most valuable information from the images [6] as well as to avoid overfitting since the number of parameters decreases [3]. The pooling type depends on the operation performed. The most common pooling technique is max pooling but sum or average pooling can be used as well [6].

The operation is similar to the convolution: a window of size $n \times n$ is moved over the image with stride $s$. Very often the size of the window is $2 \times 2$ and the stride size 2. If the selected pooling type is max pooling, then the largest value of each window region becomes the output as illustrated in Figure 4.

C. Fully-connected layer

The output of the final convolutional or pooling layer is flattened into a one-dimensional vector and fed to one or more fully-connected layers. For a classification task, the final layer is comprised of $n$ neurons where $n$ is the number of classes the dataset contains.

IV. CONVOLUTIONAL NEURAL NETWORKS FOR IMAGE CLASSIFICATION

This section describes three convolutional neural network models that were trained on the ImageNet data used for the ILSVRC [1]. In section V the pre-trained versions of these networks are applied to a classification problem.

A. VGG16 and VGG19

The VGG neural network [7] architecture was introduced in 2014 as a result of an investigation into how the depth of the CNN affects its accuracy. For the evaluation the authors proposed six network configurations with different number of weight layers. The best models were the ones that had a depth of 16 and 19 weight layers.

The network used kernels of size $3 \times 3$ in every convolutional layer with ReLU activation. Several convolutional layers were stacked on top of each other before performing $2 \times 2$ max pooling. Having multiple sequential $3 \times 3$ convolutions works just as well as having a single larger convolution. In addition, it results in fewer parameters.

B. ResNet

The VGG neural networks [7] showed that increased network depth can lead to better results. Adding more and more layers might, however, end up with decreasing training accuracy of the network that is not caused by overfitting. The
authors of ResNet [8] proposed a method for training neural networks that are even deeper (e.g. 152 layers) than the VGG networks: deep residual networks.

C. OverFeat

The OverFeat network [9] was designed to not only solve classification problems but also to perform localization and detection of objects in images. The localization task guesses a single object in the image and additionally returns its bounding box, whereas the detection task has to locate all possible objects. Additionally, the authors released a feature extractor that has two available models: fast and accurate.

For the classification task the convolutional and pooling layers of the network are similar to the AlexNet [10]. The authors brought in the following changes: reduced the pooling stride of the 1st and 2nd layer, used non-overlapping pooling and omitted contrast normalization. The final three layers are fully-connected. For the localization task the fully-connected layers of the classification model are replaced by a regression network which consists of two fully-connected layers and outputs the coordinates for the bounding box.

V. EXPERIMENTS

This section describes the performed tests on pretrained CNNs and their results.

A. Methods

The CNN training requires a lot of data and therefore can take a long time and resources to be trained to perfection. Fortunately, there are pretrained models available that can serve as a starting point for transfer learning and fine-tuning or simply used as classifiers and feature extractors [3]. The pretrained versions of ResNet50 [8] and VGG networks [7] are available in Keras, a Python library for neural networks. The OverFeat [9] network is available in a standalone repository where the code can be either built from source or already precompiled binaries can be used instead.

The chosen CNNs were trained using the ImageNet dataset, thus the classification outputs should ideally have similar results. To compare the results the four networks were tested by feeding them images of different vehicle types and taking the top three guesses. The images were selected from [11] and included a minivan, a jeep, a pickup, a sports car and a sedan.

B. Results

The results are available in Table I. From the results, it is evident that the sports car and pickup images were the easiest to classify since all of the models got them right. Interestingly, the VGG networks and ResNet50 had exactly the same guesses for the sports car. The jeep image was classified correctly by three of the networks. VGG16 classified it as a minivan which was also among the secondary guesses from the other models. The minivan picture was confusing: only ResNet50 labeled the image correctly while the other models thought it was a minibus instead.

Overall, judging from the results, all of the networks are capable of distinguishing various types of cars. Among the guesses there are also classes such as a limousine, a moving van and even a separate class for a police van. Therefore, it can be concluded that the networks were able to learn a vast set of vehicle features. Another thing to try would be tuning one of the pretrained networks to work as a vehicle classifier only.

VI. CONCLUSION

This work introduced artificial neural networks and specifically convolutional neural networks since they are widely used for image classification and other computer vision problems. In addition, three network models were introduced: VGG networks, ResNet and OverFeat. All of the models have pretrained versions easily accessible. The pretrained networks were tested on images of different vehicle types. The results of the classification were mostly accurate and using pretrained networks can be a good base for developing a vehicle classifier.

REFERENCES

<table>
<thead>
<tr>
<th>type</th>
<th>VGG16</th>
<th>VGG19</th>
<th>ResNet50</th>
<th>OverFeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>jeep</td>
<td>minivan, 0.31316641, jeep, 0.22675458,</td>
<td>jeep, 0.39384496, minivan, 0.24313863,</td>
<td>jeep, 0.97427434, pickup, 0.014005006,</td>
<td>jeep, landrover 0.71491,</td>
</tr>
<tr>
<td></td>
<td>beach_wagon, 0.084146477</td>
<td>minibus, 0.1024888</td>
<td>minivan, 0.0065984836</td>
<td>minivan 0.0664544,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>minibus 0.0204166</td>
</tr>
<tr>
<td>pickup</td>
<td>pickup, 0.76730949, tow_truck, 0.064050697,</td>
<td>pickup, 0.85349578, tow_truck, 0.02732683,</td>
<td>pickup, 0.98227906, grille, 0.0053238943,</td>
<td>pickup 0.468685,</td>
</tr>
<tr>
<td></td>
<td>limousine, 0.045928322</td>
<td>minivan, 0.01445452</td>
<td>tow_truck, 0.0046486654</td>
<td>limousine, limo 0.244896,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>tow truck 0.157212</td>
</tr>
<tr>
<td>minivan</td>
<td>minibus, 0.56778377, minivan, 0.39037687,</td>
<td>minibus, 0.76680142, minivan, 0.19920361,</td>
<td>minivan, 0.59616947, minibus, 0.38474712,</td>
<td>minibus 0.877915,</td>
</tr>
<tr>
<td></td>
<td>jeep, 0.011574605</td>
<td>moving_van, 0.016416755</td>
<td>pickup, 0.0071328329</td>
<td>minivan 0.0844457,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>police van 0.0088912</td>
</tr>
<tr>
<td>sports car</td>
<td>sports_car, 0.641114046, racer, 0.2517136,</td>
<td>sports_car, 0.5597716, racer, 0.37773848,</td>
<td>sports_car, 0.74752909, racer, 0.23821115,</td>
<td>sports car 0.904889,</td>
</tr>
<tr>
<td></td>
<td>car_wheel, 0.0544127</td>
<td>car_wheel, 0.04475775</td>
<td>car_wheel, 0.0072815763</td>
<td>racer 0.0822441,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>convertible 0.0104115</td>
</tr>
<tr>
<td>sedan</td>
<td>minivan’, 0.30450383, beach_wagon’, 0.244647,</td>
<td>limousine’, 0.27984712, convertible’, 0.16525631,</td>
<td>limousine’, 0.51297551, minivan, 0.094495483,</td>
<td>convertible 0.23289,</td>
</tr>
<tr>
<td></td>
<td>limousine’, 0.18945754</td>
<td>beach_wagon’, 0.12707476</td>
<td>beach_wagon’, 0.048630554</td>
<td>beach_wagon’, 0.172313,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>sports car 0.157339</td>
</tr>
</tbody>
</table>

TABLE I: Top 3 classification results of five types of vehicles.