Kerem Buyukdereli 64666

Onat Yapici 63898

Dr. Banu Yobaş

ENGR 350

20/08/2019

**Identifying Volkswagen Passat and Fiat Doblo using Convolutional Neural Network**

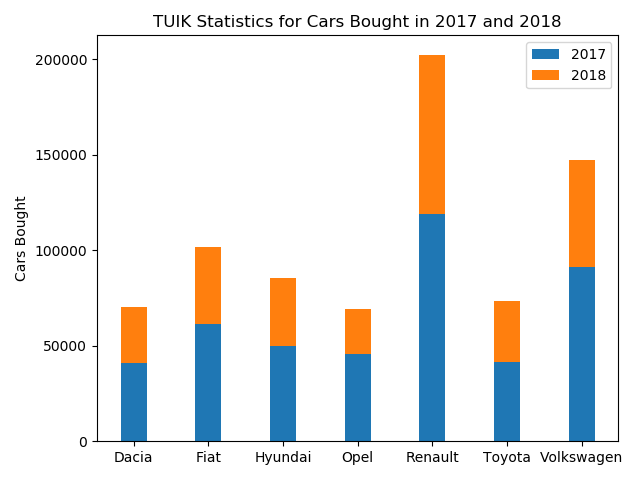
As PickCar group we first decided to focus on a suggestion engine for cars. It was planned to take specifications of the car you would like to have, and specifications of the car the user currently owns. However, considering the current advancements and saturation in the suggestion engines we decided that a neural network that would identify a car from its photos would be a more useful tool at the moment in Turkey. We are planning to put together a neural network that could make a binary choice, whether it is a Volkswagen Passat or a Fiat Doblo, given the car photos. This neural network, when trained for more car brands and models, can be used in security cameras placed in the roads and streets to collect data about how owners of each car model react to different traffic situations, whether they prefer to take a longer road with less traffic or if they would choose to go directly with more traffic, and help manufacturers tweak their cars according to driver preferences.

Currently car identification in Turkey is based on license plates in front and back of each car. This method has some shortcomings considering how small the license plates are compared to cars. Some of the problems bounded to using only license plates for identification can be listed as:

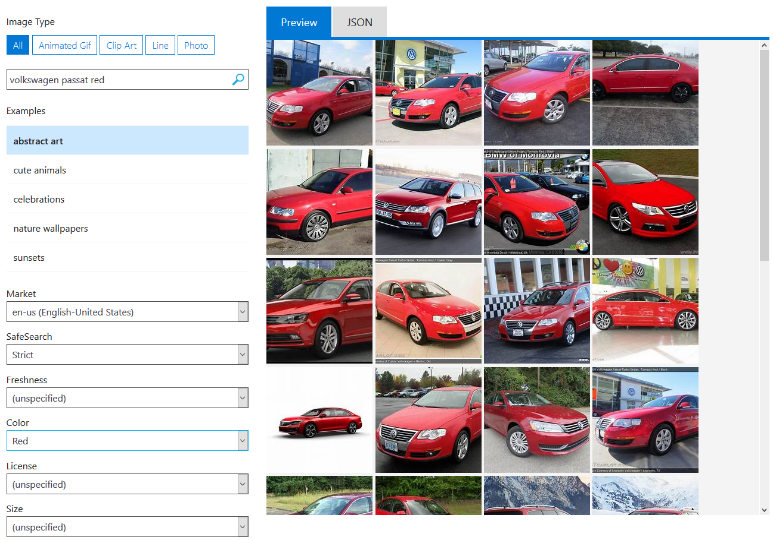
* not always in the same height or angle
* some drivers put them inside their cars
* can be easily obscured with snow and other external effects.
* difficult to capture while the vehicle is turning

These inadequacies cause wrong license plate numbers to be identified and sometimes lead to wrongly issued (Larson, n.d.)traffic tickets. This problem can be averted by utilizing a cross check with car model and brand using the neural network we trained.

**Picking Which Car Brands and Models to Train Neural Network**

Volkswagen and Renault are the two leading brands that automobile drivers in Turkey prefer for last two years according to TUIK. However, another option that is dominant is from panel van class is Fiat Doblo. In order to work on two different classes of cars we picked Volkswagen Passat and Fiat Doblo for a binary approach. The photos of cars that Stanford supplies in Stanford Car Dataset are similar to what we need to train our network however this dataset doesn’t include the photos of most bought cars in Turkey.

**Creating a Dataset for Most Popular Cars in Turkey**

In order to create a dataset with enough photos of the cars we need we searched dataset sites however decided to use Bing Image Search API as it contains thousands of photos submitted by users. Although this helps building an immense dataset of pictures there are also images that are “related” to our query, but unrelated to our dataset, such as headlights and toys of the car brands we are looking for. In order to filter unrelated photos, we added a color term to query as well as a color filter to pick red, white and black cars. As a last check we have gone through photos as much as we can to make sure nothing slips by. We accomplished to compile a dataset of approximately 20000 photos for the most bought car brands in Turkey.

**Extractable Features to Focus On**

A car parked in a parking lot

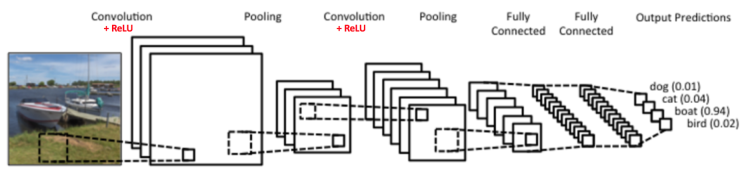
Description automatically generatedAs a proof of concept, we chose Volkswagen Passat and Fiat Doblo, cars from two different classes, with distinct features.

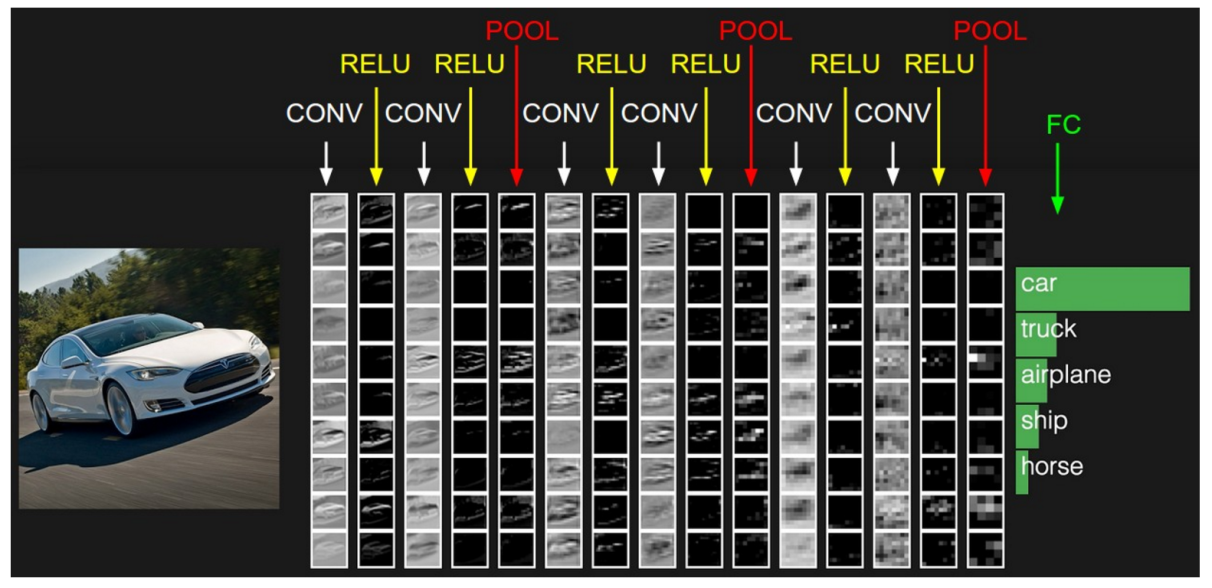
As we are going to use a Convolutional Neural Network we cannot pick specifically what the neural network should focus on but instead we should decide on the distinct features and figure out how we can enhance these features using image processing. The hyperparameters of layers of neural network should be arranged accordingly to achieve most accurate weights. To achieve these end results, we decided to first make the neural network look at the image in parts as big as possible. This is related to the fact that two cars have different sizes and shapes. In a deeper layer it should be able to extract a fairly smaller part, compared to layer before, that still is big considering the size of the image. This layer is supposed to differentiate between smaller details such as headlights, rims and rearview mirrors.

**Preparing the Photos to Process**

As the images are already categorized and picked with colors the only process was resizing them to 128x128. This was done to ensure all the input images are one type so that the neural network could apply same weights to each input. We used suggested parameters for data similar to ours for values of shear\_range and zoom\_range. Dataset was read through a directory using flow\_from\_directory with parameter class\_mode= ‘binary’ as we are planning to classify two different cars.

**Deciding on Neural Network and Hyperparameters**

Convolutional Neural Networks are the most used type of neural networks in image processing. 



As a brief introduction they are composed of several steps such as:

* convolution
* activation
* pooling

In convolution step the image is “convolved”, a matrix with (elements specific to convolution type) a size of user choice goes through each pixel to extract different features of image. As matrix goes through the image, the information of n-by-n pixels get condensed into a single pixel such as an edge. After convolution different types of filters can be applied where in different neural network different filters are applied. These filters are named neurons and the functions they hold are called activation functions which determines whether they will fire or not. Pooling is used for several reasons such as to make feature dimensions smaller and more manageable, avoid overfitting by averaging and getting the maximum of several elements, making the network invariant to small transformations, distortions and translations. By layering these steps a convolutional neural network is built. Training is done using backpropagation which is similar to arranging and modifying weights until the most accurate results are obtained.

The hyperparameters of our CNN is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2D Convolution | 2D Convolution | 2D Convolution | Flatten  Dense (64)  ReLU  Dropout (0.5) | Dense (1)  Sigmoid |
| 32 (Big portions of image) | 64 (Mid-sized portions) | 32 (Small features) |
| Rectified Linear Unit | Rectified Linear Unit | Rectified Linear Unit |
| Max pooling | Max pooling | Max pooling |

In order to program this CNN we used numpy, pandas, matplotlib, seaborn, sklearn, tensorflow, keras, cv2, IPython, PIL.

**Challenges**

Considering the availability and ease of use of Stanford Car Dataset it was tempting to use it compared to compiling our own dataset. After deciding to make our own dataset, although the photos submitted by users helped us gather a large dataset, it was hard to filter images of cars that would yield a good training set. Another challenge was to decide on hyperparameters of the neural network that would yield the best results. After going through websites and articles we achieved a high accuracy result. After getting good results we saw that there were some problems caused by overfitting, this is where we utilized dropout command to make sure the neural network used different neurons instead of heavily relying on several of them.

**Potential Improvement**

If we had several more weeks, we could have trained the neural network to identify different brands and models of cars until eventually finishing most of the cars used in Turkey. A later step would be to use a Raspberry-Pi and camera to identify different cars in real time.

**Resources**

1. “Be Awesome at OpenCV, Python, Deep Learning, and Computer Vision.” *PyImageSearch*, [www.pyimagesearch.com/](http://www.pyimagesearch.com/).
2. Britz, Denny. “Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs.” *WildML*, 8 July 2016, [www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/](http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/).
3. *Index of /computer\_vision*, [www.machinelearninguru.com/computer\_vision](http://www.machinelearninguru.com/computer_vision).
4. Geva. “Understanding a 3D CNN and Its Uses.” *MissingLink.ai*, missinglink.ai/guides/neural-network-concepts.
5. “Convolutional Neural Networks .” *CS231n Convolutional Neural Networks for Visual Recognition*, cs231n.github.io/convolutional-networks/.
6. Ujjwalkarn. “An Intuitive Explanation of Convolutional Neural Networks.” *The Data Science Blog*, 29 May 2017, ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/.
7. *Should You Fight Your Traffic Ticket? | ExpertLaw*. [www.expertlaw.com/library/traffic\_tickets/fighting\_tickets.html](http://www.expertlaw.com/library/traffic_tickets/fighting_tickets.html).
8. “IBM Knowledge Center.” *IBM Knowledge Center*, [www.ibm.com/support/knowledgecenter/en/SS3RA7\_15.0.0/com.ibm.spss.modeler.help/neuralnet\_modeltab.htm](http://www.ibm.com/support/knowledgecenter/en/SS3RA7_15.0.0/com.ibm.spss.modeler.help/neuralnet_modeltab.htm).
9. Keras Documentation
10. [TUIK](https://biruni.tuik.gov.tr/medas/?kn=89&locale=tr) Motorlu Taşıt Sayısı