Vehicle Make, Model and Color Recognition based on Deep Learning

Jiupeng Sun
Stanford University

CS230

Introduction to MMCR

Vehicle make, model and color recognition can be widely leveraged by traffic control department for statistic and suspicious vehicles tracking. Also it can benefit common consumers as well. Imagine users can know the detailed information about a car they saw in the street by just taking a photo. In this paper I will finalize an end-to-end application, the input would be a RGB picture which displays a car in arbitrary angle, and output would be an explicit string that describes the make, model and color of that car.

Dataset

Stanford vehicle dataset [2] is used in this project as it is free and easy to access. This dataset contains 16,185 images from 196 classes. And the data is split into 8,144 training images and 8,041 test images. Each car is described with Make, Model and Year, which means the 196 classes may contain cars of same make or model but different years.

Fig. 1: Sample images from Stanford Vehicle dataset

Loss Function

As a classification problem, cross entropy is enough to be the loss function. Firstly, the softmax function produces:

$$softmax(s_i) = \frac{e_i^s}{\sum_{j=1}^{N} e_j^s}, i = 1, ..., N \tag{1}$$

where s_i is the similarity of the class i, N is the number of the classes.
The cross entropy measures the quality of a model for a probability distribution.

$$CE = -\frac{1}{M}\sum_{k=1}^{M}\sum_{i=1}^{N} N y_{ik} \log(softmax(s_{ik})) \tag{2}$$

where k is the index of the element in the batch, M is the batch size.

References

[1] François Chollet. *keras*. `https://github.com/fchollet/keras`. 2015.

[2] Jonathan Krause et al. "3D Object Representations for Fine-Grained Categorization". In: *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*. Sydney, Australia, 2013.

[3] Reza Fuad Rachmadi and I Purnama. "Vehicle color recognition using convolutional neural network". In: *arXiv preprint arXiv:1510.07391* (2015).

Architecture

The overall architecture of the application is shown as Figure 2, a YOLO network will be leveraged to output the coordinate of the box which contains the main object in the image. Then I resized the output to (224, 224, 3) before feeding it to the neural network. [3] has already gave an efficient and highly accurate CNN (97% accuracy) to classify the color of vehicle. Benefit from transfer learning, I chose several mature networks from the Application model of Keras[1], which can help mitigate the time constraint on my work. The networks I used includes VGG, SqeezeNet, ResNet50, InceptionResNetV2 and MobileNet. To leverage transfer learning, I used ImageNet weight and **freeze** the whole network as base model, and add two blocks at the end of base model, each block contains one dense layer, one batch normalization layer and one dropout layer, so I only need to train the parameters in last two blocks for each network 3. In addition I also implemented one logistic regression model and two simple CNNs as the baseline models.

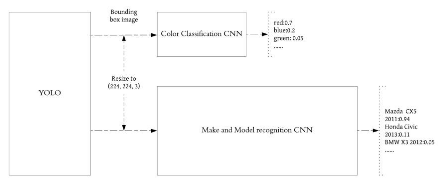

Fig. 2: Overall Architecture of System

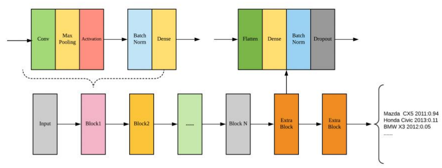

Fig. 3: Models Architecture

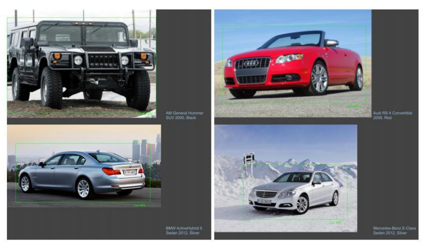

Fig. 4: Prediction Results

Experiments

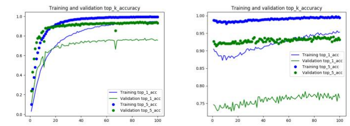

Fig. 5: Accuracy of VGG network

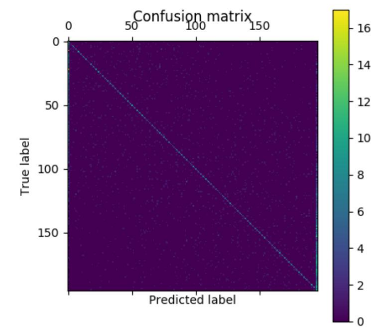

Fig. 6: Confusion matrix

Conclusion

Baseline models are either not sophisticated to learn the models, or cannot handle overfitting well. The complex models perform better but need the assistant of transfer learning, otherwise it may take too long to tune the hyper parameters and train the network. The final prediction accuracy is 79.53%, some images are wrongly labeled, either because of the low quality of original images or two cars with same model & make but different manufacture years have extremely similar appearance, which is even hard to be distinguished by human beings. In the future I may collect more data to mitigate overfitting and test more networks such as GoogleNet, then insert new layers in the middle and tune the parameters to obtain a better performance.

Acknowledgements

I do appreciate CS230 course staffs especially my mentor **Tugce Tasci** for their dedicated work to make this class successful.