A Baseline for the Commands For Autonomous Vehicles Challenge.

Simon Vandenhende
KU Leuven - ESAT/PSI
`simon.vandenhende@kuleuven.be`

Thierry Deruyttere
KU Leuven - LIIR
`thierry.deruyttere@kuleuven.be`

Dusan Grujicic
KU Leuven - ESAT/PSI
`dusan.grujicic@kuleuven.be`

Abstract

The Commands For Autonomous Vehicles (C4AV) challenge requires participants to solve an object referral task in a real-world setting. More specifically, we consider a scenario where a passenger can pass free-form natural language commands to a self-driving car. This problem is particularly challenging, as the language is much less constrained compared to existing benchmarks, and object references are often implicit. The challenge is based on the recent `Talk2Car` dataset. This document provides a technical overview of a model that we released to help participants get started in the competition. The code can be found at `https://github.com/talk2car/Talk2Car`.

1. Introduction

The *Commands For Autonomous Vehicles* (C4AV) challenge is build on top of the *Talk2Car* dataset [2]. Given a command C and an image I, we are required to predict the coordinates of a bounding box drawn around the object the command is referring to. Some examples can be found in Figure 1. The model is evaluated based on its AP50 score. More details about the benchmark and the dataset are provided in the original works [2, 1].

The remainder of this text discusses the implementation of a baseline model to help participants get started in the competition. The model was implemented in PyTorch [4].

2. Model

2.1. Overview

Figure 2 shows an overview of the method. The procedure is similar to the work from Karpathy et al. [3]. First, an object detector is used to extract region proposals from the input image. Second, we match the region proposals with the command. In particular, we compute the feature cor-

relations between an encoding of the region proposals and the command. The feature correlations are considered as a score for how well the bounding box fits the command. The model is trained by mapping the feature correlations to zero or one, depending on whether the proposed region has an IoU overlap of at least 0.5 with the ground-truth bounding box.

2.2. Implementation Details

2.2.1 Region Proposals

We obtain 64 region proposals for every image by fine-tuning the CenterNet [5] model on the object detection task. Notice that we only use images from the Talk2Car dataset. The annotations are provided by nuScenes [1]. We deliver the region proposals for every image as part of the code package. Duplicate regions are removed through non-maximum suppression.

2.2.2 Method

For every image, we select the 16 regions with the highest confidence score. The region proposals are divided into a set of positive ($IoU \geq 0.5$) and negative boxes ($IoU < 0.5$). A ResNet-18 model that was pretrained on ImageNet is used to encode the regions. The command is encoded using a bi-directional GRU. The feature correlations are measured after L2 normalization and linearly remapped to lie between 0 and 1. The loss for a sample, which is an image with a command and bounding boxes, is calculated with a binary cross-entropy as follows

$$ -\frac{1}{|\mathcal{P}|}\sum_{x\in\mathcal{P}} x\log x - \frac{1}{|\mathcal{N}|}\sum_{x\in\mathcal{N}}(1-x)\log(1-x) \quad (1) $$

where $\mathcal{P}$ contains the feature correlations for positive boxes, and $\mathcal{N}$ contains the feature correlations for negative boxes. As most region proposals do not overlap with the ground

(a) You can park up ahead behind **the silver car**, next to that lamppost with the orange sign on it.

(b) Turn around and park in front of **that vehicle in the shade**.

(c) There's **my mum**, on the left! The one walking closest to us. Park near **her**, she might want a lift.

Figure 1: Examples from the `Talk2Car` dataset. Each command describes a change of direction for the vehicle, relevant to a referred object found in the scene (here indicated by the red 3D-bounding box). The C4AV challenge requires to predict the correct bounding box from the image and the command.
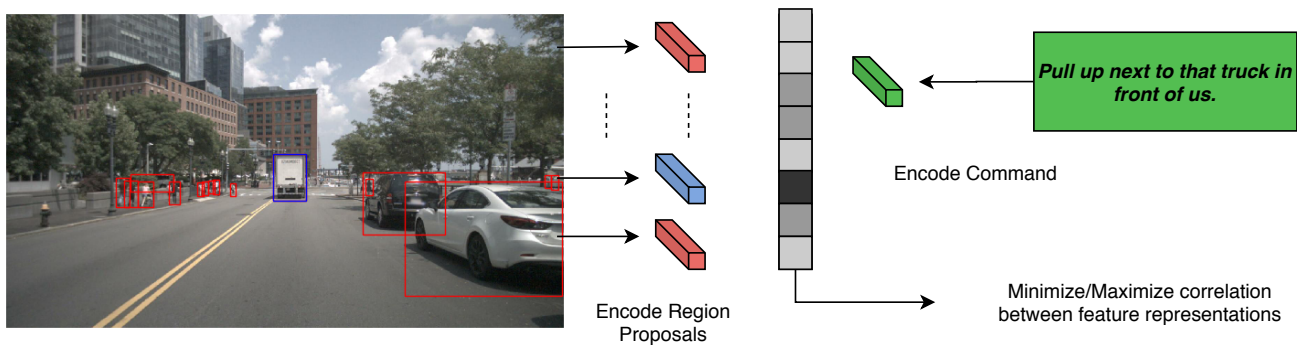

Figure 2: Overview of the method. First, we extract object region proposals. Second, every region proposal is encoded using an off-the-shelf image encoder. In parallel, we encode the command using a recurrent neural network. Finally, we measure the correlation between the feature representations of the region proposals and the command. The training minimizes or maximizes the feature correlations between the regions and the commands depending on whether the object of interest is included in the bounding box ($IoU > 0.5$) or not.

truth bounding box, the number of negative boxes greatly exceeds the number of positive boxes. Therefore, we average the loss of the positive and negative boxes separately, in order to avoid the network from getting biased towards negative predictions.

We train the model for 10 epochs using SGD with nesterov momentum 0.9. The initial learning is set to 0.01 and decayed by a factor 10 every 4 epochs. We use batches of size 18 and a weight decay term 1e-4. The model can be trained in two hours on a single Nvidia 1080ti GPu. Our best model obtains 43.5% AP50 on the validation set.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 1

[2] Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie Francine Moens. Talk2car: Taking control of your self-driving car. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2088–2098, 2019. 1

[3] Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in neural information processing systems*, pages 1889–1897, 2014. 1

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 1

[5] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 1