Have I Reached the Intersection: A Deep Learning-Based Approach for Intersection Detection from Monocular Cameras

by

Dhaivat Bhatt, danish.sodhi , Arghya Pal, Vineeth Balasubramanian, Madhava Krishna

in

International Conference on Intelligent Robots and Systems
(*IROS-2017*)

Vancouver, BC, Canada

Report No: IIIT/TR/2017/-1

Centre for Robotics
International Institute of Information Technology
Hyderabad - 500 032, INDIA
September 2017

Have I Reached the Intersection: A Deep Learning-Based Approach for Intersection Detection from Monocular Cameras

Dhaivat Bhatt*[1] Danish Sodhi*[1] Arghya Pal[2] Vineeth Balasubramanian[2] Madhava Krishna[1]

***Abstract*— Long-short term memory networks(LSTM) models have shown considerable performance on variety of problems dealing with sequential data. In this paper, we propose a variant of Long-Term Recurrent Convolutional Network(LRCN) to detect road intersection. We call this network as IntersectNet. We pose road intersection detection as binary classification task over sequence of frames. The model combines deep hierarchical visual feature extractor with recurrent sequence model. The model is end to end trainable with capability of capturing the temporal dynamics of the system. We exploit this capability to identify road intersection in a sequence of temporally consistent images. The model has been rigorously trained and tested on various different datasets. We think that our findings could be useful to model behavior of autonomous agent in the real-world.**

I. INTRODUCTION

The last few years have confirmed the long-touted claim that autonomous navigation systems will be an integral part of the future of our daily transportation. As efforts on autonomous navigation proliferate across the world, an important problem in this domain is the reliable detection of road intersections. For a robust and complete outdoor exploration, it is essential to detect road intersections as one of the key landmarks. This can help in route planning of the autonomous agent as well as in localization. While it's relatively simpler to localize robots in static indoor environments, where measurements can give us a good sense of an agent's location, this problem can be harder in the outdoor environment.

Another motivation to pursue this problem is towards safe navigation of a driverless vehicle. Traffic junctions are recognized as one of the leading causes for road accidents. According to the report[1], 50% of collisions occur at road junctions. Traffic intersections have a high degree of unpredictable behavior in the dynamic world. When modeling the behavior of the autonomous agent, it is essential to enable the agent to behave more cautiously near intersections, which motivates the problem in this work too.

A key enabler of safety in manually driven vehicles is the ability to detect the onset of an intersection in the near future. However, providing such a capability to an autonomous agent requires modeling road dynamics such as broadening or forking. This necessitates an approach that can position this problem as one of spatio-temporal understanding, where the spatial configuration of the current scene, along with the temporal changes in the scene over a past window, are considered together to label a scene as an intersection. While there has been earlier work (discussed in Section II), all the prior efforts use Lidar or other sensors to achieve this objective. In this work, we propose to detect intersections using only monocular video streams (which has not been done before).

Following the success of deep learning in recent years, we adopt deep learning for the intersection detection problem. We propose the use of a model that brings together spatial understanding (using Convolutional Neural Networks, CNN) and temporal modeling (using Recurrent Neural Networks, RNN) in this work. In particular, we use a variant of the Long-Term Recurrent Convolutional Networks (LRCN) [2] to tackle the problem of detection of road intersections, and pose this problem as a recognition problem with two classes: intersection and non-intersection videos (we also demonstrate how this model can be extended to further classify intersections as four-road crossing or T-junctions). We call our network the IntersectNet. By using the LRCN approach to model the problem, we advocate the claim that visual features learned by CNN with writable memory support can model the internal representation and dynamics of video streams, as relevant to detection of traffic intersections. Due to the 'end-to-end' training capability, this model can update its parameters of the visual feature extractor as well as the parameters of the memory module together. We extensively test the model on video sequence datasets that contain intersections, and show the practical usefulness of this approach. We also show scenarios where the model is trained on one dataset and tested on another to test its generalizability.

The paper contributes in the following ways.

- It provides a solution to the sparsely studied yet a pertinent problem of intersection detection based on monocular camera as the sensing modality. This is in contrast with previous approaches that have primarily used laser range finders and point cloud features.
- This is also the first such method to use a very current architecture combining CNN with LSTM in the form of LRCN for intersection detection.
- By reporting performance gain vis a vis single frame CNN approaches it establishes the role of the temporal dimension in intersection/junction detection.
- Significantly accurate classification across a variety of datasets and across varied geography showcases the efficacy of the proposed method.

The remainder of this paper is organized as follows. Sec-

*Equal contribution
[1]Affiliated with KCIS, Robotics Research Center, IIIT Hyderabad
[2]Affiliated with IIT-Hyderabad

tion II discusses earlier work that have attempted intersection detection using other sensors, and motivates this work. Section III describes the proposed IntersectNet in detail. Section IV presents the results on popular road video datasets, and also compares this against using a spatial model (CNN) without considering the temporal dependencies. Section V presents additional results of classifying intersections further, as well as generalization results; and Section VI concludes with paper with pointers to future work directions.

II. Related work

There have been a few attempts to detect road intersections over the last few years, although with other sensing modalities. Mukhija et al. [3] combined camera data with lidar data for intersection detection. Techniques from image processing and computational geometry were used to extract a skeleton of the navigable region for intersection detection. Habermann et al. [4] extracted features from 3D point clouds collected using laser range finders. The features are first classified using conventional classifiers like support vector machines, adaptive boosting or artificial neural networks. Structured classifiers are then used to incorporate contextual information from neighboring frames for refined classification. Nie et al. [5] proposed another camera and lidar fusion approach for road intersection detection, where both the sensing modalities are together used to extract lane information, and intersection branches are then detected from the fusion of lane information.

Most of the existing algorithms are geometry-based and may not generalize well to varied intersections that an autonomous vehicle may encounter on its path. On the contrary, a data-driven approach, such as the proposed approach, is not limited in this sense. Considering the increasing access to large amounts of video data, data-driven methods for such problems may provide a viable and generalizable solution. Further, Lidar is expensive, and it may not be easy for all vehicles of the future to have such a system. Having an intersection detection system based on cameras alone is necessary, which we focus on in this work. Also, Lidar cannot be used to detect intersections from a longer distance as the density of the point cloud decreases with increasing distance from the sensor. An alternate solution to this problem can be proposed by combining GPS information with aerial maps to localize a vehicle near an intersection. However, the imprecision of GPS data and the reliance of this approach on an external source limits such an approach for practical use.

III. Methodology

Figure 1 presents the overall architecture of the proposed IntersectNet that brings together a Convolutional Neural Network (CNN) that spatially models the scene, and a Long Short-Term Memory (LSTM) network (which is an RNN variant) that models temporal relationships.

A. Convolutional Neural Network

Recent success of models based on Convolutional Neural Networks (CNN) in tasks ranging from object classification [6] to semantic segmentation [7], [8] suggests that composing highly non-linear layers in a sequential fashion acts as a very powerful visual feature extractor[9]. In this paper, we use CNN as a feature extractor which can be further utilized by memory based models for classification of road intersection. (Specific details of the network architecture are discussed later in this section.)

B. Long Short-Term Memory Network

Recurrent Neural Networks (RNNs) are designed with an idea of incorporating a feedback loop in their architecture. The problem with a traditional RNN is that they can not infer from long-term dependencies. As investigated in [10], assuming the network accepts a window of inputs ranging from time points, say, $t-\tau$ to t, the gradient flows from the output layer at time t all the way until the units at time $t-\tau$ during backpropagation. However, gradient values tend to vanish after a few time steps during the backpropagation, creating the '*vanishing gradient*' problem. This results in tiny to almost no change of weights for distant units, nullifying the impact of long-term dependencies on the output at time t. In order to tackle this problem, Hochreiter and Schmidhuber came up with the idea of Long Short-Term Memory networks (LSTMs), which are a special kind of RNNs with the capability to carry long-term dependencies [11][12]. Figure 2 shows a single LSTM unit. An LSTM contains a cell-state, C_t, which carries information from previous units, forgets unnecessary information and incorporates new information at each step (each of which is implemented as a layer of neurons with sigmoid activation functions, which act as gating functions). Information can travel easily along cell-state without being a subject of significant modification.

Here are the update equations of LSTM module,

$$
\begin{aligned}
\mathbf{i_t} &= \sigma(\mathbf{W_{xi}x_t} + \mathbf{W_{hi}h_{t_1}} + \mathbf{b_i}) \\
\mathbf{f_t} &= \sigma(\mathbf{W_{xf}x_t} + \mathbf{W_{hf}h_{t_1}} + \mathbf{b_f}) \\
\mathbf{o_t} &= \sigma(\mathbf{W_{xo}x_t} + \mathbf{W_{ho}h_{t_1}} + \mathbf{b_o}) \\
\mathbf{g_t} &= \tanh(\mathbf{W_{xc}x_t} + \mathbf{W_{hc}h_{t_1}} + \mathbf{b_c}) \\
\mathbf{c_t} &= \mathbf{f_t} \odot \mathbf{c_{t-1}} + \mathbf{i_t} \odot \mathbf{g_t} \\
\mathbf{h_t} &= \mathbf{o_t} \odot \tanh(\mathbf{c_t})
\end{aligned}
$$

Here, $\sigma(\mathbf{x})$ is a sigmoid unit. Each sigmoid unit output a vector with values between $[\mathbf{0}, \mathbf{1}]$. $\mathbf{f_t}$ is a forget gate which removes unnecessary information from the previous cell state ($\mathbf{c_{t-1}}$). Input gate $\mathbf{i_t}$ along with $\mathbf{g_t}$ add new information obtained at time $\mathbf{t}$ to the cell state. The updated cell state($\mathbf{c_t}$) and output gate ($\mathbf{o_t}$) yield $\mathbf{h_t}$. Here, $\odot$ indicates element-wise dot product.

C. Long-Term Recurrent Convolutional Networks (LRCN)

Long-Term Recurrent Convolutional Networks (LRCN) [2] provides a combination of CNNs and LSTMs, and we use a variant of this architecture in this work. In its most basic form, the LRCN has a visual feature extractor that projects input space dimension to the fixed length feature representation of dimension $\mathbf{V}$. The fixed length vector representations are time-invariant and independent. $\mathbf{V}$ is then

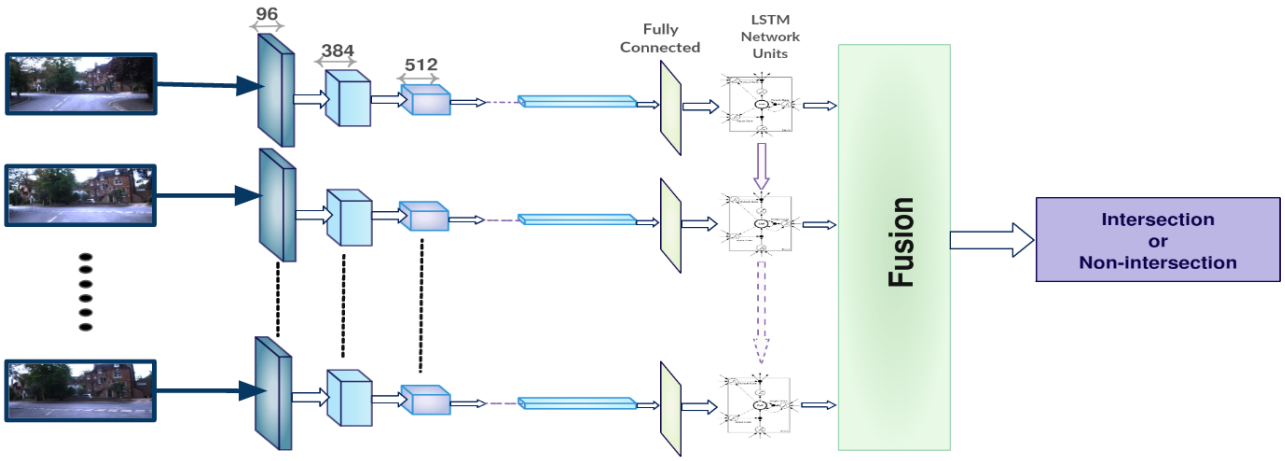

Fig. 1. Pipeline of the proposed IntersectNet. Each frame goes to a single instance of a CNN. At a time, 16 frames are fed to the network. The CNN gives rich set of features which are fed to LSTM units tied across time. These LSTM units are connected to a fully-connected layer. The softmax on fully connected layer gives label probabilities. In general, if T is the length of the input sequence and L is the total number of classes, output will be of size $T \times L$ (16×2 in our case).

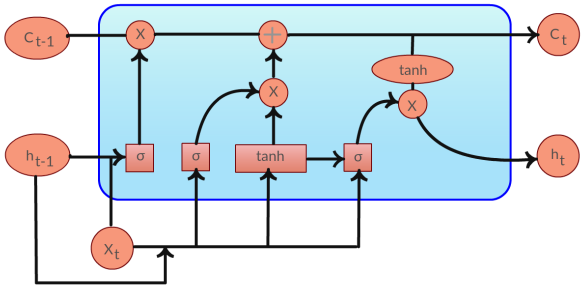

Fig. 2. Basic LSTM unit. Inputs are given by x_t, cell state at the previous time instant is C_{t-1}, history is h_{t-1}.

fed to a memory network (LSTM) along with the previous hidden state (which is obtained by running a CNN on the previous video frame), so as to produce next hidden state and output. The fusion of the deep hierarchical visual feature extractor and LSTM network enables end-to-end optimization of visual and sequential model parameters. The representation learned by the LSTM is then provided to a linear prediction layer. The predictions obtained by the LSTM network on each of the video frames of a given sequence are then provided to a fusion layer, whose output finally provides the classification result.

D. Why LRCN?

CNNs have been successfully applied on image recognition tasks, and it appears that one could use CNNs effectively on an image from the camera feed to recognize an intersection. However, we claim in this work that reliable detection of road intersections requires a spatiotemporal understanding, than just a spatial one. Road intersections could have various salient characteristics such as a human crossing the road, a car taking a turn, or just broadening of the road. These features can be captured over a video sequence (than just a single frame), thus necessitating a spatio-temporal model. We compare the performance of the LRCN model with single-frame CNN model for individual classes in Section IV, and show that the LRCN model outperforms the predictions from the single frame model by significant margin.

E. IntersectNet: Training and Network Parameters

This problem was approached using a variant of the LRCN, which is obtained by combining a CNN with an LSTM network[2]. This is an end-to-end trainable network. The CNN base of the network is a minor variant of AlexNet [6]. Since CNNs need very large amounts of data to be trained effectively (which we do not have here), we used the AlexNet CNN model trained on 1.2 million Imagnet LSVRC-2012 [13] classification dataset, which is a subset of ImageNet [14] dataset as our base, and finetuned this model on the datasets for road intersections. The weight initialization from a pre-trained network enables faster training with minimal over-fitting. Two variants of Long-term Recurrent convolutional networks(**LRCN**) were proposed in [2]. In IntersectNet, we use the first variant where the LSTM modules are placed immediately after fc_6 layer of the CNN modules (as in Figure 1).

The training part has **2000** epochs, and takes roughly **6** hours for Oxford dataset. While finetuning takes **1.4** hours. The training and testing had been carried out on Nvidia Titan X GPU. Caffe[15] was used to program the proposed architecture.

The proposed IntersectNet model is trained using Stochastic Gradient Descent(SGD). During backpropagation, gradients of the objective function are computed with respect to both, visual (CNN) as well as the temporal (LSTM) parameters. The LRCN network faces the '*exploding gradient*' problem, which is one of the key issues with LSTMs, where

the product of the gradients in each layer can explode when each value is relatively high. Pascanu et al. [16] proposed norm clipping to deal with this exploding gradient issue. In our work, we clip a gradient down to the value of $\mathbf{5}$.

Each input sequence to IntersectNet contains a video sequence of $\mathbf{16}$ contiguous frames. Each frame is an RGB image with size $\mathbf{240x320x3}$. A stride of $\mathbf{8}$ is used in the CNN. Each frame is further cropped to $\mathbf{227x227x3}$. Each CNN instance gives an output feature vector of size $\mathbf{1x4096}$. The features from all instances of CNN are then fed as input to the LSTM network. The size of input to the LSTM network is hence $\mathbf{16x4096}$. The output of each LSTM cell is then passed through a fully-connected layer and, finally, a soft-max layer which makes the output prediction. In the end, we introduce a fusion layer that combines outputs of all LSTM cells to provide the final prediction (in this work, we simply use average-pooling in the fusion layer).

IV. Experimental Results

A. Dataset Description

Considering that there is no explicit dataset publicly available for evaluating the performance of an intersection detection method, we extracted $\mathbf{110}$ intersection sequences and $\mathbf{200}$ non-intersection sequences from the Oxford Robot-Car dataset[17]. Each intersection sequence is temporally contiguous and consistent, and contains a road intersection. We define an intersection sequence as starting from a location where the intersection is fairly visible and reaching the junction.In addition, we also obtained intersection sequences from the popular Lara [18] traffic-light detection dataset. We extracted $\mathbf{22}$ intersection sequences and $\mathbf{40}$ non-intersection sequences from this dataset with the same defnition of an intersection sequence as before. Non-intersection sequences essentially have a road without any junction. Each Oxford sequence has $\mathbf{40-60}$ frames, while the Lara sequences have number of frames varying between $\mathbf{20-60}$. The Lara[18] dataset was collected with a speed $< \mathbf{30mph}$ and $\mathbf{25}$ frames per second.

B. Results

In this paper we show results to verify the superior performance of LSTM based models. We first trained adapted version of LRCN model on Lara dataset and carried out testing on Oxford dataset. Similarly the model trained on Oxford sequences was tested on Lara sequences. We also carried out testing on joint subset of both the Lara and Oxford datasets. The prediction accuracy are summarized in Table I

	ID1	ID2	ID3	ID4	ID5
Accuracy	78.25%	82.07%	94.14%	92.16%	72.052%
S.D.	1.01	2.71	2.7836	1.46	2.13

TABLE I

C. Dataset augmentation

As a part of dataset preprocessing, We flipped all the images of all the sequences around vertical axis passing through the center of the image. We subsequently added Gaussian noise with $\mathbf{0}$ mean and $[\mathbf{0.01}, \mathbf{0.1}, \mathbf{0.01}]$ as variance to **R, G, B** channels respectively. The noise was added to both original as well as flipped image. This way, we get $\mathbf{4}$ versions of same sequence with minor variations.

After data augmentation, we had $\mathbf{440}$ intersection and $\mathbf{800}$ non-intersection sequences from Oxford dataset, $\mathbf{88}$ intersection and $\mathbf{160}$ non-intersection sequences from Lara dataset.

D. Single-Frame vs LRCN

In this section, we compare the performance of LSTM based models with conventional single frame convolutional models. For single frame model, N individual frames are classified independently by CNN and final classification is done by averaging scores across all video frames. Table II reports the accuracy of performance. As evident from the numbers, IntersectNet outperforms the conventional single frame CNN network by a margin of nearly $\mathbf{5.5}\%$ when trained on Lara dataset and roughly a margin of $\mathbf{2.5}\%$ when trained on Oxford datasets.

	Trained on Lara	Trained on Oxford
Single Frame	72.72%	69.58%
LRCN	78.25%	72.05%

TABLE II

ID1 : IntersectNet trained on Lara sequences and tested on Oxford sequences
ID2 : IntersectNet trained on Lara sequences and tested on Lara and oxford sequences
ID3 : IntersectNet trained on Oxford, finetuned on Lara sequences and tested on Lara
ID4 : IntersectNet trained on Lara, finetuned on Oxford sequences and tested on Oxford
ID5 : IntersectNet trained on Oxford and tested on different Lara sequences
S.D. : Standard Deviation

V. Discussion and More Results

A. 3 class classification

Apart from detecting the potential intersections, it is crucial for the autonomous vehicle to categorize the intersection based on its geometry. We extend the problem of binary classification of road intersection to 3-class classification problem. We classify the road geometry as into one of following classes:

1) Non Intersection
2) T-junction
3) Cross junction

At road junctions, behavior of autonomous agents should change based on busyness of the road. The agent should be equipped with ability to adapt itself with dynamic environment. A small junction of $\mathbf{T}$ shape might not be as dangerous as multiple roads meeting at a common junction.

(a) Classified as Intersection (b) Classified as Non-Intersection (c) Classified as Intersection

(d) Classified as Intersection (e) Classified as Intersection (f) Classified as Non-Intersection

(g) Classified as Intersection (h) Classified as Non-Intersection (i) Classified as Intersection

(j) Classified as Intersection (k) Classified as Non-Intersection (l) Classified as Intersection

Fig. 3. Qualitative results: Our trained model was tested on various sequences from different datasets, spread over different continents.

It is essential to enable field agent to understand complexity of the surrounding environment.

If the vehicle has only two degrees of freedom to move, it can either turn on left/right or could continue on the same path. We are defining such junctions as **T** shaped junction. In other words, these junctions have only single turn.

If the vehicle has multiple degrees of freedom to move, we categorize such intersections as Cross(+) junction. Any junction with multiple roads meeting will be classified as cross junction. The autonomous agent needs to be more careful near such a junction, as there are higher chances of road accidents because of its complex nature.

For this proposal, we used Oxford dataset for training and testing. There were **60** Cross junction sequences, **47** (T) junction sequences and **200** non-intersection sequences. We randomly sampled non-overlapping set of sequences for training and test set. The sampling was carried out for **10** times. The training was carried out for **10** times.Table III summarizes the average accuracy.

	3-Class Classification
Accuracy	91.7208%
S.D.	2.1772

TABLE III

B. Finetuning and Generalization

In this section, we investigate the ability of network to adapt on new and relatively smaller dataset. In table I, we summarized the performance of IntersectNet trained on Lara sequences. The Lara dataset was extensively tested on Oxford sequences. In this section, our key idea is to adapt the Lara trained model using a small subset of Oxford dataset. The Oxford and Lara datasets have really different temporal features. Lara dataset has lower frame rate, more vehicle speed and significantly different surrounding as compared to Oxford.We assess on how we can effectively transfer learned model of Lara dataset to Oxford. With as less as 150 Oxford sequences, we finetuned a model trained on Lara and tested on 700 Oxford sequences. Table IV compares the performance of IntersectNet on Oxford dataset before(**ID1**) and after(**ID4**) finetuning. As evident from the figures, the model show significant boost in accuracy after finetuning. We demonstrate that the network quickly adapts new sequential features and generalizes for a larger dataset.

	ID1	**ID4**
Accuracy	78.25 %	92.16 %
Precision	0.7565	0.9364
Recall	0.8593	0.9581
F1 measure	0.8046	0.9471

TABLE IV

C. Qualitative Results

Figure 3 shows the qualitative results for the intersection recognition. Our proposed method makes fairly well predictions over diverse road geometries under various illumination conditions. The top row of 3 shows the classification over sequences from Oxford dataset. Second row shows results on Lara dataset. Third row shows accurate classification on sequences collected from a youtube videos. In the last row, we show results of our model evaluated on sequences collected in Indian road conditions. Even if the road in front of the vehicle is partially occluded, as in Figure 3(c), we are able to make correct predictions by learning the temporal dynamics of the environment.

VI. CONCLUSIONS

This paper presents a robust method for road intersection detection. The temporal dimension to the intersection problem gets vividly captured through superior performance of the LRCN architecture vis a vis a single frame CNN classifier. Our approach gives efficient and extensive results on a variety of road junctions. As self driving cars get close to reality, the ability to detect a diverse category of road junctions becomes inevitable. The scalability and generalizability of the proposed architecture makes it competitive for the intersection detection problem

While we have attempted to detect an intersection reliably in this work, we believe that the proposed approach can also be used to detect various stages of approaching an intersection and leaving an intersection, which we will attempt in future work.

REFERENCES

[1] "Statistics on intersection accidents." [Online]. Available: https://www.autoaccident.com/statistics-on-intersection-accidents.html

[2] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[3] P. Mukhija, S. Tourani, and K. M. Krishna, "Outdoor intersection detection for autonomous exploration," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, Sept 2012, pp. 218–223.

[4] D. Habermann, C. E. O. Vido, F. S. Osrio, and F. Ramos, "Road junction detection from 3d point clouds," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 4934–4940.

[5] Y. Nie, Q. Chen, T. Chen, Z. Sun, and B. Dai, "Camera and lidar fusion for road intersection detection," in *2012 IEEE Symposium on Electrical Electronics Engineering (EEESYM)*, June 2012, pp. 273–276.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[8] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[10] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] "Understanding lstm networks," Aug 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[16] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks." *ICML (3)*, vol. 28, pp. 1310–1318, 2013.

[17] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017. [Online]. Available: http://dx.doi.org/10.1177/0278364916679498

[18] L. team (mmClean template by Marcin Mierzejewski @ Zenzire), "Traffic lights recognition (tlr) public benchmarks." [Online]. Available: http://www.lara.prd.fr/benchmarks/trafficlights recognition