Ego-motion Classification for Driving Vehicle

Li Du*, Wenhui Jiang*, Zhicheng Zhao*, Fei Su*
*Beijing University of Posts and Telecommunications, Beijing, China

***Abstract*—Accurate prediction of vehicle ego-motion in real time is crucial for an autonomous driving system. In this paper, we formulate the problem of ego-motion classification as video event detection, and we propose an end-to-end deep model to address this problem. In this model, we utilize Convolutional Neural Networks (CNNs) to extract semantic visual feature of each video frame, and employ a Long Short Term Memory (LSTM) to model the temporal correlation of the video streams. To study the performance of ego-motion classification, we constructed a video dataset-Campus20, which captured in general driving conditions. Experimental results on Campus20 verifies the superior performance of our proposed model over well established baselines.**

***Keywords*-ego-motion classification; Convolutional Neural Networks (CNNs); Long Short Term Memory (LSTM);**

I. Introduction

Autonomous driving has become a promising research topic in the field of public transport. Accurate and efficient classification of vehicle ego-motion is one of the most important parts of a self-driving system. Compared with recognizing video events from a static camera, detecting vehicle ego-motion is more challenging, because videos recorded by a vehicle camera contains large amounts of fast moving objects due to the continuous moving nature of a driving car.

In this paper, we formulate the problem of ego-motion classification as event detection in video streams:given video streams recorded in real time, we categorize each frame into one of possible action states (turning, lane-changing, reversing, lane-following, crossing, turn-left and turn-right). Towards this goal, we propose an end-to-end deep learning architecture based on CNN and a single layer LSTM. In this model, each video frame is fed into a CNN part in sequential order to extract semantic visual features individually. Then the represented visual features are connected to a LSTM part to model the temporal correlation of the video sequence. Finally, we estimate a probability distribution over all possible ego-motion action. A brief illustration of our proposed model is shown in Figure 1.

Our architecture design is based on the following observations. Firstly, CNNs excel at capturing complex spatial characters of a input images with their multiple layer receptive fields, which enable them to obtain a better representation of image. Secondly, Long Short Term Memory (LSTM) is good at capturing long-range temporal relationships from the input sequence, by using memory cells to store, modify and access internal state in visual feature based task [13].

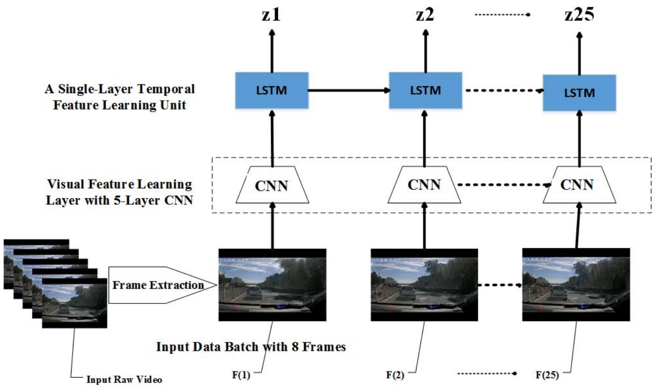
z1
z2
z25
A Single-Layer Temporal Feature Learning Unit
LSTM
LSTM
LSTM
Visual Feature Learning Layer with 5-Layer CNN
CNN
CNN
CNN
Frame Extraction
Input Data Batch with 8 Frames
Input Raw Video
F(1)
F(2)
F(25)

Figure 1. The deep learning architecture of our model.

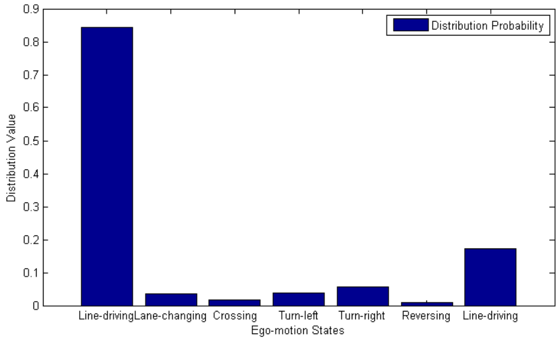

Figure 2. The ego-motion distribution probability bar image of dataset.

To provide a better benchmark on ego-motion classification, we collect a new video dataset, named Campus20. The data was captured by a single front automobile data recorder mounted on a vehicle driving around BUPT in Beijing. We compare our method with various baselines (e.g SVMs [10] and ELM [6]) based on traditional hand-crafted features) on Campus20. Experimental results show that our model contributes to significant performance improvements compared with traditional feature-based baselines.

II. Related Work

Ego-motion, also named camera motion, is one of the most important research areas in computer vision, which benefits from advanced image sensor research and has potential for completing ego-motion related tasks in autonomous driving system. Johua et al. [5] proposed a method to recover

978-1-5090-6549-3/17 $31.00 © 2017 IEEE
DOI 10.1109/BigMM.2017.25

observer's ego-motion using omnidirectional cameras, which showed the possibility of ego-motion calculation based on efficient visual scene capture device.

In 2000, MobileEye [14] proposed a robust approach to calculate the vehicle ego-motion relative to the road based on camera videos, and built a reliable method which can ignore large number of outliers happening in real driving conditions. In 2015, to make out how images of objects and scenes behave in response to specific ego-motion, Jayaraman et al. [7] made motor signals as unsupervised information in CNNs, for learning visual representation from egocentric video. To address the problem of visual ego-motion estimation or briefly Visual Odometry (VO), in [2], an approach to learn both the best visual features and the best estimator based on Neural Networks was proposed.

So far, newly emerging autonomous driving research platforms like KITTI Vision Benchmark Suite [4], Google CityScape [1] and Oxford RobotCar dataset [11], all developed relative novel challenging real-world computer vision benchmarks related to autonomous driving platform. To accomplish a more accurate real-time ego-motion prediction task, we come up with a simple deep learning architecture based on CNNs and LSTM.

The contribution of this paper as follows: we proposed an efficient ego-motion classification method based on traditional hand-crafted features, and used the low dimensional temporal combination features to complete ego-motion classification task about driving relying on practical experience, which was verified with a relative stable performance on randomly recorded dataset. And it also demonstrated that visual features can be used in driving event classification task; At the same time, ELM was also used to verified the potential of neural network in dealing with ego-motion classification task under complex dynamic conditions; Then a stacked deep learning architecture based on CNN and RNN (LSTM) was also proposed to complete the same classification task.

III. Dataset

To our knowledge, there is no dataset public available for ego-motion classification. In this paper, to provide a better benchmark on ego-motion classification, we collected a new video dataset, named Campus20. The videos are recorded on some typical roads in 20 different days. Data was collected in the clear days, in the day time.

The dataset consists of 15 videos for training and 5 videos for testing, each lasts for 5 mins.The frame rate is 18 FPS, and the spatial resolution is 320×240 pixels. Each frame is labeled with a specific state of ego-motion action. As shown in Figure 2, from statistical view, the distribution probabilities of ego-motion in Campus20 like line-driving, lane-changing, crossing, turn-left, turn-right, reversing, are respectively 84.381%, 3.469% , 1.585%, 3.923%, 5.610%, 1.028% and 17.133%.

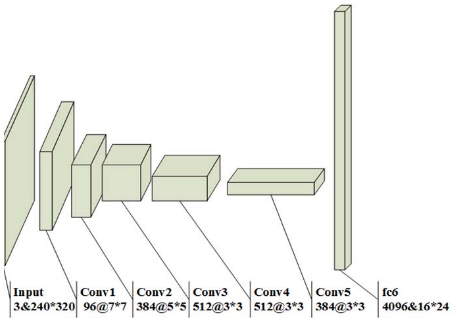

Figure 3. The architecture of our model based on CNN+LSTM, when applied to process a 25-frame video sequence.

IV. Model Description

We propose an end-to-end deep model to address the problem of ego-motion classification. The overall architecture is shown is Figure 2. It takes raw video sequences as inputs and output is the probability distribution of the corresponding video frames. The network consists of two parts, a CNN part which works as a visual feature extractor and a LSTM which acts as a temporal feature extractor. We will explain both parts in detail in the following subsections.

A. CNN Part

We build the CNN part from one of the popular backbone architectures, i.e., AlexNet [9]. As shown in Figure 3, it mainly contains 5 convolutional layers, one fully connected layer. The kernel size of each convolutional layer is 7×7, 5×5, 3×3, 3×3 and 3×3. At the same time, a max pooling layer with 3×3 kernel and 2 stride interval is connected to the 1st, 2nd and 5th Relu layer. A dropout layer is appended to the fully connected layer to prevent over-fitting.

B. LSTM Part

The LSTM part is designed to extract continuous temporal features of video sequences given the visual features extracted from the CNN part. As depicted in Figure 4, each LSTM cell remember a single floating point value (t). This value may be diminished or erased through a multiplicative interaction with the forget gate (t) or additively modified by the current input (t) multiplied by the activation of the input gate $\imath(t)$.The output gate $\o(t)$ controls the emission of (t) [15]. When fed a visual feature sequence, the LSTM part computes the hidden vector sequence and outputs a continuous temporal feature sequence.

In this part, a single-layer LSTM with 256 hidden unites is used to extract temporal relationships features within each frame sequence. Besides, a single-layer fully connected layer is designed to function as a controller for classification.

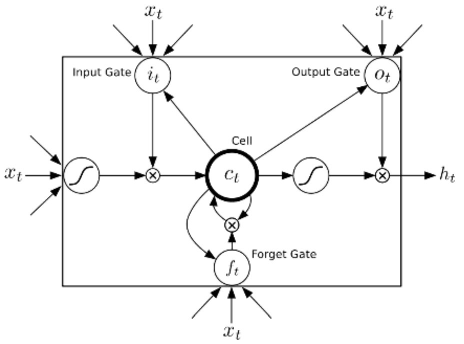

Figure 4. A simple LSTM block with only input, output, and forget gates.

C. Objective Function

The whole network is supervised by softmax loss, the loss function is shown as Equation (1) below:

$$\mathcal{L} = \frac{1}{N}\sum_{i=0}^{N-1}\log(\sum_{j=0}^{C-1}e^{z_{i,j}} - z_{i,y_i}) \tag{1}$$

In the equation above, N is the length of an LSTM sequence, C is the number of action states. We define z as the output of our model, z_{i,y_i} stands for the predicted probability value of the ith frame belongs to action state y_i and $z_{i,j}$ represents the ground truth probability value when the ith frame belongs to the jth state.

The optimization can be done by applying Stochastic Gradient Descent (SGD) algorithm with Back-propagation (BP), we will explain the training details in next section.

V. Experiment

A. Training Details

We use CNN parameters pre-trained on UCF-101 [3] to initial the visual CNN part of our model. We used a GTX 1080 GPU and implement the network in Caffe [8]. We trained our model jointly with stochastic gradient descent(SGD)algorithm with an initial learning rate of 10^{-3}. The learning rate is decreased to 10^{-4} after one epoch. The learning lasts for 4 epochs.

As shown in Section 3, the total distribution probability of lane-driving is nearly 80%. To avoid the learned model biased towards 'lane-driving', we adopt the balance sampling strategy. Specifically, we set the distribution probability of negative sample(lane-driving) to positive samples(crossing, reversing, lane-changing, turn-left and turn-right) as 3:1, and each mini-batch contains 100 video frames.

The CNN part in our model output 25×4096 data, when fed into a 25-frame video sequence each time step. And the LSTM part, which follows the CNN part output a data sequence consists of 25×7 data. Besides, the output of each time step is a class sequence consisting of 25 numbers, and the range of each number from 0 to 6.

1) Baseline: To reveal the effectiveness of deep neural network in dealing with ego-motion classification task in complex dynamic conditions, traditional method based on kernel SVMS and ELM with 150 hidden nodes, were chosen as the baselines.

A synthesized model based on four hand-crafted temporal features, namely motion [10], optical flow [12], velocity and improved dense trajectories [12] is proposed. Each kind of feature is normalized separately and concatenated to form the final feature vectors.

The primitive classifiers for the six action states(turning, lane-changing, reversing, lane-following, crossing, turn-left and turn-right) are trained by kernel Support Vector Machines (SVMs) and multiple cross validation is used to select the optimal parameters including the cost C and relative weight of positive and negative sample sets. To verified the higher calculation efficiency of ELM over SVM in dealing with ego-motion classification task under complex dynamic conditions, ELM is also used as a baseline based on its rigorous theory demonstration [6]. Besides, ego-motion classification result come from a similar classification model in [3] is also used as a comparison to verify the efficiency of our model.

2) Analysis: We evaluate the efficiency of the proposed model on Campus20, quantifying the number of the processing frames on ego-motion classification performance and revealing the importance of temporal relationships in continuous sequence in accurate ego-motion classification task. The performance is evaluated by classification error rate, which is defined as follows:

$$Err = \ N_i/N_t \tag{2}$$

Where N_i is the number of frames being incorrect classified, and N_t stands for the total frame number of the corresponding classification dataset.

B. Experimental Results

As shows in TABLE I, for ELM with 150 hidden nodes and kernel SVMs, both two methods get the nearly equal results in ego-motion classification task and ELM gets a lower classification error rate than kernel SVMs. At the same time, when processing data with a same scale, ELM gets a mild classification accuracy improvement and a less frame processing time (4×10^{-7} s to 0.006 s) compared with SVMs. The results demonstrate that as one kind of single layer feedforward neural network, ELM is more efficient in dealing with complex computation problems than SVMs, and single layer neural network also perform in our task.

Due to the fact that data in real-world is almost complex. Efficient traditional hand-craft features extraction and representation approaches are not always reliable, for they rely heavily on expert knowledge of researchers. So the traditional feature-based method may hold little potential for meeting practical need of ego-motion classification task.

Table I
Overview of baseline and proposed Models

Method	Error Rate	Testing Time
SVM	0.2375	0.006 s
ELM	0.2473	4×10^{-7} s
CNN+LSTM [3]	0.1368	0.04 s
CNN+LSTM	0.0417	0.034 s

In comparison, we can see that the multiple ego-motion classification error rate of our CNN-LSTM architecture is 0.0417and the result of the similar model in [3] is 0.1368. And the result of SVMs is 0.2375 and ELM is 0.2473. The results above show that models based on deep learning have better visual and temporal features classification ability over traditional methods, and the classification ability of our model is superior to the similar model architecture in [3].

And as verified in experiments above, deep learning method may work well in dealing with complex ego-motion classification task, depending on its more efficient learning progress and the powerful calculation ability of GPU.

VI. Conclusion

We propose a real-time ego-motion prediction task related to autonomous driving. To complete the task, we come up with a CNN+LSTM model and collected a new dataset to demonstrate the efficiency of our model. Comparing experiments between traditional method (SVMs and ELM) and our model verified the power classification ability of our method, and the proposed architecture get more reliable features based on powerful automatic feature learning ability of CNN and LSTM, which contribute to accurate ego-motion classification task.

Acknowledgment

This work is supported by Chinese National Natural Science Foundation (61372169 and 61532018).

References

[1] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] G Costante, M Mancini, P Valigi, and T. A Ciarfuglia. Exploring representation learning with cnns for frame-to-frame ego-motion estimation. *Robotics Automation Letters IEEE*, 1(1):18–25, 2016.

[3] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] Joshua Gluckman and Shree K Nayar. Ego-motion and omnidirectional cameras. In *Computer Vision, 1998. Sixth International Conference on*, pages 999–1005. IEEE, 1998.

[6] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

[7] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1413–1421, 2015.

[8] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross B. Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *CoRR*, abs/1408.5093, 2014.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *International Conference on Neural Information Processing Systems*, pages 1097–1105, 2012.

[10] Jianjiang Lu, Yulong Tian, Yang Li, Yafei Zhang, and Zining Lu. A framework for video event detection using weighted svm classifiers. In *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*, volume 4, pages 255–259. IEEE, 2009.

[11] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[12] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015.

[13] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pages 2204–2212, 2014.

[14] Gideon P Stein, Ofer Mano, and Amnon Shashua. A robust method for computing vehicle ego-motion. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 362–368. IEEE, 2000.

[15] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.