ARTICLE IN PRESS
JID:VEHCOM AID:104 /FLA [m5G; v1.224; Prn:8/11/2017; 10:56] P.1 (1-7)
Vehicular Communications ••• (••••) •••–•••

Contents lists available at ScienceDirect

Vehicular Communications

www.elsevier.com/locate/vehcom

Smart in-car camera system using mobile cloud computing framework for deep learning

Chien-Hung Chen, Che-Rung Lee *, Walter Chen-Hua Lu

National Tsing Hua University, HsinChu, 30013, Taiwan

ARTICLE INFO

Article history:
Received 17 December 2016
Received in revised form 9 July 2017
Accepted 29 October 2017
Available online xxxx

Keywords:
Deep learning
GPU
Mobile cloud

ABSTRACT

Deep learning is becoming a popular technology in various applications, such as image recognition, gaming, information retrieval, for intelligent data processing. However, huge amount of data and complex computations prevent deep learning from being practical on mobile devices. In this paper, we designed a smart in-car camera system that utilizes mobile cloud computing framework for deep learning. The smart in-car camera can detect objects in recorded videos during driving, and can decide which part of videos needs to be stored in cloud platforms to save local storage space. The system puts the training process and model repository in cloud platforms, and the recognition process and data gathering in mobile devices. The mobile side is implemented in NVIDIA Jetson TK1, and the communication is carried out via Git protocol to ensure the success of data transmission in unstable network environments. Experimental results show that detection rate can achieve up to four frame-per-second with Faster R-CNN, and the system can work well even when the network connection is unstable. We also compared the performance of system with and without GPU, and found that GPU still plays a critical role in the recognition side for deep learning.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

Deep learning has become the most promising machine learning approach. From LeCun's LeNet [1] to Khrizevsky AlexNet [2] and recent GoogleNet [3], deep learning has shown its capability of solving difficult computer vision problems [4]. Its detection performance surpasses other artificial classifiers which reply on the hand-crafted features. With the success in vision work, deep learning also attracts attentions from other fields, such as sentiment analysis [5], language processing [6], region of interest localization and description [7], and medical use [8].

The success of deep learning is brought off by three factors: the advance of numerical optimization methods, growth of data volume, and fast computational hardware [9]. New numerical methods solved the convergence problems, which are more and more critical when the number of layers goes deeper and deeper. Large enough training data sets make the extracted features by deep learning sufficiently representative. These required data can be continuously collected by the sensors equipped in modern embedded systems and mobile devices. Last, the accelerators, such as Graphic Processing Units (GPUs), provide strong computing power to support the training of deep learning model.

In the era of Internet of Things (IOT), the deployment of deep models to mobile devices is a nature request. One the one hand, the mobile devices can be smarter with the deep learning ability. Moreover, they also help the data gathering from various sources. On the other hand, the limited power, storage, and computational resources of mobile devices prevent the complex computation, like deep learning, from being practical. Therefore, the mobile cloud computing model, which combines the mobility of mobile devices and the computational resources of cloud platforms via ubiquitously accessible network, becomes a practical solution for mobile applications that utilize deep learning.

Mobile cloud computing has three types of models to coordinate the works between mobile devices and cloud platforms [10]. The first one is to off-load all the work to the cloud platforms, and the mobile devices take care of data input and output only. However, such model usually require frequent data communication, which is not suitable for heavy data transmission. In addition, when the network is unstable, mobile devices cannot work alone. The second model relies on mobile devices to handle most of the works. Such model is only limited to light weighted works that can be processed on mobile devices. The last one splits the works to mobile devices and cloud platforms. This model could balance the

* Corresponding author.
E-mail address: cherung@cs.nthu.edu.tw (W.C.-H. Lu).

https://doi.org/10.1016/j.vehcom.2017.10.001
2214-2096/© 2017 Elsevier Inc. All rights reserved.

computation and communication, but requires good synergy from both sides.

Most of the current solution of mobile deep learning utilizes the first solution since the computation of deep learning is heavy. In this paper, we utilized the third model for deep learning, which puts the training and model repository on the cloud platform, and recognition and data gathering on the mobile devices. Such architecture cuts the storage and computation requirement of mobile devices and the data transmission between cloud and end users. In addition, we employed the Git protocol for data communication data so that the transmissions can be resumed even the network connection is not available sometimes.

We used the smart in-car camera as an example application to demonstrate how the framework works. The smart in-car camera can select suitable deep learning models for video recognition, decide which part of video clips contain important information and send them to cloud platform, and update the deep learning models when necessary. We have implemented the system on NVIDIA Jetson TK1 embedded development board. Experimental results show that detection rate can achieve 2.8 to 4 FPS (frame-per-second) with Faster R-CNN, and the system can work well even when the network connection is unstable. We also compared the performance of system with and without GPU. The result shows about 20 times speedup can be obtained with the acceleration of GPU.

The major contributions of this paper are

- Proposed a split mobile cloud model for deep learning.
- Demonstrate how to use Git protocol in the mobile cloud architecture.
- Implement a smart in-car camera system that can save storage and computation.
- Compare the performance of deep learning inference with and without accelerators.

The rest of this paper is organized as follows. Section 2 introduces background knowledge and related works. Section 3 presents the framework and the implementation details of the smart in-car camera. Section 4 shows the experimental results of system performance. The conclusion and future directions are given in the last section.

2. Background

2.1. Deep learning

Deep learning, or deep neural network, refers to a neural network that consists of one input layer, one output layer and multiple hidden layers. The earliest appearance of deep neural network can be traced to late 1960s in the work published by Alexey Grigorevich and V.G. Lapa [11]. However, deep learning grows at a slow pace due to immature training scheme and architecture in the next few decades. In the 1990s, LeCun trained LeNet, a deep neural net with convolution and pooling layers, with back-propagation algorithm for digit recognition [1]. Stochastic gradient descent was invented in the 1970s to train artificial neural networks.

LeNet is the earliest work to take deep learning into recognition task. In 2006, layer-wise unsupervised pre-training was proposed for parameter initialization [12]. The new training method has two phases. In the pre-training phase, every two consecutive hidden layers are considered a restricted Boltzmann machine and weights are adjusted with an unsupervised learning manner. After pre-training, the whole model is fine-tuned with an additional supervised layer in the second phase. Pre-training makes layer parameters get better values compared to random initialization. As a result, trained models reach more sensible local minimum and quality gets more stable. In 2012, Krizhevsky et al. developed the eight-layer AlexNet and won the ILSVRC 2012 prize with about 85% classification and 66% location accuracy [2]. AlexNet won their competitors who used linear classifier over ten percentage. And the winners of ILSVRC classification task in the following years all used deep neural networks. Instead of using handcrafted classifiers, deep neural network learns high level features during training time. And the ILSVRC challenge results prove its high performance.

With big success in image classification, deep learning attracts focus from other fields. Beside from classification, deep learning is also used in object detection [13–15], speech recognition [16,17], language processing [6], sentiment analysis [5], and many other works. Recent deep learning bloom can be ascribed to new optimization methods, appearance of more powerful processor and rapid growing data volume. With more powerful CPU and multicore processor, especially general purpose GPU, training time can be cut down from months to days. With growing of various data, under-fitting can be prevented and makes deep learning be applied to solve different types of problem. Latest publications not only improve accuracy but boost performance. PReLU activation function pushes classification accuracy over 95% and saves time for deriving gradient [18]. Dropout prevents training from over-fitting [19]. Also, new initialization schemes make pre-training phase unnecessary [20].

The applications of deep learning to object detection for vehicles driving also receives many investigations [21–23]. Unlike them which concerns how to improve the performance of recognition or speed, this paper addresses the issue of system integration for practical considerations, such as package loss and model updates.

2.2. Deep learning packages

2.2.1. OverFeat

OverFeat [15] claims that it is the first work to solve location and classification problem via deep neural network. The team used two networks to handle two vision tasks correspondingly. Regression network and classification network share previous five convolutions with max pooling layers. Then the two networks predict object class and appearance confidence using the same feature map separately. To handle object not at the center of the predicted box or only part of it in the box, input images would be resized to generate different scales of original image. Various scales of images then fed into shared feature extraction layers. Next, max pooling would be conducted on feature maps with different sizes. A fixed sized sliding window goes over all locations of the pooled feature maps and predicts results at each spot. Final result can be determined after merging. OverFeat won ILSVRC 2013 classification and location prize with an under 0.3 error rate.

2.2.2. R-CNN, fast R-CNN and faster R-CNN

Motivated by Alex's success in ILSVRC 12, Girshick et al. developed R-CNN [13] for generalizing convolutional neural network based method to object detection. R-CNN consists of three parts: candidate selection, feature extraction and classification. First, R-CNN chooses selective search to find object proposals in the input images for convenience of comparing with other works. In the feature extraction part, proposals are warped to fix size images and fed into a convolutional neural network. In the last part, a supporting vector machine (SVM) classifies proposals according to feature vectors from previous stage.

Despite R-CNN achieves relatively better result, it is very time consuming during both training and detection time. Fast R-CNN [24] was published to boost performance in 2015. The main bottleneck of R-CNN performance is feeding proposals into deep neural net and calculating features separately. Fast R-CNN introduced ROI pooling layer to share calculated result generated from convolutional neural net. ROI layer extracts fix sized small feature

Please cite this article in press as: C.-H. Chen et al., Smart in-car camera system using mobile cloud computing framework for deep learning, Veh. Commun. (2017), https://doi.org/10.1016/j.vehcom.2017.10.001

Download English Version:

https://daneshyari.com/en/article/6890179

Download Persian Version:

https://daneshyari.com/article/6890179

Daneshyari.com