Caffe Framework on the Jetson TK1: Using Deep Learning for Real Time Object Detection

Christopher Alicea-Nieves (University of Puerto Rico at Mayaguez, Computer Engineering, *SUNFEST Fellow*)

Dr. Camillo J. Taylor, Computer and Information Science, University of Pennsylvania

Abstract— **Deep Neural Networks are an approach to using state-of-the-art machine learning algorithms in order to improve the applications of the computer vision field. These neural networks are complex mathematical models which can be trained to identify certain objects within a frame, and some of them are also able to identify where on the image is the object that they identified. This technical report focuses on presenting the implementation of a deep neural network on a low-power embedded computer system, specifically the NVIDIA Jetson TK1, in order to run convolutions in real-time for object detection. This can be achieved by implementing Fast R-CNN–a state-of-the-art model for convolutions–in the machine learning framework called Caffe.**

***Index Terms*—**caffe; convolutions; Jetson; machine learning; neural networks.

I. Introduction

Computer vision is a field of computation that has experienced a steady growth of popularity over the past few years because of all the useful applications that have been and are being developed. From counting cars to recognizing people's faces, Computer vision is a powerful tool that has not yet reached its maximum potential. In the past few years, since the ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC) [2], the integration of Deep Learning algorithms based on convolutions has been an increasingly popular approach for various tasks, like object detection and labeling. That has been a motivation for developing approaches that make use of Graphic Processing Units (GPUs), since they excel at running these computations. In order to achieve real-time implementation on our system, we must explore the most memory-efficient methods and train networks specifically for the tasks we wish to achieve. We are using the NVIDIA Jetson TK1 because it is a low-power system equipped with a dedicated GPU.

II. Background

2.1 Deep Neural Networks

A Convolutional Neural Network is based on different layers that look for features in an image. Regions with Convolutional Neural Networks (R-CNN) [6] are a variation of the regular CNNs. They first analyze the frame and propose different regions for different objects, and then proceed to run the CNN on each separate region. This model is more accurate in the PASCAL VOC [3], which is another dataset challenge but for object detection, hence the reason why we will be using it. The region proposal process is very expensive; it is a computational bottleneck in running R-CNNs. Fast-Regions with Convolutional Networks (Fast R-CNNs) [7] are designed to avoid this bottleneck and speed up the process of running convolutions using this network architecture. Most of the pre-trained models are very big because they were trained for large-scale datasets, resulting in a slower performance for a real-time implementation. To achieve real-time performance, we will have to create our own model based on this architecture and train it on the specific categories that we will need.

2.2 Caffe Framework

Caffe [1] is a modular machine learning framework, which means that it is open source and modifiable. It provides access to state-of-the-art deep learning algorithms in order to develop, train, and test models. The framework is based on a convolutional architecture because it has been the leading state-of-the-art deep architecture for image processing. These Convolutional Neural Networks (CNNs) are being used to train models in order to detect objects and label them, besides other implementations like gender recognition and age estimation. One of the challenges with the use of these CNNs is that they require significant computational power. There are many approaches that use variations of the CNNs in order to achieve better performance. We will be exploring these approaches and implement them on the Jetson TK1.

2.3 The datasets

- **ImageNet Large Scale Visual Recognition Challenge (LSVRC)**

ImageNet [2] is a database of hundreds of thousands of pictures labeled in over a thousand categories. This challenge serves as a benchmark for testing different Deep Learning algorithms, especially those based on CNNs. The challenge is simple, ImageNet gives the company/researcher access to a set of 50,000 images used to train the model, and then that model is deployed to analyze and label 1.2 million images that it has never seen before. The labels are then analyzed and two results are used to describe the efficiency of the model: the accuracy, and the error rate. This challenge is not the only one, but it is the most commonly used for comparison purposes. The mission is to promote the development of better algorithms, techniques, and models to achieve increasingly better performance. The main issue is that most of these methods are designed to run on super-computers, making them very difficult to implement in real-time applications.

- **The Pattern Analysis, Statistical Modelling and Computational Learning Visual Object Classes Challenge (Pascal VOC)**

The PASCAL VOC [3] is another database of thousands of images. The main detection challenge consists of 20 categories of labeling and detection, which not only means being able to recognize what is in the frame, but also where it is. This is the main difference between ImageNet and PASCAL. The datasets of training images and validation images are accessible to use for research purposes through the development kits, which include the training images with their annotation files, and testing images.

2.4 Implementation with the Jetson TK1

We are using the NVIDIA Jetson TK1 embedded computer system. It is a low-power computer system, and our interest is to use it for mobile applications. The goal is to deploy a trained model using Caffe that will analyze the frames from a camera as they are being captured, and detect and localize the object(s) within the frame. The advantage of using CNNs for this task is that it is more flexible and intelligent.

III. Related Work

Besides Convolutional Neural Networks, which have been at the vanguard of object classification and detection in the past few years, there are other approaches to object detection, DeformablePartModels(DPMs) [10] being one of those approaches. DPMs have a wide variety of implementations using different techniques, for example Histogram of Orientated Gradients (HOG) [14], or the recent Generalized Sparselet Models[13]. We are using Fast R-CNN, which is based on R-CNNs, which is not the only convolutional network structure. Other CNN models are based on Very Deep Convolutional Networks for Large-Scale Visual Recognition[11], commonly known as the Visual Geometry Group (VGG) networks, and the DeepPyramid DPM described in [12].

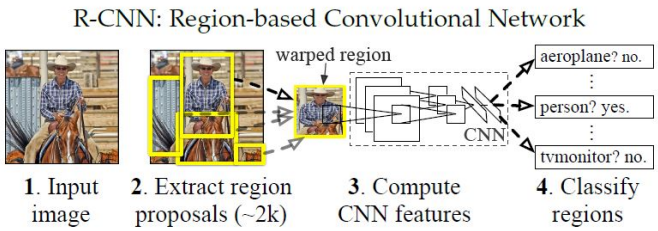

Figure 1: Object Detection System based on Regions with Convolutional Neural Networks [6]. The system takes an input image, and proceeds to extract region proposals for it. Afterwards, it processes the proposed regions in to the Convolutional Neural Network for classification and localization.

IV. Experimental Results

Caffe [1] is an open source framework for machine learning algorithms. It is used to design, train, and test models on various applications. This framework was installed on the Jetson TK1 for deploying in real-time applications. We are using the Jetson TK1 because the chip (TK1) has a CUDA-based GPU–unlike any other low power embedded system–enabling us to run the computations in it to achieve faster performance. In order to have an effective application, we must follow three steps. The first step involves the design and training of a model for deployment. The second is writing the application to feed frames from the windows USB webcam that we were using. Once achieved, we proceed to the third step which is benchmarking the frames per second (fps) in which the application is able to perform, and the latency of the process–the amount of time required for processing a single frame.

4.1 Design and training

For this implementation, we tested the performance of a pre-trained network. When training a model based on the architecture described by Girshick [6] we must pre-compute the object proposals on the dataset. The most common method of doing this is to use Selective Search [8], which is based on MATLAB. We can use Fast R-CNN [7] to train CNNs in the PASCAL VOC [3] dataset. The model that we tested was the CaffeNet, which is a variation of the AlexNet model presented by Kryzhevsky [5], fine-tuned on the PASCAL dataset.

4.2 Real-time Frame Capture

In order to do this we wrote a simple python application that feeds the frames to the CaffeNet model. After the model runs the convolutions, labels the object, and localizes it, it outputs the resulting frame. This model has the capability to label and localize within the frame, any object in the 20 categories of the PASCAL VOC Challenge [3], including people and chairs. The code is written to detect persons only, for testing purposes.

4.3 Results

The performance of the CaffeNet model can be

benchmarked by analyzing the amount of frames-per-second (fps) displayed in the output window and the latency, or the amount required to process a single frame. This model has a performance of <1 frame/second, roughly speaking. For the latency, it takes an average of 3.5 seconds to 4 seconds to process every single frame and display it in the output window.

V. Discussion and Conclusion

The continuous work by various computer scientists and mathematicians on Deep Neural Networks opens many opportunities for implementing these new networks on real-time applications for object detection. These new designs are efficient and more accurate at detecting objects, which means that they are able to do it faster. Our objective is to implement these deep learning networks on low-power systems, like the Jetson TK1, and use them in real-time applications for various applications, especially object detection. The results will show that it is possible to achieve near real-time performance and accurate object detection using deep learning networks. With the continuous development of new networks and models, the performance will increase, and it will open the opportunity to develop new applications.

This project is in a work-in-progress state, and as we keep moving forward, new and more efficient methods are being developed. The current performance is pretty impressive for a system like the Jetson TK1. Most of these deep learning models are used on high-end computers with state-of-the-art GPUs to analyze images in a fraction of a second. Nonetheless, the current performance has room for many improvements in order to achieve better performance. The system is currently running a pre-trained CaffeNet model, fine-tuned on the PASCAL VOC dataset. Although it is relatively small, the CaffeNet model was trained on the 1000-category of the ImageNet Challenge. It has a lot of data, parameters and labels which result in a bigger network that contains many categories that we will not be using. CaffeNet is based on AlexNet [5], which runs CNNs, not R-CNNs, being R-CNNs better for object detection.

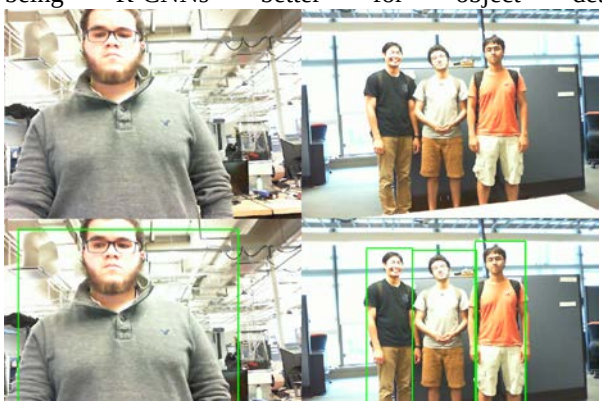

Figure 2: Person detection: On the top row of the image appears the input frame of the USB Camera. On the lower row appears the frame after being processed by the convolutional network. This detection is for “person”. There can be multiple people in the frame detected at once.

Our future goals include designing a network based on the Fast R-CNN detector, and training that network on the INRIA [4] Person dataset to create a person/civilian detector. Also, we can create our own dataset using the ImageNet dataset with their annotations. We can pre-compute object proposals using Selective Search or other methods available like Learning Object Proposals (LPO) [9], and use this to train our own Fast R-CNN detector on a custom set of categories that fit our specific requirements.

Acknowledgment

The authors would like to acknowledge the support of the National Science Foundation, through NSF REU grant no. 1062672.

References

[1] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “ Caffe: Convolutional architecture for fast feature embedding.”, arXiv:1408.5093, 2014.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database”, IEEE Computer Vision and Pattern Recognition (CVPR), 2009.

[3] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A., “The Pascal Visual Object Classes (VOC) Challenge”, International journal of computer vision, vol. 88, no 2, p. 303-338, 2010.

[4] N. Dalal, and B. Triggs, "INRIA person dataset, 2011-04-10]”, http://pascal, inrialpes, fr/data/'human, 2005.

[5] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks”, NIPS, 2012.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR, 2014.

[7] R. Girshick. “Fast R-CNN,” arXiv e-prints, vol. arXiv:1504.08083v1 [cs.CV], 2015.

[8] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A.W. Smeulders, “Selective search for object recognition”, IJCV, 2013.

[9] P. Krähenbühl, and V. Koltun, “Learning to Propose Objects”, CVPR, 2015.

[10] R. J. López-Sastre, T. Tuytelaars, and S. Savarese, "Deformable part models revisited: A performance evaluation for object category pose estimatio", Computer Vision Workshops (ICCV Workshops), IEEE International Conference, 2011.

[11] K. Simonyan, and A. Zisserman, “Very deep convolutional networks for large-scale image recognition”, arXiv preprint arXiv:1409.1556, 2014.

[12] R. Girshick, F. Iandola, T. Darrell, and J. Malik, “Deformable part models are convolutional neural networks”, arXiv preprint arXiv:1409.5403, 2014.

[13] H. O. Song, R. Girshick, S. Zickler, C. Geyer, P. Felzenszwalb, and T. Darrell, “Generalized Sparselet Models for Real-Time Multiclass Object Recognition”, Pattern Analysis and Machine Intelligence, IEEE Transactions on,37(5), 1001-1012, 2015.

[14] N. Dalal, and B. Triggs, “Histograms of oriented gradients for human detection”, IEEE Computer Vision and Pattern Recognition (CVPR), 2005.