SQUEEZING DEEP LEARNING INTO MOBILE DEVICES

Ramya Venkatesh, Ramesh Ragala, R.Jagadeesh Kannan

***Abstract*_ Advancement in technology has led to generation of data in huge amount every second. Most of the data is generated by mobile devices. Various sensors in mobile devices continuously keep generating the data. This data can be used for analyzing various activities of the user. The real time applications require analysis of the continuously data generated on mobile devices. Deep learning helps to extract useful information from the huge data being generated. This paper discusses about various deep learning models available for object detection. The idea of measuring relative distances of objects in the image along with object identification is proposed. The state-of-art models only detect the objects in the images, whereas to know the relative position of these objects along with its identification becomes useful and helps to build better vision applications. The paper also describes various methods to run this deep learning model on mobile devices.**

Keywords_ Deep Neural Network (DNN), Convolution Neural Network(CNN), Deep Learning, Object Detection, MaskRCNN.

I. INTRODUCTION

Deep learning is a subset of machine learning algorithms that attempts to model high-level abstractions in data by using architectures composed of multiple nonlinear transformations. Deep learning automatically finds out the important features that are important for classification and deep learning does end to end processing. Deep learning model consists of multiple layers which learn the data hierarchically. This is also called as Deep neural network. In contrast to traditional neural network, deep neural network has multiple layers which process the input data at varying level of abstraction.

Convolution Neural network consists one or more convolution layers. This may be optionally followed by fully connected layers. CNN is mainly designed to take advantage of 2D input like image or speech. Convolution neural networks can be used for image processing. An image is divided into multiple overlapping grids which is fed into convolution layers. Sampling and Max pooling of the data is done. The output is then given to another neural network called fully connected layer which finally tells whether an image is match or not.

Most of the real world applications like object identification, speech recognition, natural language processing, video analysis etc. uses deep learning to provide accurate results. The conventional approach for such applications is to offload the computation and storage on cloud server, only input and output is processed at the mobile device. This method is useful as deep learning is computationally expensive and it more space to store the data. This method has its own disadvantages when it comes to highly interactive real time and responsive applications like if an application has to identify the object in front of the user, it has to process the image and produce the output immediately. Similarly, an application has to react to the voice instructions of the users in less time. Such applications have to process large amount of input data and produce output immediately. In traditional approach as the data is stored and processed on server side processing gets slower. As the data received is sent to the server security of the data becomes an issue. All these applications perform computation on cloud and provide the result to the end user, this demands for continuous network connectivity, without which the processing fails.

The drawbacks of the traditional approach mentioned above makes it necessary to find the alternate solution which overcomes these drawbacks. Running the deep learning models on mobile or embedded devices could be a potential solution for the above problem. The main challenge to run deep learning models on mobile devices is that, deep learning models are computational, memory and resource expensive, which makes it difficult to run these models on resource and memory constrained devices. This paper performs the study on various techniques available to run the deep learning model on mobile devices.

This paper conducts the study on object detection model. Mask R-CNN[ref] network is used to segment the instances of objects in the image. The object detection model is extended to get the relative positioning of objects with respect to the user. Model compression and optimization techniques can be used to compress this model so as to run on Mobile CPU systems. In this paper study of various object detection models is done and new idea of obtaining relative distance of objects in the image is proposed.

II. RELATED WORK

Mohammad Abu Alsheikh et al. [2] in their research work say about mobile big data. Mobile big data is the huge amount of data generated by mobile devices. The data generated by mobile devices are mostly unstructured or semi structured data. Deep learning models can be used to analyze this data to obtain useful insights about the data. Volume, variety, volatility of the data has to be handled in order to obtain accurate results. According to Mohammad Abu Alsheikh et al. [2] the learning of deep models can be

Revised Manuscript Received on March 10, 2019.

RAMYA VENKATESH, M.Tech SCSE, VIT University, Chennai Campus (Email: ramya.2017a@vitstudent.ac.com)

RAMESH RAGALA, AsstProf SCSE at VIT University, Chennai Campus (Email: ramesh.ragala@vit.ac.in)

R.JAGADEESH KANNAN, Prof SCSE at VIT University, Chennai Campus (Email: jagadeeshkannan.r@vit.ac.in)

parallelized to reduce the time taken for making decision. The data is divided into several parts and stored in different machines. Then mapreduce tasks are executed on each chunk of data in different machines. By extending the streamlining capacity with higher bandwidth, the huge data produced every second can be handled. The intermediate data of mapreduce task is volatile. To handle this the training of deep learning model can be speeded up [2].

The training of the models can be done on GPU supported systems and only inference models can be run on mobile devices. The main bottlenecks to run the inference model on mobile devices is memory and computational energy [3]. To run the inference model on mobile devices, each layer of the model can be compressed at runtime. Nicholas D. Lane et al [3] proposes a method where an estimator can be implemented which regulated the level of dimensionality reduction. Based on the speed and accuracy required the optimized decomposition technique can be used to divide the deep model layers into unit block, and later execute these layers on processors either locally or remotely. SVD based compression can be used for dynamically scaling the resources according to availability. The redundancy can be removed by using SVD based layer compression. Finally, the models can be merged and the output can be inferred. This type of layer compression is commonly used for training models which do not require retraining.

One of the useful applications of deep learning is object identification. Most of the real time applications uses object identification or image recognition to provide advanced user friendly solutions. Akhil Mathurz et al. [4] in their work Deep eye propose idea of a small wearable size camera which can run multiple cloud-scale deep learning models on embedded devices. The camera captures real time images and perform analysis locally without offloading the computation of cloud. Various optimization techniques can be used to achieve the above mentioned. The key feature of CNN model is that the execution of layers follows strict order. Fully connected layer is applied on output of convolution layer. Akhil Mathurz et al. [4] proposes a solution which aims at interleaving execution of these two layers. There are two threads that are spawned during execution. One being convolution-execution thread and other data-load thread. Once the convolution – execution thread completes the output of the final convolution layer of each model is passed to data-load thread which loads FC parameters on to the memory and proceeds to obtain final convolution results. The k largest FC layer is loaded into the memory, if there is no runtime memory available then k-1th largest layer is loaded. SVD based model compression technique can be used which reduces the loading of memory heavy FC layers into the memory. Since a video data is continuously collected there are chances of multiple images of same thing being taken many times. The deep learning model need not be executed for all the similar images. Akhil Mathurz et al. [4] proposes a method to achieve this. The perceptual hash vector of an image captures is compared with the hash values of previous five images and if the value is greater than some threshold then inference pipeline is terminated and the result of nearing hash value is returned. This technique is used to avoid unnecessary loading and execution of models when the continuous capturing of image is same.

Huynh Nguyen Loc et al. [5] proposes a GPU-based framework for running deep convolutional neural networks on mobile devices. This framework being called DeepSense is built on an Open CL framework. It is mainly used for object identification, image recognition and face recognition. The draw back with GPU processing is that it processes both the blocks of conditional code and this is called as branch divergence, due to this computation time becomes more. So the proposed framework allocates new block of memory. The input data is copied to that location before executing on GPU kernel this may avoid branch divergence but as memory copying operations become more the time for execution will also increase. Explicit padding of input data can be done to avoid branch divergence. The experimental results [5] shows that the execution time is reduces by following explicit padding. The execution time can be reduced by floating point conversion [5]. The floating point values can be converted to half its size that is 32 bit to 16 bit. Experiment shows that by doing so the accuracy loss is less than 5% and the inference time is significantly reduced [5].

Kaiming He[7] et al. a research group from Facebook Artificial Intelligence Research team developed a network called Mask R-CNN which is extension of Faster R-CNN model. In their work they add an extra branch to the object prediction of Faster-RCNN. The network Mask R-CNN provides the masks for each object instance in the image. The mask is returned in the form of matrix, every region of interest RoI [7] returns its own mask. This helps in semantic segmentation of objects in images.

Region Based CNN also known as R-CNN [8] attends the candidate object regions [9,10] and Convolution networks is independently evaluated for every RoI[11,12]. Faster speed and accuracy is obtained by extending R-CNN to allow RoI on feature maps using RoIPool[13,14]. Faster R-CNN is a flexible framework which learns the attention mechanism by Region Proposal Network[15]. Faster R-CNN is the leading framework in present benchmarks.

III. METHODOLOGY

Mask R-CNN is used for object detection and image segmentation. The model is built using Mask R-CNN. The present working model provides the bounding box for each object instance in the image. The bounding box contains the objects in the images along with a mask. Initially the model is built using Mask R-CNN network. The first step is Region Proposal Network that displays anchor box refinements. Then in next step the solid bounding boxes are obtained around each object in the image. In next step the image is segmented using Mask R-CNN and the masks of each object instance is obtained. Every mask is differentiated with the color of the mask.

The model is trained with COCO 35K dataset. The COCO dataset contains class names of every object along with the class id. The class names of COCO dataset are discontinuous and random. To overcome this problem, the model consists of its own sequence number for each class of COCO dataset. There are 81 classes of objects in the entire collection of image dataset. Model can be trained with other dataset as well by changing the configuration file settings. In

the implementation model weights are directly obtained from pre trained model to reduce the training time.

This model detects the objects and displays the score. For ex: Person: 0.993 which means the object detected is 0.993 that is 99.3% a Person. Along with object detection segmentation of object is also done and mask to that segment is obtained. This segmented object instances can be used for human poses recognition [7] and many other similar applications. The model is extended to provide an additional functionality of predicting the relative position of the object in the image. For this purpose, we consider the co-ordinates of the bounding box obtained from object detection model. According to the image co-ordinate system the mid-point of the image bottom line is considered, from the image height, the image reference point is taken to be bottom most line of the image. The camera is supposed to be in front when the image was taken hence it becomes sensible to consider the bottom most boundary of the image to be the reference plane where the user is standing.

Figure 1. Vertical Distance Measurement from Plane of Reference

Figure 1. Shows the image in which multiple objects are detected and they are displayed in the bounding box along with their scores. The image also shows the segmentation mask being obtained. The pixel distance of the object is considered to be the vertical distance of object from the reference plane as shown in fig 1. Mid-point of the bottom line of image is considered and its Euclidian distance from the bottom most line of image is obtained. Based on the distances, it can be observed that the relative positioning of the objects can be obtained. This method helps to tell which object is nearer and which object is farthest. Thus it helps to know the relative positioning of each object with respect to the user.

Model Compression technique is used to compress the image, every layer in the model is compressed so that it can be run on the mobile devices which are resource constrained. With less memory and computational power the CPU systems must be able to run this model and predict the outputs within a time limit.

IV. WORKFLOW

Figure 2. shows the workflow of the project being carried out. The initial step to be performed is to get the appropriate training data. Preprocess the data if necessary, so as to obtain the information required for the study. Once the dataset is ready next step is to build the desired model using Mask R-CNN network. This model has to be trained with the preprocessed data. Once training is done, some test images data can be given as input to the model. The model has to recognize all possible objects present in the image, draw the bounding box around the image providing class name information and scores of each object. Since as extra layer of segmentation is added, masks around the objects are obtained which can be further used for some applications which needs only the object instances in the image.

Once the objects are identified, the model is extended to obtain relative distances of the objects in the image. The Euclidian distance is calculated appropriate positioning information of the object is displayed. The obtained model is of a bigger size and may not run efficiently on mobile devices. Hence quantization can be done on the model and model compression can be performed to reduce the size and complexity of model so as to run on the resource constrained devices like mobile.

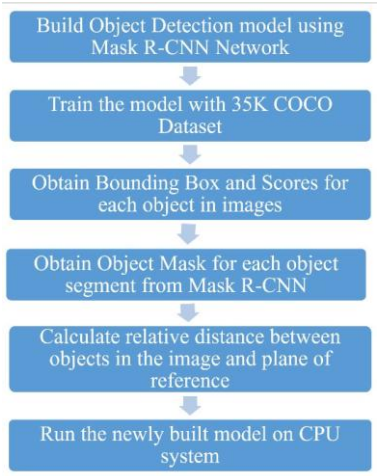

Figure 2. Workflow

V. RESULTS

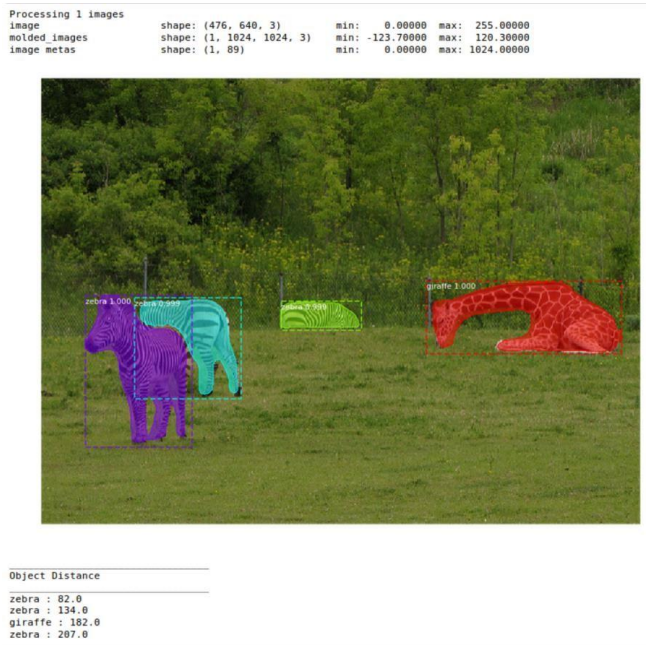

Figure 3: Output of image1

Figure 4: Output of image2

Figure 3 and Figure 4 shows the output of the model. Figure 3 shows zebras and giraffe in a forest. Below the image the results are displayed. The result shows the object names and the relative distance of the objects from the plane of reference. The order of the object is sorted based on their relative distances. In Figure 3. Zebra is the near most object hence its distance obtained is the least which is 82. The farthest one is another zebra with distance 207. Similarly, Fig 4. Shows an outdoor image of some group of people standing. In this image two persons are standing at the same distance hence their relative distance obtained is same.

This shows the object detection and object segmentation along with the scores and distances. The time for prediction is on an average 1.3 seconds. This is tested on the CPU system with 32 GB RAM. Further developments would involve compressing this model to squeeze it in mobile devices and to check for performance on mobile devices.

VI. Future Enhancement

The state-of-art object detection models helps to identify the objects in the images more accurately, presently wellknown object detection model for resource constrained devices is MobileNet which provides upto 86% accuracy. The proposed model also performs with around 79% accuracy, while having a comparatively larger size than mobile net model. The future development can include, model compression techniques to compress the model and run it on mobile devices.

The image has a inherent problem of occlusion that is overlapping of objects. Occlusion causes problem when finding the relative distances of the object. Methods have to be discovered to solve the occlusion problem. As of this paper Euclidian distance is considered for the relative measurement, but this may not hold good under all the circumstances and all the image conditions. Hence a better depth recognition algorithm can be used to identify the depth of an object in a 2D image, which is again another big area of research

The model can be extended for 3D images, it can also be run on the video stream. The 3D images and video stream can provide only object detection and segmentation whereas the distance measurement using the depth information has to incorporated in the extended model. Video stream object detection would be helpful for vision applications instead of 2D object detection

VII. CONCLUSION

The paper discusses about the advantages and disadvantages of state-of-art applications which uses cloud based systems. It describes how squeezing deep learning into mobile devices can overcome these issues. Paper describes the various methods available for compressing deep learning models, various optimization techniques are discussed in Section 2. A new model is built over the existing object detection model, which extends the existing model to measure the relative distance of the object in the image. The newly built model performs with 79% accuracy on CPU system. Section 6, describes the various drawbacks and possible future enhancements of the proposed model.

REFERENCES

1. Nicholas D. Lane, University College London and Nokia Bell Labs Sourav Bhattacharya and Akhil Mathur, Nokia Bell Labs Petko Georgiev “Squeezing Deep Learning into Mobile and Embedded Devices” Published by the IEEE CS n 1536-1268/17 2017 IEEE.
2. Mohammad Abu Alsheikh, Dusit Niyato, Shaowei Lin, Hwee-Pink Tan, and Zhu Han. “Mobile Big Data Analytics Using Deep Learning and Apache Spark”. 0890-8044/16/ 2016 IEEE.
3. Nicholas D. Lane, Sourav Bhattacharya, Petko Georgiev, Claudio Forlivesi, Lei Jiao, Lorena Qendro, and Fahim Kawsar. “DeepX: A Software Accelerator for Low-Power Deep Learning Inference on Mobile Devices”. 978-1-5090-0802-5/16/ 2016 IEEE.
4. Akhil Mathurz, Nicholas D. Lanezy, Sourav Bhattacharyaz,Aidan Boranz, Claudio Forlivesiz, Fahim Kawsarz. “DeepEye:
5. Resource Efficient Local Execution of Multiple Deep Vision Models Using Wearable Commodity Hardware,” to appear in Proc. 15th Int’l Conf. Mobile Systems, Applications, and Services (MobiSys), 2017.
6. Huynh Nguyen Loc, Balan, Rajesh Krishna, and LEE, Youngki. “DeepSense: A GPU-based deep convolutional neural network framework on commodity mobile devices.” (2016).WearSys'16: Proceedings of the 2016.
7. Chien-Hung Chen, Che-Rung Lee(B), and Walter Chen-Hua Lu. “A Mobile Cloud Framework for Deep Learning and Its Application to Smart Car Camera”. Springer International Publishing AG 2016 C.-H. Hsu et al. (Eds.): IOV 2016, LNCS 10036, pp. 14–25, 2016. DOI: 10.1007/978-3-319-51969-2 2.
8. He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn." In Computer Vision (ICCV), 2017 IEEE International Conference on, pp. 2980-2988. IEEE, 2017.
9. R. Girshick, J. Donahue, T. Darrell, and J. Malik. “Rich feature hierarchies for accurate object detection and semanticsegmentation”. In CVPR, 2014.
10. J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. “Selective search for object recognition”. IJCV,2013.

11. J. Hosang, R. Benenson, P. Doll´ar, and B. Schiele. “What makes for effective detection proposals?” PAMI, 2015.
12. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. “Back propagation applied to handwritten zip code recognition”. Neural computation, 1989.
13. A. Krizhevsky, I. Sutskever, and G. Hinton. “ImageNet classification with deep convolutional neural networks”. In NIPS, 2012.
14. K. He, X. Zhang, S. Ren, and J. Sun. “Spatial pyramid pooling in deep convolutional networks for visual recognition”. In ECCV. 2014
15. R. Girshick. “Fast R-CNN”. In ICCV, 2015
16. S. Ren, K. He, R. Girshick, and J. Sun. “Faster R-CNN: Towards real-time object detection with region proposal networks”. In NIPS,2015