Face Recognition and Machine Learning at the Edge

Face Recognition and Machine Learning at the Edge

Joanne Ling Sin Yee[1], Usman Ullah Sheikh[2], Musa Mohd Mokji[3] and Syed Abd Rahman[4]
[1, 2] Intel (M) Sdn Bhd, Pulau Pinang, MALAYSIA.
(E-mail: joanne 0330@hotmail.com)

[2, 3, 4] School of Electrical Engineering, Faculty of Engineering, Universiti Teknologi Malaysia, MALAYSIA.
(E-mail: usman@utm.my, musa@fke.utm.my, syed@fke.utm.my)

Abstract. The number of IoT is expected to reach 20 billion by year 2020. This is due to data that log in the sensors or cameras are all send to the cloud for further processing. Cloud computing is not able to support big data analytic anymore due to network bandwidth. Face recognition is chosen as a case study to demonstrate the challenges to shift the application to the edge. The objective of this project is to develop a face recognition system that is suitable to be used at the edge using a deep neural network. Secondly, investigate the performance in terms of model size, speed and inference time after different bit-width fixed point quantization on the weights of the network. Lastly, deploy the model to Raspberry Pi 3 and test the performance. The chosen data set is AT&T. MATLAB is used to train the network in laptop with i5-7300 CPU while OpenCV-python is used to load and test the network in Raspberry Pi3 and laptop. The proposed system is designed by doing transfer learning on SqueezeNet to classify face. Fixed-point quantization is being applied to the weights of the layers to reduce the size of the model. From the experiment result, it is recommended to use 8-bit fixed-point quantization to the weights in all the layers in the model to compress the size of the network up to 2.5 times while maintaining the original accuracy 90%. That is only 1.1x speed up of the model on Raspberry Pi 3 after different bit-width weight quantization.

Keywords: Edge Computing, face recognition, weight quantization

1. Introduction

Nowadays, with the emergence of the IoT, most of the data from sensors and cameras are logged into the cloud for further data analytics. Based on the Cisco Global Cloud Index, the number of IoTs are expected to reach 20 billion in the year 2020 [1]. The main reason of this success is due to the advancement of doing computer task using machine learning with their ability to process large volume of data in several applications such as speech recognition and video classification. The high volume of data basically includes typical deep learning applications such as face recognition, image classification and human tracking that significantly pushes pressure on the state of the art cloud computing paradigm [2]. For example, in order to process an 8-bit uncompressed video frame in High Definition (HD) and Full High Definition (FHD) camera, it will need 553 Mbps and 1.24 Gbps bandwidth respectively for a one minute video. The size is expected to grow exponentially with the invention of 3D and 4k cameras and will account 82% to IP Internet traffic in the year 2020 according to Cisco Global [3]. Continuously sending whole data in the video to the cloud for analytics will cause a high burden on network congestion, high demands on network bandwidth and increased response time. Cloud computing is now facing a big challenge to support prompt analytics on the big volume of data.

In addition, there are some challenges to preserve the security and privacy when using cloud computing because of cloud computing might need to transmit the data in a long distance. The connection might be lost during data processing and will cause it to expose to the risk of privacy leakage. Also, in some applications such as wearable biomedical devices that includes private data and requires extremely lower power, it is more suitable to be processed at the edge because it can avoid the need to process the data to cloud for analytics. Another application which includes privacy protection is smart home as the thing can be connected in home easily and more perfectly to be implemented locally. Hence, there is a need to bring some of the applications to the edge in order to address the security and latency problem. Edge computing is known as the method to enable the computation to be implemented towards the edge of the network. The ”edge” basically refers to the source of the data such as cameras and sensors in the homes or offices. Edge Intelligence in the term introduced to describe the ability using machine learning algorithm at the edge to process the data obtained to overcome the issue such as cybersecurity, personalized or customized issues [4].

Edge computing can offer many benefits in terms of performance, bandwidth and security. Edge intelligence can help to reduce the response time to action. The latency can be reduced up to milliseconds while reducing the network bandwidth which decreases the risk of data bottleneck. Compared to cloud computing, edge intelligence can preserve the privacy and security of data access by encrypting data closer to the core. This can prevent inappropriate information to be transmitted to the whole network and reduce the delay. Also, the cost is much lower compared to cloud computing as the data from sensors are used locally and less data need to be transferred to the cloud, which means lower power consumption in needed.

Edge intelligence is expected to grow exponentially in many applications with the growth of the optimization techniques in designing a deep neural network. In this thesis, a face recognition system is being selected as a case study because it is one of the useful application that is being used widely nowadays in access control, video analytics, surveillance system and social media system. It can present the opportunity and challenges to shift this system to the edge. For example, in a video surveillance system, face detection and face recognition will consume a large amount of computational resources when every frame needs to be processed. Sending whole video to the cloud for analytics will utilise a lot of network bandwidth. Also, for criminal investigation purpose, there is a need to analyze large video in a short time to provide safety of public [5]. The edge challenges such as memory and performance are being addressed for implementing a CNN face recognition system in this work.

2. Related Works

Edge intelligence means the ability to process the acquired data using a machine learning algorithm at the embedded edge device. Edge computing is one of the promising technology in IoT service and it gained popularity in research nowadays due to the potential to address the problems in cloud infrastructure. The performance of centralized cloud computing infrastructure degrades in processing the data from IoT when the data is transferred with limited network performance. There are two main improvements brought by edge computing to the current cloud computing. First, by pre-processing the data before transferring to the central cloud of the server.In cloud computing, the images taken by the camera will be transferred to cloud with pre-processing. The processing of the images and extracting the extra info such as time, location is being implemented in cloud. However, in the edge computing, the pre-processing is implemented on the camera and only transfer the actual number to the remote infrastructure which helps to alleviate the workload in the cloud. The second one is the computing capability in the edge to optimize the resources in the cloud. Deep Learning is one of the emerging technology that is suitable to be used in edge computing due to the ability to transmit the reduced immediate to the cloud after offloading the parts of the learning layers in the edge [6]. There are some optimization of deep learning techniques such as low-rank approximation, pruning, quantization, knowledge distillation are being implemented in CNN model in edge device [7].

There are some related works regarding edge computing to overcome the resource constraint at the edge. George Plastirae et al. presented a case study of object detection of Aerial Vehicle (UAV) using CNN [8]. He proposed the selection of input size, object size and tile processing in the CNN model to compromise with the performance, computational and memory in the edge. For example, processing all tiles before proceeding to the next frame will improve the accuracy but degrades the performance such as the inference time of the detector. The limitation of the research is it does not discuss the co-optimization of an algorithm that can be used to build a resource-efficient system in order to achieve the goals of edge intelligence. Camille et al. found that the performance is affected by the size of the frame as the algorithm complexity increases in the video analytics at the edge for crowd monitoring. Hence, re-scaling the video frame is suggested [9]. However, this paper does not cover the technique to enhance the re-scaling method in order to fasten the deep learning process. Song Han et al. proposed the use of deep compression techniques that include network pruning, quantization and Huffman encoding. By using this technique, the AlexNet weight is reduced by 35 times without compromise the accuracy [10].

Xiangyu Zhang proposed the use of low-rank approximation in order to speed up the VGG model. Although the convolutional speed up by 4%, the accuracy is dropped by 0.8% [11]. Also, it is difficult to implement global parameter compression on the layers using low-rank approximation as the different layers hold different data. Knowledge distillation is one of the model compression techniques that is being used to deploy the model in the edge device such as mobile. It is one of the promising way to obtain small model while retaining the accuracy compared to large model. However, the performance is affected when a large pre-trained model (a.ka. teacher) is used to train small model (a.k.a student). SeyedIman Mirzadeh et al. proposed a Teacher Assistant Knowledge Distillation by adding intermediate model as teacher assistant to fill the gap when there is large difference in teacher and student model size.

Face classification or face recognition is one the application that is being used widely in many potential applications such as biometrics, social media, video surveillance, home or office access security, law enforcement or human-computer interaction activities [12]. Basically, face recognition is divided into five steps which are image acquisition, image pre-processing, face detection, feature extraction and classification.

However, face recognition is one challenging task due to two main reasons. The main challenge is the method that can be used to improve the recognition performance under the unconstrained conditions such as variation of poses, illumination, rotation and scale. The second challenge for the face classification is large data set is needed for face identification. Thus, face classification is more challenging compared to other classification tasks as it requires more resources and complex algorithms.

There are many types of traditional feature extraction techniques that have been implemented in face recognition such as LBP, PCA, LDA and ICA. Most of the techniques implemented in the early ’90s are able to achieve excellent performance. However, the accuracy of the recognition decreased gradually in the unconstrained environments such as variation of pose, illumination, rotation and scale. For example, the PCA technique proposed by Ramandeep Kaur is easier to be implemented with fast computational time, the performance is always affected by the background of image and illumination [13]. Due to face recognition system accuracy is always affected by the unconstrained condition, Venkat R Peddigari et al. proposed the used of pre-processing techniques which include Mirror Image Superposition (MIS), Histogram Equalization (HE) and Gaussian Filtering (GF) in training data in order to overcome those limitations [14]. However, the preprocessing techniques are complex and other issues like aging or occlusion which will affect the face recognition still need to be addressed.

CNN is one of the deep learning methods that achieved impressive results on face recognition in a constrained environments because it is more resilient with intra-personal variations [15]. A.R.Syaffez et al. suggested to fuse the subsampling layer and convolution layer to simplify CNN [16]. This technique is able to recognize a face in 0.01s. However, the accuracy of AT&T is much more accurate than FERNET. Ya Wang et al. proposed fine-tuning VGG pre-trained model which the full-connected layer is fined tuned with a new dataset [17]. The recognition rate is higher after fine-tuning. Umme Aiman et al. proposed to construct a database by adding Poison or Gaussian Noise to the training set [18]. The model is able to achieve 95.21% recognition rate. However, the limitation of the model is the data is human annotated.

Fixed-point quantization is introduced in the recent year in order to alleviate the workload of hardware implementation by converting the data into a fixed-point format. The weights in the CNN are stored in the floating point which contribute a lot of resources in the embedded devices in terms of memory, powers and increase the computational cost. Hence, there are some fixed-point approaches to convert the weights of CNN in fixed-point format such as BinaryNet. The weights were binarized to to -1,1,0 but that is a significant degradation in the accuracy [19]. Bit-width is one of the fixed-point technique quantization that represents the weight in the shorter bitwidth format. The low precision format reduce the memory footprints and allow large model to be fit in a given space when implementing in the hardware. There are two types of fixedpoint quantization which are static fixed-point and dynamic fixed-point. For static fixed-point, the feature maps, kernels and weights have same bit-width. Whereas, the feature maps, kernels and connection weights have different bit-width in the dynamic-fixed point quantization [20]. Post-quantization can be implemented in the CNN, which means the network can still be trained in the floating point format but quantization algorithm is applied on the network after training [21].

3. Methodology

The main objective of the project is to develop a face recognition system that is suitable to be used at the edge, which means the model size needs to be small while maintaining the performance in terms of speed and accuracy. The adopted methodology is inspired by the methodology proposed by Song Han et al. which compressed the model of AlexNet and VGGNet using quantization method without compromising much on accuracy in the architecture using deep compression methodology such as weight quantization. The same methodology can be applied to compress the model of Squeeze Net on data set AT&T in this system with some modifications. Transfer learning is used to develop the system based on the methodology proposed by YaWang in this face recognition system to improve the accuracy. Also, to fine-tune a network is more efficient than constructing a network from scratch.

SqueezeNet is chosen as the pre-trained model for the transfer learning system because it has 50x fewer parameters than AlexNet but it can achieve the same accuracy in the ImageNet. The smaller CNN model is easier to be deployed at edge devices such as Raspberry Pi 3.

The design model is divided into the training and testing part. First of all, data acquisition is needed. Pre-processing the input image in order to be fed as the input for training and testing. For the training part, instead of training from scratch and consuming a lot of time, transfer learning is applied. The pre-tained SqueezeNet acts as the base network and the final output layer is replaced with the layer adopted with new data set to predict the classes. Due to the training set and validation set are distributed randomly, there is a slight difference in the validation accuracy. The training process is executed a few times to get the best model with the best accuracy to act as the base model.

After that, different bit-width fixed-point quantization is applied to the weights in different layers of the network model to investigate the performance in terms of model size and accuracy.

The model is saved after quantization and converted to ONNX format before exporting to

Raspberry Pi 3. Next, install Open-CV library and use Python to implement the ONNX model in Raspberry Pi 3.

For the training part, it is implemented using an Intel Core i5-7300CPU. For the testing part, in is done on both Raspberry Pi 3 and Intel Core i5-7300CPU to investigate the difference in performance. Raspberry Pi 3 is a single Board Computer (SBC) that supports high level programming language such as Python to run in Linux operating system. It has peripherals such as CPU, memory to execute the instruction without the need of additional hardware.

4. RESULTS AND DISCUSSION

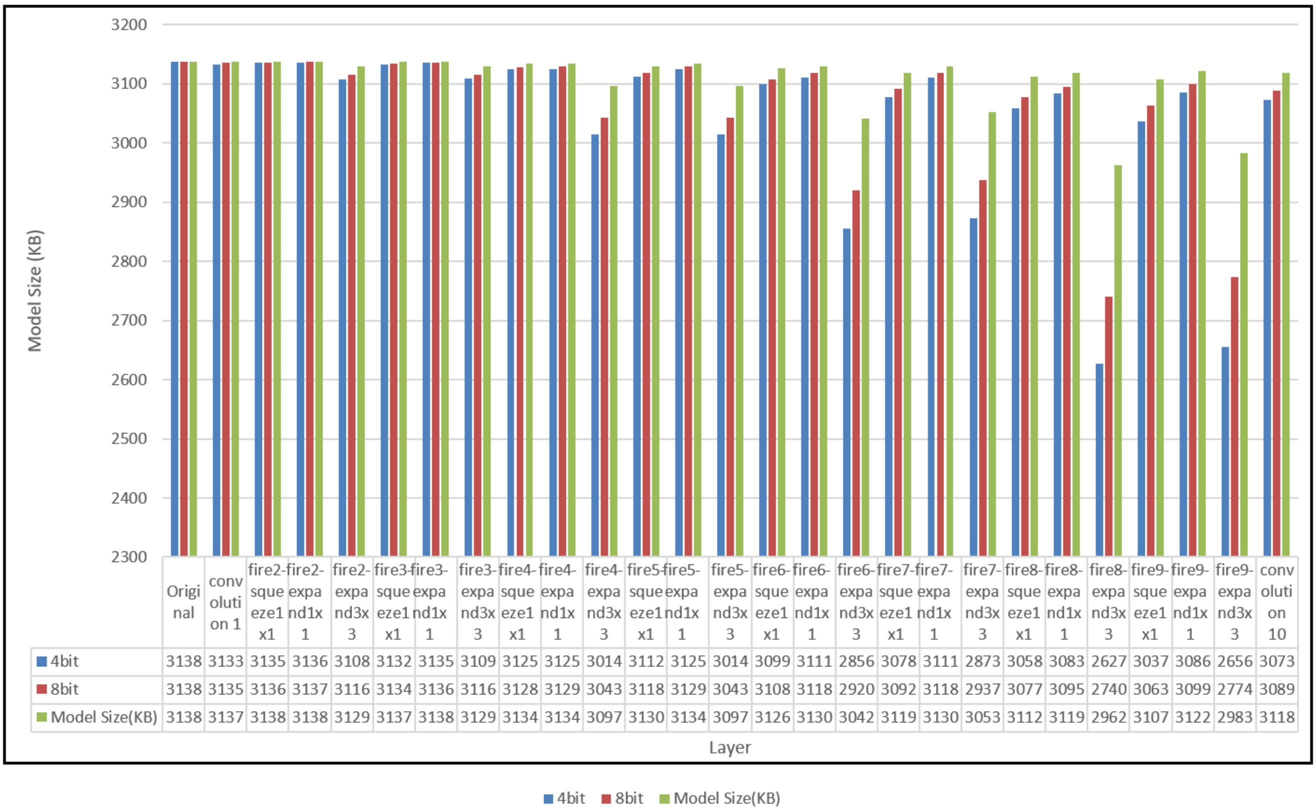
Figure 1 compares the difference in model size after different bit-width quantization are applied on the weights in a single layer. Model size reduced most after apply 4-bit fixed point quantization in all of the layers compared with 8-bit and 16bit because the weights are stored in lower precision format which means it consumes less memory storage. For 4-bit fixed-point quantization, the accuracy drops to 72.5% and 85% from the original 90% after quantization on the weights in the layer 'fire8-expand3x3' and 'fire9expand3x3'. The accuracy drops the to the lowest which is 42.5% after 4-bit fixed point quantization is being applied on the weights in the layer 'fire6-expand3x3'. This might due to the large value changes in total weight after quantization. Although the model size is compressed the most after applying weight quantization in the layer 'fire8-expand3x3', but there is 17.5% accuracy drop. Hence, it is recommended to apply 4-bit quantization on weights in layer 'fire-9-expand3x3' when doing 4-bit fixed point quantization on the weights in a single layer of the network as it causes a slight drop in accuracy while reducing the model size. For 8-bit and 16-bit fixed-point quantization, there is no change even when weight quantization is applied on any of the layers. Hence, it is recommended to apply 8-bit or 16-bit fixed point quantization on the weights in layer 'fire-8-expand3x3' when implementing weight quantization in a single layer as they can help to reduce model size efficiency.

5. Conclusion

This project presented contributions in the face recognition system using deep neural network and edge intelligence. The main objective of the project is to develop a face recognition system using a deep neural network that is suitable to be used at the edge. This is achieved by reducing the size of the network by applying different bit-width fixed point quantization on the weights in the convolution layer of the network without compromising the accuracy. From the experimental results, the size reduced the most when 4-bit fixed point quantization is applied to the weights of any layer compared to 8-bit and 16-bit. However, that is a significant drop in accuracy when 4-bit quantization is applied on the weights in all the layers. In terms of inference time, it is only around 0.03s faster even the network size has been compressed to 4.5x times smaller on Raspberry Pi 3. Hence, it can be summarised that weight quantization is an effective technique to compress the CNN network, but speed up brought from quantization method seem like not as high as expected.

Acknowledgment

The authors would like to thank Universiti Teknologi Malaysia and Intel Microelectronic (M) Sdn Bhd for the support.



	Original	convolution 1	fire2-squeeze1x1	fire2-expand1x1	fire2-expand3x3	fire3-squeeze1x1	fire3-expand1x1	fire3-expand3x3	fire4-squeeze1x1	fire4-expand1x1	fire4-expand3x3	fire5-squeeze1x1	fire5-expand1x1	fire5-expand3x3	fire6-squeeze1x1	fire6-expand1x1	fire6-expand3x3	fire7-squeeze1x1	fire7-expand1x1	fire7-expand3x3	fire8-squeeze1x1	fire8-expand1x1	fire8-expand3x3	fire9-squeeze1x1	fire9-expand1x1	fire9-expand3x3	convolution 10
4bit	3138	3133	3135	3136	3108	3132	3135	3109	3125	3125	3014	3112	3125	3014	3099	3111	2856	3078	3111	2873	3058	3083	2627	3037	3086	2656	3073
8bit	3138	3135	3136	3137	3116	3134	3136	3116	3128	3129	3043	3118	3129	3043	3108	3118	2920	3092	3118	2937	3077	3095	2740	3063	3099	2774	3089
Model Size(KB)	3138	3137	3138	3138	3129	3137	3138	3129	3134	3134	3097	3130	3134	3097	3126	3130	3042	3119	3130	3053	3112	3119	2962	3107	3122	2983	3118

Figure 1. Comparison of the Model Size after different bit-width Quantization

References

[1] Shi, W., Cao, J., Zhang, Q., Li, Y. and Xu, L. Edge computing: Vision and challenges. IEEE Internet of Things Journal, 2016. 3(5): 637646

[2] Huang, Y., Ma, X., Fan, X., Liu, J. and Gong, W. When deep learning meets edge computing. 2017 IEEE 25th International Conference on Network Protocols (ICNP). IEEE. 2017. 12.

[3] Ali, M., Anjum, A., Yaseen, M. U., Zamani, A. R., Balouek-Thomert, D., Rana,O. and Parashar, M. Edge enhanced deep learning system for largescale video stream analytics. 2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC). IEEE. 2018. 110.

[4] Li, E., Zhou, Z. and Chen, X. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy. Proceedings of the 2018 Workshop on Mobile Edge Communications. ACM. 2018.

[5] 3136.

[6] Qi, X., Liu, C. and Schuckers, S. CNN based key frame extraction for face in video recognition. 2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA). IEEE. 2018. 18.

[7] Li, H., Ota, K. and Dong, M. Learning IoT in edge: deep learning for the internet of things with edge computing. IEEE Network, 2018. 32(1): 96101.

[8] Cheng, Y., Wang, D., Zhou, P. and Zhang, T. A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282,2017.

[9] Plastiras, G., Terzi, M., Kyrkou, C. and Theocharidcs, T. Edge Intelligence: Challenges and Opportunities of Near-Sensor Machine Learning Applications.2018 IEEE 29th International

Conference on Applicationspecific Systems, Architectures and Processors (ASAP). IEEE. 2018. 17.

[10] Bailas, C., Marsden, M., Zhang, D., OConnor, N. E. and Little, S. Performance of video processing at the edge for crowd-monitoring applications. 2018 IEEE 4th World Forum on Internet of Things (WFIoT). IEEE. 2018. 482487.

[11] Han, S., Mao, H. and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149, 2015.

[12] Zhang, X., Zou, J., He, K. and Sun, J. Accelerating very deep convolutional networks for classification and detection. IEEE transactions on pattern analysis and machine intelligence, 2016. 38(10): 19431955.

[13] Hassaballah, M. and Aly, S. Face recognition: challenges, achievements and future directions. IET Computer Vision, 2015. 9(4): 614626.

[14] Kaur, R. and Himanshi, E. Face recognition using principal component analysis.2015 IEEE international advance computing conference (IACC). IEEE. 2015. 585589.

[15] Peddigari, V. R., Srinivasa, P. and Kumar, R. Enhanced ICA based face recognition using histogram equalization and mirror image superposition. 2015 IEEE International Conference on Consumer Electronics (ICCE). IEEE. 2015.625628.

[16] Schmidhuber, J. Deep learning in neural networks: An overview. Neural networks, 2015. 61: 85117.

[17] Syafeeza, A., Khalil-Hani, M., Liew, S. and Bakhteri, R. Convolutional neural network for face recognition with pose and illumination variation. International Journal of Engineering Technology, 2014. 6(1): 09754024.

[18] Wang, Y., Bao, T., Ding, C. and Zhu, M. Face recognition in real-world surveillance videos with deep learning method. 2017 2nd International Conference on Image, Vision and Computing (ICIVC). IEEE. 2017. 239 243.

[19] Aiman, U. and Vishwakarma, V. P. Face recognition using modified deep learning neural network. 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE. 2017. 15.

[20] Al-Hami, M., Pietron, M., Casas, R. A., Hijazi, S. L. and Kaul, P. Towards a Stable Quantized Convolutional Neural Networks: An Embedded Perspective. ICAART (2). 2018. 573580

[21] Abdelouahab, K., Pelcat, M., Serot, J. and Berry, F. Accelerating CNN inference on FPGAs: A Survey. arXiv preprint arXiv:1806.01683, 2018.

[22] Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. and Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[23] 2018. 27042713.

[24] Intel Core i5-7300HQ Processor.

[25] Pi, R. Raspberry Pi Model B, 2015