Embedded Implementation and Evaluation of Deep Neural Network of Federated Learning

Zhuoyue Zhao, Feiyu Wu, Chao Dong, Yuben Qu

College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, China

Abstract. Compared with traditional distributed machine learning, federated learning (or joint learning) enables multiple computing nodes to cooperate and train a shared machine learning model without transmitting original data. At present, the research work of federated learning mainly focuses on the theoretical method, and the system implementation is less, and only for the text data or simple image such as medical institution information sharing, handwriting font recognition and other simple neural network applications. Aiming at more complex deep neural networks, this project implements a multi-node federated learning system on embedded device, and evaluates its key performance indicators such as training accuracy, delay and loss. The research method mainly uses embedded computer both as client and server, adjusts and groups the Visdrone datasets as training samples, and then trains the model on the client based on YOLOv4 algorithm, realizes the encrypted transmission of information through TCP protocol, and achieves the aggregation update of the model on the server with FedAvg algorithm.

Keywords: Deep Neural Network; Federated Learning; Embedded Implementation; Object Recognition.

1. Introduction

The concept of federated learning [1] comes from federated learning technology pioneered by Google, aiming to solve two major challenges in the development of machine learning technology: One is data security, and the other is data sharing, which effectively guarantees the privacy of users by transferring the data storage and model training stage of machine learning to local users, and only interacting with the central server to update the model [2]. The year 2019 is known as the first year of federated learning. As a new research hot spot in the field of network security, federated learning has attracted a lot of attention and research. Our work also conducts in-depth learning and research on neural networks for object recognition under the background of federated learning, which is more secure and efficient.

Therefore, innovation point of this project have the following: First, compared with the existing combined neural network learning system for mostly simple depth [3], we want to realize the depth of the complex neural network, namely for the application of visual target recognition. Second, different from other similar federated learning studies, which are mainly theoretical, this project is more about implementing a multi-node federated learning system through embedded deployment, and evaluating its training accuracy, delay and other key performance indicators.

2. Related Work

2.1 Neural Network Selection for Visual Object Recognition

As a lightweight neural network structure, YOLOv4[4] is shown in the figure 1 below. Compared with other network models, YOLOv4 designs a powerful and efficient detection model with low hardware requirements. Embedded devices such as Jetson series can easily train this super-fast and accurate model. This model verifies and modifies many deep learning object detection and training techniques of SOTA in recent years, such as CmBN, PAN, SAM etc. to make them more efficient for single GPU training. Therefore, this project plans to improve and optimize the model based on this model and apply it to online federated learning system.

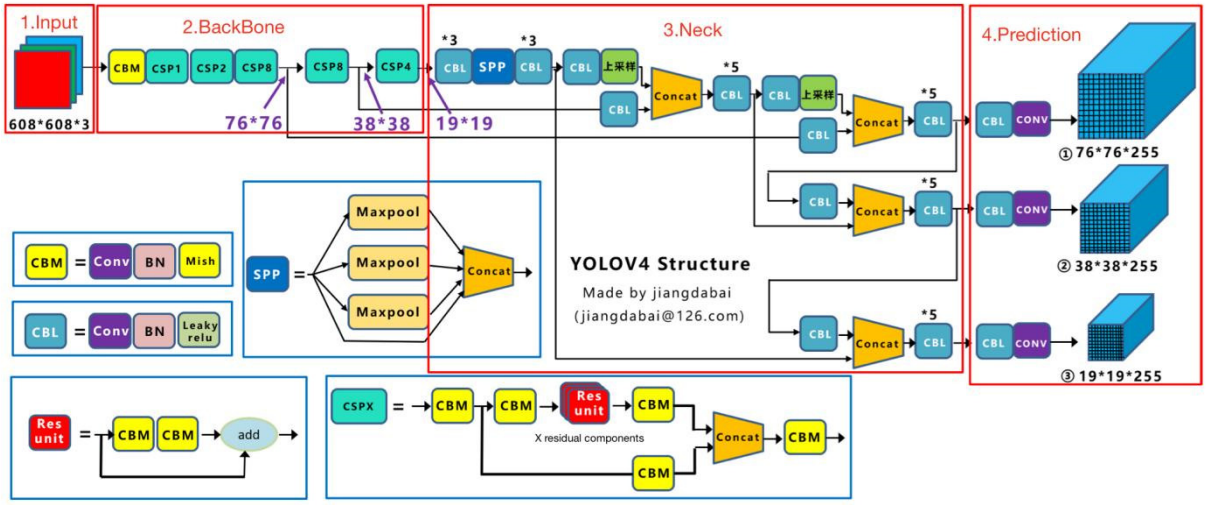

Figure 1. The structure of YOLOv4[4]

2.2 The Hardware System

In terms of hardware selection, low-power embedded artificial intelligence devices such as Nvidia Jetson series are planned to be used as node devices, which are responsible for the reasoning of complex models and the training of lightweight models. For example, Jetson Nano [5] is equipped with quad-core Cortex-A57 CPU and Maxwell architecture GPU along with 4GB LPDDR4 memory and 16GB storage. The graphics card supports high-resolution sensors, can process multiple sensors in parallel, and can run multiple modern neural networks on each sensor stream. It also supports many common AI frameworks, making it ideal for deployment as a node or the more capable equipment such as Jetson TX2[10].

In addition, Ad hoc network communication modules are installed on node devices through cable network ports. The communication module of Ad hoc network includes Beidou positioning, serial port, network port and other peripheral circuits, which are responsible for the communication and data transmission between nodes. It can realize the transmission and acceptance of wireless communication signals and the modulation and demodulation of baseband signals based on the dedicated link of Ad hoc network module. The transmission delay is not more than 100ms, and the access success rate is more than 95%. Moreover, it is equipped with large-capacity FPGA, which can process high-speed modulation and demodulation algorithms and realize broadband communication.

3. Method

3.1 Overall Steps

Figure 2 shows our overall research idea. First of all, we see a server and multiple clients in the middle of the square that are our core part of this federated learning, which is built by Jetson TX2. And then adjust and group the data that was collected in order to form the datasets which are simply put on each client. Through YOLOv4 algorithm, model training is completed. Subsequently, the trained parameter information is encrypted and transmitted to the server through TCP protocol. Finally, the server will aggregate and update the transmitted information through FedAvg[6] algorithm.

In figure 2, only overall idea is shown, and there are more detailed steps of the whole system.

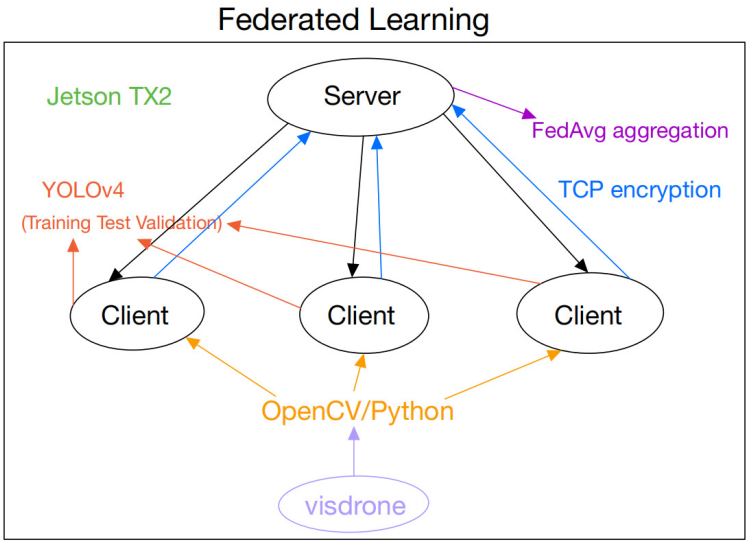

Figure 2. The overall framework of embedded implementation of deep neural network of federated learning system

3.2 Federated Learning Steps

First of all, Specific steps of federated learning are as follows:

As depicted in Figure 3, there are four steps;

Step 1: The client downloads the global model from the server.

Step 2: The client k trains the local data to achieve the local model.

Step 3: The client uploads the local model to update to the central server.

Step 4: The server complete weighted aggregation operation after having received the data from all clients to get the global model [2].

Among other things, $W_{t,k}$ represents the client k of the t round of communications of the local model update. W_{t-1} represents the global model update of the t round communication.

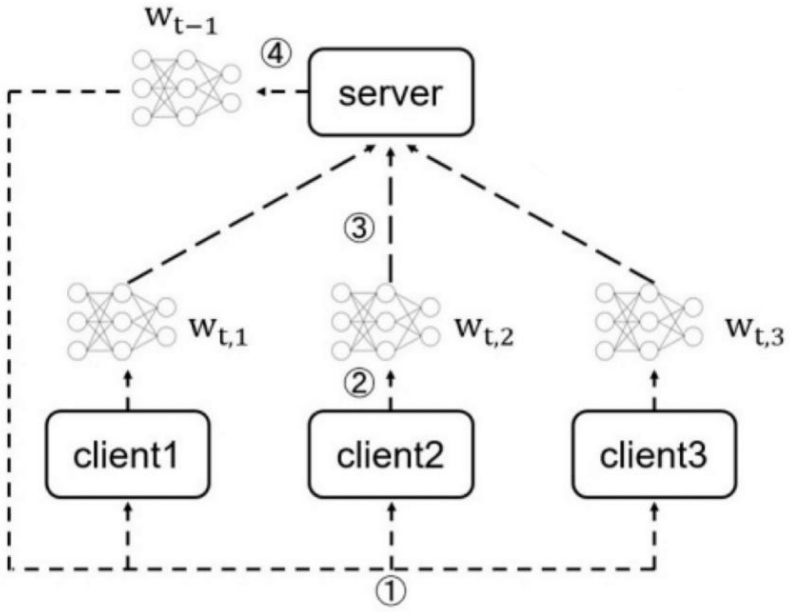

Figure 3. The model of federated learning model [2]

3.3 Model Training for Image Recognition

The following is about the model training of Python and OpenCV for UAV visual image recognition on Jetson TX2[10]:

Step 1: Obtain the initial coordinates and size of the object in the image through the classifier provided by OpenCV for object detection.

Step 2: Select the message of multidimensional array pixel information of targeted images, unifying the size of the images, and after layers of convolution and pooling, the information of arrays is stored into the folder.

Step 3: All the targeted data can be divided into training sets and test sets, and then mess up the order.

Step 4: Build a model of the convolutional neural network, using classified image sets to train and validate the neural network.

3.4 Construction of Neural Networks

The steps of construction of neural networks are as follows;

Step 1: Build TensorFlow environment based on Python language.

Step 2: Import the sample datasets: import a large amount of information of the targeted image for the following deep learning.

Step 3: Conversion and unification of data: unprocessed original image samples often do not conform to the expected shape of TensorFlow, so it is necessary to transform datasets;

Step 4: Divide the sample datasets: the image samples are divided into training sample sets, test sample sets and validation sample sets.

Step 5: Set learning parameters: Set constant parameters for the model, such as learning rate, weight parameters and iteration times.

Step 6: Initialize variables and placeholders: In the process of model optimization, TensorFlow will obtain data information through placeholders, and then adjust data by adjusting variables and weight deviation values. It is necessary to make a reasonable trade-off between data accuracy and training speed.

4. Experimental Results and Analysis

4.1 Organize Data Sets and Set up Federated Learning Systems

Firstly, the training set data is written in YOLOv4 format, and its data information is transmitted to the device with Jetson TX2 as the client. Similar to the text format in Figure 4, each stored image corresponds to an address in YOLOv4 format.

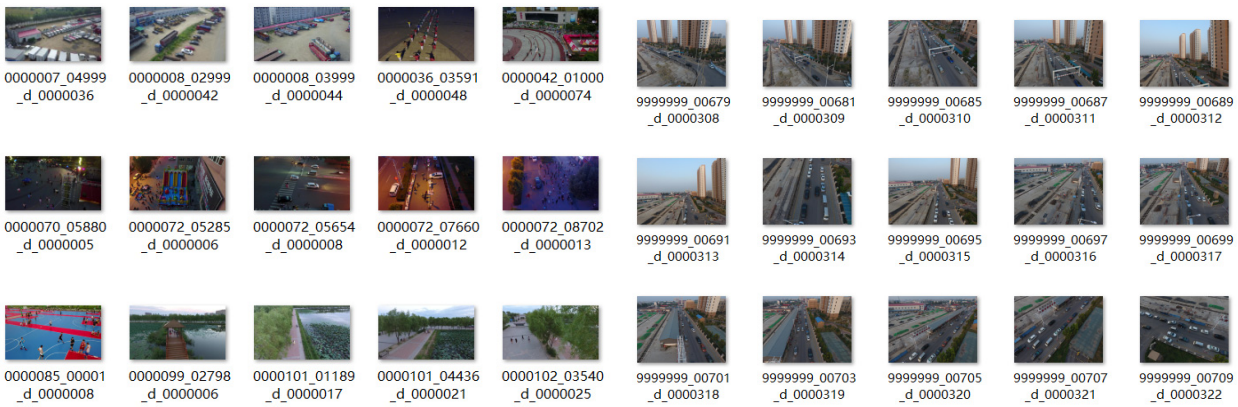

Figure 4. Data set in YOLOv4 format

After sorting out the data set, we set up a federated learning system and wrote codes to realize the realizable program. On this basis, we constantly modified and improved the code, added the timing function and invoked the warm-up learning strategy. In addition, we also added the breakpoints in order to continue training, so that training can continue after an unexpected interruption.

The federated learning system adopts the training model of one server connecting three clients, as shown in Figure 5. Figure 6 is the result of 15 iterations, it is not difficult to find that when the first round of training AUC [7] is close to 0.5, authenticity is the lowest. And when the number of training rounds gradually accumulate, AUC value increases until 13 rounds of iterations are gradually stabilized. And after that, training results have been able to meet the authenticity. So even if the rounds of training increase, there will be no big difference.

At the same time, we can also see that the loss value decreases as the number of iterations increases. The final test results show that the training effect is the best when the number of iterations is the 13th, the AUC is 0.8978, and the loss value is about 1.304.

However, when recording the time-consuming of each training stage, it is found that the GPU utilization rate is relatively low, the loading time is more than 17 seconds, and the GIoU[8] loss value reaches 96.57. After improving the algorithm, we preliminarily improve the GPU utilization rate.

Figure 5. Training terminal information of Federated Learning

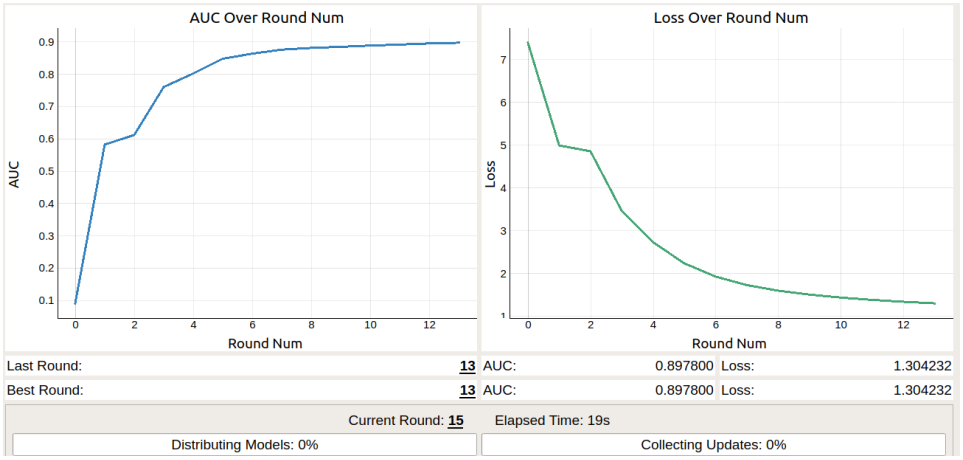

Figure 6. AUC and Loss correspond to the relation of training times respectively

4.2 The Result of Object Recognition

After several rounds of training, we can see the test results of the samples in Figure 7. The recognition content on the left side is relatively simple, so it is obvious to recognize the car, and the corresponding recognition accuracy is also higher, except for some cars with bad angles and relatively close to each other, their accuracy is slightly lower. In addition to the car, a variety of objects such as the van also appeared in the right of the photograph. So, the depth of the neural network algorithm is needed for detecting more complex objects. According to the pictures, a van and pedestrians in the vicinity were successfully detected, but unfortunately pedestrians in the distance were not to recognize, However, the detection accuracy is higher among the identified targets.

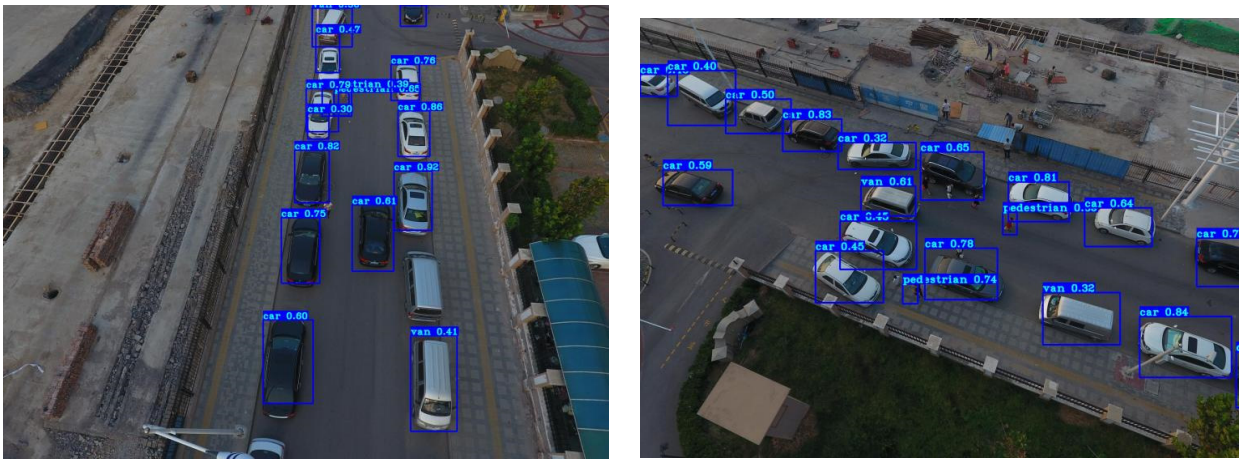

Figure 7. The result of object recognition

4.3 The Result of Loss and mAP for Two Clients

According to the detection results in the sample pictures, the visual interface and multi-thread processing are adopted to feedback the loss value and mAP [9] value. Figure 8 shows the data information when two clients are used. With the increase of training rounds, the loss gradually decreases, and the mAP gradually increases. After 15 training rounds, the best loss value is 181.186813, and the best mAP value is 5.085144.

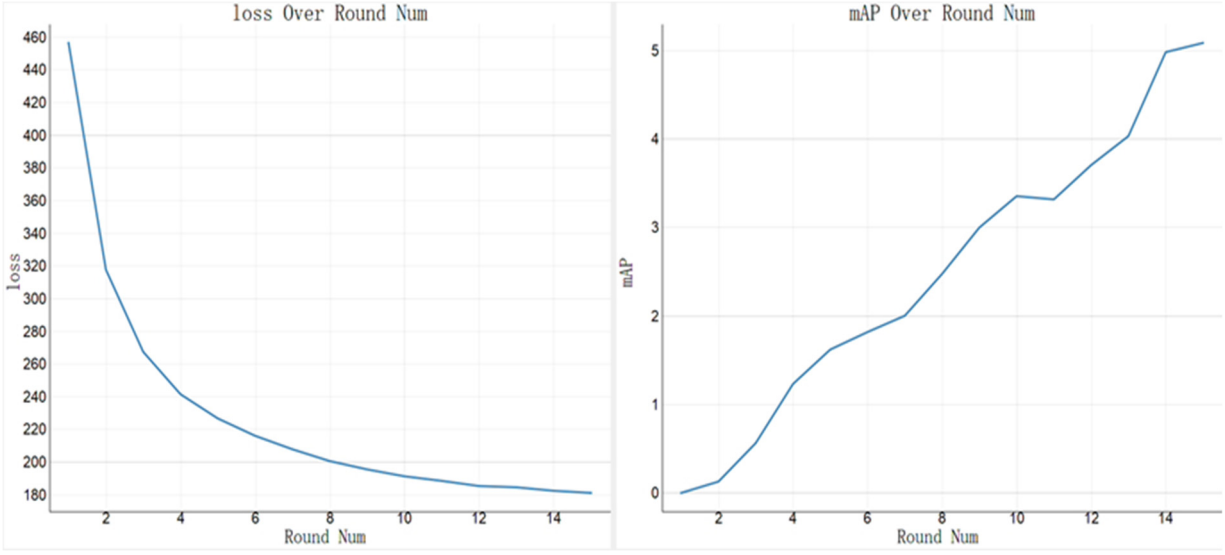

Figure 8. Loss and mAP over Round Num for two clients

4.4 The Comparison between Different Numbers of Nodes

Then the number of clients is increased to test and the differences of the training models are compared. According to Figure 9, it is found that the mAP value of single node 1 is the smallest. With the continuous increase of training rounds, the mAP value of double node 1 is the highest, which can reach about 1.95. The situation of single node 2 and double node 2 is basically the same. For the loss value, the results of the four nodes are almost the same, which all decrease with the increase of the training rounds. Finally, after the tenth training, the loss value drops to about 225.

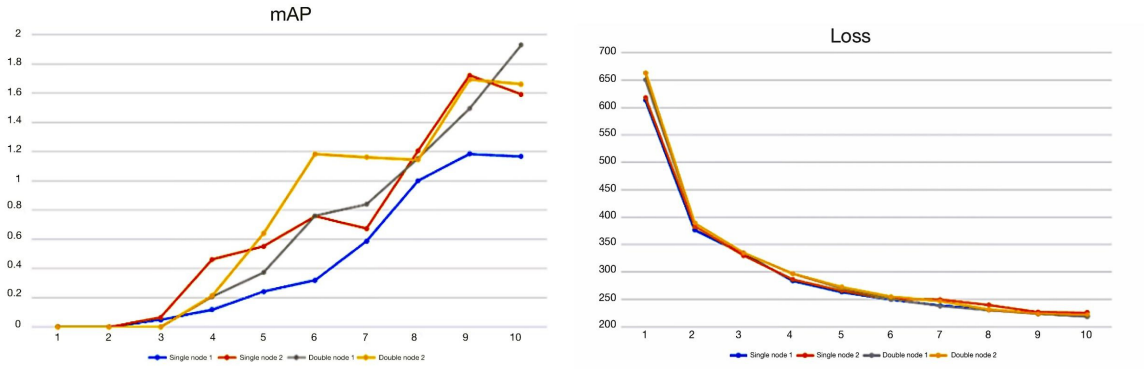

Figure 9. The comparison between different numbers of nodes

5. Conclusion

On balance, the significance of this project is to detect more complex objects. Therefore, the original algorithm was improved by adopting deep neural network CNN and citing YOLOv4 algorithm. At the same time, the process of training data is on the embedded device, because the embedded device is very small compared with the ordinary computer, which is very suitable for loading on the UAV to realize the function of directly processing data for target detection after aerial photography. And federated learning system realizes the function of uploading the trained model parameters without transmitting the original data to the server. It is found that the model accuracy is improved by 10% and the training time is shortened by 10%, which can also reflect the better performance of the federated learning system under deep neural network in a certain sense.

However, the system is flawed. Due to the heterogeneity of each node device, the slow speed of training node leads to slowing down the whole iteration speed, and the fast speed of training node is in standby state for a long time. So further work is needed to optimize the federated learning code, allows the client to calculate according to their own strength and the differences between communication speed regulating the number of training samples.

Acknowledgments

My sincere thanks to Chao Dong and Yuben Qu for providing the research ideas and equipment support, and also to Feiyu Wu for his detailed guidance, and finally to Jamie Vicary and Ekrem for their guidance in writing the paper.

References

[1] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[C]//Artificial Intelligence and Statistics. PMLR, 2017: 1273-1282.

[2] ZHOU C X, SUN Y, WANG D G, et al. Survey of federated learning research[J]. Chinese Journal of Network and Information Security, 2021, 7(2).

[3] Feki I, Ammar S, Kessentini Y, et al. Federated learning for COVID-19 screening from Chest X-ray images[J]. Applied Soft Computing, 2021, 106: 107330.

[4] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, et al. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

[5] https://www.nvidia.com/zh-tw/autonomous-machines/embedded-systems/jetson-nano/.

[6] Wang H, Yurochkin M, Sun Y et al. Federated learning with matched averaging, 2020: arXiv:2002.06440.

[7] Hanley, J.A., McNeil, B.J., The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 143, 29–36. 1982.

[8] D. R. Prado, "The Generalized Intersection Approach for Electromagnetic Array Antenna Beam-Shaping Synthesis: A Review," in IEEE Access, vol. 10, pp. 87053-87068, 2022, doi: 10.1109/ ACCESS. 2022. 3199734.

[9] N. Pereira, "PereiraASLNet: ASL letter recognition with YOLOX taking Mean Average Precision and Inference Time considerations," 2022 2nd International Conference on Artificial Intelligence and Signal Processing (AISP), 2022, pp. 1-6, doi: 10.1109/AISP53593.2022.9760665.

[10] Reprinted from https://blog.csdn.net/smartvxworks/article/details/121899008.

[11] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [C] // Advances in neural information processing systems. 2012: 1097-1105.

Appendix

A. Jetson TX2

The Jetson TX2 is NIVDIA's AI targeting update after the launch of Jetson TK1 and TX1.

The GPU and CPU of TX2 have been upgraded, the memory has increased to 8GB, the storage has increased to 32GB, the TX2 supports Wifi and Bluetooth, the code supports H.265, and the body size is also small.

According to NVIDIA official introduction, Jetson TX2 provides two operating modes: one is MAX Q, energy efficiency ratio can reach the highest, which is the previous generation of TX1 twice. And power consumption is below 7.5W; The other is MAX P, the performance can achieve the highest, of which energy efficiency ratio can also do the previous generation of 2 times. And the power consumption is below 15W. [10]

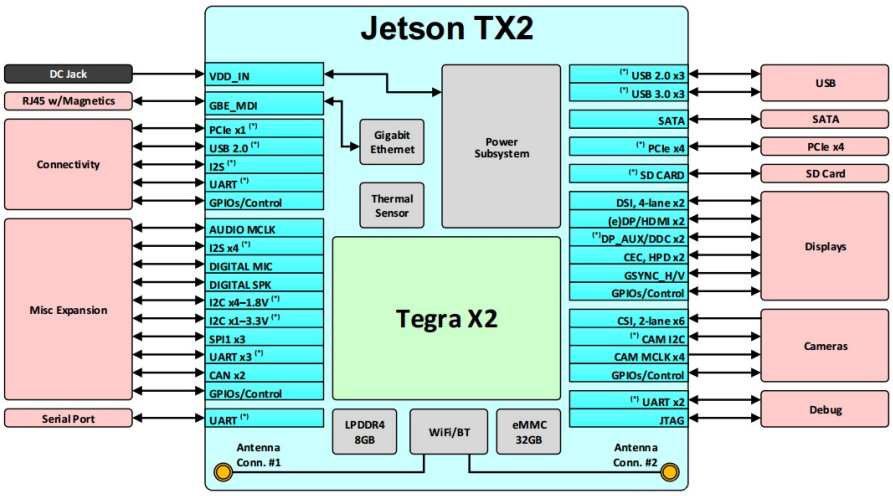

Figure 10. The architecture of Jetson TX2[10]

B. Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs) constitute one such class of models. Their capacity can be controlled by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies). Thus, compared to standard feedforward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters and so they are easier to train, while their theoretically-best performance is likely to be only slightly worse.[11]