Electronic Design

By MICHAEL TRUONG LE, Software AI Lead, Recogni, www.recogni.com

Developing Advanced 3D Object Detection for Autonomous Vehicles

Real-world vision systems rely on 3D object detection, which is critical in developing perception capabilities for AVs and mobile autonomous robots. But what's the best approach to achieve the best quality data for the most accurate detection?

3D object detection (3DOD) is central to real-world vision systems and a critical component in the development of perception capabilities for autonomous vehicles (AVs) and mobile autonomous robots. Real-time 3DOD performed at the frame rates required by AVs presents a very difficult engineering problem, though, because it demands significant computing resources from resource-constrained systems. Constraints in AVs include cost, size, power consumption, performance, and accuracy.

While developing 3DOD capabilities for Recogni's machine-learning (ML) perception chip—an embedded inference engine designed specifically for vehicles and robots—the engineering team developed a software architecture based on inputs from a high-resolution, high-dynamic-range (HDR), high-frame-rate stereo camera pair that meets the unique constraints and requirements for AV-based systems while delivering excellent accuracy.

Recogni's end goal is to develop highly accurate, AI-based perception for AVs. The company's experience in developing an architecture to reach this goal illustrates the significant differences between conventional and general-purpose versus purpose-built approaches that yield high throughput, low latency, and low power implementation.

One key decision that drove the architectural design for the ML perception system was to use a stereo pair of high-resolution video cameras as the sole inputs for the system. Although 3D LiDAR sensors are frequently paired with stereo cameras to implement perception systems, LiDAR sensors remain expensive. Thus, implementing machine perception with the required performance and accuracy using only a high-resolution stereo camera pair can deliver significant cost savings.

Picking a 3DOD Approach

Recogni's team evaluated several approaches for stereo 3DOD and ultimately developed their own approach for high accuracy and throughput as a single-shot stereo 3D detection network. The left and right image streams from a stereo video camera pair provide the only inputs to the architecture.

The system operates directly on distorted and non-rectified images (DNR), which drastically reduces system latency by eliminating the need for a separate hardware warping unit. By working on DNR data, the computational load shifts to the convolutional accelerator. Although this design increases the accelerator's overall computational load, sufficient resources in the accelerator are available to handle the load, so the design remains feasible.

The perception model passes both the left and right image through a shared-weights multiresolution-FPN (feature pyramid network), with a ResNet32 as a backbone. The resulting pairs of left and right multiresolution feature maps are then combined in so-called disparity cost volumes, effectively embedding the concept of depth/disparity into the neural network's architecture itself.

You can think about generating cost volumes as the process of overlaying the left and right camera images, then shifting the right image to the left one column at a time until the images overlay. The number of shift operations needed to align the left and right images is then used to infer depth information.

We then create 3D bounding boxes and classify objects using the output of these multiple cost volumes and subsequent detection heads. Once bounding boxes are accurately predicted around objects and the objects are classified, the ML perception task is complete.

High-Resolution, High-Frame-Rate Video Streamlines 3DOD

High-resolution HDR video at high frame rates is central to making this architecture practical and effective. Fortunately, the advent of cellphones has made low-cost, high-resolution, HDR video sensors readily available from several vendors, including Omnivision, onsemi, and Sony. Normally, the video streams from the stereo camera pair would be preprocessed using rectification and undistortion to simplify the convolutional neural network's (CNN) 3DOD task by making it easier to identify matching points in stereo video frames.

Rectification involves warping the stereo frame pairs so that matching points in each frame align on epipolar lines corresponding to pixel rows in the video frames. This transformation reduces the point-matching problem to a single dimension. The undistortion transformation reverses the effects of imperfect lenses on a captured frame so that straight lines are actually straight and not curved, for example.

Both rectification and undistortion require many computing cycles, so they're costly in terms of computing capability and power consumption. On top of that, both transformations introduce latency. All of these attributes are undesirable for a latency optimized system.

The human brain performs 3DOD operations without resorting to rectification and undistortion on an image level, and it's quite possible to train a highly capable CNN to detect and identify 3D objects without either rectification or undistortion at the input stage. The tradeoff is that additional processing power is needed for training and inference.

Training is only performed once for the CNN, so the additional processing required to train a CNN to perform the 3DOD task without rectification and undistortion is not a significant cost. For inference, working with DNR stereo image frames puts additional burden on the CNN accelerator. Therefore, the CNN must be implemented with a perception chip that has sufficient processing power to handle the task.

Training and Inference Need Different Numeric Representations

Training using DNR stereo image frames can be accomplished with GPUs and floating-point numeric representation, as is the rule with CNN training. However, inference needs to use a more compact numeric representation format to fit the needs of an accelerator chip. The compact number format minimizes off-chip storage and reduces the complexity of on-chip multiplication operations within the CNN accelerator, which in turn cuts the required power consumption and reduces latency.

Recogni's ML perception chip employs a compact number format based on logarithmic math and is therefore able to replace the multiplication operations in the CNN with addition. Optimally clustering weights in the trained CNN achieves further optimizations.

Using data types that are smaller than the standard Float32 values employed for training (known as quantization) is a frequently used strategy for inferencing on edge devices, but quantization presents its own challenges. Typically, the quantization strategy involves reducing weights to 8-bit floating-point numbers or integers to reduce the required amount of memory for storing weights by 75% while significantly reducing computational needs. However, quantizing weights in this way makes accurate regression prediction far more difficult.

For example, imagine that you want to predict the distance of objects relative to the AV at ranges to 250 meters. Using 8-bit integers to encode the distance prediction gives only 256 distinct values, with a resulting resolution of roughly one meter. That's clearly not an acceptable resolution for navigation or collision avoidance.

Recogni solved this issue using an approach called hybrid regression, which defines a number of bins to first get a coarse localization of a detected object. This is a classification problem for CNNs. To then achieve the necessary accuracy for automated driving, the CNN also predicts an offset from the bin's center. This offset is derived by regression; such a hybrid approach achieves a possible accuracy of 2 cm.

For AV applications, there's a huge difference between 1-m and 2-cm resolution. However, generating multiple classification and regression outputs to solve a regression problem with a CNN comes at an increased computational cost. This extra cost is easily absorbed by the computational resources in the Recogni ML perception chip.

Recogni's Seefar Dataset

The computational resources of the Recogni ML perception chip enable it to perform real-time inference on stereo pairs of 8-Mpixel images. The high resolution of these stereo video frames are essential to making the YoloStereo3D model practical for real-time navigation and collision avoidance. However, none of the industry and open-source datasets met the company's training requirements because they're all based on low-resolution images.

Consequently, Recogni recorded its own training dataset. The data-collection rig uses two cameras with onsemi 8.3-Mpixel (3848 x 2168 pixel) image sensors as a stereo video camera pair. The rig also incorporates a Hesai Pandar128 LiDAR for direct distance measurements that are incorporated into the dataset. While the LIDAR data isn't directly used for inference, it makes labeling much easier. Recogni works with a partner for data annotation. The partner provides 3D bounding boxes for a large number of object classes.

One interesting aspect of working with this high-resolution dataset is that it reveals unsuspected performance bottlenecks. The Recogni dataset's high-resolution, high-frame-rate, HDR video images at a resolution of 1920 × 1080 × 4 are quite different from the standard ImageNet dataset's 256

x 256 × 3 images, yet the higher resolution is essential in this application. At about 85X the size per tensor, data-transfer rate suddenly becomes extremely significant.

Measuring 3DOD Performance

Detailed system performance measurements guide development decisions, suggest architecture improvements, establish CNN training steps, and help to indicate when training has reached a stopping point.

Measuring performance was even more important. That's because it allowed Recogni to assess the resulting performance and accuracy due to numeric conversion from the 32-bit floating-point numbers used for training weights to the converted, fixed-point, logarithmic weights used by the company's ML perception chip for inference.

Researchers apply several metrics to evaluate 3DOD performance. For example, the metrics used in the KITTI challenge or the NuScenes detection metrics match ground-truth objects with predicted objects and then compute metrics based on the match between ground truth and prediction.

While the engineering team used both the KITTI and the NuScenes metrics, it was discovered that they have limited usability in an AV application because they're built for synthetic object detection challenges Object detection insights that reflect the real world and real-world performance were needed.

Here's an example to illustrate the difference between synthetic challenges and real-world performance. For a single car and a single prediction, the KITTI metrics require a 70% 3D-IoU (3D intersection over union) score to associate ground truth and the predicted bounding boxes. Assuming we predict the object's dimensions of 4.5 × 2 meters perfectly, a lateral offset error of only 80 cm prevents the ground truth and predicted bounding boxes from matching, according to the KITTI matching criteria.

As a result, the KITTI metric returns both a false negative result (an actual, ground-truth car that hasn't been detected by the ML perception model) and a false positive object (an additional detection that's not based on a true object). In addition, the KITTI metric produces the same outcome whether the prediction is off by 80 cm or 80 m. For our purposes, we need to know the actual lateral offset between the ground truth and the prediction, which isn't provided by the KITTI metric.

Customizing Performance Metrics

Consequently, the engineering team developed its own metrics to provide real-world performance measurements. Custom metrics were based on two principles:

1. Matching bounding boxes from the model's perspective, based on 2D image data. 2D-IoU works well for this metric because it's fairly invariant to distance. The Recogni ML perception model matches both close predictions and distant predictions similarly.

2. Building a set of human-interpretable metrics. A human should be able to look at a metric or a metric change and easily see what the change means. For example, decreasing rotation error from 4 degrees to 2 degrees is a clearly understandable result, while a decrease in the area under the rotation error curve from 0.2 to 0.15 clearly is not.

Relevant scores, calibrated to be between 0 and 1, are computed, with 0.5 being the "minimum acceptable error." This score makes it possible to see how well the CNN is making predictions at a glance.

Finally, a weighted score is computed from all computed scores that serves as a proxy value for general quality. While it's impossible to accurately represent a dozen metrics in a single scalar value, an overall score allows for quick comparisons of test runs. Experience has shown that team members converge to one metric anyway when lacking a single predefined metric, so one metric is generated that helps to compare experimental runs.

Of course, these metrics aren't just computable for all of the data—care was taken to keep them meaningful for any arbitrary group of samples. If a team member is interested in which single frame produced the worst result, or the best-performing recorded sequence, or the recorded day with the worst depth error, metric computation can answer all of these queries.

Conclusion

Over the past several years, university researchers have developed many useful datasets and performance metrics for training and evaluating CNNs. However, the datasets and metrics generated by academia may not be the best tools for developing real-world applications such as 3DOD. You may need to develop your own datasets for CNN training.

Such datasets should reflect the actual real-world data a system will provide to the CNN, with the correct bit resolution and the right sensor update rate. Finally, the metrics you use to evaluate CNNs in these applications should reveal real-world performance with easily understood results.

References

Liu Y, Wang L, Liu M. "Yolostereo3d: A step back to 2d for efficient stereo 3d detection." In 2021 IEEE International Conference on Robotics and Automation (ICRA) 2021 May 30 (pp. 13018-13024)

Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry. "End-to-End Learning of Geometry and Context for Deep Stereo Regression." In 2017 IEEE International Conference on Computer Vision (ICCV) 2017: (pp. 66-75)