The 2020 Low-Power Computer Vision Challenge

Xiao Hu[1], Ming-Ching Chang[2], Yuwei Chen[2], Rahul Sridhar[3], Zhenyu Hu[3], Yunhe Xue[3], Zhenyu Wu[3], Pengcheng Pi[3], Jiayi Shen[3], Jianchao Tan[4], Xiangru Lian[4], Ji Liu[4], Zhangyang Wang[5], Chia-Hsiang Liu[6], Yu-Shin Han[6], Yuan-Yao Sung[6], Yi Lee[6], Kai-Chiang Wu[6], Wei-Xiang Guo[7], Rick Lee[8], Shengwen Liang[8], Zerun Wang[9], Guiguang Ding[9], Gang Zhang[10], Teng Xi[10], Yubei Chen[11], Han Cai[12], Ligeng Zhu[12], Zhekai Zhang[12], Song Han[12], Seonghwan Jeong[13], YoungMin Kwon[13], Tianzhe Wang[14], Jeffery Pan[15], Changcheng Tang[16], Tianyi Lu[16], Shuang Liang[16], Xuefei Ning[17]

[1]*Purdue University, West Lafayette, IN, USA* [2]*University at Albany, State University of New York, NY, USA*
[3]*Texas A&M University, College Station, TX, USA* [4]*Kwai Inc. Seattle AI lab, Seattle, WA, USA* [5]*The University of Texas at Austin, Austin, TX, USA*
[6]*National Chiao Tung University, Hsinchu, Taiwan* [7]*National Tsing Hua University, Hsinchu, Taiwan*
[8]*SKLCA, Institute of Computing Technology, CAS, Beijing* [9]*School of Software, Tsinghua University, Beijing, China*
[10]*Department of Computer Vision Technology (VIS), Baidu Inc.* [11]*University of California, Berkeley, CA, USA*
[12]*Massachusetts Institute of Technology, Cambridge, MA, USA* [13]*The State University of New York, Korea* [14]*Georgia Institute of Technology, Atlanta, GA, USA*
[15]*Phillips Academy, Andover, MA, USA* [16]*Novauto Technology Co.Ltd. Beijing* [17]*Tsinghua University, NICS-EFC Lab*

***Abstract*—AI computer vision has advanced significantly in recent years. IoT and edge computing devices such as mobile phones have become the primary computing platform for many end users. Mobile devices such as robots and drones that rely on batteries demand for energy efficient computation. Since 2015, the IEEE Annual International Low-Power Computer Vision Challenge (LPCVC) was held to identify energy-efficient AI and computer vision solutions. The 2020 LPCVC includes three challenge tracks: (1) PyTorch UAV Video Track, (2) FPGA Image Track, and (3) On-device Visual Intelligence Competition (OVIC) Tenforflow Track. This paper summarizes the 2020 winning solutions from the three tracks of LPCVC competitions. Methods and future directions for energy-efficient AI and computer vision research are discussed.**

***Index Terms*—Low-power, computer vision, challenge, drone, scene text, FPGA, model compression, knowledge distilling, NAS.**

I. Introduction

Low-Power Copmuter Vision Challenge (LPCVC) [1] aims to identify energy-efficient solutions for computer vision. These solutions have a wide range of applications in mobile phones, drones, autonomous robots, or any intelligent systems equipped with digital cameras carrying limited energy [1].

The 2020 LPCVC contests were hosted online with submission window opened from 7/1 to 7/31 in 2020, featuring three challenge tracks sponsored by Facebook, Xilinx, and Google.

1) **PyTorch UAV Video Track:** Optical character recognition of English letters and numbers in videos captured by an unmanned aerial vehicle (UAV, also called drone) using PyTorch Mobile.
2) **FPGA Image Track:** Classification of objects from 1000 ImageNet categories using Field Programming Gate Array (FPGA).
3) **On-device Visual Intelligence Competition (OVIC) Tensorflow Track:** Real-time object detection and image classification using Tensorflow.

[1]LPCV: http://lpcv.ai is previously known as the Low-Power Image Recognition Challenge (LPIRC). All 2020 LPCV evaluation results are at https://lpcv.ai/scoreboard/Video20.

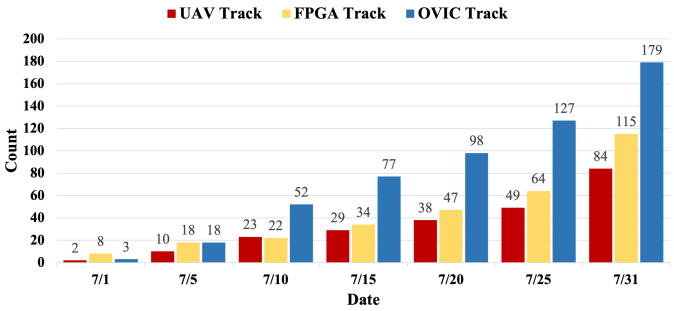

Fig. 1. Submission count for all 3 LPCVC tracks during the opening month.

During the month-long challenge, many teams submitted multiple solutions and witnessed the growing scores. In total, 46 teams submitted 378 solutions in all tracks, see Fig. 1. Notably, the newly added UAV Video Track adds video processing contest, which expands the image-based LPIRC contests that were held annually since 2015.

II. PyTorch UAV Video Track

In this contest, participant teams must recognize English letters and numbers on the walls from the provided videos recorded by an unmanned aerial vehicle (UAV). Submission must be performed using PyTorch to recognize the question phrases and identify the respective answers (letters and numbers) appearing in the video frames. Fig. 2 shows two examples from the challenge. Evaluation was conducted on Rawspberry Pi 3B+. The accuracy score was calculated using the Levenshtein distance, and the energy consumption was recorded on the Yokogawa WT310 Power Meter. Eventually, 11 teams submitted 84 solutions in this contest.

A. Team LPNet

Team LPNet from **ByteDance Inc.** (*Xuefeng Xiao, Xing Wang, Li Lin, Tianyu Zhao, Ni Zhuang, Huixia Li, Xin Xia, Can Huang, Linfu Wen*) won the 1st place of this contest.

Improvement from team focused on (1) an efficient video decoder runnable on embedded device and (2) key frame filtering. We fine-tuned the model given by Facebook with

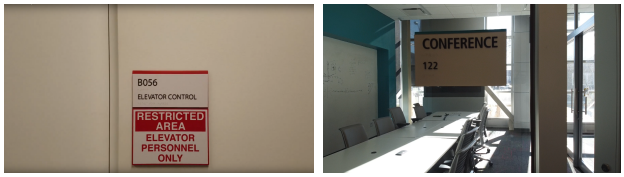

Fig. 2. **Track 1.** Example video frames used in the PyTorch UAV Video Track. For the question of "RESTRICTED AREA", the answer should be "B056 ELEVATOR CONTROL ELEVATOR PERSONNEL ONLY". For the question of "CONFERENCE", the answer should be "122".

other open-source OCR datasets, and apply the Connectionist Temporal Classification (CTC) to achieve better results with efficient performance. For key frame selection, we process key frames using dense optical flow to filter out low-quality frames (*e.g.* blurry characters), and perform OCR on frames remaining still and clear. Eventually, we managed to select 100 key frames from the 5000 frames from the sample video that produced higher accuracy.

B. Team TAMU-KWAI

Team TAMU-KWAI from **Texas A&M University and Kwai Inc.** (*Yunhe Xue, Zhenyu Hu, Rahul Sridhar, Zhenyu Wu, Pengheng Pi, Jiayi Shen, Jianchao Tan, Xiangru Lian, Zhangyang Wang, Ji Liu*) won the 2nd place of this contest.

The proposed approach includes a two-step spotting pipeline that addresses energy efficiency from both the data and model levels. The E^2VTS two-step text spotting system adopts Efficient and Accurate Scene Text Detector (EAST) as the text detector, and Convolutional Recurrent Neural Network (CRNN) as the text recognizer. Crop+Resize is adopted instead of aligned RoI-Pooling because feature sharing and joint training in the aligned RoI pooling will lead to sub-optimal performance for both detection and recognition.

Model-level efficiency is improved through an early-exiting mechanism that terminates a running text-spotting pipeline for ineffective cases. A proper sparsity rate is set and $\ell 1$ filter is used to prune both EAST and CRNN model with little performance drop. For quantization, on Raspberry Pi with ARM CPUs, PyTorch provides QNNPACK backend for acceleration. Static post quantization is applied on all convolutional and fully-connected layers. Dynamic post quantization is applied on the LSTM modules of CRNN. Data-level efficiency is improved by filtering out the text-free and poor-quality images before fed to the text spotting pipeline. Non-text regions are cropped out to improve the signal-to-noise ratio (SNR) in the image.

C. Team C408

Team C408 from **The State University of New York, Korea** (*Seonghwan Jeong, Jay Hoon Jung, YoungMin Kwon*) won the 3rd place of this contest.

The proposed scene text detection and recognition process consists of three stages: (1) *image pre-processing* where images are resized and reshaped, (2) *scene text detection* where the texts are detected and localized, (3) *optical character recognition (OCR)* where the texts are identified and recognized. Performance improvement is achieved by applying a pre-processing and reorganizing the pipeline in Fig. 3. The proposed pipeline achieves about 1.8 times higher performance score against the baseline. Accuracy score was increased by 17%, while the energy consumption was decreased by 30%. The increased accuracy is largely due to the duplicated text detection and the text recognition stages. The decreased energy consumption is mainly obtained by pruning blurry images. The inference computation cost is reduced by 45%.

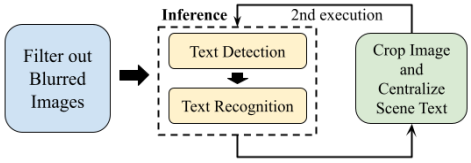

Fig. 3. **Team C408** processing pipeline.

III. FPGA Image Track

This track uses Ultra96-V2 + Xilinx Deep Learning Processor Unit (DPU) running software PYNQ + Vitis AI. The challenge recognizes objects in images. In total, 19 teams submitted 115 solutions.

A. Team OnceForAll

Team OnceForAll from the **MIT Han Lab.** (*Zhekai Zhang, Han Cai, Yubei Chen, Jeffrey Pan, Ligeng Zhu, Tianzhe Wang, Song Han*) won the 1st place of this contest.

We employ the Once-For-All (OFA) AutoML https://github.com/mit-han-lab/once-for-all to automatically design the neural networks for this contest. OFA is an efficient AutoML technique that decouples model training from architecture search. By selecting different parts from the OFA network, many different sub-networks with zero training cost can be obtained, while maintaining the same level or even higher accuracy than the counterparts trained from scratch. This allows us to quickly obtain a serial of neural networks that cover different latency ranges. We use the MobileNetV2 design space to build our OFA network, which includes elastic kernel size ([3, 5, 7]), elastic expand ratio ([4, 5, 6]), elastic depth ([2, 3, 4]), and elastic resolution (128 to 224). We also optimize the search space according to the DPU hardware parameters. We use the B1600 configuration of DPU, which has 10 input channel parallelism $\times$ 10 output channel parallelism. We make the number of channels of all layers be the multiple of 10 (instead of 8) to improve the utilization.

To obtain a specialized neural network for a given latency constraint, we build a lookup-table-based latency predictor. We enumerate all possible layers in our search space and collect their latency information on FPGA. Since DPU processes the neural network in a layer-by-layer manner, the whole network's latency can be accurately estimated by adding up each layer's latency. Based on the latency predictor and the OFA network, we use evolutionary search to select a sub-network for each latency constraint. After specialized sub-networks are obtained, we fine-tune them on the ImageNet dataset. Finally, tools provided in Vitis AI are used to conduct quantization-aware fine-tuning and final deployment.

B. Team MAXX

Team MAXX from **National Chiao Tung University (Taiwan)** (*Shih-Yu Wei, Xuan-Hong Li, Yu-Da Ju, Juinn-Dar Huang*) won the 2nd place of this contest.

HarDNet is chosen as the proposed backbone for low-memory traffic requirement. Specifically, HarDNet-39 comes with lower latency, while HarDNet-68 provides higher performance accuracy. We made two substantial changes to the network for optimization. For low latency, we reduced the HarDBlks size by fine-tuning k. We also removed the first 3 1×1 convolutions in the 5 available HarDBlks. To maximize the utilization of the reduced model, the last two 1×1 convolutions were replaced with 3×3 convolutions. We can thus lower the latency while preserving the accuracy of the baseline network. We chose Vitis-AI for quantizer and compilation. Due to the 8-bit quantization provided by the DPU, accuracy drop is expected. To this end, we increased the use of calibration images. Accuracy drop was reduced to be within 1% after the use of $10,000$ calibration images.

C. Team Water

Team Water from **SKLCA, Institute of Computing Technology, and CAS** (*Shengwen Liang, Rick Lee, Ying Wang, Cheng Liu, Huawei Li*) won the 3rd place of this contest.

Multiple DPU configurations were tested to determine the best setting regarding logic elements and power supply to run on the Ultra-96 v2 board. The optimal setting is B2304 working at 400MHz, which utilizes 90.56% of the DSP resource, which worked on the board with limited power supply after upgrading the firmware of PMIC. EfficientNet-lite is used as a baseline considering the trade-off between latency and accuracy. However, the architecture of original Google EfficientNet-lite0 cannot meet the requirement of the configuration determined before. We shrunk the output dimension of the 6-th layer from 192 to 162, to ensure that the model can run on the Ultra96-V2 board. To reduce the latency, we profiled the whole pipeline including image pre-processing, data copy, inference and label calculation. We found the running time is mostly spent on image processing, which also causes the low utilization of DPU. A new 4-thread configuration is re-implemented to best reduces the execution time of the image pre-processing. Source code is provided in https://github.com/shengwenLeong/lpcvc2020_water.

IV. OVIC TensorFlow Track

The On-device Visual Intelligence (OVIC) Tensorflow competition includes three different subtracks. In total, 16 teams submitted 179 solutions to all subtracks.

A. Subtrack - Interactive Object Detection

This contest subtrack is on COCO detection models operating at 30 ms per image on Pixel 4 smartphone CPU.

1) Team OnceForAll: Team OnceForAll from the **MIT Han Lab.** (*Yubei Chen, Jeffrey Pan, Han Cai, Zhekai Zhang, Tianzhe Wang, Ligeng Zhu, Song Han*) won the 1st place of this contest.

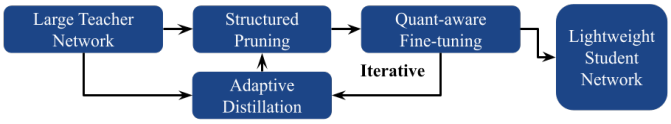

Fig. 4. **Team Novauto:** Detector compression pipeline with adaptive distillation.

A variation of the one-for-all NAS is used to design the network within the MobilenetV2+SSD model space for this contest. As described in § III-A, OFA decomoses the NAS search and model training, which enables us to find a series of optimal models in a short amount of time. Based on the score formula, we set $<$ 24ms as a hard limit in our model search. And the best model we found achieved 23.2 mAP on COCO validation set with a latency 23.92ms. Due to the library difference, our internal latency measurement was slightly slower than the official measurement and the model final latency was 23.48 ms. Our training include an imagenet pretraining, multi-stage full-precision training, quantization aware training. Additionally, we also used the best model (X-101-64x4d-FPN) from the open-mmlab library for distillation on additional data. For the final submission, we combined the training set, validation set, and the pseudo labeled data into one big training set for fine-tuning.

2) Team SjtuBicasl: Team SjtuBicasl from **Shanghai Jiao Tong University** (*Lining Hu, Xinzi Xu, Yongfu Li, Weihong Yan, Xiaocui Li, Yuxin Ji*) won the 2nd place of this contest.

3) Team Novauto: Team Novauto from **Novauto Technology Co. Ltd. Beijing.** (*Changcheng Tang, Tianyi Lu, Zhiyong Zhang, Shuang Liang, Tianchen Zhao, Xuefei Ning, Shiyao Li, Hanbo Sun, Shulin Zeng*) won the 3rd place of this contest.

Model pruning and adaptive distillation on the baseline model are implemented to improve the accuracy and speed. Compared with the baseline model of 23.0% mAP with 26.5ms latency, we achieved 22.7% mAP on COCO val2017 at a latency of 24.2ms on Pixel 4 smartphone CPU. This brings 0.0326 improvement over the current pareto frontier.

Adaptive distillation is used to boost the performance of the lightweight student network for object detection, by transferring the knowledge of the high-precision teacher network to the student network. This mitigates the class-imbalance problem caused by the direct calculation of all prediction results to compute the classification distillation loss. In our implementation, all negative anchors are excluded. Only the loss of matched anchors are accumulated as the soft loss. This design brings about 1.5% mAP improvement.

A **detector compression** pipeline is combined with adaptive distillation as in Fig. 4, to obtain maximal compression and speed-up. This mitigates the gap between the required calculation and the limited computation resource on an edge device. A structured pruning strategy is adopted to automatically prune the large teacher network based on the results of sensitivity analysis. Fine-tuning of the pruned detection network is coupled with quantization aware training. Our lightweight object detector achieves 22.7% mAP on the COCO object detection dataset at a latency of 24.2ms when running on the Pixel 4 smartphone CPU.

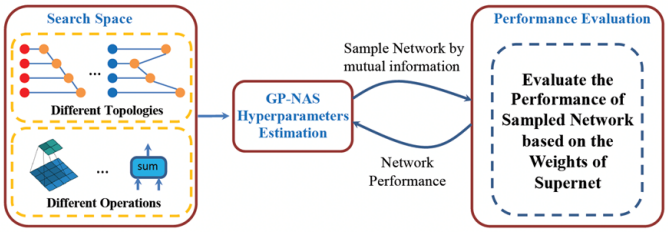

Fig. 5. **Team BAIDU & THU.**: the proposed GP-NAS based approach.

B. Subtrack - Real-time Image Classification on Pixel 4

This contest subtrack is on ImageNet classification models operating at 10 ms per image on Pixel 4 smartphone CPU.

1) BAIDU & THU: Team BAIDU & THU from **Baidu Inc. & Tsinghua University** (*Zerun Wang, Teng Xi, Dahan Gong, Fei Tian, Zizhou Jia, Cheng Cui, Xiaoyu Huang, Zhihang Li, Fan Yang, Tianxiang Hao, Shengzhao Wen, Huiyue Yang, Gang Zhang, Guiguang Ding*) won the 1st place of this contest.

A Gaussian Process based Neural Architecture Search (GP-NAS) is adopted in our pipeline as in Fig. 5. The correlation between the network architecture and its performance is modeled in the GP Bayesian formulation. In GP-NAS, such correlation is modeled using the kernel function and mean function. Specifically, based on the hyperparameters of GP-NAS, we have the ability to efficiently predict the performance of any child network in the search space. The NAS problem is thus converted as the hyper-parameter estimation of GP-NAS. Sub-networks can be effectively sampled via a mutual information maximization sampling algorithm. According to the performance of the sampling network, posterior distribution of the hyperparameters of GP-NAS can be updated gradually and effectively. Based on the estimated GP-NAS hyperparameters, optimal model structure that satisfies specific latency constraints can be predicted. To improve search efficiency, performance evaluation is carried out based on weight-sharing super-net solutions.

2) Team Imchinfei: Team Imchinfei from **Beihang University** (*Fei Qin, Yangyang Kuang*) won the 2nd place of this contest.

3) Team LPNet: Team LPNet from **ByteDance Inc.** (*Xuefeng Xiao, Jiashi Li, Xin Xia, Huixia Li, Linfu Wen*) won the 3rd place of this contest.

C. Subtrack - Real-time Image Classification on LG G8

This contest subtrack is on ImageNet classification models operating at 7ms per image on LG G8 smartphone DSP.

1) Team LPNet: Team LPNet from **ByteDance Inc.** (*Xuefeng Xiao, Jiashi Li, Xin Xia, Huixia Li, Linfu Wen*) won the 1st place of this contest.

2) Team Imchinfei: Team Imchinfei from **Beihang University** (*Fei Qin, Yangyagn Kuang*) won the 2nd place of this contest.

3) Team OnceForAll: Team OnceForAll from the **MIT Han Lab.** (**Yubei Chen, Jeffrey Pan, Han Cai, Zhekai Zhang, Tianzhe Wang, Ligeng Zhu, Song Han**) won the 3rd place (Tie) of this context.

The OFA NAS search algorithm is used to find the optimal model. The search space has to be modified due to the limit of supported operators on the DSP. Considering the potential miss match between our latency measurement and the official latency measurement, we set < 6.5ms instead of < 7ms as the our latency constraint. The final model reaches 74.5 top-1 accuracy trained on the ImageNet training set with latency 6.5ms. The training and validation sets are combined for additional fine-tuning for final submission.

4) Team FoxPanda: Team FoxPanda from **National Chiao Tung Univ. and National Tsing Hua Univ.** (*Chia-Hsiang Liu, Wei-Xiang Guo, Yuan-Yao Sung, Yi Lee, Kai-Chiang Wu*) won the 3rd place (Tie).

The proposed search methodology is based on *multivariate regression* to reduce the NAS space and time. We randomly sampled 300 different sub-networks from the super-network for data collection required only 0.5 GPU hour with a NVIDIA 2080Ti. Multivariate regression analysis generates the predictions of accuracy and latency based on the collected data. The multivariate regression analysis leads to promising predictors with much less required data. Notably, our multivariate regression analysis is *explainable* regarding the network architecture parameters, which leads to significant reduction of NAS search space.

We next use the correlation coefficient and P-value of each variable obtained from multivariate regression analysis to reduce our search space. The correlation coefficient of a variable represents the influence of the variable, and P-value < 0.05 indicates high correlation with the output variable. Due to the reduced search space and the effectiveness of our predictors, our NAS converges fast oon the search of optimal architecture. Detailed explanation of our solutions and source code are provided in https://github.com/great8nctu/lpcvc20.

V. Discussions and Conclusion

In this 2020 LPCV CVPR Workshop, an online virtual panel were hosted, where speakers from Facebook, Google, and Xilinx shared their experiences on developing low-power computer vision systems. 100 + participants attended the video call, showing increasing interest from the academic and industry in low-power computer vision that can operate on battery-powered systems. The initial success of our new UAV video track in LPCVC also suggests the increasing interest of development of smart AI drones.

Acknowledgement. The LPCV organization thanks Facebook, Xilinx, and Google for sponsorship and support.

References

[1] S. Alyamkin, M. Ardi, A. C. Berg, A. Brighton, B. Chen, Y. Chen, H. Cheng, Z. Fan, C. Feng, B. Fu, K. Gauen, A. Goel, A. Goncharenko, X. Guo, S. Ha, A. Howard, X. Hu, Y. Huang, D. Kang, J. Kim, J. G. Ko, A. Kondratyev, J. Lee, S. Lee, S. Lee, Z. Li, Z. Liang, J. Liu, X. Liu, Y. Lu, Y. Lu, D. Malik, H. H. Nguyen, E. Park, D. Repin, L. Shen, T. Sheng, F. Sun, D. Svitov, G. K. Thiruvathukal, B. Zhang, J. Zhang, X. Zhang, and S. Zhuo. Low-power computer vision: Status, challenges, and opportunities. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(2):411–421, 2019. 1