EFFICIENT FPGA IMPLEMENTATION FOR VISUAL OBJECT SEGMENTATION

Team Members:

- Keerthi Priya Lankapalli (klankapalli)
- Sowmya Saraswathi Tippi Reddy (tippireddy)
- Harshit Sharma (hsharma27)

PROBLEM STATEMENT

Computer Vision is being used in a variety of domains like agriculture, autonomous vehicles, security etc. These algorithms work on large scale datasets and complex models. Traditionally the processing and computation required by these algorithms were being done on cloud, as they provide scalability and flexibility. In recent times, edge computing is becoming increasingly popular for computer vision applications, especially for those applications that require real-time results. It provides benefits like low power consumption, reduced latency and better security as data processing is done closer to the source.

In this project, we plan to perform the task of image analysis from video frames taken from various sources such as traffic cameras for identifying objects using effective deep learning models and then implement viable models on an FPGA simulation through VIVADO and metric against task key metrics (like accuracy, mean IoU, Jaccard Score, etc) and constraint performance metrics like resource consumption (e.g. latency, DSP, LUT). Using video object segmentation, we can automate tasks which would otherwise require tedious manual effort. Like in the case of surveillance videos, it can be used to alert security personnel to potential threats or suspicious activities in such use cases performing visual object segmentation on cloud adds latency, which is not a viable option.

FPGA implementation of these segmentation algorithms can provide faster results with a reduced power consumption. FPGAs facilitate parallel processing and for tasks like semantic segmentation of videos where large amounts of data need to be processed in parallel.

CURRENT STATE-OF-THE-ART MODELS

In this paper [1], a Quality-aware Dynamic Memory Network (QDMN) was proposed for video object segmentation. The authors address the mask error accumulation problem, where frames with poor segmentation masks are likely to be memorized by the model. The solution to prevent this issue is to evaluate the segmentation quality of each frame and selectively store the segmented frames. This model achieved state-of-the-art performance on DAVIS and YouTube-VOS benchmark datasets by achieving a mean Intersection over union (IoU) score of 91% on DAVIS 2016 dataset and 82% on YouTube-VOS dataset.

The authors in [2] presented a semi-supervised framework for visual object segmentation that was designed with a dynamically scalable architecture for speed-accuracy trade-offs. Associating Objects with Scalable Transformers approach was used to match and segment multiple objects with online network scalability. This model achieved a state-of-the-art performance of 93% IoU on DAVIS 2016 dataset. In [3], XMEM architecture was proposed to perform video object segmentation on long video datasets. Existing architectures use a single feature memory model that could result in a trade-off between memory consumption and segmentation accuracy. The XMEM model consolidates actively used working memory elements to a long-term memory which avoids the memory explosion problem. This model achieves an IoU score of 92% on the DAVIS 2016 dataset.

PROPOSED SOLUTION AND IMPLEMENTATION

Since there are many different state-of-the-art models available for visual object segmentation, it is important to choose the one that is most suitable for the task at hand. The first step in this plan is to establish a baseline by implementing the most pragmatic state-of-the-art (SOA) model for visual object segmentation. This model will serve as the starting point for further development and optimization towards FPGA implementation.

The implementation of the current state of the art models was not viable as the current code infrastructure for enabling the conversion of deep learning models for FPGA use have various limitations. Firstly, they do not work with models written as custom classes and the support for pytorch is limited to basic layers and networks only [10]. Hence, most of these models would have to be retrained in tensorflow requiring code conversion and heavy compute resources for model training.

Alternatively, we implemented the most implemented state of the art model UNET [11] for the task of visual object segmentation. Since the model was lighter in comparison with the other state of the art models, we were able to train the model with our personal compute resources over selected vehicle specific classes from DAVIS 2016 class. Additionally, due to the nascent state of the research in FPGA implementation of ML models, not all the layers are available in the code infrastructure for conversion including Conv2DTranspose. To tackle this issue we have replaced it with a combination of Upsampling 2D and Conv2D layers in the architecture which also proves to be more memory-efficient than using a Conv2D transpose layer. The Conv2D transpose layer requires more memory because it needs to compute a dense matrix multiplication, while UpSampling followed by a Conv2D layer only requires computing a sparse matrix multiplication. For training we used 50 epoch cycles with a total dataset of size 400 car images as seen on the road. These images were split into training and test datasets with a ratio of 9:1 respectively.

The hls4ml (High-Level Synthesis for Machine Learning) [10] library was used to convert the trained machine learning model into hardware design that could be implemented on FPGA. The HLS model plot is described in Fig 1.

The next step is to modify its architecture as necessary so that it is suitable for FPGA implementation. Since FPGA devices have limited resources compared to traditional computing devices, it may be necessary to optimize the architecture of the model to reduce resource utilization and ensure real-time performance. The optimizations of the model like pruning that is used to reduce the complexity and size of the model are still work in progress. The results described in the Evaluation section are the results of a non-optimized model. After optimizing the model, this model would be implemented on FPGA.

EVALUATION

The Tensorflow U-Net model was modified as stated above and this model was used to perform the object segmentation task on DAVIS 2016 dataset. The model was trained on 300 car images from the dataset. The results of evaluating the model on the test set are described in Table 1. The results of some sample predictions are displayed in Fig 2. We observed a continuous decrease in loss and continual increase in the accuracy values during the training process. These results prove that this model outperformed the existing state-of-the-art models for object segmentation tasks.

Test Metrics	Value
Accuracy	0.9856
IoU Coefficient	0.9669
Jaccard Index	0.9667

Table 1: Evaluation Results

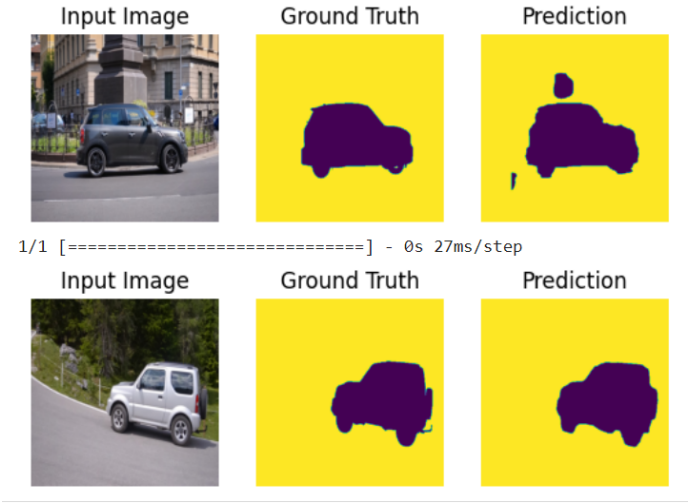

Fig 2: Sample predictions from images of DAVIS-2016

The baseline model implementation will give us the required metrics to begin our development for optimization. We have considered both software/task performance metrics and computer resource performance metrics for our study. For task performance we shall adopt metrics such as Accuracy, IoU (Intersection over Union) and F1 score. For FPGA implementation, the model was first compiled into a Vivado project using hls4ml by setting the current FPGA to Xilinx xcu250-figd2104-2L-e for virtual simulation. Later, it was synthesized and simulated using Vivado high level synthesis extension (VIVADO HLS) which shall deliver us the usage metrics such as latency, DSP, LUT etc. We believe that the two metrics will present themselves as a trade-off against one another.

NET PROPOSED CHANGES + NEXT STEPS:

1. Video Segmentation Task to Visual Segmentation Task
2. From Xmem to UNET based model implementation
3. Reporting Usage Metrics Through VIVADO HLS
4. Added Model Training step to ensure HLS4ML compatibility
5. Optimizing model obtained using pruning and other relevant techniques

TIMELINE

Task	Description	Date
Project Scoping & Proposal	Finalizing the project topic, execution plan, timeline and generate initial project proposal report	02/24/2023

Literature Review and Datasets Acquisition	Gathering more information on light weight state of the art models, acquiring data and analysis	03/03/2023
Implementing Baseline through model selected	End-to-End Baseline Development: GPU and task metric evaluation	03/21/2023
FPGA Simulation pipeline	Developing FPGA implementation strategy and compute performance evaluation of baseline	03/31/2023
Mid Term Report	Report the current challenges and metrics in developing an end-to-end solution and its current metrics.	04/04/2023
Model Development and Optimization	After obtaining the baseline and establishing testing pipelines, we can begin the model and data iteration process to optimize for task and computer performance.	04/24/2023
Final Presentation	Report the results obtained through model iterations or new architecture development.	04/25/2023
Project Webpage	Compile results and ensure replicability of project through project organization via github, documentation and web page development	05/05/2023

REFERENCES

1. Liu, Y., Yu, R., Yin, F., Zhao, X., Zhao, W., Xia, W. and Yang, Y., 2022, October. "**Learning quality-aware dynamic memory for video object segmentation**." In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX (pp. 468-486). Cham: Springer Nature Switzerland.
2. Yang, Z., Miao, J., Wang, X., Wei, Y. and Yang, Y., 2022. "**Associating objects with scalable transformers for video object segmentation."** *arXiv preprint arXiv:2203.11442*.
3. Cheng, H.K. and Schwing, A.G., 2022, October. "**XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model."** In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII (pp. 640-658). Cham: Springer Nature Switzerland.
4. Elumalai, Naveen, ***FPGA Implementation for Real Time Video Segmentation Using Gaussian Mixture Model*** (March 9, 2015). Available at SSRN: https://ssrn.com/abstract=2575998
5. Xiaohao Xu, Jinglu Wang, Xiang Ming, and Yan Lu. 2022. ***Towards Robust Video Object Segmentation with Adaptive Object Calibration***. In Proceedings of the 30th ACM International Conference on Multimedia (MM '22). Association for Computing Machinery, New York, NY, USA, 2709–2718. https://doi.org/10.1145/3503161.3547824
6. Ghielmetti, Nicolò, Vladimir Loncar, Maurizio Pierini, Marcel Roed, Sioni Summers, Thea Aarrestad, Christoffer Petersson et al. "**Real-time semantic segmentation on FPGAs** for autonomous vehicles with hls4ml." Machine Learning: Science and Technology 3, no. 4 (2022): 045011.
7. El Hajjouji, Ismaïl & Mars, Salah & Asrih, Zakariae & El Mourabit, A.. (2019). ***A novel FPGA implementation of Hough Transform for straight lane detection***. Engineering Science and Technology, an International Journal. 23. 10.1016/j.jestch.2019.05.008.

8. Q. Xu, S. Varadarajan, C. Chakrabarti, and L. J. Karam, "**A Distributed Canny Edge Detector: Algorithm and FPGA Implementation**," *IEEE Transactions on Image Processing*, vol. 23, no. 7, pp. 2944–2960, Jul. 2014, doi: 10.1109/tip.2014.2311656
9. A. Ahamad, C. -C. Sun, H. M. Nguyen and W. -K. Kuo, "***Q-SegNet: Quantized deep convolutional neural network for image segmentation on FPGA,***" 2021 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Hualien City, Taiwan, 2021, pp. 1-2, doi: 10.1109/ISPACS51563.2021.9650929.
10. HLS4ML Software: https://fastmachinelearning.org/hls4ml
11. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "***U-net: Convolutional networks for biomedical image segmentation.***" In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pp. 234-241. Springer International Publishing, 2015.