Unified Framework for Vision Inference on the Edge

UNIFIED FRAMEWORK FOR VISION INFERENCE ON THE EDGE

A PREPRINT

Vysakh S. Mohan
Head of Artificial Intelligence
Accubits Invent - Artificial Intelligence R&D Lab,
Accubits Technologies Inc, Trivandrum, India
vysakh@accubits.com,
vsmo92@gmail.com

Kelvin Jose
Jr. AI Researcher
Accubits Invent - Artificial Intelligence R&D Lab,
Accubits Technologies Inc, Trivandrum, India
kelvin@accubits.com,
kelvinkonoor@gmail.com

March 11, 2020

ABSTRACT

Edge processing for computer vision systems enable incorporating visual intelligence to mobile robotics platforms. Demand for low power, low cost and small form factor devices are on the rise. This work proposes a unified platform to generate deep learning models compatible on edge devices from Intel, NVIDIA and XaLogic. The platform enables users to create custom data annotations, train neural networks and generate edge compatible inference models. As a testimony to the tool ease of use and flexibility, we explore two use cases — vision powered prosthetic hand and drone vision. Neural network models for these use cases will be built using the proposed pipeline and will be open-sourced. Online and offline versions of the tool and corresponding inference modules for edge devices will also be made public for users to create custom computer vision use cases.

***K*eywords** artificial intelligence · edge devices · computer vision · neural networks

1 Introduction

Artificial Intelligence (AI) systems are widespread in the current day and age. They solve problems in the areas of computer vision, natural language processing, speech/audio analysis, pattern recognition etc to name a few [10]. Technology has matured so much that AI use cases can be designed and executed by technologists with few lines of code. Popular deep learning libraries like Keras [5] and Google's TensorFlow [11] makes it easy to come up with AI solutions. Such widespread acceptance of this technology is due to several reasons, cheaper computing hardware is one among them. Ground works for deep learning and machine learning algorithms were laid a decade ago, but practical implementations surfaced after the introduction of GPUs and its supported libraries.

Computer vision applications majorly took advantage of Convolutional Neural networks (CNN) [16], which helped researchers solve problems like object detection, image classification, segmentation, feature extraction etc. Research materials often come with newer algorithms or tweaks to existing algorithms, as a way to solve or perfect an existing use case. Although we see overwhelming results, these algorithms or solutions are often constrained to just the research and most of these do not grow as a consumer/market ready application. Scalability often comes in the way of transferring these technologies to a market ready solution. As mentioned before, artificial intelligence solutions are complemented by the hardware they are deployed on. Neural network inference (when model is deployed live) requires the model to perform real time predictions without latency, but training a model is often an offline exercise, not performed regularly so hardware scalability may not be a direct constraint. In the test field, computing devices are often compact and low power devices. In this case a heavy neural network model could be a disastrous ordeal. One cannot expect the computing device to scale in such a scenario. It's hard to imagine a huge server grade GPU system running full-time inside an autonomous car, generating insights for the self driving system. Addressing this issue would enhance performance of such autonomous platforms and this is makes 'AI inference on the edge', the prime focus of this work. Newer scalable neural architectures will soon find its way to compact and more power efficient devices. The Tensor RT framework from NVIDIA, TensorFlow Lite from Google, OpenVINO from Intel are all aimed at achieving AI inference at edge

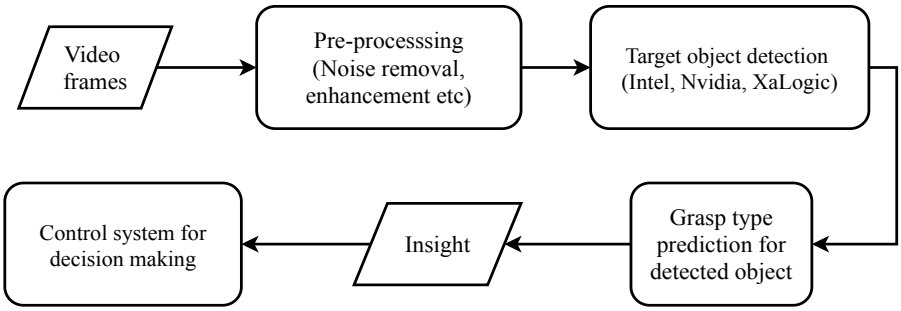

Figure 1: Workflow depicting video processing with on-board vision processing modules on prosthetic arm.

on their respective architectures. This work aims to build a unified platform that lets people build neural network models for specific use cases and deploy these models on edge devices like NVIDIA's Jetson Nano/TX2 and Intel's NUC/Movidious. This work will also be focusing on a new edge device from XaLogic, promising real time machine learning inference.

This work hypothesizes a central platform where users can add their custom image data, annotate them and produce neural networks compatible in Intel, NVIDIA and XaLogic devices. Two novel use cases are proposed, where the model generation pipeline solveing computer vision tasks in the field of prosthetics and drone vision is theorised.

2 Background and use cases

The use cases discussed here are explained with regards to the proposed system. It is assumed that the proposed system exists and the impacts on aforementioned technologies are discussed. Two key use cases for this work are – Vision powered prosthetics and Drone vision.

2.1 Vision powered prosthetics

Human prosthesis has been in research for a long time. Researchers have been experimenting with newer ways to build human body parts that augment/replace the functions of real ones. Prosthetic hand is one of those and is one key focus of this work. For upper-limb amputees and people with acute motor deficit, these prosthetic arms could prove to be a means of rehabilitation and can improve the persons life by enabling them to carry out daily activities with ease. Commercial prosthetic arms enable user control via myoelectric signals, where electric signals from muscle movements are recorded from skin surface of the residual limb [13, 7]. Myoelectric control of prosthetic arm was first implemented by Reinhold Reiter [14]. Due to funding constraints and lack of technology at the time, Reiter's work was abandoned midway. Advancements in sensor technology enabled compact and more technically capable prosthetic hands to be built, but their controls were limited [7, 12]. Accuracy close to 90%[1] was achieved for classification and decoding of myoelectric signals for control of finger movements, wrist control, grasp and elbow motion. These pattern recognition systems were usually laboratory tested and fails when scaled to clinical use.

A computer vision enabled prosthetic hands was designed to convert visual information of target objects to determine its size and predict appropriate grasp type. Došen *et al* [3, 4] presented a vision based prosthetic arm which triggers the camera using myoelectric signal to do a basic object recognition and distance calculation to target object with ultrasound. A two channel myoelectric arm was used by Ghazal *et al* [8] while developing a computer vision assisted mechanism to classify objects with respect to grasp type, without taking into account the dimensions of detected object. While the work achieved an accuracy of 84%, the neural network for prediction was developed, tested and benchmarked for a system with following configuration — Intel Core i5-47670 CPU (3.4 GHz), running a 64-bit Windows 7 operating system, with 32 GB RAM. Since prosthetic arms need to be modular, portable and compact, the system proposed in [8] may not be scaled to a commercial prosthetic arm because the processing hardware required is huge, cumbersome, power intensive and ergonomically non-feasible. This work aims to offload such intense neural network inferences to edge devices that are compact, require low power and are low cost. The proposed system can be used to annotate data related to grasp type based on target object, train neural networks and deploy it to edge devices from Intel, NVIDIA and XaLogic. This can greatly enhance the prosthetic arms design and make it a commercially feasible product. A proposed workflow for video processing with vision processing module attached to the prosthetic hand is illustrated in Figure 1. The flexibility offered by our proposed platform enables users to train the prosthetic arms to recognise custom objects,

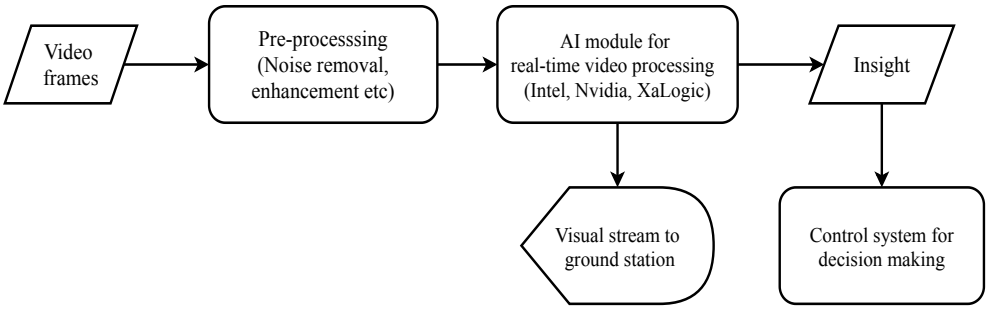

Figure 2: Workflow depicting video processing with on-board vision processing modules on UAV.

define new grasp types and experiment with newer designs. The tool will be fully open sourced to let users develop new use cases as well as contribute to incorporate new functionalities.

2.2 Drone vision

Drone or unmanned aerial vehicle (UAV) technology offers novel disruptive ways to solve problems in the field of logistics, surveillance, military, civilian applications etc. The technology has become mainstream and its easy to purchase these drones off-the-shelf based on what a users requirements are. Often times, these drones are limited in terms of flexibility. All commercially available drones are equipped with cameras and the footage captured during flight time is a valuable data. In current drones, these footage needs to be processed post flight, each time, at a base station to produce valuable insight. These video files received can often be noisy which delays its processing [15]. Having a real-time vision processing system as payload on these drones can incorporate some level of visual intelligence to these UAVs and generate in-situ analytical insights about the environment. Visual processing on-board UAVs are currently limited to basic image processing due to availability of constrained computing resources [6].

This work aims to enhance UAV vision processing by incorporating small form factor edge devices from Intel, NVIDIA or XaLogic, which are small footprint, tiny payloads, often consuming very low power. Edge processors enable processing deep neural nets to process video frames with acceptable frames per second (fps). Different object detection neural nets could be deployed on these devices and then used for real-time processing of video from on-board camera. Key feature of these devices are that they are modular and low-power consuming. Drones are battery operated and any power hungry devices on-board the UAV, could disrupt its total flight time. With the proposed platform, user can generate custom neural network architectures or train existing deep nets to detect/identify custom objects with ease. This level of flexibility can greatly improve user experience, thereby increasing the commercial feasibility of the drone and letting the users to fine tune the capabilities of the vision system. Open sourcing the proposed platform lets collaborators and developers scale functionality as well as build custom products. Basic workflow of the video processing proposed, is shown in Figure 2.

3 Methodology

Central platforms for training and testing machine learning (ML) or deep learning (DL) models are provided by Microsoft Azure, Google Cloud Platform (GCP), Amazon Rekognition etc. Users can train new deep nets or ML models with new data and deploy on any of these services for inference. These services are often cloud based and do not allow user to seamlessly integrate ML or DL models to edge devices. Annotation of data, training/testing the model and moving it to an edge device, needs to be done separately and requires domain expertise to execute. Consumer products like UAVs or prosthetic hands cannot rely on this model because it demands the user to be a domain expert in order to explore full potential of the product they purchased. Our proposed architecture allows the user to effortlessly login to the platform, create annotated data for the aforementioned use cases, train the model for any particular edge hardware of their choice and deploy it within the device, all with the help of our guided click through user interface.

Proposed tool incorporates third-party services, but is not limited to them. For data storage Amazon S3 service will be used and model training is done on Amazon Web Service (AWS) servers. Users could either choose the proposed tool, available as freeware and choose these third-party solutions by opting to their costing structure or maintain storage and processing on local devices by downloading the entire platform to work offline. Although it is recommended to use the former method for ease of use, offline version of the tool is also provided to comply with user privacy and data security. Logging in to the platform enables users to upload custom image data. The platform will provide support for generating

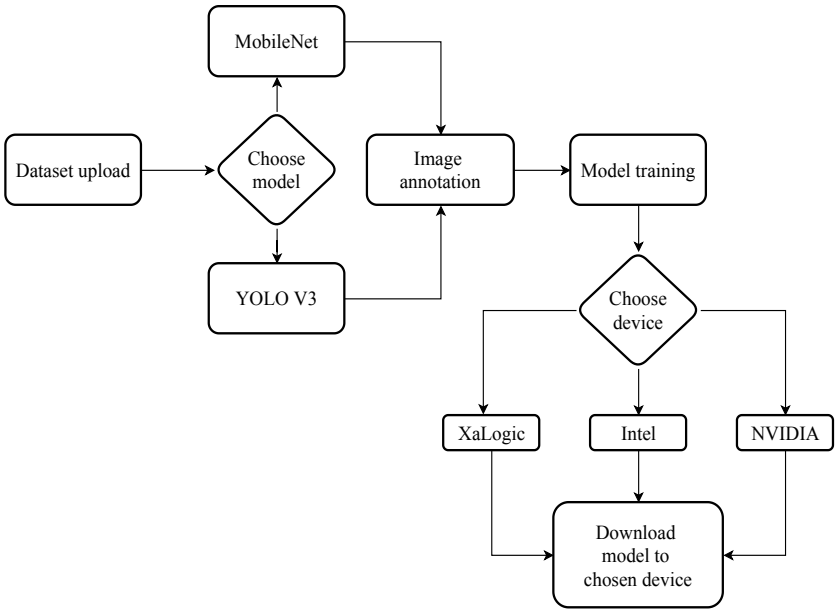

Figure 3: Workflow depicting proposed tools data flow and model generation.

two custom object detection models in the initial build – MobileNet [9] and YOLO V3 [2]. Users can choose their preferred edge device (Intel, NVIDIA or XaLogic) and the tool will generate inference graphs/models compatible in the chosen platform. Platform specific inference modules will be released through our GitHub repositories and can be offloaded to these devices. Pre-configured devices can be send to users on demand. Opting to use our online version of the tool, the users will be billed on an hourly basis in order to cover our expenses while using third-party solutions mentioned above. An abstract workflow is illustrated in Figure 3.

Users can compile an image dataset to recognise custom objects. On uploading the data to our online tool,and choosing a preferred object detection model, the user can start annotating the data. After completing the annotation, the user can proceed to the model training phase, where the annotated data is uploaded to an S3 bucket and our model training API in AWS gets triggered, thereby starting the model training. On completion of the training cycle, based on the user selected device, a hardware specific model file is generated and a link is sent to the user. In the offline version, user can acquire the model from a pre-defined location after the training is complete. Model inference/deployment can be done by just copying the model to the hardware of choice, which has been pre-configured with inference modules available through our GitHub repositories. Pre-trained models for use cases discussed here will also be made available through our Git repo along with steps to re-train them using the tool.

4 Conclusion

Computer vision solutions developed with AI are often cumbersome, slow and has large storage footprint. This drastically affects their real time performance and makes them unsuitable for commercial use cases. Proposed tool can greatly improve the ease of developing custom computer vision solutions for edge processing on hardware provided by Intel, NVIDIA or XaLogic. Use cases discussed in this work (vision powered prosthetic hand and drone vision), will serve as the testimony for real world problem solving capabilities of the tool. Users are given full access to the tools source code and complementary inference modules for use cases discussed in this work. Tool can be integrated to new/existing application as per users choice and crowd sourced development can extend functionality of the tool. AI inference on the edge will allow anyone to integrate visual analytics and automated insight generation for decision making in portable and mobile robotic platforms.

References

[1] Al-Timemy A H, Bugmann G, Escudero J and Outram N. 2013 Classification of finger movements for the dexterous hand prosthesis control with surface electromyography *IEEE J. Biomed. Health Inf.* 17 608–18

[2] Andrew G. Howard and Menglong Zhu and Bo Chen and Dmitry Kalenichenko and Weijun Wang and Tobias Weyand and Marco Andreetto and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications *arXiv eprint arXiv:1704.04861*, 2017.

[3] Došen S and Popović D B 2011 Transradial prosthesis: artificial vision for control of prehension *Artif. Organs* 35 37–48

[4] Došen S, Cipriani C, Kostić M, Controzzi M, Carrozza M C and Popović D B 2010 Cognitive vision system for control of dexterous prosthetic hands: experimental evaluation *J. Neuroeng. Rehabil.* 7 42

[5] François Chollet et al. Keras.https://github.com/fchollet/keras, 2015

[6] Shoaib Ehsan and Klaus D. McDonald-Maier. On-board vision processing for small uavs: Time to rethink strategy,2015

[7] Farina D, Jiang N, Rehbaum H, Holobar A, Graimann B, Dietl H and Aszmann O. 2014 The extraction of neural information from the surface EMG for the control of upperlimb prostheses: emerging avenues and challenges *IEEE Trans. Neural Syst. Rehabil. Eng.* 22 798–809

[8] Ghazal Ghazaei et al 2017 *J. Neural Eng.* 14 036025

[9] Joseph Redmon and Santosh Divvala and Ross Girshick and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection *arXiv eprint arXiv:1506.02640*, 2015.

[10] Mahbubul Alam and Manar D. Samad and Lasitha Vidyaratne and Alexander Glandon and Khan M. Iftekharuddin. Survey on Deep Neural Networks in Speech and Vision Systems *arXiv eprint arXiv:1908.07656*, 2019.

[11] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[12] Nazarpour K, Cipriani C, Farina D and Kuiken T. 2014 Advances in control of multi-functional powered upperlimb prostheses *IEEE Trans. Neural Syst. Rehabil. Eng.* 22 711–15

[13] Oskoei M A and Hu H. 2007 Myoelectric control systems—a survey *Biomed. Signal Process. Control* 2 275–94

[14] Reiter R (1948) Eine neue Elektrokunsthand. Grenzgeb Med 4:133–135

[15] Tippetts, Beau J. "Real-Time Implementation of Vision algorithms for Control, Stabilization and Target Tracking, for a Hovering Micro-UAV," Master of Science Thesis, Brigham Young University, USA, 2008.

[16] Hinton, Geoffrey E. and Osindero, Simon and Teh, Yee-Whye. "LeNet-5, convolutional neural networks," November 2013.