Deep Learning Systems for Identifying and Localizing Objects in Real-World Environments

Nicholas Sagers[1], Juan Andrade[2], Robert LiKamWa[1,2], and Andreas Spanias[2]

[1]School of Arts, Media and Engineering (AME), Arizona State University
[2]SenSIP Center, School of ECEE, Arizona State University

***Abstract*—The intersection of deep learning and mobile/embedded systems has received significant attention in recent years. Development in the areas of deep learning and mobile architecture has made possible an entirely new ecosystem of technological solutions. We propose one of those solutions, a mobile system to predict the whereabouts of objects in the environment. Using recently developed techniques, we are able to efficiently utilize computer vision and mobile processing to assist in the task of interacting with the natural environment. This solution provides a means of assisting those with memory impairments and those without memory impairment alike.**

***Index Terms*—accessibility, augmented reality, computer vision, mobile, convolutional neural networks**

I. Introduction

New advanced object detection and natural language processing techniques have allowed for a plethora of more reliable and better performing technologies, such as energy requirement reduction via selective image processing leading to smart cameras in mobile and embedded systems. [1, 2]. These innovations are foundational for future applications that involve solving human accessibility concerns and more naturalistic human-machine interfacing. Existing technologies have utilized object detection for security, automation, entertainment, and natural language processing for hands-free interaction and real-time translation. We propose a system to improve the quality of life of both memory impaired and non-memory impaired persons through the use of deep learning techniques and mobile systems.

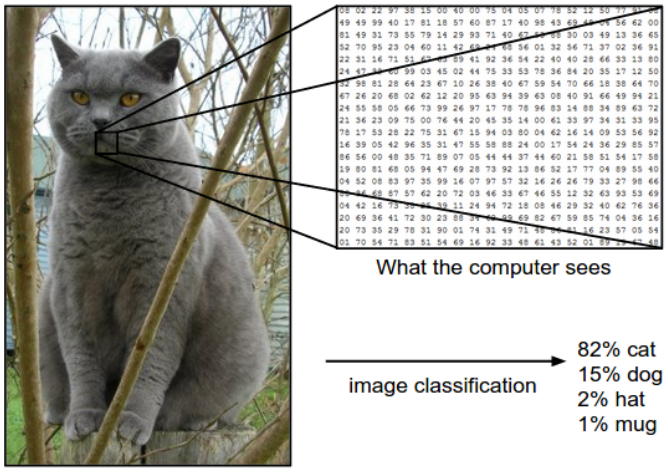

Figure 1. Image Classification

II. Motivation

Care of elderly and disabled persons remains a critical challenge to this day. However, not all elderly and disabled persons require total and constant supervision, as in the case of a nursing home. Many simply need help with day-to-day activities that require assumed skills. Specifically, tasks of interacting with the immediate environment pose a great obstacle for unassisted living. Tasks such as holding and using objects, remembering in-progress tasks, and locating objects in the immediate environment are all activities that can cause independent or semi-independent living to become overly difficult. The Real World Search Engine (RWSE) aims to ease the burden of independent living by allowing users a way to "search" for things they need, and in doing so addressing two concerns of independent living: object location and object interaction.

This project is incredibly useful for those who wish to remain in independent living, but are heavily burdened by the task of identifying and interacting with the objects around them.

III. Requirements

In order for the project to be successful, it must provide a solution to a multitude of challenges and seamlessly integrate these solutions into a feasible package for the end user. Object identification must occur in real-time, defined herein as a lack of noticeable delay by the user, as too must data transfer between the client and server. The client must efficiently deliver (camera) sensor data to the server in order to generate valuable information for the program and the user. The visual user interface must be accessible to all manner of users, specifically those who would most benefit from the Real World Search Engine.

IV. Implementation

The Real-World Search Engine utilizes the Microsoft HoloLens in order to provide visual feedback to the user and as a source of data for a network connected, server hosted neural network. The HoloLens is a relatively recent hardware development that has numerous, unexplored uses in industry and accessibility systems [3]. Techniques such as down-sampling [4] and motion estimation will be used in the future to ensure a positive user experience in terms of energy performance, functionality, and reliability. Object identification will be carried out by YOLOv3 via the Darknet frame work [5]. This project will focus on the application

Figure 2. The Microsoft HoloLens

of these models, the feasibility and implementation of deep learning tasks on low powered devices (Microsoft HoloLens), and future applications of this system. The future challenges of the project, outside of design and testing, are meeting of the power and bandwidth requirements, as well as further expandability with other accessibility tasks.

Figure 3. Prototype example view of Real World Search Engine.

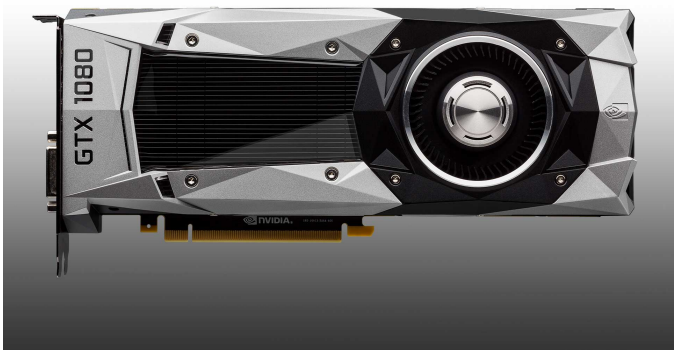

Figure 4. NVIDIA GTX 1080 graphics card used for image classification

V. Preliminary Results

Although much of the project so far has been developing the platform for the Real World Search Engine, we have been able to perform initial testing on its image detection capabilities so far. Currently, the Real World Search Engine is able to identify objects in the environment, report those finding back to the end-user via visual cue, and selectively screen for certain objects based on user input. The Real World Search Engine is able to perform these tasks in real-time with moderate power usage client side.

We started the project using OpenCV's deep learning module (dnn) via Python 3. Unfortunately, the deep learning module from OpenCV does not utilize discrete graphics card resourced. Instead, it relies on the central processing unit for image classification. With OpenCV, image classification occurred at roughly one frame per second. In order to process image data in real time we needed to utilize GPU resources. For this reason, we decided to switch to Darknet for our image classification framework. Darknet allows us to utilize not only GPU resources, via NVIDIA CUDA and cuDNN, but also OpenMP for program multiprocessing and OpenCV for frame processing on buffer images [5]. OpenCV support through Darknet allowed us to work on frame data stored in memory rather than saved to physical storage, dramatically improving performance.

With Darknet, and the above features utilized, we were able to process frames at a rate of approximately 10 frames per second on a single NVIDIA GeForce GTX 1080 graphics card. While performance can always be improved, this was more than adequate for a proof of concept for the Real World Search Engine's server component. In order for this aspect of the project to meet deployment performance needs we need to increase the frame calculations per second. This goal would most likely be achieved by supplementing additional graphics cards to the server or by even leveraging client hardware to assist in the task.

VI. Project Challenges

The most time consuming aspect of the project, so far, has been setting up the server environment. Many of the tools we wanted to utilize required significant time to diagnose and treat software deprecation and incompatibility. Additionally, identifying preexisting tools to assist in the program and customizing them to the program's exact needs was more of a challenge than previously thought. Before arriving at Darknet, we tried Caffe2, PyTorch, and Tensorflow. For each of these frameworks there was a period of testing and experimentation that ultimately led to shifting to a new framework that was more inline with the project goals. Migrating to Darknet allowed us to utilize CUDA, cuDNN, OpenMP, and OpenCV by simply modifying values in a makefile. The ease at which we could modify Darknet to suit our task allowed us to spend time elsewhere.

VII. Future Developments

Short term project goals for the Real World Search Engine include faster client/server network operations and server image processing, using image compression techniques to reduce the amount of data transferred between the client and server, and reducing client power requirements via selective frame capturing and other techniques. Along with these algorithmic