Project ID: 26767

Developing an Automated Microscope Integrated with Deep Learning Postprocessing to Conduct Preliminary Cancer Diagnosis

A Major Qualifying Project Report submitted to the faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Electrical & Computer Engineering

April 26, 2018

Prepared by:

Jamison LaRoche

Brian Westgate

Presented to:

Professor Kwonmoo Lee, Ph.D., Advisor
Department of Biomedical Engineering
Affiliated with Electrical & Computer Engineering

Acknowledgements

The authors would like to thank University of Massachusetts Medical School for providing information and access to samples. In particular, Dr. Ali Akalin and Dr. Young Kim provided valuable insight into what requirements and purpose the prototype could fulfill. The authors would also like to thank Jason McKinney for his advice and assistance with additive manufacturing. All of the faculty and staff at Worcester Polytechnic Institute deserve thanks for helping to make this project possible. Thank you to Bill Appleyard for assistance in acquiring parts. Thank you to Xitong Zhang, Chauncey Wang, and Bolun Lin for assisting with deep learning development. The authors would like to also thank Professor Kwonmoo Lee for serving as project advisor. His help and support throughout the course of the entire project was invaluable.

Abstract

Cancer is a disease that affects nearly 40% of the United States population. The current diagnosis process can take several days, and a highly trained pathologists must allocate time to observe and classify all samples collected from biopsies. Occasionally, biopsy samples do not even have enough cellular material to make a proper diagnosis, and the patients must undergo a secondary procedure before a confident cancer diagnosis can be composed. This project aims to accelerate the process by using an automated microscope outfitted with image-processing artificial intelligence to complete a preliminary stage of diagnosis to determine if adequate amounts of cell clusters have been collected from the procedure. A prototype integrated with regional convolutional neural networks was constructed that successfully images microscope slides and places bounding boxes around cell clusters found in fine needle aspirations. This device is expected to greatly assist pathologists in providing faster more accurate diagnoses. This prototype also serves as a development tool that can be easily configured with different neural network architectures, and be adapted to employ different imaging techniques.

Executive Summary

Introduction

There are many existing technologies that support the diagnosis of cancer. Pathological studies, however, remain among the most common ways of detecting cancer, and are often crucial for an accurate diagnosis. In many cases, a tissue sample, or biopsy, is required to determine if a patient is positive for particular diseases. A fine needle aspiration is a procedure where a needle and syringe are used to extract soft tissue or fluid from the body, and is commonly used to retrieve thyroid samples for cancer diagnosis.

Once the samples have been stained, they can be viewed under the microscope by pathologists to identify any specific characteristics or abnormalities. A pathology report is created for each specimen tested in the lab, and results are added to other medical records in order to make a proper diagnosis. The microscopy portion of a pathology report assesses size, shape, organization, mitotic rate, and staining characteristics of cells. The pathologist determines grade, stage, tumor margin, whether the cancer is invasive, and if the cancer is in lymph nodes all within the pathology report [1]. A wealth of knowledge is collected from these histopathological inspections and used to support diagnosis and treatment methods. This process currently takes between 2 and 10 days for the pathologist to prepare the sample and take a diagnosis [2]. The diagnosis could take even longer if the sample taken does not have enough data to make an accurate diagnosis, thus possibly delaying the necessary treatment if the patient has cancer.

In order to expedite this process, medical professionals believe that artificial intelligence may be used on site in order to see if the sample is valid, and conduct a preliminary diagnosis on

the sample. Artificial intelligence has been evolving over the years, and one form of artificial intelligence that seems promising for this application is a Region-Based Convolutional Neural Network (RCNN). This is a form of machine learning which overlays a support vector network (also known as support vector machine, or SVM) in order to classify the image [3]. This works similarly to a line of best fit, where the input variables include contour vectors and colors.

Design Approach

The core design requirement for the automated microscope was to autonomously image a microscope slide and reliably identify and quantify cell clusters. Additional requirements include having a compact size, fast scanning and processing time, changeable imaging equipment, low cost, and ability to be run without operation by a trained specialist. The imaging feed from the device had to also be processed on site, as opposed to cloud computing. These requirements were recognized by designing a three axis gantry system capable of focusing and moving the entire area of a standard microscope slide beneath a camera.

Final Prototype

The frame of the prototype was constructed from 10x10mm extruded aluminum, which provides solid support and convenient mounting locations. The volume of the device is 200x250x250mm. The gantry mechanism was implemented by employing three sets of 8mm diameter lead screws and linear rails. The lead screws are powered by NEMA 17 stepper motors. An Arduino outfitted with a 3 axis GRBL driver shield is used to drive the motors. The Arduino receives G-code commands from Python programs via serial port connection to control the X-Y motion of the microscope slide holder and controls the Z axis to focus the images. A 12V, 240W power supply provides power to the motors through the shield. A CMOS digital microscope is

utilized in this project to capture images at an approximate 10x magnification. This microscope camera was mounted to the moveable platform above the slide area. Images from the USB microscope are sent to and saved in a Nvidia Jetson. The Jetson executes the motor control code and processes the images within the neural network.

Training and Testing:

Functionality of the prototype was tested by training the neural network with preliminary samples, and verified by testing new samples. The pathology division at University of Massachusetts Medical School provided biopsy specimens from fine needle aspirations. These samples were collected to test patients for thyroid cancer. Ten slides were scanned using the device, and employed for training and initial verification. All of the images were visually inspected for cell clusters, and coordinates of the boxes input into the training algorithms of the region-based convolutional neural network. The neural network is trained through back propagation, by passing forward and back images through many convolutional layers in order to extract specific characteristics associated with the desired regions. The neural network is optimized or trained by reducing the error in epoch after epoch, as the desired outcomes are predetermined. 20 percent of bounded samples were not included in the training sets and used to validate accurate identification of cell clusters by the neural network. After the neural network had been trained, it was loaded on the NVIDIA Jetson to provide immediate identification of cell clusters on newly scanned samples.

Results

The team successfully produced an automated microscope with an artificial intelligence post-processing. The device takes approximately two and a half minutes to scan an entire

microscope, and an additional three minutes to process the images through the artificial intelligence. If internet connection is available, the device will upload the files to Google Drive, however the speed of that is limited by the internet connection in the current area of operation.

The artificial intelligence is able to identify clusters of cells in many different shapes, sizes and colors, however it struggles with smaller or lighter cell clusters even though they were included in the sample. This is adequate for the device's desired operation. A secondary segmentation deep learning architecture that was also investigated, which demonstrated improved performance in identifying clusters.

Future Work

This project is the first step towards what could be a possible breakthrough in the field of medicine. The artificial intelligence can be adapted to identify any kind of cell (stained or unstained) given the proper camera and magnification. This could identify cancer cells, sickle cells, and anything you are able to place under a microscope.

This project could be enhanced by outfitting the prototype with a secondary camera of higher magnification. This allows the cell clusters to be identified more easily by the low magnification camera, allowing for faster and more accurate analysis This would support much more analysis at the cellular level, and support greater confidence in classification.

Additionally, more training data for the artificial intelligence postprocessing is always beneficial. Future work could implement training on unstained samples so that the staining process can be omitted altogether, allowing for immediate diagnosis. The mechanism that drives the automated microscope could also be improved by having its size decreased and using quieter

motors. Additionally, if a camera with a high enough framerate is used, the device will not have to stop to take a picture and instead run continuously.

Table of Contents

Table of Figures

Table of Tables

Glossary of Terms

Artificial Intelligence (AI) - Algorithms that can complete human tasks

Neural Network (NN) - Computing structure composed on many convolutional layers to extract specific features

NVIDIA Jetson - Embedded supercomputer with sufficient GPU

Regional Convolutional Neural Network (RCNN) - Neural Network trained through backpropagation to place bounding boxes about identified features

Chapter 1: Introduction

There exists a great need for fast and accurate cancer diagnosis. The National Cancer Institutes that approximately 1,685,210 Americans will be diagnosed with cancer each year [4]. Millions more are screened each year, as it is important to find and treat cancer as early as possible. Though, cancer diagnosis is a very intensive and monotonous process. Patients may not receive the results of biopsies for weeks because each must be handled by a pathology lab and evaluated by experienced professionals. Additionally, if adequate specimens were not collected from the biopsy, patients will have to undergo the procedure a second time, delaying possible treatments for those that have cancer. The evaluation time and secondary procedures place a significant burden on hospitals and health centers as well.

This project focused specifically on improving the evaluation process for thyroid cancer. Patients who may be positive for thyroid cancer undergo a fine needle aspiration (FNA) to extract a biopsy from the thyroid. A large needle is inserted into the neck, and cellular material from the thyroid extracted. These samples are smeared onto microscope slides and sent to pathologists for staining and evaluation. This process can be improved by having a device to evaluate slides on the spot to determine if enough cellular material has been collected to make a proper diagnosis.

A prototype automated microscope was developed in order to complete a pre-diagnosis of stained fine needle aspirations. The device was programmed to scan the entire area of a microscope slide, and used deep learning in order to identify cell clusters in the collected images. This quickly informs doctors if enough cellular material has been collected, and the locations of identified clusters.

Doctors are often hesitant to rely solely upon Artificial intelligence for cancer diagnosis, which is why this device simply offers a preliminary stage of processing of cell cluster identification. In the future, the device could easily be adapted to include a secondary lens of higher magnification to image at the cellular level, and be trained to recognize cells exhibiting cancerous traits. Hesitation to rely on artificial intelligence in the medical community stems from the inability to conceptualize the specific algorithms and results that a neural net may return. Though, certain standards such as ISO/IEC JTC 1/SC 42 on Artificial intelligence exist which can support for greater confidence in processing that relies on neural networks [5]. Meeting these certifications means that serious consideration has been made for network architecture and computational methods, big data, societal concerns, and trustworthiness. Implementing systems equipped with artificial intelligence often require additional translation and communication about factors that contributed to decisions made.

The use of deep learning within the medical field is a very recent proposition, and FDA just approved the first diagnostics process relying upon artificial intelligence over the course of this project [6]. On April 11, 2018, the FDA approved a device call IDx-DR that screens and diagnoses patients with diabetic retinopathy. This is the first device that diagnoses a medical condition using artificial intelligence without relying upon a clinician to look at images or data from the patient [7]. This is a major development for the artificial intelligence market, and medical devices equipped with artificial intelligence are on the cutting edge of technology.

Chapter 2: Background

Presentation of Cancer

Cancer is the given name for a collection of over 100 diseases in which mutated cells divide uncontrollably and may invade neighboring tissues [8] [9]. The mutation in a cell is occurs when the DNA is unintentionally modified during cell division. These mutations can be a predisposition of genetics, however it is more commonly caused by exposure to a carcinogen [10]. Carcinogens are a class of substances which "are directly responsible for damaging DNA" [1].

There are four key types of genes that may lead to cancer when mutated. These are: oncogenes, tumor suppressor genes, cell suicide genes, and DNA repair genes [1]. The oncogenes contain instructions that tell the cell when to divide. Tumor suppressor genes contain instructions that tell the cells when not to divide. Some genes control apoptosis, also known as programmed cell death, thus mutations to these genes can remove the pre-defined cell control instructions normally contained in these genes [11]. The fourth type of gene is the DNA repair gene, which instructs a cell when to repair damaged DNA. Mutations to any of these four types of genes can lead to the rapid reproduction of cells, thus forming an invasive growth of cells called a malignant neoplasm [12], more commonly known as a cancerous tumor.

Due to the broad definition of cancer, the disease may present numerous symptoms depending on what kind of cell is mutated. However, most forms of cancer present themselves similarly on the cellular level [13]. Cancerous cells may be of irregular size and/or shape, as they do not function as intended and normal cells are specially shaped to help them carry out their designated task. This also affects the arrangement of cancerous cells, as many cell

structures are in place to aid with the typical function of the cells. The nuclei of cancer cells may be also abnormal- this is most commonly presented in the form of a darker and/or larger nucleus, as the nucleus of a cancer cell tends to contain more DNA than the average cell. These physical anomalies presented by cancer cells are what allow medical professionals to identify cancer cells optically.

Pathological Diagnosis of Cancer

There are many existing technologies that support the diagnosis of cancer. Pathological studies, however, remain among the most common ways of detecting cancer, and are often crucial for an accurate diagnosis. In many cases, a tissue sample, or biopsy, is required to determine if a patient is positive for particular diseases. The three main types of biopsies are excisional biopsies, incisional biopsies, and fine needle aspirations [13]. Excisional biopsies are tissue removals where an entire lump or affected area is removed, while only a section is removed during incisional biopsies. A fine needle aspiration is a procedure where a needle and syringe are used to extract soft tissue or fluid from the body.

The samples from fine needle aspirations can be smeared onto glass slides, but other types of biopsies must be treated before they can be examined under a microscope. There are two common strategies for slicing a specimen and preparing it for a histopathological examination. One method involves freezing the sample before it decays, and sectioning it using a device called a cryostat [14]. This is a relatively fast procedure, and used in situations such as surgery where results need to be promptly delivered. Quality of frozen samples can sometimes be impaired. If a cleaner slice is required, or if the specimen must be preserved, it can alternatively be treated with a fixative and then infiltrated with paraffin wax [13].

After samples have been reduced to single-cell thickness, they can be immunostained. The routine staining procedure involves using hematoxylin and eosin (H&E). H&E staining enhances the visibility of cellular structure and provides much greater detail to support diagnosis. Hematoxylin is a basic dye that colors the nucleus and organelles containing DNA or RNA a purple or blue. Eosin is an acidic dye that stains acidophilic parts of the cell such as the cytoplasm or cell walls pink or red [15]. The combination of these two dyes supports a much more comprehensive understanding of cellular structure within a biopsy sample. There are also a wide variety of special staining techniques that have been developed for specific types of cancer detection. Immunocytochemistry, for example, is often used to observe specific proteins and antigen in cells in order to identify tumor histogenesis and subtype [16]. This sort of information is used to tailor treatments to specific patients and cases.

Once samples have been stained, they can be viewed under the microscope by pathologists to identify any specific characteristics or abnormalities. A pathology report is created for each specimen tested in the lab, and results are added to other medical records in order to make a proper diagnosis. The microscopy portion of a pathology report assesses size, shape, organization, mitotic rate, and staining characteristics of cells. The pathologist determines grade, stage, tumor margin, whether the cancer is invasive, and if the cancer is in lymph nodes all within the pathology report [17]. A wealth of knowledge is collected from these histopathological inspections and used to support diagnosis and treatment methods.

Artificial Intelligence, Deep Learning & Image Recognition

In order to understand the purpose of an artificial intelligence-based system, you must first understand how it operates. Artificial intelligence (AI) is defined as “[t]he theory and

development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages" [18]. Modern day artificial intelligence applications fall under the concept of Narrow artificial intelligence, which is a technology that performs a specific task [19].

One subset of artificial intelligence called machine learning (ML) uses a algorithms to parse through large amounts of data, learn from the data, then makes a prediction or determination about the data with a certain confidence [19]. The "learning" in machine learning comes from its attempt to optimize the results by using algorithms that recognize successes and errors in the outcome [20]. These algorithms determine successes/errors by using predefined identifiers. machine learning is commonly used in the form of computer vision i.e. image recognition; however, this still requires coding additional identifiers in order for the machine learning to fully and accurately complete the program's objective [19].

In order to minimize the hard-coding needed and maximize the accuracy of the outcome, a structure called a neural network is used. This is based on a framework which manipulates inputs with different weight factors in order to get a better understanding of the implication of each input [20]. The artificial neurons are structured in layers, where once one neuron interprets the data, it sends it to the next layer of neurons. The program then draws conclusions based on the modified inputs, while a cost function determines if it the output is correct or not. At first the initial outputs are wrong and are compared to outputs that are known to be correct, such as imputed data that has been examined by a human and predetermined to have the desired output. This allows the program to measure how correct its outputs are, and it can modify the algorithm in an attempt to reduce errors. The accuracy is represented by a probability vector (i.e. a highly

educated guess), and this accuracy becomes greater each time data is passed through the neural network [19].

As more neurons are used in the neural network, the amount of data it can process increases thus increasing the speed at which the neural network learns. Deep learning is a subset of machine learning that refers to the use of neural networks with many layers of artificial neurons. Additionally, there can be multiple layers of these 'neurons,' which allows the AI to deepen its understanding. Each layer allows the neural network to devise both a larger quantity of and more precise concepts for identification [21]. The output of the first neuron becomes the input to the neurons in the next layer, and so on and so forth. This revolutionized the field of AI and machine learning, as instead of having one large recognition computation, the neurons divide the single complex algorithm into a series of much simpler calculations [21]. ""With traditional methods, the machine just compares the pixels. Deep learning allows learning on more abstract features than pixel values, which it will build itself" [21].

RCNN and Image Recognition

Regions with Convolutional Neural Networks, or RCNN, was the first stepping stone for image recognition using artificial intelligence. It was developed in 2014 by Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik [22]. By using a CNN to detect the image features, such as color intensities, within a designated region, they created an artificial intelligence which was able object recognition. This CNN accounts for the shapes and colors of the item in a given region, thus allowing a quantification of data and a way of pattern detection for the given region. These color intensities and patterns are processed through a Support Vector Machine (SVM) in

order to categorize certain shapes and color vectors [3]. An example of this would be detecting the relatively straight lines that the edges of a tree trunk has.

RCNN is designed to detect more than one kind of object at a time. This is what is called a class, and it can consist of just about anything- a dog, an apple, or an airplane. By using machine learning to train the artificial intelligence on different classes, RCNN has nearly limitless image detection possibilities. Some examples of RCNN in academia include facial recognition and detecting if someone is using their phone while they are driving [23].

Further research by the same team lead them to develop faster versions of RCNN, aptly name Fast RCNN followed by Faster RCNN [25] [26]. After Faster RCNN, some of the team members sought to further increase the processing speed of the RCNN, so they developed a new artificial intelligence called RetinaNet [27]. Though they initially wanted to trade accuracy for speed, the end product was both faster and more accurate than their Faster RCNN [27].

Automated Cancer Diagnosis

Recent advances in deep learning have opened new opportunities for advancement within the medical field [28] [29]. Pathological and cytological examinations are very monotonous and time consuming and require careful attention from highly trained physicians and technicians. These laboratory examinations are costly and time consuming, as many health clinics must outsource examinations to other departments, and place limitations on the level of care that can be provided. In response to these challenges, some industry leaders and researchers have been attempting to integrate automated imaging techniques into the healthcare system to speed up the diagnosis process and provide thorough and consistent pathological evaluations.

Automated cancer diagnosis is typically divided into 3 stages consisting of pre-processing, feature extraction, and diagnosis [30]. Preprocessing mainly consists of identifying which parts of a sample slide is background or white space and which areas must be examined in greater detail. A computer aided diagnosis study completed by researchers at MIT found that on average 82% of whole slide biopsy images were covered with white space [31]. Adequate focusing is required for the pre-processing and later stages in automated cancer diagnosis. There is a delicate balance between the magnification required to extract pertinent information from a sample and the processing time or memory required to image a specimen at high resolution. In many cases the pre-processing stage is crucial for determining which zones are significant and which can be ignored. A major challenge while competing this, and all imaging aspects of automated diagnosis, is that a clear and focused view must be extracted for detection. Typically this is accomplished using image contrast methods utilizing images collected by altering the distance between the samples with respect to the lens. However, other strategies exist to focus images at a much faster rate. UCONN's Smart Imaging lab has devised a technique using two pinhole modulated cameras to instantly detect focal planes [33]. Recurrent neural networks offer another alternative for processing images and instantly reacting to maintain focus [33] [34].

Feature extraction is the next step in completing an automated cancer diagnosis. Features can either be extracted at the tissue or cellular level. Different information relevant to diagnosis is presented at both the cellular level and wider assortment of cells. RPI researchers classified samples into five main categories consisting of morphological features, textural features, fractal-based features, topological features, and intensity-based features when attempting to build

a program to automatically diagnose cancer [30]. One major challenge other than extracting these features is weighting them and determining confidence margins. A paper released by researchers at Villanova and University of Pennsylvania specifically focused on image retrieval and classification of thyroid cytopathology [35]. Features used to classify FNA biopsies as either benign or malignant included looking at cell patterns, cellular characteristics, nucleus size and shape, and background material [35]. Some observations that fall into these categories include colors, cellular density and position, and roundness. One major challenge that complicates classification and identification of cellular characteristics though, is that there are many different techniques for preserving, staining, and viewing biopsies. Any automated cancer diagnosis system must be thoroughly trained to handle specific types of cancer and treatment methods.

The culmination of the preprocessing and feature extraction stages is the final cancer diagnosis and communication of results. There are a multitude of ways to evaluate the data collected by these imaging techniques, but the two most common include statistical analysis and machine learning [30]. The statistical approach consists of drawing together information from feature extraction phase and using complex equations to determine the probability of cancer. The more recent and promising approach involves using neural networks to diagnose. This method involves training a system to use deep learning to identify cancer. Numerous samples are processed by the neural network such that it can "learn" what each variation of healthy specimen, benign abnormality, or malignant cancer may look like. Each result is classified as either true positive, false positive, true negative, or false negative in order for the network to reconfigure weightings and reduce error [30]. This method yields a confidence margin that indicates the likelihood of cancer. The MIT research group developed an evaluation system that delivered

very high confidence margins and proved to be very accurate. It was noted in their paper that when the automated cancer diagnosis was paired with that of a physician's pathology examination, the overall accuracy in diagnosis was much higher than both parties independent [31]. This supports the concept that the results form automated cancer diagnosis could prove invaluable to the medical industry.

Chapter 3: Goals and Specifications

Customer Requirements

Through discussion with the project advisor, the MQP team devised the initial customer requirements for the deep learning based imaging system for point-of-care cancer diagnosis:

- Inexpensive
- Automated process
- Compact & portable
- Reliable & consistent results
- Fast operation

Inexpensive to build:

The customer requests that the device is inexpensive to manufacture, as it would make the device more commercially accessible. The ideal price point for this product is less than $700, however due to the scope of this project the team will simply be keeping costs to a minimum.

Automated process:

The customer requests that this device should be simple enough for average medical personnel to operate. This requires the data collection to be automated. The request of the customer is to use a lens system with a camera to take pictures of blood samples, so this must be done in an automated fashion using artificial intelligence.

Device must be compact:

As the device will be point-of-care, it must be small enough to easily transport between different locations.

Reliable & consistent results:

It is critical that the cell count of this system is accurate, reliable, and consistent as a slide with an insufficient sample will cost the patient and the hospital thousands of dollars.

Fast operation:

The device must be able to quickly produce the cancer diagnosis of the inputted sample. This time must be approximately 5 to 10 minutes.

Design Concepts

After consulting with Dr. Ali Akalin and Dr Young Kim at UMASS Medical School as well as the advisor, the team developed the initial plans for the design. All of the design requirements were assess, and attempted to be met in the fullest extent. The block diagram for this design can be seen in Figure 1 on the following page.

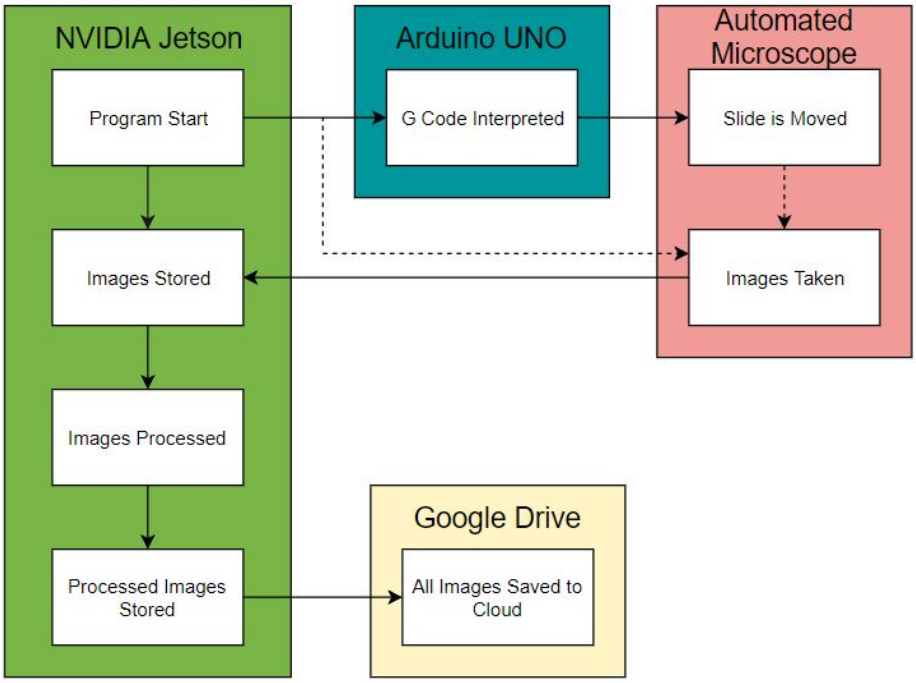

Figure 1: Control Block Diagram

As requested by the advisor, the team used the NVIDIA Jetson TX2 as the base control system for the entire device. The NVIDIA Jetson was chosen as it allows for a compact yet very powerful system for artificial intelligence implementations. It has the following specifications [39]:

- GPU: NVIDIA Pascal with 256 CUDA cores
- CPU: 2.0 GHz
- Memory: 8GB
- Storage: 30GB

This allows the team to have a single device to both control the automated microscope as and use artificial intelligence to process the collected data. This is important as having a single device capable of all allows the system to be transported easily, which is requested by the customer.

Chapter 5: Methodology and Design

Mechanical Design Considerations

Many possible designs were evaluated while determining what method was best for creating an automated system to image a microscope slide. The concept of a three axis system was decided because the microscope required an X and Y axis to scan the entire area of the slide, and a Z axis to focus the images. The proposed microscope has the X and Y axes mounted together on the base of the device. This is the case because it allows complete left and right, forward and back, motion of the test bed carrying the slide beneath a "fixed" camera and lens position. The Z axis holds the microscope lens and camera, and is capable of moving in the up and down relative to the slide test bed. The axes were isolated as such in order to minimize shake, to preserve high quality images. The base material used for the frame is 10mm by 10mm beams of MakerBeam extruded aluminum. These are T-slot pieces of aluminium which provide stability and ease of connectivity. The mechanical design was created using the 3D modeling software Solidworks. A rendering of the model can be found in Figure 2 on the following page.

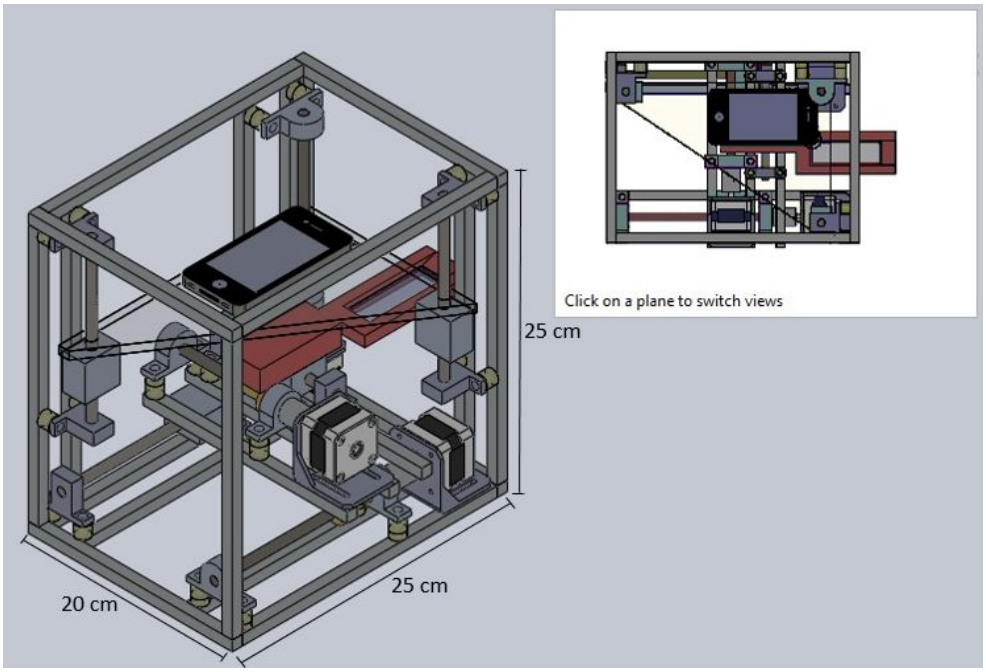

Figure 2: Solidworks Model of Automated Microscope

Mechanical motion of the axes is completed through the use of NEMA 17 stepper motors, T8 lead screws, and anti-backlash nuts. The chosen motors are suitable for the task because they have 26 Ncm of torque and support 200 steps per rotation. NEMA 17 is also a standard mounting size, which simplifies the design. The axis are moving very small loads, so the 26 Ncm of torque is sufficient. The microscope requires very fine incremental control, so the the 200 steps per rotation is also required. When paired with a T8 lead screw and 8 increments of microstepping capability from the Synthetos driver shield, each axis can attain a resolution of 5 microns. This is far more precise than even needed for this application. Additionally, the anti-backlash nut is required in order to prevent slippage, and guarantee that the nut always returns to the expected location after moving back and forth. Limit switches were not included with this design, but could be placed at the ends of the axis to detect when each axis has returned home. This would ensures that numerous trials are consistent with each other. Linear rails are

used opposite the lead screw on each axis in order to support smooth and stable motion.

A Synthetos gShield V5 Driver Board was chosen due to its incorporation of 3 axis and integration with GRBL Arduino. The Synthetos shield was attached to an Arduino Uno, which was flashed with GRBL. GRBL allows for the arduino to interpret G-Code over serial port connection. G-Code is a numerical control language that is commonly used in computer aided manufacturing. The G-Code allows for configuration of speed, acceleration, and travel distance for the motors. The shield also supports up to 8 microsteps and can regulate power delivered to each motor.

The Alunar ALSP20A05 12V 20A 240W Power Supply was chosen at it meets the requirements to power the Synthetos Shield as well as the Nema 17 motors. The input voltage required for the shield driver is between -0.5 and 29V, so 12V is sufficient. From the shield, each motor will draw 0.4ARMS and 12V, for a total system required power of approximately 21W. Though 240W is more than necessary, this particular power supply costs less than others and has easy-access outputs for development purposes. The jetson relies upon its own independent power supply.

The microscope used in this project was a Monoprice USB digital microscope. The microscope has a CMOS sensor and a maximum resolution on 1600x1200 pixels. The effective viewing area of the microscope was approximately 4x6mm. In order to image an entire slide area, not including patient label, 54 photos had to be taken. The test bed for the slide will also be a custom part and include a holder for the microscope slide. Below the slide there will be a LED lamp. The light source is covered with a thin white plastic material in order to adequately diffuse the light and illuminate the sample.

Neural Network Development

For the artificial intelligence used for the device, the team needed to have an image-based deep learning system. The two choices available are a regional convolution neural network (RCNN), or a segmentation-based artificial intelligence. The team decided to use an RCNN, as it would be faster to process and have potential uses for future works. If an RCNN is used to box potential areas, it would allow for a segmentation artificial intelligence to process only in the designated areas, thus decreasing the total processing time for the segmentation artificial intelligence.

The chosen RCNN was Keras RetinaNet [36]. This is a Keras-based RCNN which uses a Tensorflow backend. It is based off of RetinaNet [27], however it uses Keras instead of Caffe. The team chose to use a Keras-based system as the lab the project was based in has been using Keras for their other projects and is familiar with how it works [37]. Tensorflow was chosen as the backend over Theano for this same reason [38].

The training method chosen was region-based Pascal VOC, as the team was familiar with this particular method. In order to train the RCNN, the team had to acquire image scans of cell samples and designate the regions of interest on each image. This was done by using the photo editing software GIMP to record the pixel coordinates of the corners of each region. By recording two points (upper right and lower left), the RCNN is able to analyze and train on a rectangular region of the given image.

The dataset that the artificial intelligence was trained on consisted of no more than three significant images taken from a twenty-four slides. Each slide was comprised of fifty-four images, and not every slide had visible cell clusters on three slides. Some had over twenty

images with visible cell clusters, while others had one or zero images with visible cell clusters. It is important to note that on samples of patients with cancer, there were significantly more cells. This means that there is more training data on cancerous cell clusters than there are with benign cell clusters. In total, there were 854 cell cluster boxed for training the RCNN. This process took approximately 30 man hours.

In order to process the data through the RCNN, the team needed hardware which has a powerful graphics card. The device chosen was the NVIDIA Jetson TX2, as the customer requested it. It has a very powerful NVIDIA Pascal GPU with 256 CUDA cores, 8GB of memory and up to 2.0GHz processing speed [39]. Though the NVIDIA Jetson is optimized for processing neural networks, it does not have the CPU required to train the artificial intelligence. Therefore, the team used an Ubuntu machine with an NVIDIA Titan X GPU card to train the neural network. This device was designed for the use of cell cluster prediction using artificial intelligence, and its compact size makes it ideal for our system.

As the customer gave the team the NVIDIA Jetson TX2 Development Kit to use, it came loaded with not just GPIO but a full operating system environment. This is ideal for the team, as the programming for the motor control and artificial intelligence can be loaded and tested on the system using Python.

Once the prototype was constructed, training samples were collected in order to teach the the neural network to extract and recognise cell clusters. The pathology division at University of Massachusetts Medical School provided biopsy specimens from fine needle aspirations. These samples were originally collected to test patients for thyroid cancer. Ten slides were scanned using the device, and used for training and initial verification. Since the purpose of this network

was to simply identify cell clusters, not all of the samples scanned were positive for cancer. Five subjects had benign colloid nodes, two had papillary thyroid cancer, and three had lymphocytic thyroiditis. A maximum of 3 images of the 54 from each scan were selected for inclusion in the training set. All of the images were visually inspected for cell clusters, and coordinates of the boxes input into the training algorithms of the region-based convolutional neural network.

The neural network is trained through back propagation, by passing forward and back images through many convolutional layers in order to extract specific characteristics associated with the desired regions. The neural network is optimized or trained by reducing the error in each iteration and epoch after epoch, as the desired outcomes are predetermined. 20 percent of the slides were not included in the training sets and instead used to validate accurate identification of cell clusters by the neural network. After the neural network had been trained, it was loaded on the NVIDIA Jetson to provide immediate identification of cell clusters on newly scanned samples.

Chapter 6: Results and Analysis

Prototype Operation

The user must first plug in and power on the NVIDIA Jetson. The USB from the Arduino Uno must be plugged into the USB3 port on the NVIDIA Jetson Development Kit, and the USB camera must be plugged into the USB2 port on the NVIDIA Jetson Development Kit using the MicroUSB/USB connector. Then, the Arduino shield must be connected to the Arduino Uno, and the automated Microscope is then to be powered on. Ensure that the NVIDIA Jetson is connected to the internet via ethernet or wireless connection. At this time, the microscope slide of the desired sample may be placed into the slide holder. If the slide has an information sticker on it, it must be placed on the side of the microscope holder which faces outwards. After waiting one minute for the NVIDIA Jetson to boot up, the button S2 (volume down) on the Development Board must be pressed. After waiting an additional five seconds, the button S3 (recovery) must be pressed. After a delay of approximately twenty seconds, the automated microscope will begin to take pictures of the slide. The processing time for each step can be seen in Table 1 below:

Step	**Time to Process**
Image Microscope Slide	~2:30 minutes
AI Post-Processing	~3:00 minutes
Upload to Google Drive	Internet Dependent (~6:00 minutes)

Table 1: Device Operation Time

The speed at which it does this is dependant on the internet speed of the current location of the device, however you are able to check the progress by watching the files upload to your Drive, and after all fifty-four images are uploaded, the process is done. If you wish to run another

sample, you may load another slide and press the button S3 (recovery) to repeat the scanning process.

Results Overview

After training, the artificial intelligence performs adequately. The device was taken back to UMASS Medical School and tested against five new samples. Three subjects had benign colloid nodes, and two had papillary thyroid cancer. Though it consistently identifies clear cell clusters, it struggles to pick up on lighter colored or less defined cell clusters as seen in Figure 3. Additionally, it struggles with oddly shaped cell clusters, often putting multiple boxes around the cell or segmenting the cell into multiple sections as seen in Figure 4. These issues can be solved by acquiring more data to train the artificial intelligence with. Even with these issues, the artificial intelligence performs well given the relatively small data set it received.

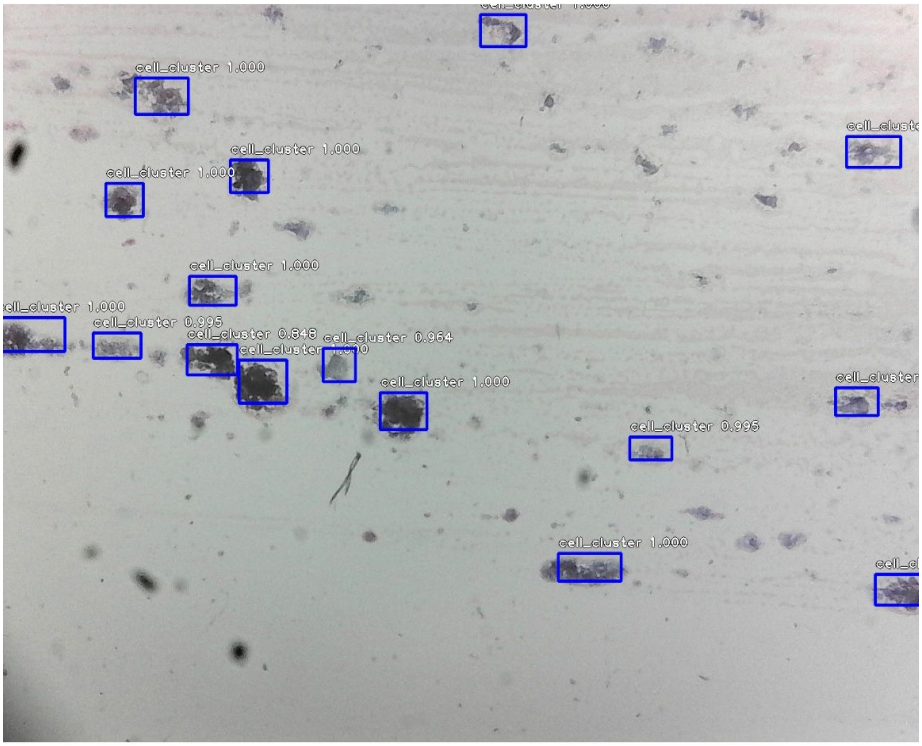

Figure 3: Testing Results Image A

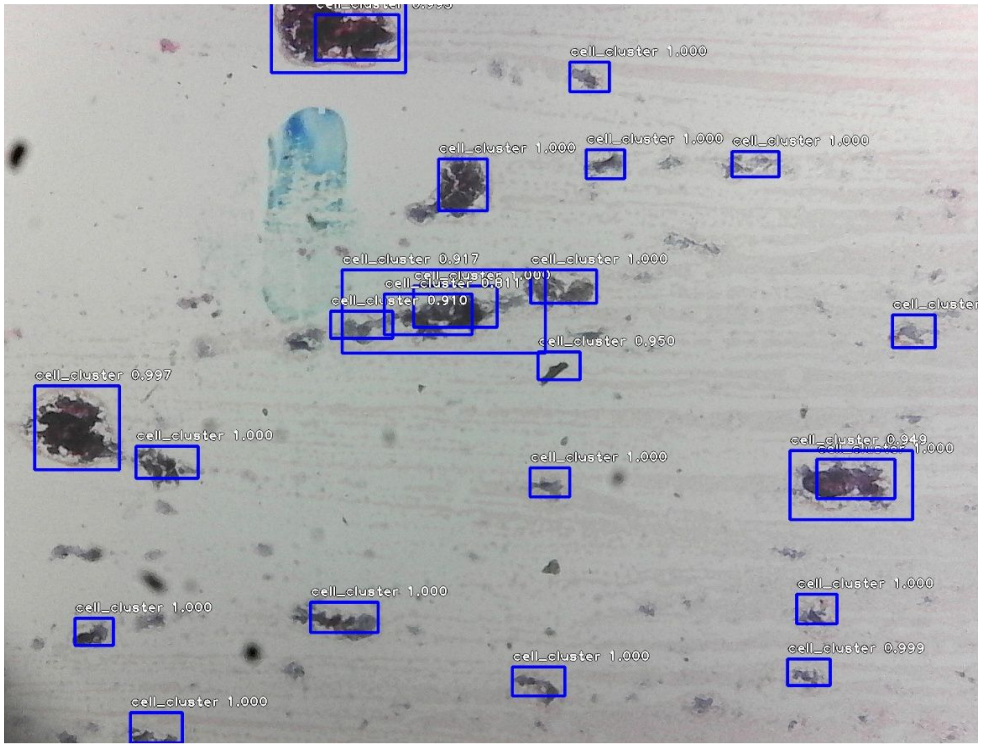

Figure 4: Testing Results Image B

Analysis

The R-CNN was trained using 726 bounding boxes found in 8 images. The validation set used included 128 bounding boxes found in 18 images. It was trained for a total of 8 epochs. The regression loss was found to be 0.0272. Secondary tests were also completed test the RCNN's learning capability. The regression loss from two smaller groupings consisting of 144 and 134 bounding boxes were compared after a single epoch. The resulting loss was 0.3491 and 0.1942 respectively. These two values were relatively close, supporting the idea that the RCNN can be trained successfully on comparable data. The RCNN also substantially improves with more training data and epochs, as evidenced by the much smaller regression loss of the original training. More comprehensive analysis was not conducted as processing time took days to train the R-CNN, and further analysis was conducted on vU-net as its results were substantially better.

Sample #	# of Images Used	Sample Quantity	Epochs Completed	Regression Loss
1(Official)	8	726	8	0.0272
2	4	144	1	0.3491
3	2	134	1	0.1942
5 (Validation)	18	128	Validation Set	Validation Set

Table 2: Regression Loss

vUnet Evaluation

The first stage of this evaluation was to develop a training set for the vU-net. This training set included images from fine needle aspirations from 15 different patients. The automated microscope was used to collect these images. Each sample scanned by the microscope generated 54 images, but only one image from each subject was used to train the vU-net. In order to train the vU-net to recognize desired cell clusters, masks were generated in order to draw attention of the algorithms to strictly specific zones. These masks were developed using an open source photo-editing tool called Gimp. Cell clusters were denoted in white, and all background denoted in black. The images below demonstrate an example of a mask that was generated in order to train the vU-net. A mask was created for each of the 15 training images.

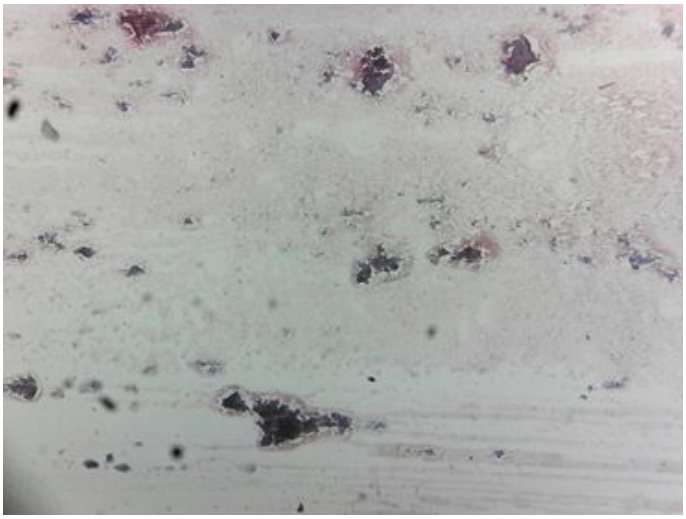

Figure 5: Sample Image

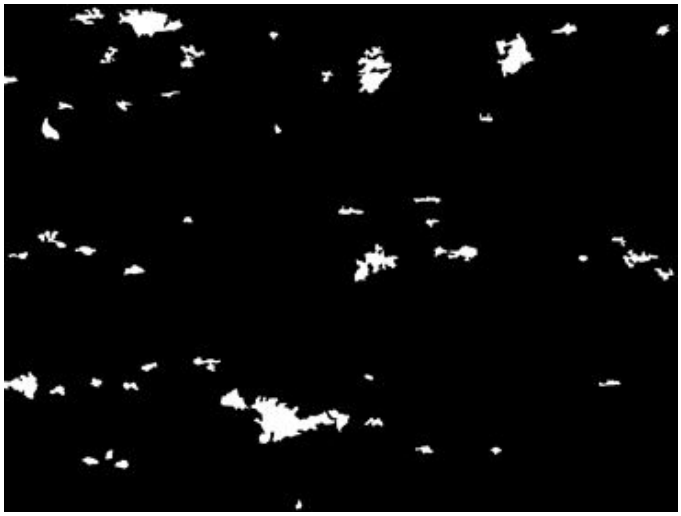

Figure 6: Mask Created for Sample Image

After creation of the masks, the vU-net was trained for 32 epochs on 12 of the images and masks. The remaining 3 images and masks were used for validation. This training was used to generate images to compare to the RCNN. A 5-fold cross validation was also completed in order to verify the robustness of the vU-net. In order to complete this computation, the 15 training images were divided into 5 groups of 3. Each group was rotated through to be used as the validation set. Training out to 5 epochs was completed for each set. The final loss errors were averaged together in order to yield the overall loss for the cross validation.

The following figures are plots of the loss and dice coefficient of the training and validation sets over the course of the 32 epochs. The final validation loss was 0.0157. This is generally considered a good fit, because the final validation error is very close to the training error. Note that the validation loss is actually lower than the training set initially.

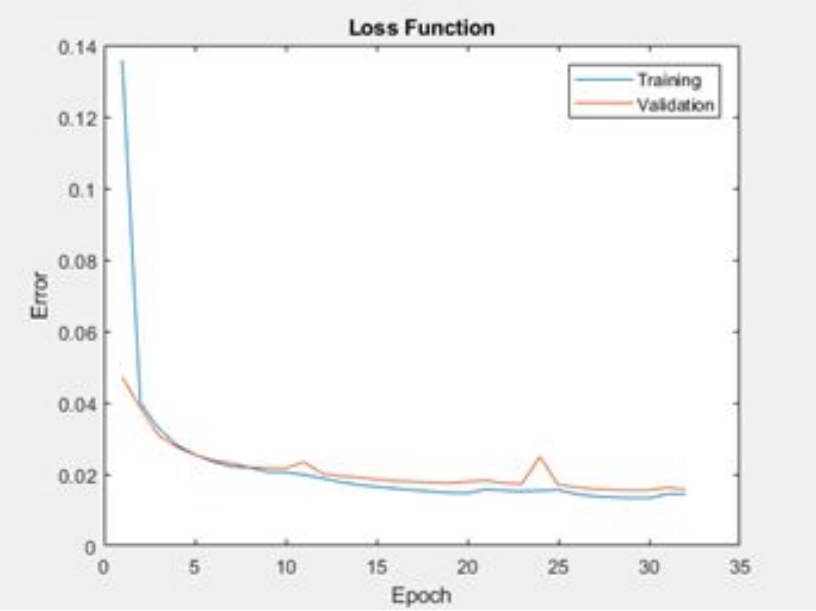

Figure 7: Training Loss Function, 0.0157

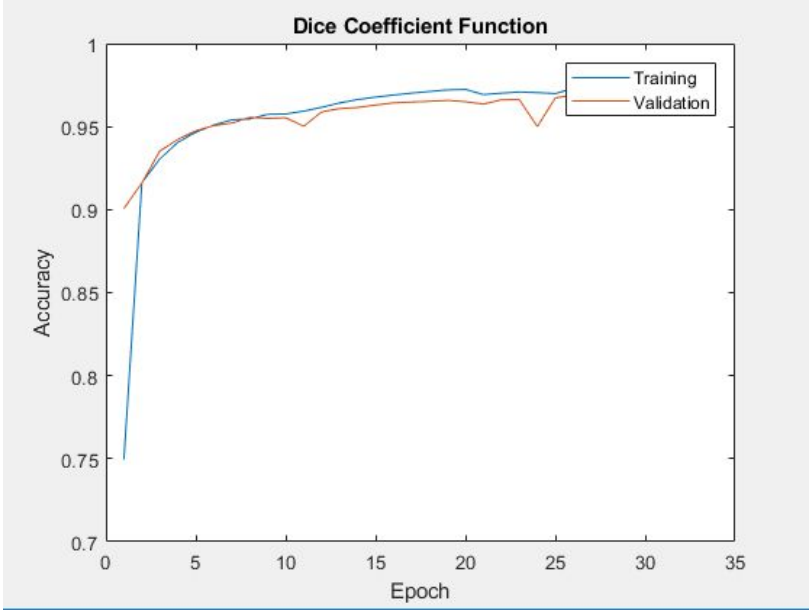

Figure 8: Dice Coefficient, 0.971

Figure 8 below is a plot of the validation loss curves for the 5-fold cross validation test. Note that 4 of the curves are very close, and that one has significantly more error. This indicates that the vU-net generally does a good job fitting to training and validation data, but is still challenged by difficult validation sets. The average error for the cross validation was 0.0391.

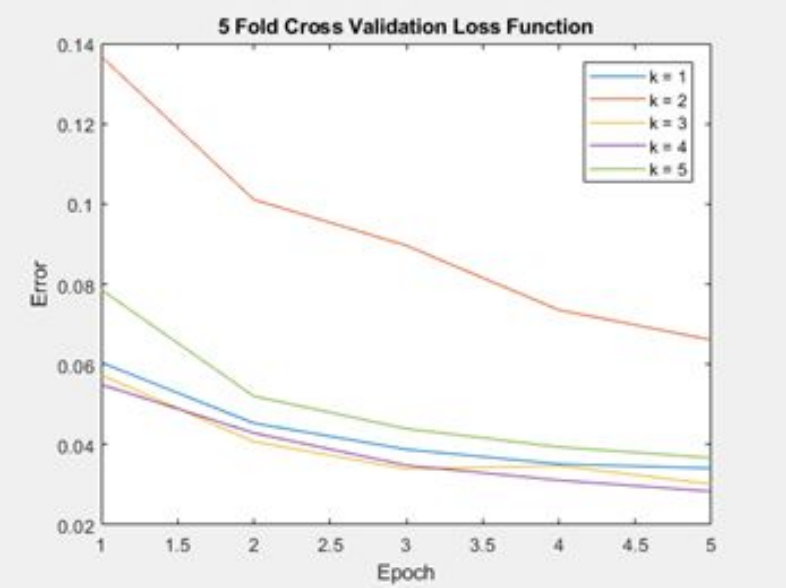

Figure 9: 5-Fold Cross Validation, 0.0391

After the vU-net was trained, it was tested against the R-CNN on testing images that were also collected from UMASS Medical School. The testing data had no truth masks generated, but allowed for visual inspection of results, and comparison to the output bounding boxes from R-CNN. The boundaries created by the vU-net were overlaid onto the original image using Gimp. Figure 5 is the output image generate by the RCNN, and Figure 6 is the image generated by the vU-net. Figure 7 shows comparison of RCNN on the left, and vU-net on the right.

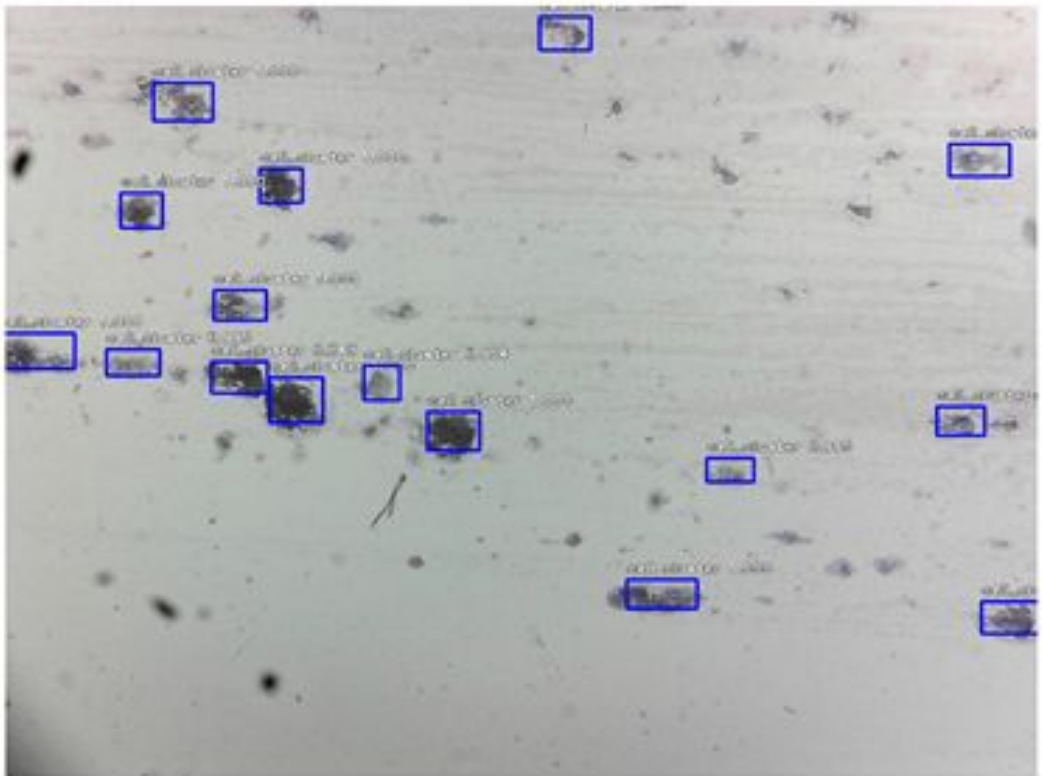

Figure 10: R-CNN Boxes

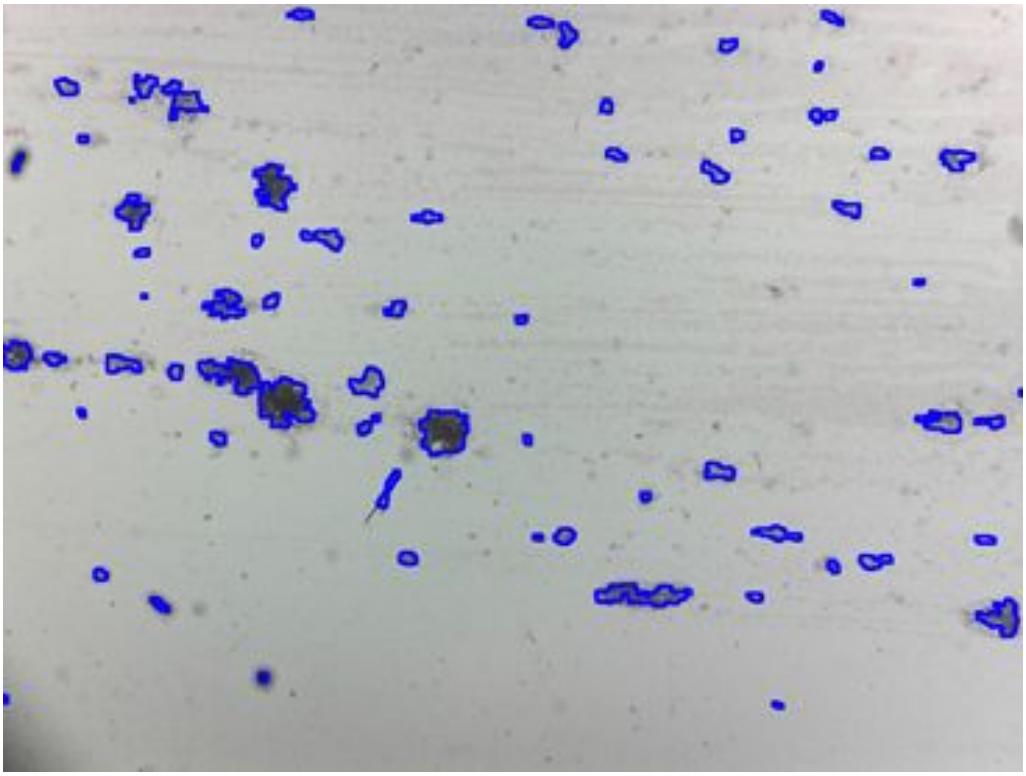

Figure 11: vU-net Segmentation

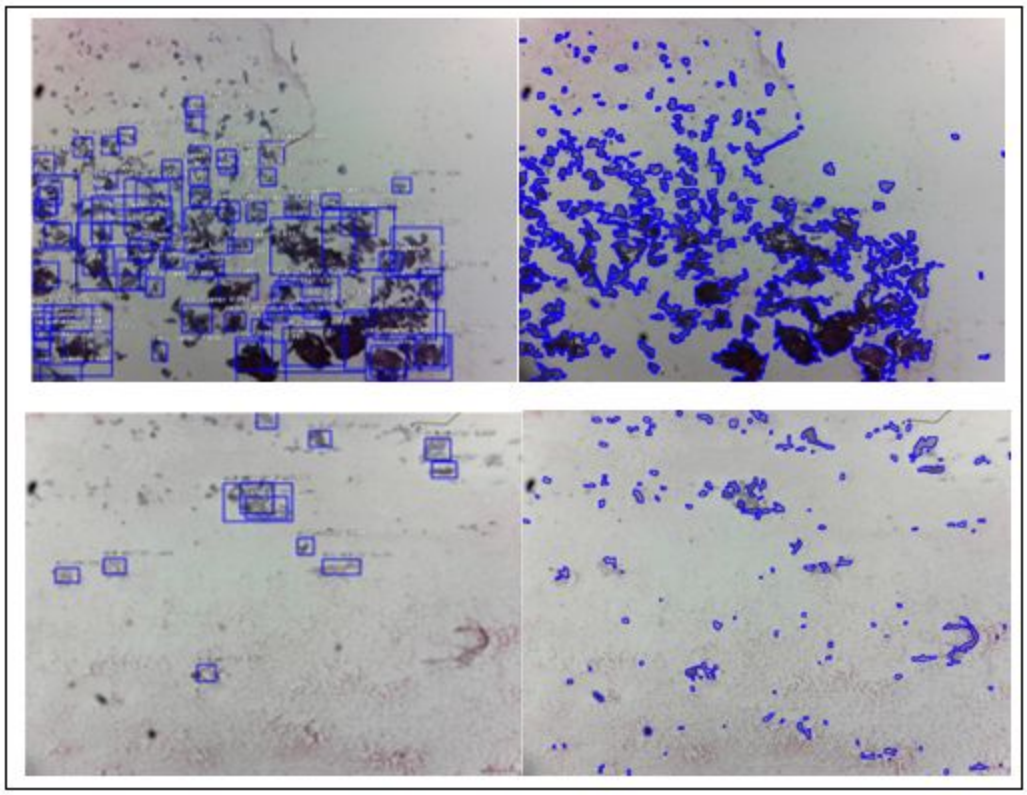

Figure 12: R-CNN vs. vU-net

Chapter 5: Recommendations

Enable Focusing

Future improvements should include the automated focusing of the microscope. The prototype is designed with a third axis to support this fine focusing. The current system is set to remain a constant distance from the sample for the entire course of the scanning, as the focus remains relatively constant throughout. This distance is set prior to the scanning, and controlled by sending G-Code commands to the Z-axis. However, this focusing could be controlled by the software as well. Images could be taken at two different distances, compared, and then evaluated for how much and in what direction the scope should be moved to focus the image. This could be accomplished by using the integrated artificial intelligence, or using other algorithms to calculate intensity or other characteristics.

Additional High Magnification Camera

This prototype could be greatly enhanced by outfitting the device with a secondary microscope of greater magnification. The current microscope is used as a scanning device and identifies clusters at the tissue level. A secondary scope should be added in order to provide images that the cellular level. It would be impractical to image the entire microscope area with a lens of 100x or 200x magnification, but after the cell clusters have been quickly identified, the secondary camera could image strictly the clusters in much greater detail. This additional scope would allow training and possible diagnosis of cancer using the integrated artificial intelligence. Benign colloid node cells could be compared to those with papillary thyroid cancer.

Other Applications

Due to the modularity of the artificial intelligence post processing, the designed system has nearly limitless capabilities for the field of pathology. The only limit is what kind of data the artificial intelligence is trained on, and the quantity of the data used to train it. Some noteworthy applications include training the AI to accomplish the following: to detect cell clusters of different types of cells; to detect cancerous cells and benign cells (however this would likely require a higher magnification camera); to detect cell clusters or make preliminary cancer diagnoses on unstained cells; and to tell if there are enough cell clusters of the correct size in the sample to warrant an accurate diagnosis. Whatever application a pathologist is needed, this device can be adapted in order to suit that need.

Chapter 6: Conclusions

Though a capstone, his project taught the team much about artificial intelligence, deep learning, and embedded systems. With the team's only relevant background consisting of introduction to embedded systems and introduction to programming, each member of the team learned much more about these field as well as the field of artificial intelligence. Though it was challenging, the end result is rewarding.

The team successfully created an automated microscope with an artificial intelligence post-processing that is able to detect cell clusters. Furthermore, the team went beyond the standard requirements by giving the device the capabilities to send the results wirelessly to other devices.

This project acts as a proof-of-concept and stepping stone for automation in the field of pathology. Though the current design only detects cell clusters, with time the system can be adapted to do more significant things like diagnose cancer without a pathologist or even count cell clusters and diagnose cancer on unstained samples. This advancement has the potential to save many lives by getting the patients their treatment faster, and the team is very proud to have a part in this design.

Bibliography

[1] The MNT Editorial Team, "Cancer: What you need to know," *MedicalNewsToday*. [Online]. Available: https://www.medicalnewstoday.com/info/cancer-oncology. [Accessed: Apr. 2, 2018].

[2] Cancer.Net, "Biopsy," *Cancer.Net*. Jan. 2018. [Online]. Available: https://www.cancer.net/navigating-cancer-care/diagnosing-cancer/tests-and-procedures/biopsy. [Accessed: Apr. 2, 2018].

[3] J. Xu, "Deep Learning for Object Detection: A Comprehensive Review," *Towards Data Science*. Sept. 11, 2017. [Online]. Available: https://towardsdatascience.com/deep-learning-for-object-detection-a-comprehensive-review-73930816d8d9. [Accessed: Apr. 2, 2018].

[4] National Cancer Institute, "Cancer Statistics," *National Cancer Institute*. Mar. 22, 2017. [Online]. Available: https://www.cancer.gov/about-cancer/understanding/statistics. [Accessed: Apr. 25, 2018].

[5] International Organization for Standards, "ISO/IEC JTC 1/SC 42 Artificial intelligence," *ISO*. 2017. [Online]. Available: https://www.iso.org/committee/6794475.html. [Accessed: Apr. 25, 2018].

[6] FDA, "FDA permits marketing of artificial intelligence-based device to detect certain diabetes-related eye problems," *Food and Drug Administration*. Apr. 11, 2018. [Online]. Available: https://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm604357.htm. [Accessed: Apr 25, 2018].

[7] McDonnell Boehnen Hulbert & Berghoff LLP, "FDA Permits Marketing of First AI-based Medical Device; Signals Fast Track Approach to Artificial Intelligence," *JDSUPRA*. [Online]. Available: https://www.jdsupra.com/legalnews/fda-permits-marketing-of-first-ai-based-68315/. [Accessed: Apr. 25, 2018].

[8] National Cancer Institute, "NCI Dictionary of Cancer Terms," *National Cancer Institute*. [Online]. Available: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/cancer. [Accessed: Apr. 25, 2018].

[9] R. A. Weinberg, "How Cancer Arises," *Scientific American Inc.* Sept. 1996. [Online]. Available: http://www.utdallas.edu/~burr/BIO2311/Weinberg-tumor%20progression.pdf.

[10] Mayo Clinic Staff, "Cancer," *Mayo Clinic*. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/cancer/symptoms-causes/syc-20370588. [Accessed: Apr. 25, 2018].

[11] B. Alberts, A. Johnson, J. Lewis, et al, "Programmed Cell Death (Apoptosis)," in Molecular Biology of the Cell, 4th edition. New York: Garland Science, 2002. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK26873/.

[12] R. Hruban, "What is cancer?," *Johns Hopkins University*. [Online]. Available: http://pathology.jhu.edu/pc/BasicTypes1.php?area=ba. [Accessed: Apr. 25, 2018].

[13] American Cancer Society, "Testing Biopsy and Cytology Specimens for Cancer," *American Cancer Society*. Jul. 30, 2015. [Online]. Available: https://www.cancer.org/treatment/understanding-your-diagnosis/tests/testing-biopsy-and-cytology-specimens-for-cancer/what-doctors-look-for.html. [Accessed: Apr. 25, 2018].

[14] G. Rolls, "An Introduction to Specimen Preparation," *Lieca Biosystems*. [Online]. Available: https://www.leicabiosystems.com/pathologyleaders/an-introduction-to-specimen-preparation/. [Accessed: Apr. 25, 2018].

[15] University of Leeds, "What is H&E," *University of Leeds*. [Online]. Available: http://histology.leeds.ac.uk/what-is-histology/H_and_E.php. [Accessed: Apr. 25, 2018].

[16] J. Duraiyan, R. Govindarajan, et al, "Applications of immunohistochemistry," *National Center for Biotechnology Information*. Aug. 4, 2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3467869/.

[17] Cancer.Net, "Reading a Pathology Report," *Cancer.Net*. Jan. 2016. [Online]. Available: https://www.cancer.net/navigating-cancer-care/diagnosing-cancer/reports-and-results/reading-pathology-report. [Accessed: Apr. 25, 2018].

[18] OED Online, "Artificial Intelligence," *Oxford University Press*. [Online]. Available: https://en.oxforddictionaries.com/definition/artificial_intelligence. [Accessed: Apr. 25, 2018].

[19] M. Copeland, "What's the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning?," *NVIDIA*. Jul. 29, 2016. [Online]. Available: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/. [Accessed: Apr. 25, 2018].

[20] DL4J, "What's the Difference Between Artificial Intelligence (AI), Machine Learning and Deep Learning?," *Deep Learning for Java*. [Online]. Available: https://deeplearning4j.org/ai-machinelearning-deeplearning. [Accessed: Apr. 25, 2018].

[21] Deepbitech, "How Deep Learning Revolutionizes Artificial Intelligence," *Deepbitech*. [Online]. Available: http://deepbitech.org/how-deep-learning-revolutionizes-artificial-intelligence/. [Accessed: Apr. 25, 2018].

[22] R. Girshick, J. Donahue, et al, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Cornell University*. Oct. 22, 2014. [Online]. Available: https://arxiv.org/abs/1311.2524.

[23] X. Sun, P. Wu, S. Hoi, "Face Detection using Deep Learning: An Improved Faster RCNN Approach," *Cornell University*. Jan. 28, 2017. [Online]. Available: https://arxiv.org/abs/1701.08289.

[24] T. H. N. Le, Y. Zheng, et al, "Multiple Scale Faster-RCNN Approach to Driver's Cell-phone Usage and Hands on Steering Wheel Detection," *Computer Vision Foundation*. June 2016. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2016_workshops/w3/papers/Le_Multiple_Scale_Faster-RCNN_CVPR_2016_paper.pdf.

[25] R. Girshick, "Fast R-CNN," *Cornell University*. Sept. 27, 2015. [Online]. Available: https://arxiv.org/abs/1504.08083.

[26] S. Ren, K. He, et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Cornell University*. Jan. 6, 2016. [Online]. Available: https://arxiv.org/abs/1506.01497.

[27] T. Lin, P. Goyal, et al, "Focal Loss for Dense Object Detection," *Cornell University*. Feb. 7, 2018. [Online]. Available: https://arxiv.org/abs/1708.02002.

[28] M Razzak, S. Naz, A. Zaib, "Deep Learning for Medical Image Processing: Overview, Challenges and Future," *Cornell University*. Apr. 22, 2017. [Online]. Available: https://arxiv.org/abs/1704.06825.

[29] R. C. S Rao, "New Product breakthroughs with recent advances in deep learning and future business opportunities," *Stanford Management Science and Engineering*. Jul. 6, 2017. [Online]. Available: https://mse238blog.stanford.edu/2017/07/ramdev10/new-product-breakthroughs-with-recent-advances-in-deep-learning-and-future-business-opportunities/.

[30] C. Demir, B. Yener, "Automated cancer diagnosis based on histopathological images: a systematic survey," *Rensselaer Polytechnic Institute*. 2005. [Online]. Available: https://pdfs.semanticscholar.org/ca4b/a745d8b2670282627828af7b8b51d7ed972c.pdf.

[31] A. Thon, U. Teichgräber, et al, "Computer aided detection in prostate cancer diagnostics: A promising alternative to biopsy? A retrospective study from 104 lesions with histological ground truth," *Public Library of Science*. Oct. 12, 2017. [Online]. Available: http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0185995.

[32] UCONN, "A low-cost, high-throughput whole slide imaging system," *University of Connecticut*. [Online]. Available: https://smartimaging.uconn.edu/instantscope/. [Accessed: Apr. 25, 2018].

[33] RSIP Vision, "What's the Difference between Computer Vision, Image Processing and Machine Learning?," *RSIP Vision*. [Online]. Available: https://www.rsipvision.com/defining-borders/.

[34] T. Pitkäaho, A. Manninen, T. J. Naughton, "Performance of Autofocus Capability of Deep Convolutional Neural Networks in Digital Holographic Microscopy," *OSA Publishing*. 2017. [Online]. Available: https://www.osapublishing.org/abstract.cfm?uri=DH-2017-W2A.5.

[35] E. Kim, M. Corte-Real, Z. Baloch, "A Deep Semantic Mobile Application for Thyroid Cytopathology," *SPIE*. Mar. 2016. [Online]. Available: http://edwardkim.net/pubs/fullpaper2016.pdf.

[36] fizyr, "keras-retinanet," *GitHub*. Jan. 2018. [Online]. Available: https://github.com/fizyr/keras-retinanet. [Accessed: Apr. 25, 2018].

[37] F. Chollet, "Keras," *GitHub*. Feb. 13, 2018. [Online]. Available: https://github.com/keras-team/keras. [Accessed: Apr. 25, 2018].

[38] M. Abadi, A. Agarwal, et al, "TensorFlow: Large-scale machine learning on heterogeneous systems," *TensorFlow.org*. [Online]. Available: https://www.tensorflow.org/. [Accessed: Apr. 25, 2018].

[39] NVIDIA, "NVIDIA Jetson," *NVIDIA*. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems-dev-kits-modules/. [Accessed: Apr. 25, 2018].

Appendix

Appendix A: Python Code for Program

```
### Load Necessary Modules
# show images inline
#%matplotlib inline

# automatically reload modules when they have changed
#%load_ext autoreload
#%autoreload 2



# import miscellaneous modules
import matplotlib.pyplot as plt
import cv2
import os
import numpy as np
import time
import datetime
import serial
import time
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive

### Function Definitions
############################################################
### Run Example Detection

def img_out(image_file, output_dir):

# load image
    image = read_image_bgr(output_dir + '/' + image_file)

# copy to draw on
    draw = image.copy()
    draw = cv2.cvtColor(draw, cv2.COLOR_BGR2RGB)

# preprocess image for network
    image = preprocess_image(image)
    image, scale = resize_image(image)

# process image
    start = time.time()
    _, _, detections = model.predict_on_batch(np.expand_dims(image, axis=0))
    print("processing time: ", time.time() - start)

# compute predicted labels and scores
    predicted_labels = np.argmax(detections[0, :, 4:], axis=1)
    scores = detections[0, np.arange(detections.shape[1]), 4 + predicted_labels]

# correct for image scale
    detections[0, :, :4] /= scale

# visualize detections
```

```
    for idx, (label, score) in enumerate(zip(predicted_labels, scores)):
        if score < 0.5:
            continue
        b = detections[0, idx, :4].astype(int)
        cv2.rectangle(draw, (b[0], b[1]), (b[2], b[3]), (0, 0, 255), 3)
        caption = "{} {:.3f}".format(labels_to_names[label], score)
        cv2.putText(draw, caption, (b[0], b[1] - 10), cv2.FONT_HERSHEY_PLAIN, 1.5, (0, 0, 0), 3)
        cv2.putText(draw, caption, (b[0], b[1] - 10), cv2.FONT_HERSHEY_PLAIN, 1.5, (255, 255, 255), 2)

    plt.figure(figsize=(15, 15))
    plt.axis('off')
    plt.imshow(draw)
    plt.savefig(output_dir + '/out' + image_file, bbox_inches='tight')
    plt.close()
    plt.close()
    return

############################################################
### Begin Program
os.system("sudo nvpmodel -m 0")

### Make New File Directory
CLK = time.localtime()
#imgdir = 'fakedir'
imgdir = str(CLK.tm_mon)+ "-" + str(CLK.tm_mday)+ "-" + str(CLK.tm_year)+ "_" + str(CLK.tm_hour)+ ":" + str(CLK.tm_min)
os.system("mkdir " + imgdir)

############################################################
### Taking Pictures
print "Initializing Hardware..."
COM = '/dev/ttyACM0' #'COM3'# /dev/ttyACM0 (Linux)
BAUD = 115200

ser = serial.Serial(COM, BAUD, timeout = .1)
Flag = 0

print('Waiting for device');
time.sleep(3)
print(ser.name)

# Wake up grbl
ser.write("\r\n\r\n")
time.sleep(2)   # Wait for grbl to initialize
ser.flushInput()  # Flush startup text in serial input

# Wake up camera  ** after -S controls timing of shots
photo = "fswebcam -d /dev/video1 -r 1600x1200 --fps 10 -S 8 --jpeg -95 --no-banner "  + imgdir +"/"

print('Set Relative Distance Mode')
ser.write('G91\n')
grbl_out = ser.readline() # Wait for grbl response with carriage return
```

```
print grbl_out.strip()

#GO HOME ----------------------------------Implement

print('Go to Zero')
y = 1
p = 1
print '(' + '1' + ',' + str(y) + ')'
ser.write('G0 X0 Y0 Z0 \n')
grbl_out = ser.readline() # Wait for grbl response with carriage return
print grbl_out.strip()
name = str(p)
name = name.rjust(6,'0')
os.system(photo+name+".jpg")
p = p+1

"Taking Pictures..."

for k in range(1,4):

    for i in range(2,10):
        print '(' + str(i) + ',' + str(y) + ')'
        ser.write('G0 X6 Y0 Z0 \n')
        grbl_out = ser.readline()
        print grbl_out.strip()
        name = str(p)
        name = name.rjust(6,'0')
        os.system(photo+name+".jpg")
        p = p+1
        #time.sleep(2)

    y = y+1
    print '(' + '9' + ',' + str(y) + ')'
    ser.write('G0 X0 Y4 Z0 \n')
    grbl_out = ser.readline()
    print grbl_out.strip()
    name = str(p)
    name = name.rjust(6,'0')
    os.system(photo+name+".jpg")
    p = p+1
    #time.sleep(2)

    for j in range(2,10):
        print '(' + str(12-j) + ',' + str(y) + ')'
        ser.write('G0 X-6 Y0 Z0 \n')
        grbl_out = ser.readline()
        print grbl_out.strip()
        name = str(p)
        name = name.rjust(6,'0')
        os.system(photo+name+".jpg")
        p = p+1
        #time.sleep(2)
```

```
    y = y+1
    print '(' + '1' + ',' + str(y) + ')'
    ser.write('G0 X0 Y4 Z0 \n')
    grbl_out = ser.readline()
    print grbl_out.strip()
    name = str(p)
    name = name.rjust(6,'0')
    if p < 54:
        os.system(photo+name+".jpg")
    p = p+1
    #time.sleep(2)

print 'Go Zero'
ser.write('G0 X0 Y-24 Z0 \n')
grbl_out = ser.readline() # Wait for grbl response with carriage return
print grbl_out.strip()

# Close file and serial port
ser.close()

############################################################
### Import Keras
print "Setting Up Keras..."
# import keras
import keras
# import keras_retinanet
from keras_retinanet.models.resnet import custom_objects
from keras_retinanet.utils.image import read_image_bgr, preprocess_image, resize_image
from keras_retinanet.utils.visualization import draw_box, draw_caption
#from keras_retinanet.utils.colors import label_color

### Setup Tensorflow Backend

# set tf backend to allow memory to grow, instead of claiming everything
import tensorflow as tf

        # definition set in loop in order to start scanning immediately
def get_session():
    gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=0.2)
    config = tf.ConfigProto(gpu_options=gpu_options)
    config.gpu_options.allow_growth = True
    return tf.Session(config=config)

# use this environment flag to change which GPU to use
#os.environ["CUDA_VISIBLE_DEVICES"] = "1"

# set the modified tf session as backend in keras
keras.backend.tensorflow_backend.set_session(get_session())

############################################################
### Load RetinaNet
print "Loading RetinaNet..."
# adjust this to point to your downloaded/trained model
```

```
model_path = os.path.join('.', 'resnet50_pascal_50.h5')


# load retinanet model
model = keras.models.load_model(model_path, custom_objects=custom_objects)
#print(model.summary())

# load label to names mapping for visualization purposes
labels_to_names = {0: 'cell_cluster', 1: 'cancer'}

############################################################
### Image Testing
print "Testing Images..."
for index in range(1,55):
        file_name = str(index)
        img_out(file_name.rjust(6,"0") + '.jpg', imgdir)
############################################################
### Google Upload
print "Uploading to Google..."
os.system('python ./auth.py &')
gauth = GoogleAuth()
gauth.LocalWebserverAuth()
drive = GoogleDrive(gauth)

#imgdir = 'fakedir'

# Create folder.
folder_metadata1 = {
    'title' : imgdir,
    # The mimetype defines this new file as a folder, so don't change this.
    'mimeType' : 'application/vnd.google-apps.folder'
}
folder1 = drive.CreateFile(folder_metadata1)
folder1.Upload()

# Create folder.
folder_metadata2 = {
    'title' : 'out_' + imgdir,
    # The mimetype defines this new file as a folder, so don't change this.
    'mimeType' : 'application/vnd.google-apps.folder'
}
folder2 = drive.CreateFile(folder_metadata2)
folder2.Upload()

# Get folder info and print to screen.
folder_title1 = folder1['title']
folder_id1 = folder1['id']
print('title: %s, id: %s' % (folder_title1, folder_id1))

# Get folder info and print to screen.
folder_title2 = folder2['title']
folder_id2 = folder2['id']
```

```
print('title: %s, id: %s' % (folder_title2, folder_id2))


for i in range(1,55):
    filestr = str(i)
    orgfile = filestr.rjust(6,'0') +'.jpg'
    outfile = "out" +  orgfile
    print'Uploading %s ...' %(orgfile)
    file1 = drive.CreateFile({"parents": [{"kind": "drive#fileLink", "id": folder_id1}]})
    file1.SetContentFile('./' + imgdir +"/" + orgfile)
    file1.Upload()
    file2 = drive.CreateFile({"parents": [{"kind": "drive#fileLink", "id": folder_id2}]})
    file2.SetContentFile('./' + imgdir +"/" + outfile)
    file2.Upload()
############################################################
### End Program
start = 1
os.system("nvpmodel -m 1")
print("Concluding...")
```

Appendix B: Bill of Materials

Part Number	Name	Discription	Manufacturer	Vender	Vender Number	Item Cost	Required	Total Cost
1	Digital_Microscope	USB Microscope	Monoprice	Monoprice	11613	$24.99	1	$24.99
2	Driver_Shield	3 Axis Motor Driver for Arduino	Synthetos	Synthetos	V5	$49.99	1	$49.99
3	Stepper_Motor	Nema 17 Motor 26Ncm 12V 0.4A	StepperOnline	Amazon	17HS13-0404S	$11.99	3	$35.97
4	Motor_Mount	[3 Pack] Nema 17 Mounting Bracket	Beauty Star	Amazon	Nema-17-Bracket	$8.99	1	$8.99
5	Lead_Screw	[Package] T8 8mm Lead Screw 4 Start 300mm	BIQU	Amazon	BIQU-KIT091	$12.98	3	$38.94
6	Coupler	8mm to 5mm Coupler	BIQU	Amazon	BIQU-KIT092	-	3	
7	Ball_Bearing	Pillow Block Mounted Ball Bearing	BIQU	Amazon	BIQU-KIT093	-	6	
8	Anti_Backlash_Nut	Anti-Backlash Nut T8 8mm 4 Start	Wangdd22	Amazon	LEE-T8	$6.95	3	$20.85
9	Power_Supply	12V 20Amp 240W Power Supply	Alunar	Amazon	ALSP20A05	$16.99	1	$16.99
10	Linear_Slide	[4 Pack] Linear Slide 8mm D 500mm L	Mergorun	Amazon	B01LPZPJ18	$38.69	1	$38.69
11	Extruded_Aluminum	[2 Pack] 10mm x 10mm x 1500mm	MakerBeam	Amazon	103442	$29.00	2	$58.00
12	T_Nuts	[25 Pack] Nuts and Screws	MakerBeam	Amazon	101619	$17.00	4	$68.00
13	Corner_Cube	[12 Pack] M3 Corner Cube and Screws	MakerBeam	Amazon	FBA_100988	$16.25	1	$16.25
14	Corner_Brackets	[12 Pack] M3 Corner Brackets and Screws	MakerBeam	Amazon	bcrp12	$9.28	1	$9.28
15	Lamp	LED Flashlight 250 lumens	Strate Biological	Amazon	ML007	$4.99	1	$4.99
16	Limit_Switch	[6 Pack] Limit Switches	Qianxin	Amazon	CYT1046	$8.93	1	$8.93
17	Bolt_M3_25	25mm M3 Flat Button Head	McMaster	McMaster	90353A150	$4.09	1	$4.09
							PO Total:	$404.95