2018 ASEE Zone IV Conference: Boulder, Colorado Mar 25

Inspiring Community College Students in Electrical and Computer Engineering Research through Live Digit Recognition using Nvidia's Jetson Tx1

Mr. Jayson Paul Mercurio, Canada College

Jayson Mercurio studies computer science at UC Santa Barbara. He is a recent transfer student from Canada College in Redwood City and interned at SFSU over the summer of 2017, working on image recognition with neural networks.

Kevin Yamada

Mr. Jose L. Guzman, Canada College

Jose L. Guzman is currently an undergraduate at Canada College. He participated in a research program at San Francisco State University where he focuses on a conventual neural network to identify the object at high and accurate results.

Dr. Xiaorong Zhang, San Francisco State University

Xiaorong Zhang received the B.S. degree in computer science from Huazhong University of Science and Technology, China, in 2006, the M.S. and the Ph.D. degrees in computer engineering from University of Rhode Island, Kingston, in 2009 and 2013 respectively. She is currently an Assistant Professor in the School of Engineering at San Francisco State University. Her research interests include embedded systems, wearable technologies, neural-machine interface, and cyber-physical systems.

Dr. Wenshen Pong P.E., San Francisco State University

Wenshen Pong received his Ph.D. in Structural Engineering from the State University of New York at Buffalo. He joined the School of Engineering at San Francisco State University in 1998. He teaches courses in Civil/Structural Engineering.

Dr. Pong is a registered Professional Engineer in California. He is a member of the American Society of Civil Engineers and the Structural Engineers Association of California. He has published over fifty technical papers in the areas of Structural Control and Earthquake Engineering. Dr. Pong has been the Director of the School of Engineering at SFSU with 20 full-time faculty and over 25 part-time faculty since 2009.

Dr. Amelito G Enriquez, Canada College

Amelito Enriquez is a professor of Engineering and Mathematics at Cañada College in Redwood City, CA. He received a BS in Geodetic Engineering from the University of the Philippines, his MS in Geodetic Science from the Ohio State University, and his PhD in Mechanical Engineering from the University of California, Irvine. His research interests include technology-enhanced instruction and increasing the representation of female, minority and other underrepresented groups in mathematics, science and engineering.

Dr. Zhaoshuo Jiang P.E., San Francisco State University

Zhaoshuo Jiang graduated from the University of Connecticut with a Ph.D. degree in Civil Engineering. Before joining San Francisco State University as an assistant professor, he worked as a structural engineering professional at Skidmore, Owings & Merrill (SOM) LLP. As a licensed professional engineer in the states of Connecticut and California, Dr. Jiang has been involved in the design of a variety of low-rise and high-rise projects. His current research interests mainly focus on Smart Structures Technology, Structural Control and Health Monitoring and Innovative Engineering Education.

Dr. Cheng Chen, San Francisco State University

Dr. Cheng Chen is currently an associate professor in the school of engineering at San Francisco State University. His research interests include earthquake engineering, structural reliability and fire structural engineering.

Dr. Kwok Siong Teh, San Francisco State University

Kwok Siong Teh received his B.S., M.S., Ph.D. degrees in Mechanical Engineering from the University of Illinois Urbana-Champaign, University of Michigan at Ann Arbor, and University of California at Berkeley in 1997, 2001, and 2004, respectively. He is currently an associate professor of mechanical engineering at San Francisco State University. His primary research interests are in: (i) the synthesis, characterization, and applications of metal oxides, conductive polymer, and low dimensional carbon nanostructures for en-

Inspiring Community College Students in Electrical and Computer Engineering Research through Live Digit Recognition using Nvidia's Jetson Tx1

Jayson Mercurio[2], Kevin Yamada[1], Alexander Choi[2], Ayesha Iqbal[2], Jose I. Guzman[2], Amelito G. Enriquez[2], Xiaorong Zhang[1], Wenshen Pong[1], Zhaoshuo Jiang[1], Cheng Chen[1], Kwok-Siong Teh[1], Hamid Mahmoodi[1] and Hao Jiang[1],

[1] San Francisco State University, San Francisco, CA
[2] Cañada College, Redwood City, CA

Abstract

Community colleges provide a beneficial foundation for undergraduate education in STEM majors. To inspire community college students to pursue a major in STEM, it is crucial to adapt strategies that help facilitate this interest. With support from the Department of Education Minority Science and Engineering Improvement program (MSEIP) and the Hispanic-Serving Institution Science, Technology, Engineering and Mathematics (HSI STEM), an internship program with multiple colleges was developed between community colleges and a public four-year university to engage community college students in cutting-edge engineering research. In the summer of 2017, four community college students participated in a ten-week electrical and computer engineering research internship project at a four-year university research lab. The summer internship project aimed to develop a real-time handwritten digit recognition system leveraging Neural Networks and Nvidia's Jetson Tx1 platform. Utilizing a modified Nvidia workflow, a robust digit recognition algorithm was designed using two industry standard programs for deep learning -- TensorFlow and DIGITS. Nvidia's live image recognition demonstration created the framework to interface a camera module that sends images to the input of the digit classifying network in real-time. The student interns designed experiments to test the robustness of the algorithm in their daily environment, from low light situations to cluttered backgrounds with the handwritten digit blending in. The internship project created a stimulating environment for student interns to gain research experiences and learn a wealth of knowledge in deep learning, real time pattern recognition systems and leading-edge hardware platforms. The experiences contained within the ten-week internship allowed the interns to drastically improve technical writing and presentations, experimental design, data analysis and management, teamwork, and perseverance. The ten-week research internship was an effective method for engaging aspiring community college students by teaching the tools and methodology for success within an engineering profession, and helping to increase the interns' confidence levels.

I. Introduction

In the United States, there is a strong consensus that a large increase in the number of professionals in the science, technology, engineering, and mathematic (STEM) community is essential to progress the stability, competitiveness, and growth of the nation's economy. Community colleges widen the STEM pipeline, increasing the student preparation to continue their education in a STEM major at a higher education university. The role of the community college in a STEM field is more prominent for individuals who are from underrepresented populations, such as females and minority groups. To increase the recruitment and retention of STEM students, it is important for community colleges to provide students with impactful opportunities, such as STEM research. The limited resources found at community colleges makes it nearly impossible to build the research infrastructure seen at four-year research universities. However, establishing collaborations between community colleges and higher education research universities is a recommended solution to address the limited resource problem. With support from the Department of Education Minority Science and Engineering Improvement program (MSEIP) and the Hispanic-Serving Institution Science, Technology, Engineering and Mathematics (HSI STEM), an internship program with multiple colleges was developed between community colleges and a public four-year university to develop and implement the Accelerated STEM Pathways through Internships, Research, Engagement, and Support (ASPIRES) program, to aid the retention and interest of STEM in underrepresented minority students. A large part of the ASPIRES program is a ten-week summer research internship for community college students in engineering to conduct leading edge research at higher education research laboratories. This paper covers the summer 2017 Electrical and Computer Engineering (ECE) research project. The research project consisted of four community college interns, a graduate mentor, and faculty advisor from the sponsoring four-year university to design a real-time live digit recognition system (RTLDRS) using Nvidia's Tx1 in the Bioelectronic Research Laboratory.

The 2017 summer ECE project aimed to develop a robust fast training neural network (NN) for live digit recognition utilizing industry standard deep learning software. The NN model would be imported to Nvidia's Jetson Tx1 for real time live digit recognition (RTLDR) on the go. The student interns used a modified Nvidia workflow and live image recognition demonstration as the framework to interface a camera module, that sends images to the input of the NN model for digit classification. The student interns designed an experiment to validate the robustness of their NN model by showing their RTLDRS different handwritten test images in various lighting conditions, cluttered background, various handwriting styles, and line thickness. The internship project exposed the student interns to numerous new engineering concepts and allowed the student interns to gain a wealth of knowledge in deep learning, real time pattern recognition systems, and leading-edge hardware platforms. The student interns improved on their technical writing, formal presentations, design methodology, expressing ideas clearly during team meetings, and perseverance throughout the course of the ten-week internship. The summer research internship was an effective method for inspiring and boosting the confidence of community college students' interest in electrical and computer engineering by teaching the tools and methodology for success within an engineering profession.

II. Internship Program Activities

The electrical and computer engineering project team consisted of one full-time intern and three part-time interns. The graduate student mentor presented an introduction of the research project on the opening day of the ASPIRES internship program. The ten-week internship activities for the RTLDRS project were divided into two weeks of learning the basic theory of machine learning and deep learning, six weeks of learning/exploring industry standard deep learning programs and Nvidia's Jetson Tx1, one week for preparing a midterm and final presentation, and one week for making the final presentation poster and writing the final report. The interns presented project updates, issues, and ideas with the graduate mentor during the team's weekly morning meetings. During each meeting, the full-time student intern and the graduate mentor assigned a weekly task to each team member followed by project presentations and open discussion. Each team member wrote meeting notes in their personal project notebook to keep track of tasks, raw data, and various mental notes. The PowerPoint based meeting presentations were task specific, for example the full-time intern would present on his/ her NN model, discuss how the code was written, the accuracy of the model, and what was the goal of this specific model. The progression of each team member was tracked weekly on a whiteboard, in which each task was checked off at the end of the day.

The outcome of all the research projects were evaluated twice during the ten-week internship program by both a midterm and final PowerPoint presentation, a poster, and a written report. The final PowerPoint presentation and poster were judged by faculty advisors and graduate student mentors, while the final report was judged solely by the directors of the ASPIRES program. Each project was ranked by a point based system, where the final presentation was worth 50%, the poster was worth 25%, and the written report was worth 25% of the overall score. The highest scoring project was the winning project among all the participating teams.

III. Theory and architecture of a Neural Network

MLP Network Overview

The field of artificial neural networks is often referred to as neural network for shorthand, and most commonly thought of as the multilayer perceptron. Neural networks are biologically inspired, which mimic simple models of the biological brains that can be used to solve difficult computationally expensive problems from image classification to a predictive stock market model. However, the goal of a neural network is not to create realistic models of the brain, but to use the brain as inspiration to develop robust algorithms and data structures that can be used to model difficult problems efficiently. The predictive power of neural networks come from their ability to learn from a given dataset and correlate the output to a category or an object that is to be classified or predicted. The predictive capability of the multi-layer perceptron comes from the hierarchical structure of the network, in a high-level sense an MLP is good at mapping. A MLP can be trained to learn features at unknown resolutions and combine them into complex multi-order features, such as complex shapes built from a collection of lines.

The basic building block for neural networks are artificial neurons, commonly referred to as nodes, units, or neurons. There are two types of neurons – one with an activation function (hidden unit neurons) and the other only containing weights. Each neuron weight can be thought of as coefficients used in a multi-order equation. Each neuron also contains a bias to help shift the network weights. The weights of neural networks are randomly initialized to a small range with high precision. Larger weight values can indicate that a neuron is very complex. In most networks, robust noncomplex neurons are desirable. Weight regularization techniques can be applied to keep weight values small.

Hidden unit neurons use a nonlinear activation function that governs the threshold at which the neuron is activated and determines how strong the output of the neuron is. Historically, nonlinear activation functions, such as logistic sigmoid and hyperbolic tangent, were used to allow the network to combine inputs in complex manners, providing robust feature detection. Today's go-to activation function is the rectified linear unit (RELU), which has shown better results.

A MLP is a feed forward neural network which contains three or more different layers (input layer, hidden layer, and output layer). A MLP is a fully connected network as each neuron in one layer connects to every neuron in the next layer by a certain weight, as shown in figure 1. In a typical MLP the input layer is a single dimensional vector which represents one neuron per input value of the dataset. The role of the input vector is to pass the input value to the next layer. The hidden layer is sandwiched between the input layer and the output layer. The hidden layer contains many hidden unit neurons that sum the input and pass the output into a nonlinear activation function. The final layer is the output layer and is responsible for outputting a vector of values that corresponds to the format required for the problem. The output layer is unique as it is dependent on the predictive or classification task. For a multi-class classification function, there is one neuron per class. The output of the neurons is pushed into a SoftMax function (probability distribution function) used to predict the probability of each class.

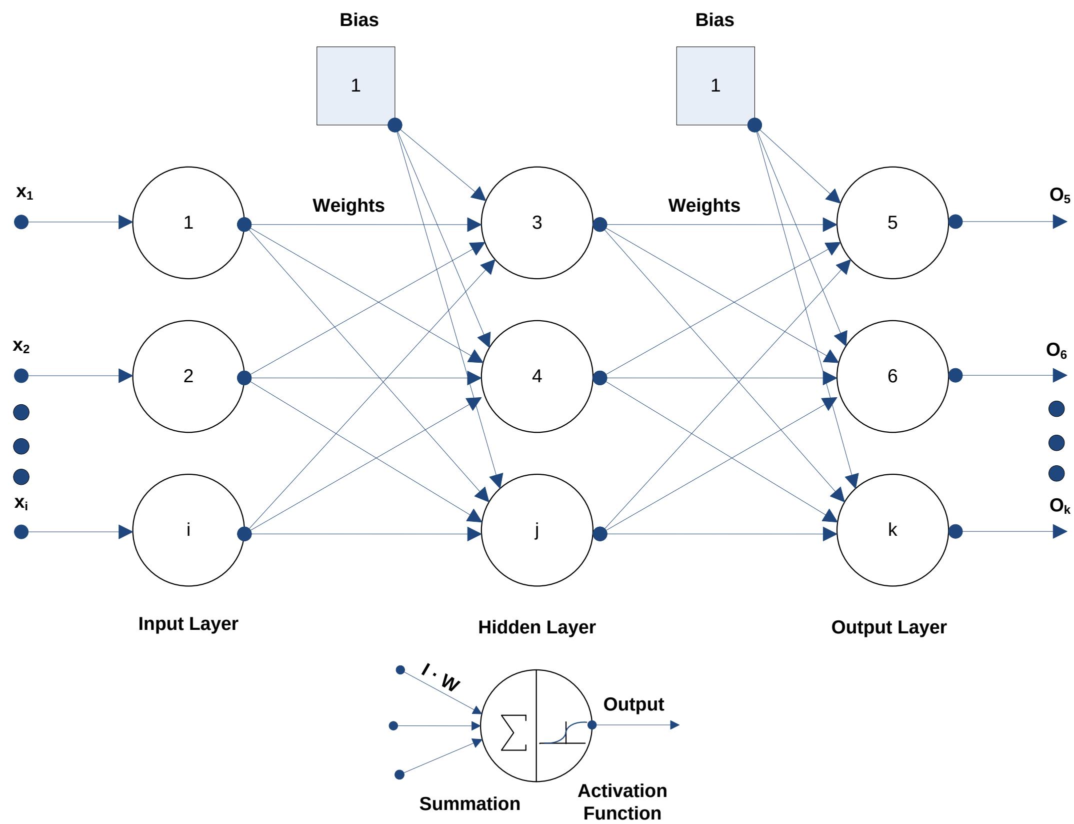

Figure 1: Multi-Layer Perceptron Model

Neural Network Training

Once the neural network model is designed it needs to be trained based on the chosen dataset. A typical dataset for image classification is split into a training set, validation set, and test set. Each image has a fixed sized and contains a label allowing for easy comparison during the training process. Every image is normalized between the range of zero and one. Once the data is prepared, the training process can begin. The idea of training a NN model is to minimize the overall error within the model and to create robust weights. To train a NN, one of the training images must be pushed into the network activating neuron as the data propagates to the output, this is called the forward pass. The output of the forward pass is compared against the expected output and the error is calculated. This error then propagates back through the NN model, layer by layer updating the weight according to the amount they contributed to the error. The process of updating weight through the model is known as the backpropagation algorithm. The process is repeated is X number of epochs. An epoch is one forward pass and one backward pass of the entire training dataset. During the training process, additional parameters can be applied, such as the amount a weight can update, how the weights are updated, and how often the weights are updated. Each change in the training process can result in a high model accuracy but a long training time, low accuracy and fast training time, or any combination in between.

IV. Design and Results of the Research Project

A. Project Background and Motivation

The popularity of machine learning and deep learning has drastically increased in recent years, as companies like Facebook, Google, and Tesla are pushing the boundaries of artificial intelligence, self-driving cars, and big data analysis. Research within the era of big data has pushed the limits of modern central processing units (CPU), driving companies like Nvidia to build massive graphic processing unit (GPU) based data centers and deep learning specific hardware. The current hardware solution for deep learning researchers are custom computers with multiple high-end Nvidia GPUs, water cooling, and massive power supplies costing upwards of ten thousand USD. These expensive deep learning machines are not a viable option for many advanced deep learning projects that need the compute power of a GPU, but the portability of a microcontroller. In late 2015, Nvidia launched the Jetson Tx1 a powerful System on Module (SOM) as the perfect solution for any mobile compute intensive system.

Nvidia's Jetson Tx1 is the perfect embedded solution to design a RTLDRS due to the compute power, low power consumption, and portability. As the Jetson Tx1 is the perfect hardware solution, Google's TensorFlow and Nvidia's Digit are industry standards for designing custom NN that can easily be exported to the Jetson. Using different datasets, NN architectures, training algorithms, and hardware determines the time it takes to train a model, the overall classification accuracy, and what images the model can classify.

This internship project aimed to develop a fast training NN designed specifically for the Jetson Tx1, that can classify handwritten digits in real time by feeding video frames into the NN model and displaying the digit classification and the confidence level on a monitor.

B. Design and Implementation

Modified Nvidia workflow for the Jetson Tx1

The approach to designing and implementing a RTLDRS for the Jetson Tx1 was inspired by the Nvidia standardized workflow for the Jetson, shown in figure 2. The approach that was used was a modified Nvidia workflow that was divided into five major tasks: 1) Choose the dataset, 2) design, train and optimize a NN model, 3) deploy the trained model to the Jetson, 4) interface a USB camera, and 5) classify test images in real-time. Using a modified Nvidia workflow allowed the NN model to be design with more flexibility in TensorFlow. The NN model parameters would input into Nvidia's proprietary deep learning software Digits, to build the final Jetson ready model.

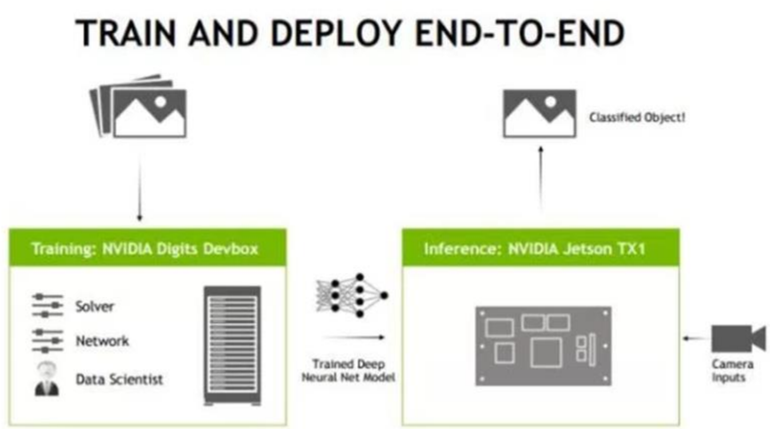

Figure 2: Nvidia Jetson Tx1 Workflow

MNIST Dataset and TensorFlow

The Modified National Institution of Standards and Technology (MNIST) is a large database of handwritten digits from 0 to 9 that originated from the NIST database. The original NIST database was created from the American Census Bureau employees and American high school students. The MNIST dataset contains 60,000 training images and 10,000 test images which were upscaled to 28x28 pixels and centered each handwritten digit. The MNIST dataset was chosen due to two major factors. The first factor was the MNIST dataset is a classically proven dataset known for high accuracy in various machine learning and deep learning models and the second factor was shallow MLP models can obtain high accuracy for the MNIST dataset.

The MNIST MLP model was based on TensorFlow's MNIST tutorial, that demonstrated the flexibility and efficient TensorFlow functions for fast NN model building. TensorFlow natively supports Nvidia GPUs and Compute Unified Device Architecture (CUDA) for fast parallel performance. Using the tutorial MLP model parameters as the baseline model allowed for good accuracy but overall slow training time. All tests were conducted using a single computer containing an Intel i7 7700, GTX 1070, 16 GB of Ram, and a solid state hard drive.

Model Validation

To create a robust fast training MLP with high accuracy for the Jetson Tx1 based on the MNIST dataset, a series of test were conducted to find the best model parameters that yield a fast training model with limited accuracy loss. Fast training time was the determining parameter above high accuracy, due to the Jetson smaller Maxwell based GPU containing 256 CUDA cores. In comparison, the GTX 1070 used to benchmark the results below contains 1920 CUDA cores of the newer, more efficient pascal based GPU architecture. Training the same NN model on the Jetson Tx1 could take upwards of 10 times longer than the benchmarking computer due to the limiting amount of GPU compute power.

To prepare the lightweight NN model for the Jetson Tx1, seven parameters were individually varied to measure the impact each parameter had on the NN model. Shown in figure 3 is the baseline model and the seven different parameters that were used to construct the NN model. Each parameter was assigned a default value from the baseline NN model, and then a single parameter was varied. Each time a single parameter was changed, the model was retrained five times and the average was taken. The first set of tests were to vary one of the seven NN parameters and to see how accuracy changed. The second set of tests took training time into account.

Baseline Network	
Batch Size	256
Training Epochs	100
Learning Rate	0.001
Optimizer	Adam
Hidden Units	100
Activation Function	ReLU
Number of Hidden Layers	1

Figure 3: Baseline NN Parameters

The number of training epochs and the number of hidden unit neurons versus accuracy results are shown in figure 4 and 5. Figure 4 indicates that the NN model accuracy reaches a steady state when the model is trained with more than 200 epochs. Figure 5 indicates that an increase in the number of hidden unit neurons marginally improves the accuracy of the NN model.

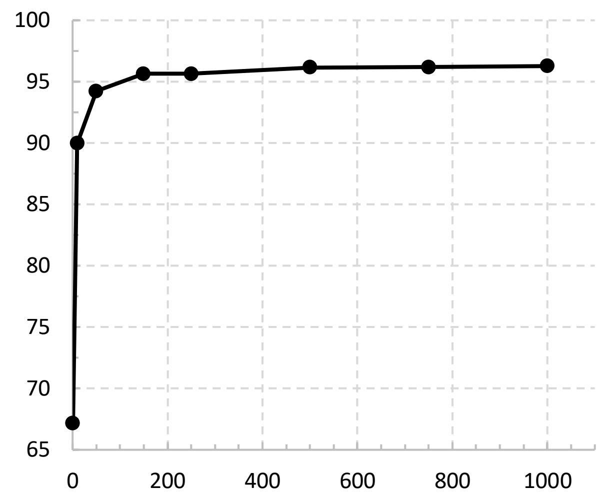

Figure 4: Number of Epochs vs. Accuracy (%)

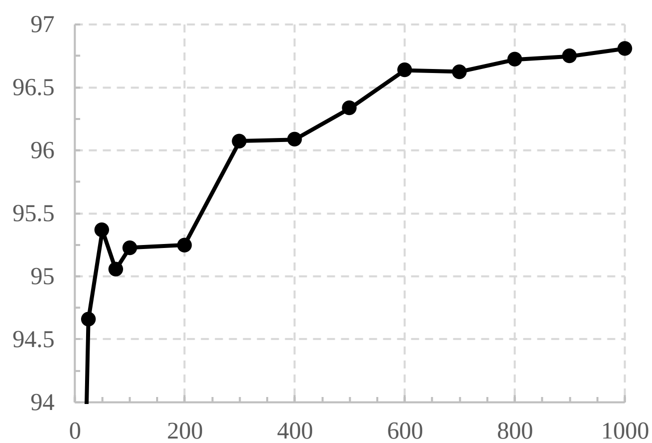

Figure 5: Number of Hidden Unit Neurons vs. Accuracy (%)

The activation functions chosen for the test represent the historically and modern day popular activation functions. The activation functions are shown in figure 6. High accuracy activation functions commonly indicate that the weights are easily differentiable. Optimizers can help converge the network to find the local minima of the cost or loss function to yield a high model accuracy. Figure 7 indicates that the Adaptive Moment Estimation (Adam) optimizer and RMSprop optimizer results in the highest NN model accuracy. The Adam optimizer is known for fast convergence while the RMSprop optimizer provides good management over the learning rate.

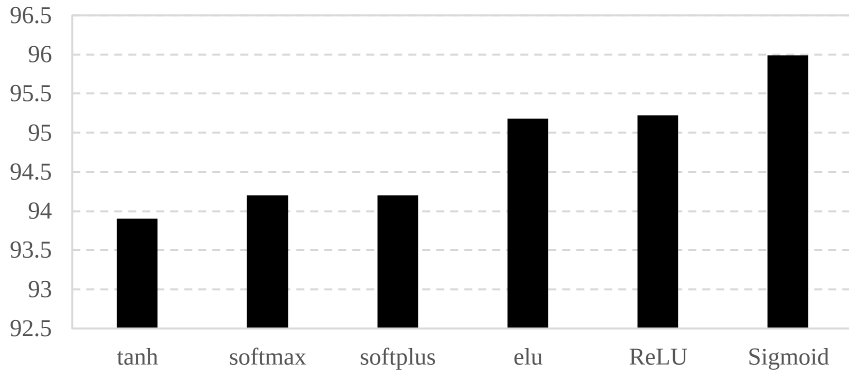

Figure 6: Activation Functions vs. Accuracy (%)

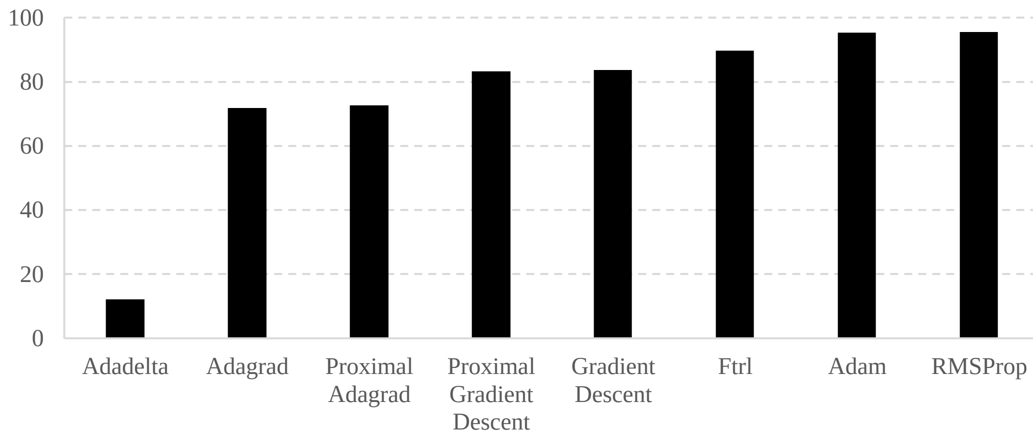

Figure 7: Optimizers vs. Accuracy (%)

The learning rate parameter influence the optimizer on how much to adjust the loss function. The loss function can be thought of as a parabola, where the local minima represent the lowest loss indicating a “perfect network”. The learning rate is then thought of as a ball moving down the parabola at some fixed increment. If the learning rate is too large the ball can keep overshooting the local minima, however if the step size is too small the ball may never reach the local minima. Figure 8. indicates with the default network parameters, a small to medium learning rate results in the best accuracy.

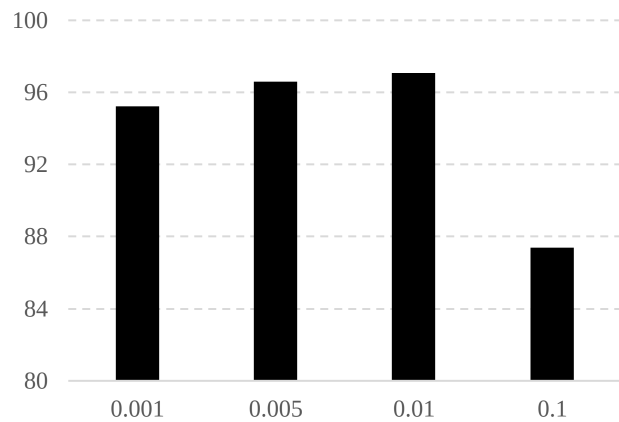

Figure 8: Learning Rate vs. Accuracy (%)

Increasing the number of hidden layers (100 hidden unit neurons per layer) caused a significant decrease in accuracy as shown in Figure 9. The decrease in accuracy could be caused by the default number of training epochs not being sufficient while increasing the total number of hidden unit neurons within the NN model.

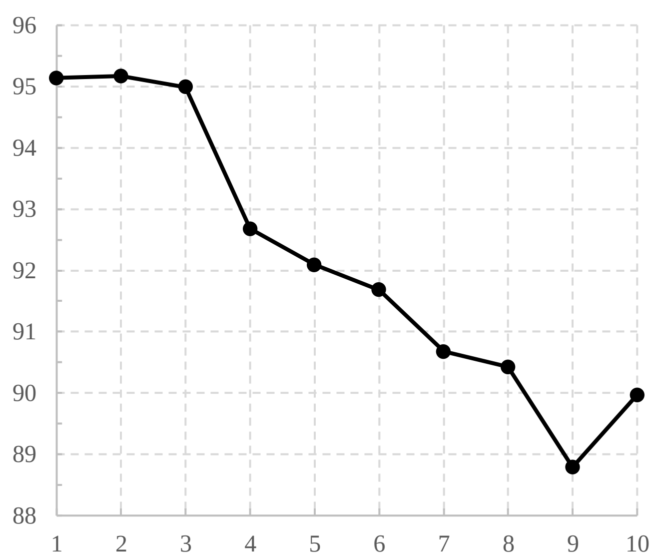

Figure 9: Number of Hidden Layers vs. Accuracy (%)

A batch size can be very influential in the time it takes to train a network. A batch size indicates how many images/input data is pushed into the network before the network updates it weights. A large batch size would have less number of weight updates in a fixed number of training epochs compared to a small batch size. A higher number of weight updates typically indicates a higher model accuracy, as shown in Figure 10.

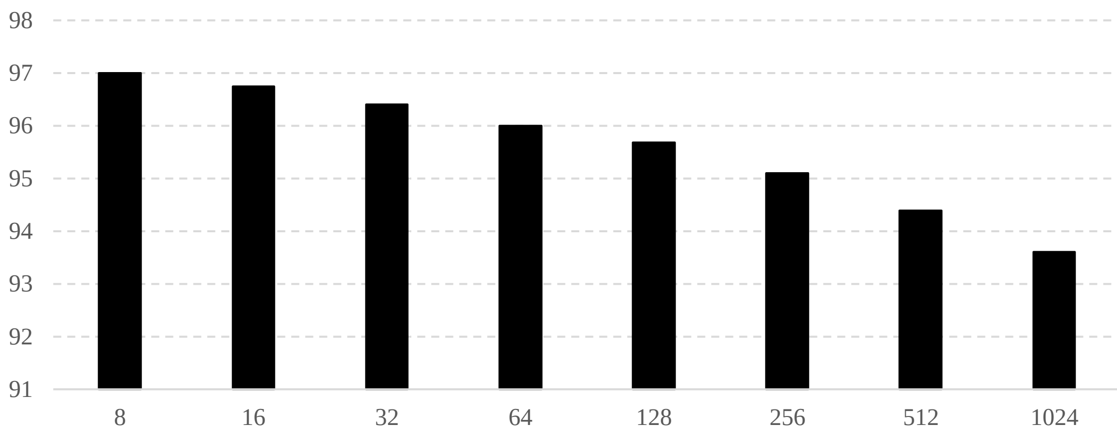

Figure 10: Batch Size vs. Accuracy (%)

The second set of tests were used to finalize the NN model in regard to training speed and accuracy. Each data point in Figure 11. represents one unique NN model. The red circle in Figure 11. shows various NN models that are the most desirable due to low training time and high accuracy. Figure 11 also indicates that model accuracy reaches steady state after 100 seconds of training time. The finalized NN model parameters shown in Figure 12, has an approximately 60 seconds training time while achieving 95%-97% accuracy on the MNIST test set. The finalized model is Jetson ready.

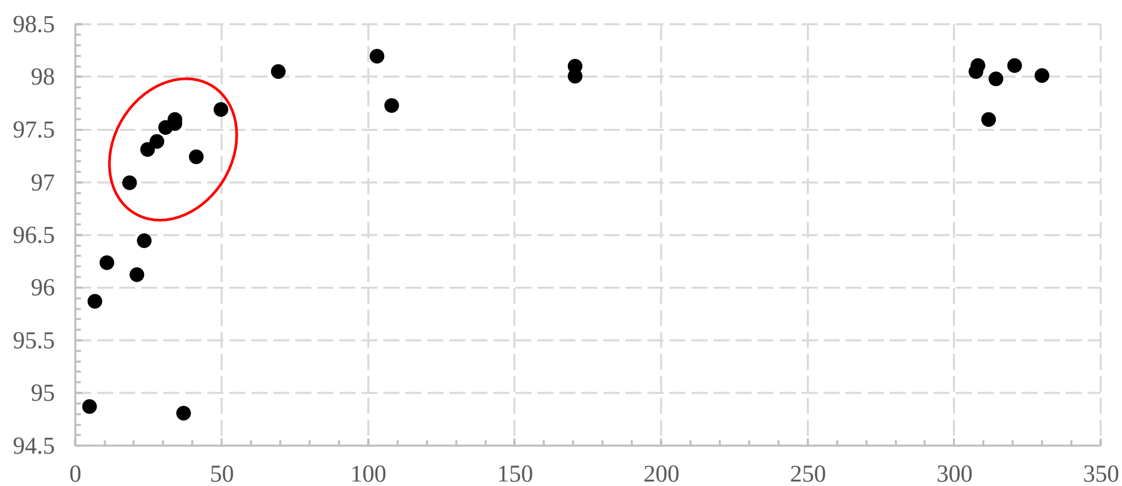

Figure 11: Runtime (s) vs. Accuracy (%)

Baseline Network	
Batch Size	1024
Training Epochs	100
Learning Rate	0.01
Optimizer	Adam

Hidden Units	800
Activation Function	ReLU
Number of Hidden Layers	1

Figure 12: Jetson Ready Model

DIGITS and Jetson

The NN model parameters from the finalized model were ported into Nvidia's DIGITS program to build a Tensor RT and Jetson Ready model that can be easily ported into Nvidia's existing image recognition demonstration. Tensor RT is Nvidia's inference program, which allow images to be recognized in real time. Once the Jetson Ready model was built, the model was swapped into the image recognition demonstration that setup the camera protocol and input data pipeline to the NN model.

C. Results and Discussion

The modified image recognition demonstration with the Jetson ready model for real-time live digit recognition work as shown in Figure 13. The upper right-hand corner on the monitor screen circled in red in Figure 13 shows the confidence level (accuracy) in percentage next to the classified digit. Figure 13 shows the handwritten digit 7 being classified in real time with the Jetson ready model being 100% confident that it is the digit 7.

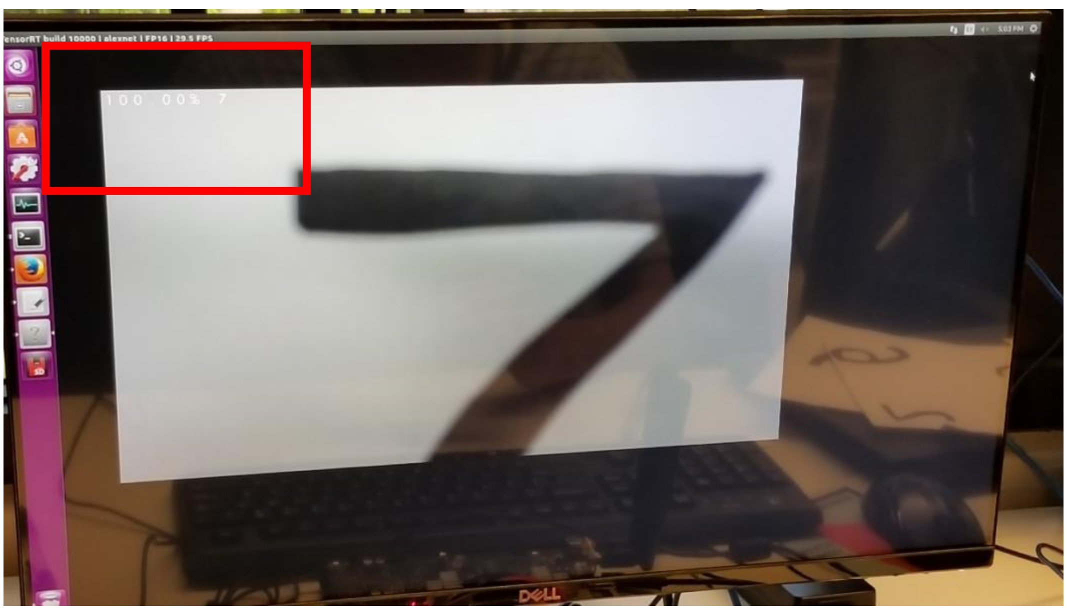

Figure 13: Digit 7 Classified by the Jetson Tx1

To test the robustness of the Jetson ready model in various situations such as inversed image colors, cluttered backgrounds, and line thickness, the model was shown various images of all digits. Figure 14 shows the Jetson ready model classifying the digit 8 with a happy face in the background at an 85% confidence level. However, through various test it was clear that the Jetson ready model was not robust enough. The model was highly dependent on thick black lines with a white background, the camera angle relevant to the hand drawn digit image, and the images had to be shown in a well-lit environment.

Figure 14: Jetson Tx1 classifying Digit 8 with a Happy face background

Designing a RTLDRS using the Jetson Tx1 proved to be a non-trivial summer internship project. Aside from the lack of robustness in the Jetson ready NN model, the summer interns completed the project that will be used for many years to come. This first implementation of a RTLDRS allows for future implementation of a robust network with better feature detection that can combat the issues of inverted colors, specific camera angles, lighting conditions, and so forth. The idea of switching to a more powerful NN architecture, such as a convolution neural network (CNN), may fix many of these issues but at a large cost of training time.

V. Assessment of the Research Internship Program

To evaluate the effectiveness of the ten-week engineering research internship, pre-and post-internship surveys were given to the all the student interns, including two civil engineering teams, two electrical engineering teams, one mechanical engineering team, and one computer engineering team. The survey was designed to measure the student interns' motivations and perception for cutting edge student research, academic goals, and skills needed for research and academic success. The survey shown in Table 1 summarizes the results for pre-and post-internship based on student motivation, expectations, and purpose for participating in the internship. Table 1 results indicate that the largest motivation for participating in the ASPIRES engineering internship was to gain hands-on experience in research, be challenged intellectually, and clarify whether graduate school would be a good choice for them. The interns found the program to be most helpful in learning how to work with others to plan and conduct scientific experiments, followed by talking to professors about science. The largest difference between pre- and post-internship survey was observed for gaining hands-on experience in research.

Table 2 summarizes the results of the pre-and post-internship survey based on the interns' perception of skills and knowledge needed for research and academic success. The highest scoring question in Table 2 was the students felt confident that they will transfer to a four-year institution after participating in the ASPIRES engineering internship. After completing the engineering internship, there were significant gains in the interns imaging themselves continuing beyond a Bachelor of Science degree towards a master's degree in a STEM field, that they were ready for more demanding research, and understood how scientists work on real problems.

Table 1. Results of the pre-and post-surveys of student motivation, expectations and purpose for participating in the internship program.

Pre-program Prompt: What do you most want to learn or gain from your internship this summer?

1 – Strongly Disagree and 5- Strongly Agree

Post-program Prompt: Please indicate the degree to which your internship experience helped you learn or gain each of the following.

1 being the LEAST helpful and 5 being the MOST helpful.

	Average Response		
Purpose of Internship	**Pre**	**Post**	**Change**
Gain hands-on experience in research	4.79	4.09	-0.69
Solidify my choice of major		3.56	
Gain skills needed to successfully complete a BS degree		3.88	
Clarify whether graduate school would be a good choice for me	4.15	3.69	-0.46
Clarify whether I wanted to pursue a STEM research career	3.79	4.06	0.27
Work more closely with a particular faculty member	3.58	3.75	0.17
Get good letters of recommendation	4.00	3.59	-0.41
Have a good intellectual challenge	4.55	4.34	-0.20
Read and understand a scientific report		4.03	
Write a scientific report		3.97	
Ask good questions related to the scientific process		3.97	
Set up a scientific experiment		3.56	
Work with others to plan and conduct scientific experiments		4.09	
Talk to professors about science		4.00	
Think like a scientist		4.03	

Table 2. Results of pre-and post-surveys on student perceptions of skills and knowledge for academic and research success.

Prompt: Please Indicate your level of agreement with the following statement.

1-Strongly Disagree and 5- Strongly Agree

Prompt	Average Response Pre	Post	Change
I was able to conduct the scientific research that is part of my summer internship.		4.28	
I am confident I will transfer to a four-year institution.	4.64	4.78	0.14
I am confident I will complete a BS in a STEM field.	4.55	4.69	0.14
I can imagine myself continuing after my BS to pursue a Master’s Degree in a STEM field.	3.85	4.38	0.53
I can imagine myself continuing after my BS to pursue a Ph.D. in a STEM field/Medical School/other education beyond the Master's level.	3.48	3.72	0.23
I have a clear career path.	3.94	4.16	0.22
I have skill in interpreting results.	4.09	4.22	0.13
I have tolerance for obstacles faced in the research process.	4.18	4.38	0.19
I am ready for more demanding research.	3.85	4.28	0.43
I understand how knowledge is constructed.	3.76	4.03	0.27
I understand the research process in my field.	3.42	3.81	0.39
I have the ability to integrate theory and practice.	3.76	4.00	0.24
I understand how scientists work on real problems.	3.70	4.13	0.43
I understand that scientific assertions require supporting evidence.	4.33	4.53	0.20
I have the ability to analyze data and other information.	4.09	4.25	0.16
I understand science.	4.12	4.28	0.16
I have learned about ethical conduct in my field.	3.97	3.84	-0.13
I have learned laboratory techniques.	3.76	3.78	0.02
I have an ability to read and understand primary literature.	4.12	4.06	-0.06
I have skill in how to give an effective oral presentation.	4.00	4.31	0.31
I have skill in science writing.	3.76	4.16	0.40
I have self-confidence.	4.27	4.22	-0.05
I understand how scientists think.	3.79	4.06	0.27
I have the ability to work independently.	4.33	4.50	0.17
I am part of a learning community.	4.33	4.34	0.01
I have a clear understanding of the career opportunities in science.	3.97	4.28	0.31

VI. Conclusion

The 2017 summer ECE community college team successfully developed a semi-robust fast training NN for live digit recognition and implemented the NN model on the Jetson Tx1 for real-time digit recognition on the go. The ASPIRES internship did not only expose community college students to leading edge research, but helped inspire, improve, motivate, and challenge each intern to continually improve their teamwork, technical writing, reading, and presentation skills. The ASPIRES engineering internship is an effective method to engage community college students in engineering research, to reinforce the intern’s confidence in perusing a higher education within a STEM field and to teach them the tools and methodology to be successful within a engineering profession.

Acknowledgement

This project was supported by the US Department of Education through the through the Minority Science and Engineering Improvement Program (MSEIP, Award No. P120A150014); and through the Hispanic-Serving Institution Science, Technology, Engineering, and Mathematics (HSI STEM) Program, Award No. P031C110159.

Bibliography

1. Bridle, J. S. (1989). Training Stochastic Model Recognition Algorithms as Networks can lead to Maximum Mutual Information Estimation of Parameters. *Advances in Neural Information Processing Systems 2*.
2. Ruder, S. (2017). An Overview of Gradient Descent Algorithms. *[1609.04747] An Overview of Gradient Descent Algorithms.*
3. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM, 60*(6), 84-90. doi:10.1145/3065386
4. Simard, P., Steinkraus, D., & Platt, J. (n.d.). Best practices for convolutional neural networks applied to visual document analysis. *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* doi:10.1109/icdar.2003.1227801
5. Schaul, Tom, Ioannis Antonoglou, and David Silver. (2014). Unit Tests for Stochastic Optimization. *[1312.6055] Unit Tests for Stochastic Optimization*.
6. Ian Goodfellow, Yoshua Bengio & Aaron Courville (2016). Deep Learning. *MIT Press*. doi:10.1038/nature14539
7. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *[1603.04467]*
8. Lecun, Y. A., Bottou, L., Orr, G. B., & Müller, K. (2012). Efficient BackProp. Lecture Notes in Computer Science *Neural Networks: Tricks of the Trade*, 9-48. doi:10.1007/978-3-642-35289-8_3
9. Stathakis, D. (2009). How many hidden layers and nodes? *International Journal of Remote Sensing*, 30(8), 2133-2147. doi:10.1080/01431160802549278