10-25-2019

A deep learning approach for motion segment estimation for pipe leak detection robot

Cihan Uyanik
Tennessee State University

Erdem Erdemir
Tennessee State University

Erkan Kaplanoglu
Marmara University

Ali Sekmen
Tennessee State University

Recommended Citation

Cihan Uyanik, Erdem Erdemir, Erkan Kaplanoglu, Ali Sekmen, "A deep learning approach for motion segment estimation for pipe leak detection robot", Procedia Computer Science, Volume 158, 2019, Pages 37-44, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2019.09.025.

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 158 (2019) 37–44

Procedia Computer Science

www.elsevier.com/locate/procedia

3rd World Conference on Technology, Innovation and Entrepreneurship (WOCTINE)

A deep learning approach for motion segment estimation for pipe leak detection robot

Cihan Uyanik[α], Erdem Erdemir[b], Erkan Kaplanoglu[c], Ali Sekmen[b]

[a]*College of Engineering, Tennessee State University Nashville 37011, USA*
[b]*Department of Computer Science, Tennessee State University, Nashville 37011, USA*
[c]*Department of Mechatronics Engineering, Marmara University, Istanbul 34000, Turkey*

Abstract

The trajectory motion of a robot can be a valuable information to estimate the localization of an autonomous robotic system, especially in a very dynamic but structurally-known environments like water pipes where the sensor readings are not reliable. The main focus of this research is to estimate the location of meso-scale robots using a deep-learning-based motion trajectory segment detection system from recorded sensory measurements while the robot travels through a pipe system. The idea is based on the classification of the motion measurements, acquired by inertial measurement unit (IMU), by exploiting the deep learning approach. Proposed idea and utilized methodology are explained in the related sections and it is observed that convolutional neural network approach is quite powerful to overcome the unreliability of IMU data.

Peer-review under responsibility of the scientific committee of the 3rd World Conference on Technology, Innovation and Entrepreneurship

Keywords: Motion Trajectory, Inertial Measurement Unit, Deep Learning, Convolutional Neural Network, Leak Detection Robot

1. Introduction

For almost any dynamic system, investigating of motion trajectory for a robotic system is very crucial process. In any application for almost any field, to acquire the motion and extract valuable information from the measured data are only possible by accurate sensor retrieval and successive data analysis. For instance, industrial and daily life activities are very important areas for the motion trajectory analysis and have strong demand on usage inertial

* Corresponding author. *E-mail address:* cuyanik@my.tnstate.edu

1877-0509
Peer-review under responsibility of the scientific committee of the 3rd World Conference on Technology, Innovation and Entrepreneurship
10.1016/j.procs.2019.09.025

measurement unit (IMU) [1]. Affordable, tiny, and energy efficient sensors such as IMU are also very popular in many practical applications especially in robotic field, although the problematic measurements acquired. In this paper we have applied a deep learning-based approach to determine robot motion trajectory via segmenting the traveling domain and performing classification procedure to each segment. The applied methodology directly utilizes convolutional neural network as a classifier for taken raw measurements and generates analyzed output for IMU measurement time instance as a ground location information of the traveling device.

1.1. Smart Mechatronics Pipe Leak Detection Systems

In the literature, there are many studies to specific issues such as water distribution, energy efficiency [2]. To overcome many problems, detection dedicated robots are considered an efficient solution to investigate the pipe quality and possible leakage detection [3]. Navigating a robot inside a pipeline system is a crucial challenge and IMU devices are widely utilized [4]. In this paper, we have investigated the applicability of a deep learning machine learning approach to acquire the motion trajectory information of a meso-scale system which is aimed to move inside a pipe system and collect measurements for future investigations about the pipe health conditions. The most critical hardness of this kind of systems are size and energy restrictions. For such small devices, it is not possible to attach highly accurate environment measurement devices such as cameras, laser devices. The environment itself also does not allow the track the motion of the robot externally. These issues force the system to utilize a small in size, low energy consumption measurement instruments such as inertial measurement unit (IMU). However, the most crucial problem of IMU devices is the measurement accuracy. The sensors itself are so noisy and taken measurements highly possible to be unacceptable especially in long time periods.

1.2. Deep Learning in Machine Learning

The ability to represent the information through network layers and self-extraction of valuable features on the data pushes the literature to consider deep learning as state of art approach in machine learning [5],[6]. Through the self-data abstraction ability, the data given as input is transformed into inner representation in the network [7]. From all over the world researchers and companies are attracted by the ability of layer-wise abstraction and transformation of the information. These situation leads to develop several neural network models [8],[9],[10],[11],[12] for diverse areas such as computer vision, speech recognition, medical investigation [13], robotics etc.

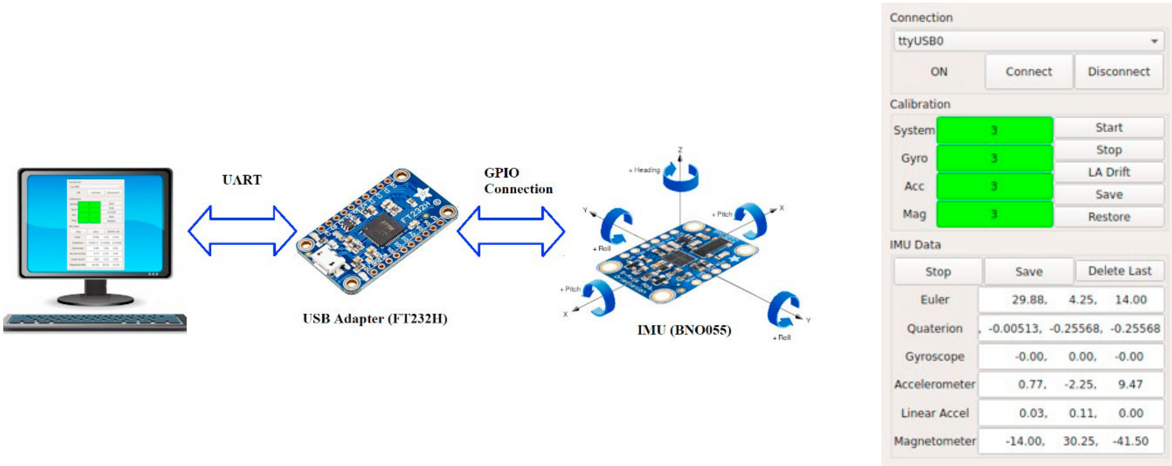
2. Experimental Study

2.1. Hardware Structure

To implement a motion segment type determination system which exploits the deep learning machine learning approach, the beginning step is build a sensory measurement setup. As indicated in the previous section, in this study, we have utilized an IMU device as a core measurement instrument. The main criteria on choosing an IMU is that the device utilized should be in low dimensional, low energy consumption. BNO055 [14] is one the most popular, highly reachable, cheap IMU device and supports I2C, SPI and serial UART communication, which allows to configure the system to utilize in different conditions and several master data retrieval devices such as single board computers like Raspberry PI. In this research, we have decided to carry the experiments via UART connection, which allows 63 Hz data acquisition rate. Also, a serial connection helper instrument (FT232H [15]) is necessary to adapt the data flow port of IMU device into the computer USB port (See Fig. 1a).

2.2. Data Collection Helper Program

When the hardware configuration successfully performed, a helper software is required to be implemented to safely acquire the motion data measurements taken from IMU device and record these measurements. The software is developed in Python 2.7 and a graphical user interface (GUI) (QT development kit) is implemented to manage the data acquisition and storing processes (See Fig. 1b). The GUI is capable of managing the connection between the measurement instrument and creates an opportunity to the user to perform successful accelerometer, gyroscope, magnetometer calibration process. The individual sensors inside the IMU requires easy to perform calibration procedures and the GUI demonstrates the health status of these calibration process in real time. Also, it allows to reload last successful calibration status. Finally, it begins and finalizes the measurement operation while storing them into storage device.



(a) (b)

Fig. 1. (a): Connection schema for hardware setup, (b): Data collection helper user interface

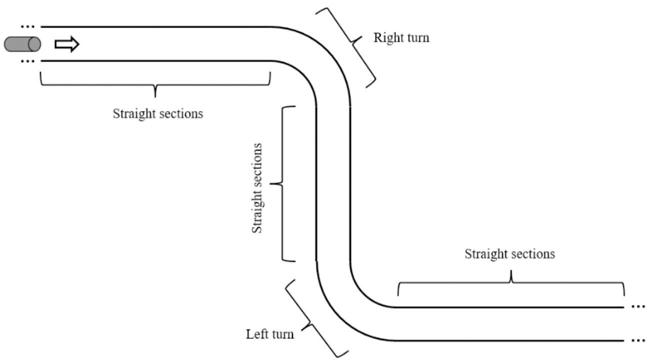


Fig. 2. General pipe structure

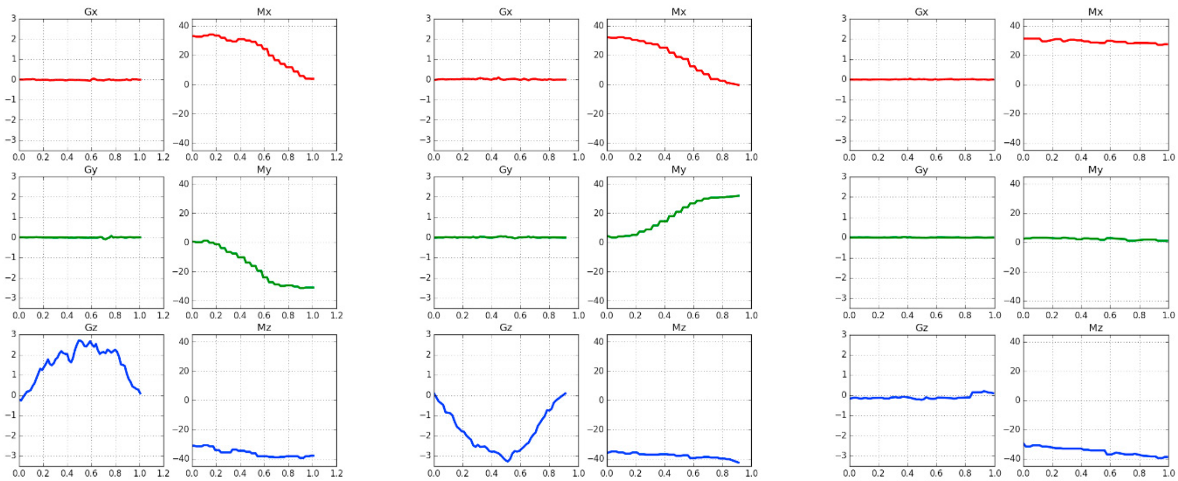
2.3. Training Data Collection

To construct a training dataset for a pipe line motion segment detection system, it is required to define a general pipe structure which the leak detection robot travelling inside. To make a general assumption for the proposed idea a pipe line system could be considered as constructed by finite geometric shapes. The Fig. 2 demonstrates a general pipe structure in upper side perspective. The structure simply constructed by straight segments and bending segments. From the travelling robot's point of view, also from the core motion measurement device (IMU), the traveling path contains straight segments, right turning segments and left turning segments. Due to this finite state geometric constraint on a pipe structure, it possible to separate measurements taken from IMU device, which also represents the traveling path of the robot inside the pipe, into 3 different classes. This idea leads to generate a training data set containing these 3 classes, straight, left turn, right turn. To build a training data set for CNN, the hardware setup explained in the previous section is manipulated to perform all the given classes separately 50 times for each class, which is totally 150 training sample of motions. All collected training samples have different length of duration, variating from 1 sec to 2 secs to simulate the velocity changing behavior of the motion.

2.4. Data Preparation

When the training dataset collection is completed, a data preparation process is conducted to obtain a standardized data structure from the raw measurements of IMU device. In this process, the main goal is to generate stacked time graphs representations for each measurement axis of the gyroscope and magnetometer. This process yields 6 channel representation of each performed training motion. While motion is carried, the IMU measurements are stored with the related time instance, due to the known time frames, it is possible to plot the motion into a time series graph for any measurement variable. These time series plots create a separable characteristic for each motion type defined above. The most important aspect of time graph plotting is value axis scaling for different kinds of graphing variables. If any graph plotting tool is utilized as default drawing characteristics, it is obvious that if any measurement range contains only small measurement changes, the resulting time graph could be magnified. To overcome this issue, it is necessary to define a drawing range for value axis. This information could be easily obtained via statistical analysis on training dataset. The process is based on finding the maximum and minimum measurement ranges of each measurement sensors (gyroscope, magnetometer).

After finding the value axis drawing ranges, the raw data of the IMU device is drawn into time graphs and these time graphs are stacked into $MxNx6$ matrix as a training input samples, where M and N are the image height and width respectively, 6 corresponds to measurement variables, 3 axis gyroscope (G_x, G_y, G_z), 3 axis magnetometer (M_x, M_y, M_z). The Fig. 3 shows the characteristics of the time graphs for each classes. Unfortunately, stacked version these images could not be visualized as a regular image.



(a) (b) (c)

Fig. 3. Training data samples, (a): left turn, (b): right turn, (c): straight

2.5. Convolutional Deep Neural Network Structure

The next phase of the process is to build a Convolutional Neural Network (CNN) model and training this model by using the data acquired. The prepared CNN model is given in the Table 1. The model generated with convolution layers to detect the data features. A $48x48x6$ data file is given to the model, the data is forwarded through successive convolution layers. 64 different filters with $11x11$ kernel size and *same* padding is chosen as first convolution layer structure, following by a non-linear activation function *ReLU*. The outcomes of the $ReLU$ activation layers fed into maximum pooling layer with $2x2$ pooling and $2x2$ stride. A dropout procedure with 0.25 drop-out rate is executed over the pooling layer to decrease the possibility of over-fitting of the model. Next two convolutional blocks follow the same structure with higher filter count and smaller kernel size to enrich the feature representation. Third convolutional layer output is utilized as input of a 512 units densely connected layer within *ReLU* activation. Also, similar over-fitting protection strategy is exploited with 0.5 drop-out rate. Finalization layer, which is also named as classification layer, is constructed with 3 fully connected units with a *Softmax* activation as output.

Table 1. Convolutional deep neural network structure

Layer	Description
Input	48x48x6
Convolution	64 filters of size 11x11
Activation	ReLU
Max Pooling	2x2 Pooling with 2x2 stride
Dropout	0.25 cut ratio to increase generality
Convolution	128 filters of size 5x5
Activation	ReLU
Max Pooling	2x2 Pooling with 2x2 stride
Dropout	0.25 cut ratio to increase generality
Convolution	256 filters of size 3x3
Activation	ReLU
Max Pooling	2x2 Pooling with 2x2 stride
Dropout	0.25 cut ratio to increase generality
Fully Connected	Fully connected dense layer with 512 units
Activation	ReLU
Dropout	0.5 cut ratio to increase generality
Fully Connected	Fully connected dense layer with 3 units
Activation	Softmax

The training phase of the CNN model explained above for the prepared training data set, we have obtained perfect classification accuracy on the test set, which is randomly separated from the training data. The related information of the training results is given in the following tables.

Table 2. Confusion matrix

	Cylind.	**Hook**	**Pinch**
Left	**8**	0	0
Right	0	**8**	0
Straight	0	0	**8**

Table 3. Precision, Recall & F1-Score

	Precision	Recall	F1-Score	Support
Left	1	1	1	8
Right	1	1	1	8
Straight	1	1	1	8

2.6. Leak Detection Robot Pipe Travelling Segment Detection

To verify the proposed idea real time experiments is also performed, the core expectation on these experiments is that the CNN model trained with the collected training dataset should distinguish the long-term motion measurements taken from IMU through a pipe path successfully. The separation of an individual measurement, that is labeling it, yields a successful identification of the robot location inside the pipe for a given time instance (See Fig. 4).

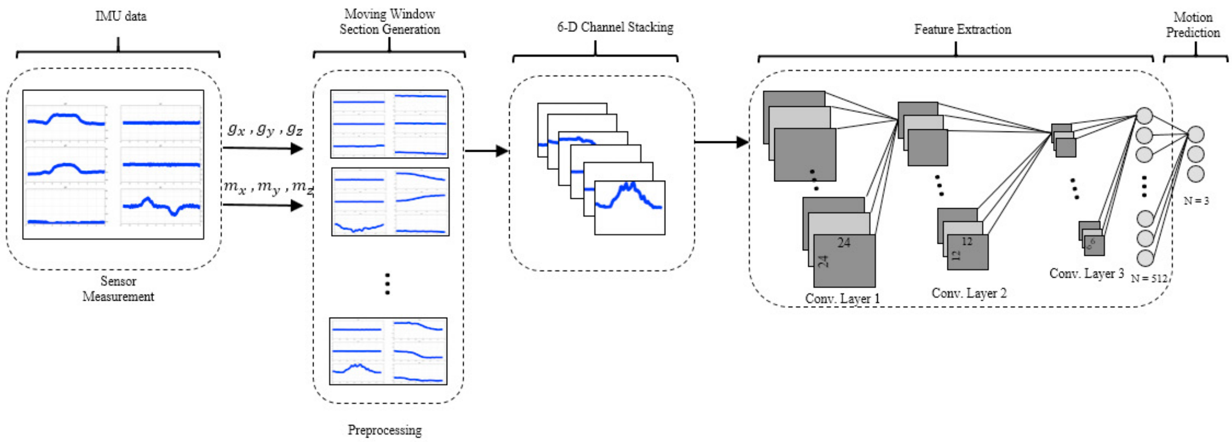


Fig. 4. Leak detection robot pipe travelling segment detection system

Consider the given example pipe structure in Fig. 2. For this pipe structure, the generated model supposed to identify the motion segments while labeling each measurement class. However, the most prominent aspect this observation is that, we cannot directly feed the long-term measurement data into the neural network model, also we need to label each measurement sample instance separately. To generate this output, we have applied a moving window segmentation approach onto the overall measurement. The most important parameter of this approach is to determine a proper window size. The window size could be easily calculated by observation taken from the training motion dataset sizes. By iterating through the time on the given test motion and performing a window segmentation by considering the current time instance as a window center, we can successfully generate motion segments. After that step each segment could be feed into training CNN model to classify the segment motion and label the measurement corresponding the window center. The resulting labeled measurements of the motion is shown in Fig. 5. In this figure, all motion measured for the given pipe structure, which contains straight sections, right turn, straight sections, left turn, straight sections in order. The expectation from the CNN motion segment classifier is to determine the motion type for each measurement by looking limited region, which is defined as window size. As shown in the given figure, classifier successfully identifies the sections and labels them into right class.

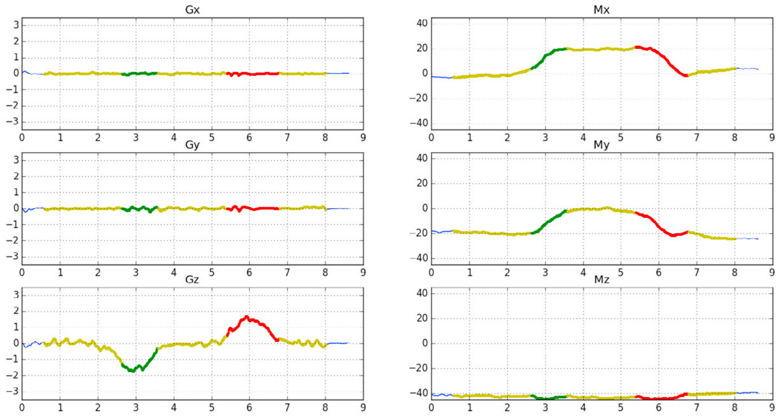
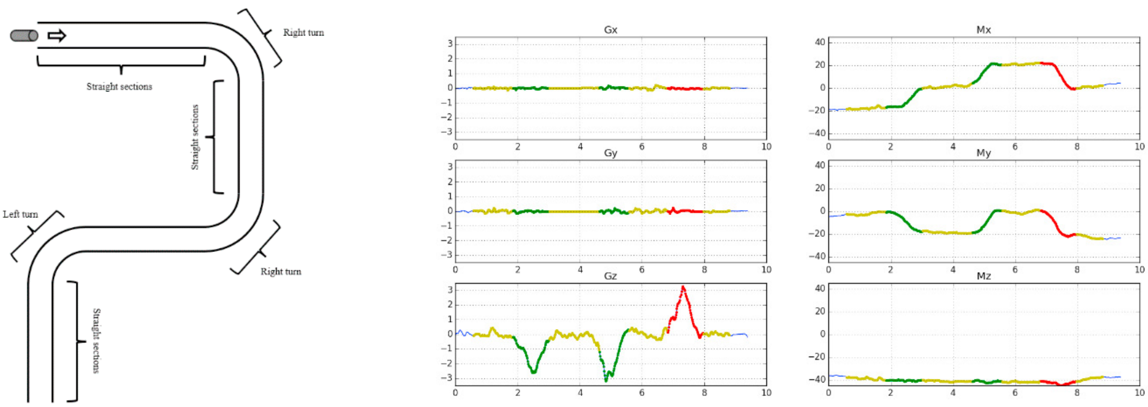


Fig. 5. Motion measurement and measurement instance classification: yellow = straight, green = right turn, red = left turn

As another test application for the given approach, we can consider the pipe structure given in Fig. 6a. In this test case, the pipe contains straight sections, right turn, straight sections right turn, straight sections, left turn and straight sections as shown in the figure. It obvious that the CNN model expected to identify the motion in segments for each measurement by using the explained approach. Corresponding result is shown in Fig. 6b.



(a) (b)

Fig. 6. (a): Pipe structure, (b): Corresponding motion measurements and measurement instance classification: yellow = straight, green = right turn, red = left turn

3. Conclusion

In this paper, a motion trajectory of a pipe line detection robot which is supposed to travel inside an unconfigurable environment is segmented into predefined subsections by using a small, low-cost inertial measurement unit. The segmentation and labeling each segment through the pipe line motion is performed by one of the most popular machine learning approach, deep learning. The proposed idea depends on the generation of artificial time graph images for the considered sensor measurement variables. The created time graphs (related to the measurement instance) successfully fed into the trained convolutional neural network model. The classifier is responsible to distinguish each individual measurement's motion type, straight, left turn, right turn, by analyzing the segment which takes the interested measurement point as its center. This approach allows the meso-scale system to obtain knowledge, within time frame information, about the traveling action through a pipe line system.

Acknowledgements

This work is partially supported by the National Institute of Food and Agriculture (NIFA), United States Department of Agriculture (USDA) Capacity Building Grant under award number CBGP-005696.

References

[1] O. Dehzangi, M. Taherisadr, and R. ChangalVala, "IMU-Based Gait Recognition Using Convolutional Neural Networks and Multi-Sensor Fusion," *Sensors*, vol. 17, no. **12**, p. 2735, 2017.

[2] A. M. Sadeghioon, N. Metje, D. N. Chapman, and C. J. Anthony, "SmartPipes: smart wireless sensor networks for leak detection in water pipelines," *J. Sens. Actuator Networks*, vol. 3, no. **1**, pp. 64–78, 2014.

[3] D. Wu, D. Chatzigeorgiou, K. Youcef-Toumi, and R. Ben-Mansour, "Node localization in robotic sensor networks for pipeline inspection," *IEEE Trans. Ind. Informatics*, vol. 12, no. **2**, pp. 809–819, 2016.

[4] J. Zheng, S. Wang, A. Hazim, and S. Dubljevic, "Pipeline Leak Detection Swimming Robot Design and Deployment," in *2018 Annual American Control Conference (ACC)*, 2018, pp. 1166–1171.

[5] L. Deng, D. Yu, and others, "Deep learning: methods and applications," *Found. Trends®in Signal Process.*, vol. 7, no. **3**--4, pp. 197–387, 2014.

[6] A. Sekmen, A. B. Koku, M. Parlaktuna, A. Abdul-Malek, and N. Vanamala, "Unsupervised deep learning for subspace clustering," in *Big Data (Big Data), 2017 IEEE International Conference on*, 2017, pp. 2089–2094.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. **7553**, p. 436, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[9] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, 2014, pp. 818–833.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.

[11] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. **7639**, p. 115, 2017.

[14] "BNO055 Overview." [Online]. Available: https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor/overview. [Accessed: 08-Apr-2019].

[15] "FT232 Overview." [Online]. Available: https://learn.adafruit.com/adafruit-ft232h-breakout/overview. [Accessed: 08-Apr-2019].