Full Body Beamsteered Wi-Fi Imaging and Threat Classifier for Public Transportation

Paul C. Proffitt
ECE Dept
University of Massachusetts
Dartmouth, MA, USA
pproffitt@umassd.edu

Honggang Wang, Ph.D.
ECE Dept
University of Massachusetts
Dartmouth, MA, USA
hwang1@umassd.edu

***Abstract*—With the population of the world increasing substantially decade after decade, the probability of terrorist attacks on people will increase. Detecting terrorists with dangerous objects is necessary. Airport security is very good, but how about other forms of public transportation such as subways and city buses? More threat detection is expected in these areas. Our research work focuses on developing a system made of two main parts to detect and classify dangerous objects placed on the chest of terrorists. The first part is made up of Wi-Fi imaging. Images are created using Wi-Fi signals scanning a person head-to-toe with a 2x2 antenna array. The second part uses a Region-based Convolutional Neural Network to classify whether the Wi-Fi image contains a dangerous object or something harmless such as a tablet. Our results demonstrate the feasibility of the proposed system. Also, our system only uses one out of many channels used in the Wi-Fi bandwidth, which allows for the remaining channels to be used for users' Internet applications on public transportation.**

***Index Terms*—Wi-Fi imaging, threat detection, classification, neural networks, public transportation**

I. Introduction

The researchers of this paper developed a project where Wi-Fi is used to image and classify dangerous and non-dangerous objects that could enter a public bus or subway on a person's body. Most public transportation systems have a turnstile where a passenger must pay for entry. At this same access point, our system could be installed, which would deny entry, lock the turnstile, to anyone that was detected with dangerous objects such as C4 explosives, pistols, or any other homemade explosive strapped to a terrorist's chest. Upon denied entry, subway security or a bus driver could require a passenger expose what they are concealing. This design is much cheaper than an airport security millimeter-wavelength setup, which has cost close to a quarter of a billion dollars and requires a lot of human interaction. Plus our system allows for the implementation of Internet access on the many spare Wi-Fi channels.

This project is made up of three main stages. The first is the Wi-Fi hardware and software to process transmit/receive signals against the passenger. The data is stored upon completion of a scan by the 2x2 receive antennas. The second stage takes the stored data and creates images via MATLAB, which could be easily replaced with a real-time image creator written in C++. These images are fed into the third stage, Detectron. Detectron is an implementation of the Mask Region-based Convolutional Neural Network (Mask-RCNN) [1, 2]. Out of Detectron comes the probability of what is strapped or held against their chest. These items are: 1) an innocent person holding a tablet, 2) a person holding a pistol, 3) a person with five soda cans strapped together containing liquid explosives, 4) C4 explosives, or 5) an innocent person with nothing on chest.

II. Background

Prior to this work a thorough look into past Wi-Fi research was conducted. Using Wi-Fi for other things beside Internet access became popular with [3]. This paper used Wi-Fi to track moving people and gestures. Many other motion tracking papers followed. Our research is different in that we are detecting static, relatively non-moving objects. We are not concerned with what a person is doing but with what they may have strapped and concealed against their chest.

One paper which did scan static objects was [4]. It used optical equations to generate images of objects. Though it did create images, their setup required moving the antennas from location to location to generate the overall image. Ours is a small 2x2 receive antenna array and a 1x2 transmit antenna array fixed in place. [4] stopped at creating images and did not attempt to classify the images. Also, it used 2.4 GHz instead of a frequency closer to 5 GHz, which would reflect better.

Another paper which looked at scanning relatively static objects on a conveyor belt was [5]. They used an array of omni-directional antennas to scan objects. Using an array of omni-directional antennas will give you measurements from both sides of the array. It will not tell you from which direction the measurement is coming from; it could be from the back or the front. Usually, array systems with omni-directional antennas use other sensors, which could be other independent antennas, to determine the direction of the approaching signal. These weren't used in [5]; thus, we assume [5] insured no other objects were in range of the omni-directional antennas except for the object of interest, being the conveyor belt items. Without these extra sensors it's hard to determine if the signal is from either side. Our setup uses small directional antennas that we designed from scratch for our transmit and receive signals, which guarantees we know the direction of our signals.

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

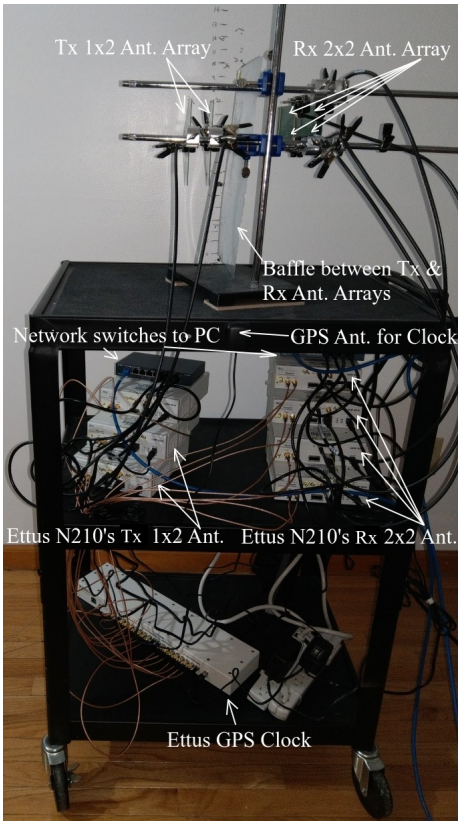
Also, our antennas use a higher frequency closer to 5 GHz, which has two benefits. One, we can create narrower beams than the 2.4 GHz used in [5] for measuring. Two, a higher frequency penetrates objects less and reflects better than 2.4 GHz. Lastly, [5] was used against objects only, not humans with the object. Humans add breathing and motion which we consider in our research.

Years ago, buses required a passenger to climb several steps where a 1x2 steered antenna array system would work well as the passenger passed through the measurement field. Now, most public buses only have one short step onto the bus with a large flat entrance area. This new bus design requires our receive 2x2 antenna array to scan the passenger from head-to-toe. We still use a simple 1x2 wide-beam transmit antenna array to light up the passenger, but our 2x2 antenna array supports present day applications.

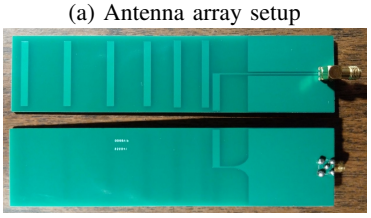
In a subway, the only chance to scan a passenger prior to entrance is at the paying turnstile, which is also a flat area. Scanning the passenger as they pass through the turnstile with a 1x2 receive antenna design would be too late as the passenger has already gained access. Using our 2x2 antenna array to scan a passenger head-to-toe prior to unlocking the turnstile would be better.

Millimeter (mm)-wavelength processing has become the popular method to measure objects. It has one major problem; it uses a very wide 2 GHz bandwidth as in [6], which is too much data for modern computers to handle at close range. Several mm-wavelength methods have been researched, but they all have real-world implementation hurdles since they use hardware oscilloscopes, spectrum analyzers, simulation models, or reduced bandwidths down to Wi-Fi data rates to gather their data as in [7], [8], [9], [10], and [11]. Airport passenger scanners take advantage of the smaller wavelength, but that's it. The high data rate is still too much to process for finite measurements at close range and require a TSA agent to determine if the object is harmful. The cost of airport mm-wavelength scanners is enormous at over $150,000 per unit with a poor track record as [12] discusses.

Our research is a balance between affordability, abundance, and practicality. Wi-Fi systems are much cheaper; thus, they can be made abundant unlike expensive airport mm-wavelength scanners. Finally, Wi-Fi can't scan tiny objects, but a box cutter on a plane does a lot more damage than it can do to passengers in a bus or subway.



(a) Antenna array setup



(b) A single, two-sided directional antenna of the 2x2 array

Fig. 1: Setup

III. System Setup and Execution

A. Antenna Array Setup

The hardware of stage one of our research is shown in Fig. 1(a). The figure labels the parts in the image.

The bottom of the figure shows an Ettus GPS clock. This clock is a single frequency reference for all the Ettus N210's especially for the 2x2 receive antenna array. Also, it provides a Pulse Per Second (PPS) signal to synchronize the sample time across the N210's. We found the Ettus GPS clock to be much more reliable than the Ettus MiMo cables, and we highly recommend the clock for your own research over the inexpensive, unreliable MiMo cables. MiMo cables only support 2 N210's while we are using 7 N210's with all clock inputs driven by this one Ettus GPS clock.

The middle of the figure shows the N210's. The right set is a group of four to support the 2x2 receive antenna array above it. Also, resting on top of these N210's is a network switch. It along with each receive N210 has a dedicated domain IP address, a 192.168.10.X address. The left N210's support the 1x2 transmit antenna array above it. There is an extra N210 in this stack used for calibrating the antennas to be discussed soon. This stack has a 192.168.20.X group of addresses.

The top-left of the figure is the transmit 1x2 antenna array, and to the right is the receive 2x2 antenna array. The transmit 1x2 antenna array is made of the common log-periodic directional antennas that Ettus sells. These were sufficient in our transmit work since they didn't violate the $\lambda/2$ spacing we needed between antennas. The wide beam coming out of these antennas are constantly pointed at 10 degrees to light up

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

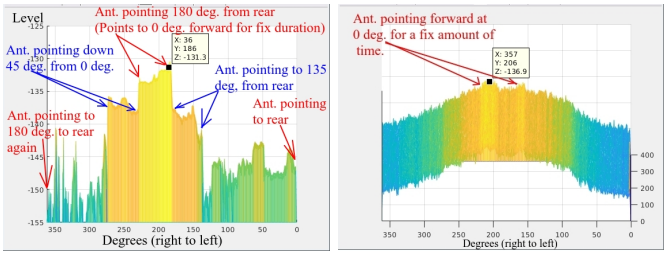
the person and object.

Now the upper right side of the figure is the receive 2x2 antenna array. In order to get the stacked antennas at $\lambda/2$ spacing we had to design our own antennas. We based our antenna on the common Yagi antenna. Our design is a two sided PCB board as shown in Fig. 1(b). We based our design on [13] and [14], which is based on the popular [15] text. Our design is for our own frequency and gain. Our antennas support a 4.3 GHz transmit and receive signal. Why 4.3 GHz and not 5 GHz? The Ettus N210 internal daughtercards only support up to 4.4 GHz so based on the Ettus N210 performance datasheet we chose 4.3 GHz. Since this frequency is lower than 5 GHz we would only get better performance if we moved up to 5 GHz someday. 5 GHz has a narrower beam and has better reflection off objects due to its shorter wavelength; thus, we don't feel running with 4.3 GHz would misrepresent 5 GHz. Yagi antennas have a very tight frequency range; thus, they must be used for the designed frequency and not some wider frequency range like the log-periodic antennas support.

Before using our antennas we measured their frequency response with a spectrum analyzer, which landed the center frequency at 4.3 GHz. Also, we measured the beam pattern of our antennas to insure when we steer the beams they are actually pointing where we want them. This is what the third Ettus N210 on the left stack of Fig. 1(a) supported. To measure the transmit pattern of our Yagi design, we put an omni-directional receive antenna at one meter from our rack. A relay board connected to our PC and controlled by our GNU Radio software would click the relay on/off to tell us to move the single Yagi to the next vertical or horizontal angle being from +180 to -180 degrees in increments of 45 degrees. The antenna would stay at that increment until the next relay click. Fig. 2(a) shows the levels of the Yagi antenna as it was rotated vertically from pointing 180 degrees to the rear initially. The antenna was rotated from 180 degrees at rear to 45 degrees up. Next, it was pointed to 90 degrees straight up and then 135 degrees forward. When the antenna was at 180 from rear, it was pointing forward directly at the omni-directional receive antenna. At this position you see the strongest level in Fig. 2(a); thus, this tells us our antenna is pointing its energy correctly. The figure continues to show we finished the rotations back to 180 degrees to rear. These steps were performed for horizontal, too. Fig. 2(b) shows the beam levels when the antenna was steered horizontally from rear 180 degrees, to forward at 0 degrees, and back to 180 degrees.

Now that we were confident our individual Yagi antenna design could transmit in the correct direction, we tested the 2x2 Yagi antenna array receive beam levels. We set the omni-directional antenna at center at 1 meter from the array and set to transmit. We then used our 2x2 array equations to steer from top to bottom and right to left. The max degrees were +90 to -90 degrees for vertical and horizontal steering. We will look at beamsteered results soon.

Safety when transmitting microwave signals is always a health concern. A microwave oven produces up to 1250 Watts of power cooking food immediately. A cell phone next to your



(a) Vertical (b) Horizontal

Fig. 2: Transmit beam levels of our Yagi antenna design

head and only millimeters from your brain produces about 1 Watt of power. A Wi-Fi router only produces about 100 milli-Watts of power. Our system only produces about 30 milli-Watts of power; thus, our system and Wi-Fi routers are much safer than using a cell phone next to your head.

B. GNU Radio

The software that controls the antenna arrays is a vital part of this system. Many people prefer to use MATLAB Communication Toolbox due to its ease of use and instant changes allowed by its scripting language, but to run real-time we chose the highly recommended GNU Radio. GNU Radio is written in C++ with a Python drag and drop flow block interface similar to Simulink. Within the C++, we wrote all of our own routines except for support tools like oscilloscopes, spectrum analyzers, and the drivers that talk to the N210's. Our system contains its own blocks for transmission processing, beamsteering, gain control, receiver processing, and so on. Our GNU Radio design is shown in Fig. 3. All the blocks with the word "my" contained in the name are the blocks we developed from scratch. Some blocks contain thousands of lines of C++ code we wrote. The blocks shown in dark gray are disabled blocks but are necessary for some of our low level, unit testing. The system goes through a full calibration prior to each run to make sure each antenna is at the same transmit/receive level. From experience the N210's and antennas will be slightly different in gain from other N210's and antennas in the system; thus, the calibration phase is necessary.

After the calibration phase, the transmit 1x2 array steers to 10 degrees to light up the object of interest. We use (1) from the very popular [16] text to steer the signal where β is defined in (2) and θ_0 = 10 degrees. Yes, (2) has a minus sign or else your signal would be steered to an unexpected location as we measured. We verified the transmit steered angle by using a receive omni-directional antenna placed at 10 degrees and 1 m away. We plotted the signal strength like we did in Fig. 2(b) which showed its strongest peak at 10 degrees.

$$x = \sum_{n=0}^{M} e^{jn\frac{2\pi}{\lambda}d(cos(\theta)+\beta)} \tag{1}$$

$$\beta = -cos(\theta_0) \tag{2}$$

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

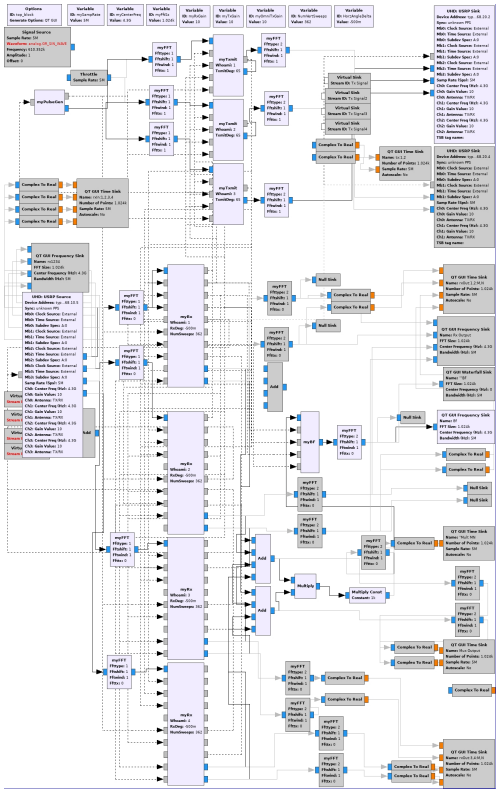


Fig. 3: Our tx 1x2 array and rx 2x2 array GNU Radio design

The receive 2x2 antenna array steering uses the planar array equation in [16] which is (3) with β_1 as (4) and β_2 as (5).

$$x = \sum_{n=0}^{M} e^{jn\frac{2\pi}{\lambda}d((cos(\psi)sin(\theta)+\beta_1)+(sin(\psi)sin(\theta)+\beta_2))} \quad (3)$$

$$\beta_1 = -cos(\psi_0)sin(\theta_0) \quad (4)$$

$$\beta_2 = -sin(\psi_0)sin(\theta_0) \quad (5)$$

The receive 2x2 antenna array starts its steering at horizontal angle of +90 degrees and a vertical angle of -90 degrees. In our case -90 degrees vertically is pointing up and +90 degrees is pointing down. We sweep the horizontal from +90 to -90 degrees. After one sweep, we move the vertical to -89 degrees. Specifically, we move the increment in even smaller increments more like 0.5 degrees; thus, the next increment sets the vertical to -89.5 degrees. We continue these horizontal sweeps from +90 to -90 until the vertical angle reaches +90 degrees and the run is done. These steering angles go into our ψ_0 and θ_0. In order to obey the Spherical Equation of [17] and many other math books, which is where (3) is derived from, we must adjust these +90 to -90 horizontal angles to be between 0 and 360 degrees. If the vertical angle is negative, we make the vertical angle positive and horizontally steer from 360 down to 180 degree. If the vertical angle is positive, we horizontally steer from 0 to 180 degrees. Since we are using directional antennas and already pointing our beams in the correct direction, unlike omni-directional antennas, we set θ and ψ to 0 degrees and control the steering with β in all our equations. As proven earlier through calibration measurements, these equations hold true along with the laws of math.

C. Signal Image Creation

The receiver array scans all the vertical angles, and at each vertical angle, the receive beam steers horizontally from right to left, +90 to -90 degrees. Each horizontal scan we call a "slice". Over the entire run, we collect all these horizontal slices and store them in a buffer to be written to a .dat file at the end of the run. Not only do we save slices, each angle we sample at is also stored. During post run processing, we slap all these slices together to form the images. We use the angles we stored to help reconstruct these slices.

A receive, beamsteered slice resulting from the four 2x2 receive channels is shown in Fig. 4 prior to extracting it from the entire frequency band. We are only concerned with each slice and nothing else in the band so we throw out the rest of the band and only keep the slice for image creation. The remaining portions of the band are freed up, which makes room for plenty of Internet surfing channels to be used by passengers.

Fig. 5(c) is an example of image creation. Here all the horizontal slices gathered from scanning vertically top to bottom were slapped together or stacked in this Fig. 5(c) top-down view. Fig. 5(c) is the result of scanning five full soda cans strapped together to form a chest mounted explosive shown in Fig. 5(a). The object isn't exactly centered to each array since the object is between the transmit and receive arrays, which is why the transmit array is set at 10 degrees. We scan the strapped side-by-side soda cans top to bottom. The slices near the top are weaker than the center. As the scan gets closer to the center, the beam hits the hard top edge of the can and the energy experiences diffraction as in [18]. When we are looking directly at the cans, the energy only hits the curved edges of the cans and the energy disperses elsewhere so the return energy is weaker. Also, some energy goes between the soda cans and gets lost behind the cans. This weakness shows up in the center of the image. The scan continues

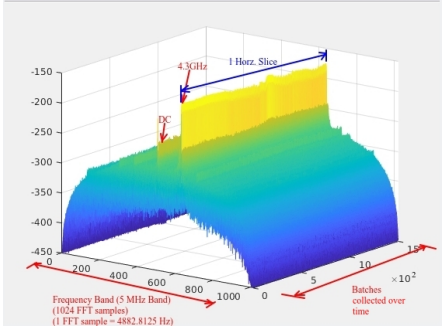
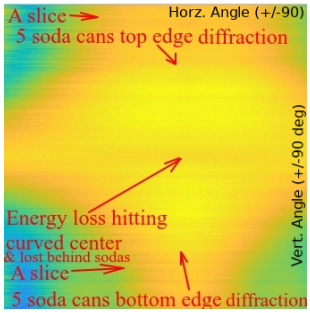


Fig. 4: One horizontal scan slice for imaging

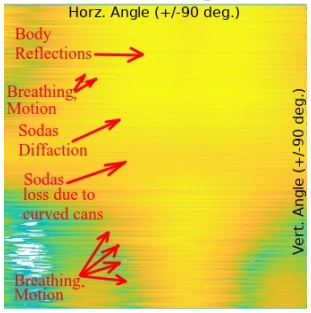
978-1-7281-5092-5/19/$31.00 ©2019 IEEE

(a) 5 sodas

(b) 5 sodas and person



(c) Scan of 5 sodas



(d) Scan of 5 sodas and person

Fig. 5: Wi-Fi scan of 5 strapped sodas

down, and the signal eventually narrows and weakens. All these slight changes in the energy and shape of the image we call "characteristics". Characteristics will help the neural network distinguish one object from another to result in a classification of the object.

The post run process of creating images with MATLAB could be replaced with a real-time image creator written within the C++ of GNU Radio. We know the angle and the level we received so it's only a matter of creating a pixel with an assigned color representing a level to form an image like Fig. 5(c).

The goal of this paper is to prove that a person strapped with dangerous objects can be detected and classified. Soda cans have been used for explosives and allow for easy carrying in public prior to an attack. Before we move on, we add a person with the 5 sodas. This is shown in Fig. 5(b). The scan produces Fig. 5(d), which has many of the same characteristics as Fig. 5(c), but now the person's chest is shown in the image. The loss of energy between and behind the five sodas is still shown in Fig. 5(d). You will notice in Fig. 5(d) the slices are less continuous from slice to slice. This is caused by the person's breathing and motion. We will discuss more results later on.

D. Neural Network Training

After many scans are made to create images, we can train a neural network. We chose the Mask-RCNN algorithm in [1] and the implementation called Detectron in [2]. Why was it chosen? Mask-RCNN has proven to be slightly better over previous RCNN algorithms. Also, Detectron only runs on GPU's. Running on GPU's speeds up the training by days if not weeks. Finally, Mask-RCNN not only puts a bounding box around the classified object, it also puts a mask over the identified object. This mask greatly helps in determining how much of the object was classified. If multiple objects are close together in the image, the masking helps the viewer distinguish the result from each object. Detectron was developed and tested with 8 GPU's. We only have 2 GPU's. If you don't have 8 GPU's and only have 1 GPU, you must multiply the number of iterations to be created by 8 and divide the learning rate by 8. This equates to 90,000 iterations x 8 = 720,000 iterations with a learning rate of 0.02 x 8 = 0.0025. Since we have 2 GPU's we only have to do 360,000 iterations with a learning rate of 0.005. With 2 GPU's our training time was 41 hours while 1 GPU was 94 hours. We ran with two ASUS GTX1060 GPU cards with 6 GB of memory each. Detectron is written in CUDA, but it has a Python interface, which runs on the CPU. We have an Intel i7-4770 CPU manufactured in 2012. So overall our GPU's and CPU are not top of the line, but Detectron runs very well and finishes training in a reasonable amount of time. Detectron for its authors, who had 8 GPU's, finished in 2.5 hours. Ours with only 2 GPU's finished in 1.7 days. These are relative numbers depending on what you may have for hardware but provides you a ballpark figure.

E. Neural Network Inference

Inferencing with Detectron is the final result of our system. It classifies what the object is. We'll get to our results soon, but here we'll state Detectron classifies new images of objects in milliseconds, making it great for real-world, real-time implementations of our system.

We will mention here we tried classifying our images with the popular K-Nearest Neighbor (KNN) algorithm, but it performed awful with a 48% probability of classifying the correct image. 48% is worse than flipping a coin with a 50/50 chance. 48% is the typical reported probability when using KNN to classify any image; thus, it's not recommended. RCNN's perform much better for image classification.

IV. Results

A. Detectron

Before looking at overall results, we'll look at the images our system produces along with the classification by Detectron.

In Fig. 6(b) you see clay material shaped into the form of C4 replica blocks. This clay is harmless but that's what C4 is, clay material that can be formed into many shapes. Even if the clay was shaped to form a child's baby doll, our system could determine something was being carried by the human, which would throw up an alert. In Fig. 6(d) you see the image our system creates. You see the breathing and motion of the human in the image as well as the C4. Detectron placed its bluish mask over the part of the image it detected as human carrying C4. The bluish mask makes it a little difficult to see the finer detail in our images, but zooming in should help. Detectron placed its probability of C4 at 96% at the top left of Fig. 6(d).

Most of the time, humans are carrying harmless objects. One very popular object these days are tablets. We have a tablet being carried on the front of a person in Fig. 6(a). The resultant signal processed image it creates with our system is shown in Fig. 6(c). A tablet is a flat surface so it reflects with

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

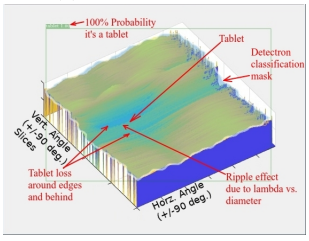
the bluish ridge in the center of Fig. 6(c). Above and below the tablet are curved edges, which causes the loss of energy being returned as shown by the very deep blue in Fig. 6(c). All these little characteristics help the RCNN classify the object. One special characteristic is the ripple effect. This is due to portions of the object being less than λ as transmitted upon by the 1x2 transmit array and viewed by the 2x2 receive array. All these characteristics support Detectron in classification.

Next we consider a human who is carrying nothing placed on the chest, which will occur most of the time. Fig. 7(a) is such a case. You would expect the signal processed image to be more consistent and repeat its self from top to bottom. This is exactly what the image produces. Fig. 7(c) shows the breathing and motion do show up but it's consistent. The top half of the vertical angles are almost a mirror image with the bottom half of the vertical angles. Detectron learned these consistent characteristics and classified the test image correctly as a person with 99% probability shown in the top-left of Fig. 7(c).

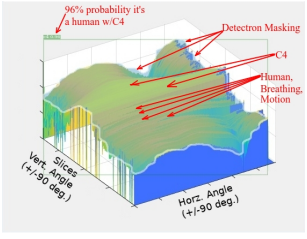
Finally, we look at the pistol/handgun replica being carried by a human in Fig. 7(b). A pistol is a smaller object which is more difficult to detect with our wavelength, λ. The shorter the wavelength, the more signal gets reflected. The longer the wavelength allows for signals to go beyond the object without being disturbed. The pistol is long but very narrow about the barrel and handle. Even with these issues, our system still detected the handgun. Why? Every object creates signal characteristics that the RCNN learns to distinguish from every other object. Here in Fig. 7(d), the handgun creates lots of slices that vary in level from one slice to the next. This is evident in breathing but is amplified here by the handgun making stronger reflections for breathing. Along with this, you will see sloping lines of breathing as indicated in Fig. 7(d), which means the reflected signal is amplified for a longer

(a) Tablet and human

(b) C4 clay replica and human



(c) Scan of tablet and human

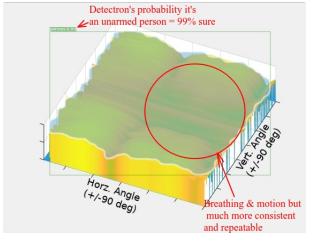


(d) Scan of C4 and human

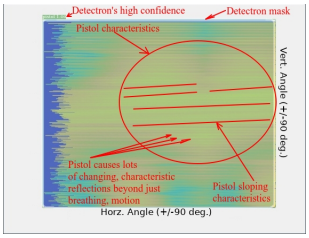
Fig. 6: Wi-Fi scan of human/tablet and human/C4 w/Detectron result

(a) Human

(b) Pistol replica and human



(c) Scan of human



(d) Scan of pistol replica and human

Fig. 7: Wi-Fi scan of human and human/pistol replica w/Detectron result

period from slice to slice than just due to breathing. Detectron placed its bluish mask over the image and classified it as a pistol with 100% probability at top-left of Fig. 7(d).

B. Overall Results

Now we'll present our overall results. In Table I you will see the probability of detecting a person with the objects we have discussed. The probability of detection may appear low since they are in the mid-60's and high-70's, but these are actually pretty good. Remember these are not your typical photograph images we are using with Detectron. We are using the signal processed images we have created from scans as we have shown in Fig. 5(c) and (d) through Fig. 7(c) and (d). We find that to the human eye the images we have created through scans are easily identified and classified; thus, this tells us, more learning is necessary in the training of our neural network. Initially, we had only done 90,000 iterations of Detectron training with 1 GPU on our training set, and we received zero correct classifications with our test data. At 180,000 iterations of learning with 1 GPU we had a few correct classifications. We boosted our number of iterations to 360,000 on 2 GPU's (Equivalent to 720,000 iterations with 1 GPU), which provided the much better results of Table I. We feel if we increase our learning to 720,000 iterations with 2 GPU's, these detection probabilities will only increase. You will notice that the two highest probabilities are the person with a tablet and a person carrying nothing; this will help in reducing false alarms. Also, even though the numbers are low for the dangerous objects, Detectron did classify the person as carrying something, which would raise suspicion and technically raise our reported probabilities. Having probability

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

TABLE I: Probability of Detection

C4 on Person	Person Alone	Pistol on Person	Sodas on Person	Tablet on Person
70.0%	73%	64%	68%	78%

of correct detection in the mid-60's to high-70's is only 2.2 to 3.6 failures per 10 people scanned, which isn't that bad. An installation of this system in a location used for public transportation would increase the sampling of humans at a rate of thousands per day. Also, training could occur on an offline system to provide weekly trained model updates. We feel our numbers in Table I will only increase.

Our system requires the environment doesn't change much. Once it's installed at a subway turnstile or at the entrance of a public bus, the environment stays the same. Typical RCNN's must be trained with thousands of pictures since the background is always changing. For example, trees, cars, etc. are always appearing in pictures besides just the object of interest. Our system is viewing one unchanging environment, which means we can train our system with hundreds of fewer images. In addition, the accuracy of the proposed system could be improved once more training data is available.

V. Conclusion

We've shown that dangerous objects placed on the chest of terrorists can be detected with our Wi-Fi imaging and classifier. These systems could be placed at subway turnstiles or at the entrance of public buses. If these systems were installed, we could increase the number of Wi-Fi images used for training since thousands of people use public transportation per day. The system could be trained weekly increasing the probability of detection far beyond our results. We concentrated on the person's chest, but this could be expanded to scanning backpacks and other parts of the body.

References

[1] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. "Mask R-CNN". *CoRR*, abs/1703.06870, 2017.

[2] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. "Detectron". https://github.com/facebookresearch/detectron, 2018.

[3] F. Adib and D. Katabi. "See Through Walls with WiFi!". In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 75–86, New York, NY, USA, 2013. ACM.

[4] D. Huang, R. Nandakumar, and S. Gollakota. "Feasibility and Limits of Wi-Fi Imaging". In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, SenSys '14, pages 266–279, New York, NY, USA, 2014. ACM.

[5] Chen Wang, Jian Liu, Yingying Chen, Hongbo Liu, and Yan Wang. "Towards In-baggage Suspicious Object Detection Using Commodity WiFi". *2018 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9, 2018.

[6] Wireless Gigabit Alliance. https://en.wikipedia.org/wiki/Wireless_Gigabit_Alliance.

[7] T. Wei and X. Zhang. "mTrack: High-Precision Passive Tracking Using Millimeter Wave Radios". In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, pages 117–129, New York, NY, USA, 2015. ACM.

[8] N. Obeid, M. Heddebaut, F. Boukour, C. Loyez, and N. Rolland. "Millimeter Wave Ultra Wide Band Short Range Radar Localization Accuracy". In *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, pages 1–5, April 2009.

[9] H. El-Sayed, G. Athanasiou, and C. Fischione. "Evaluation of localization methods in millimeter-wave wireless systems". In *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pages 345–349, Dec 2014.

[10] M. Shinotsuka, Y. Wang, X. Ma, and G. T. Zhou. "Designing radio interferometric positioning systems for indoor localizations in millimeter wave bands". In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pages 1184–1188, Nov 2014.

[11] F. Lemic, J. Martin, C. Yarp, D. Chan, V. Handziski, R. Brodersen, G. Fettweis, A. Wolisz, and J. Wawrzynek. "Localization as a feature of mmWave communication". In *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1033–1038, Sept 2016.

[12] "Price for TSA's failed body scanners: $160 million". https://www.politico.com/story/2015/08/airport-security-price-for-tsa-failed-body-scanners-160-million-121385.

[13] Richard Wallace and Steve Dunbar. "2.4 GHz YAGI PCB Antenna". http://www.ti.com/lit/an/swra350/swra350.pdf, 2010.

[14] Ajarn Changpuak. "Yagi Uda Antenna Calculator". https://www.changpuak.ch/electronics/yagi_uda_antenna_DL6WU.php, 2014.

[15] K. Rothammel and A. Krischke. *"Antennenbuch"*. DARC-Verlag, 2002.

[16] Constantine A Balanis. *"Antenna Theory: Analysis and Design"*. Wiley-Interscience, 2005.

[17] P.V. O'Neil. *"Advanced Engineering Mathematics"*. Wadsworth Publishing Company, 1987.

[18] J. Schiller. *"Mobile Communications"*. Addison-Wesley, 2nd edition, 2004.

978-1-7281-5092-5/19/$31.00 ©2019 IEEE