ISSN 2347 - 3983
Volume 7, No. 9 September 2019
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter05792019.pdf
https://doi.org/10.30534/ijeter/2019/05792019

Defensive Turret Defensive Turret with Fully Automated Motion Detection Using Infrared Technology

Hazel Mae A. Adriano[1], Dustin A. Reyes[2], Kristelee Jem B. Tan[3], Daniel D. Zagada[4], Reggie C. Gustilo[5]

[1]ECE Department, De La Salle University, Manila, Philippines, hazel_adriano@dlsu.edu.ph
[2]ECE Department, De La Salle University, Manila, Philippines, dustin_reyes@dlsu.edu.ph
[3]ECE Department, De La Salle University, Manila, Philippines, kristelee_tan@dlsu.edu.ph
[4]ECE Department, De La Salle University, Manila, Philippines, daniel_zagada@dlsu.edu.ph
[5]ECE Department, De La Salle University, Manila, Philippines, reggie.gustilo@dlsu.edu.ph

ABSTRACT

This research presents an automated defensive turret that has human detection capabilities even in low-light or night-time conditions through the use of infrared technology. Motion detection through Python and Open CV is used to detect humans in the Area of Interest (AOI) shown covered by digital cameras than can become possible targets. The target detection can reach as far as 100 feet from the system. The turret can be operated on both automatic and manual control modes. In automatic control mode, the turret is triggered upon detection of the first human and then automatically fires continuously. On manual mode, the user has full control of the movements of the turret like the pan and tilt using the computer's pointing device. The turret is also capable of continuously shooting the target it is tracking. During testing of the programs and hardware prototype, the targets were placed randomly in different distances from the turret. These tests were done in both standard and infrared modes of the camera. Results show that the turret can effectively operate in both automatic mode and manual control mode. This project aims to assist the Armed Forces during the clearing and defensive operations against national or international aggressors.

Key words: automated turret, infrared, motion detection, target tracking

1. INTRODUCTION

The information era brings us interesting research topics in the fields of robotics [1], [2], [3], automation and artificial intelligent mechanism that help us simplify and daily lives and help us in some dangerous tasks. Some scientists use machine vision and intelligent control systems in the fields of aquaculture [4],[5], medical technology [6], [7] or simply purely robotics systems.

With machine vision, a number of intelligent algorithms approaches are developed for human, object or animal recognition and tracking [8], [9], [10], [11], [12], [13].[14]. Lastly, these research trends in computer technology and artificial intelligence came along with the improvement of communications system with IoT. Continuously, the design, simulations, test and analysis on how to improve communications links and performance are still very important topic for the research community [15], [16], [17], [18]. These advancements in technology led us to develop and improve a robotic system that can be used to support military operations and minimize military casualties against dangerous aggressors.

Existing defensive turret technologies need to be enhanced to cater to the military and security needs and problems that the local community is facing. Not only will the technology be developed by the group to help the research and development of military organizations or authorities for the purpose of public security, but it will also have the capability to perform mechanical functions that could help assist subduing or disabling the security threats in government institutions.

This study aims to alleviate problems about security threats that can cause major casualties, loss of property and facility breaches. Military applications can be considered as one of the most significant uses of this study. Since the military engage in warfare of different settings, the infrared tracking system may be of use to track enemies during the night-time and the standard tracking system may be used during daytime.

Equipped with a camera and infrared capabilities, the ability to automatically detect, track and continuously fire at targets will enable users to operate and monitor potentially deadly zones with confidence with minimal labor. With the optical camera, the prototype can provide real-time video feed which is then streamed to a web GUI via internet connection. This feature can then provide data to the users for military planning on the environment.

2. DESCRIPTION AND METHODOLOGY

The prototype assembly diagram in Figure 1 shows the connections of each component of the prototype. The turret can only be controlled using direct access of the computer's program or through remote access via a wireless communications link. A personal computer serves as the main hub of all the components used in the prototype. The system also uses machine vision and human recognition algorithm to detect enemy presence in the area. A graphical user interface was designed to show video streamed from the camera and the human detection made by the program. Several parameters are calibrated in the system namely the target coordinates, the tracking and locking in of targets, the trigger button mechanism, and the choice between using the standard camera or the infrared camera. The image processing code sends the position coordinates of the target to the motors holding the firearm for proper aiming and firing. A pointing device is used to navigate and control the movement of the turret while in manual mode. The turret also has a continuous triggering capability wherein it will continuously push the trigger while the target is locked and being tracked. For safety purposes, the firearm used in this research project is a M4 Airsoft.

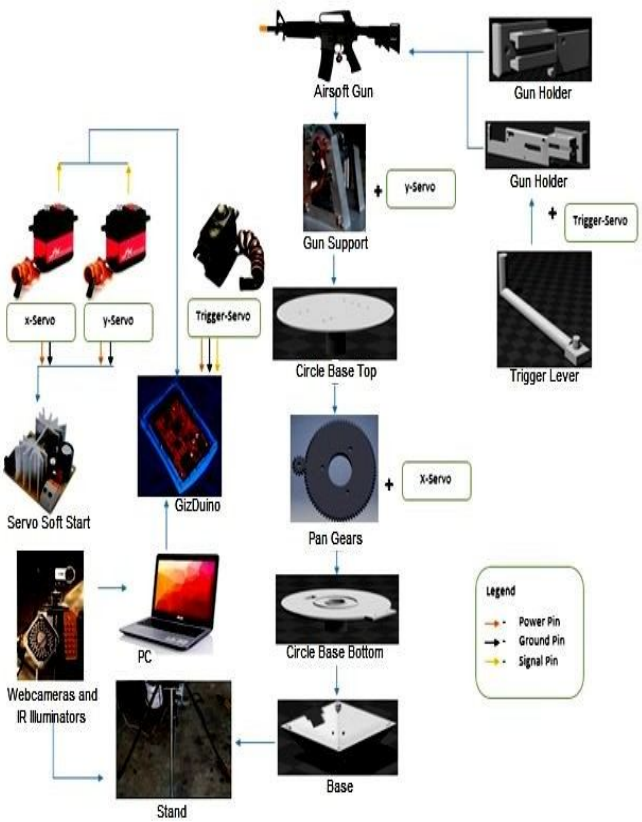

Figure 1: Sentry Gun Assembly Diagram

Figure 2: Sentry Gun Prototype

Figure 2 above shows the prototype of the system. The design for the physical layout of the system was done using 3D CAD software. The design of the platform of the system included important parameter considerations namely the structural integrity of the base and the weight of the turret.

Figure 3: Airsoft Gun Web camera showing Streamed Video with Crosshair

Figure 3 shows the webcam attached above the airsoft gun to show the line of vision of the turret. A crosshair is shown at the center of the video feed to determine whether the target is properly tracked or not. The webcam was calibrated by using the laser pointer as a reference to the center of the webcam's video feed. A target stood 10 feet away from the turret and the gun was manually aimed to the target. The webcam was adjusted until the crosshair was aligned with the laser pointed at the target's body.

At the start of the operation, the turret's default position is initialized. The user must first set whether the turret will be used in manual mode or in fully automatic mode. When in a fully automatic mode, the turret will autonomously scan and lock the target detected. It will remain idle or stationary until the algorithm detects motion within the camera's field of vision. If multiple targets are detected, the user will be given an option to aim or lock in to a preferred target meaning only one target can be locked at an instance.

Once a specific locked-in target is chosen, the turret will become fully automatic and continuously track the target automatically. If multiple targets are detected and the user chooses not to switch to another preferred target, the turret will simply follow the first target it detects as a default and then operate normally either in fully automatic and manual mode. The triggering is done continuously once a target is tracked.

The general algorithm used for motion detection is background subtraction method. The background subtraction algorithm was used to detect the motion of the person once it enters the camera's field of vision. Its main technique is to calculate the difference between the current image and a reference background image. A stored background image of the surroundings captured by the camera is used as the reference image. Incoming video frames are subtracted from the reference image. A certain threshold value for the difference shall be set so that negligible environmental movements will not be considered as movement that will trigger the firearm. A remote desktop is used to access the program of the turret over the internet. The user may control the turret in automatic or manual mode as well as view the video, change the offset settings, camera settings, and trigger the turret. To test the accuracy of the prototype created, the motion detection and target shooting were tested by simulating a target walk every 10 feet until a 100 feet is reached. This was done in daytime and night-time conditions using both standard camera and an infrared camera. The limitations of the turret was tested by making a target jog at varying speeds until the turret cannot successfully follow and shoot the target anymore. These trials were documented and the maximum speed of the target that the turret can track was recorded. The motion detection was also tested by testing whether or not the algorithm had a false positive, false negative, true positive, or true negative output.

2.1 Hardware Considerations

During tests and simulations of the defensive turret, several components were needed to be fabricated and assembled. The hardware of the turret considered important parameters for stability of the system such as the design of the supporting mechanism of the airsoft gun, the range of motion, and the connection of the data center (PC) to the system. The hardware base consists of the gun supports attached to the rotating base has the gear attached on its protruding component. These are then attached to the supporting base structure to hold the weight of the gun. A four legged stand was used to provide stability and serve as a foundation for the system. The camera stand was connected to the four legged stand by a distance of 22 inches and it has a vertical height of 56 inches. The visual feedback given by the cameras from the camera stand was then inputted to a personal computer. A GUI was created to view the visual output taken by the camera.

A GizDuino microcontroller was used to provide control to the servo motors which is directly interfaced to the computer. Three servo motors were used for this setup: pan, tilt, and trigger servos. The trigger servos had all three of its pins connected to the GizDuino while the pan and tilt servos only had the signal pins connected to the microcontroller. The reason for this setup is that the power and the ground pins of the pan and tilt servos were connected to the servo soft start circuit. Each of these servo motors had their own slots on the fabricated bases. The x-servo motor is placed near the gear of the stationary base so that it can control the x-axis movement of the turret while the y- servo motor is placed near the gear of the gun support to provide vertical motion. The trigger servo is connected to a lever to provide firing action and is placed within the gun holders. The gun holders are used to hold and support the gun during its motion of operation.

The model of the gun used is a Golden Eagle M4A1 assault airsoft rifle AEG having a weight of 4.88lbs which is somewhat close to a 6.88 pound real M4A1 assault rifle. Having the same physical appearance to a real M4A1, this airsoft gun would serve as the ideal substitute for the purpose of simulating the function of a real weapon. The Golden Eagle M4A1 assault rifle is also accurate within the maximum range of 100 feet with respect to the objective target.

2.2 Software Considerations

One of the objectives of the study is to produce latency not greater than 300 milliseconds. With this, background subtraction algorithm was chosen due to its computational efficiency. However, it must be indicated that the decision for this algorithm would entail low performance in others areas to be considered in motion detection such as background lightning changes, occlusion and camouflage of detected moving objects. Background subtraction has a low processing speed requirement but has poor performance in the said circumstances. The purpose of the sentry gun is to be operating in a real-time environment and must continuously track detected targets; hence processing speed was prioritized as the main basis for choosing background subtraction as the turret' motion detection algorithm.

Figure 4: Background Image Reference

An example background is shown in Figure 4. This is the reference background where all future foreground objects or targets will be based. Whenever a target walks in front of a camera, that current frame will be compared with the reference background. An example of a moving target in front of a camera is shown in Figure 5.

Figure 5: Example Input from Camera with Target

Using OpenCV‟s "BackgroundSubtractorMOG2" function, continuous foreground frames fed from the camera are applied with the said function. The function's output is a thresholded black and white image where white areas are the detected moving or foreground objects. BackgroundSubtractorMOG2 is a Gaussian Mixture-based Background/Foreground Segmentation algorithm which has the capability to select the appropriate number of Gaussian distribution for each pixel. Due to this, there is better adaptability to illumination and environment changes. The output image from BackgroundSubtractorMOG2 also contains gray areas which represent shadows. An example of an output from the BackgroundSubtractorMOG2 function is shown in Figure 6.

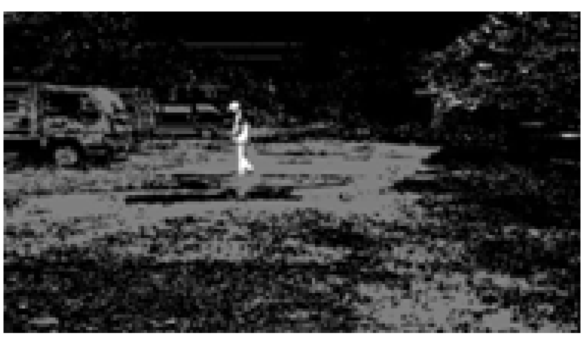

Figure 6: Output Image from the BackgroundSubtractorMOG2 function

The shadows appeared after the background subtraction are removed by thresholding the image and remove the gray color. The input of this function is the output from the BackgroundSubtractorMOG2 function.

THRESH_BINARY is used so that if the corresponding pixel is greater than threshValue2, the pixel would be set to white or 255. The value for the threshold "threshValue2" is calibrated in the GUI. The output of this function still contains noises from movements of leaves as shown in Figure 7.

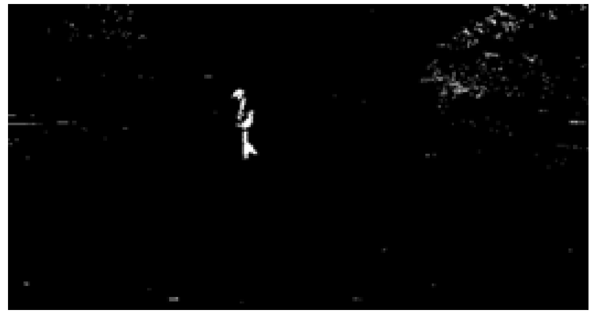

Figure 7: Output Image from the Threshold function

Morphological processing is then performed to the image to improve the features of the blobs. A blob is a term used to refer to the white area of the output image which represents the foreground or moving object. Morphological opening is done to remove small noises detected in the environment. The input to this function is the output from the Threshold function. The structuring element or kernel determines the nature of operation of the morphological opening. The kernelSize is fully calibrated in the GUI and determines how many blocks of pixels are affected the morphology process through the process of convolution. The result of morphological processing is shown in figure 8.

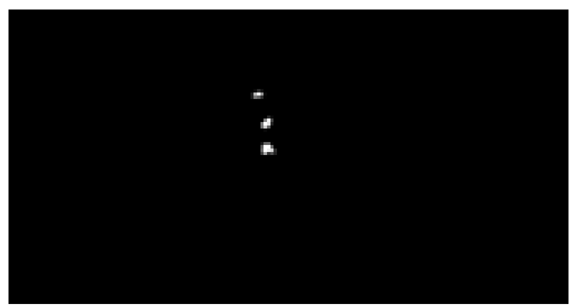

Figure 8: Output Image from the Morphological Opening

Dilation is done to increase the area of the blob. The different parameters in the function determine the nature of the dilation. Figure 9 shows the output of dilation.

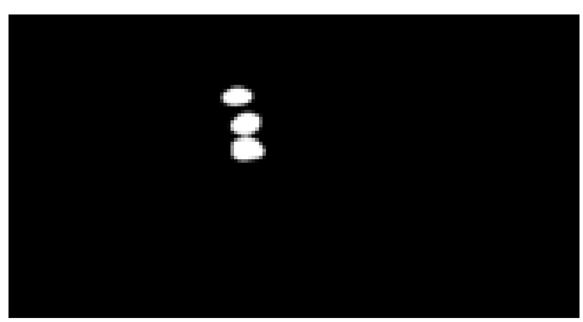

Figure 9: Output Image from Dilation

Morphological closing is then performed to fill the holes inside each blob so that it will be a concrete white area. It is the opposite of the Morphological Opening function. Instead of removing white areas outside the blob, it fills up holes inside the blob itself so that when the blob of a single person is separated into two, it will join and become only one blob. The output of morphological closing is shown in figure 10.

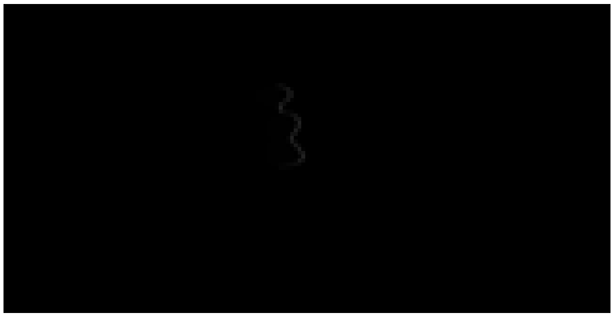

Figure 10: Output Image from Morphological Closing

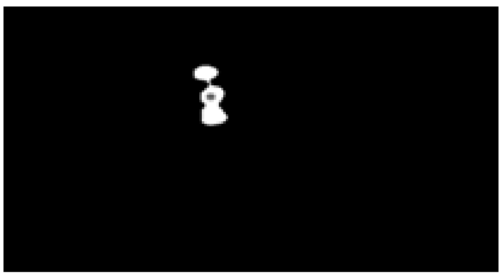

Figure 11: Output Image of Contours

Figure 11 shows the output after using the countours function. The contours function is able to detect individual blobs by making use of a curve to join all continuous points together. The input to the cv2.findContours function is the output from the Morphological Closing. To find the center coordinate of each blob, an automatic function is already available through the OpenCV library using the function "cv2.moments". The output of this function stores all necessary data such as data needed to calculate the center of mass of the blob.

Figure 12: Final Output Image from Motion Detection Algorithm using OpenCV functions

All centroid coordinates of detected blobs will be stored in an array and drawn with a circle on the original image coming from the camera and the blobs‟ corresponding IDs. The final output image is shown in Figure 12.

One of the objectives of this study is to be able to detect multiple targets. The user of the turret must be able to choose the correct target.

A specific ID must be associated with each target once it is detected within the camera's field of view. As long as the target is within the camera's field of view, the same ID must be assigned to each specific target. This mechanism is necessary because after OpenCV is able to detect the blobs of the moving objects through the motion detection algorithm, it will not able to associate a blob as the same blob detected from the previous frames. For example, two persons were detected by the algorithm. The first person was assigned as Target 1 while the other one was assigned as Target 2. For the next camera frame, if both persons are still in the camera's field of view, the first person must still be assigned as Target 1 and the second person is still Target 2. This algorithm was implemented by simply using logical statements through Python.

The main purpose of the Kalman Filter in the turret's processing code is to predict where the target will be in case that target suddenly gets undetected because of the camouflage problem. This prediction will only occur for a specific amount of time and for any time greater than the specified period, it will be assumed that the target has hidden behind an obstruction or moved out of the camera's field of view which are situations not covered by the camouflage problem.

Kalman Filter will only be utilized on the specific target that the user has picked for the turret to lock onto. This is because processing speed might degrade if all targets detected within the camera's field of view are being processed with Kalman Filter. The target that the turret is currently locked onto will be the only target processed with Kalman Filter. There is already an existing available code for use which implemented the Kalman Filter for mouse tracking using C++. Mouse tracking is similar with the purpose of this algorithm where there is one single coordinate to be followed by the Kalman Filter. In the case of the mouse tracking code, the coordinates were the mouse coordinates. The existing code was then converted into its Python equivalent and some changes were made for the code to be able to accept coordinates from targets and not from a mouse.

2.3 Calibration from GUI Coordinates to Real World Implementation

To start the calibration, a target should be shown on the bottom left side of the GUI video feed. Through manual control, the user will adjust the turret until the laser is pointed at the area shown on the bottom left coordinate. The coordinates of the servo motor at this point will be recorded and set as an input for both the X Min degrees and Y Min degrees section on the GUI. This process is repeated to find the maximum X degree and maximum Y degree which will then be set as inputs to the X Max degrees and Y Max degrees section on the GUI respectively.

3. RESULTS AND DISCUSSION

To test the accuracy of the prototype created, the motion detection and target shooting were tested by making a target walk and jog every 10 feet until a 100 feet is reached. The results presented below will show several video frames showing the operation of the actual turret, morphological operations done on the video feed during testing, several video frames showing the tests done on each distance when the target was walking during daytime and night-time, several video frames showing the tests done when the target was jogging, and several video frames showing the tests done for multi-target tracking. A summary of the turret's specifications including the average maximum speed of the target that the turret can successfully track, angle of coverage of the turret, target engage accuracy, target tracking precision, motion detection software accuracy, and motion detection software latency will be shown on a table below.

A webcam attached on the airsoft gun shows field view of the system. This was used to see whether or not the turret was able to track and shoot the target. This webcam shows a video feed with a crosshair to allow users to see if the target can be tracked by the turret or not. The results are presented by showing several frames taken from the videos recorded by the webcam during testing. These will also be presented for every distance tested in both daytime and night-time tests

Figure 13: Multi-target Test

Figure 13 shows several frames from the streamed video with 2 targets. The system can successfully identify the two targets one at a time.

Figure 14: Daytime Testing Sample

Figure 14 shows multiple frames from day time video streaming. The morphological process on each frame is shown in figure 15.

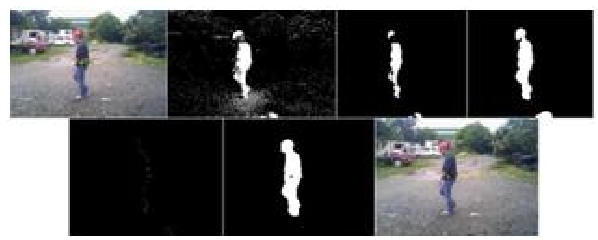

Figure 15: Morphological Operations Applied to the Video Feed

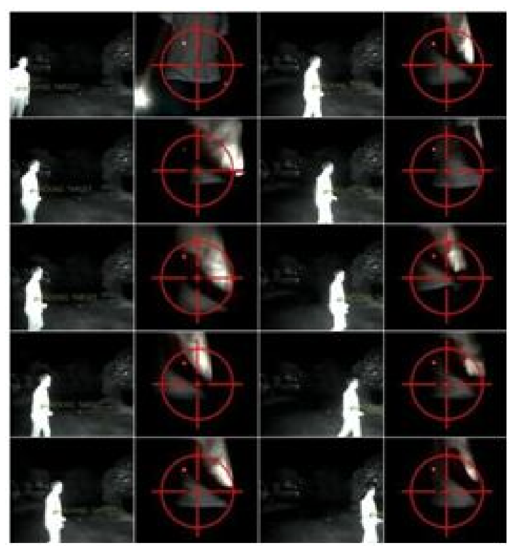

Figure 16: Night-time Testing Sample

Figure 16 shows how effective the program is in detecting and tracking a target in night-time. The overall performance of the turret system is shown in table 1.

Table 1: Turret Performance Specifications Summary

Turret Performance Specifications				
Minimum Distance				10 feet
Maximum Distance				100 feet
Maximum Speed	Lateral Movement	10 ft	3.048 m	1.289 m/s
		20 ft	6.096 m	2.392 m/s
		30 ft	9.144 m	2.909 m/s
		40 ft	12.192m	3.217 m/s
		50 ft	15.24 m	2.516 m/s
		60 ft	18.288m	2.408 m/s
		70 ft	21.366m	2.356 m/s
		80 ft	24.384m	2.032 m/s
		90 ft	27.432m	2.04 m/s
		100 ft	30.48 m	2.035 m/s
Average Maximum Target Speed				2.319 m/s
Turret Angle Coverage	Pan	94 degrees		
	Tilt	46 degrees		
Target Engage Time				2.01 sec
Target Engage Accuracy				100%
Target Tracking Precision				8.85 cm
Probability of Shooting miss				17.26%
Motion detection software latency				210.46ms

4. CONCLUSION

In this research a program design and sentry gun prototype are presented that is able to detect motion and track humans during daytime and night-time conditions. Normal and infrared vision cameras were used respectively to enable such capabilities in those conditions. Manual control was also successfully implemented with the use of the computer's pointing device. During testing, the automated turret's motion detection and tracking, target shooting, multi- target tracking, and remote PC control capabilities show the effectiveness and accuracy of the system. The performance limitations of the turret were also tested and it is found that the maximum speed of the target that the turret can successfully track and shoot is at 2.319m/s. This research is intended to assists military operations during their dangerous encounters with terrorists.

REFERENCES

1. Celebre, A.M.D., Dubouzet, A.Z.D., Medina, I.B.A., Surposa, A.N.M., Gustilo, R.C., **Home automation using raspberry Pi through Siri enabled mobile devices**, *8th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2015*, (2016) https://doi.org/10.1109/HNICEM.2015.7393270
2. Dulay, A., Sze, R., Tan, A., Yap, R., Materum, L., **Development of a wideband PLC channel emulator with random noise scenarios**, *Journal of Telecommunication, Electronic and Computer Engineering*, . 2018
3. Dulay, A.E., Sze, R., Tan, A., Huang, Y.-H., Yap, R., Materum, L., **FPGA implementation of an indoor broadband power line channel emulator**, *Proceedings of KICS-IEEE International Conference on Information and Communications with Samsung LTE and 5G* Special Workshop, ICIC 2017. https://doi.org/10.1109/INFOC.2017.8001678
4. Gustilo, R.C., Dadios, E.P., **Behavioural response analysis using vision engineering (BRAVENet)**, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 21(2), pp. 211-220, 2017 https://doi.org/10.20965/jaciii.2017.p0211
5. A. Africa, J. Aguilar, C. Lim Jr, P. Pacheco, and S. Rodrin, **Automated Aquaculture System that Regulates Ph, Temperature and Ammonia.** *9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM).* 2017. https://doi.org/10.1109/HNICEM.2017.8269494
6. E. Mohammadi, E. P. Dadios, L. A. G. Lim, M. K. Cabatuan, R. N. G. Naguib, J. M. C. Avila, and A. Oikonomou, **Real-Time Evaluation of Breast Self-Examination Using Computer Vision,** *International Journal of Biomedical Imaging*, vol. 2014, pp. 1–12, 2014. https://doi.org/10.1155/2014/924759
7. R. A. A. Masilang, M. K. Cabatuan, and E. P. Dadios, **Hand initialization and tracking using a modified KLT tracker for a computer vision-based breast self-examination system,** *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 2014. https://doi.org/10.1109/HNICEM.2014.7016244
8. Hilado, S.D.F., Dadios, E.P., Gustilo, R.C., **Face detection using neural networks with skin segmentation**, *Proceedings of the 2011 IEEE 5th International Conference on Cybernetics and Intelligent Systems*, CIS 2011 pp. 261-265, (2011) https://doi.org/10.1109/ICCIS.2011.6070338
9. J. A. C. Jose, M. K. Cabatuan, E. P. Dadios, and L. A. G. Lim, **Depth estimation in monocular Breast Self-Examination image sequence using optical flow,** *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, 2014.
10. M. N. Eman, M. K. Cabatuan, E. P. Dadios, and L. A. G. Lim, **Detecting and tracking female breasts using neural network in real-time,** *2013 IEEE International Conference of IEEE Region 10* (TENCON 2013), 2013.
11. Hanpinitsak, P., Saito, K., Takada, J.-I., Kim, M., Materum, L., **Multipath clustering and cluster tracking for geometry-based stochastic channel modeling,** *IEEE Transactions on Antennas and Propagation*, . 2017 https://doi.org/10.1109/TAP.2017.2754417
12. P. Loresco and A. Africa, **ECG Print-out Features Extraction Using Spatial-Oriented Image Processing Techniques.** *Journal of Telecommunication, Electronic and Computer Engineering.* Vol. 10, Nos. 1-5, pp. 15-

20, 2018.

13. Praveen Kumar Kollu and R. Satya Prasad, **Intrusion Detection System Using Recurrent Neural Networks and Attention Mechanism**, *International Journal of Emerging Trends in Engineering Research*, Volume 7, No. 8, 2019 https://doi.org/10.30534/ijeter/2019/12782019
14. Kuntal Mondal and Dr Padmavathi S, **Wild animal Detection and Recognition from Aerial Videos using Computer Vision Technique**, *International Journal of Emerging Trends in Engineering Research*, Volume 7, No. 5, 2019
15. Dy, H.C., and Gustilo, R.C., **Characterization of signal response for surface water movements in underwater optical wireless communications**, *IEEE Region 10 Annual International Conference, Proceedings/ TENCON*, (2012) https://doi.org/10.1109/TENCON.2012.6412238
16. Baykas, T., Materum, L., Kato, S., **Performance evaluation of mmWave single carrier systems with a novel NLOS channel model**, *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, . 2013 https://doi.org/10.1109/PIMRC.2013.6666157
17. Dulay, A.E., Yap, R., Materum, L., **Hardware Modelling of a PLC Multipath Channel Transfer Function**, *Journal of Telecommunication, Electronic and Computer Engineering*, 2017
18. A. Africa, S. Bautista, F. Lardizabal, J. Patron, and A. Santos, **Minimizing Passenger Congestion in Train Stations through Radio Frequency Identification (RFID) coupled with Database Monitoring System**. *ARPN Journal of Engineering and Applied Sciences*. Vol. 12, No. 9, pp. 2863-2869, 2017.