Prepared by:MADIHA KANWAL

mmWave Radar & Machine Vision Fusion for Pedestrian Collision Warning

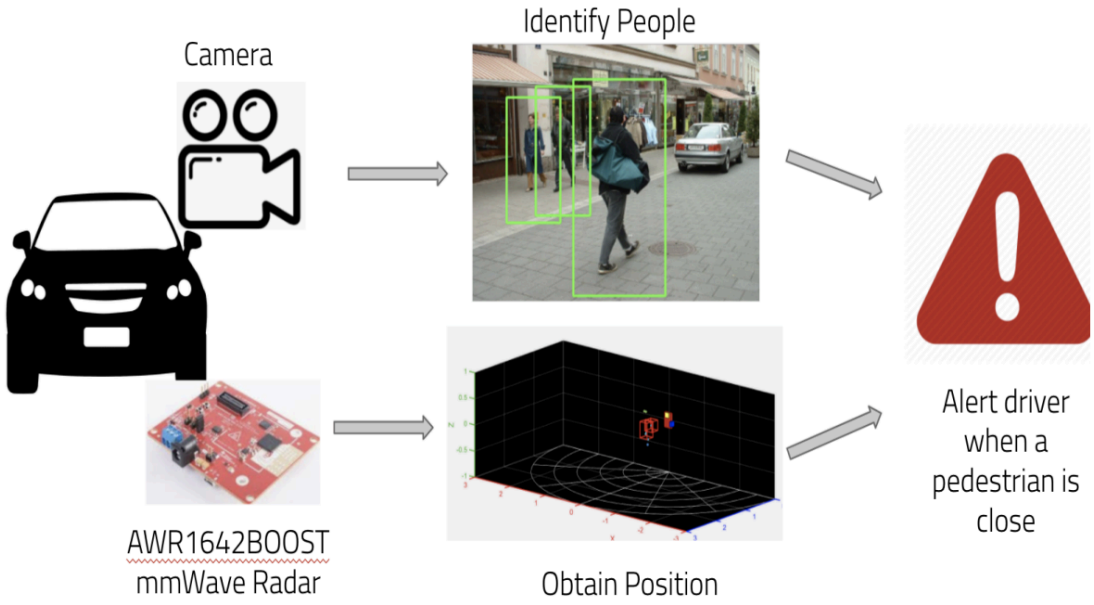
ABSTRACT:

The autonomous-automotive or Electric Vehicle (EV) industry is and will continue to be one of the most regulated industries especially as the concern for human others safety behind modern stateful computers that manipulate nearly every driver's move with the intention that it may potentially be even more safer than a human driver.

Technology advancements made by the hardware creator, Texas Instruments (TI) had designed and developed a product helping and incorporating vehicles and other safety versed industries for example; oil & gas for object avoidance, construction and manufacturing for aid in object tracking and verification. For this report a proposed system is designed to take advantage of the Radar AWR1642BOOST by TI, the mmWave sensor, and harness the parallel-computing power of Jetson Nano with the robustness of Machine Vision, a subset of Machine Learning (ML) in Computer Vision. The fusion system would warn the driver of potential obstacles, pedestrians, or even pot-holes because of the hardware's capability to "feel" and detect depth by utilizing pedestrian data points provided by the ML application, angle deviation, and velocity.

INTRODUCTION:

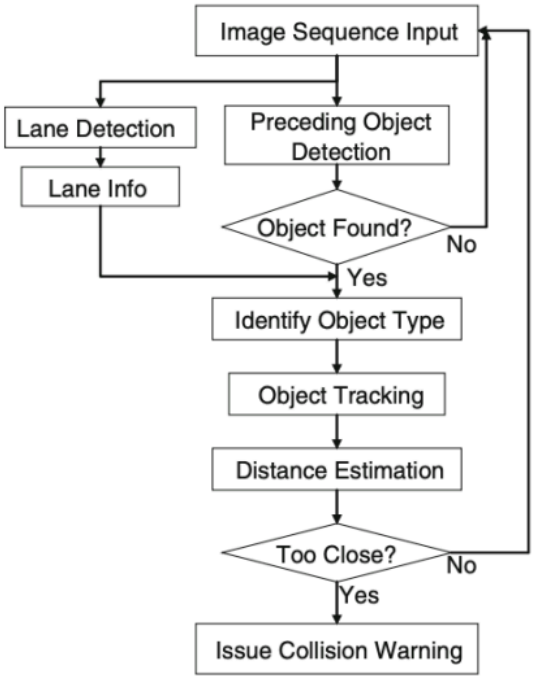
General structure of our project is the shown in the next figure below:



It is imperative the safety and the protection of the pedestrian must be adhered to at all times. As the EV industry advances and the use of supplemental preemptive safety systems becomes critical, millions of data points are consumed by any given EV autonomous system. The use in Machine Learning systems which tap into data points within a shared Convolutional Neural Networks or CNN, powered by a shared Open Source Software(s) produced by companies like Google and Amazon. The Jetson Nano's ability to process and

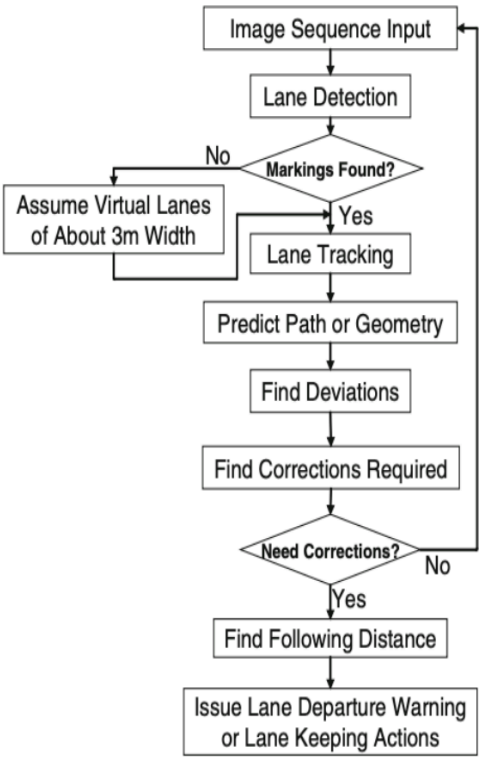
compute multiple object-oriented applications (threads) gives the system an advantage as it is a data and process flow intensive controller running a base version of Ubuntu as the operating system. The computing Hardware coupled by the TI AWR1642boost provides us another subset of data, from velocity, to radar-vision data space alignments. The mentioned data points along with fusing information received by the common USB Camera, we can connect multiple safety related points together. This information enables the system to realize pedestrian, or objection detection and feed in that stream of information through a series or “pipelines" of algorithms. We then process the information received by the TI radar hardware through the intelligence system (ML) and make useful predictive behaviors provided to us by the application in turn are funneled by
H.O.G (histogram of gradient) for radar-vision information. The system is effective because it takes two lightweight module components like the Jetson Nano, Machine Learning, and a robust Radar sensor like the AWR1642boost in our system. The combined system has the ability to flatten and heterogenous data points crucial for machine learning models. As more pedestrian safety information is consumed by the fusion system your model would improve overtime just by the innate nature of an intelligent system tapping into an inter-frame tracking algorithm as part of the pipeline to extract and validate target signal from radar data with noise. This computer vision algorithm uses the information from TI’s radar hardware and ML fusion system to sense and detect pedestrians.

RELATED WORKS:
Forward collision warning (FCW) or collision avoidance
The FCW system detects any objects in the same lane, calculates distance between the object and the vehicle, and issues a collision warning in order to avoid accidents if the distance is quickly becoming shorter than a threshold value. In this way, FCW will issue collision warning only when it finds that the vehicle will collide with another vehicle or object if it continues to move with the current speed.



Lane departure warning (LDW):

The LDW system detects and tracks lane markings, predicts lane geometry, finds any deviation from path, and issues lane departure warning or lane keeping action while keeping an eye on the vehicles behind to avoid rear-collision.



Adaptive cruise control (ACC):
Adaptive cruise control system automatically slows down the vehicle when it approaches another vehicle in front and accelerates again to achieve the preset speed when traffic allows. Traditional ACC systems use laser or radar technologies to measure the distance and speed of the vehicle in front. However, they have proposed a camera-based implementation of ACC

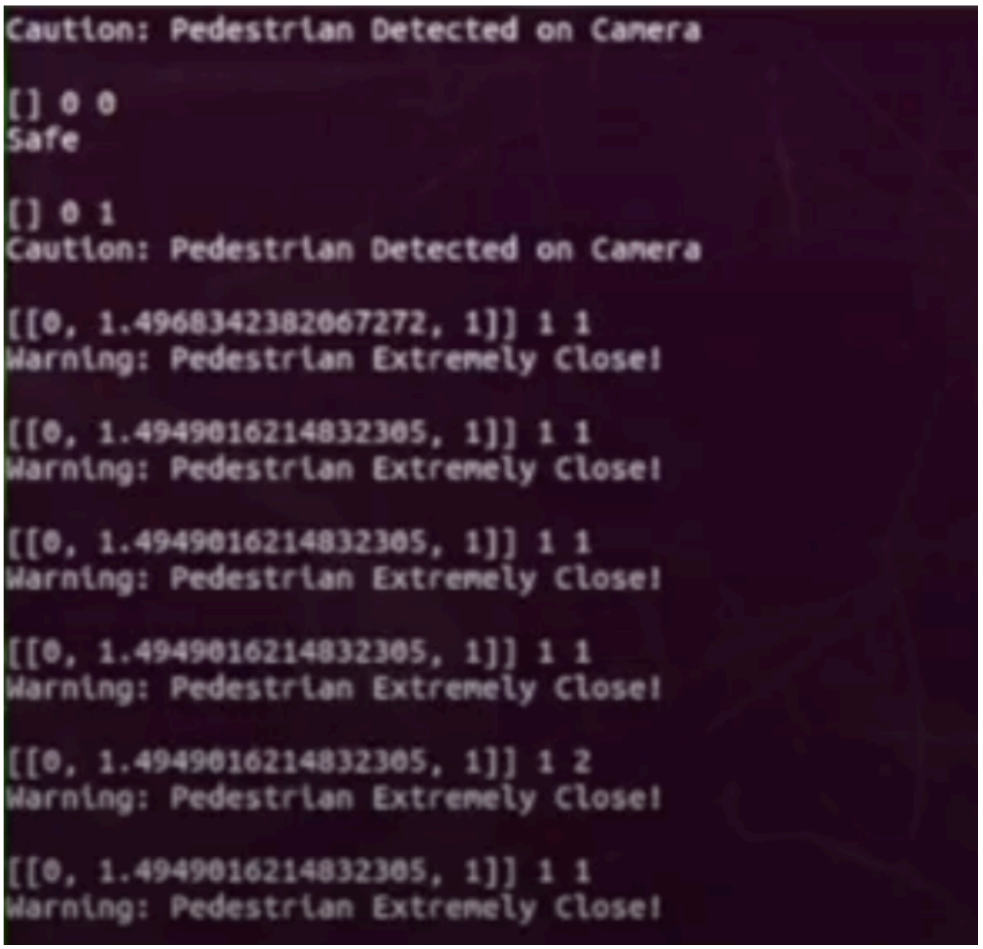
Intelligent speed adaptation (ISA):
The ISA system looks for any speed-limit sign on the road, and compares the speed limit with the speed of the vehicle. If the vehicle is too fast, it issues a warning and reduces the vehicle speed while keeping an eye on the vehicles behind to avoid rear-collision.

Parking assistance (PA):
The PA system identifies any objects in the very close proximity of the vehicle, tracks them to find their distance, and issues a collision warning if the vehicle comes very close to the objects.

Traffic sign recognition (TSR):
The TSR system selects a region of interest (RoI), finds and tracks any candidates, extracts features, and classifies the sign. It then shows this sign on the display, if valid.

SYSTEM DESCRIPTION:
The camera is first postionsioned and mounted to the vehicle which will initiate the live video stream, overlaid by both the ML and Radar Sensor systems. As the pipeline of systems

processes the flow of live information it can begin to detect in a fusion of data points potential obstacles like pot-holes, and more importantly, pedestrians.
The information use the infThe systems has the ability to preemptively warn the driver of machine learned detections, for the lab presented an output is displayed via the Jetson Nano's Terminal window output:



Hardware Description:

Hardware Components:

- TI AWR1642BOOST-ODS mmWave radar sensor
- NVIDIA Jetson Nano single-board
- Camera
- Monitor, cables, micro SD card, and other peripherals.

Examples of mmWave and Machine Vision Hybrid Solutions:

- Child presence detection with mmWave Radar and camera. Camera-based driver monitoring system can be used to detect the child's presence; it might not be able to

function properly if it’s not under the camera’s view. sensors along with a camera to get the right result

- Driver drowsiness and distraction with mmwave radar and camera. Driver drowsiness, fatigue and distraction at the wheel are often the cause of serious accidents worldwide.these in-cabin sensors play a very important role in occupant detection and driver monitoring.
- Obstacle Detection and Identification.

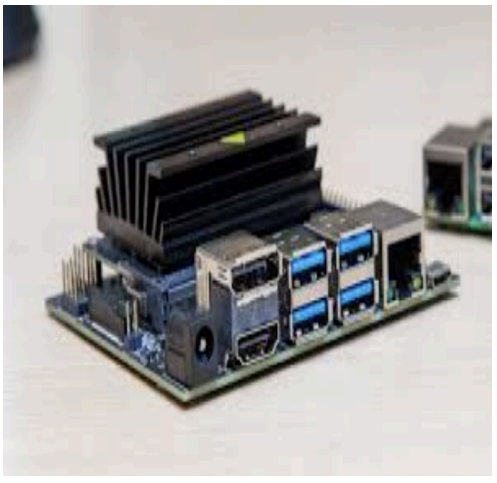
Millimeter-wave radar to detect the position and velocity of the obstacle. Afterwards, the image processing module uses the bounding box regression algorithm in deep learning to precisely locate and identify the obstacles.

System Components

A. Machine Vision Pedestrian Classification Subsystem
B. mmWave Pedestrian Localization Subsystem
C. PCW Fusion System

A. Machine Vision Pedestrian Classification Subsystem

JETSON NANO SINGLE BOARD: INTRODUCTION:
Jetson Nano is a small, powerful single-board Linux computer developed by NVIDIA, which is optimized for image processing and neural networks. [1] Jetson Nano lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. Compared with other single-board computers in wide use like Raspberry Pi, Jetson Nano has better performance in all applications of image processing and neural network, even when an extra compute stick is added to the Raspberry Pi.



A Jetson Nano is equipped with the following components:

- GPU
- CPU
- Memory
- Storage
- Camera
- Connectivity
- Display
- USB
- Others
- Mechanical

KEY FEATURES

128-core Maxwell
Quad-core ARM A57 @ 1.43 GHz

4 GB 64-bit LPDDR4 25.6 GB/s microSD (up to 1TB) 2x MIPI CSI-2 DPHY lanes
Gigabit Ethernet, M.2 Key E HDMI and display port
4x USB 3.0, USB 2.0 Micro-B GPIO, I2C, I2S, SPI, UART
69 mm x 45 mm, 260-pin edge connector
The image/video processing capability of Jetson Nano is as follows:

- Video Encode 4K@30 | 4x1080p@30 | 9x720p@30 (H.264/H.265)

DEMOS AND OBSERVATION:

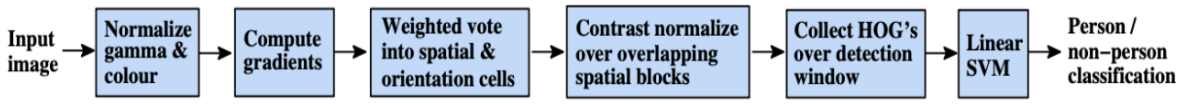
Pedestrian Detection using OpenCV-Python:

OpenCV is an open-source library, which is aimed at real-time computer vision. This library is developed by Intel and is cross-platform – it can support Python, C++, Java, etc. OpenCV is one of the most widely used libraries for Computer Vision tasks like face recognition, motion detection, object detection.

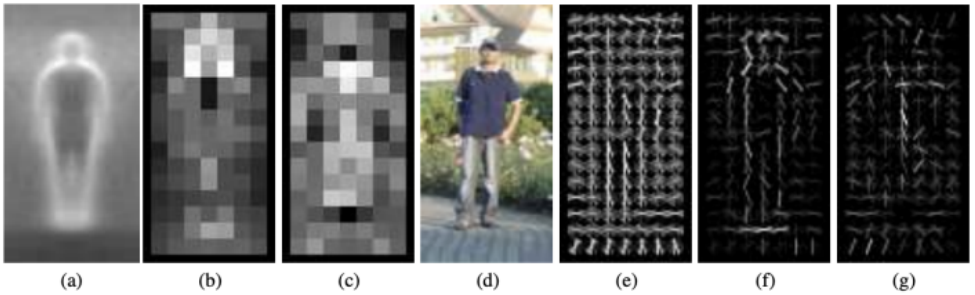
Pedestrian detection is a very important area of research because it can enhance the functionality of a pedestrian protection system in Self Driving Cars.We can extract features like head, two arms, two legs, etc, from an image of a human body and pass them to train a machine learning model. After training, the model can be used to detect and track humans in images and video streams. However, OpenCV has a
built-in method to detect pedestrians. It has a pre-trained HOG(Histogram of Oriented Gradients) + Linear SVM model to detect pedestrians in images,video and real time video streams.

Histogram of Oriented Gradients:

This algorithm checks directly surrounding pixels of every single pixel. The goal is to check how darker is the current pixel compared to the surrounding pixels. The algorithm draws arrows showing the direction of the image getting darker. It repeats the process for each and every pixel in the image. At last, every pixel would be replaced by an arrow, these arrows are called Gradients. These gradients show the flow of light from light to dark. By using these gradients algorithms perform further analysis.

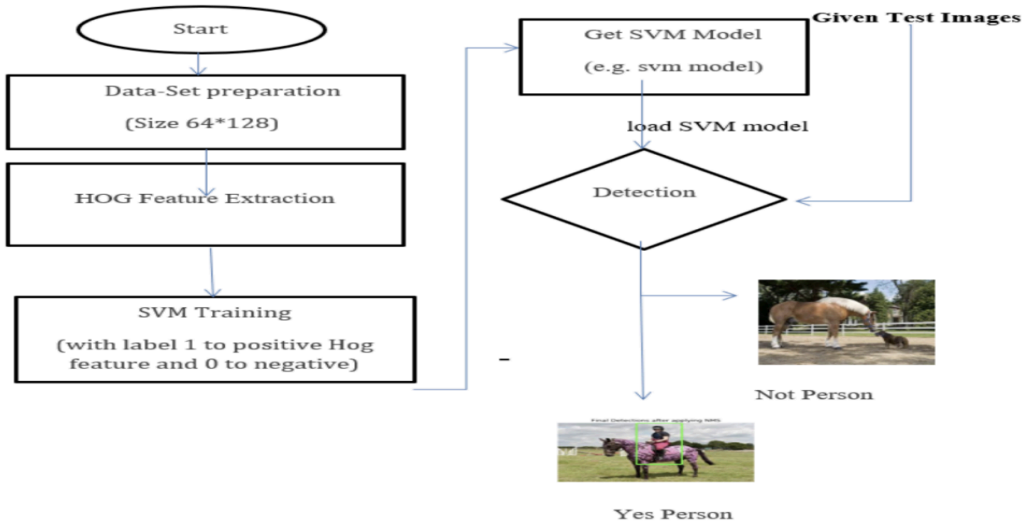


This figure is an overview of our feature extraction and pedestrian detection chain. The detector window is tiled with a grid of overlapping blocks in which Histogram of Oriented Gradient feature vectors are extracted. The combined vectors are fed to a linear SVM for object/non-object classification. The detection window is scanned across the image at all positions and scale conventional non-maximum suppression is run on the output pyramid to detect human/pedestrian



Support Vector Machine (SVM) Classifier:

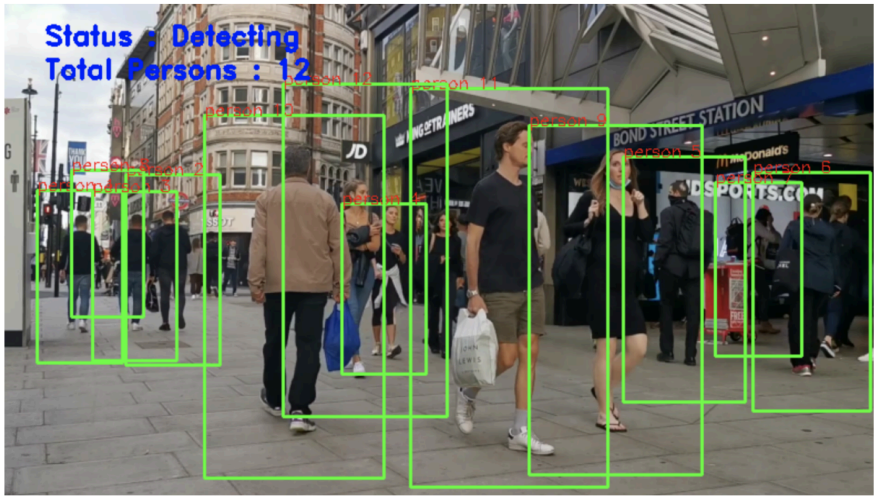
“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for either classification or regression challenges. However, it is mostly used in classification problems. There are various machine learning algorithms e.g. the k-nearest neighbor, Support Vector Machine,….etc. which can be used for training. Here the Linear SVM is used for training and we come up with a model which can be used in the detection algorithm to detect the person.

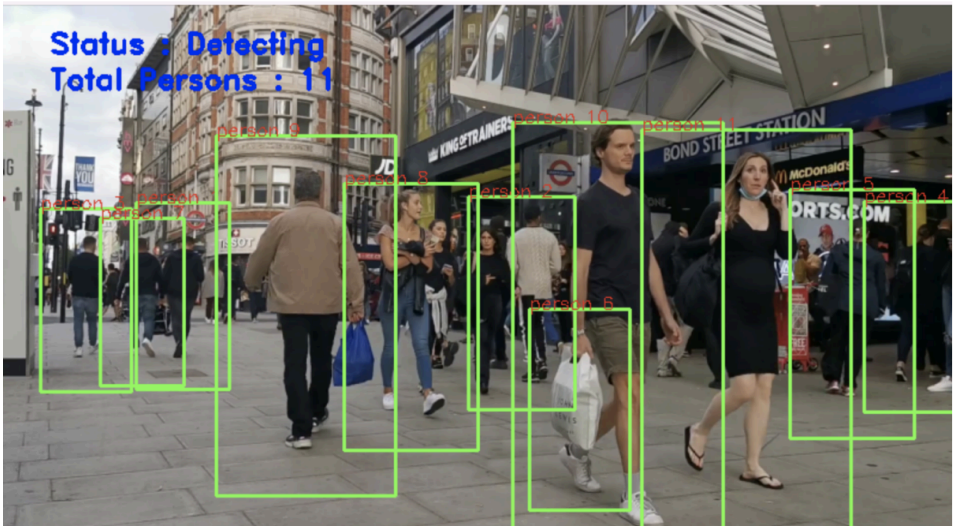


In detection, trained classifiers are used with a sliding window to localize persons in an image or video. Detection results can give more than one detection for a single person, that’s why the NMS algorithm(non-maximum separation algorithm) is used further which will give one detection for each person.
Sliding Window & Non-Maximum Separation Algorithm for Object Detection:
Sliding window is a rectangular region of fixed width and height that “slides” across an image, such as in the following figure:

PEOPLE COUNTING AND DETECTING USING MV:

DEMO ON VIDEOS:





DEMO WITH OTHER VIDEOS:

REALTIME FEED RESULTS

Observations: Classification system works best when full body is in the camera's field of view.As you can see both images have the correctly identified object detected, because of general social distancing and remote learning, we weren't both able to be in a room to test multiple people in a single stream

B. mmWave Pedestrian Localization Subsystem

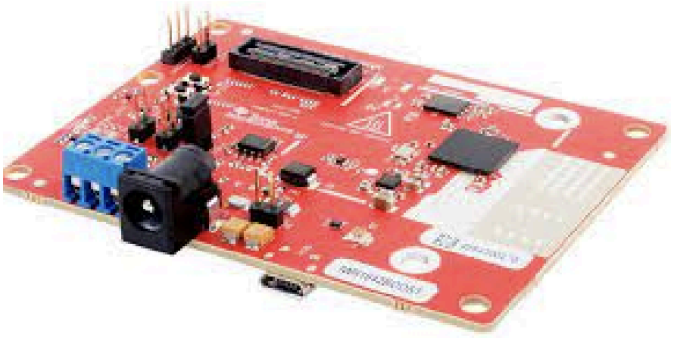
AWR1642BOOST-ODS Evaluation Board Demonstrations

Introduction:

The AWR1642 Obstacle Detection Sensor from Texas Instruments is an easy-to-use evaluation board for the AWR1642 mmWave sensing device, with direct connectivity to the microcontroller (MCU) LaunchPad Development Kit. The Obstacle Detection Sensor contains everything required to start developing software for on-chip C67x DSP core and low-power ARM R4F controllers, including onboard emulation for programming and

debugging as well as onboard buttons and LEDs for quick integration of a simple user interface.
The standard 20-pin BoosterPack headers make the device compatible with a wide variety of TI MCU LaunchPads and enables easy prototyping.
Key Features :

- Two 20-pin LaundPad connectors that leverages the ecosystem of the TI LaunchPad
- XDS110 based JTAG emulation with a serial port for onboard QSPI flash programming
- Back-channel UART through USB-to-PC for logging purposes
- Onboard antenna
- 60-pin, high-density (HD) connector for raw analog-to-digital converter (ADC) data over LVDS and trace-data capability
- Onboard CAN-FD transceiver
- One button and two LEDs for basic user interface



- 5-V power jack to power the board

SOFTWARE:

- Code Composer Studio v10.3
- Uniflash v6.3
- TI mmWave SDK v3.5
- TI mmWave Studio v2.01
- Matlab Runtime v9.2
- TI mmWave Automotive Toolbox v3.4
- TI mmWave Industrial Toolbox v4.7

APPLICATIONS:

- Occupancy detection
- Motion detector
- Automated doors and gates
- IP network camera
- Smoke and heat detector
- Lighting sensors

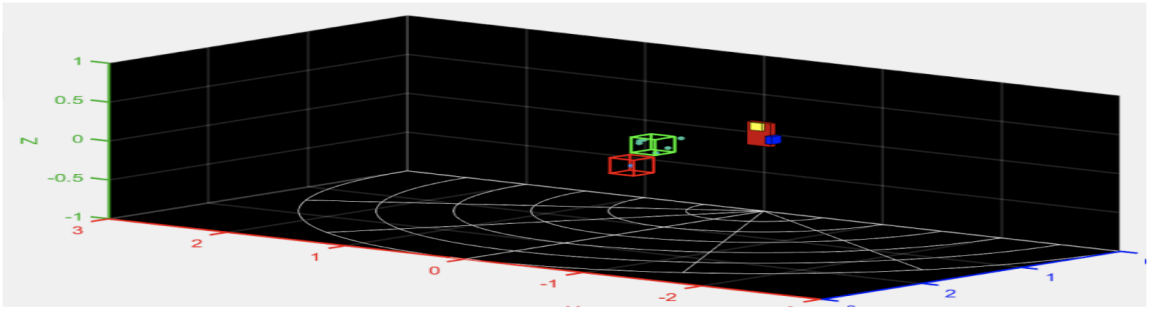
System Description:

Industrial and building automation security systems can use radar to detect and track humans and other objects. In a security system, mmWave technology provides range, velocity, and angle information that is immune to environmental effects.Human monitoring

has become an important area of exploration, due to its potential for understanding people's count, activities, intents, and health issues.
Accuracy and precision play an important role in these applications. While sensors such as passive infra-red (PIR) and time of flight (TOF) are in use, they suffer from limitations in accuracy, false alarms, and environmental changes such as darkness, brightness, and smoke. Radars allow an accurate measurement of distances, relative velocities of people, and other objects. They are relatively immune to environmental conditions such as the effects of rain, dust, or smoke. Additionally, they can work in complete darkness or in bright daylight. They are therefore useful for building automation applications such as people counting, motion detection, IP network cameras, and safety guards

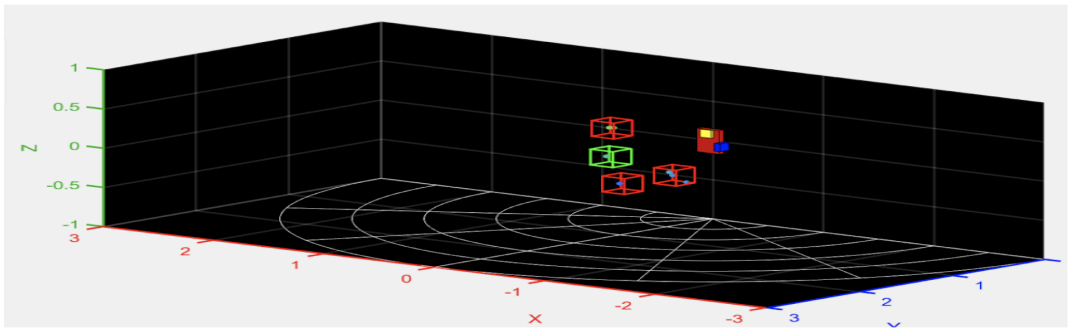
To test the capabilities of the AWR evaluation board, we first ran some of Texas Instrument's pre-designed demonstrations:

Demo #1 TI's Obstacle Detection: Detects obstacles in close proximity to the sensor

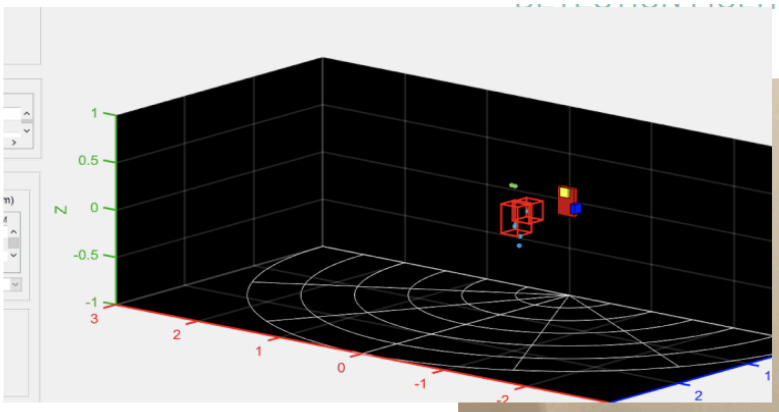


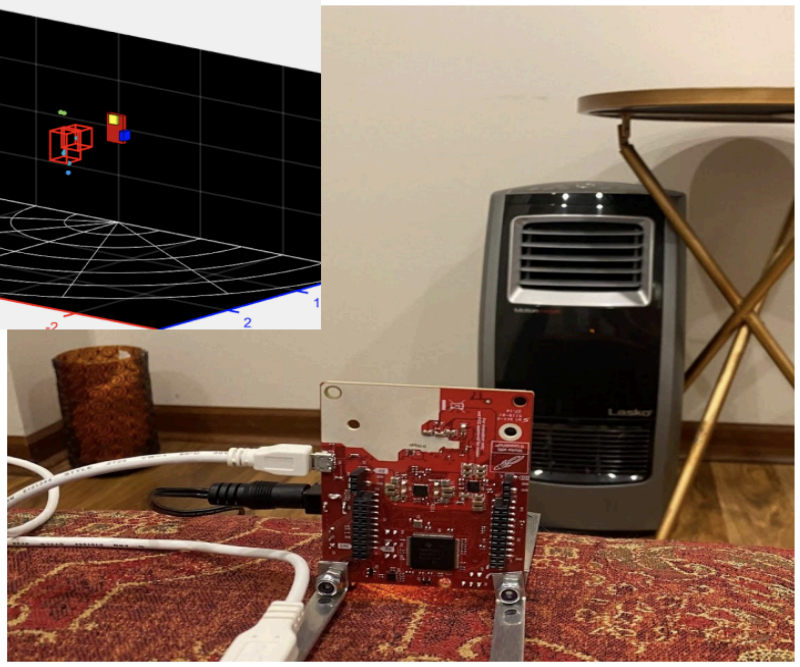
Single Object

The first one being obstacle detection within 5m of the sensor



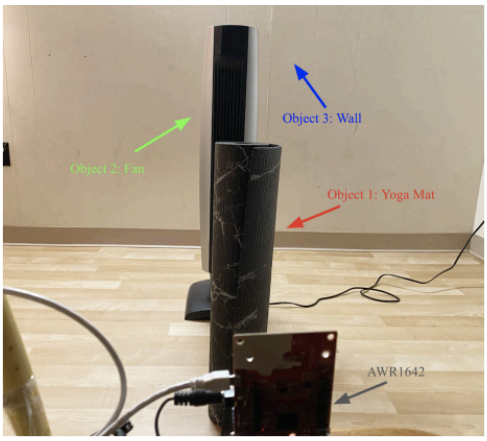
DETECTION MULTIPLE OBJECTS

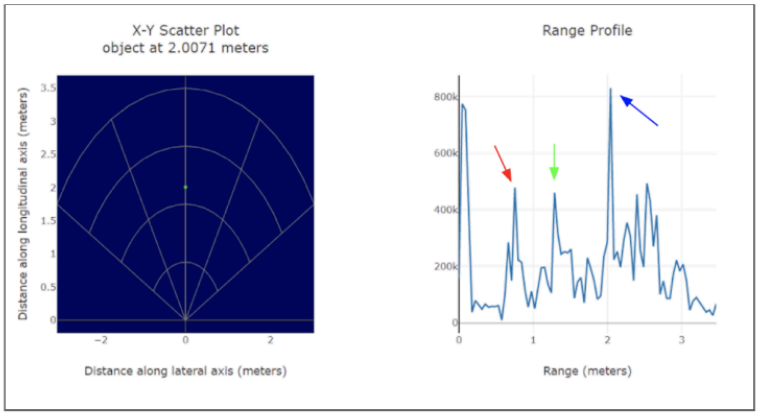




OBJECTS CLOSE TOGETHER

Demo #2 TI's High Accuracy Range: Detects objects with high range accuracy to the level of millimeters and returns the measurement for the highest peak.



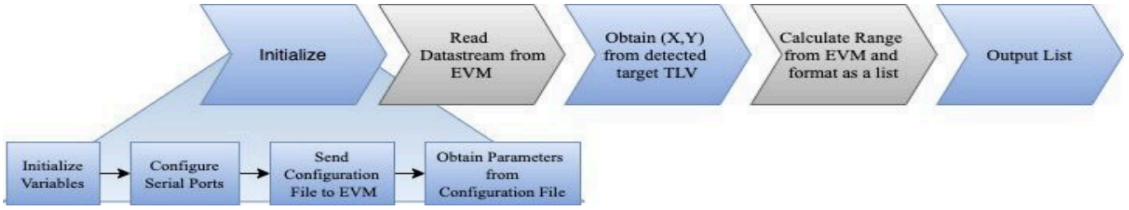


mmWAVE SENSOR PEOPLE COUNTING AND PEDESTRIAN DETECTION:

AWR1642BOOST Programming:

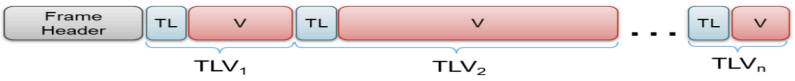
The AWR1642BOOST EVM is programmed with prebuilt binaries from TI's "16xx - People Counting" demo in the mmWave Industrial Toolbox v4.1,it Performs low level digital signal processing on the C674x to get the cloud points and Performs high level group tracking algorithms on the ARM Cortex-R4

The EVM outputs a stream of Type-Length-Value packets with information such as Cartesian position and velocity for the point cloud and the detected targets



Dataframe output:

A TLV(type-length-value) encoding scheme is used with little endian byte order. For every frame, a packet is sent consisting of a fixed sized Frame Header and then a variable number of TLVs depending on what was detected in that scene. The TLVs can be of types representing the 2D point cloud, target list object, and associated points. It outputs one frame every frame period. The frame has a fixed header, followed by a variable number of segments in tag, length, value (TLV) format. Each TLV has a fixed header, followed by a variable-size payload



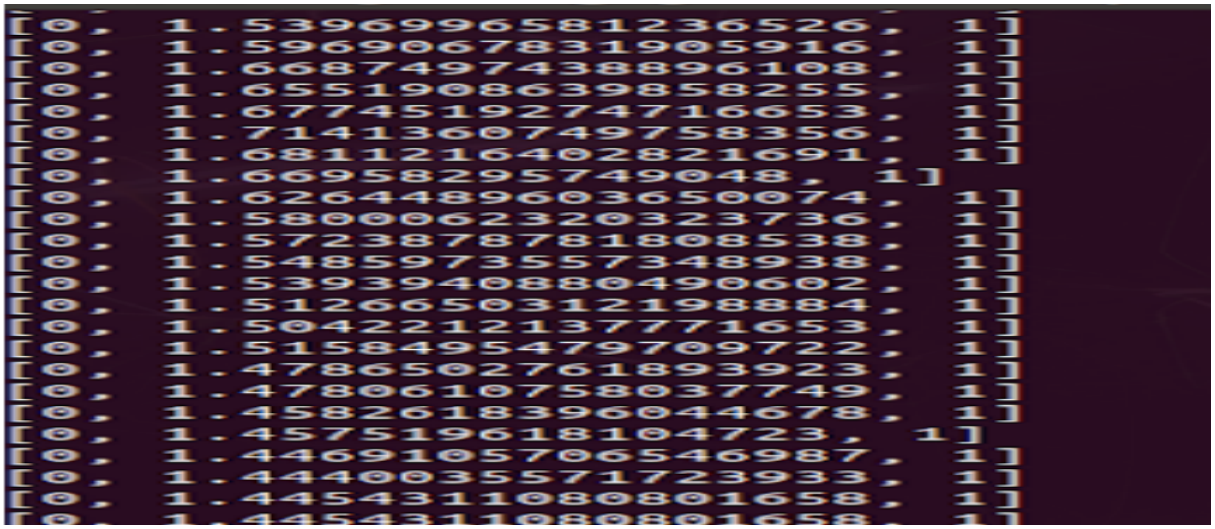
The frame header is a fixed size (52 bytes) ,Each TLV has a fixed header (8 bytes) Each sensor packet includes 4 bytes of the frame number, 4 bytes of the number of objects detected, and 12 bytes of tracking information for each object.The tracking information for each object is 12 bytes which include 4 bytes track ID, 4 bytes target position in the X dimension, and 4 bytes target position in the Y dimension.

Python Data Extraction:

- Establishes serial connection with the EVM to send the configuration file and retrieve the data
- Parses the datastream to extract the information from the detected target's TLV packets.
- Calculates range from EVM to target using the Cartesian position.
- Format as a list

Function Output:

[ID, Rand]Format: ID, range, # of targets (ID is given every time new target is detected) TI Demo GUI Tracking Screenshot

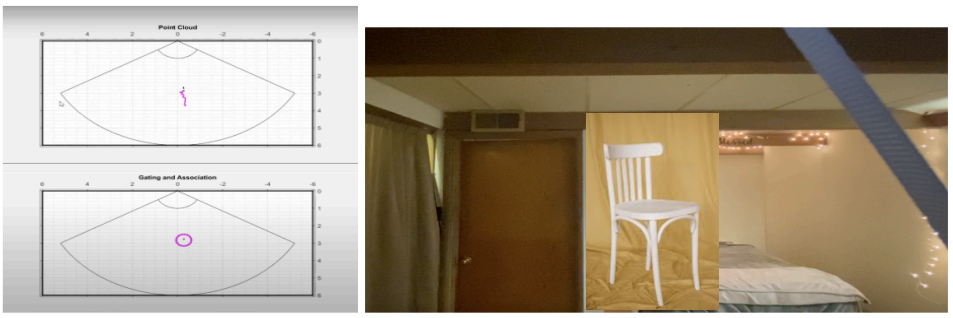


#2

```
[0, 1.5396996581236526, 1]
[0, 1.5969067831905916, 1]
[0, 1.6687497438896108, 1]
[0, 1.6551908639858255, 1]
[0, 1.6774519274716653, 1]
[0, 1.7141360749758356, 1]
[0, 1.6811216402821691, 1]
[0, 1.66958295749048, 1]
[0, 1.6264489603650074, 1]
[0, 1.5800062320323736, 1]
[0, 1.5723878781808538, 1]
[0, 1.5485973557348938, 1]
[0, 1.5393940880490602, 1]
[0, 1.5126650312198884, 1]
[0, 1.5042212137771653, 1]
[0, 1.5158495479709722, 1]
[0, 1.4786502761893923, 1]
[0, 1.4780610758037749, 1]
[0, 1.4582618396044678, 1]
[0, 1.457519618104723, 1]
[0, 1.4469105706546987, 1]
[0, 1.4440035571723933, 1]
[0, 1.4454311080801658, 1]
[0, 1.4454311080801658, 1]
```

Note: Terminal Output filters out when no targets found

Test using TI's provided Matlab Visualizer:



detected

Missed detection

The Python program extracts the x and y coordinates of the targeted people from the datastream. The code to extract the information from the datastream is based from ibaiGorodo's Github. Instead of showing a GUI, it calculates the range using the x,y coordinates from the EVM. The output of the data extraction code is a list [targetId, range, numTargets] where numTargets is the total number of people detected and targetId is their identifier.

mmWave Pedestrian Localization Subsystem

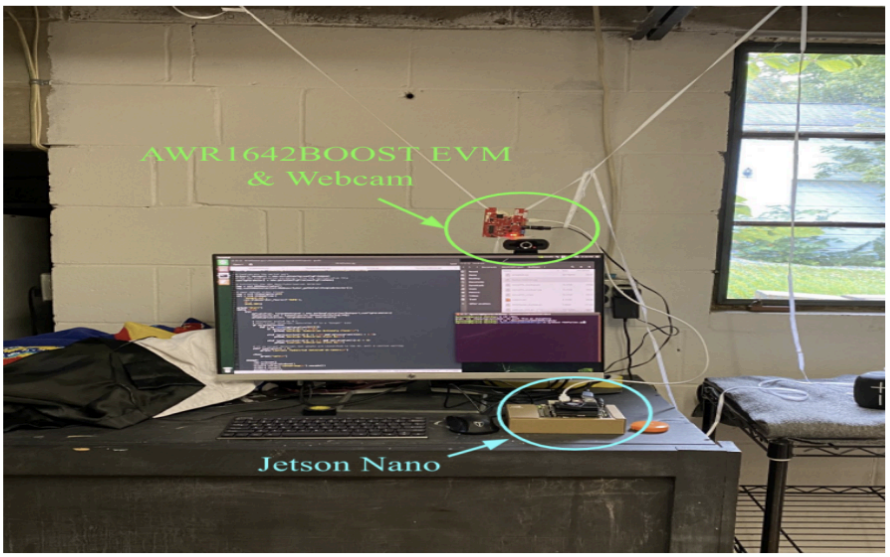
[WARNING DISTANCE CALCULATIONS:
(nacto = national association of city transportation officials) 25 mph = 85 ft (25m) to stop
55 ft perception reaction time = 16.7 m

Constraints & Challenges:

1. Real-time Constraint: The MV Pedestrian Classification Subsystem takes approximately 0.6 seconds to process a video frame
   a. In 0.6 seconds, a car going 25 mph would travel 6.7m which is 80% of its Stopping Distance
2. Testing Environment Constraints:
   a. Limited space
   b. Limited testing personnel
   c. Make sure your hardware is good before use, give your time for shipping and processing in the event a bad board or hardware is received by the vendor.
3. More AWR miss detections in the Fusion System then individually
   a. Possible Cause: the real-time delay of the MV subsystem is affecting the AWR output
   b. Possible Solution: run both subsystems in parallel and thread their outputs together to make a decision

Pedestrian Collision Warning Fusion Results:

SYSTEM SETUP:



Sensor setup ~1.5m above the ground

```
Caution: Pedestrian Detected on Camera

[] 0 1
Caution: Pedestrian Detected on Camera

[[0, 2.0360612745901085, 1]] 1 1
Warning: Pedestrian Close By

[] 0 1
Caution: Pedestrian Detected on Camera

[] 0 1
Caution: Pedestrian Detected on Camera

[[0, 2.0953290425750772, 1]] 1 1
Warning: Pedestrian Close By

[] 0 1
Caution: Pedestrian Detected on Camera

[] 0 1
Caution: Pedestrian Detected on Camera

[[0, 2.1706363189463858, 1]] 1 1
Warning: Pedestrian Close By

[] 0 1
Caution: Pedestrian Detected on Camera
```

Future Work:

Make the system headless for the Vision Impaired, or have a difficult time remembering, as the system can “sense” what’s in front of the user and warn of potential obstacles like a long flight of stairs or a crosswalk. Machine Learned Seizure Response system. Assisted Sensory

with Computer Vision. Ability to detect if something fell on the floor. Avoiding pot-holes can be a machine learned data point, but verified by the computer vision system enhanced by headless or retrofittable models. As Privacy Protections and concerns arise, they be alleviated with the help of these the sensor systems as they do not necessarily need a camera to perform regulated health measures like occupancy control counters in a pre-defined space, i.e, movie theater lines, classrooms, or DMV lines all without a camera and still live output that information to a centralized and ML modeling. This system can later alert authorities or users if a limit has been detected, again without the camera's input. Low light camera settings where an expensive and unreliable camera video stream could be avoided. All information consumed by EVM sensors, and processed through the ML platform. Peripheral vision detection without a camera for privacy protocols. Only sensor, no camera is needed to satisfy the privacy concerns (restrooms) but still need operational assistance i.e if an elderly has fallen in the bathroom tub. The sensors were tested upto 3 meters for this experiment.
Make the PCW System headless for operational use on vehicles. Have the system return the driver's display or automatic braking functionality. Implement different Machine Learning algorithms on the Machine Vision Subsystem to compare their performance in accuracy and processing time.

References

- https://developer.nvidia.com/embedded/jetson-nano-developer-kit
- https://www.researchgate.net/publication/329113452_Pedestrian_Detection_Based_on_Fusion_of_Millimeter_Wave_Radar_and_Vision
- https://www.researchgate.net/publication/257581231_Designing_an_integrated_driver_assistance_system_using_image_sensors
- https://medium.com/@richa.agrawal228/person-detection-in-various-posture-using-hog-feature-and-svm-classifier-2c3a3991022c
- https://github.com/ibaiGorordo/AWR1642-Read-Data-Python-MMWAVE-SDK-2/blob/master/People%20counting%20demo/peopleCountingDemo.py
- https://dev.ti.com/tirex/explore/node?a=VLyFKFf 4.1.0&node=AO1DWYiQ.dLcRazksGRNng VLyFKFf 4.1.0&r=VLyFKFf LATEST
- https://training.ti.com/sites/default/files/docs/MainDemoSlidesMAINEDIT_V1p11_0.pdf
- https://thedatafrog.com/en/articles/human-detection-video/
- https://morioh.com/p/f7343c9642c3
- https://dev.ti.com/tirex/explore
- H. Panduranga, D. Stocchero. "AWR1642BOOST mmWave Radar Sensor Research and Application" ECE597-130 Final Report.
- https://www.compel.ru/wordpress/wp-content/uploads/2018/04/mmwave-tehnologiya-iskusstvennogo---zreniya---dlya-promyishlennyih-i-ohrannyih-sistem. pdf
- https://www.ti.com/lit/wp/spyy005a/spyy005a.pdf?ts=1624046639883&ref_url=https%253A%252F%252Fwww.google.de%252F
- https://www.pathpartnertech.com/mmwave-radar-and-camera-based-in-cabin-sensing/