Parameter Setting and Reliability Test of a Sensor System for Driver Height Estimation in a Car Using Convolutional Neural Network

Master of Engineering
Information Technology
Shafait Azam
Matriculation No: 1345243
shafait.azam@stud.fra-uas.de

Master of Engineering
Information Technology
Mashnunul Huq
Matriculation No: 1384042
Mashnunul.huq@stud.fra-uas.de

***Abstract*—An ultrasonic sensor is a device that uses high-frequency sound waves to detect objects and measure distances. It works by emitting sound waves that bounce off nearby objects and return to the sensor, which then calculates the distance based on the time it takes for the sound waves to return. Red Pitaya is a programmable hardware platform that can be used for a wide range of applications, including signal processing, data acquisition and control. A Convolutional Neural Network (CNN) is a type of deep neural network commonly used in image recognition. It is designed to automatically learn and extract features from images through a series of convolutional and pooling layers. The purpose of this paper is to demonstrate a mechanism that uses a sensor system (ultrasonic sensors connected with Red Pitaya) to estimate the height of driver in a car with the help of Convolutional Neural Network.**

Keywords—Ultrasonic sensor, Red Pitaya, Convolutional Neural Network (CNN), Signal Processing.

I. INTRODUCTION

Ultrasonic sensors are a type of non-contact sensor that utilize sound waves to measure distance, presence, or level of objects. These sensors emit high-frequency sound waves that bounce off objects and return to the sensor, allowing it to calculate the distance between itself and the object. Ultrasonic sensors are commonly used in a variety of applications, including automotive parking assistance, driver's height measurement, weight estimation and various fields of robotics. They can detect objects in a wide range of materials, including metal, plastic, and liquids, making them versatile and useful in many different settings. One of the key advantages of ultrasonic sensors is their ability to operate in harsh environments. Ultrasonic sensor follows rudimentary principle of sound propagation and reflection within ultrasonic frequency range. It can function promisingly in conditions where light intensity is low and dark. To determine the height of an object or human, one can arrange some ultrasonic sensors, mounted parallel, and then place an object within the detection area of these ultrasonic sensors [1]. Ultrasonic sensor provides accurate distance measurement of an object. The transmitter of the sensor transmits ultrasonic waves, and when the waves hit an object, part of their energy reflected back to receiver of the sensor as echo signal. The distance to the object can be calculated through the speed of the ultrasonic waves in the media and the angle [1]. Fig. 1. demonstrates the working principle of the ultrasonic sensor. Ultrasonic sensors can measure up to 10m target distance approximately. The inclination surface of the object as the waves hit may cause deflected part of it away from the sensor's receiver and then the sensing accuracy is decrease.

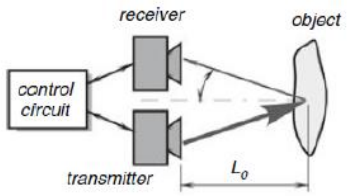

Fig. 1. Ultrasonic sensor's working principle [1].

The frequency ranges of ultrasonic waves are from 10 kHz to 1 GHz [2]. Because of the frequencies cannot be heard by human and the noise of sound from surrounding cannot affect the performance of this sensor as its frequency still below of the sensors operating frequency, ultrasonic sensor is a great tool for data acquisition in harsh environments [1].

In this project, FIUS Ultrasonic sensors is mounted with the Red Pitaya in order to detect human and measure the approximate height. Red Pitaya is powerful open-source hardware platform that can perform a wide range of tasks, including scientific research and industrial automation. It is based on a field-programmable gate array (FPGA) and a dual-core ARM Cortex-A9 processor, allowing for real-time signal processing and control. Red Pitaya platform also includes various peripherals, such as high-speed analog-to-digital converters (ADCs) and digital-to-analog converters (DACs), as well as GPIO, Ethernet, USB, and HDMI interfaces. One of the major advantages of Red Pitaya is its adaptability and modularity. The hardware can be customized and extended with various plug-in modules, including RF front ends, power amplifiers, and sensors. Additionally, the platform's software is open source, meaning it can be modified and extended to meet specific needs. This makes Red Pitaya an excellent choice for prototyping and testing new ideas, as well as for creating custom solutions for specific applications. The combination of two ultrasonic sensors and Red Pitaya have been used to acquire raw data from the environment, which is then pre-processed and tuned in order to fetch them in a CNN model. CNN model is responsible for the detection of human and non-human objects. Depending on the position of the two sensors, the approximate height of the person is then measured, and confusion metrics have been generated.

II. METHODOLOGY

Two sensors were used in this project to receive the overlapped interference of the ultrasonic signal which is essential for the height estimation.

A. Data Acquisition

In the height estimation problem, two sensors were needed because the one FIUS sensor was not enough to cover the whole experimental space.

Fig. 2. Working environment which was used for height detection project.

From Fig. 2, two sensors are mounted in the dashboard in a way that these two can cover the whole driver's seat from top of the head rest to the bottom of the seat. To detect the overlapping of these to sensor's signal interference an experiment was conducted. In the Fig. 2, there is a box which is hanging on the head rest was at first at the top of the head rest. Only the upper sensor then could detect the box. As the length of the rope was gradually increased the box started to fall. This whole experiment was monitored in the FIUS Ultrasonic Sensor UI. When the lower sensor started to detect the box, that position has been marked because that was our overlapping of two sensors. In the Fig. 2, the purple line indicates the length, calculated from the roof of the car where lower sensor can not detect anything. The red line indicates the length of car from the roof top to the bottom of seat. The green line indicates the length where slanted upper sensor can detect. The blue line indicates the length where slanted lower sensor can detect. To collect some, signal overlapped data upper sensor was tilled down approximately 4.5 degree because it was impossible to change the angle of the lower sensor because it was hard mounted on the dashboard. Theta 1 indicates the slanting angle of the upper sensor with horizontal ground which is approximately 19.47 degree. Theta 2 indicates the slanting angle of the lower sensor with horizontal ground which is approximately 17.46 degree. Distance from the dashboard to the seat is around 89 cm.

9500 samples were combinedly collected for the height estimation (upper sensor 3750 and lower sensor 5750). First, there were no person present, the seat was adjusted in the standard position and tilted exactly 90 from the ground. Both sensors captured the ADC (analog-to-digital) data. Then, there was a person present in the car whose presence can be detected by both sensors. Same procedures were followed, and ADC data were collected. A person of very low height was also present to be only detect by the lower sensor which resemblances the infant or very small person detection.

Again, same procedures were followed, and ADC data were collected. For the crossing, Some of the FFT data were also collected from this experiment.

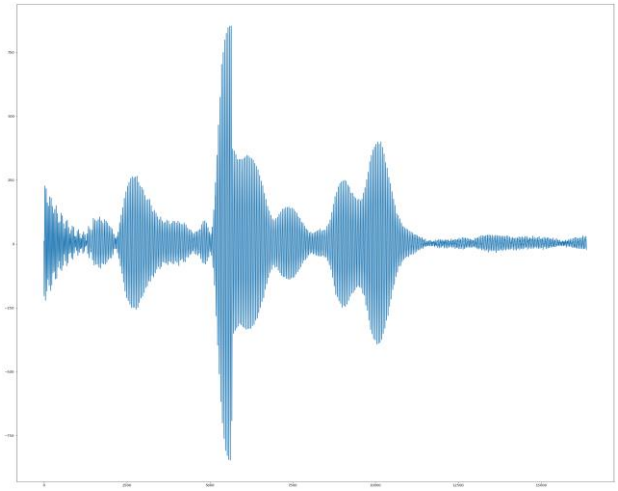

Fig. 3. ADC signal of the Upper Sensor when a person was present in car.

From the Fig. 3, the reflection or the echo captured by the sensor is clearly visible when a person is present in the car.

B. Data Preprocessing

In order to mine feasible information from a ADC signal for specific application purpose any signal needs to be pre-processed. Later, these processed data were fetched to the CNN to detect a person in the car.

1) Labelling the Data: In order to label the data, human eye observation was used. If a person or very small persent was present during the experiment, signal acquired by both sensors were labelled as person. If a person was not present during the esperiment, signal acquired by both sensors were labelled as nonperson.

2) Removing Header from the Raw Signal: In order to clean the data set and use the ADC data for further processing first 16 bit header where removed.

3) Generating Power Spectral Density: To measure the power spectral density the sliding window technique was used. Applied fix window size of 1024 on the ADC signal with sliding step of 512 and calculated FFT of the applied windowed portion of the signal. By squaring the FFT portions and devide them by window size PSD was calculated.

4) Generating Spectrogram: Spectrogram was generated only with the first 512 data points of PSD. Because the lower frequencies of the power spectral density array contain infromation about the overall trend or behaviour of the signal over time. The PSD is a plot of the power of a signal as a function of frequency, and lower frequencies

correspond to slower changes in the signal. After generating spectrograms, they were then resized in a shape which is similar to the input shape(256*256*1) of a CNN. These spectrograms then converted to logarithmic scale for better view.

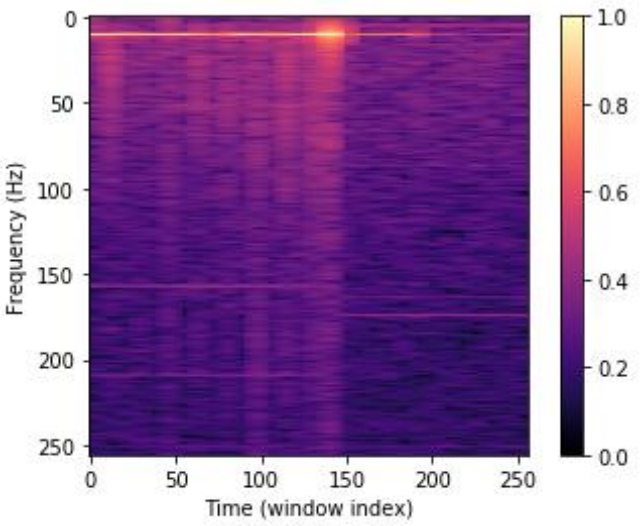

Fig. 4. Spectrogram in Logarithmic scale.

5) Convolutional Neural Network Architecture: CNN are built to process data that come in the form of multi-dimentional arrays. There are four advantages of the CNN that makes it so popular: local connections, shared weights, pooling and the use of many layers. The usual architecture of a typical CNN has convolutional layers with its corresponding pooling layers [3]. At the end of these layers, fully connected layers are also defined. Units in a convolutional layer are devided in feature maps, within which each unit is connected to local patches in the feature maps of the pevious layer through a set of weights called filtetr bank. The result of the local weight sum is then passed through a non-linearity such as a ReLU. All units in a feature map share the same filter banks [3]. Deep neural networks exploit the property that many natural signals are compositional heirarchies, in which higher-level features are obtained by lower -level ones.In images, local combinations of edges from motifs, motifs assemble into parts, and parts form objects [3]. Signal processing systems with deep convolutional neural network architectures are composed of multiple layers of non-linear processing stages, where each lower layer's outputs are fed to its immediate higher leayer as the input [4]. For this project, we also have defined our own neural network. As we have two sesnors, we have to defined to CNN architectures each for one sensor's generated spectrogram.

a) CNN ONE for Upper Sensor Spectrograms: For the training and testing of the spectrograms CNN one has been defined with its own convolutional layers, pooling layers and two fully connected dense layers. To counter the overfitting problem we have also attached two dropout layers. In all the convolutional layers ReLU(Rectified Linear Unit) activation is used. In the first fully connected dense layer used activation function is ReLU. And the last layer which is fully connected dense layer, sigmoid function has been used as an activation function.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 254, 254, 32)	320
max_pooling2d_1 (MaxPooling2	(None, 127, 127, 32)	0
conv2d_2 (Conv2D)	(None, 125, 125, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 62, 62, 64)	0
dropout_1 (Dropout)	(None, 62, 62, 64)	0
conv2d_3 (Conv2D)	(None, 60, 60, 128)	73856
max_pooling2d_3 (MaxPooling2	(None, 30, 30, 128)	0
conv2d_4 (Conv2D)	(None, 28, 28, 256)	295168
max_pooling2d_4 (MaxPooling2	(None, 14, 14, 256)	0
flatten_1 (Flatten)	(None, 50176)	0
dense_1 (Dense)	(None, 64)	3211328
dropout_2 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 1)	65

Total params: 3,599,233
Trainable params: 3,599,233
Non-trainable params: 0

Fig. 5. Convolutional neural Network for Upper Sensor's Spectrograms.

Before fetching the spectrograms to CNN, we have encoded our dataset with binary values where person = 1(reflection or echo) and nonperson = 0(without reflection or no echo). Then we have reshaped the data for CNN. Then we have split our data set by taking 15% of whole data set for testing purpose and 85% for training. In the training phase we have used Adaptive Moment Estimation (ADAM). Because ADAM is widely used optimization algorithm in machine learning and deep learning [5]. It is a stochastic gradient descent optimizer that utilizers adaptive learning rates and momentum-based updates. ADAM optimizer maintains an exponentially decaying average of the past gradients and past squared gradients of the parameters, which are used to calculate the current gradient update. Specially, it computes the first and second moments of the gradients, known as the mean and variance. These moments are used to adjust the learning rate for each parameter in the model, with higher learning rate for less frequently updated parameters. The ADAM optimizer also incorporates momentum updates to help accelerate convergence and smooth out noisy gradients [5]. For the loss function we have used Binary cross entropy because it is commonly used for binary classification problems. It is used to measure the difference between the predicted probability distribution and the true probability distribution. In binary classification problems, the output of the model is a single scaler values between 0 and 1 that represents the predicted probability of belonging to one of the two classes. The true label is represented as a binary value, either 0 or 1 [6]. Our defined CNN in Fig. 5, has been trained with batch size of 32 and for validation 20% of training data was used. In the Fig. 6, all the ten epochs of the first CNN can be seen, where the

training accuracy, loss, validation accuracy and validation loss has been depicted.

```
Train on 2548 samples, validate on 638 samples
Epoch 1/10
2548/2548 [==============================] - 110s 43ms/step - loss: 0.5521 - accuracy: 0.6817 - val_loss: 0.2029 - val_accuracy: 0.9295
Epoch 2/10
2548/2548 [==============================] - 111s 43ms/step - loss: 0.2452 - accuracy: 0.9042 - val_loss: 0.1795 - val_accuracy: 0.9342
Epoch 3/10
2548/2548 [==============================] - 115s 45ms/step - loss: 0.1963 - accuracy: 0.9325 - val_loss: 0.1064 - val_accuracy: 0.9561
Epoch 4/10
2548/2548 [==============================] - 112s 44ms/step - loss: 0.1739 - accuracy: 0.9576 - val_loss: 0.0867 - val_accuracy: 0.9765
Epoch 5/10
2548/2548 [==============================] - 113s 44ms/step - loss: 0.1692 - accuracy: 0.9588 - val_loss: 0.0647 - val_accuracy: 0.9796
Epoch 6/10
2548/2548 [==============================] - 115s 45ms/step - loss: 0.1473 - accuracy: 0.9572 - val_loss: 0.1116 - val_accuracy: 0.9514
Epoch 7/10
2548/2548 [==============================] - 111s 44ms/step - loss: 0.1300 - accuracy: 0.9717 - val_loss: 0.0538 - val_accuracy: 0.9828
Epoch 8/10
2548/2548 [==============================] - 112s 44ms/step - loss: 0.1271 - accuracy: 0.9765 - val_loss: 0.0499 - val_accuracy: 0.9843
Epoch 9/10
2548/2548 [==============================] - 113s 44ms/step - loss: 0.1236 - accuracy: 0.9765 - val_loss: 0.0374 - val_accuracy: 0.9906
Epoch 10/10
2548/2548 [==============================] - 113s 44ms/step - loss: 0.0982 - accuracy: 0.9823 - val_loss: 0.0327 - val_accuracy: 0.9937
563/563 [==============================] - 6s 11ms/step
Test accuracy: 0.9840142130851746
```

Fig. 6. CNN ONE Training Phase.

b) CNN TWO for Lower Sensor Spectrograms: For the training and testing of the spectrograms CNN two has been defined with its own convolutional layers, pooling layers and two fully connected dense layers. To counter the overfitting problem we have also attached two dropout layers. The design of CNN two is same as CNN one depicted in the Fig. 5. ADAM has been used as an optimizer and for loss function Binary Cross entropy has been used.

```
Train on 3568 samples, validate on 893 samples
Epoch 1/10
3568/3568 [==============================] - 159s 45ms/step - loss: 0.5209 - accuracy: 0.7287 - val_loss: 0.2005 - val_accuracy: 0.9507
Epoch 2/10
3568/3568 [==============================] - 168s 47ms/step - loss: 0.1322 - accuracy: 0.9546 - val_loss: 0.0195 - val_accuracy: 1.0000
Epoch 3/10
3568/3568 [==============================] - 175s 49ms/step - loss: 0.1049 - accuracy: 0.9922 - val_loss: 0.0079 - val_accuracy: 1.0000
Epoch 4/10
3568/3568 [==============================] - 163s 46ms/step - loss: 0.0922 - accuracy: 0.9978 - val_loss: 0.0112 - val_accuracy: 1.0000
Epoch 5/10
3568/3568 [==============================] - 161s 45ms/step - loss: 0.0925 - accuracy: 0.9994 - val_loss: 0.0373 - val_accuracy: 1.0000
Epoch 6/10
3568/3568 [==============================] - 160s 45ms/step - loss: 0.0881 - accuracy: 0.9983 - val_loss: 0.0062 - val_accuracy: 1.0000
Epoch 7/10
3568/3568 [==============================] - 160s 45ms/step - loss: 0.0859 - accuracy: 0.9969 - val_loss: 0.0328 - val_accuracy: 1.0000
Epoch 8/10
3568/3568 [==============================] - 160s 45ms/step - loss: 0.0773 - accuracy: 0.9989 - val_loss: 0.0049 - val_accuracy: 1.0000
Epoch 9/10
3568/3568 [==============================] - 161s 45ms/step - loss: 0.0799 - accuracy: 0.9994 - val_loss: 0.0032 - val_accuracy: 1.0000
Epoch 10/10
3568/3568 [==============================] - 159s 45ms/step - loss: 0.0781 - accuracy: 0.9989 - val_loss: 0.0072 - val_accuracy: 1.0000
788/788 [==============================] - 9s 11ms/step
Test accuracy: 1.0
```

Fig. 7. CNN TWO Training Phase.

In the Fig. 7, all the ten epochs of the second CNN can be seen, where the training accuracy, loss, validation accuracy and validation loss has also been depicted.

III. RESULT ANALYSIS

This project is basically consisting of two phases. First, is to detect a person. Second, depending on the position and the angle of the sensor estimate approximate height such as small/infant, average sized person, and no person. As we have created two CNN to process each sensor's data, so our generated confusion matrix is also two.

A. Upper Sensor's Testing Result

As we have splitted the whole data set into train and test set, we have tested 563 data. The test set accuracy is around 98.40% depicted in Fig. 6. In Fig. 8, the confusion matrix of the CNN ONE has been generated, where 0 = without reflection/ nonperson and 1 = reflection/ person. Here, True Positive (TP) = 212, True Negative (TN) = 342, False Negative (FN) = 6 and False Positive (FP) = 3.

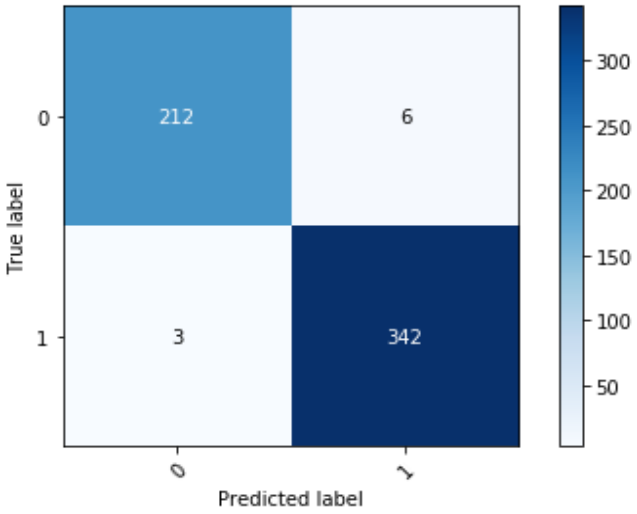

Fig. 8. Confusion Matrix of test Data in CNN ONE.

B. Lower Sensor's Testing Result

As we have splitted the whole data set into train and test set, we have tested 788 data. The test set accuracy is around 100% depicted in Fig. 7. In Fig. 9, the confusion matrix of the CNN ONE has been generated, where 0 = without reflection/ nonperson and 1 = reflection/ person. Here, True Positive (TP) = 215, True Negative (TN) = 573, False Negative (FN) = 0 and False Positive (FP) = 0.

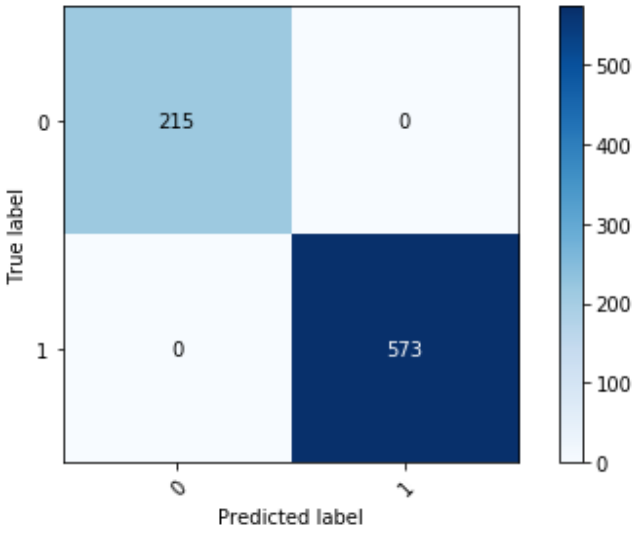

Fig. 9. Confusion Matrix of test Data in CNN TWO.

C. Height Estimation Result

For the detection of height, we had to run two CNN at the same time with their corresponding spectrograms. It requests significant amount of computational power. So, we have tested only 500 combined data, captured while both sensors were active. Among them 250 data was for regular sized human, 200 was for infant or very small and 50 was for no person. In the table I, we can see that 248 out of 250 regular sized humans were detected correctly. We can see

accuracy in this case 99.2%. In case of infant/very small person, 195 out of 200 were detected correctly by lower sensor and upper sensor detected nothing (means label 0). So, the accuracy is 97.5%. And in the case of no person, 45 out of 50 have been detected correctly. This because in some cases different kind of noise interferes with the sonar signal and creates distortion in the spectrogram. That's why the accuracy plummeted down to 90%.

TABLE I: Height Estimation Results.

Actual		Predict	
Size	No of test Samples	Size	Predicted Correct Samples
Regular Sized Human (UpperSensor = 1, LowerSensor = 1)	250	Regular Sized Human (UpperSensor = 1, LowerSensor = 1)	248
Infant/ very small Human (UpperSensor = 1, LowerSensor = 1)	200	Infant/ very small Human (UpperSensor = 1, LowerSensor = 1)	195
No Human (UpperSensor = 1, LowerSensor = 1)	50	No Human (UpperSensor = 1, LowerSensor = 1)	45

IV. FUTURE SCOPE

When we are dealing with the Ultrasonic signals, it is essential to detect the time duration of the reflected signal. For this reason, the calculation of the first peak of the reflected signal or echo is necessary. We have successfully designed an algorithm which can successfully detect the first reflection of a signal in a spectrogram. This has been done by both thresholding and morphological operations. In the Fig. 10, the '+' sign denotes the first reflection of a signal. This is because of the super impose of the frequencies in that area in much higher. After this we have tried to find the width of the echo by trial-and-error method. We have then generated spectrogram through sliding window only with the width of the echo from the ADC signal. We wanted to create a automated system to detect the first echo in a unknown ADC signal with CNN. But the actual time duration value is not accurate after prediction. In future, more R&D must be done to solve this problem. We are currently working to solve this issue.

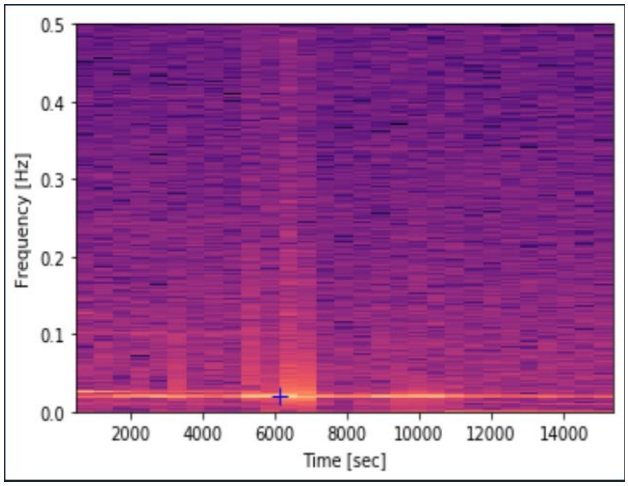

Fig. 10. First Reflection Detection in Spectrogram.

V. ACKNOWLEDGEMENT

We would like to thank Prof. Dr. Andreas Pech for Providing us an opportunity to work on this topic and also for providing me the proper guidance to write this paper.

VI. REFERENCES

[1] Stiawan, Roni; Kusumadjati, Adhi; Aminah, Nina Siti; Djamal, Mitra; Viridi, Sparisoma (2019). An Ultrasonic Sensor System for Vehicle Detection Application. Journal of Physics: Conference Series, 1204(), 012017–. doi:10.1088/1742-6596/1204/1/012017.

[2] Cheeke, J. D. N. (2012). Fundamentals and applications of ultrasonic waves. CRC press.

[3] Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning Nature, 521(7553), 436-444. doi:10.1038/nature14539.

[4] Yu, D., & Deng, L. (2010). Deep learning and its applications to signal and information processing [exploratory dsp]. IEEE Signal Processing Magazine, 28(1), 145-154.

[5] Bock, Sebastian, Josef Goppold, and Martin Weiß. "An improvement of the convergence proof of the ADAM-Optimizer." arXiv preprint arXiv:1804.10587 (2018).

[6] Ruby, Usha, and Vamsidhar Yendapalli. "Binary cross entropy with deep learning technique for image classification." Int. J. Adv. Trends Comput. Sci. Eng 9.10 (2020).