ENGR 498-07 Research in Artificial Intelligence and Deep Learning

SMART INFANT MONITORING SYSTEM

TEAM MEMBERS:

DIPAN MUKHOPADHYAY

GURPREET SINGH

ABHISHEK RAJ SHEKHAR

Advisor: Dr. Jafar Saniie

Assistant Mentor: Xinrui Yu

SUMMER 2022

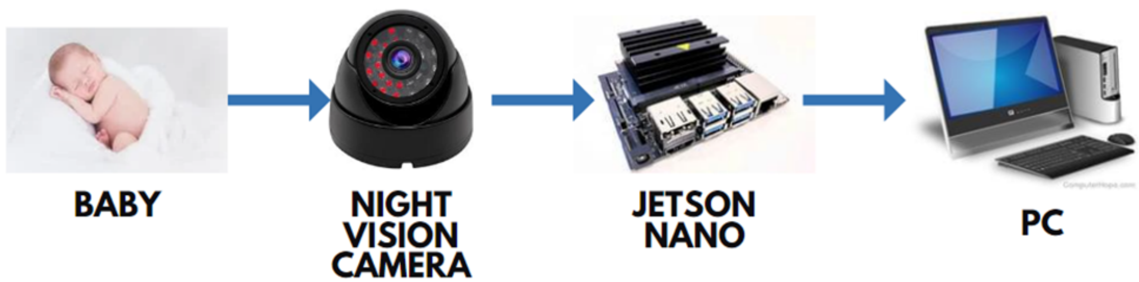
SMART INFANT MONITORING SYSTEM | FINAL REPORT

ABSTRACT

The goal of this project is to design a smart infant monitoring system that can help parents to have a watch on their baby and can notify them about the same. A Non-contact-based baby monitoring system using image Artificial Intelligence and Image processing is proposed in this project which is used for proper safety and monitoring the activity of babies by their busy parents. This is done by sending the video feed to the caregiver's smartphone. The system detects the unwanted positions in which the baby is in. The following would be monitored here: (a) whether the baby's face was covered due to sleeping on the stomach; (b) whether the baby threw off the blanket from the body; (c) whether the baby was moving frequently; (d) whether the baby's eyes were opened due to awakening.(e) baby crying detection is also proposed to detect its crying patterns. Thus the caregivers were not required to monitor their babies after regular intervals. They were given an alert message when their presence was required. The device was developed using NVIDIA's Jetson Nano microcontroller. A night vision camera(USB system) with distinctive features and secured Wi-Fi connectivity was interfaced. For this we have used Python as a programming language and OpenCV as an Image Processing Library.

INTRODUCTION

Smart baby monitoring devices are being used to obtain and send video and audio data of the baby to the caregiver’s smartphone, but most of these devices are unable to recognize or understand the data. In this project, a novel baby monitoring device is developed which automatically recognizes the undesired and harmful postures of the baby by image processing and sends an alert to the caregiver’s smartphone. Deep learning-based object detection algorithms are implemented in the hardware, and a smartphone app is developed. The overall system is shown in Figure below -



This shows the flow used here

The objectives that have been mentioned in the introduction are based on deep learning models that are to be run in embedded systems where resources such as memory and speed are limited. The methods must work in an embedded system with low latency. The system should also work in both day and night conditions. The detection methods should not be biased and be inclusive to all races of babies. The objective of this study is to develop a baby monitoring device that will automatically recognize the harmful postures of the baby by image processing and send an alert to the caregiver’s smartphone. The work will be considered successful when the proposed baby monitor can automatically detect the targeted harmful postures and send a notification to the smartphone. Experiments with different postures will be conducted and the latency of the detection algorithms will be measured.

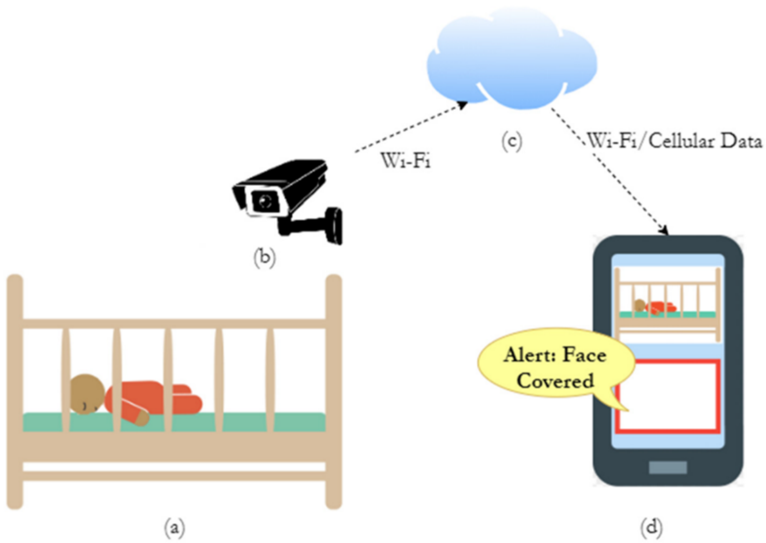


Figure depicts the overall smart baby monitoring system: (a) baby sleeping; (b) smart baby monitoring device automatically detects harmful postures such as face covered, thrown off the blanket, frequently moving or awake; (c) image data are sent to the smartphone through the Internet with the message of any harmful situation; (d) a real-time video of the baby is shown in the smartphone, and notifications and alerts are generated whenever the caregiver receives a message of the harmful situation.

The needs and significances of the proposed system are mentioned below:

- About 1300 babies died due to sudden infant death syndrome (SIDS), about 1300 deaths were due to unknown causes, and about 800 deaths were caused by accidental suffocation and strangulation in bed in 2018 in the USA . Babies are at higher risk for SIDS if they sleep on their stomachs as it causes them to breathe less air. The best and only position for a baby to sleep is on the back—which the American Academy of Pediatrics recommends through the baby's first year . Sleeping on the back improves airflow. To reduce the risk of SIDS, the baby's face should be uncovered, and body temperature should be appropriate . The proposed baby monitor will automatically detect these harmful postures of the baby and notify the caregiver. This will help to reduce SIDS.
- Babies may wake up in the middle of the night due to hunger, pain, or just to play with the parent. There is an increasing call in the medical community to pay attention to parents when they say their babies do not sleep . The smart baby monitor detects whether the baby's eyes are open and sends an alert. Thus, it helps the parents know when the baby is awake even if he/she is not crying.
- When a baby sleeps in a different room, the caregivers need to check the sleeping condition of the baby after a regular interval. Parents lose an average of six months' sleep during the first 24 months of their child's life. Approximately

10% of parents manage to get only 2.5 h of continuous sleep each night. Over 60% of parents with babies aged less than 24 months get no more than 3.25 h of sleep each night. A lack of sleep can affect the quality of work and driving; create mental health problems, such as anxiety disorders and depression; and cause physical health problems, such as obesity, high blood pressure, diabetes, and heart disease . The proposed smart device will automatically detect the situations when the caregiver's attention is required and generate alerts. Thus, it will reduce the stress of checking the baby at regular intervals and help the caregiver to have better sleep.

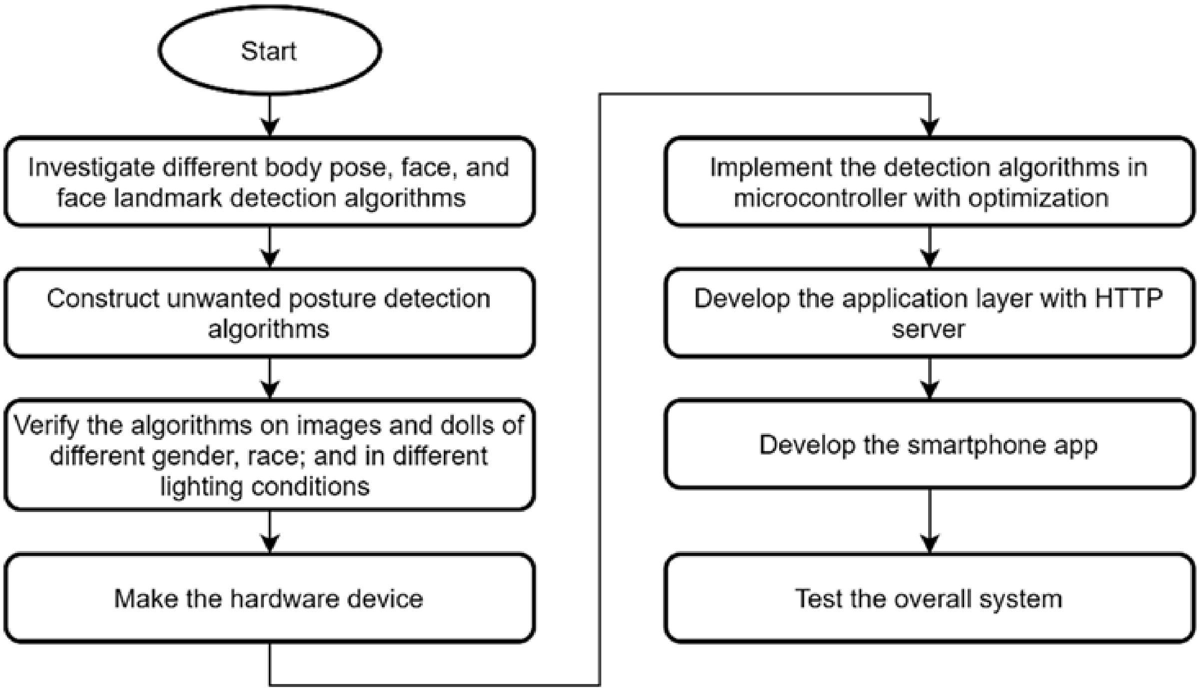
RELATED WORKS:

Several pieces of research have been conducted regarding this project by various companies who are using this project in an actual implementation with differing outcomes which are depicted in this following table as shown below:

Work	Motorola [9]	Infant Optics [10]	Nanit [11]	Lollipop [12]	Cubo Ai [49]	Proposed
Live Video	Yes	Yes	Yes	Yes	Yes	Yes
Boundary Cross Detection	No	No	No	Yes	Yes	No
Cry detection	No	No	No	Yes	Yes	No
Breathing Monitoring	No	No	Yes	No	No	No
Face Covered Detection	No	No	No	No	Yes	Yes
Blanket Removed Detection	No	No	No	No	No	Yes
Frequent Moving Detection	No	No	Yes	No	No	Yes
Awake Detection from Eye	No	No	No	No	No	Yes

MATERIALS AND METHODS:

The steps taken to develop the detection algorithms of harmful and undesired sleeping postures from image data and prototype development of the smart baby monitor are briefly shown in Flowchart. They are described below:



TECHNICAL DESCRIPTION:

The offered system's performance was tested on two different types of hardware i.e., on the laptop and on NVIDIA Jetson Nano, which provides Maxwell 128 core GPU, capable of running multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech recognition.

Nvidia Jetson Nano & attached components -

(a) The whole setup

(b) Jetson Nano

Hardware: The single-board computer-NVIDIA® Jetson NanoTM is used as the main processing unit. It is a small size and low-power embedded platform where neural network models can run efficiently for applications such as image classification, object detection, segmentation, etc. It contains a Quad-core ARM A57 microprocessor running at 1.43 GHz, 4 GB of RAM, a 128-core Maxwell graphics processing unit (GPU), a micro SD card slot, USB ports, and other built-in hardware peripherals. A night-vision camera is interfaced with the Jetson Nano using a USB. When the surrounding light is enough, such as in the daytime, it captures color images. This camera has a built-in light sensor and infrared (IR) LEDs. When the surrounding light is low, the IR LEDs automatically turn on and it captures grayscale images. To connect with the Internet wirelessly, a Wi-Fi adaptor is connected to the USB port of the Jetson Nano. A 110V AC to 5V 4A DC adapter is used as the power supply.

Software: Linux4Tegra (L4T)—a version of Ubuntu operating system (OS)—was installed on a SD card (32 GB) of the Jetson Nano board. All the required packages like opencv and basic models were also installed in our Jetson Nano.For pose estimation initially the mediapipe library was used. However, after making some efforts it was unable to complete our objectives. Then we made a shift to Nvidia's pretrained model for pose estimation called TRTPose (TensorRT Pose Estimation). One of the libraries used in our project is Dlib, which is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real-world problems. It is used in industries and academia in various domains like robotics, embedded devices, mobile phones, and large high- performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge. Although it

is implemented in C++ it can be used with python as well as python is lucid and it is very much useful for developing complex scientific and numeric applications. It supports many operating systems and its large and robust standard library makes python score over other programming languages. Therefore, the sheer code of the proposed system is connected in python.We have used PyCharm IDE as our working environment as well.

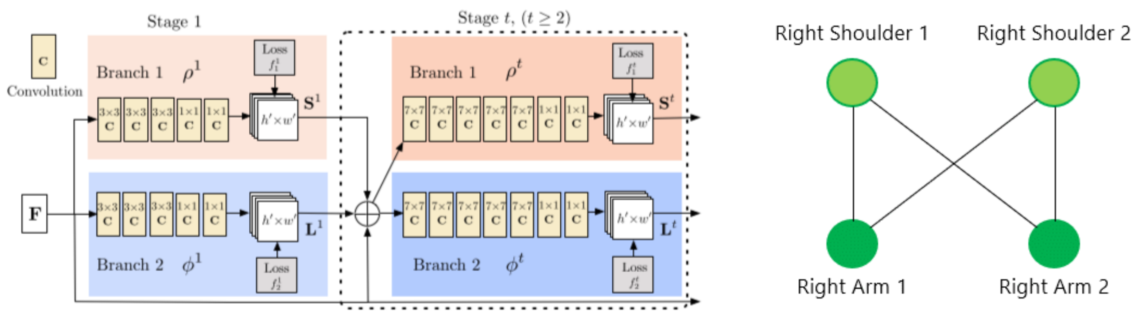
POSE ESTIMATION :

Human pose estimation is the computer vision task of estimating the configuration ('the pose') of the human body by localizing certain key points on a body within a video or a photo.This localization can be used to predict if a person is standing, sitting, lying down, or doing some activity like dancing or jumping.

TensorRT Pose Estimation is used for our use case. This project features multi-instance pose estimation accelerated by NVIDIA TensorRT. It is ideal for applications where low latency is necessary. There are two approaches to building a pose estimation model. A top-down approach places bounding boxes around all humans detected in a frame, and then their respective body parts are localized within that bounding box. A bottom-up approach does the opposite. You would first detect all human body parts within a frame and then group parts that belong to a specific person after the fact.

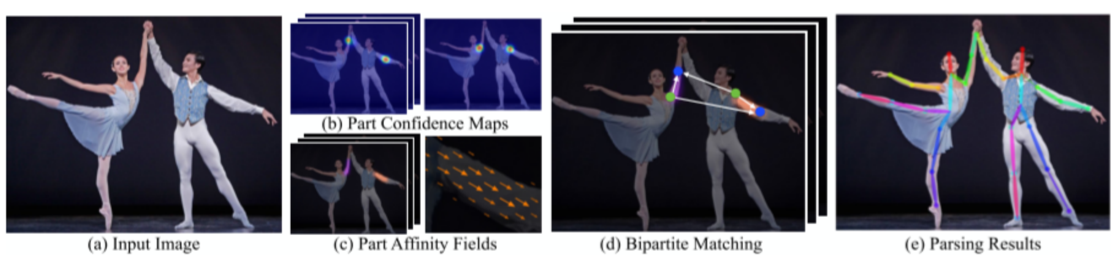
TRTPose takes the bottom-up approach towards pose estimation. The model first detects key points for every body part present in a frame, and then figures out which parts belong to which individual within that frame. The method takes the entire image as the input for a two-branch CNN to jointly predict confidence maps for body part detection, and part affinity fields for parts association. Each PAF has a component in the x direction as well as the y direction, thus representing a vector.Then the parsing step

performs a set of bipartite matchings to associate body parts candidates. Finally, we assemble them into full body poses for all people in the image.



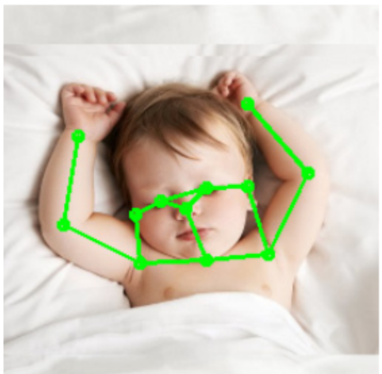
(a) (b)

(a) Architecture of the two-branch multistage CNN. Each stage in the first branch predicts confidence maps S t , and each stage in the second branch predicts PAFs L t . After each stage, the predictions from the two branches, along with the image features, are concatenated for the next stage.
(b) Sample subgraph depicting possible relations between detected keypoints.

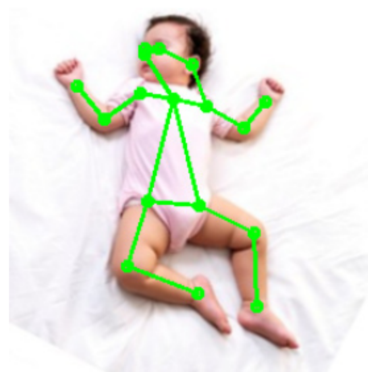


To achieve our objective of this project here, the nose of the baby is detected from the image to decide whether the face is covered due to sleeping on the stomach or for other reasons. To detect a blanket removed, the visibility of the lower body parts such as the hip, knee, and ankle are detected. This is done by the TRT pose estimation model by Nvidia. The features are fed into two-branch multi-stage transposed convolution networks. These branch networks then simultaneously predict the heatmap and Part Affinity Field (PAF) matrices. Finally, a collection of human sets is found, where each human is a set of parts, and where each part contains its relative information. And after the detection and running our algorithms on it, we can now get the alert of the situation whenever the baby is in a vulnerable state. Whether the baby having its blanket removed or the baby sleeping on its stomach, the caretaker can now get notified of it

without monitoring the infant 24x7. Thus, the baby remains in a safe state preventing any accidents from happening.

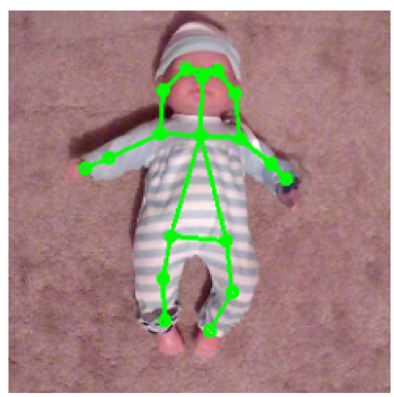


(a)



(b)

(c)



(d)

(a)Baby with blanket on (b)Baby with blanket off (c)Baby sleeping on the back, nose not detected (d)Baby sleeping pose estimated.

AWAKE DETECTION :

The suggested method for detecting whether the baby is awake or not operates on two levels. The procedure begins with the camera recording live video frames, which are

then transferred to a local server. The Dlib library is utilized on the server to identify facial landmarks, and a threshold value is used to determine whether or not the baby is awake. The EAR (Eye Aspect Ratio) is then computed using these face landmarks and given to the driver. The EAR value obtained at the application's end is compared to a threshold value of 0.20 in our system. The baby is regarded to be awake if the EAR value is greater than the threshold value. An alert message will be sent to the parent notifying them that the baby is awake and needs assistance.
For awake detection we have used OpenCV and Python. The Dlib library is used to detect and isolate the facial landmarks using Dlib pre- trained facial landmark detectors. In this approach , 68 facial landmarks have been used.

Facial Landmark marking -
Dlib library is imported and used for the extraction of facial landmarks. Adlib uses a pre-trained face detector , that is an improvement of the histogram of oriented gradients. It consists of two shape predictor models trained on the i-Bug 300-W dataset ,that each localize 68 and 5 landmark points respectively within a face image .We have used the 68 facial landmarks.
In this method, frequencies of gradient direction of an image in localized regions are used to form histograms. It is especially suitable for face detection; it can describe contour and edge features exceptionally in various objects.
For recording the Facial Landmarks, the Facial Landmark Predictor was used by the system to calculate lengths for the EAR values.

The Facial landmark points of the Dlib library which are used to compute EAR are represented in the figure below :

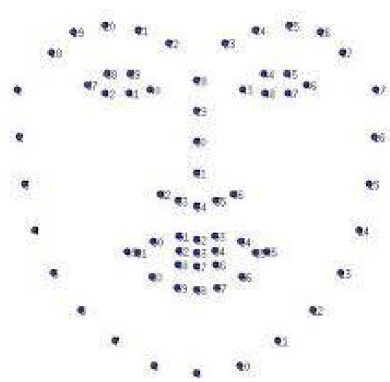


Fig (a) : Facial Landmarks

Algorithm used -

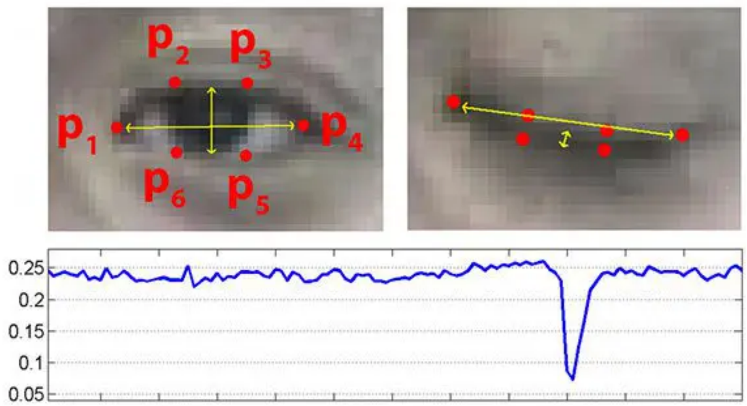


Fig (b) : Eye Aspect Ratio

Here P1,P2,P3,P4,P5,P6 are the pupil coordinates.EAR is generally a constant when eyes are open and is near about 0.5- 0.10.When EAR is greater than 0.20. It is concluded that the baby is awake.

Eye Aspect Ratio(EAR) is the calculated for both the eyes,

$$\frac{(|P2 - P6| + |P3 - P5|)}{2(|P1 - P4|)}$$

The numerator determines the distance between the upper and lower eyelids using equation. The horizontal distance of the eye is represented by the denominator. EAR values are utilized to identify whether the baby is awake or asleep in this framework. The average of the EAR values of the left and right eyes is obtained. The EAR is monitored using a counter that sees whether the value falls exceeds the threshold value in the following frame . The infant has closed its eyes and is sleepy,as indicated by the aforementioned circumstances.In contrast, if the EAR values does not maintains the higher value for a following few frames, it means that the baby went back to sleep after a few seconds.The block design of our suggested technique to detect whether the baby is awake or asleep is shown in Figure 3 (Block Diagram).

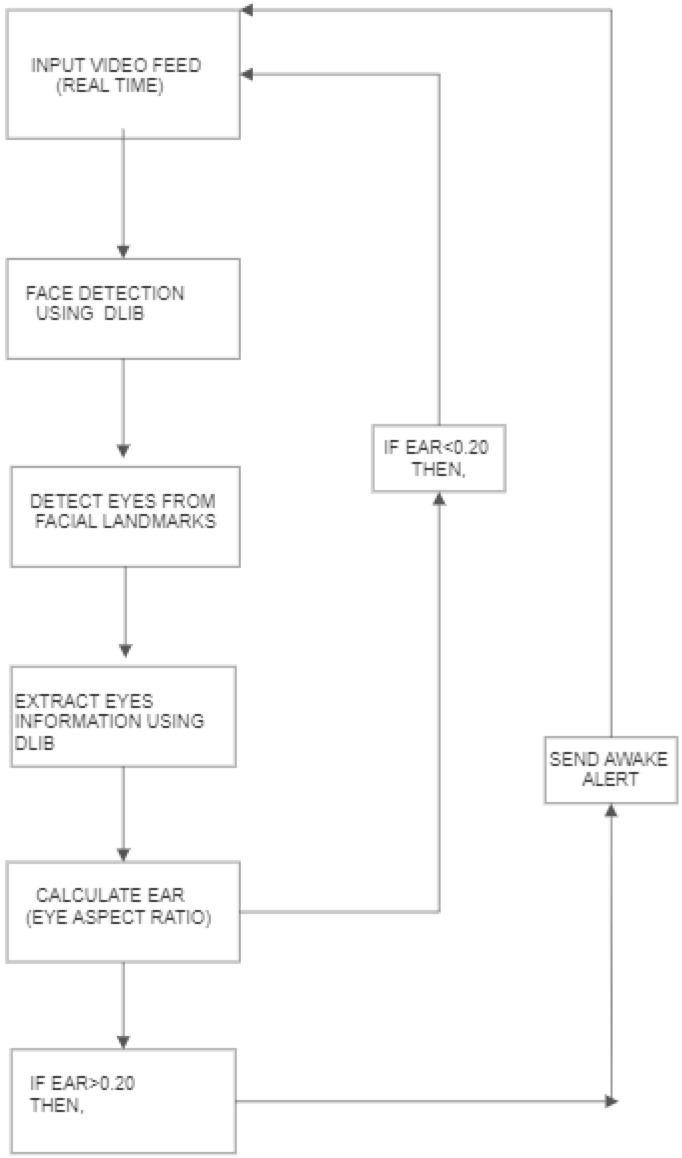


Fig (c) : Block Diagram - Awake Detection

For this project we have utilized a webcam associated with the Jetson Nano .The framework was tried for different samples in various light surrounding lighting conditions.

Eye Detection Accuracy =

$$\frac{\text{Total no. of times eye detected}}{\text{(Total no. of time eye detected + total no. of times eye not detected)}}$$

Awake Detection accuracy =

$$\frac{\text{Total no. of times alert message was displayed}}{\text{(Total no. of times alert message was displayed +Total no. of times alert messages was not sent)}}$$

Result Table for accuracy

Serial No:	Eye Detection accuracy	Awake Detection accuracy
Sample 1	82.34%	80.53%
Sample 2	81%	64.5%
Sample 3	73%	82.34%
Sample 4	77%	65.67%
Sample 5	89.5%	100%
Total	80.56%	78.60%

After training and testing the model rigorously we attained an accuracy of about 78.60%.

Obtained results -

Fig (a) Sample Image

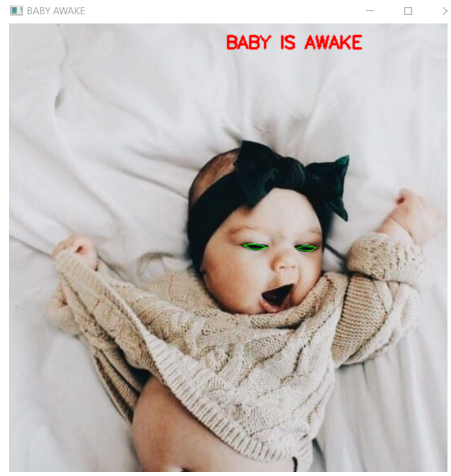


Fig (b) Output Image

We also faced certain limitation mentioned as follows -

- The OpenCV can detect live images up to only an approximate distance up to 25 cm from the Webcam to face. So, the placement of the webcam should be such that this limitation does not affect the model accuracy.
- With poor lighting conditions occasionally, the system is unfit to perceive the eyes. So, it gives a wrong result which must be managed.
- If the orientation of the face is tilted to a specific degree it will in general be perceived , anyway past this framework the framework is unable to identify the face of the baby.

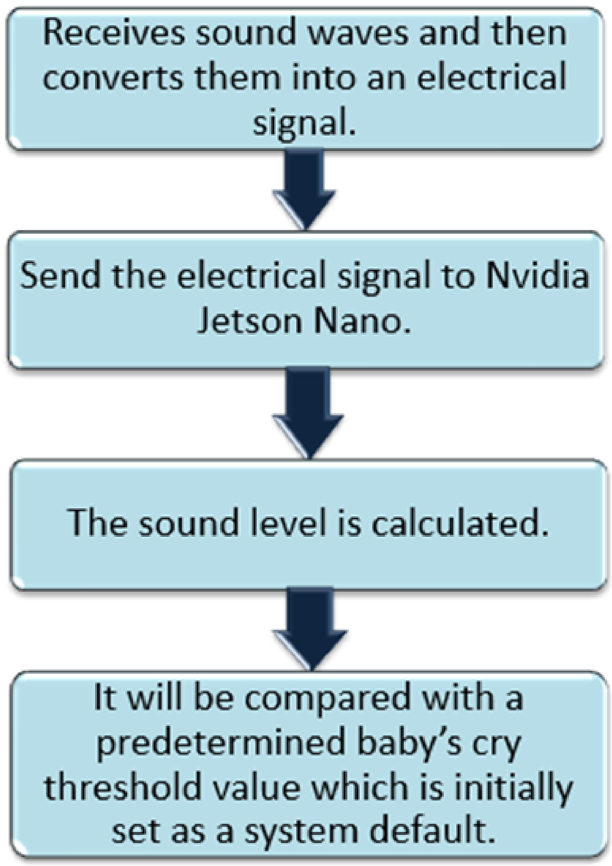
CRYING DETECTION:

An objective, automated and unobtrusive method to quantify crying behavior in an at-home and clinical setting may improve the diagnostic process in excessively crying infants, allow for objective determination of treatment effects by physicians, and enable researchers to include objectively determined cry duration as digital biomarker in clinical trials. Given the importance for researchers to study the relationship between an infant's crying patterns and their health, automatic detection and quantification of infant cries from an audio signal is an essential step in remote baby monitoring applications .

Automatic cry detection has been reported in the form of remote baby monitors for non-intrusive clinical assessments of infants in hospital settings, and several researchers have shown that classification of cry- and non-cry-sounds is possible with machine-learning algorithms.

The aim is to automatically recognize a baby crying while sleeping. In such a case, a soothing sound is played to calm the baby down. This is done by implementing a machine learning algorithm on Jetson Nano.The idea is to train a model on a computer and to deploy it on Jetson Nano,which is used to record a signal and use the model to predict if it is a baby cry or not. In the former case a soothing sound,the caretaker is also notified about the same, in the latter the process(recording and predicting steps) starts again.

A review of infant cry analysis and classification:



Training of the model -

It includes all the steps required to train a machine learning model. First, it reads the data, it performs feature engineering and it trains the model. The model is saved to be used in the prediction step. The _training step_ is performed on a powerful machine, such as our personal computer.

Simulation of the model -

There is a script to test the prediction step on our computer before deployment on Jetson Nano.

The trained model was then deployed from my personal computer to Jetson Nano using FlieClient@Zilla having a suitable Wi-Fi connection and IP address of the Jetson Nano. The trained model is named as “model.pkl”.

After the deployment, “model.pkl” is opened on Jetson Nano where the prediction of the model is done.

Prediction -

It includes all the steps needed to make a prediction on a new signal.It reads a new signal, it divides it into 5 overlapping signals, it applies the pipeline saved from the training step to make the prediction.The prediction is performed on Jetson Nano.

FUTURE WORKS

- Our first focus is to develop a mobile application so that we can directly interact with caretakers and notify them whenever the baby needs their attention.
- To improve our pose estimation model and revise our algorithm for obtaining better results.
- To try other cry detection methods for better results.
- We also aim on performing frequent moving detection
- Use sentiment analysis by using machine learning models trained with baby’s expression for example: if he is making a particularly suckling expression in sleep which would mean he is hungry so notification of hunger would be sent to parents. This will be done through audio and video capturing.

REFERENCES

- https://github.com/NVIDIA-AI-IOT/deepstream_pose_estimation
- https://developer.nvidia.com/blog/creating-a-human-pose-estimation-application-with-deepstream-sdk/
- https://github.com/NVIDIA-AI-IOT/trt_pose

- Tareq Khan, Article An Intelligent Baby Monitor with Automatic Sleeping Posture Detection and Notification, School of Engineering, Eastern Michigan University, Ypsilanti, MI 48197, USA; tareq.khan@emich.edu
- https://pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/
- Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh, Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, The Robotics Institute, Carnegie Mellon University
- https://www.researchgate.net/publication/336989127_Baby_Cry_Detection_Deep_Learning_and_Classical_Approaches
- https://www.frontiersin.org/articles/10.3389/fped.2021.651356/full
- http://dlib.net/ml.html
- Jose Gomes da Silva Neto, Joao Marcelo Xavier Natario Teixeira, Veronica Teichrieb, Analyzing embedded pose estimation solutions for human behavior understanding

UNCOVERING THE BUILDING BLOCKS OF TURBULENCE & MODELING UNSTEADY AERODYNAMIC SYSTEM

FAHIM YOUSUF CHOUDHURY, KOUSTAV MONDAL, ANIMESH ALOK PRABHAKAR, SHUBHANKAR SINHA & KESHAW THAKUR

MENTOR: SCOTT DAWSON

FINAL GROUP REPORT

Introduction

The main goal of this research work is to minimize the computational complexity and cost that goes into representing various fluid mechanics and aerodynamic systems in a low dimensional form in order to understand and analyse them better.

Methodology

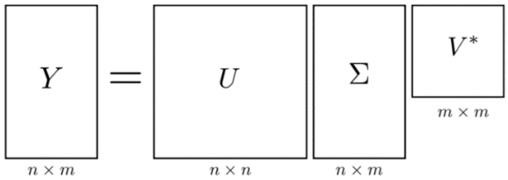
The datasets that we are dealing with are mostly functions of both space and time. In order to understand pertinent features of such datasets as well as to fulfil our goal of building a reduced-complexity model, our first step is to separate the variables using Singular Value Decomposition (SVD) and then using various curve fitting tools in MATLAB and PYTHON we create an analytical function that fits our dataset.

The decomposition methods presented in this report are founded on the singular value decompositions of matrices or operators.

Singular Value Decomposition

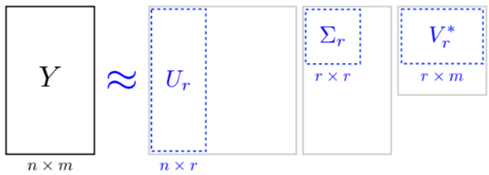
$$y(x,t) = \sum_{j=1}^{m} u_j(x) a_j(t)$$

In order to achieve this decomposition, we use SVD. The main idea of this is to break the matrix containing spatial and time coordinates into two separate vectors i.e.,



U and ***V*** are unitary matrices and $\sum$ is diagonal matrix with decreasing and nonnegative diagonal entries.

We truncate the above matrices in order to eliminate noise and we get the following,



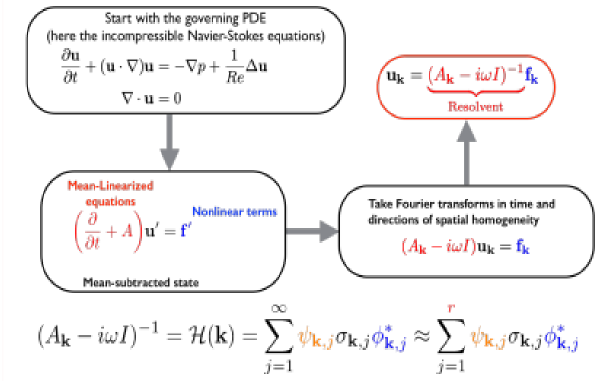
We then solve this expression and obtain

$$\begin{aligned} Y_r &= U_r \Sigma_r V_r^* \\ &= \sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^* \end{aligned}$$

Here uj and vj are vectors that are the j-th columns of Ur and Vr.

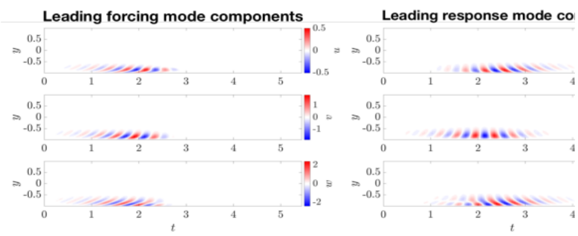
We use predicted coherent structures from a method called resolvent analysis, which looks at a singular value decomposition of a linear operator obtained from the governing Navier-Stokes equations

FINAL GROUP REPORT



The leading left and right singular vectors give the optimal response and forcing modes of the system.

Sample space-time modes in turbulent channel flow



Curve Fitting

The absolute and real components of u_j *and* v_j matrices are fitted using gaussian functions with the help of **curve_fit** function in Python and MATLAB.

The general form of the gaussian function assumed for fitting the components is,

$$\psi(y,t) = c\exp\left[ia_y y - b_y(y-y_0)^2\right]\exp\left[ia_t t - b_t(t -\right.$$

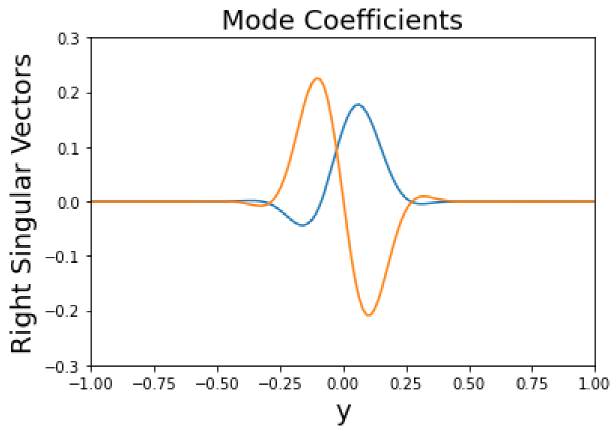
The unknown parameters of these functions are determined using curve fitting tools. For a better fit we enforce additional boundary conditions by adding additional image wavepackets.

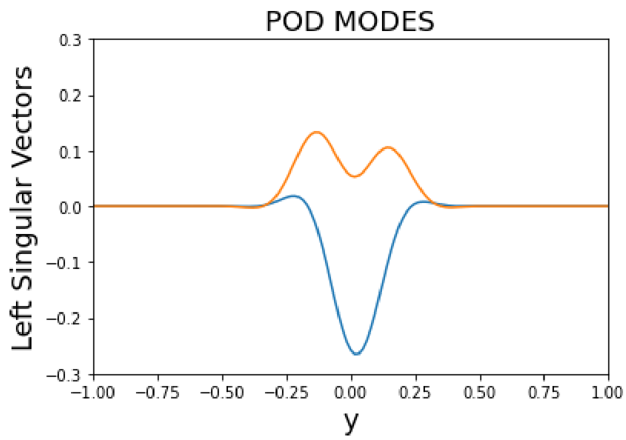
Results

Using the method discussed in the previous section, we have succeeded in creating analytical functions for various datasets. The results are shown and discussed below –

For TestMatrix –

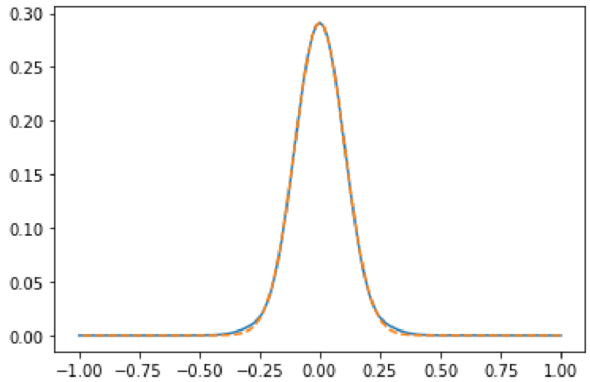
For our first dataset, we extracted the left and right singular vectors using built-in SVD tools in Matlab and python.



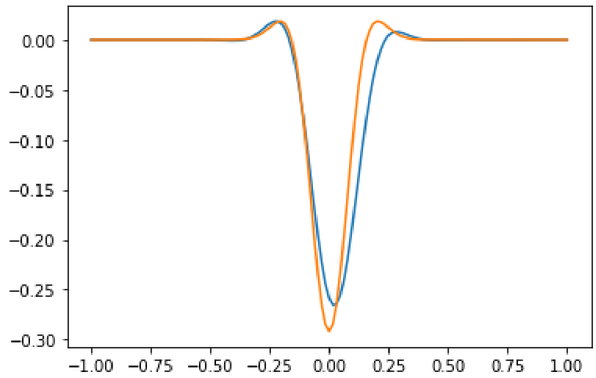


Now since we have separated space and time variables, we are in a position to fit them. Left Singular Vector is a function of space while Right Singular Vector is a function of time. For our first dataset, we fitted the absolute value of the above

vectors using one gaussian and the results were-

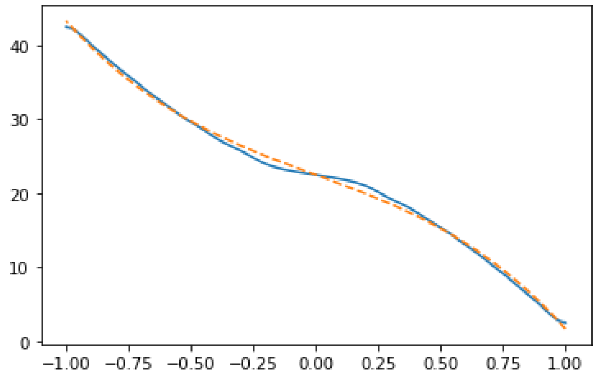


Fit of Left Singular Vector



Fit of Right Singular Vector

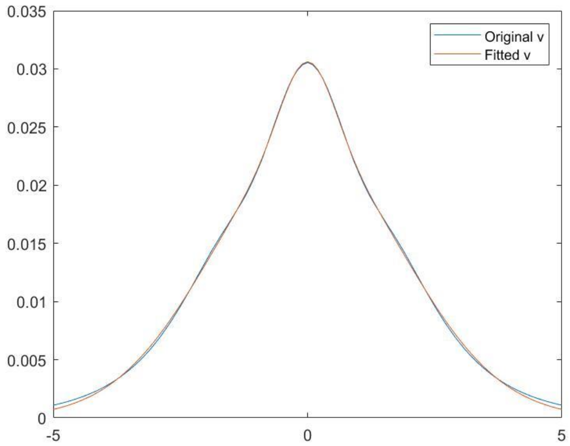
We also looked at the phase of the Left Singular Vector and fitted it.



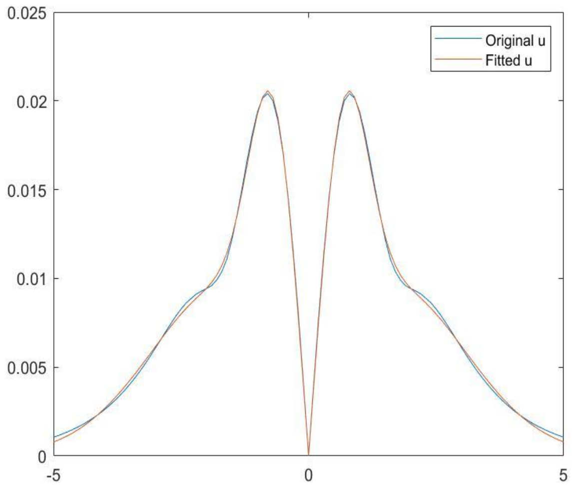
Now that we also have the phase, we can fit the real part of the vectors also. This framework has been used by us for the other datasets also.

For Flow over a Cylinder Data –

Here, we are going over some real world data and using the same analysis as before -

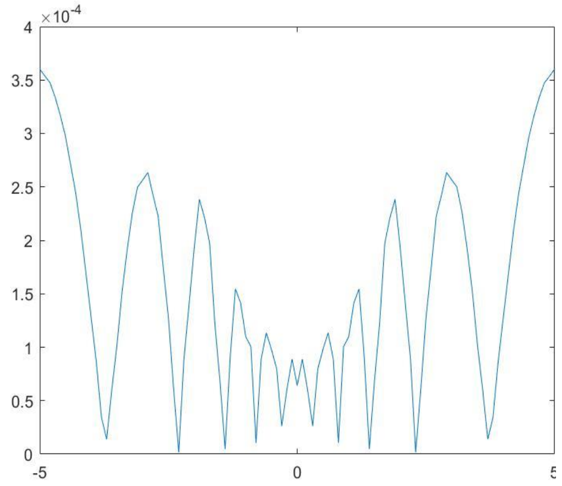


This fit was achieved using two gaussians, one for the inner region and the other accounting for the outer.

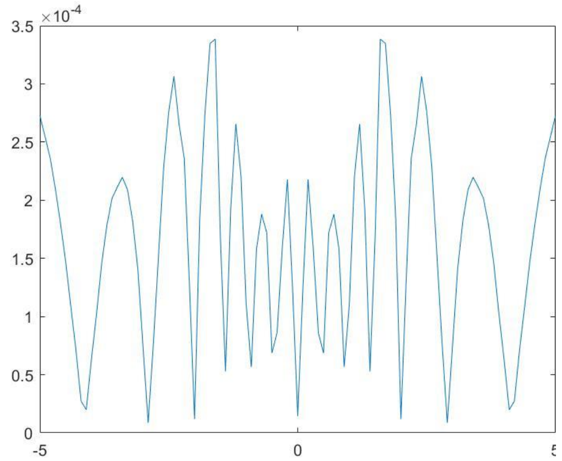


The above graph was obtained using 4 gaussians paired with a starting points guess as well as upper and lower bounds. These two graphs represent the u and v component of velocity of flow over a cylinder.

The error in both these cases is shown by the following graphs-



Error for v component of velocity

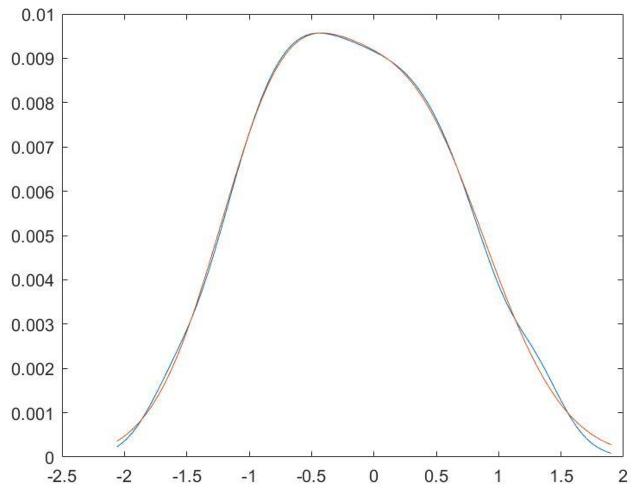


Error for u component of velocity

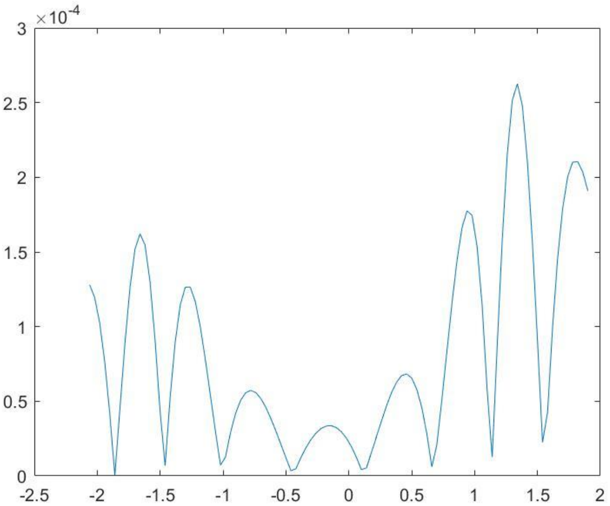
As we can clearly see, the error is negligible. That means our analytical function works and can be used for studying the dataset. Similarly, we were able to successfully create functions for other datasets also beginning with airfoil data.

For Flow over an Airfoil Data -

For this first plot, only 2 gaussians were required to achieve the necessary fit.



Velocity fitting graph



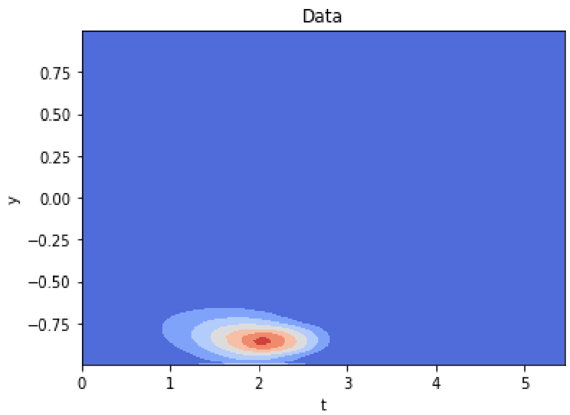
Error plot

As we can see, the error in this case is also negligible which is a good thing as our analysis has now worked on two sets of real world data. We will now look at some other data to test our theory even further.

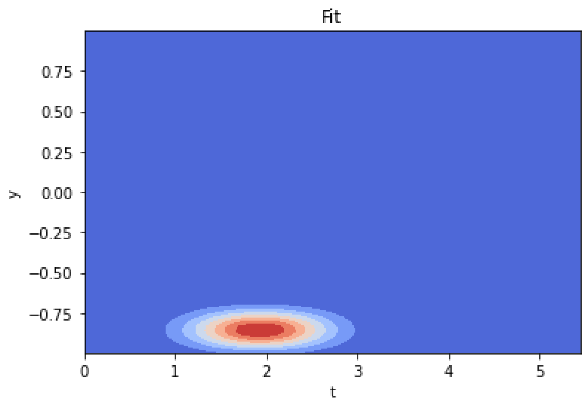
FINAL GROUP REPORT

For DataModes –

The below graph is a contour graph of the absolute value of PhiU. The second graph is obtained after finishing our fitting in time with one gaussian and two gaussian plus boundary conditions for space. As we can see, they match to a great extent.

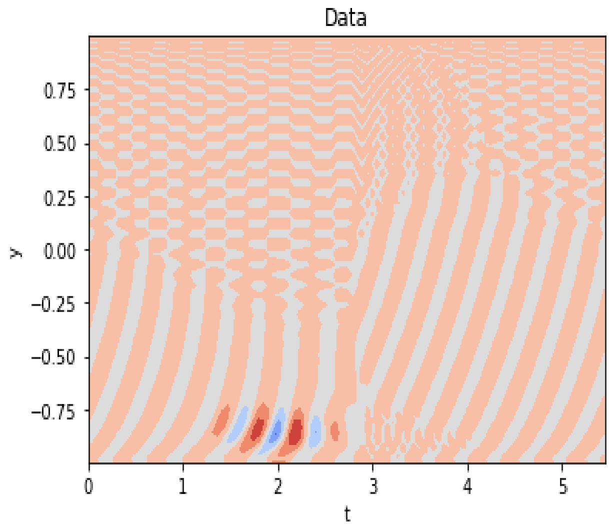


Original abs PhiU velocity

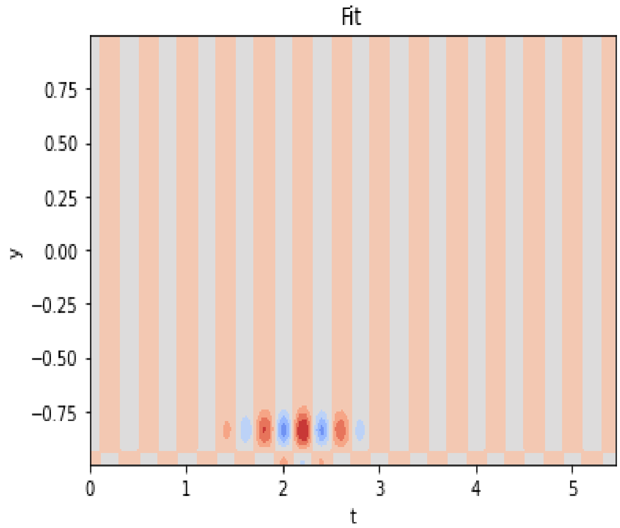


Fitted abs PhiU velocity

The next two graphs are of the real part of PhiU where we have extracted the phase and then fit a linear function to it using our fitting tools and combined it with our absolute valued function to get the fit of the real part. The graph matches the original in the x-direction but deviates a little in the y due to the phase.

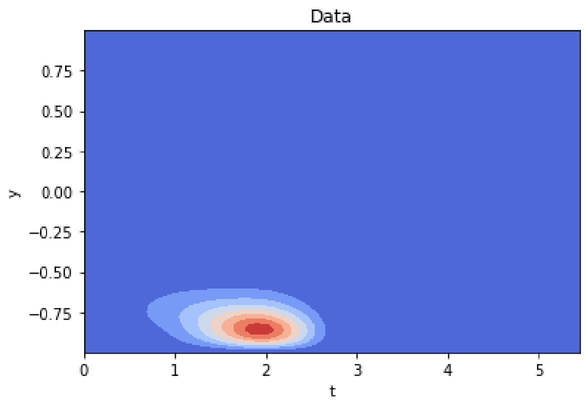


Original real PhiU velocity

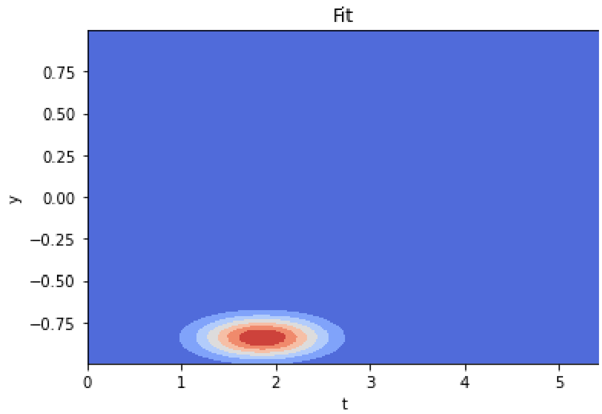


Fitted real PhiU velocity

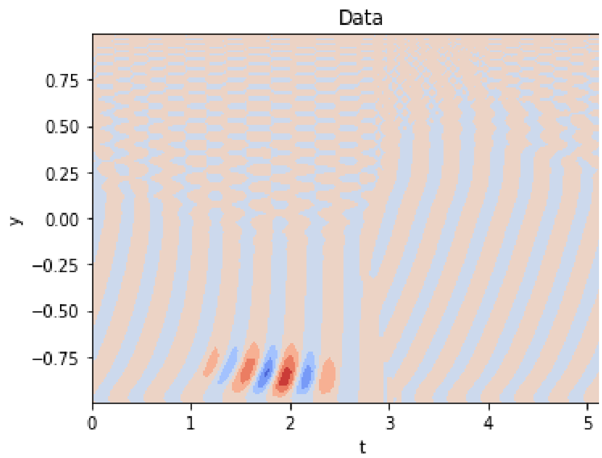
The same trend continues for PhiV, PsiU, PsiV and PsiW wherein we select that number of gaussian which reduces error and if necessary boundary conditions as shown by the following plots in the next few pages -



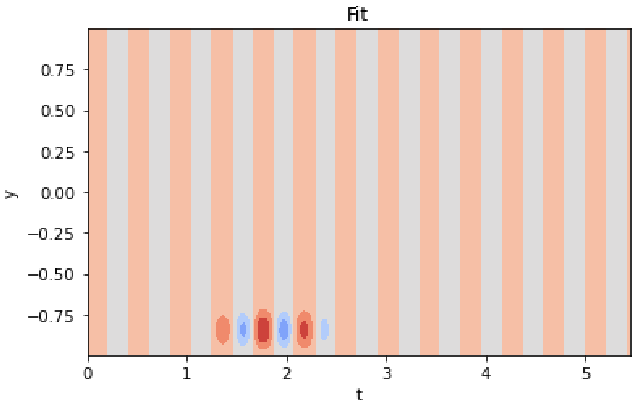
Original abs PhiV velocity



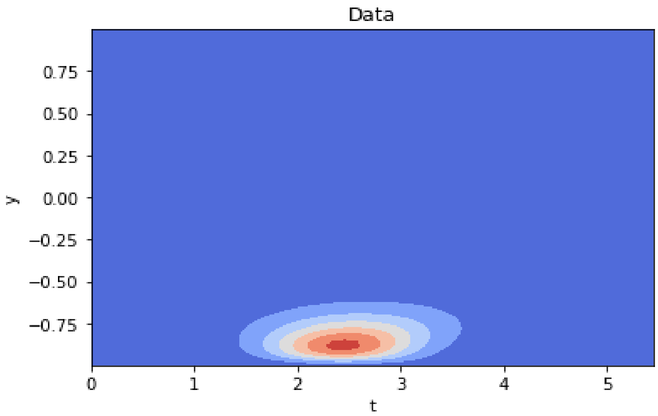
Fitted abs PhiV velocity



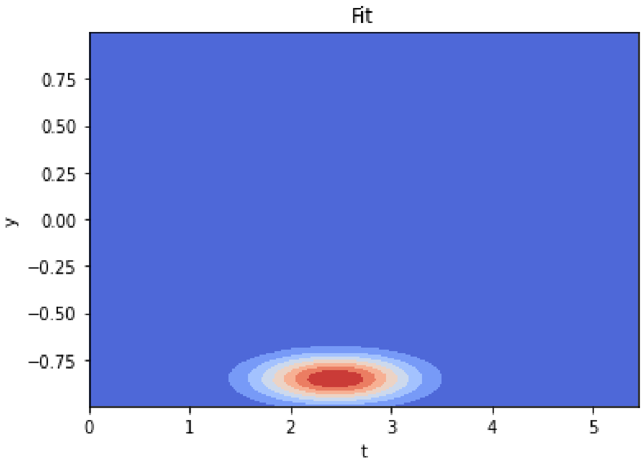
Original real PhiV velocity



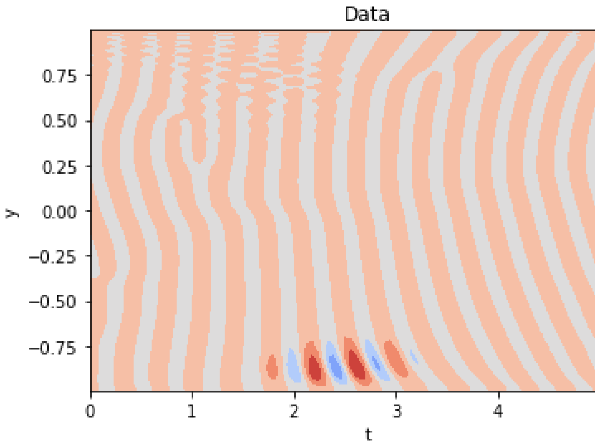
Fitted real PhiV velocity



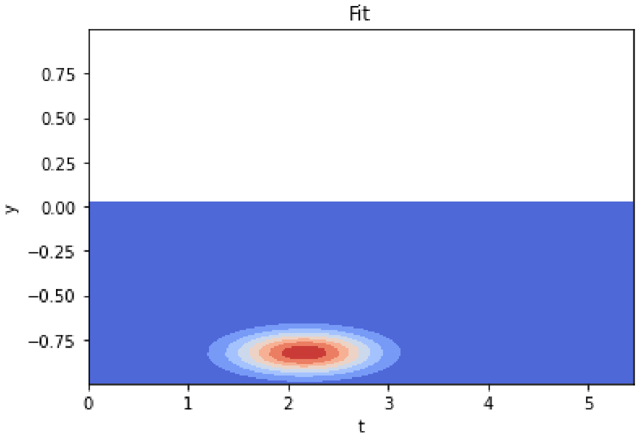
Original abs PsiU velocity



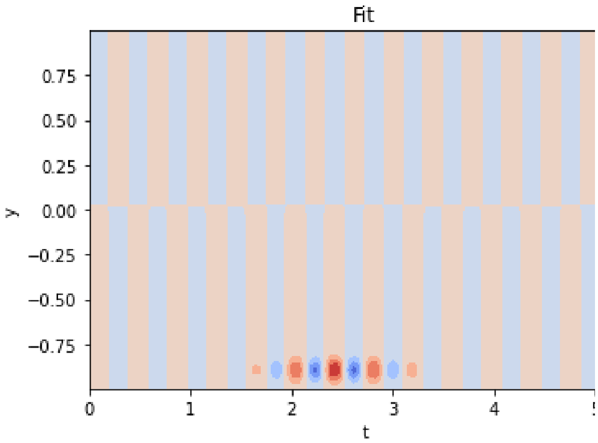
Fitted abs PsiU velocity



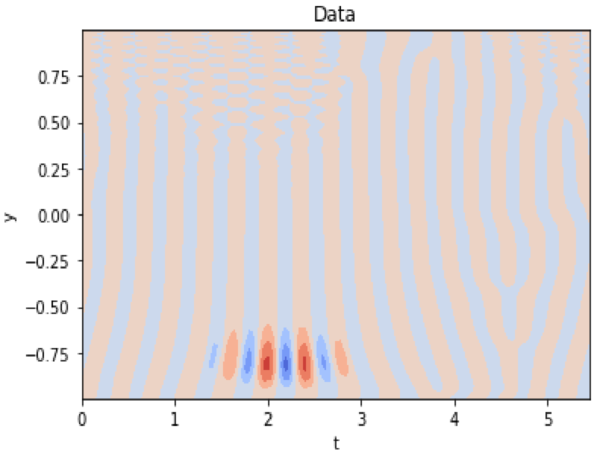
Original real PsiU velocity



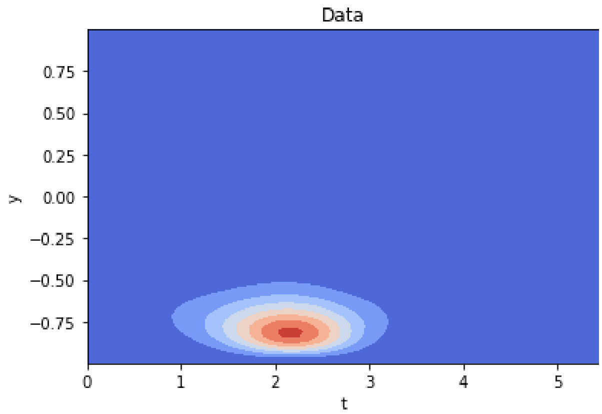
Fitted abs PsiV velocity



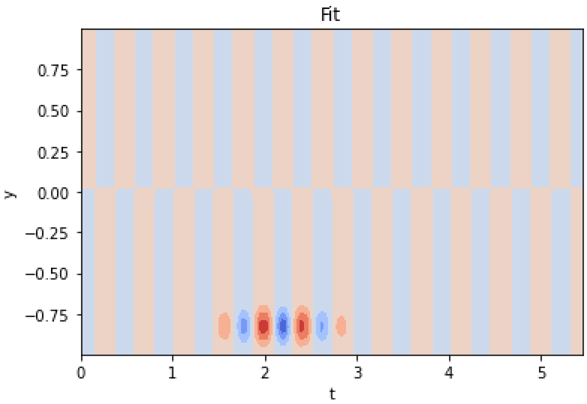
Fitted real PsiU velocity



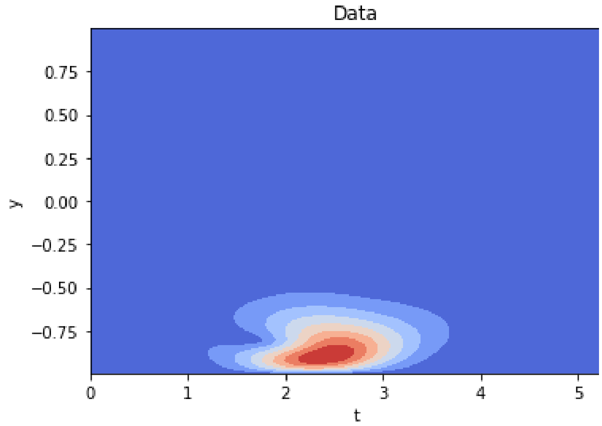
Original real PsiV velocity



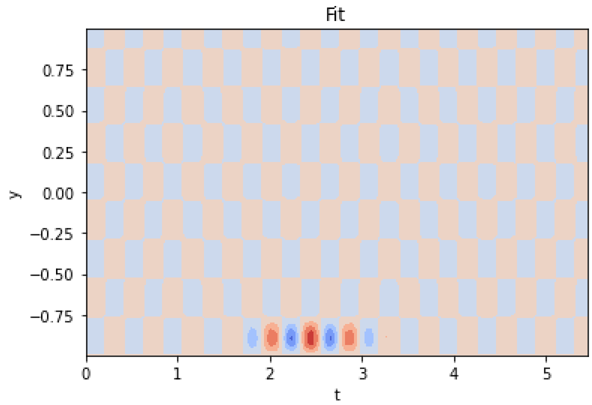
Original abs PsiV velocity



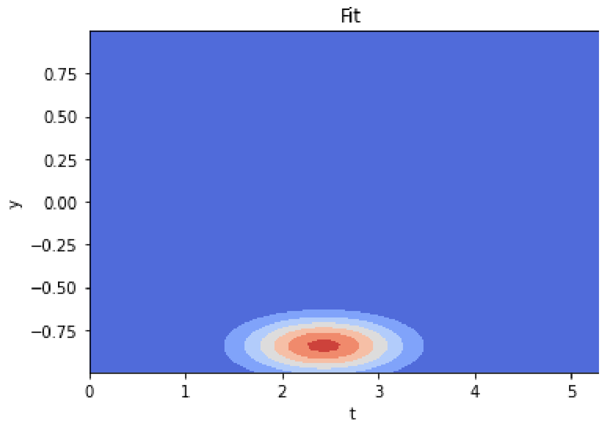
Fitted real PsiV velocity



Original abs PsiW velocity



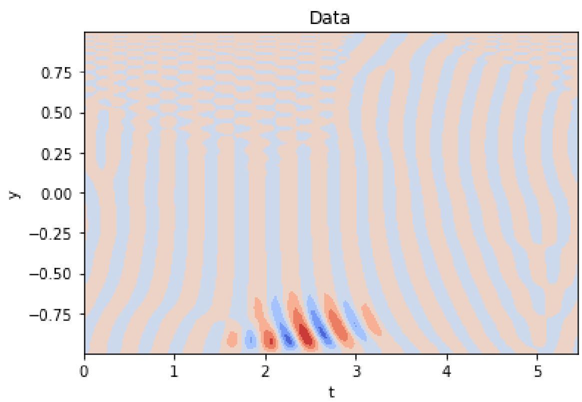
Fitted real PsiW velocity



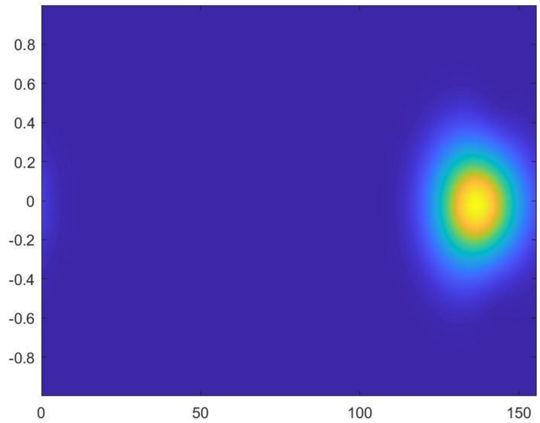
Fitted abs PsiW velocity

For DataModesV3-

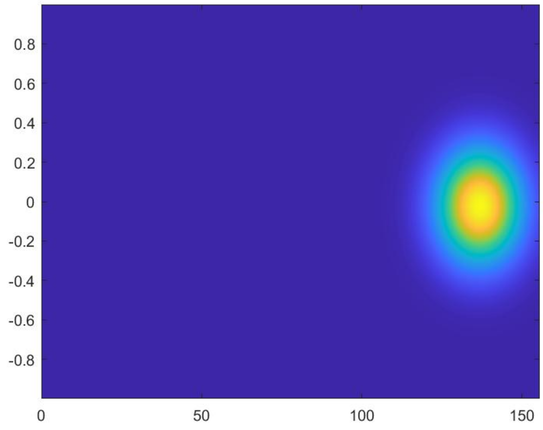
As evident by the name, DataModesV3 is an updated dataset of our previous DataModes. We have once again used our framework to create functions that represent the original data to a great extent as evident by the following plots-



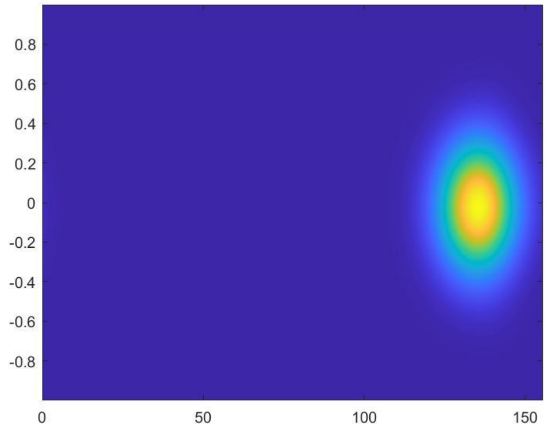
Original real PsiW velocity



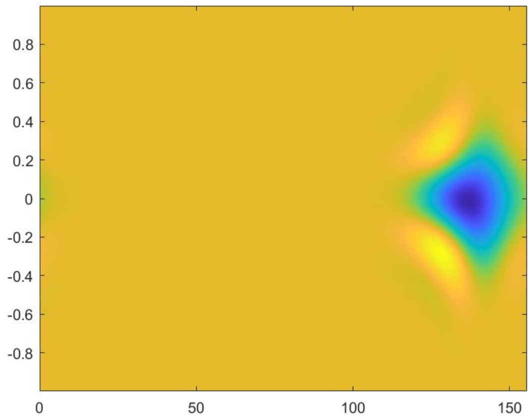
Original abs PhiU velocity



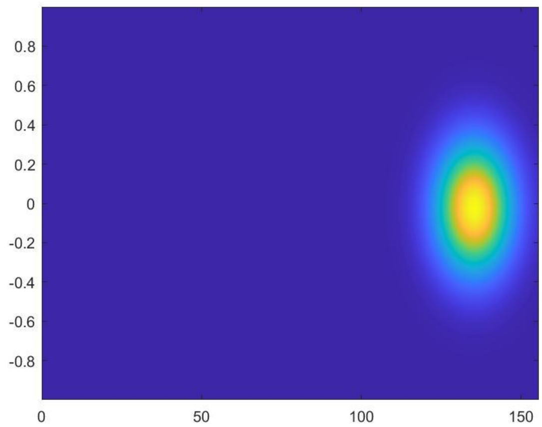
Fitted abs PhiU velocity



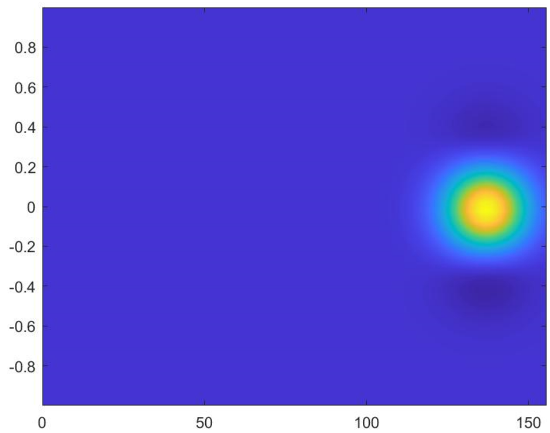
Original abs PhiV velocity



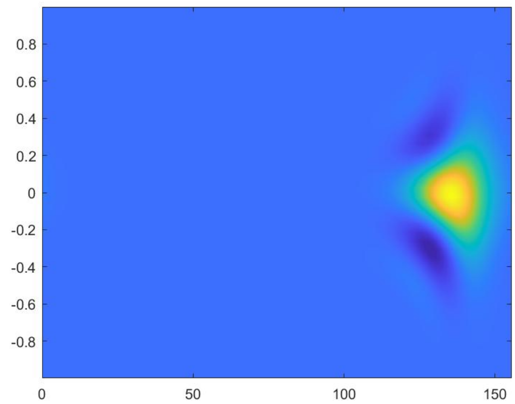
Original real PhiU velocity



Fitted abs PhiV velocity

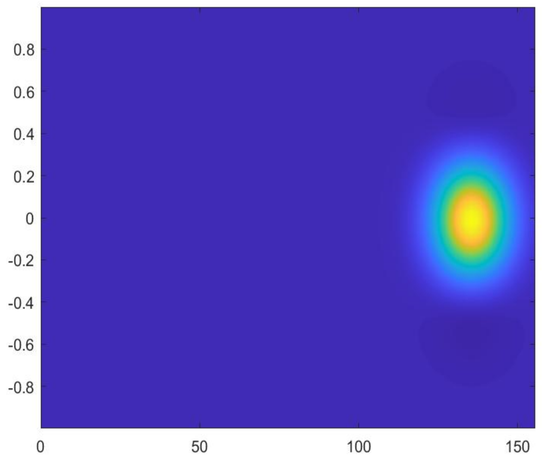


Fitted real PhiU velocity



Original real PhiV velocity

FINAL GROUP REPORT



Fitted real PhiV velocity

Conclusion –

After performing our analysis on a variety of data including both real world and theoretical and looking at the results that we have obtained, it is clear that our theory works and error is also minimal. But this theory does not provide us with a straight forward path as we have seen that in many cases, we had to select more than one gaussian sometimes even four and on top of that we also had to use boundary conditions and start point guesses sometimes in order to achieve a better fit. But our analysis also has its flaws as when it came to the real part of a function in DataModes, we were not able to get a great fit perhaps due to the phase not matching to a great extent or some other underlying issue. But apart from this, our error was less than 1% in flow over a cylinder, air-foil data, DataModesV3 and TestMatrix dataset.

Further Work –

In our work, we have done our analysis mostly on datasets that resemble a gaussian function in some form or the other. Further work may be done to expand our theory to cover a wide range of functions and eliminate some of the issues we encountered in our present work.

Acknowledgement –

We would like to thank you sir for giving us the opportunity to do this research which allowed us to hone our skills in Matlab and python and we also got to know about the basics of research work for which we are eternally grateful. We would also like to wish you good health and success in your future work.

ENGR 498-07 Research in Artificial Intelligence and Deep Learning

SMART INFANT MONITORING SYSTEM

TEAM MEMBERS:

DIPAN MUKHOPADHYAY

GURPREET SINGH

ABHISHEK RAJ SHEKHAR

Advisor: Dr. Jafar Saniie

Assistant Mentor: Xinrui Yu

SUMMER 2022

SMART INFANT MONITORING SYSTEM | FINAL REPORT

ABSTRACT

The goal of this project is to design a smart infant monitoring system that can help parents to have a watch on their baby and can notify them about the same. A Non-contact-based baby monitoring system using image Artificial Intelligence and Image processing is proposed in this project which is used for proper safety and monitoring the activity of babies by their busy parents. This is done by sending the video feed to the caregiver's smartphone. The system detects the unwanted positions in which the baby is in. The following would be monitored here: (a) whether the baby's face was covered due to sleeping on the stomach; (b) whether the baby threw off the blanket from the body; (c) whether the baby was moving frequently; (d) whether the baby's eyes were opened due to awakening.(e) baby crying detection is also proposed to detect its crying patterns. Thus the caregivers were not required to monitor their babies after regular intervals. They were given an alert message when their presence was required. The device was developed using NVIDIA's Jetson Nano microcontroller. A night vision camera(USB system) with distinctive features and secured Wi-Fi connectivity was interfaced. For this we have used Python as a programming language and OpenCV as an Image Processing Library.

INTRODUCTION

Smart baby monitoring devices are being used to obtain and send video and audio data of the baby to the caregiver's smartphone, but most of these devices are unable to recognize or understand the data. In this project, a novel baby monitoring device is developed which automatically recognizes the undesired and harmful postures of the baby by image processing and sends an alert to the caregiver's smartphone. Deep learning-based object detection algorithms are implemented in the hardware, and a smartphone app is developed. The overall system is shown in Figure below -

This shows the flow used here

The objectives that have been mentioned in the introduction are based on deep learning models that are to be run in embedded systems where resources such as memory and speed are limited. The methods must work in an embedded system with low latency. The system should also work in both day and night conditions. The detection methods should not be biased and be inclusive to all races of babies. The objective of this study is to develop a baby monitoring device that will automatically recognize the harmful postures of the baby by image processing and send an alert to the caregiver's smartphone. The work will be considered successful when the proposed baby monitor can automatically detect the targeted harmful postures and send a notification to the smartphone. Experiments with different postures will be conducted and the latency of the detection algorithms will be measured.

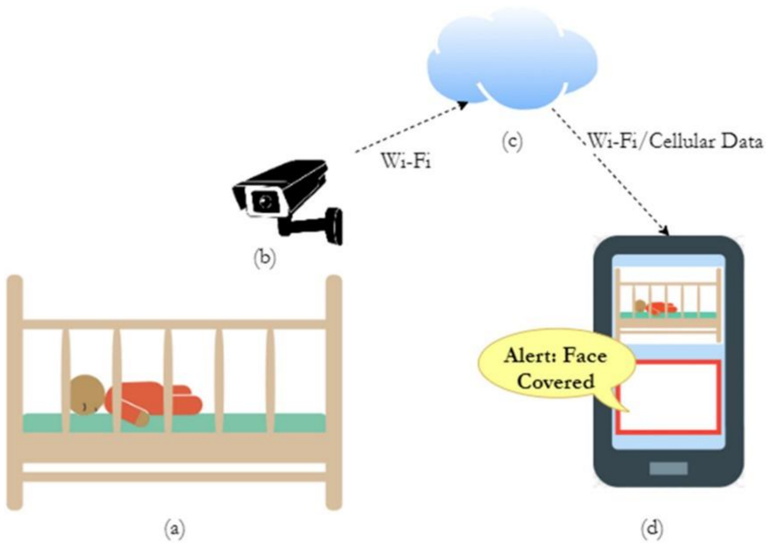


Figure depicts the overall smart baby monitoring system: (a) baby sleeping; (b) smart baby monitoring device automatically detects harmful postures such as face covered, thrown off the blanket, frequently moving or awake; (c) image data are sent to the smartphone through the Internet with the message of any harmful situation; (d) a real-time video of the baby is shown in the smartphone, and notifications and alerts are generated whenever the caregiver receives a message of the harmful situation.

The needs and significances of the proposed system are mentioned below:

- About 1300 babies died due to sudden infant death syndrome (SIDS), about 1300 deaths were due to unknown causes, and about 800 deaths were caused by accidental suffocation and strangulation in bed in 2018 in the USA . Babies are at higher risk for SIDS if they sleep on their stomachs as it causes them to breathe less air. The best and only position for a baby to sleep is on the back—which the American Academy of Pediatrics recommends through the baby's first year . Sleeping on the back improves airflow. To reduce the risk of SIDS, the baby's face should be uncovered, and body temperature should be appropriate . The proposed baby monitor will automatically detect these harmful postures of the baby and notify the caregiver. This will help to reduce SIDS.
- Babies may wake up in the middle of the night due to hunger, pain, or just to play with the parent. There is an increasing call in the medical community to pay attention to parents when they say their babies do not sleep . The smart baby monitor detects whether the baby's eyes are open and sends an alert. Thus, it helps the parents know when the baby is awake even if he/she is not crying.
- When a baby sleeps in a different room, the caregivers need to check the sleeping condition of the baby after a regular interval. Parents lose an average of six months' sleep during the first 24 months of their child's life. Approximately

10% of parents manage to get only 2.5 h of continuous sleep each night. Over 60% of parents with babies aged less than 24 months get no more than 3.25 h of sleep each night. A lack of sleep can affect the quality of work and driving; create mental health problems, such as anxiety disorders and depression; and cause physical health problems, such as obesity, high blood pressure, diabetes, and heart disease . The proposed smart device will automatically detect the situations when the caregiver's attention is required and generate alerts. Thus, it will reduce the stress of checking the baby at regular intervals and help the caregiver to have better sleep.

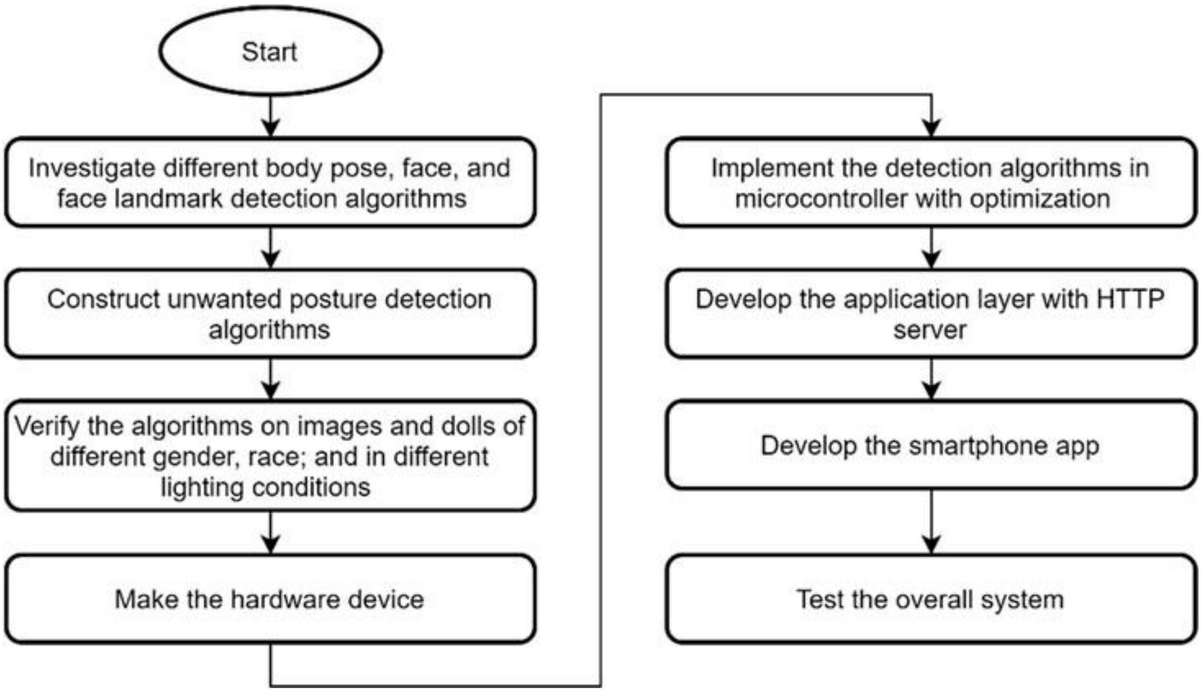
RELATED WORKS:

Several pieces of research have been conducted regarding this project by various companies who are using this project in an actual implementation with differing outcomes which are depicted in this following table as shown below:

Work	Motorola [9]	Infant Optics [10]	Nanit [11]	Lollipop [12]	Cubo Ai [49]	Proposed
Live Video	Yes	Yes	Yes	Yes	Yes	Yes
Boundary Cross Detection	No	No	No	Yes	Yes	No
Cry detection	No	No	No	Yes	Yes	No
Breathing Monitoring	No	No	Yes	No	No	No
Face Covered Detection	No	No	No	No	Yes	Yes
Blanket Removed Detection	No	No	No	No	No	Yes
Frequent Moving Detection	No	No	Yes	No	No	Yes
Awake Detection from Eye	No	No	No	No	No	Yes

MATERIALS AND METHODS:

The steps taken to develop the detection algorithms of harmful and undesired sleeping postures from image data and prototype development of the smart baby monitor are briefly shown in Flowchart. They are described below:



TECHNICAL DESCRIPTION:

The offered system's performance was tested on two different types of hardware i.e., on the laptop and on NVIDIA Jetson Nano, which provides Maxwell 128 core GPU, capable of running multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech recognition.

Nvidia Jetson Nano & attached components -

(a) The whole setup

(b) Jetson Nano

Hardware: The single-board computer-NVIDIA® Jetson NanoTM is used as the main processing unit. It is a small size and low-power embedded platform where neural network models can run efficiently for applications such as image classification, object detection, segmentation, etc. It contains a Quad-core ARM A57 microprocessor running at 1.43 GHz, 4 GB of RAM, a 128-core Maxwell graphics processing unit (GPU), a micro SD card slot, USB ports, and other built-in hardware peripherals. A night-vision camera is interfaced with the Jetson Nano using a USB. When the surrounding light is enough, such as in the daytime, it captures color images. This camera has a built-in light sensor and infrared (IR) LEDs. When the surrounding light is low, the IR LEDs automatically turn on and it captures grayscale images. To connect with the Internet wirelessly, a Wi-Fi adaptor is connected to the USB port of the Jetson Nano. A 110V AC to 5V 4A DC adapter is used as the power supply.

Software: Linux4Tegra (L4T)—a version of Ubuntu operating system (OS)—was installed on a SD card (32 GB) of the Jetson Nano board. All the required packages like opencv and basic models were also installed in our Jetson Nano.For pose estimation initially the mediapipe library was used. However, after making some efforts it was unable to complete our objectives. Then we made a shift to Nvidia's pretrained model for pose estimation called TRTPose (TensorRT Pose Estimation). One of the libraries used in our project is Dlib, which is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real-world problems. It is used in industries and academia in various domains like robotics, embedded devices, mobile phones, and large high- performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge. Although it

is implemented in C++ it can be used with python as well as python is lucid and it is very much useful for developing complex scientific and numeric applications. It supports many operating systems and its large and robust standard library makes python score over other programming languages. Therefore, the sheer code of the proposed system is connected in python.We have used PyCharm IDE as our working environment as well.

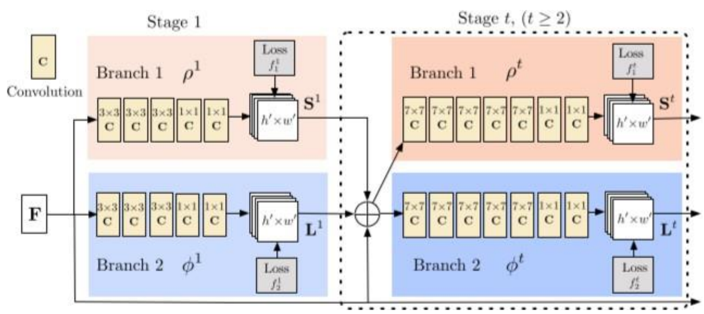
POSE ESTIMATION :

Human pose estimation is the computer vision task of estimating the configuration ('the pose') of the human body by localizing certain key points on a body within a video or a photo.This localization can be used to predict if a person is standing, sitting, lying down, or doing some activity like dancing or jumping.

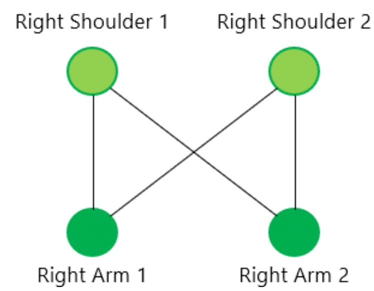
TensorRT Pose Estimation is used for our use case. This project features multi-instance pose estimation accelerated by NVIDIA TensorRT. It is ideal for applications where low latency is necessary. There are two approaches to building a pose estimation model. A top-down approach places bounding boxes around all humans detected in a frame, and then their respective body parts are localized within that bounding box. A bottom-up approach does the opposite. You would first detect all human body parts within a frame and then group parts that belong to a specific person after the fact.

TRTPose takes the bottom-up approach towards pose estimation. The model first detects key points for every body part present in a frame, and then figures out which parts belong to which individual within that frame. The method takes the entire image as the input for a two-branch CNN to jointly predict confidence maps for body part detection, and part affinity fields for parts association. Each PAF has a component in the x direction as well as the y direction, thus representing a vector.Then the parsing step

performs a set of bipartite matchings to associate body parts candidates. Finally, we assemble them into full body poses for all people in the image.

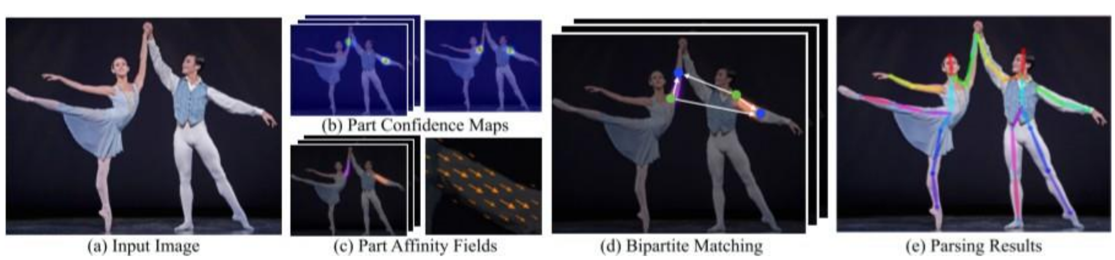


(a)



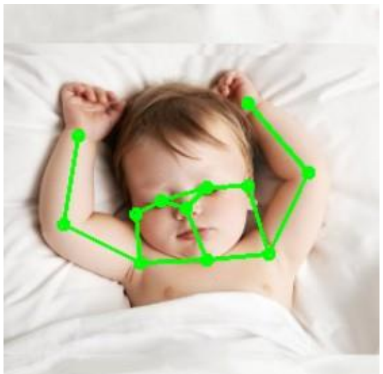
(b)

(a) Architecture of the two-branch multistage CNN. Each stage in the first branch predicts confidence maps S t , and each stage in the second branch predicts PAFs L t . After each stage, the predictions from the two branches, along with the image features, are concatenated for the next stage.
(b) Sample subgraph depicting possible relations between detected keypoints.

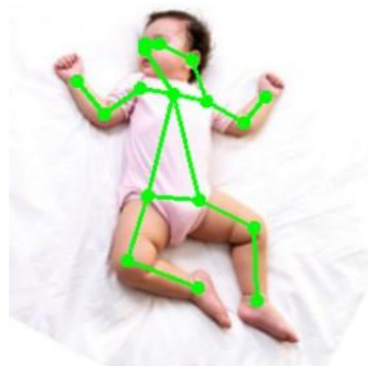


To achieve our objective of this project here, the nose of the baby is detected from the image to decide whether the face is covered due to sleeping on the stomach or for other reasons. To detect a blanket removed, the visibility of the lower body parts such as the hip, knee, and ankle are detected. This is done by the TRT pose estimation model by Nvidia. The features are fed into two-branch multi-stage transposed convolution networks. These branch networks then simultaneously predict the heatmap and Part Affinity Field (PAF) matrices. Finally, a collection of human sets is found, where each human is a set of parts, and where each part contains its relative information. And after the detection and running our algorithms on it, we can now get the alert of the situation whenever the baby is in a vulnerable state. Whether the baby having its blanket removed or the baby sleeping on its stomach, the caretaker can now get notified of it

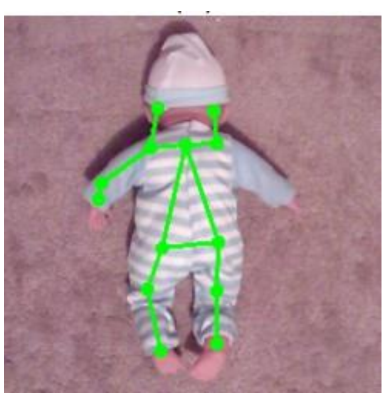
without monitoring the infant 24x7. Thus, the baby remains in a safe state preventing any accidents from happening.



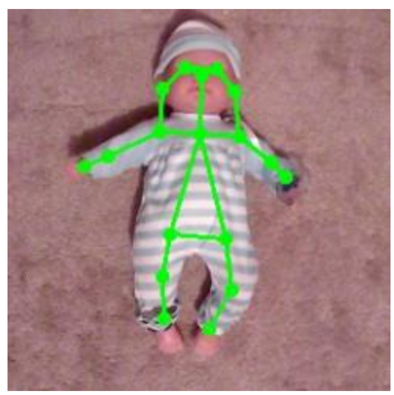
(a)



(b)



(c)



(d)

(a)Baby with blanket on (b)Baby with blanket off (c)Baby sleeping on the back, nose not detected (d)Baby sleeping pose estimated.

AWAKE DETECTION :

The suggested method for detecting whether the baby is awake or not operates on two levels. The procedure begins with the camera recording live video frames, which are

then transferred to a local server. The Dlib library is utilized on the server to identify facial landmarks, and a threshold value is used to determine whether or not the baby is awake. The EAR (Eye Aspect Ratio) is then computed using these face landmarks and given to the driver. The EAR value obtained at the application's end is compared to a threshold value of 0.20 in our system. The baby is regarded to be awake if the EAR value is greater than the threshold value. An alert message will be sent to the parent notifying them that the baby is awake and needs assistance.
For awake detection we have used OpenCV and Python. The Dlib library is used to detect and isolate the facial landmarks using Dlib pre- trained facial landmark detectors. In this approach , 68 facial landmarks have been used.

Facial Landmark marking -
Dlib library is imported and used for the extraction of facial landmarks. Adlib uses a pre-trained face detector , that is an improvement of the histogram of oriented gradients. It consists of two shape predictor models trained on the i-Bug 300-W dataset ,that each localize 68 and 5 landmark points respectively within a face image .We have used the 68 facial landmarks.
In this method, frequencies of gradient direction of an image in localized regions are used to form histograms. It is especially suitable for face detection; it can describe contour and edge features exceptionally in various objects.
For recording the Facial Landmarks, the Facial Landmark Predictor was used by the system to calculate lengths for the EAR values.

The Facial landmark points of the Dlib library which are used to compute EAR are represented in the figure below :

Fig (a) : Facial Landmarks

Algorithm used -

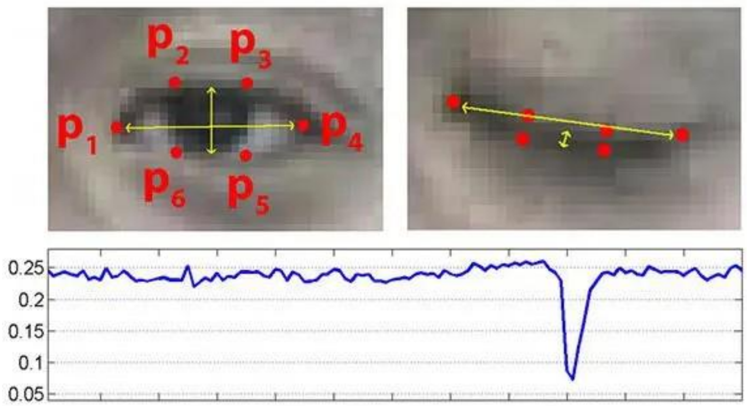


Fig (b) : Eye Aspect Ratio

Here P1,P2,P3,P4,P5,P6 are the pupil coordinates.EAR is generally a constant when eyes are open and is near about 0.5- 0.10.When EAR is greater than 0.20. It is concluded that the baby is awake.

Eye Aspect Ratio(EAR) is the calculated for both the eyes,

$$\frac{(|P2 - P6| + |P3 - P5|)}{2(|P1 - \ P4|)}$$

The numerator determines the distance between the upper and lower eyelids using equation. The horizontal distance of the eye is represented by the denominator. EAR values are utilized to identify whether the baby is awake or asleep in this framework. The average of the EAR values of the left and right eyes is obtained. The EAR is monitored using a counter that sees whether the value falls exceeds the threshold value in the following frame . The infant has closed its eyes and is sleepy,as indicated by the aforementioned circumstances.In contrast, if the EAR values does not maintains the higher value for a following few frames, it means that the baby went back to sleep after a few seconds.The block design of our suggested technique to detect whether the baby is awake or asleep is shown in Figure 3 (Block Diagram).

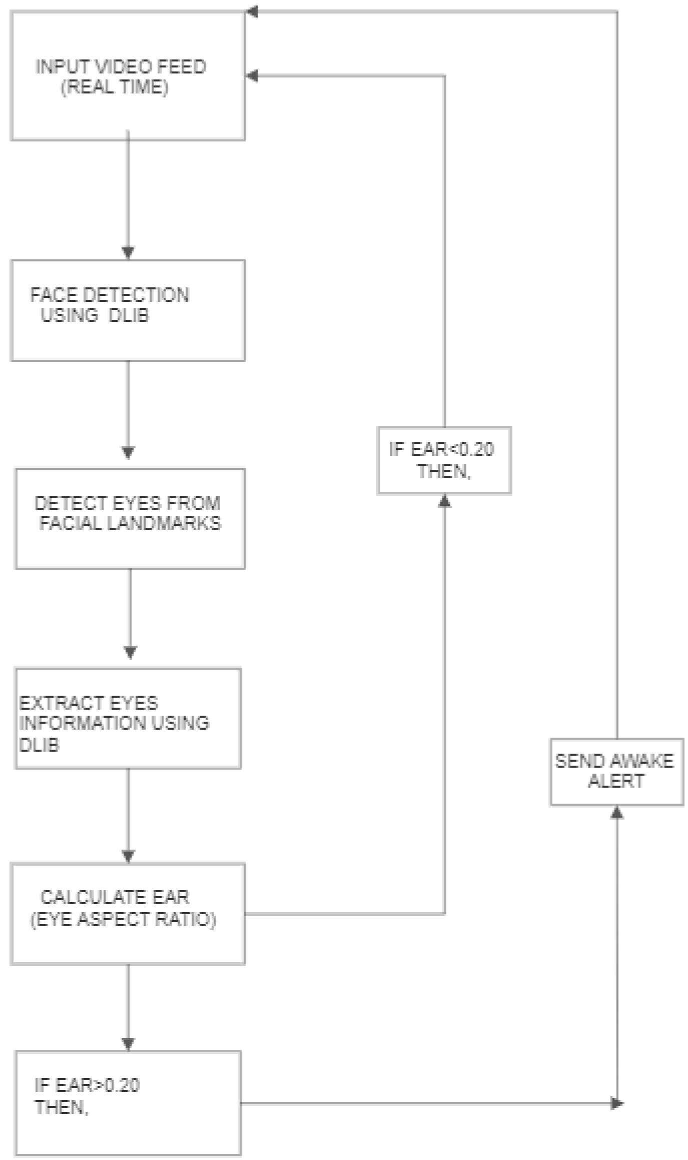


Fig (c) : Block Diagram - Awake Detection

For this project we have utilized a webcam associated with the Jetson Nano .The framework was tried for different samples in various light surrounding lighting conditions.

Eye Detection Accuracy =

$$\frac{\text{Total no. of times eye detected}}{\text{(Total no. of time eye detected + total no. of times eye not detected)}}$$

Awake Detection accuracy =

$$\frac{\text{Total no. of times alert message was displayed}}{\text{(Total no. of times alert message was displayed +Total no. of times alert messages was not sent)}}$$

Result Table for accuracy

Serial No:	Eye Detection accuracy	Awake Detection accuracy
Sample 1	82.34%	80.53%
Sample 2	81%	64.5%
Sample 3	73%	82.34%
Sample 4	77%	65.67%
Sample 5	89.5%	100%
Total	80.56%	78.60%

After training and testing the model rigorously we attained an accuracy of about 78.60%.

Obtained results -

Fig (a) Sample Image

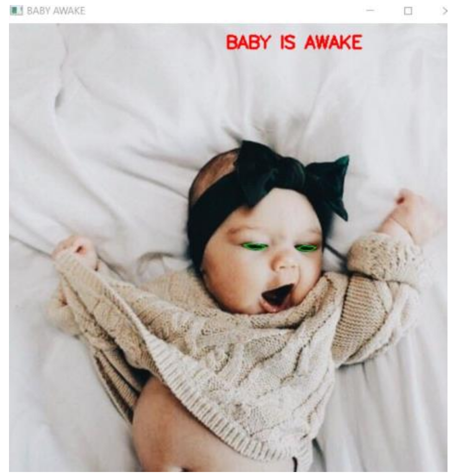


Fig (b) Output Image

We also faced certain limitation mentioned as follows -

- The OpenCV can detect live images up to only an approximate distance up to 25 cm from the Webcam to face. So, the placement of the webcam should be such that this limitation does not affect the model accuracy.
- With poor lighting conditions occasionally, the system is unfit to perceive the eyes. So, it gives a wrong result which must be managed.
- If the orientation of the face is tilted to a specific degree it will in general be perceived , anyway past this framework the framework is unable to identify the face of the baby.

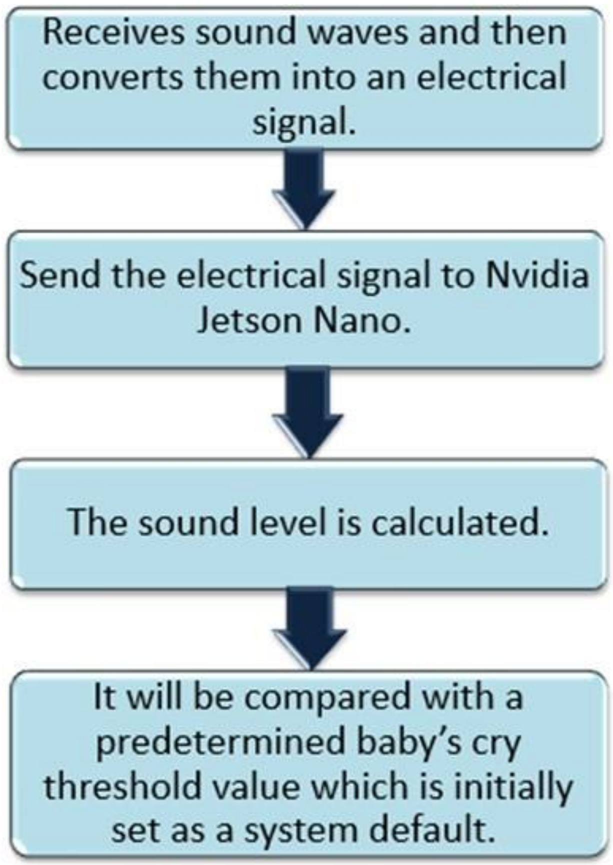
CRYING DETECTION:

An objective, automated and unobtrusive method to quantify crying behavior in an at-home and clinical setting may improve the diagnostic process in excessively crying infants, allow for objective determination of treatment effects by physicians, and enable researchers to include objectively determined cry duration as digital biomarker in clinical trials. Given the importance for researchers to study the relationship between an infant's crying patterns and their health, automatic detection and quantification of infant cries from an audio signal is an essential step in remote baby monitoring applications .

Automatic cry detection has been reported in the form of remote baby monitors for non-intrusive clinical assessments of infants in hospital settings, and several researchers have shown that classification of cry- and non-cry-sounds is possible with machine-learning algorithms.

The aim is to automatically recognize a baby crying while sleeping. In such a case, a soothing sound is played to calm the baby down. This is done by implementing a machine learning algorithm on Jetson Nano.The idea is to train a model on a computer and to deploy it on Jetson Nano,which is used to record a signal and use the model to predict if it is a baby cry or not. In the former case a soothing sound,the caretaker is also notified about the same, in the latter the process(recording and predicting steps) starts again.

A review of infant cry analysis and classification:



Training of the model -

It includes all the steps required to train a machine learning model. First, it reads the data, it performs feature engineering and it trains the model. The model is saved to be used in the prediction step. The _training step_ is performed on a powerful machine, such as our personal computer.

Simulation of the model -

There is a script to test the prediction step on our computer before deployment on Jetson Nano.

The trained model was then deployed from my personal computer to Jetson Nano using FlieClient@Zilla having a suitable Wi-Fi connection and IP address of the Jetson Nano. The trained model is named “model.pkl”.

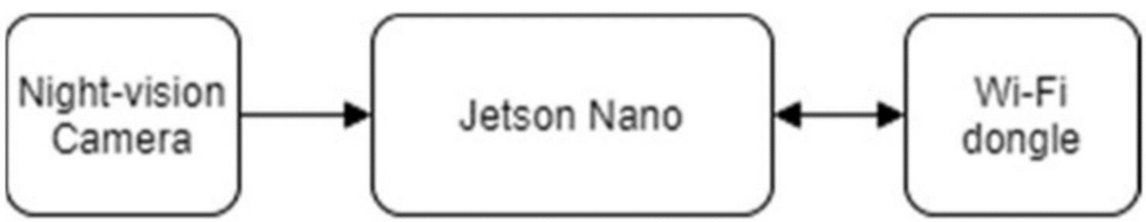
After the deployment, “model.pkl” is opened on Jetson Nano where the prediction of the model is done.

Prediction -

It includes all the steps needed to make a prediction on a new signal.It reads a new signal, it divides it into 5 overlapping signals, it applies the pipeline saved from the training step to make the prediction.The prediction is performed on Jetson Nano.

CONCLUSION:

We proposed in this project an intelligent system that allows parents to monitor a baby from a web application either on a local network or on a remote network. The main advantage of this system is that it is more user friendly and no harm to the baby. This system is designed using the Nvidia Jetson Nano module which has huge advantages over other embedded systems. It can be applicable for the home environment as well as in the hospital or baby nursing care.



Monitoring system

FUTURE WORKS

- Our first focus is to develop a mobile application so that we can directly interact with caretakers and notify them whenever the baby needs their attention.
- To improve our pose estimation model and revise our algorithm for obtaining better results.
- To try other cry detection methods for better results.
- We also aim on performing frequent moving detection
- Use sentiment analysis by using machine learning models trained with baby's expression for example: if he is making a particularly suckling expression in sleep which would mean he is hungry so notification of hunger would be sent to parents. This will be done through audio and video capturing.

REFERENCES

- https://github.com/NVIDIA-AI-IOT/deepstream_pose_estimation
- https://developer.nvidia.com/blog/creating-a-human-pose-estimation-application-with-deepstream-sdk/
- https://github.com/NVIDIA-AI-IOT/trt_pose
- Tareq Khan, Article An Intelligent Baby Monitor with Automatic Sleeping Posture Detection and Notification, School of Engineering, Eastern Michigan University, Ypsilanti, MI 48197, USA; tareq.khan@emich.edu
- https://pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/
- Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh, Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, The Robotics Institute, Carnegie Mellon University
- https://www.researchgate.net/publication/336989127_Baby_Cry_Detection_Deep_Learning_and_Classical_Approaches
- https://www.frontiersin.org/articles/10.3389/fped.2021.651356/full
- http://dlib.net/ml.html
- Jose Gomes da Silva Neto, Joao Marcelo Xavier Natario Teixeira, Veronica Teichrieb, Analyzing embedded pose estimation solutions for human behavior understanding

UNCOVERING THE BUILDING BLOCKS OF TURBULENCE

FAHIM YOUSUF CHOUDHURY
ROLL NO- BTECH/10529/19
MENTOR: SCOTT DAWSON

FINAL GROUP REPORT

Introduction

The main goal of this research work is to minimize the computational complexity and cost that goes into representing various fluid mechanics and aerodynamic systems in a low dimensional form in order to understand and analyse them better.

Methodology

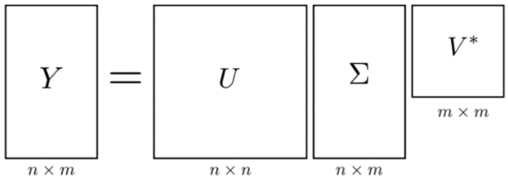
The datasets that we are dealing with are mostly functions of both space and time. In order to understand pertinent features of such datasets as well as to fulfil our goal of building a reduced-complexity model, our first step is to separate the variables using Singular Value Decomposition (SVD) and then using various curve fitting tools in MATLAB and PYTHON we create an analytical function that fits our dataset.

The decomposition methods presented in this report are founded on the singular value decompositions of matrices or operators.

Singular Value Decomposition

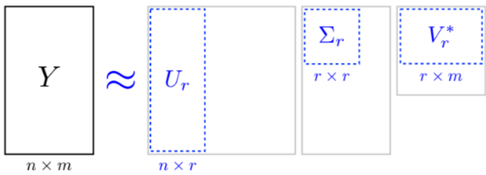
$$y(x,t) = \sum_{j=1}^{m} u_j(x) a_j(t)$$

In order to achieve this decomposition, we use SVD. The main idea of this is to break the matrix containing spatial and time coordinates into two separate vectors i.e.,



U and ***V*** are unitary matrices and $\sum$ is diagonal matrix with decreasing and nonnegative diagonal entries.

We truncate the above matrices in order to eliminate noise and we get the following,



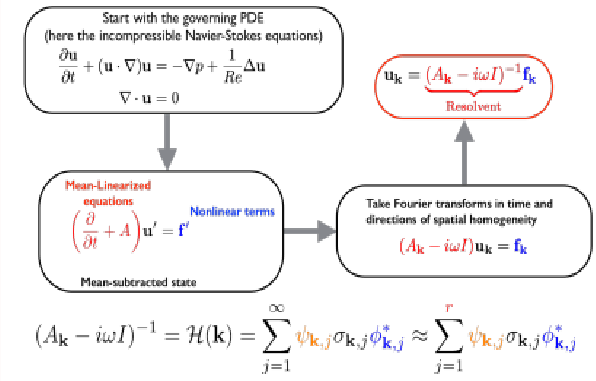
We then solve this expression and obtain

$$Y_r = U_r \Sigma_r V_r^* = \sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^*$$

Here uj and vj are vectors that are the j-th columns of Ur and Vr.

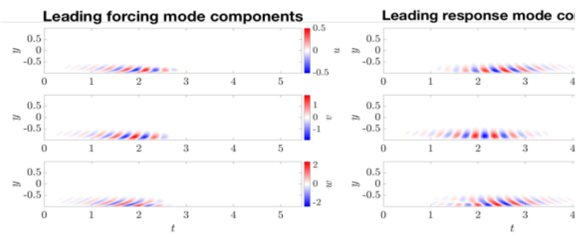
We use predicted coherent structures from a method called resolvent analysis, which looks at a singular value decomposition of a linear operator obtained from the governing Navier-Stokes equations

FINAL GROUP REPORT



The leading left and right singular vectors give the optimal response and forcing modes of the system.

Sample space-time modes in turbulent channel flow



Curve Fitting

The absolute and real components of u_j *and* v_j matrices are fitted using gaussian functions with the help of **curve_fit** function in Python and MATLAB.

The general form of the gaussian function assumed for fitting the components is,

$$\psi(y,t) = c\exp\left[ia_y y - b_y(y-y_0)^2\right]\exp\left[ia_t t - b_t(t -\right.$$

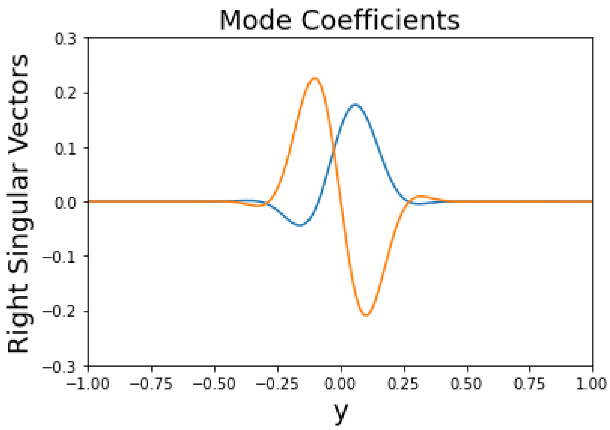
The unknown parameters of these functions are determined using curve fitting tools. For a better fit we enforce additional boundary conditions by adding additional image wavepackets.

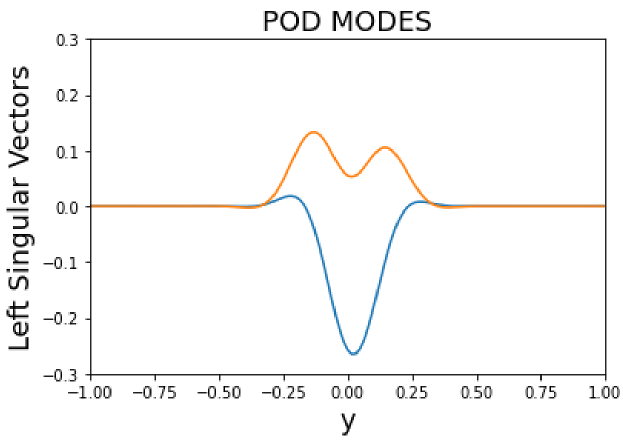
Results

Using the method discussed in the previous section, we have succeeded in creating analytical functions for various datasets. The results are shown and discussed below –

For TestMatrix –

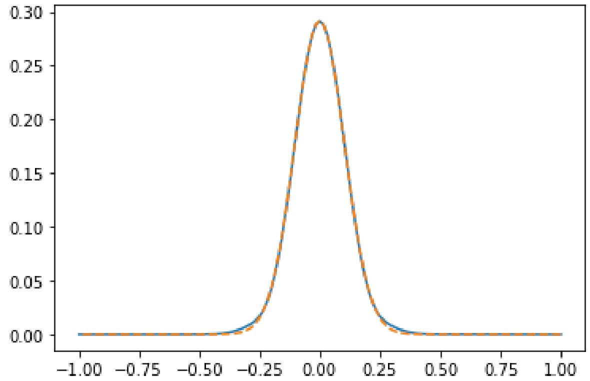
For our first dataset, we extracted the left and right singular vectors using built-in SVD tools in Matlab and python.



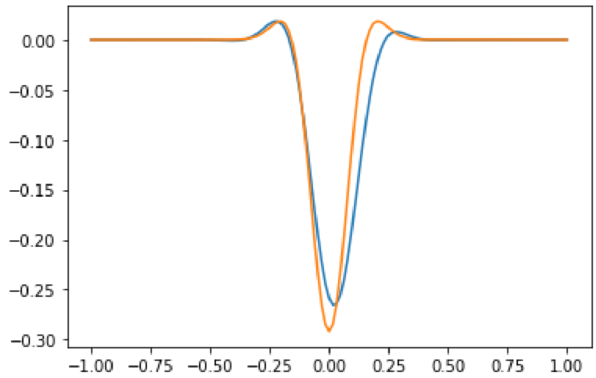


Now since we have separated space and time variables, we are in a position to fit them. Left Singular Vector is a function of space while Right Singular Vector is a function of time. For our first dataset, we fitted the absolute value of the above

vectors using one gaussian and the results were-

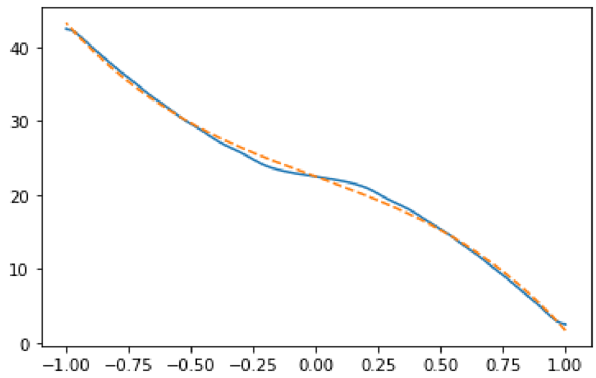


Fit of abs Left Singular Vector



Fit of real Left Singular Vector

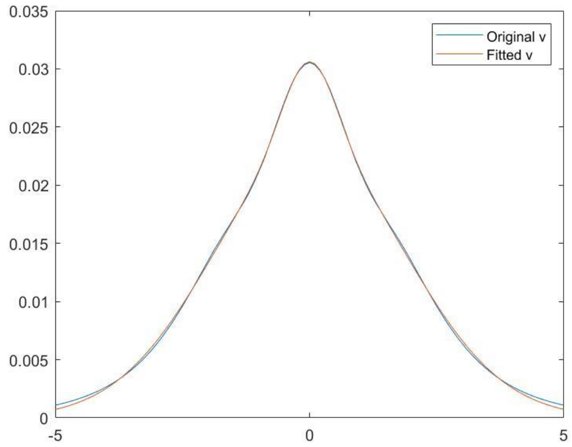
We also looked at the phase of the Left Singular Vector and fitted it.



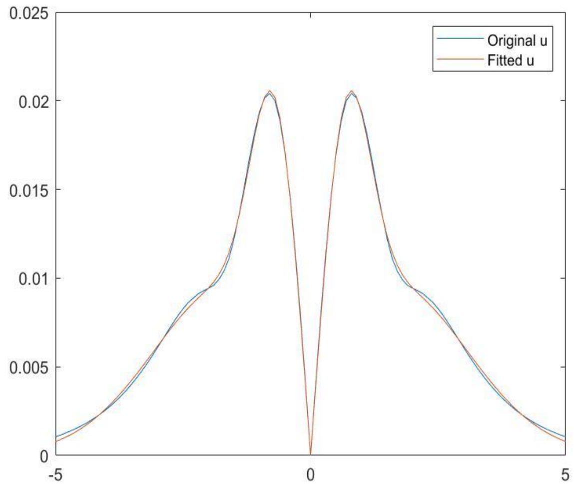
Now that we also have the phase, we can fit the real part of the vectors also. This framework has been used by us for the other datasets also.

For Flow over a Cylinder Data –

Here, we are going over some real world data and using the same analysis as before -

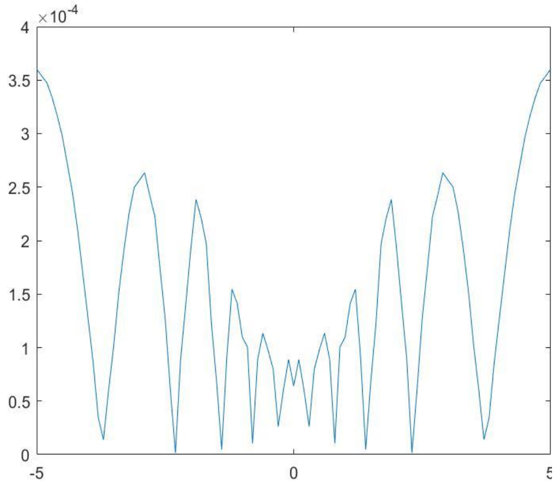


This fit was achieved using two gaussians, one for the inner region and the other accounting for the outer.

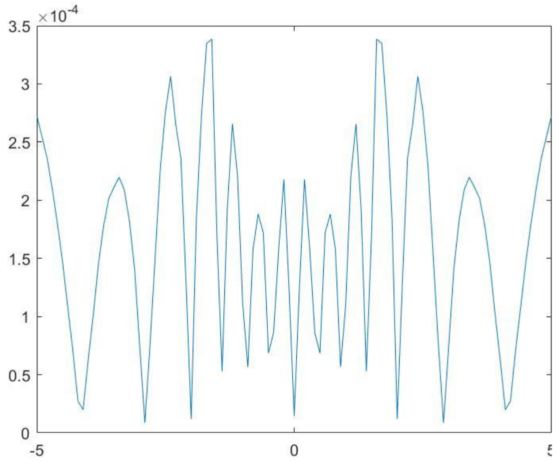


The above graph was obtained using 4 gaussians paired with a starting points guess as well as upper and lower bounds. These two graphs represent the u and v component of velocity of flow over a cylinder.

The error in both these cases is shown by the following graphs-



Error for v component of velocity

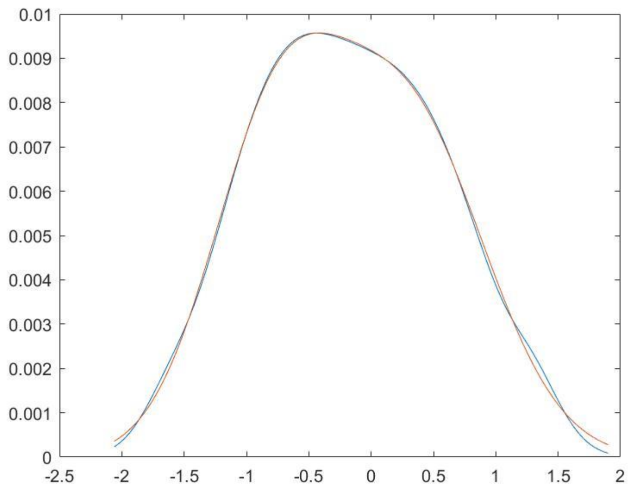


Error for u component of velocity

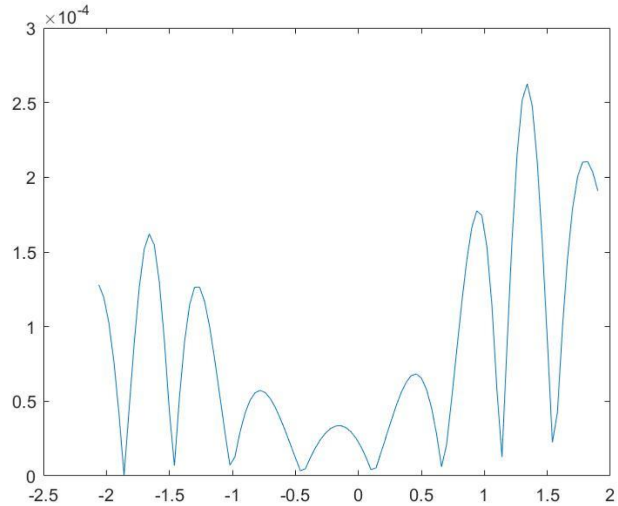
As we can clearly see, the error is negligible. That means our analytical function works and can be used for studying the dataset. Similarly, we were able to successfully create functions for other datasets also beginning with airfoil data.

For Flow over an Airfoil Data -

For this first plot, only 2 gaussians were required to achieve the necessary fit.



Velocity fitting graph



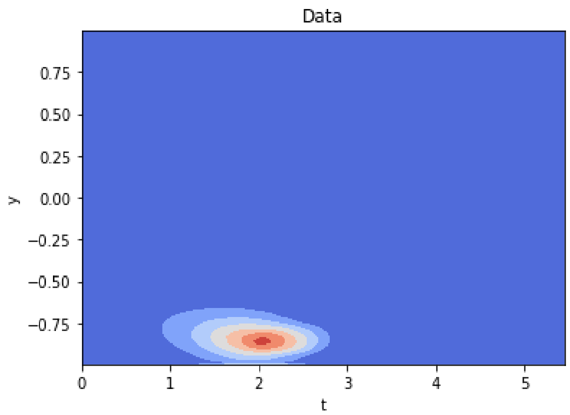
Error plot

As we can see, the error in this case is also negligible which is a good thing as our analysis has now worked on two sets of real world data. We will now look at some other data to test our theory even further.

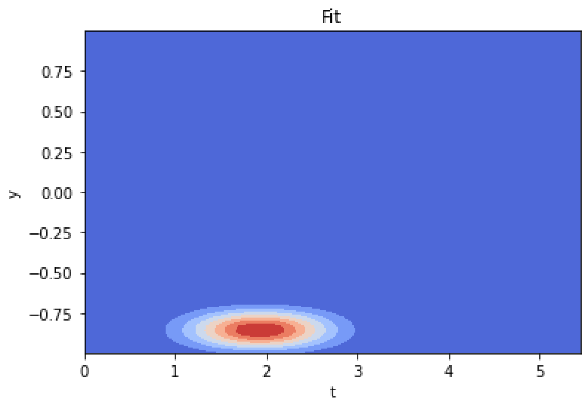
FINAL GROUP REPORT

For DataModes –

The below graph is a contour graph of the absolute value of PhiU. The second graph is obtained after finishing our fitting in time with one gaussian and two gaussian plus boundary conditions for space. As we can see, they match to a great extent.

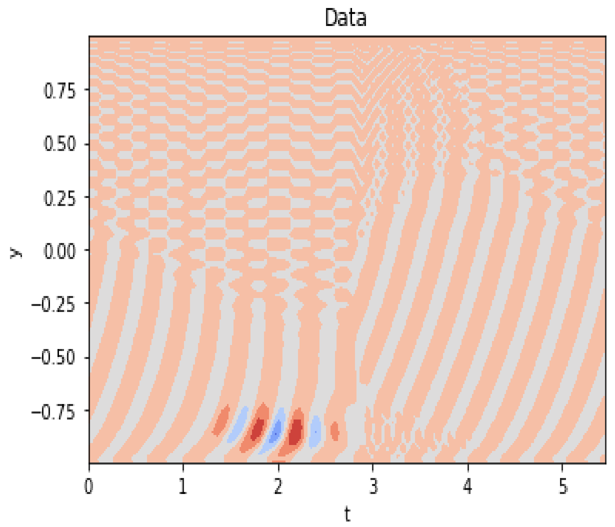


Original abs PhiU velocity

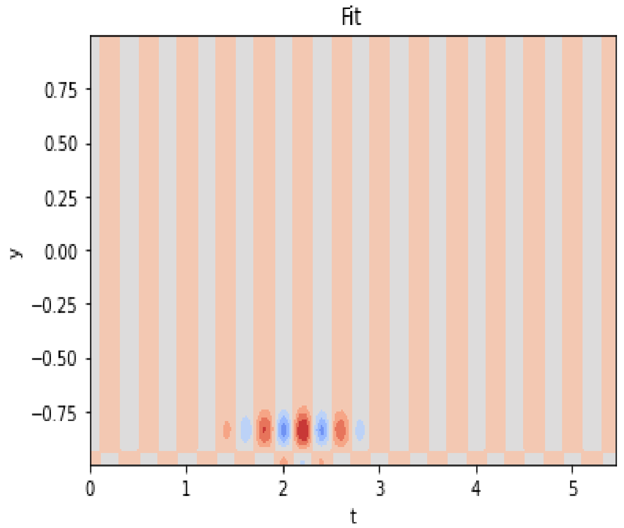


Fitted abs PhiU velocity

The next two graphs are of the real part of PhiU where we have extracted the phase and then fit a linear function to it using our fitting tools and combined it with our absolute valued function to get the fit of the real part. The graph matches the original in the x-direction but deviates a little in the y due to the phase.

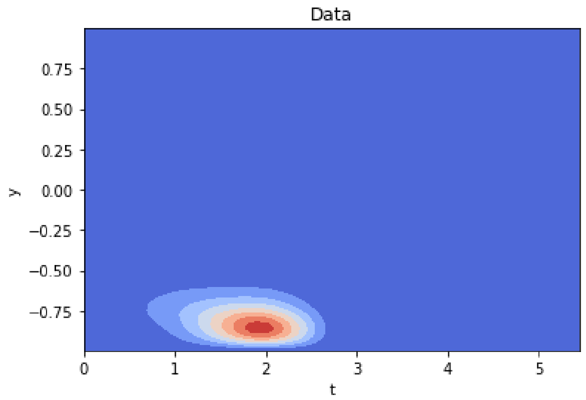


Original real PhiU velocity

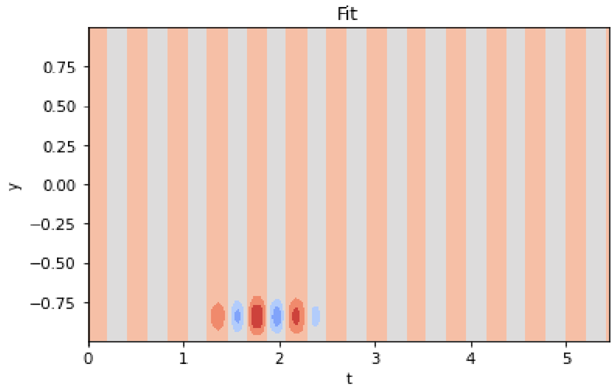


Fitted real PhiU velocity

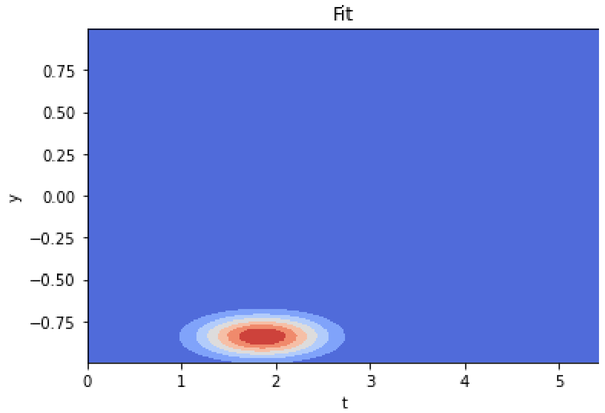
The same trend continues for PhiV, PsiU, PsiV and PsiW wherein we select that number of gaussian which reduces error and if necessary boundary conditions as shown by the following plots in the next few pages -



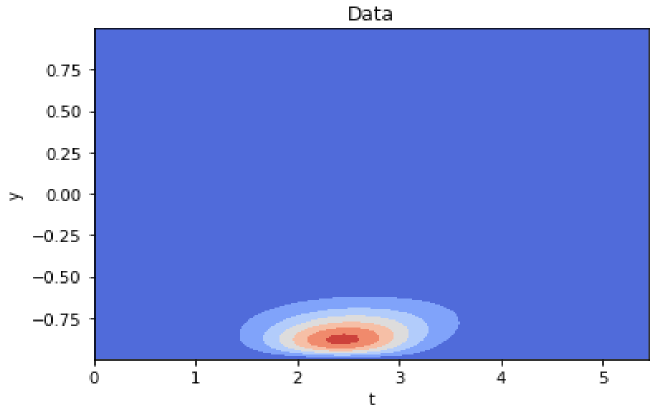
Original abs PhiV velocity



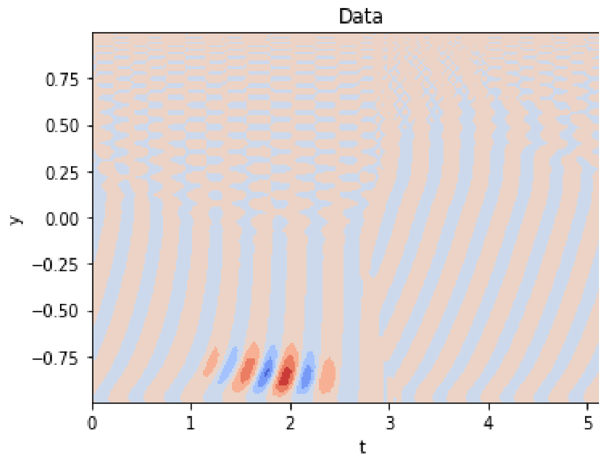
Fitted real PhiV velocity



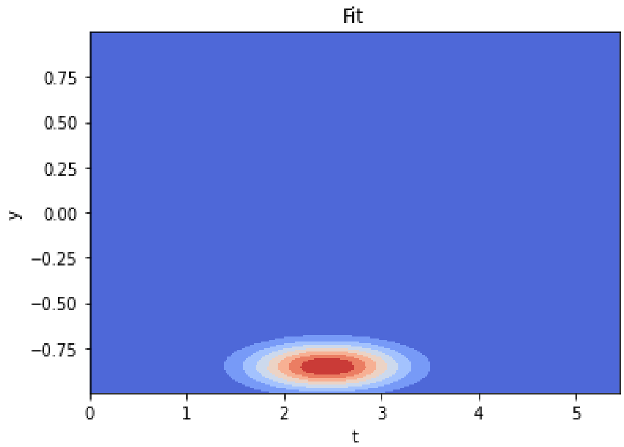
Fitted abs PhiV velocity



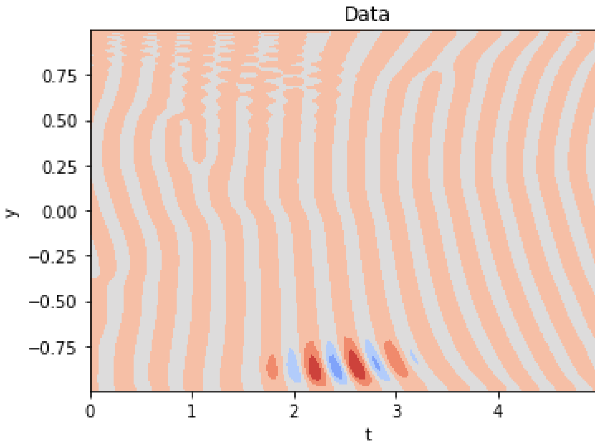
Original abs PsiU velocity



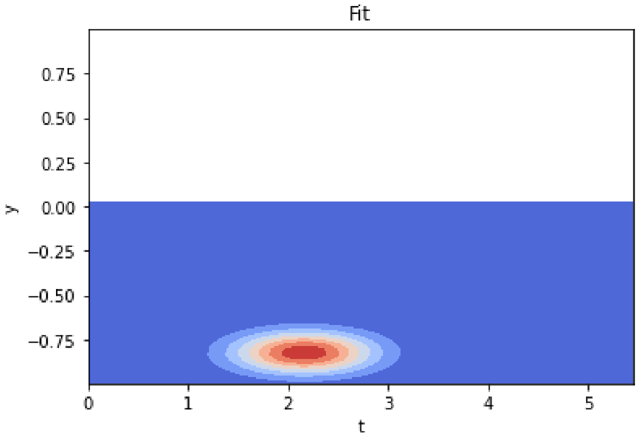
Original real PhiV velocity



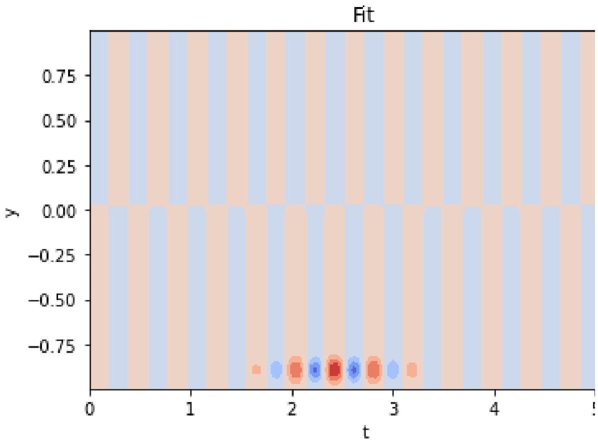
Fitted abs PsiU velocity



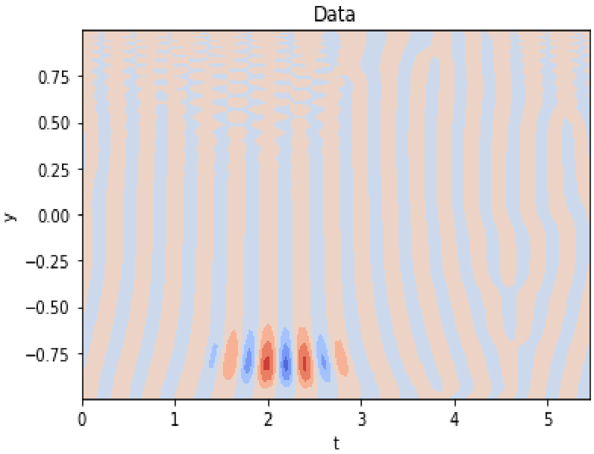
Original real PsiU velocity



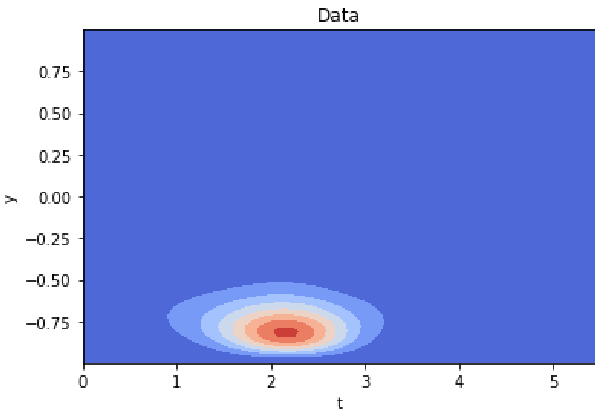
Fitted abs PsiV velocity



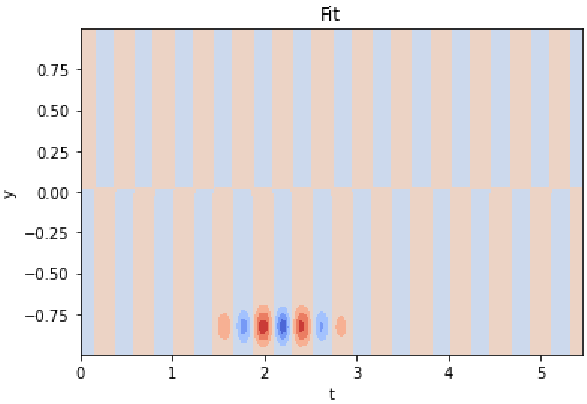
Fitted real PsiU velocity



Original real PsiV velocity

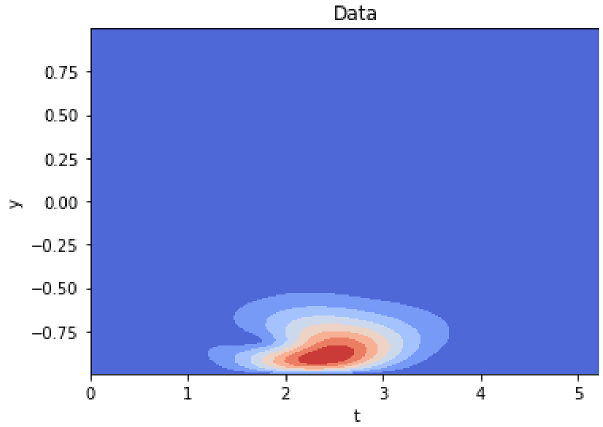


Original abs PsiV velocity

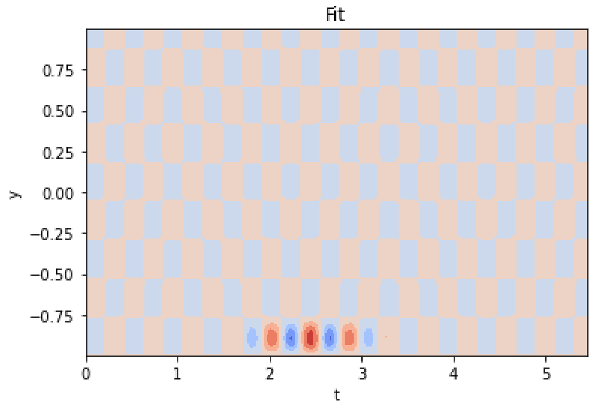


Fitted real PsiV velocity

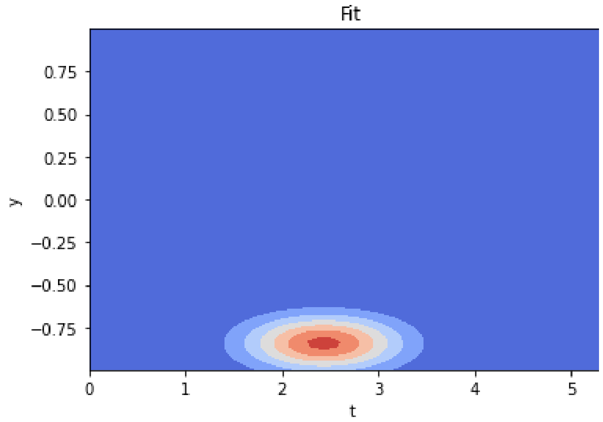
FINAL GROUP REPORT



Original abs PsiW velocity



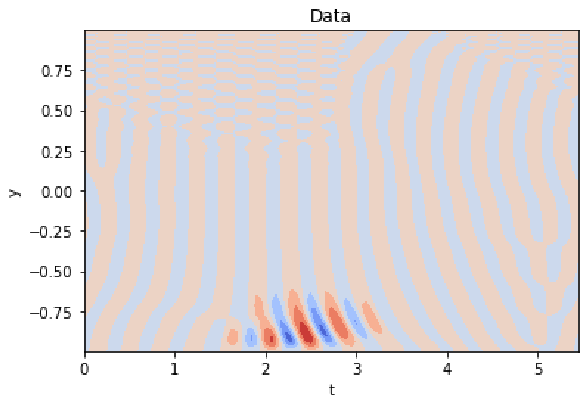
Fitted real PsiW velocity



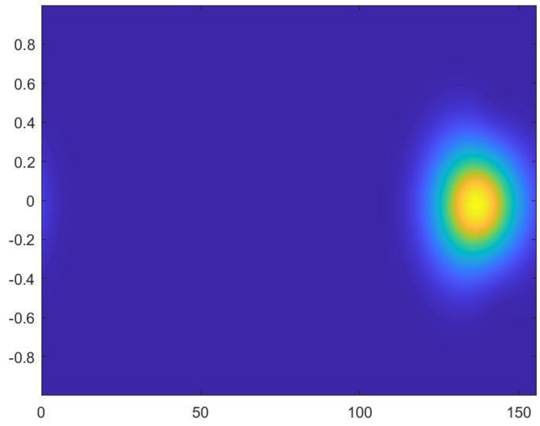
Fitted abs PsiW velocity

For DataModesV3-

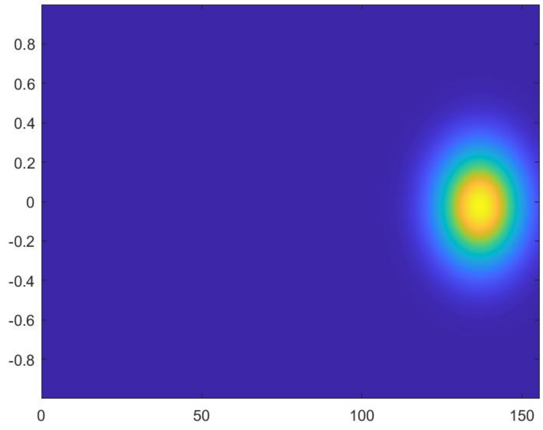
As evident by the name, DataModesV3 is an updated dataset of our previous DataModes. We have once again used our framework to create functions that represent the original data to a great extent as evident by the following plots-



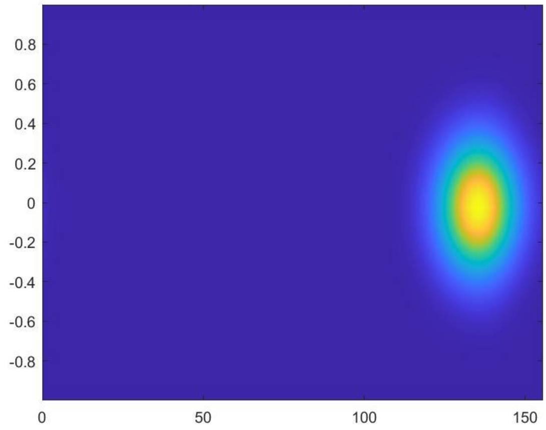
Original real PsiW velocity



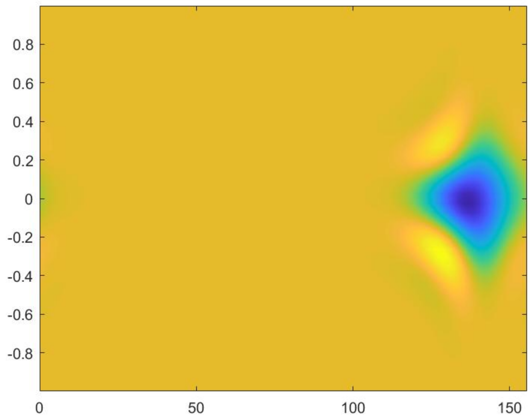
Original abs PhiU velocity



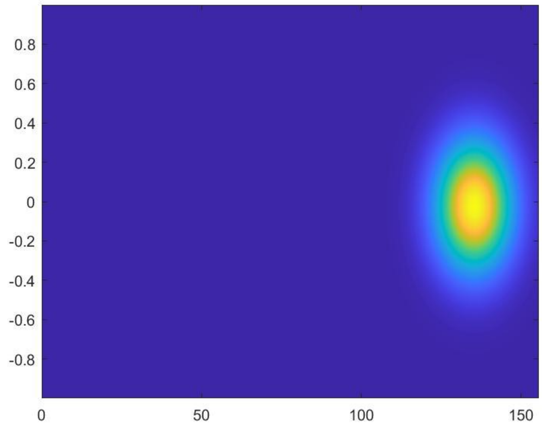
Fitted abs PhiU velocity



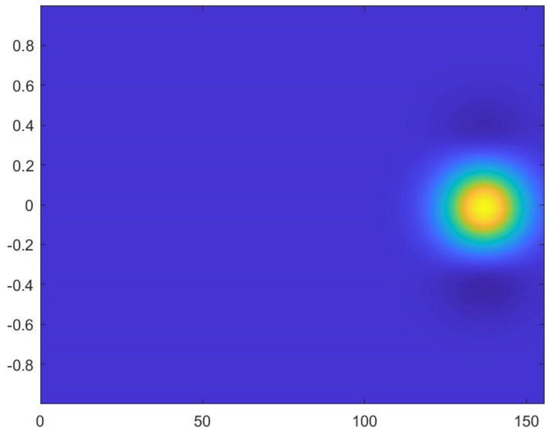
Original abs PhiV velocity



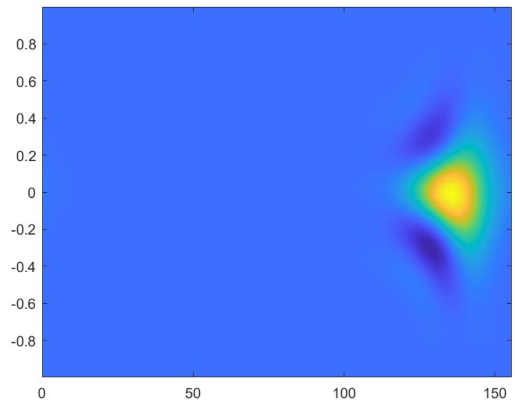
Original real PhiU velocity



Fitted abs PhiV velocity

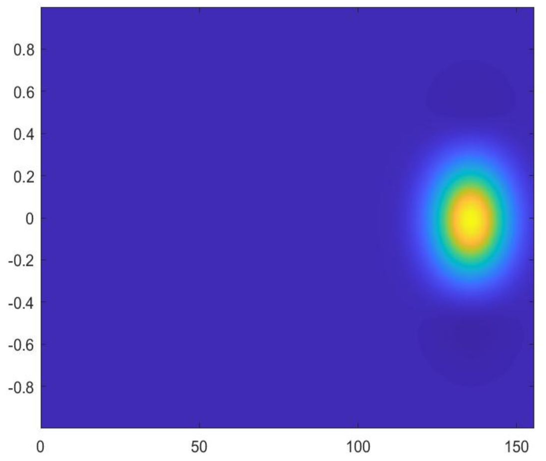


Fitted real PhiU velocity



Original real PhiV velocity

FINAL GROUP REPORT



Fitted real PhiV velocity

Conclusion –

After performing our analysis on a variety of data including both real world and theoretical and looking at the results that we have obtained, it is clear that our theory works and error is also minimal. But this theory does not provide us with a straight forward path as we have seen that in many cases, we had to select more than one gaussian sometimes even four and on top of that we also had to use boundary conditions and start point guesses sometimes in order to achieve a better fit. But our analysis also has its flaws as when it came to the real part of a function in DataModes, we were not able to get a great fit perhaps due to the phase not matching to a great extent or some other underlying issue. But apart from this, our error was less than 1% in flow over a cylinder, air-foil data, DataModesV3 and TestMatrix dataset.

Further Work –

In our work, we have done our analysis mostly on datasets that resemble a gaussian function in some form or the other. Further work may be done to expand our theory to cover a wide range of functions and eliminate some of the issues we encountered in our present work.

Acknowledgement –

We would like to thank you sir for giving us the opportunity to do this research which allowed us to hone our skills in Matlab and python and we also got to know about the basics of research work for which we are eternally grateful. We would also like to wish you good health and success in your future work.

FINAL GROUP REPORT

CERTIFICATE-

ILLINOIS TECH | Armour College of Engineering

Upon the recommendation of the faculty of the
Armour College of Engineering
Of Illinois Institute of Technology

Fahim Yousuf Choudhury
Is recognized as achieving

Summer Undergraduate Engineering
Research Immersion Program
Certificate of Participation

With all the rights, privileges and honors thereunto appertaining
Awarded at Chicago, in the State of Illinois of the United States of America
July 15, 2022

Dean, Kevin W. Cassel

Armour R&D

ENGR 498-07 Research in Artificial Intelligence and Deep Learning

SMART INFANT MONITORING SYSTEM

TEAM MEMBERS:

DIPAN MUKHOPADHYAY

GURPREET SINGH

ABHISHEK RAJ SHEKHAR

Advisor: Dr. Jafar Saniie

Assistant Mentor: Xinrui Yu

SUMMER 2022

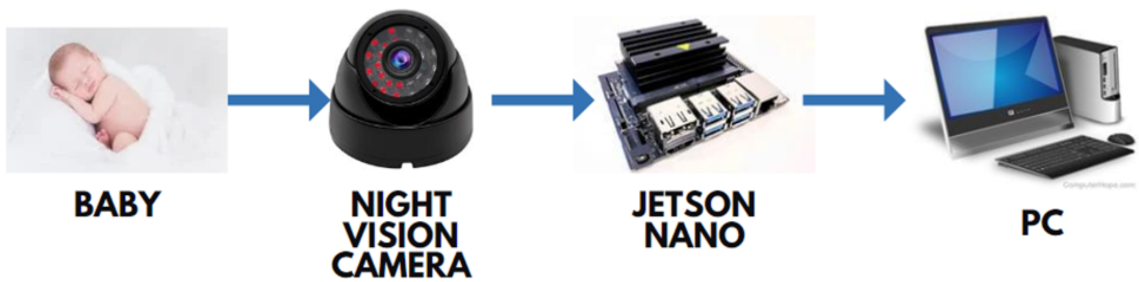
SMART INFANT MONITORING SYSTEM | FINAL REPORT

ABSTRACT

The goal of this project is to design a smart infant monitoring system that can help parents to have a watch on their baby and can notify them about the same. A Non-contact-based baby monitoring system using image Artificial Intelligence and Image processing is proposed in this project which is used for proper safety and monitoring the activity of babies by their busy parents. This is done by sending the video feed to the caregiver's smartphone. The system detects the unwanted positions in which the baby is in. The following would be monitored here: (a) whether the baby's face was covered due to sleeping on the stomach; (b) whether the baby threw off the blanket from the body; (c) whether the baby was moving frequently; (d) whether the baby's eyes were opened due to awakening.(e) baby crying detection is also proposed to detect its crying patterns. Thus the caregivers were not required to monitor their babies after regular intervals. They were given an alert message when their presence was required. The device was developed using NVIDIA's Jetson Nano microcontroller. A night vision camera(USB system) with distinctive features and secured Wi-Fi connectivity was interfaced. For this we have used Python as a programming language and OpenCV as an Image Processing Library.

INTRODUCTION

Smart baby monitoring devices are being used to obtain and send video and audio data of the baby to the caregiver's smartphone, but most of these devices are unable to recognize or understand the data. In this project, a novel baby monitoring device is developed which automatically recognizes the undesired and harmful postures of the baby by image processing and sends an alert to the caregiver's smartphone. Deep learning-based object detection algorithms are implemented in the hardware, and a smartphone app is developed. The overall system is shown in Figure below -



This shows the flow used here

The objectives that have been mentioned in the introduction are based on deep learning models that are to be run in embedded systems where resources such as memory and speed are limited. The methods must work in an embedded system with low latency. The system should also work in both day and night conditions. The detection methods should not be biased and be inclusive to all races of babies. The objective of this study is to develop a baby monitoring device that will automatically recognize the harmful postures of the baby by image processing and send an alert to the caregiver's smartphone. The work will be considered successful when the proposed baby monitor can automatically detect the targeted harmful postures and send a notification to the smartphone. Experiments with different postures will be conducted and the latency of the detection algorithms will be measured.

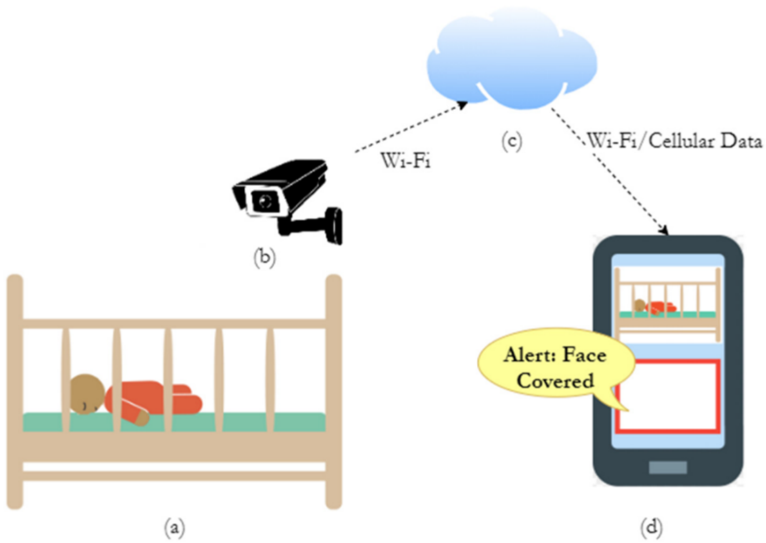


Figure depicts the overall smart baby monitoring system: (a) baby sleeping; (b) smart baby monitoring device automatically detects harmful postures such as face covered, thrown off the blanket, frequently moving or awake; (c) image data are sent to the smartphone through the Internet with the message of any harmful situation; (d) a real-time video of the baby is shown in the smartphone, and notifications and alerts are generated whenever the caregiver receives a message of the harmful situation.

The needs and significances of the proposed system are mentioned below:

- About 1300 babies died due to sudden infant death syndrome (SIDS), about 1300 deaths were due to unknown causes, and about 800 deaths were caused by accidental suffocation and strangulation in bed in 2018 in the USA . Babies are at higher risk for SIDS if they sleep on their stomachs as it causes them to breathe less air. The best and only position for a baby to sleep is on the back—which the American Academy of Pediatrics recommends through the baby's first year . Sleeping on the back improves airflow. To reduce the risk of SIDS, the baby's face should be uncovered, and body temperature should be appropriate . The proposed baby monitor will automatically detect these harmful postures of the baby and notify the caregiver. This will help to reduce SIDS.
- Babies may wake up in the middle of the night due to hunger, pain, or just to play with the parent. There is an increasing call in the medical community to pay attention to parents when they say their babies do not sleep . The smart baby monitor detects whether the baby's eyes are open and sends an alert. Thus, it helps the parents know when the baby is awake even if he/she is not crying.
- When a baby sleeps in a different room, the caregivers need to check the sleeping condition of the baby after a regular interval. Parents lose an average of six months' sleep during the first 24 months of their child's life. Approximately

10% of parents manage to get only 2.5 h of continuous sleep each night. Over 60% of parents with babies aged less than 24 months get no more than 3.25 h of sleep each night. A lack of sleep can affect the quality of work and driving; create mental health problems, such as anxiety disorders and depression; and cause physical health problems, such as obesity, high blood pressure, diabetes, and heart disease . The proposed smart device will automatically detect the situations when the caregiver's attention is required and generate alerts. Thus, it will reduce the stress of checking the baby at regular intervals and help the caregiver to have better sleep.

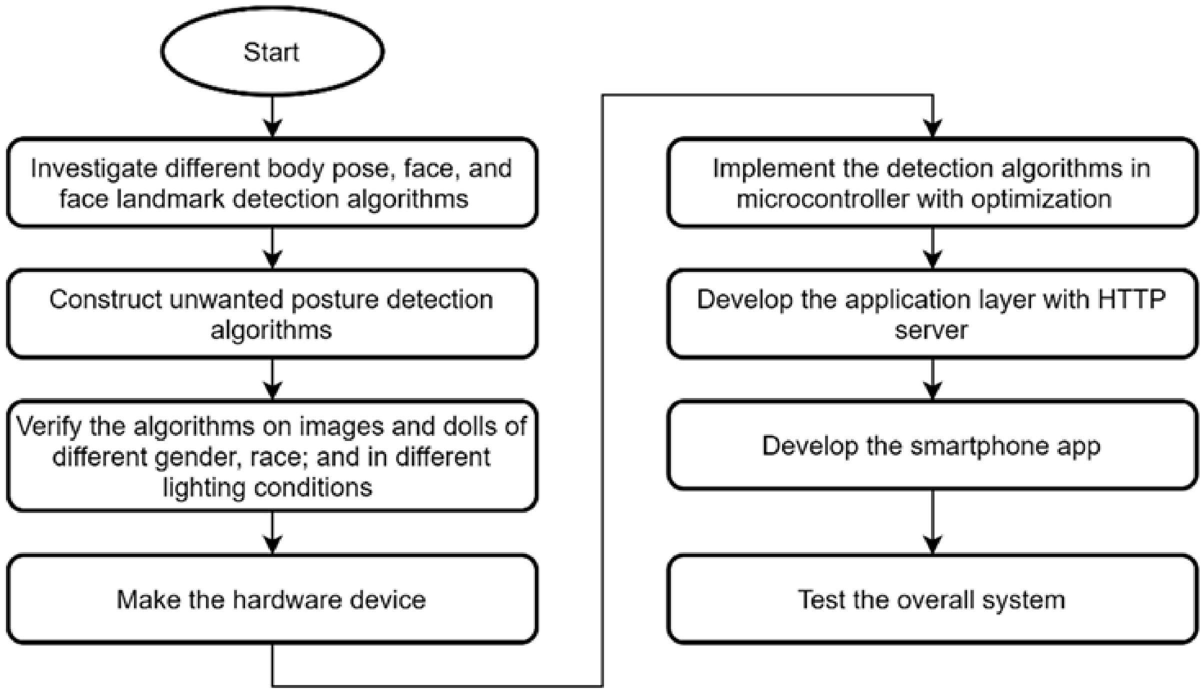
RELATED WORKS:

Several pieces of research have been conducted regarding this project by various companies who are using this project in an actual implementation with differing outcomes which are depicted in this following table as shown below:

Work	Motorola [9]	Infant Optics [10]	Nanit [11]	Lollipop [12]	Cubo Ai [49]	Proposed
Live Video	Yes	Yes	Yes	Yes	Yes	Yes
Boundary Cross Detection	No	No	No	Yes	Yes	No
Cry detection	No	No	No	Yes	Yes	No
Breathing Monitoring	No	No	Yes	No	No	No
Face Covered Detection	No	No	No	No	Yes	Yes
Blanket Removed Detection	No	No	No	No	No	Yes
Frequent Moving Detection	No	No	Yes	No	No	Yes
Awake Detection from Eye	No	No	No	No	No	Yes

MATERIALS AND METHODS:

The steps taken to develop the detection algorithms of harmful and undesired sleeping postures from image data and prototype development of the smart baby monitor are briefly shown in Flowchart. They are described below:



TECHNICAL DESCRIPTION:

The offered system's performance was tested on two different types of hardware i.e., on the laptop and on NVIDIA Jetson Nano, which provides Maxwell 128 core GPU, capable of running multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech recognition.

Nvidia Jetson Nano & attached components -

(a) The whole setup

(b) Jetson Nano

Hardware: The single-board computer-NVIDIA® Jetson NanoTM is used as the main processing unit. It is a small size and low-power embedded platform where neural network models can run efficiently for applications such as image classification, object detection, segmentation, etc. It contains a Quad-core ARM A57 microprocessor running at 1.43 GHz, 4 GB of RAM, a 128-core Maxwell graphics processing unit (GPU), a micro SD card slot, USB ports, and other built-in hardware peripherals. A night-vision camera is interfaced with the Jetson Nano using a USB. When the surrounding light is enough, such as in the daytime, it captures color images. This camera has a built-in light sensor and infrared (IR) LEDs. When the surrounding light is low, the IR LEDs automatically turn on and it captures grayscale images. To connect with the Internet wirelessly, a Wi-Fi adaptor is connected to the USB port of the Jetson Nano. A 110V AC to 5V 4A DC adapter is used as the power supply.

Software: Linux4Tegra (L4T)—a version of Ubuntu operating system (OS)—was installed on a SD card (32 GB) of the Jetson Nano board. All the required packages like opencv and basic models were also installed in our Jetson Nano.For pose estimation initially the mediapipe library was used. However, after making some efforts it was unable to complete our objectives. Then we made a shift to Nvidia's pretrained model for pose estimation called TRTPose (TensorRT Pose Estimation). One of the libraries used in our project is Dlib, which is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real-world problems. It is used in industries and academia in various domains like robotics, embedded devices, mobile phones, and large high- performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge. Although it

is implemented in C++ it can be used with python as well as python is lucid and it is very much useful for developing complex scientific and numeric applications. It supports many operating systems and its large and robust standard library makes python score over other programming languages. Therefore, the sheer code of the proposed system is connected in python.We have used PyCharm IDE as our working environment as well.

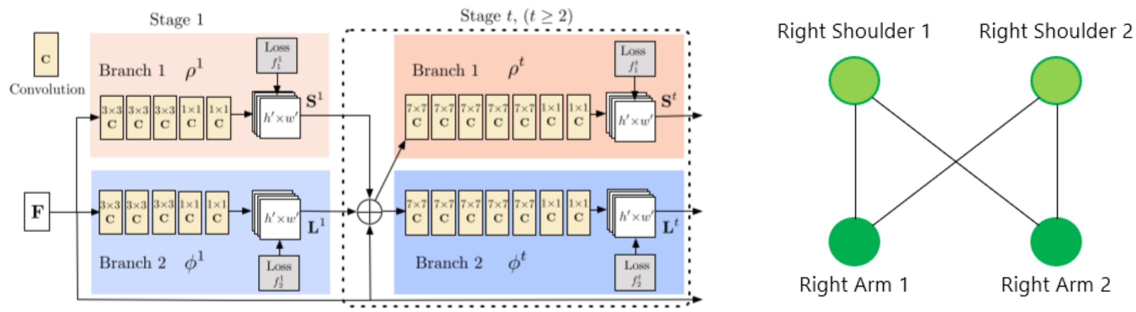
POSE ESTIMATION :

Human pose estimation is the computer vision task of estimating the configuration ('the pose') of the human body by localizing certain key points on a body within a video or a photo.This localization can be used to predict if a person is standing, sitting, lying down, or doing some activity like dancing or jumping.

TensorRT Pose Estimation is used for our use case. This project features multi-instance pose estimation accelerated by NVIDIA TensorRT. It is ideal for applications where low latency is necessary. There are two approaches to building a pose estimation model. A top-down approach places bounding boxes around all humans detected in a frame, and then their respective body parts are localized within that bounding box. A bottom-up approach does the opposite. You would first detect all human body parts within a frame and then group parts that belong to a specific person after the fact.

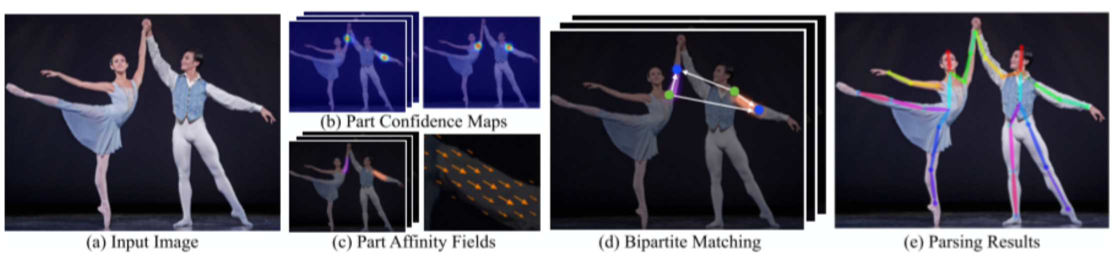
TRTPose takes the bottom-up approach towards pose estimation. The model first detects key points for every body part present in a frame, and then figures out which parts belong to which individual within that frame. The method takes the entire image as the input for a two-branch CNN to jointly predict confidence maps for body part detection, and part affinity fields for parts association. Each PAF has a component in the x direction as well as the y direction, thus representing a vector.Then the parsing step

performs a set of bipartite matchings to associate body parts candidates. Finally, we assemble them into full body poses for all people in the image.



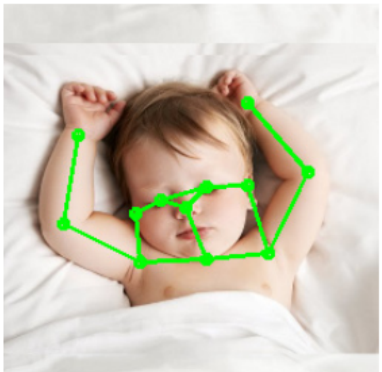
(a) (b)

(a) Architecture of the two-branch multistage CNN. Each stage in the first branch predicts confidence maps S t , and each stage in the second branch predicts PAFs L t . After each stage, the predictions from the two branches, along with the image features, are concatenated for the next stage.
(b) Sample subgraph depicting possible relations between detected keypoints.

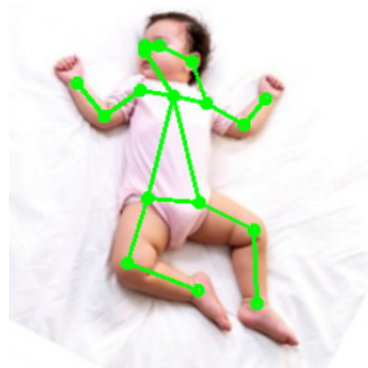


To achieve our objective of this project here, the nose of the baby is detected from the image to decide whether the face is covered due to sleeping on the stomach or for other reasons. To detect a blanket removed, the visibility of the lower body parts such as the hip, knee, and ankle are detected. This is done by the TRT pose estimation model by Nvidia. The features are fed into two-branch multi-stage transposed convolution networks. These branch networks then simultaneously predict the heatmap and Part Affinity Field (PAF) matrices. Finally, a collection of human sets is found, where each human is a set of parts, and where each part contains its relative information. And after the detection and running our algorithms on it, we can now get the alert of the situation whenever the baby is in a vulnerable state. Whether the baby having its blanket removed or the baby sleeping on its stomach, the caretaker can now get notified of it

without monitoring the infant 24x7. Thus, the baby remains in a safe state preventing any accidents from happening.



(a)



(b)

(c)

(d)

(a)Baby with blanket on (b)Baby with blanket off (c)Baby sleeping on the back, nose not detected (d)Baby sleeping pose estimated.

AWAKE DETECTION :

The suggested method for detecting whether the baby is awake or not operates on two levels. The procedure begins with the camera recording live video frames, which are

then transferred to a local server. The Dlib library is utilized on the server to identify facial landmarks, and a threshold value is used to determine whether or not the baby is awake. The EAR (Eye Aspect Ratio) is then computed using these face landmarks and given to the driver. The EAR value obtained at the application's end is compared to a threshold value of 0.20 in our system. The baby is regarded to be awake if the EAR value is greater than the threshold value. An alert message will be sent to the parent notifying them that the baby is awake and needs assistance.
For awake detection we have used OpenCV and Python. The Dlib library is used to detect and isolate the facial landmarks using Dlib pre- trained facial landmark detectors. In this approach , 68 facial landmarks have been used.

Facial Landmark marking -
Dlib library is imported and used for the extraction of facial landmarks. Adlib uses a pre-trained face detector , that is an improvement of the histogram of oriented gradients. It consists of two shape predictor models trained on the i-Bug 300-W dataset ,that each localize 68 and 5 landmark points respectively within a face image .We have used the 68 facial landmarks.
In this method, frequencies of gradient direction of an image in localized regions are used to form histograms. It is especially suitable for face detection; it can describe contour and edge features exceptionally in various objects.
For recording the Facial Landmarks, the Facial Landmark Predictor was used by the system to calculate lengths for the EAR values.

The Facial landmark points of the Dlib library which are used to compute EAR are represented in the figure below :

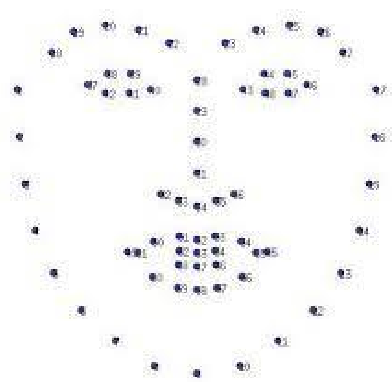


Fig (a) : Facial Landmarks

Algorithm used -

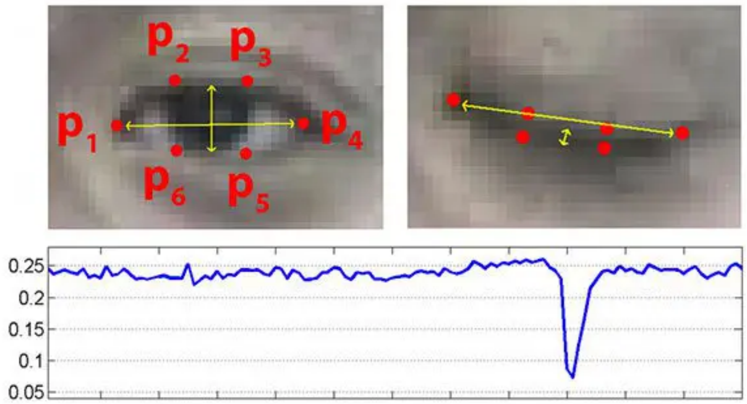


Fig (b) : Eye Aspect Ratio

Here P1,P2,P3,P4,P5,P6 are the pupil coordinates.EAR is generally a constant when eyes are open and is near about 0.5- 0.10.When EAR is greater than 0.20. It is concluded that the baby is awake.

Eye Aspect Ratio(EAR) is the calculated for both the eyes,

$$\frac{(|P2 - P6| + |P3 - P5|)}{2(|P1 - P4|)}$$

The numerator determines the distance between the upper and lower eyelids using equation. The horizontal distance of the eye is represented by the denominator. EAR values are utilized to identify whether the baby is awake or asleep in this framework. The average of the EAR values of the left and right eyes is obtained. The EAR is monitored using a counter that sees whether the value falls exceeds the threshold value in the following frame . The infant has closed its eyes and is sleepy,as indicated by the aforementioned circumstances.In contrast, if the EAR values does not maintains the higher value for a following few frames, it means that the baby went back to sleep after a few seconds.The block design of our suggested technique to detect whether the baby is awake or asleep is shown in Figure 3 (Block Diagram).

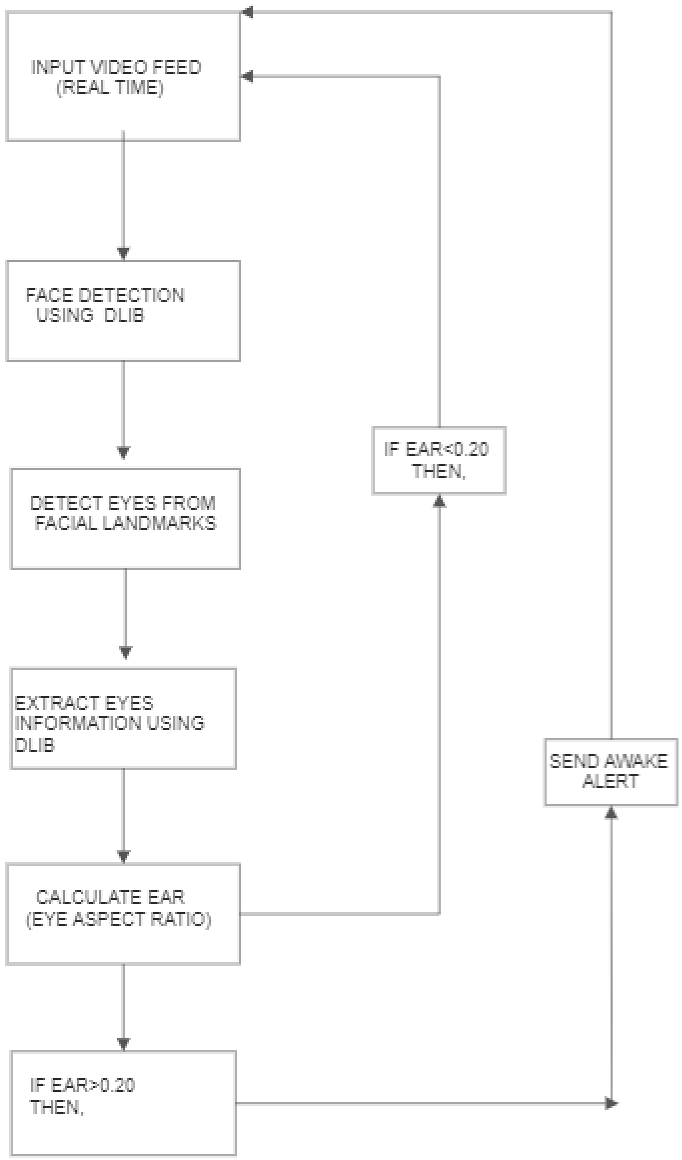


Fig (c) : Block Diagram - Awake Detection

For this project we have utilized a webcam associated with the Jetson Nano .The framework was tried for different samples in various light surrounding lighting conditions.

Eye Detection Accuracy =

$$\frac{\text{Total no. of times eye detected}}{\text{(Total no. of time eye detected + total no. of times eye not detected)}}$$

Awake Detection accuracy =

$$\frac{\text{Total no. of times alert message was displayed}}{\text{(Total no. of times alert message was displayed +Total no. of times alert messages was not sent)}}$$

Result Table for accuracy

Serial No:	Eye Detection accuracy	Awake Detection accuracy
Sample 1	82.34%	80.53%
Sample 2	81%	64.5%
Sample 3	73%	82.34%
Sample 4	77%	65.67%
Sample 5	89.5%	100%
Total	80.56%	78.60%

After training and testing the model rigorously we attained an accuracy of about 78.60%.

Obtained results -

Fig (a) Sample Image

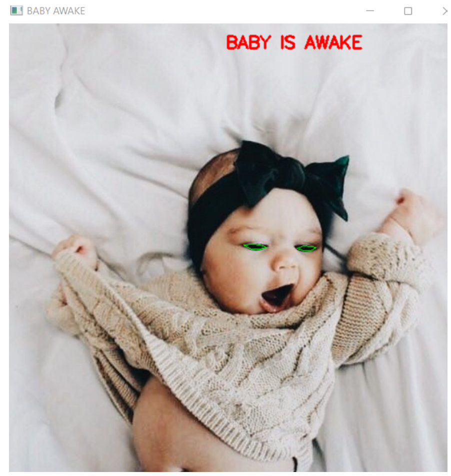


Fig (b) Output Image

We also faced certain limitation mentioned as follows -

- The OpenCV can detect live images up to only an approximate distance up to 25 cm from the Webcam to face. So, the placement of the webcam should be such that this limitation does not affect the model accuracy.
- With poor lighting conditions occasionally, the system is unfit to perceive the eyes. So, it gives a wrong result which must be managed.
- If the orientation of the face is tilted to a specific degree it will in general be perceived , anyway past this framework the framework is unable to identify the face of the baby.

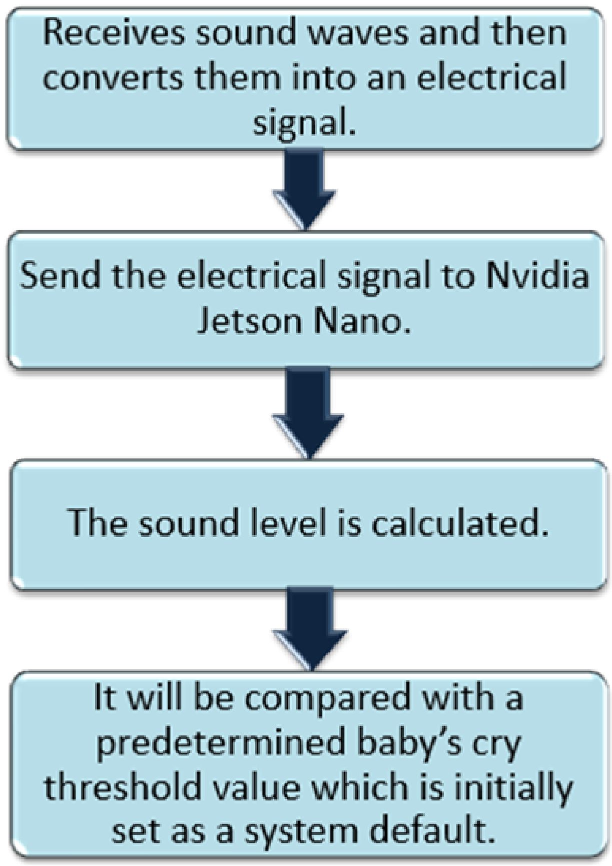
CRYING DETECTION:

An objective, automated and unobtrusive method to quantify crying behavior in an at-home and clinical setting may improve the diagnostic process in excessively crying infants, allow for objective determination of treatment effects by physicians, and enable researchers to include objectively determined cry duration as digital biomarker in clinical trials. Given the importance for researchers to study the relationship between an infant's crying patterns and their health, automatic detection and quantification of infant cries from an audio signal is an essential step in remote baby monitoring applications .

Automatic cry detection has been reported in the form of remote baby monitors for non-intrusive clinical assessments of infants in hospital settings, and several researchers have shown that classification of cry- and non-cry-sounds is possible with machine-learning algorithms.

The aim is to automatically recognize a baby crying while sleeping. In such a case, a soothing sound is played to calm the baby down. This is done by implementing a machine learning algorithm on Jetson Nano.The idea is to train a model on a computer and to deploy it on Jetson Nano,which is used to record a signal and use the model to predict if it is a baby cry or not. In the former case a soothing sound,the caretaker is also notified about the same, in the latter the process(recording and predicting steps) starts again.

A review of infant cry analysis and classification:



Training of the model -

It includes all the steps required to train a machine learning model. First, it reads the data, it performs feature engineering and it trains the model. The model is saved to be used in the prediction step. The _training step_ is performed on a powerful machine, such as our personal computer.

Simulation of the model -

There is a script to test the prediction step on our computer before deployment on Jetson Nano.

The trained model was then deployed from my personal computer to Jetson Nano using FlieClient@Zilla having a suitable Wi-Fi connection and IP address of the Jetson Nano. The trained model is named “model.pkl”.

After the deployment, “model.pkl” is opened on Jetson Nano where the prediction of the model is done.

Prediction -

It includes all the steps needed to make a prediction on a new signal.It reads a new signal, it divides it into 5 overlapping signals, it applies the pipeline saved from the training step to make the prediction.The prediction is performed on Jetson Nano.

CONCLUSION:

We proposed in this project an intelligent system that allows parents to monitor a baby from a web application either on a local network or on a remote network. The main advantage of this system is that it is more user friendly and no harm to the baby. This system is designed using the Nvidia Jetson Nano module which has huge advantages over other embedded systems. It can be applicable for the home environment as well as in the hospital or baby nursing care.

Monitoring system

FUTURE WORKS

- Our first focus is to develop a mobile application so that we can directly interact with caretakers and notify them whenever the baby needs their attention.
- To improve our pose estimation model and revise our algorithm for obtaining better results.
- To try other cry detection methods for better results.
- We also aim on performing frequent moving detection
- Use sentiment analysis by using machine learning models trained with baby's expression for example: if he is making a particularly suckling expression in sleep which would mean he is hungry so notification of hunger would be sent to parents. This will be done through audio and video capturing.

REFERENCES

- https://github.com/NVIDIA-AI-IOT/deepstream_pose_estimation
- https://developer.nvidia.com/blog/creating-a-human-pose-estimation-application-with-deepstream-sdk/
- https://github.com/NVIDIA-AI-IOT/trt_pose
- Tareq Khan, Article An Intelligent Baby Monitor with Automatic Sleeping Posture Detection and Notification, School of Engineering, Eastern Michigan University, Ypsilanti, MI 48197, USA; tareq.khan@emich.edu
- https://pyimagesearch.com/2017/05/08/drowsiness-detection-opencv/
- Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh, Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, The Robotics Institute, Carnegie Mellon University
- https://www.researchgate.net/publication/336989127_Baby_Cry_Detection_Deep_Learning_and_Classical_Approaches
- https://www.frontiersin.org/articles/10.3389/fped.2021.651356/full
- http://dlib.net/ml.html
- Jose Gomes da Silva Neto, Joao Marcelo Xavier Natario Teixeira, Veronica Teichrieb, Analyzing embedded pose estimation solutions for human behavior understanding

UNCOVERING THE BUILDING BLOCKS OF TURBULENCE & MODELING UNSTEADY AERODYNAMIC SYSTEM

FAHIM YOUSUF CHOUDHURY, KOUSTAV MONDAL, ANIMESH ALOK PRABHAKAR, SHUBHANKAR SINHA & KESHAW THAKUR

MENTOR: SCOTT DAWSON

FINAL GROUP REPORT

Introduction

The main goal of this research work is to minimize the computational complexity and cost that goes into representing various fluid mechanics and aerodynamic systems in a low dimensional form in order to understand and analyse them better.

Methodology

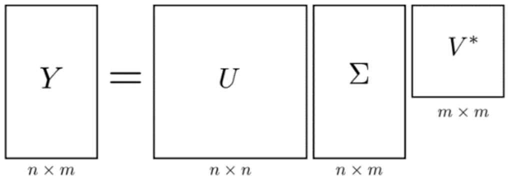
The datasets that we are dealing with are mostly functions of both space and time. In order to understand pertinent features of such datasets as well as to fulfil our goal of building a reduced-complexity model, our first step is to separate the variables using Singular Value Decomposition (SVD) and then using various curve fitting tools in MATLAB and PYTHON we create an analytical function that fits our dataset.

The decomposition methods presented in this report are founded on the singular value decompositions of matrices or operators.

Singular Value Decomposition

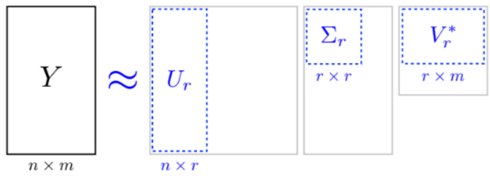
$$y(x,t) = \sum_{j=1}^{m} u_j(x)a_j(t)$$

In order to achieve this decomposition, we use SVD. The main idea of this is to break the matrix containing spatial and time coordinates into two separate vectors i.e.,



U and ***V*** are unitary matrices and $\sum$ is diagonal matrix with decreasing and nonnegative diagonal entries.

We truncate the above matrices in order to eliminate noise and we get the following,



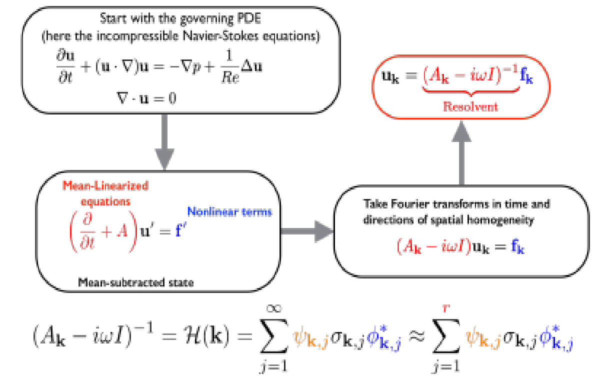
We then solve this expression and obtain

$$\begin{aligned} Y_r &= U_r \Sigma_r V_r^* \\ &= \sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^* \end{aligned}$$

Here uj and vj are vectors that are the j-th columns of Ur and Vr.

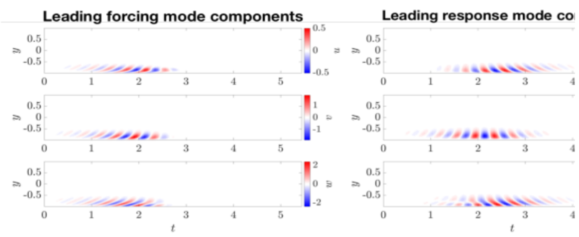
We use predicted coherent structures from a method called resolvent analysis, which looks at a singular value decomposition of a linear operator obtained from the governing Navier-Stokes equations

FINAL GROUP REPORT



The leading left and right singular vectors give the optimal response and forcing modes of the system.

Sample space-time modes in turbulent channel flow



Curve Fitting

The absolute and real components of u_j *and* v_j matrices are fitted using gaussian functions with the help of **curve_fit** function in Python and MATLAB.

The general form of the gaussian function assumed for fitting the components is,

$$\psi(y,t) = c\exp\left[ia_y y - b_y(y-y_0)^2\right]\exp\left[ia_t t - b_t(t -\right.$$

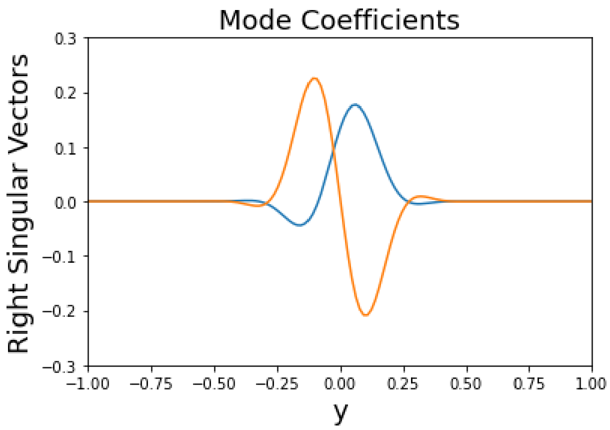
The unknown parameters of these functions are determined using curve fitting tools. For a better fit we enforce additional boundary conditions by adding additional image wavepackets.

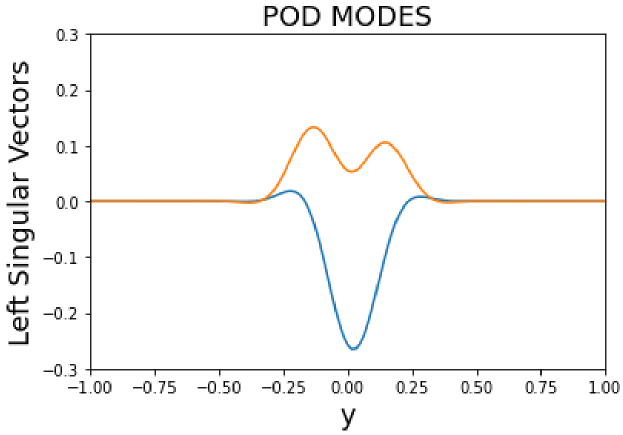
Results

Using the method discussed in the previous section, we have succeeded in creating analytical functions for various datasets. The results are shown and discussed below –

For TestMatrix –

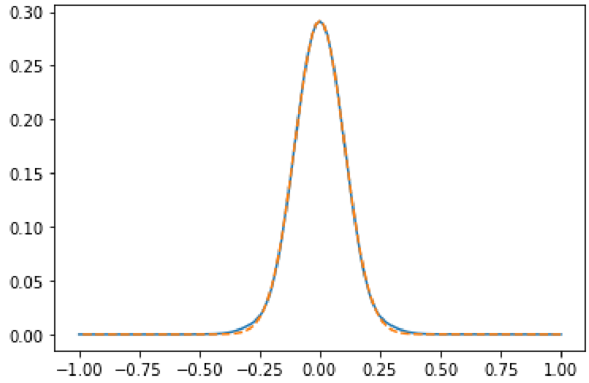
For our first dataset, we extracted the left and right singular vectors using built-in SVD tools in Matlab and python.



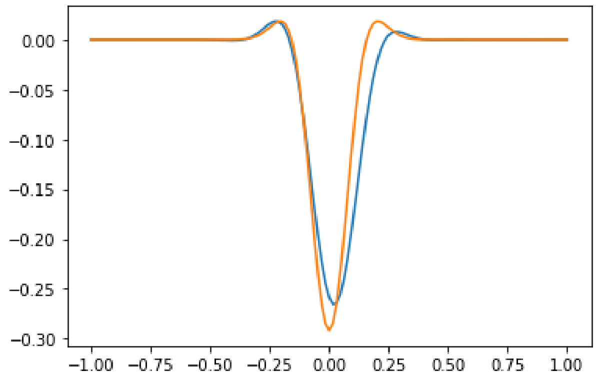


Now since we have separated space and time variables, we are in a position to fit them. Left Singular Vector is a function of space while Right Singular Vector is a function of time. For our first dataset, we fitted the absolute value of the above

vectors using one gaussian and the results were-

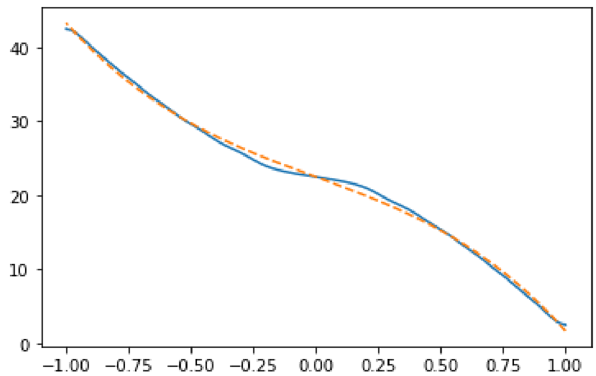


Fit of Left Singular Vector



Fit of Right Singular Vector

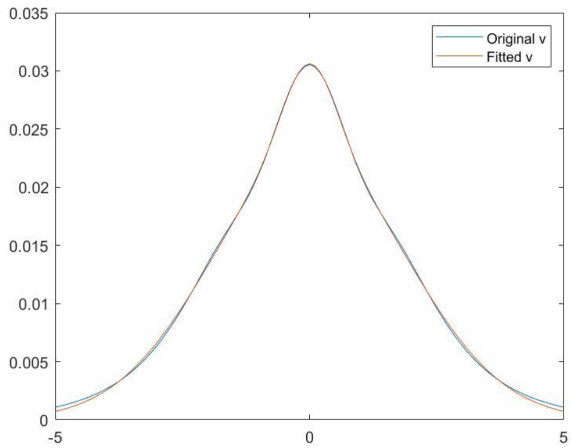
We also looked at the phase of the Left Singular Vector and fitted it.



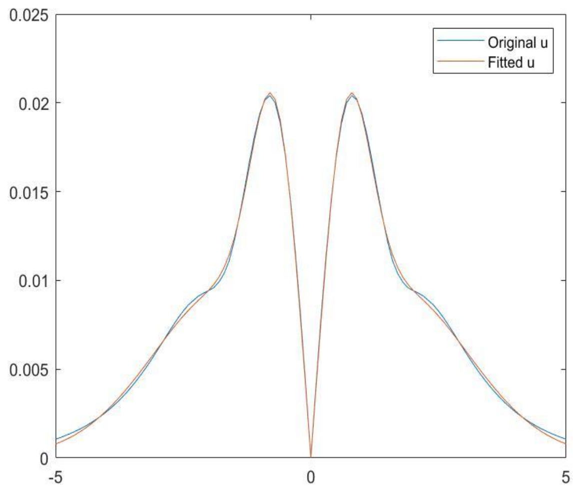
Now that we also have the phase, we can fit the real part of the vectors also. This framework has been used by us for the other datasets also.

For Flow over a Cylinder Data –

Here, we are going over some real world data and using the same analysis as before -



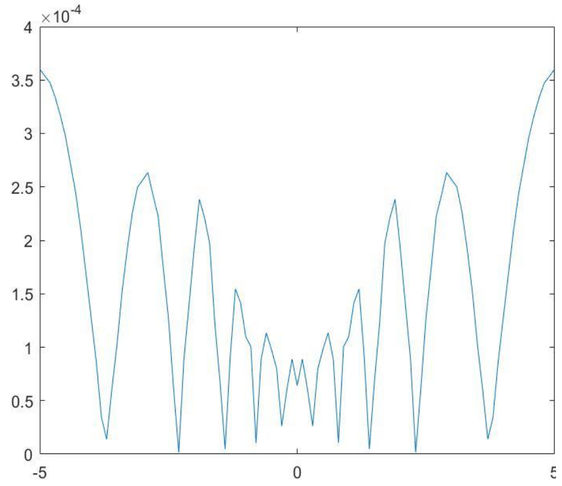
This fit was achieved using two gaussians, one for the inner region and the other accounting for the outer.



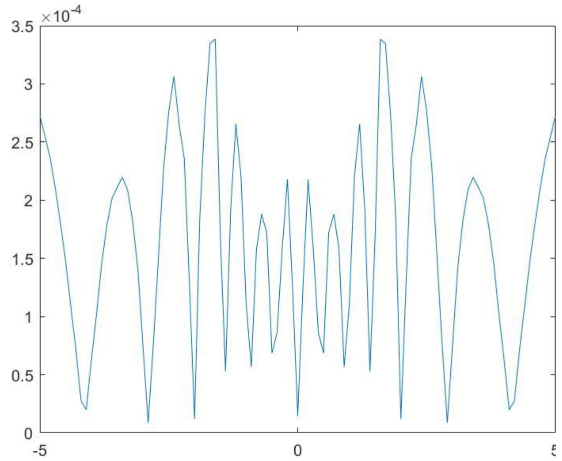
The above graph was obtained using 4 gaussians paired with a starting points guess as well as upper and lower bounds. These two graphs represent the u and v component of velocity of flow over a cylinder.

FINAL GROUP REPORT

The error in both these cases is shown by the following graphs-



Error for v component of velocity

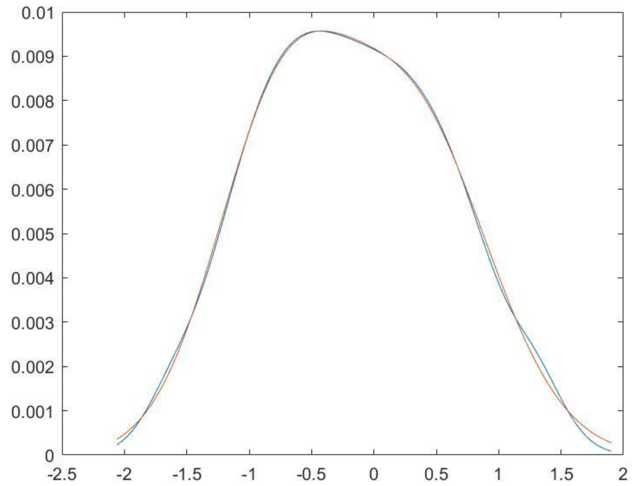


Error for u component of velocity

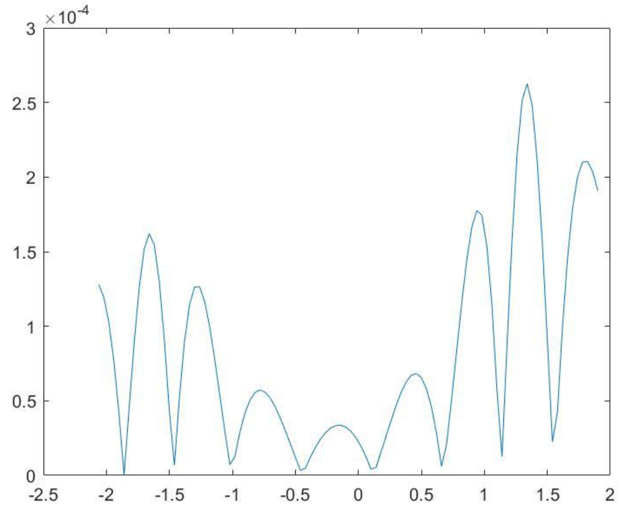
As we can clearly see, the error is negligible. That means our analytical function works and can be used for studying the dataset. Similarly, we were able to successfully create functions for other datasets also beginning with airfoil data.

For Flow over an Airfoil Data -

For this first plot, only 2 gaussians were required to achieve the necessary fit.



Velocity fitting graph



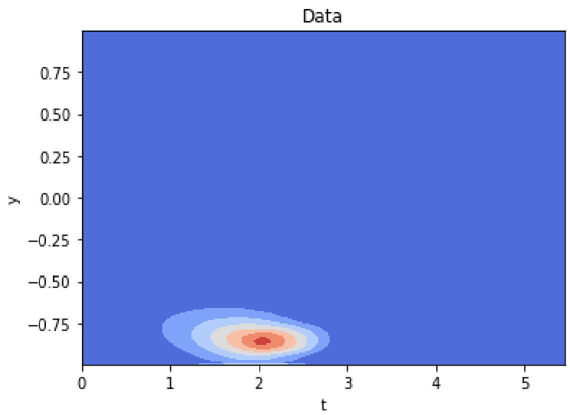
Error plot

As we can see, the error in this case is also negligible which is a good thing as our analysis has now worked on two sets of real world data. We will now look at some other data to test our theory even further.

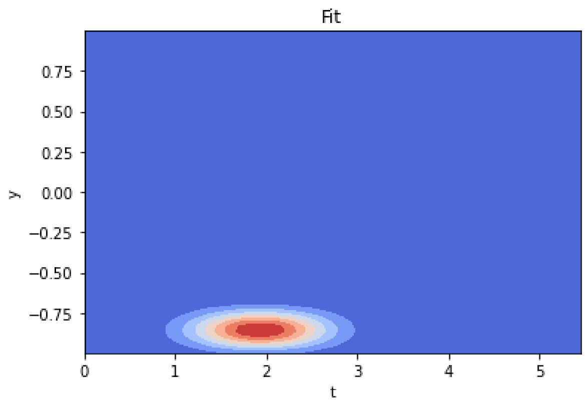
FINAL GROUP REPORT

For DataModes –

The below graph is a contour graph of the absolute value of PhiU. The second graph is obtained after finishing our fitting in time with one gaussian and two gaussian plus boundary conditions for space. As we can see, they match to a great extent.

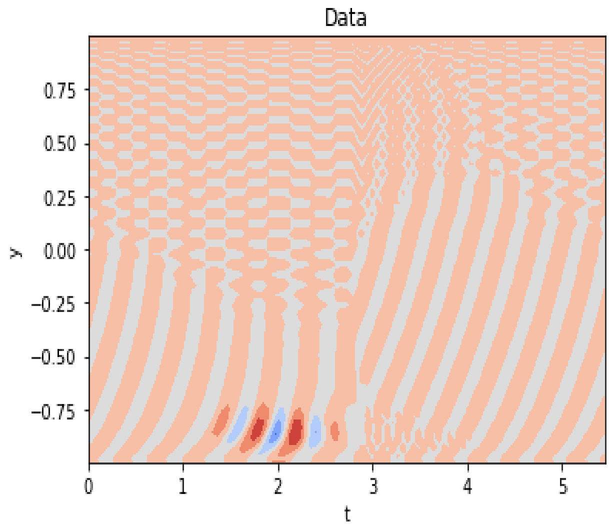


Original abs PhiU velocity

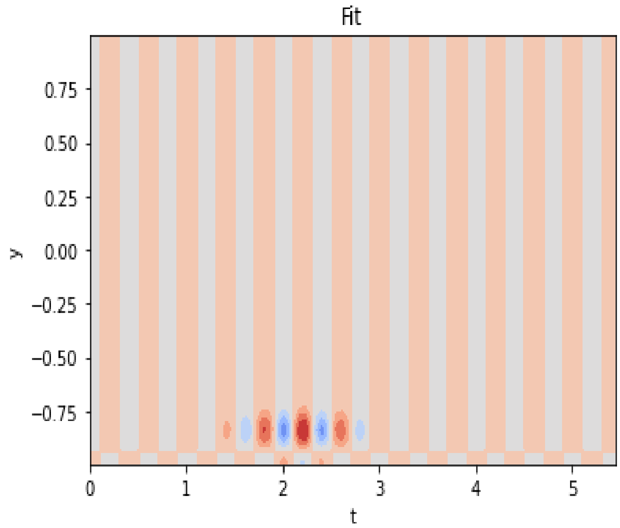


Fitted abs PhiU velocity

The next two graphs are of the real part of PhiU where we have extracted the phase and then fit a linear function to it using our fitting tools and combined it with our absolute valued function to get the fit of the real part. The graph matches the original in the x-direction but deviates a little in the y due to the phase.



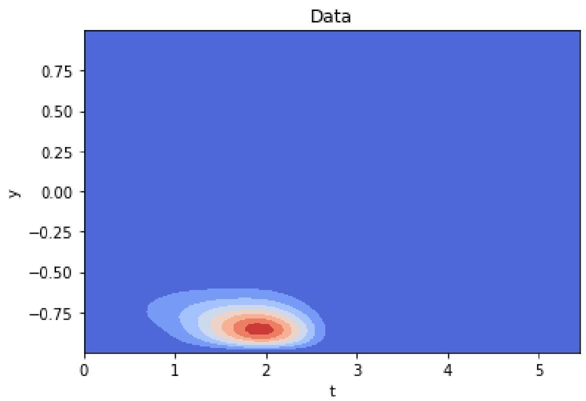
Original real PhiU velocity



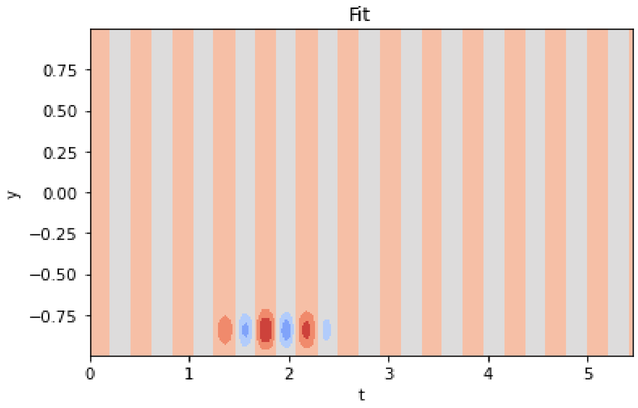
Fitted real PhiU velocity

The same trend continues for PhiV, PsiU, PsiV and PsiW wherein we select that number of gaussian which reduces error and if necessary boundary conditions as shown by the following plots in the next few pages -

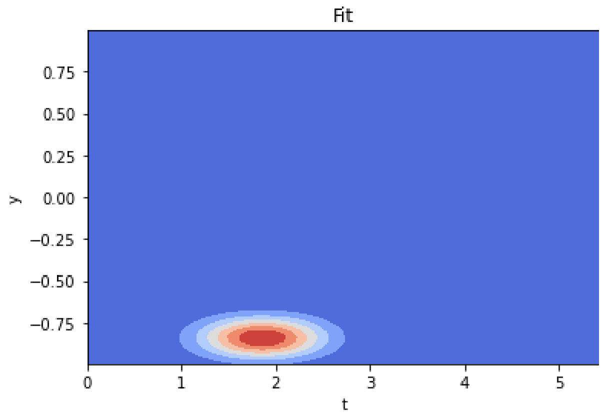
FINAL GROUP REPORT



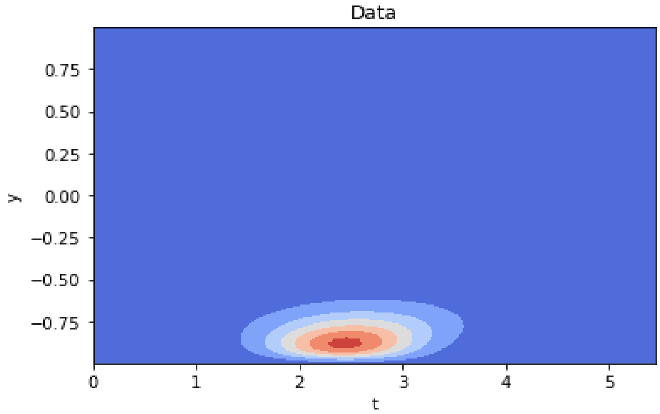
Original abs PhiV velocity



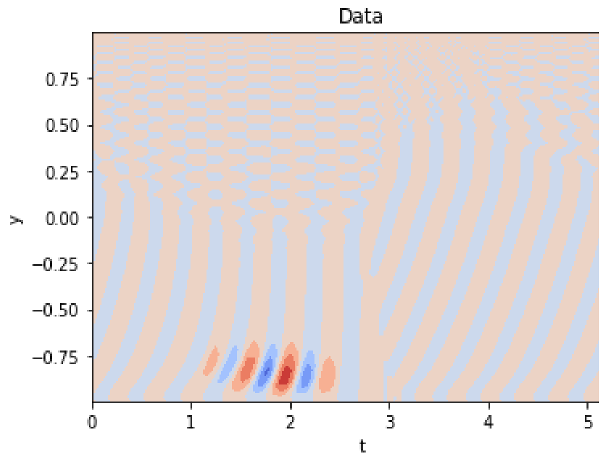
Fitted real PhiV velocity



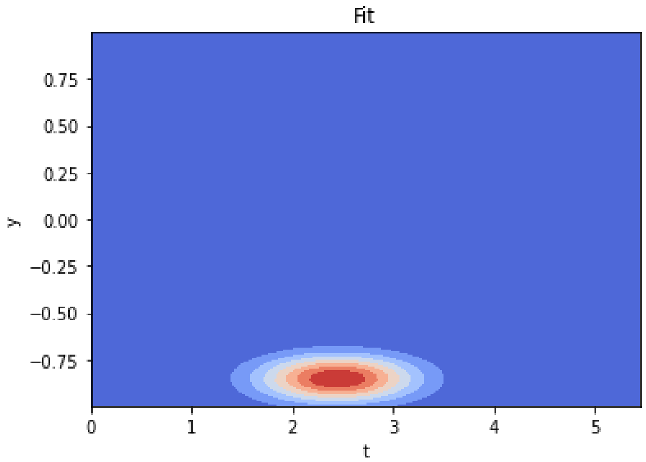
Fitted abs PhiV velocity



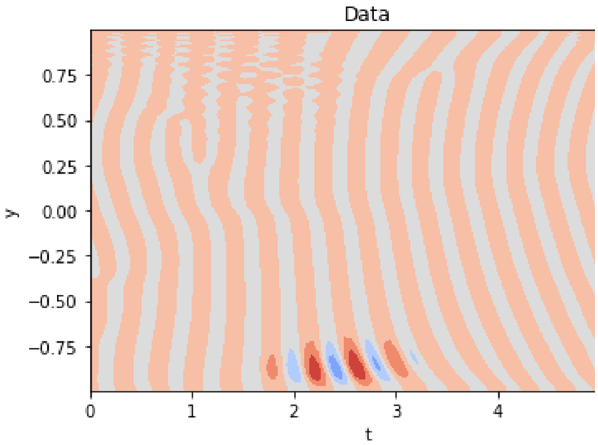
Original abs PsiU velocity



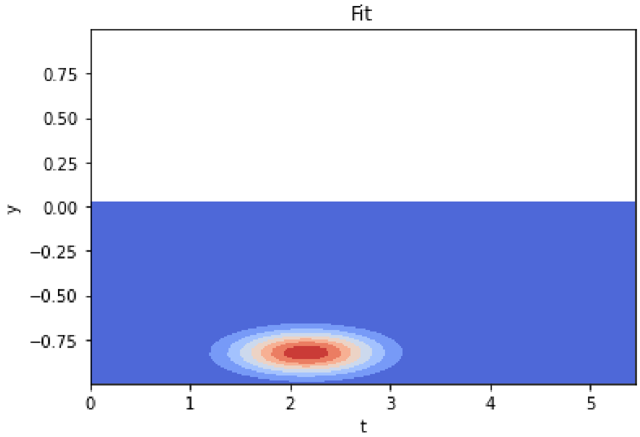
Original real PhiV velocity



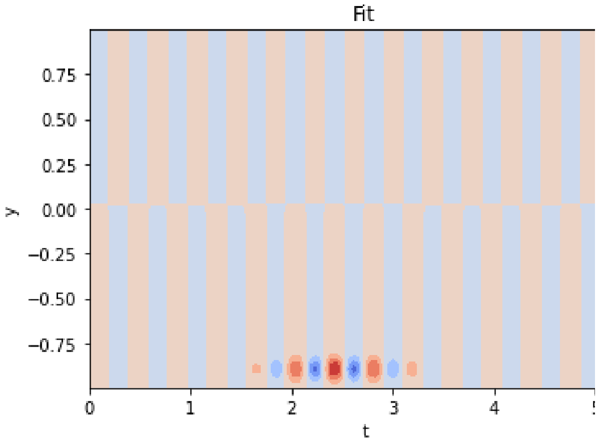
Fitted abs PsiU velocity



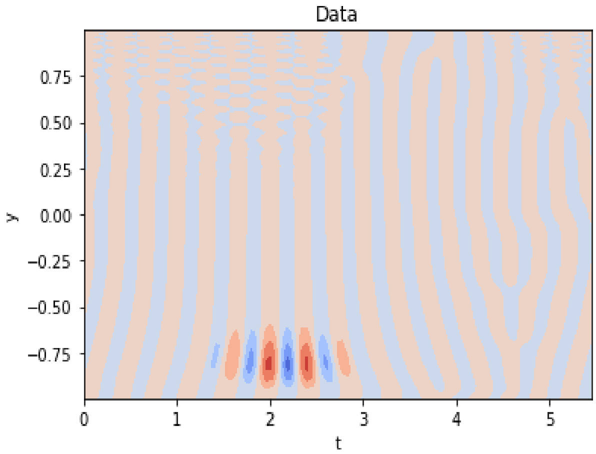
Original real PsiU velocity



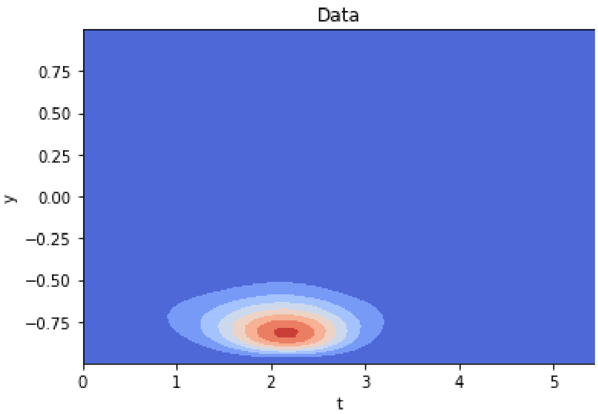
Fitted abs PsiV velocity



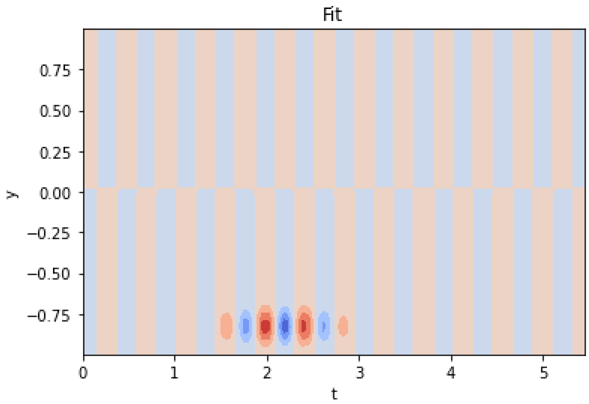
Fitted real PsiU velocity



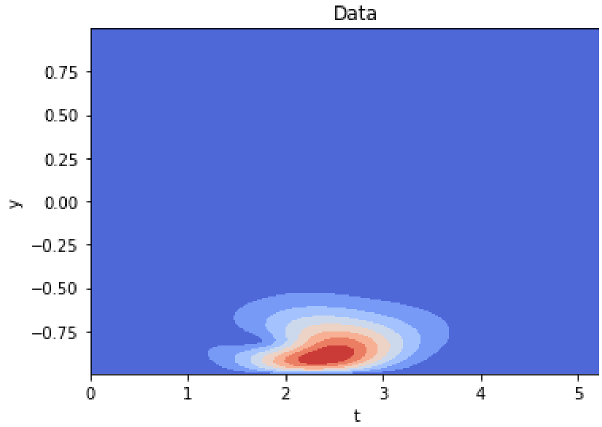
Original real PsiV velocity



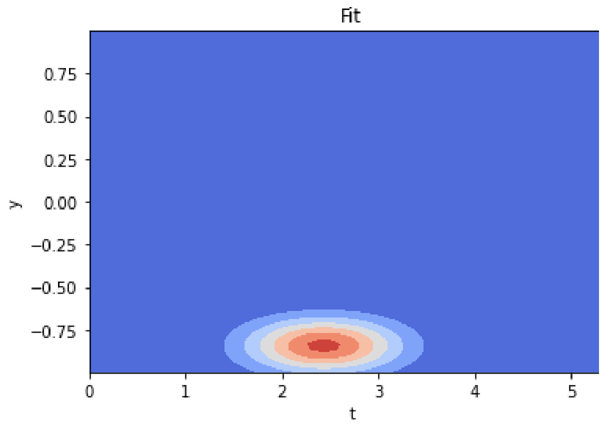
Original abs PsiV velocity



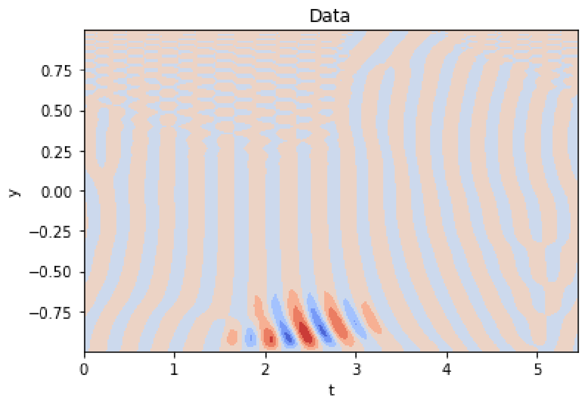
Fitted real PsiV velocity



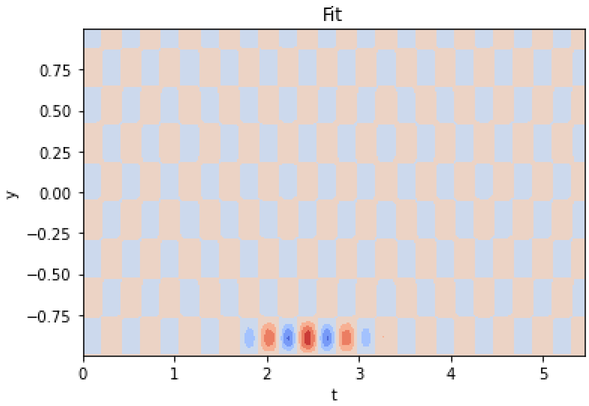
Original abs PsiW velocity



Fitted abs PsiW velocity



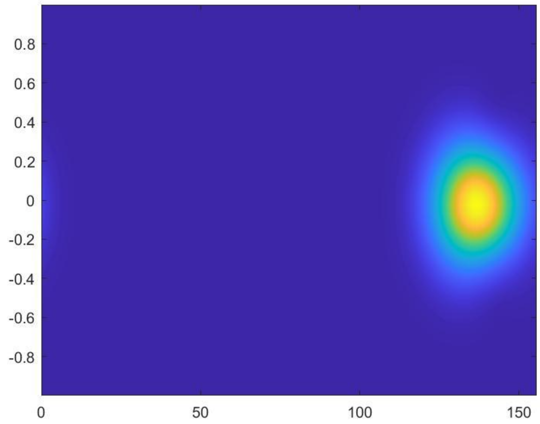
Original real PsiW velocity



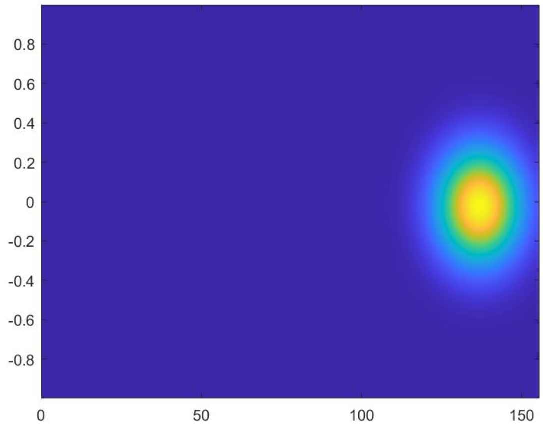
Fitted real PsiW velocity

For DataModesV3-

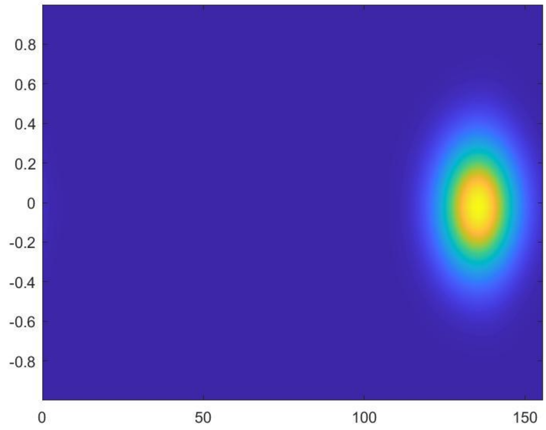
As evident by the name, DataModesV3 is an updated dataset of our previous DataModes. We have once again used our framework to create functions that represent the original data to a great extent as evident by the following plots-



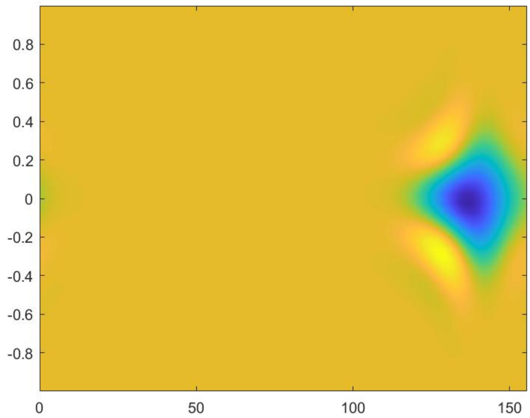
Original abs PhiU velocity



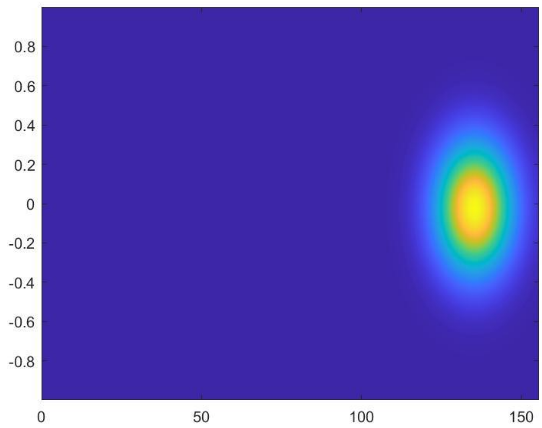
Fitted abs PhiU velocity



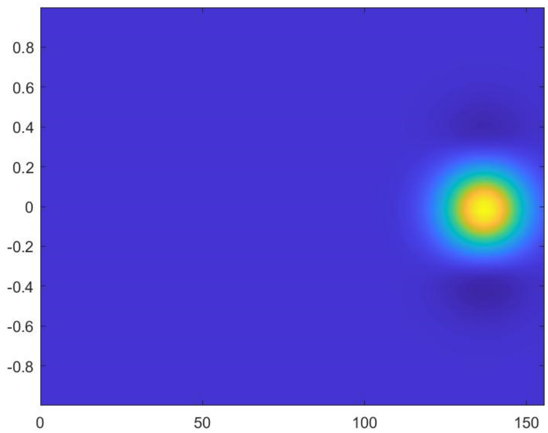
Original abs PhiV velocity



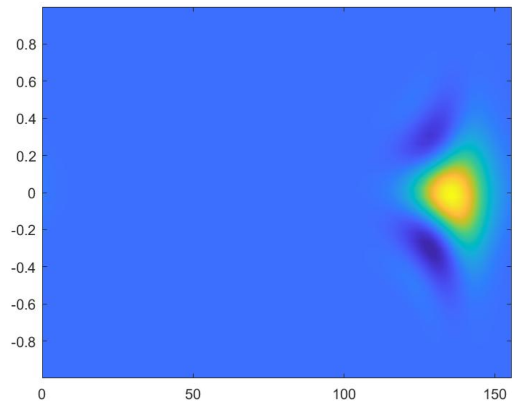
Original real PhiU velocity



Fitted abs PhiV velocity

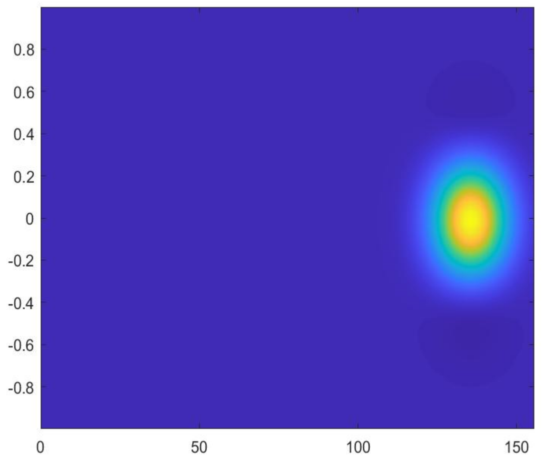


Fitted real PhiU velocity



Original real PhiV velocity

FINAL GROUP REPORT



Fitted real PhiV velocity

Conclusion –

After performing our analysis on a variety of data including both real world and theoretical and looking at the results that we have obtained, it is clear that our theory works and error is also minimal. But this theory does not provide us with a straight forward path as we have seen that in many cases, we had to select more than one gaussian sometimes even four and on top of that we also had to use boundary conditions and start point guesses sometimes in order to achieve a better fit. But our analysis also has its flaws as when it came to the real part of a function in DataModes, we were not able to get a great fit perhaps due to the phase not matching to a great extent or some other underlying issue. But apart from this, our error was less than 1% in flow over a cylinder, air-foil data, DataModesV3 and TestMatrix dataset.

Further Work –

In our work, we have done our analysis mostly on datasets that resemble a gaussian function in some form or the other. Further work may be done to expand our theory to cover a wide range of functions and eliminate some of the issues we encountered in our present work.

Acknowledgement –

We would like to thank you sir for giving us the opportunity to do this research which allowed us to hone our skills in Matlab and python and we also got to know about the basics of research work for which we are eternally grateful. We would also like to wish you good health and success in your future work.

UNCOVERING THE BUILDING BLOCKS OF TURBULENCE & MODELING UNSTEADY AERODYNAMIC SYSTEM

FAHIM YOUSUF CHOUDHURY, KOUSTAV MONDAL, ANIMESH ALOK PRABHAKAR, SHUBHANKAR SINHA & KESHAW THAKUR

MENTOR: SCOTT DAWSON

FINAL GROUP REPORT

Introduction

The main goal of this research work is to minimize the computational complexity and cost that goes into representing various fluid mechanics and aerodynamic systems in a low dimensional form in order to understand and analyse them better.

Methodology

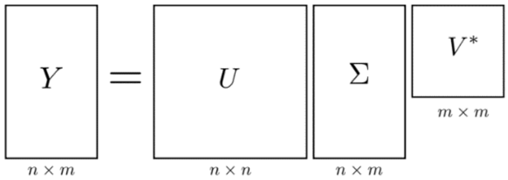
The datasets that we are dealing with are mostly functions of both space and time. In order to understand pertinent features of such datasets as well as to fulfil our goal of building a reduced-complexity model, our first step is to separate the variables using Singular Value Decomposition (SVD) and then using various curve fitting tools in MATLAB and PYTHON we create an analytical function that fits our dataset.

The decomposition methods presented in this report are founded on the singular value decompositions of matrices or operators.

Singular Value Decomposition

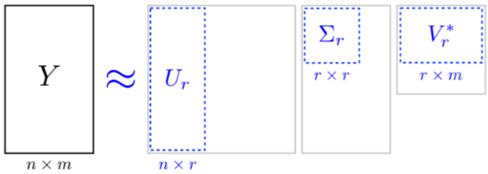
$$y(x,t) = \sum_{j=1}^{m} u_j(x) a_j(t)$$

In order to achieve this decomposition, we use SVD. The main idea of this is to break the matrix containing spatial and time coordinates into two separate vectors i.e.,



U and ***V*** are unitary matrices and $\sum$ is diagonal matrix with decreasing and nonnegative diagonal entries.

We truncate the above matrices in order to eliminate noise and we get the following,



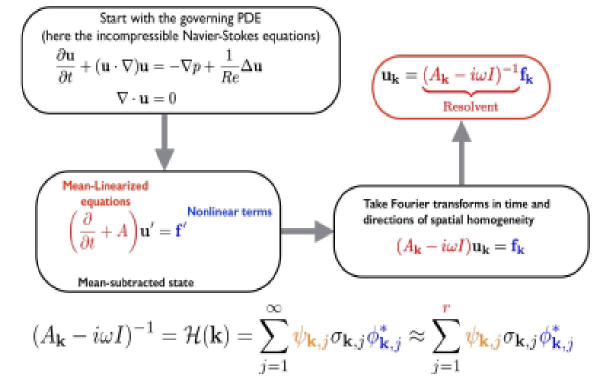
We then solve this expression and obtain

$$Y_r = U_r \Sigma_r V_r^* = \sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^*$$

Here uj and vj are vectors that are the j-th columns of Ur and Vr.

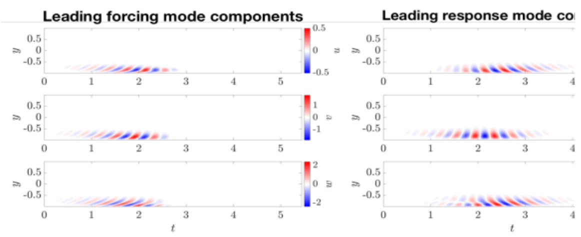
We use predicted coherent structures from a method called resolvent analysis, which looks at a singular value decomposition of a linear operator obtained from the governing Navier-Stokes equations

FINAL GROUP REPORT



The leading left and right singular vectors give the optimal response and forcing modes of the system.

Sample space-time modes in turbulent channel flow



Curve Fitting

The absolute and real components of u_j *and* v_j matrices are fitted using gaussian functions with the help of **curve_fit** function in Python and MATLAB.

The general form of the gaussian function assumed for fitting the components is,

$$\psi(y,t) = c\exp\left[ia_y y - b_y(y-y_0)^2\right]\exp\left[ia_t t - b_t(t -\right.$$

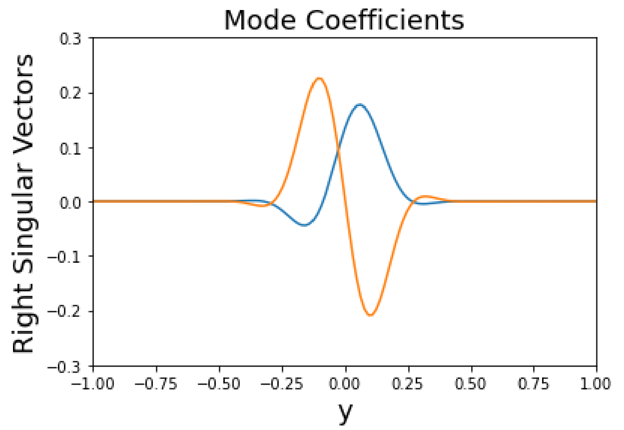
The unknown parameters of these functions are determined using curve fitting tools. For a better fit we enforce additional boundary conditions by adding additional image wavepackets.

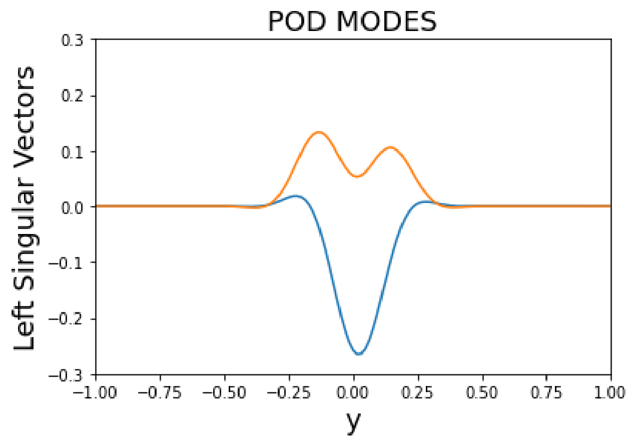
Results

Using the method discussed in the previous section, we have succeeded in creating analytical functions for various datasets. The results are shown and discussed below –

For TestMatrix –

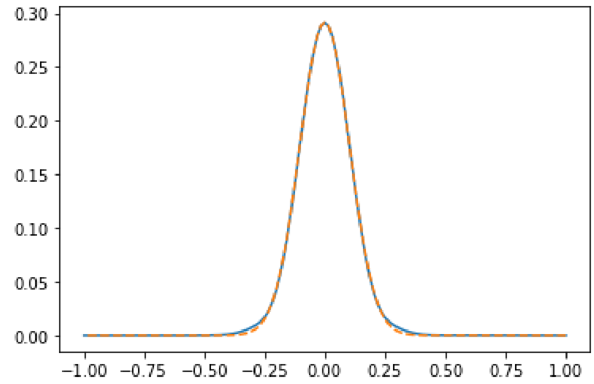
For our first dataset, we extracted the left and right singular vectors using built-in SVD tools in Matlab and python.



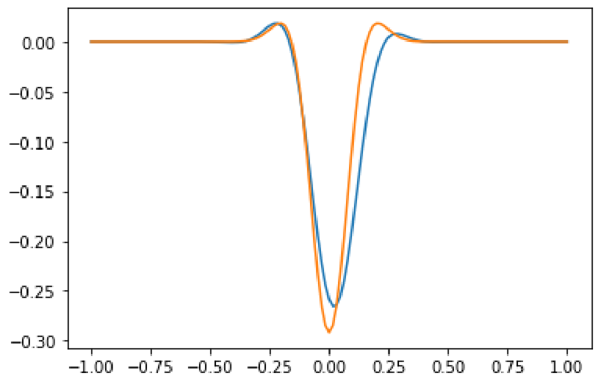


Now since we have separated space and time variables, we are in a position to fit them. Left Singular Vector is a function of space while Right Singular Vector is a function of time. For our first dataset, we fitted the absolute value of the above

vectors using one gaussian and the results were-

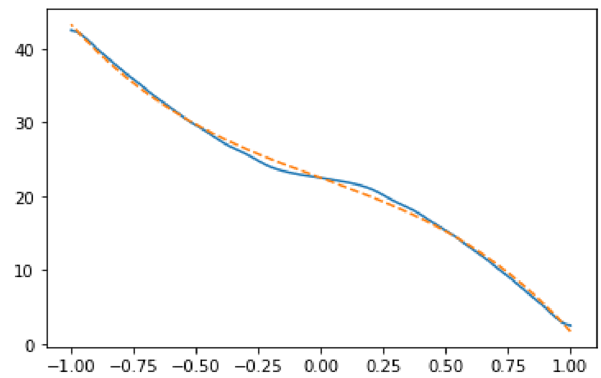


Fit of Left Singular Vector



Fit of Right Singular Vector

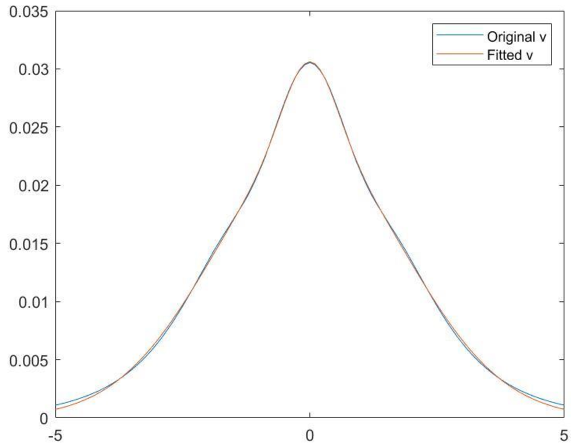
We also looked at the phase of the Left Singular Vector and fitted it.



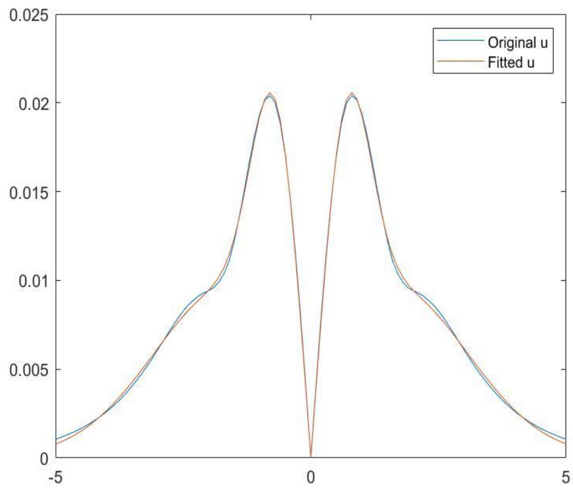
Now that we also have the phase, we can fit the real part of the vectors also. This framework has been used by us for the other datasets also.

For Flow over a Cylinder Data –

Here, we are going over some real world data and using the same analysis as before -



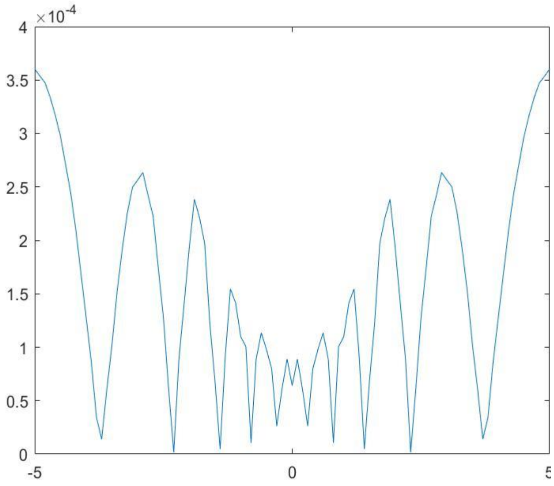
This fit was achieved using two gaussians, one for the inner region and the other accounting for the outer.



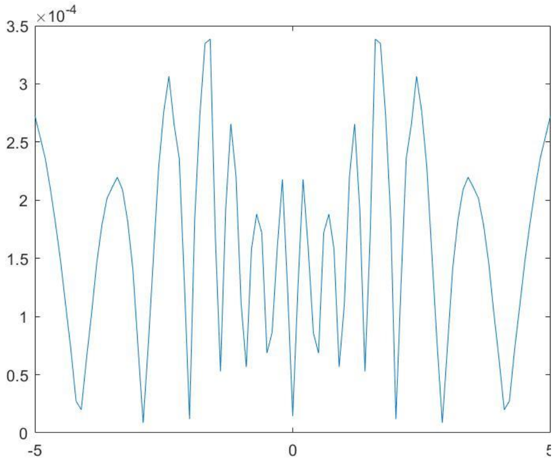
The above graph was obtained using 4 gaussians paired with a starting points guess as well as upper and lower bounds. These two graphs represent the u and v component of velocity of flow over a cylinder.

FINAL GROUP REPORT

The error in both these cases is shown by the following graphs-



Error for v component of velocity

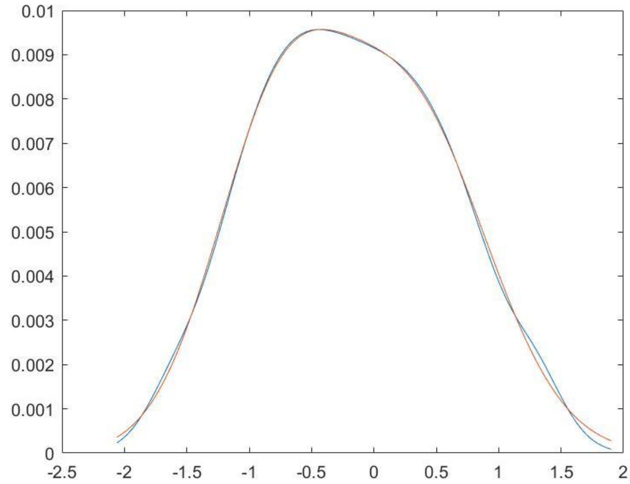


Error for u component of velocity

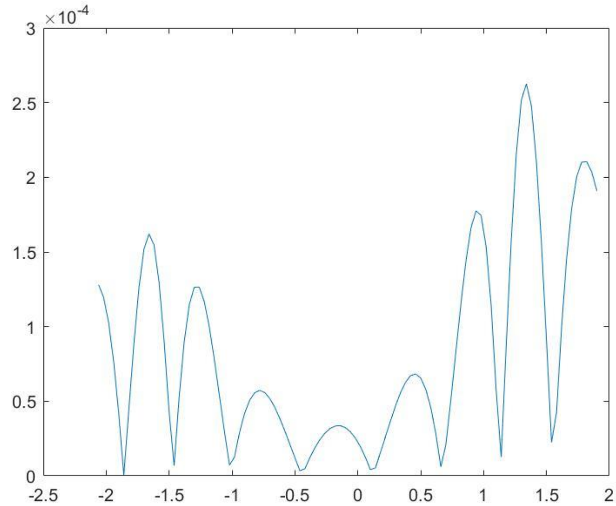
As we can clearly see, the error is negligible. That means our analytical function works and can be used for studying the dataset. Similarly, we were able to successfully create functions for other datasets also beginning with airfoil data.

For Flow over an Airfoil Data -

For this first plot, only 2 gaussians were required to achieve the necessary fit.



Velocity fitting graph



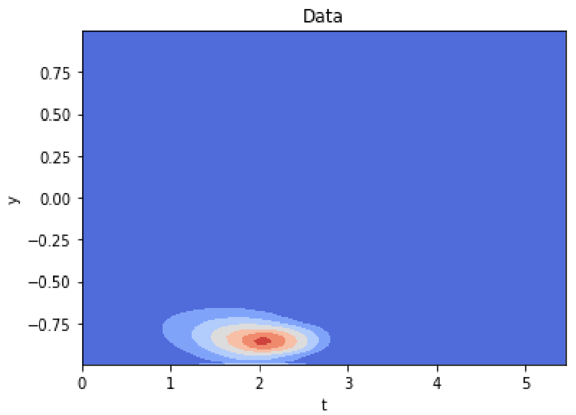
Error plot

As we can see, the error in this case is also negligible which is a good thing as our analysis has now worked on two sets of real world data. We will now look at some other data to test our theory even further.

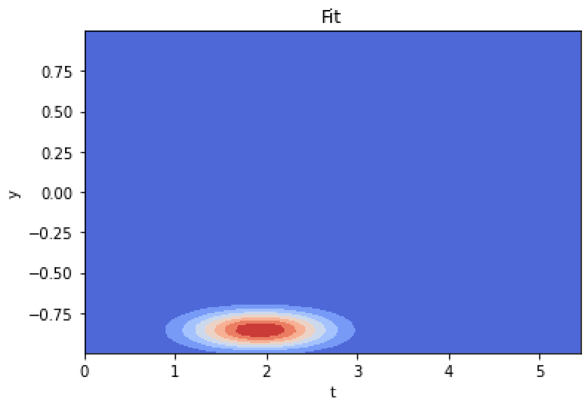
FINAL GROUP REPORT

For DataModes –

The below graph is a contour graph of the absolute value of PhiU. The second graph is obtained after finishing our fitting in time with one gaussian and two gaussian plus boundary conditions for space. As we can see, they match to a great extent.

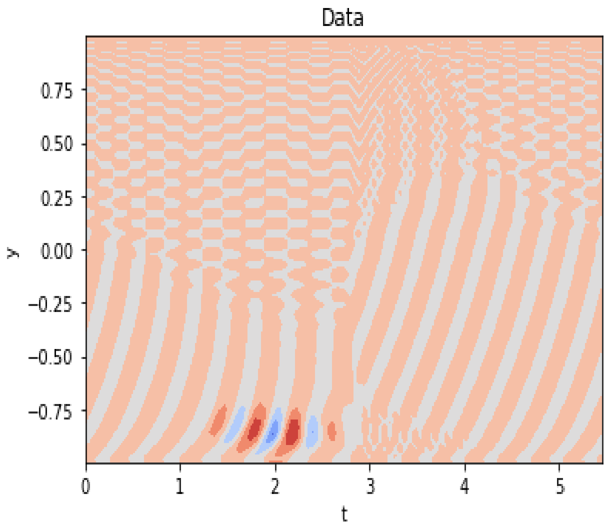


Original abs PhiU velocity

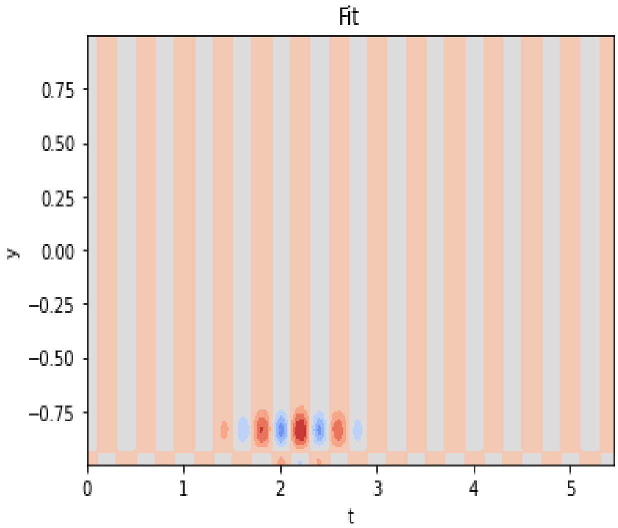


Fitted abs PhiU velocity

The next two graphs are of the real part of PhiU where we have extracted the phase and then fit a linear function to it using our fitting tools and combined it with our absolute valued function to get the fit of the real part. The graph matches the original in the x-direction but deviates a little in the y due to the phase.

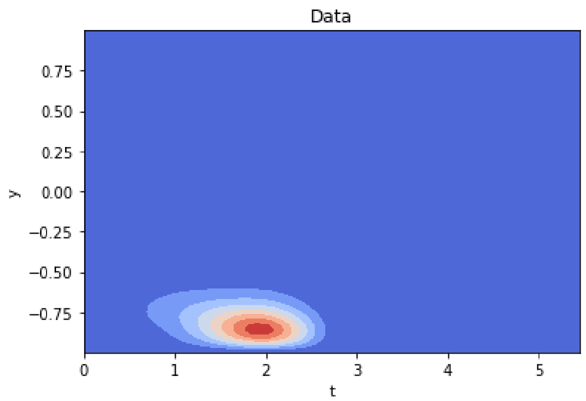


Original real PhiU velocity

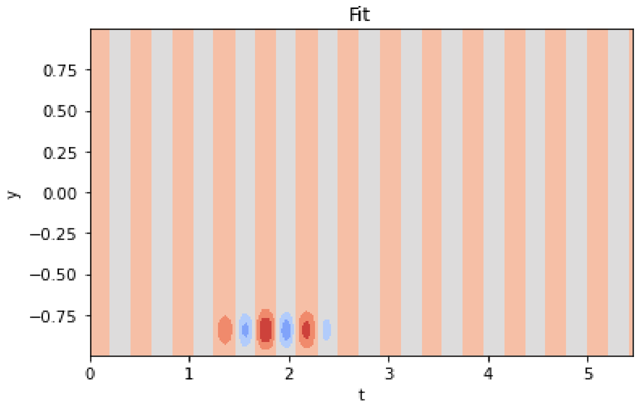


Fitted real PhiU velocity

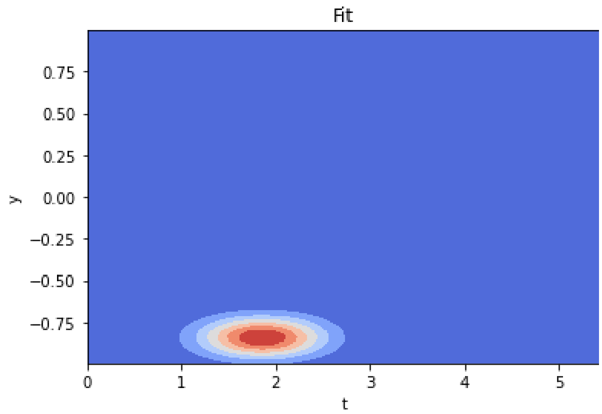
The same trend continues for PhiV, PsiU, PsiV and PsiW wherein we select that number of gaussian which reduces error and if necessary boundary conditions as shown by the following plots in the next few pages -



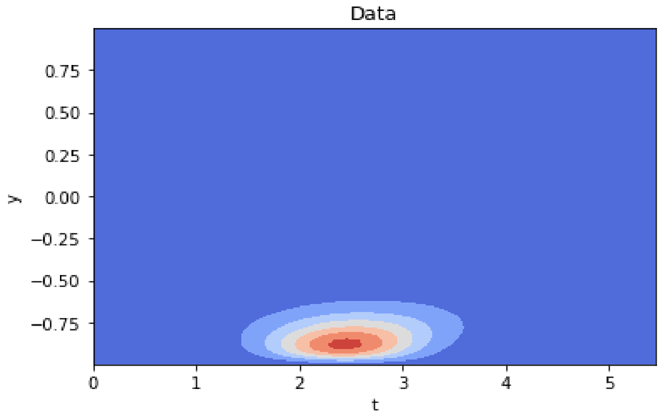
Original abs PhiV velocity



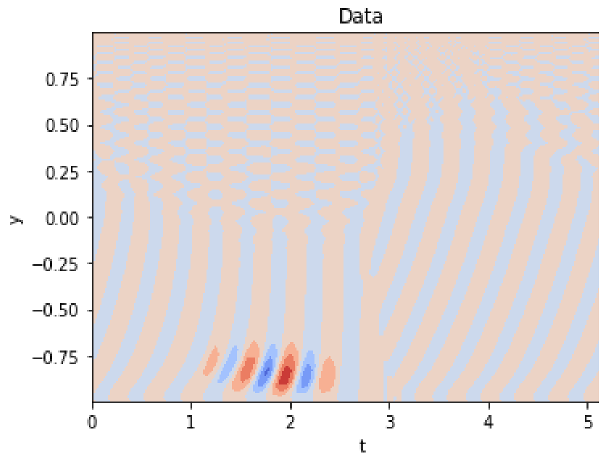
Fitted real PhiV velocity



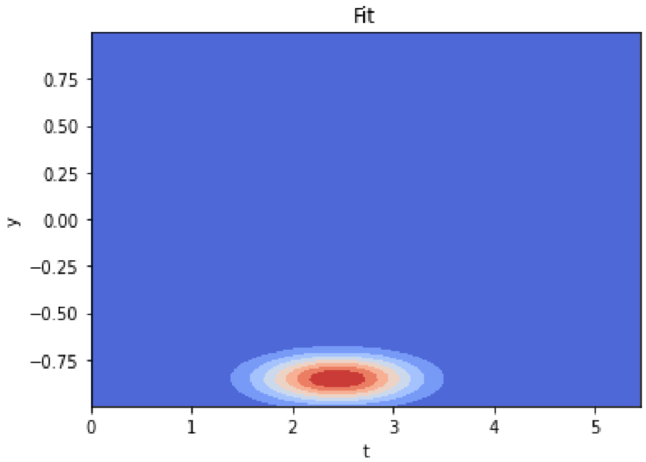
Fitted abs PhiV velocity



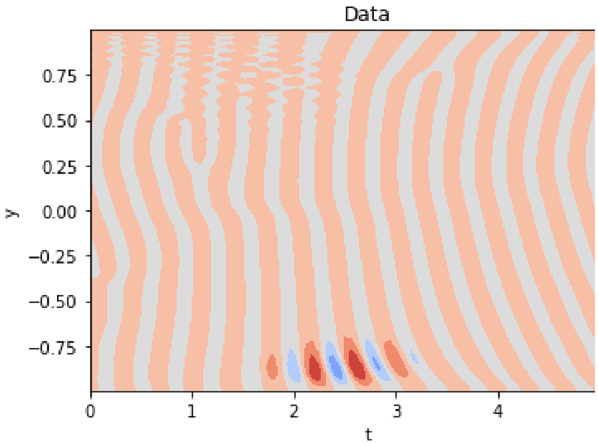
Original abs PsiU velocity



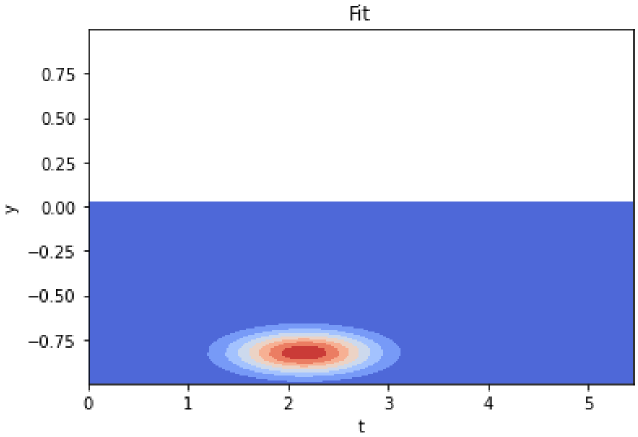
Original real PhiV velocity



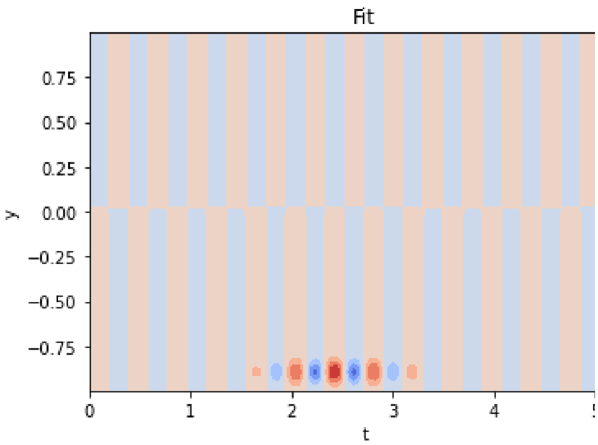
Fitted abs PsiU velocity



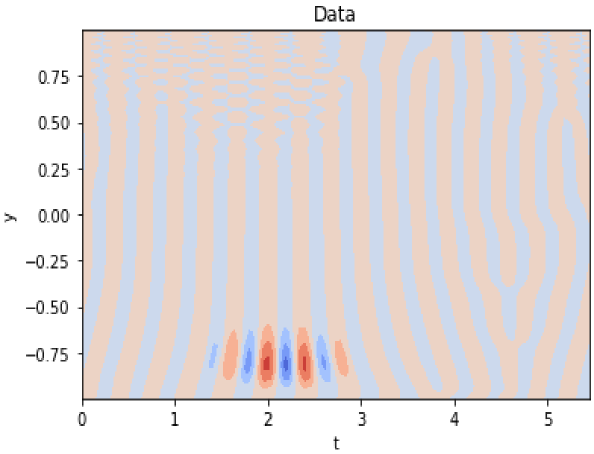
Original real PsiU velocity



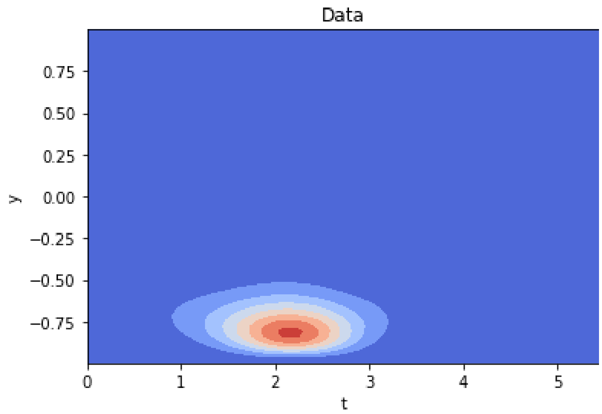
Fitted abs PsiV velocity



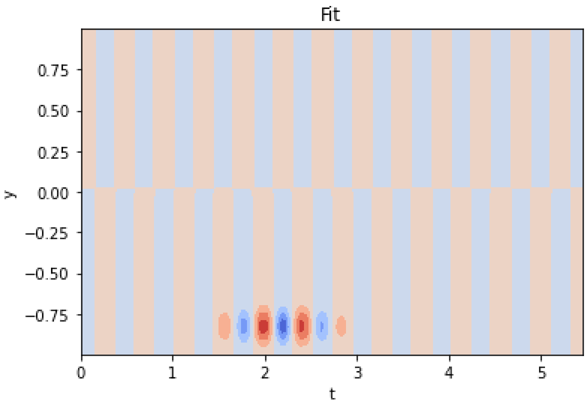
Fitted real PsiU velocity



Original real PsiV velocity

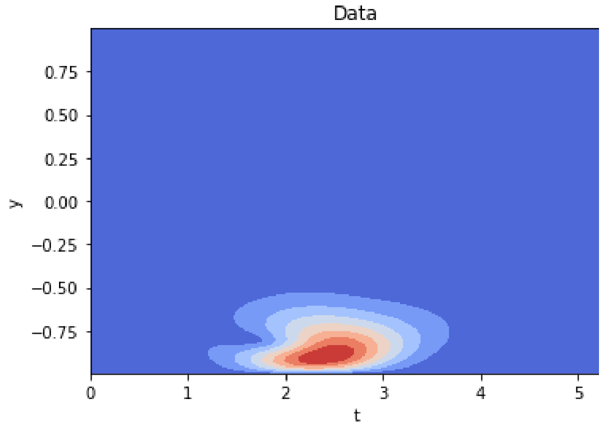


Original abs PsiV velocity

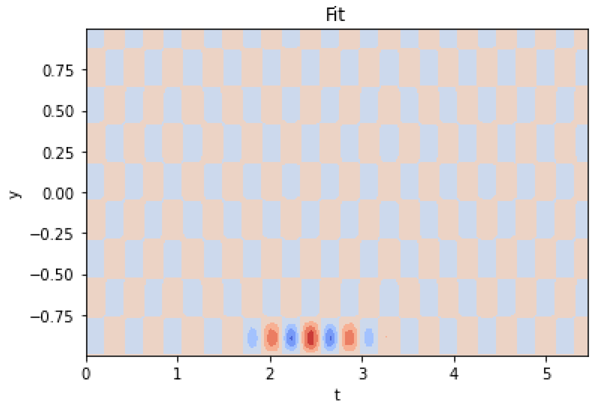


Fitted real PsiV velocity

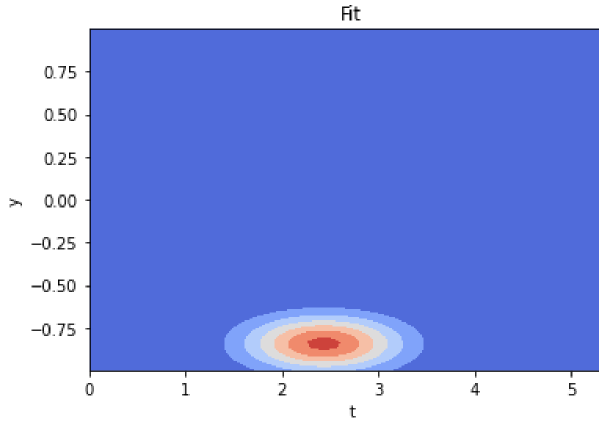
FINAL GROUP REPORT



Original abs PsiW velocity



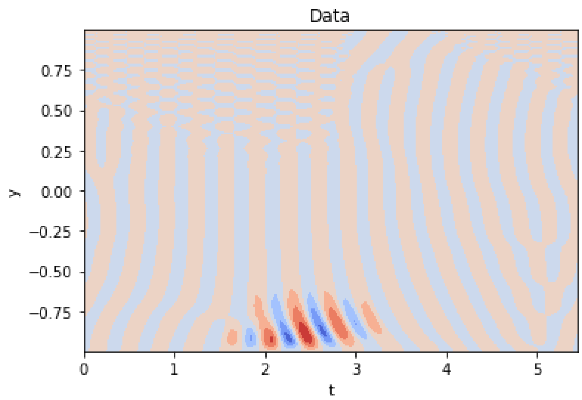
Fitted real PsiW velocity



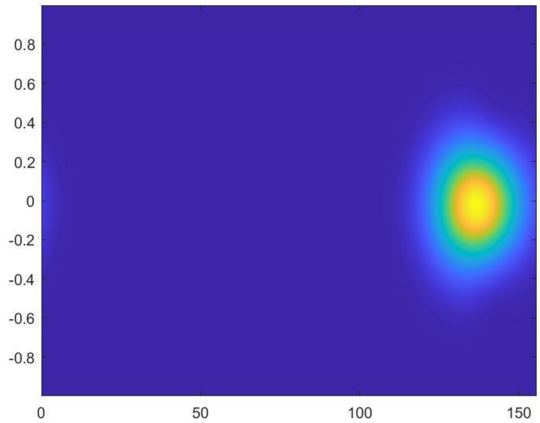
Fitted abs PsiW velocity

For DataModesV3-

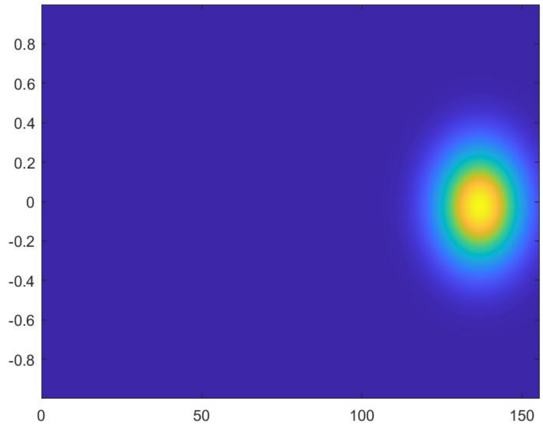
As evident by the name, DataModesV3 is an updated dataset of our previous DataModes. We have once again used our framework to create functions that represent the original data to a great extent as evident by the following plots-



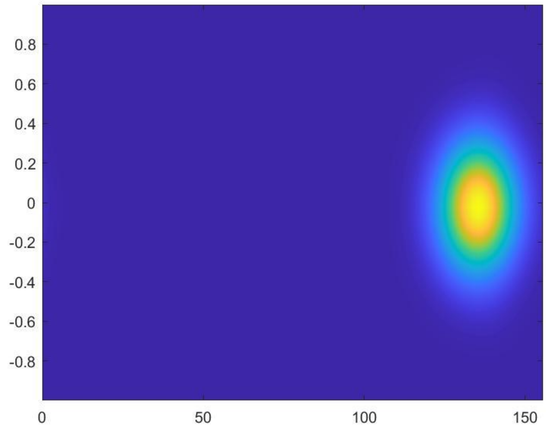
Original real PsiW velocity



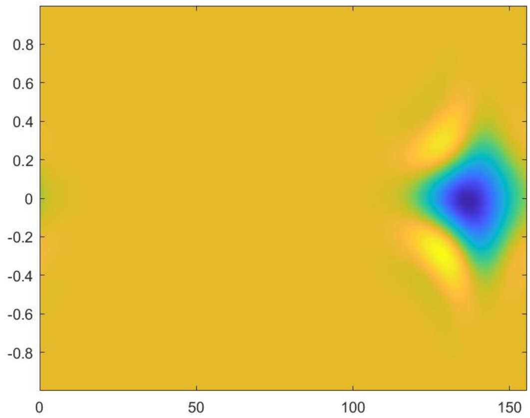
Original abs PhiU velocity



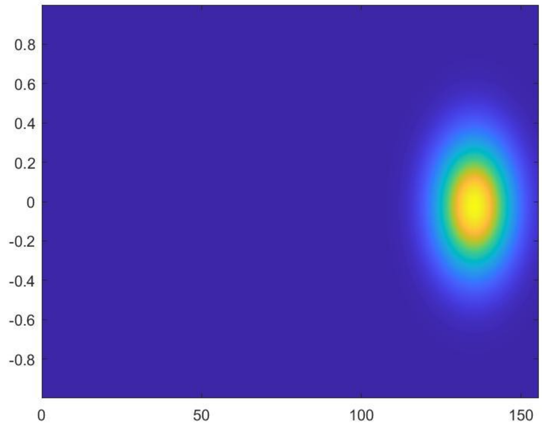
Fitted abs PhiU velocity



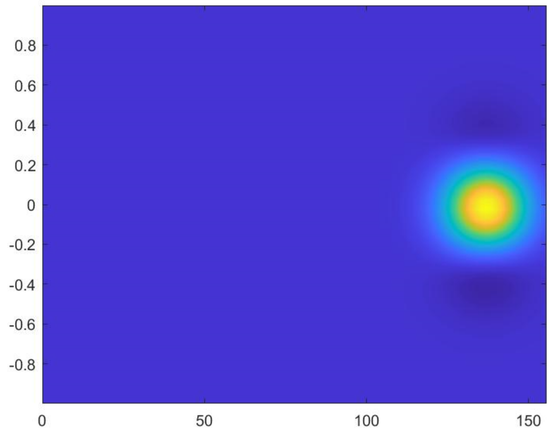
Original abs PhiV velocity



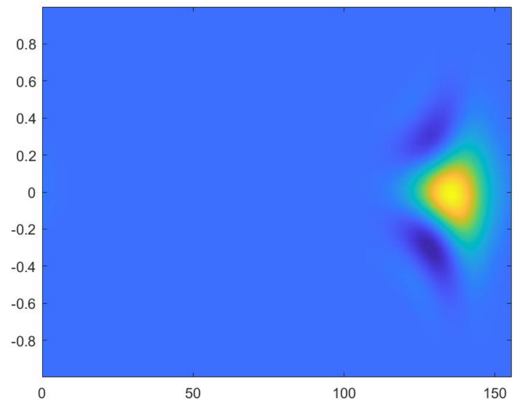
Original real PhiU velocity



Fitted abs PhiV velocity

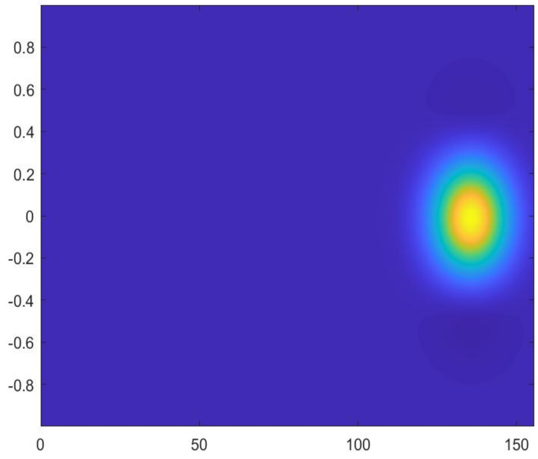


Fitted real PhiU velocity



Original real PhiV velocity

FINAL GROUP REPORT



Fitted real PhiV velocity

Conclusion –

After performing our analysis on a variety of data including both real world and theoretical and looking at the results that we have obtained, it is clear that our theory works and error is also minimal. But this theory does not provide us with a straight forward path as we have seen that in many cases, we had to select more than one gaussian sometimes even four and on top of that we also had to use boundary conditions and start point guesses sometimes in order to achieve a better fit. But our analysis also has its flaws as when it came to the real part of a function in DataModes, we were not able to get a great fit perhaps due to the phase not matching to a great extent or some other underlying issue. But apart from this, our error was less than 1% in flow over a cylinder, air-foil data, DataModesV3 and TestMatrix dataset.

Further Work –

In our work, we have done our analysis mostly on datasets that resemble a gaussian function in some form or the other. Further work may be done to expand our theory to cover a wide range of functions and eliminate some of the issues we encountered in our present work.

Acknowledgement –

We would like to thank you sir for giving us the opportunity to do this research which allowed us to hone our skills in Matlab and python and we also got to know about the basics of research work for which we are eternally grateful. We would also like to wish you good health and success in your future work.

TEMPERATURE SENSOR DESIGN TO CHECK CORRELATION BETWEEN ACTIVITY AND BODY SURFACE TEMPERATURE CHANGE
ECLTSB.1

A Report on Summer training submitted in partial fulfilment of the Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL & ELECTRONICS ENGINEERING

BY

PUSHP RAJ (BTECH/10567/19)

Under the guidance of
Dr. Aatmesh Shrivastava

Energy Efficient Circuits and Systems Lab College of Engineering Northeastern University Boston MA. US

And the supervision of

Dr. S. SHIVA KUMAR (Assistant professor)

EEE Department BIT MESRA

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
BIRLA INSTITUTE OF TECHNOLOGY MESRA,
RANCHI - 835215. (INDIA)

STUDENT DECLARATION

It is hereby declared that the **“Internship report”** submitted by **PUSHP RAJ (BTECH/10567/19)** is work done by him and submitted during 2019 – 2023 academic year, in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY in ELECTRICAL AND ELECTRONICS ENGINEERING,

at Energy Efficient Circuits and Systems Lab College of Engineering, Northeastern University

Northeastern University Interdisciplinary Science & Engineering Complex
805 Columbus Ave, Boston, MA 02120, United States

College Internship Coordinator - Dr. S. SHIVA KUMAR (Assistant professor, EEE)

Head of the Department Department of EEE -Professor T. Ghose

ACKNOWLEDGEMENT

I would like to express my deep gratitude to our project guide and supervisor, **Dr. Aatmesh Shrivastava and Dr. S.Shiva Kumar** under whose guidance I was able to pursue this endeavour .

I am very grateful to **Dr. Pankaj Mishra**, Birla Institue of Technology for extending all faculties and giving valuable suggestions at all times for pursuing this project.

I would also like to thank **Prof. T Ghose, HOD Electrical and Electronics, Birla Institute of Technology Mesra.**

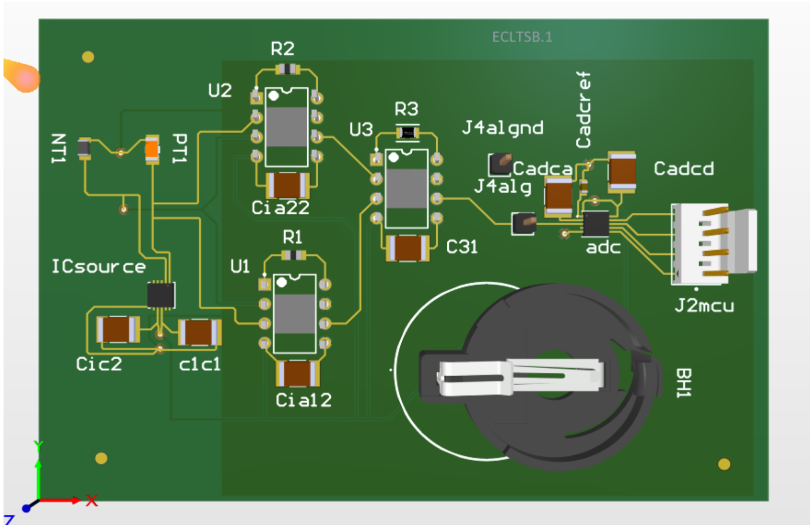
It feels impossible to express how grateful I am to the extent that it overwhelms me to realise the sheer support and positivity which was bestowed leading to the environment conducive for the success of this project

TABLE OF CONTENTS

INTRODUCTION

This report is drafted with the intention to outline my contributions in the development and design of *high resolution temperature sensor design for data acquisition in investigation of correlation between neurological activity and minute body surface temperature change*. During my internship at **Energy Efficient Circuits and Systems Lab College of Engineering, Northeastern University** Interdisciplinary Science & Engineering Complex 805 Columbus Ave, Boston, MA 02120, United States.

The proposed design achieved a staggering resolution of 0.0005 degree of temperature change while only utilising a 17-bit analog to digital conversion. The compact design of the sensor board makes it ideal for using it in biomedical applications.



ECLTSB.1

ABOUT THE LABORATORY

The EECS LAB at Northeastern University, College of Engineering is led by Prof. Aatmesh Shrivastava, performing research focussed on self-powered and ultra-low power circuits and system for Internet-of-things (IoTs). We are currently working to solve the power issue, in order to make IoT devices easily deployable in our environment. We are also focussing on analog computing hardware for AI applications, ultra-low power bio-medical and neural circuits, system modelling, and high-reliability circuits and system design. With 20 U.S granted patents and and multiple awards as well as grants from prestigious organisations like NSF, This lab is a powerhouse of innovation in analog circuits

ABOUT THE PROJECT

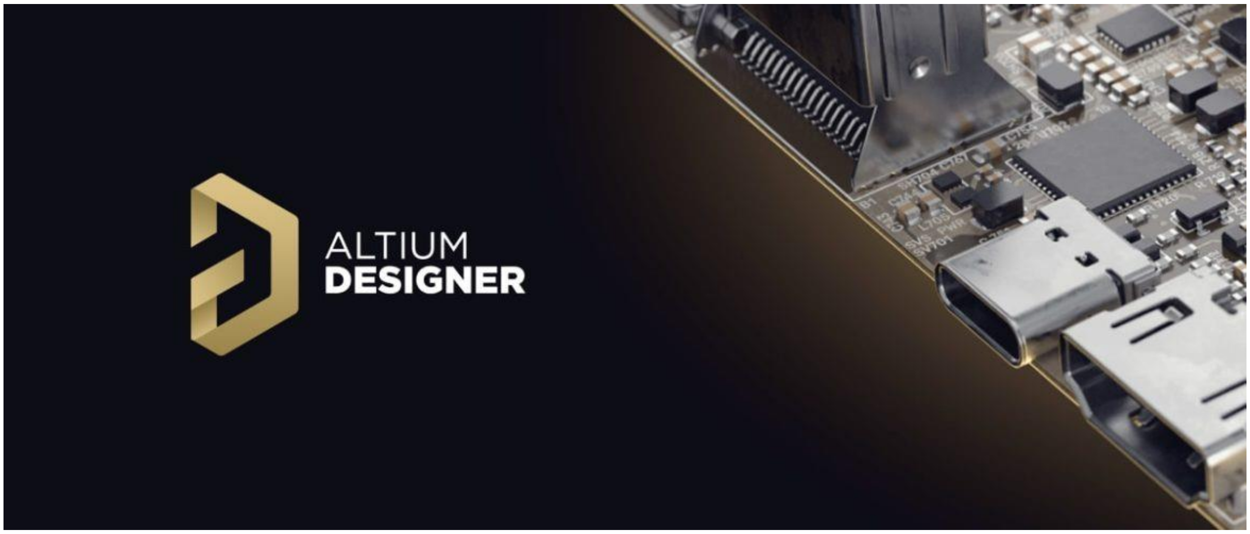
The temperature sensor design was based on the properties of thermistors. In which we utilised the caliber of both positive and negative temperature coefficient thermistors to create a difference in voltage whenever the temperature changed. Then this voltage difference was amplified to a level where the 17th bit of 18-bit ADC with 1LSB error range was enough to assign a value till the 0.0005 degree change in temperature. the ability of the board design is of high utility since the it works proficiently even with a small 3V coin cell and approximately 100 uA of current.

SOFTWARE TOOLS UTILISED

1. LT SPICE

LTspice is a SPICE-based analog electronic circuit simulator computer software, produced by semiconductor manufacturer Analog Devices (originally by Linear Technology). It is the most widely distributed and used SPICE software in the industry. Though it is freeware, LTspice is not artificially restricted to limit its capabilities (no feature limits, no node limits, no component limits, no subcircuit limits). It ships with a library of SPICE models from Analog Devices, Linear Technology, Maxim Integrated, and 3rd party sources too. In our case it was utilised for simulating the thermistor-based circuit including the amplification in a transient manner with step parametric temperature change.

2. ALTIUM DESIGNER



Altium Designer is a PCB and electronic design automation software package for printed circuit boards. It is developed by Australian software company Altium Limited.was originally launched in 2005 by Altium, known at the time as Protel Systems Pty Ltd. It has roots back to 1985 when the company launched the DOS-based PCB design tool known as Protel PCB (which later emerged into Autotrax and Easytrax). Originally it was sold only in Australia. Protel PCB was marketed internationally by HST Technology since 1986.The product became available in the United States, Canada, and Mexico beginning in 1986, marketed by San Diego-based ACCEL Technologies, Inc. under the name Tango PCB.In 1987, Protel launched the circuit diagram editor Protel Schematic for DOS.

In 1991, Protel released Advanced Schematic and Advanced PCB 1.0 for Windows (1991–1993), followed by Advanced Schematic/PCB 2.x (1993–1995) and 3.x (1995–1998). In 1998, Protel 98 consolidated all components, including Advanced Schematic and Advanced PCB, into a single environment. Protel 99 in 1999 introduced the first integrated 3D visualization of the PCB assembly. It was followed by Protel 99 SE in 2000. Protel DXP was issued in 2003, Protel 2004 in 2004, Altium Designer 6.0 in 2005. Altium Designer version 6.8 from 2007 was the first to offer 3D visualization and clearance checking of PCBs directly within the PCB editor.

HARDWARE COMPONENTS UTILISED FOR THE DESIGN

1. **NTC Thermistor**
2. **TFPT Thermistor**
3. **INA128 Instrumantation amplifier**
4. **Decoupling capacitors**
5. **Gain resistors**
6. **Connecting head**
7. **Current source IC**
8. **Battery holder**

**Please note that the datasheets for all these components are mentioned in the BIBLIOGRAPHY containing all the requisite and relevant information

BILL OF MATERIALS – This sheet comprises of all the requisite information about the hardware components required for assembly and fabrication including supplier as well as manufacturing details.

BOM:

Line #	Name	Description	Designator	Quantity	Manufacturer 1	Manufacturer Part Number 1	Supplier 1	Supplier Part Number 1
	ADS8881IDRCR	anlg to digi con	adc	1	TEXAS INSTRUMEN	ADS8887IDRCT	digikey	296-37265-2-ND
	1066	cell holder	BH1	1	keystone	CH25-2032LF	digi-key	36-1066-ND
	18125C105JAT2A	Ceramic Capac	c1c1, C31, Cadca	7	KYOCERA AVX	18125C105JAT2A	digikey	478-7984-2-ND
	LMK107BJ106MALTD	Capacitor	Cadcref	1	taiyo yuden	LMK107BJ106MALTD	digikey	587-2562-2-ND
	REF200AU_2K5	Integrated Circu	ICsource	1	TEXAS INSTRUMEN	REF200AU/2K5	digikey	296-27726-2-ND
	22-29-2041	CONN HEADER	J2mcu	1	Molex	22-29-2041	Arrow Electro	22-29-2041
	61300111121	THT Vertical Pir	J4alg, J4algnd	2	Wurth Electronics	61300111121	Mouser	710-61300111121
	NTCS0805E3222FMT	Thermistor	NT1	1	VISHAY	NTCS0805E3222FMT	digikey	BC2569TR-ND
	TFPT1206L5601FV	TFPT	PT1	1	VISHAY	TFPT1206L5601FV	MOUSER	71-TFPT1206L5601FV
	RT0805BRD0756KL	RESISTOR	R1, R2	2	YAGEO	RT0805BRD0756KL	digikey	YAG1933TR-ND
	RP73PF2A12K4BTDF	Resistor	R3	1	te connectivity	1-2176093-3	digikey	A110648TR-ND
	TI-INA1XX-8	Instrumentatio	U1, U2, U3	3	TEXAS INSTRUMEN	INA128P	digikey	INA128P-ND

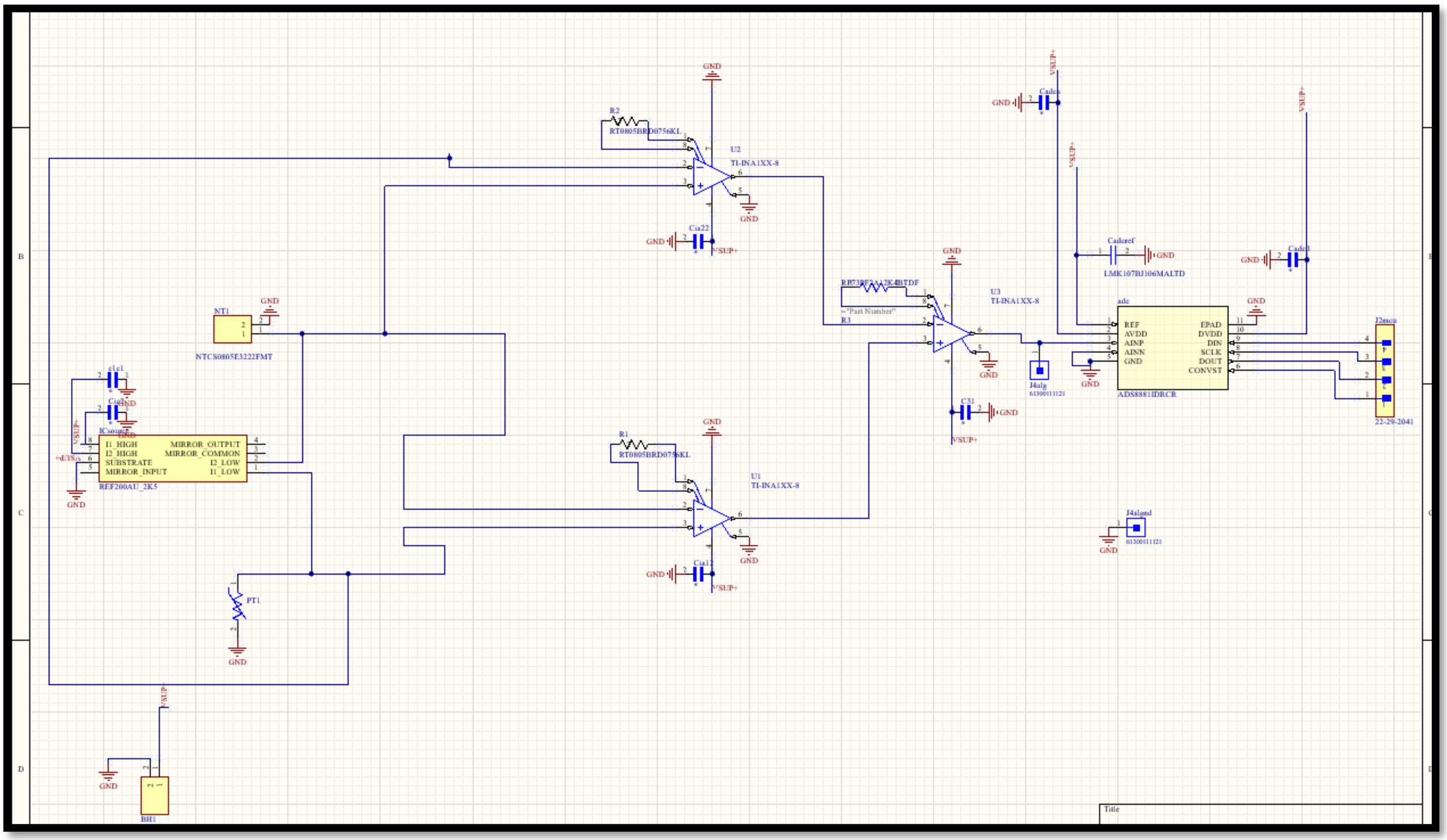
THE CIRCUIT

The circuit intends to make use of both increasing as well as decreasing temperature change and hence we utilise both a NTC as well as PTC thermistor to maximise the voltage difference between the points where we tap in the voltage from. Moreover, our design consists of 3 instrumentation amplifiers connected in opposing polarities to the thermistors with respect to each other, which in turn helps our temperature sensor to have a larger temperature sensing range even at extremely low temperatures without compromising the integrity of the signal/ differential voltage.

This analog acquisition unit is then driven into an 18bit – ADC with given error range of 1LSB making it effectively 17-bit. Considering all the factors, all the components in the circuit are decoupled using capacitors mentioned in their advised applications.

The current is drawn from a current source IC which operates on the same supply as rest of the components on the board i.e 3Volts which is supplied by a coin cell .

THE CIRCUIT SCHEMATIC



THE PCB DESIGN

The PCB was designed using Altium designer wherein the number of layers was restricted to 2. This itself contributed immensely in making the board ergonomic.

The components were placed ton the board such that the thermistors are a little away from the other components since otherwise this would've caused unnecessary deviation in the temperature measurements.

Specific design rules were followed to reduced the cost of fabrication for the board and maintaining the integrity while minimizing heat dissipation as well as current leakage.

In the the adc is connected to a header with the requisite output as well as SPI pins.

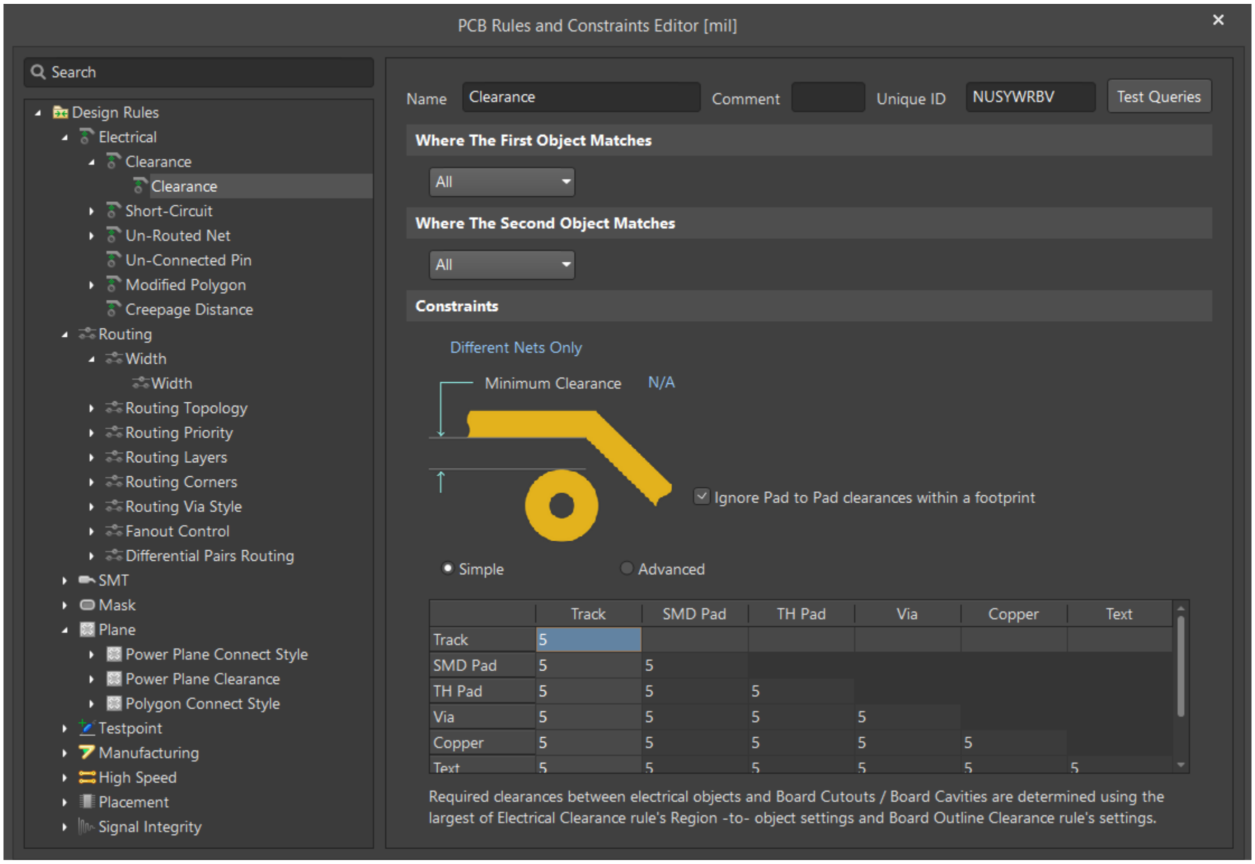
INDEX OF DESIGNATORS FOR CIRCUIT COMPONENTS

Description	Designator
analog to digital converter	adc
cell holder	BH1
Ceramic Capacitor X7R 1812 (4532 Metric)1uF	c1c1, C31, Cadca, Cadcd, Cia12, Cia22, Cic2
Capacitor 10uF	Cadcref
Integrated Circuit	ICsource
CONN HEADER VERT 4POS 2.54MM	J2mcu
THT Vertical Pin Header WR-PHD, Pitch 2.54 mm, Single Row, 1 pins	J4alg, J4algnd
Thermistor	NT1
TFPT	PT1
RESISTOR for IA gain 1st stage	R1, R2
Resistor for IA gain 2nd stage	R3
Instrumentation amplifiers	U1, U2, U3

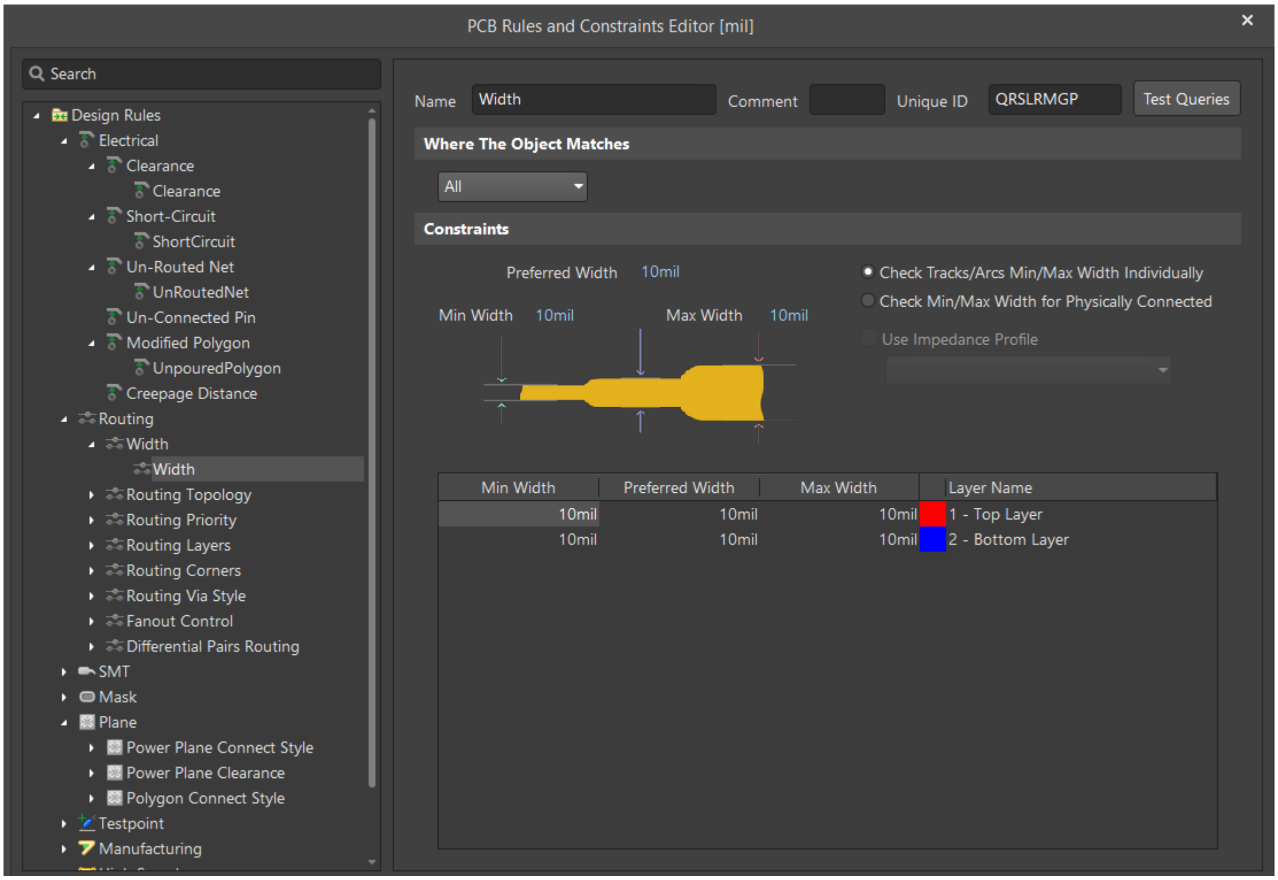
DESIGN RULES

***ALL BOARD MEASUREMENTS OF LENGTH ARE IN MILS**

1. CLEARANCE



2. ROUTING



***** the rules as evident in the images were set as following respectively:**

Clearance- 5 mils

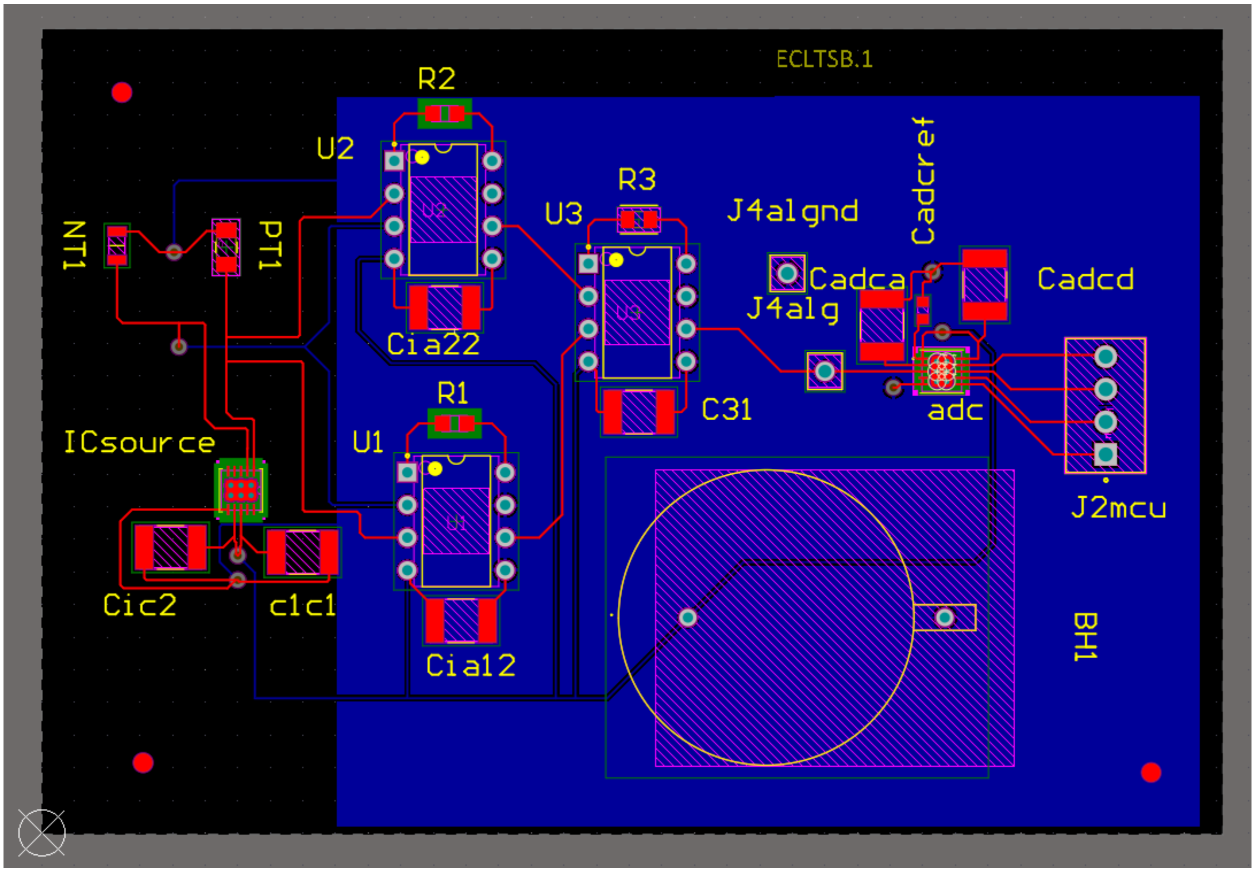
Routing – 10 mils

These were the values indicated by the fabricating facility and therefore we proceeded with these.

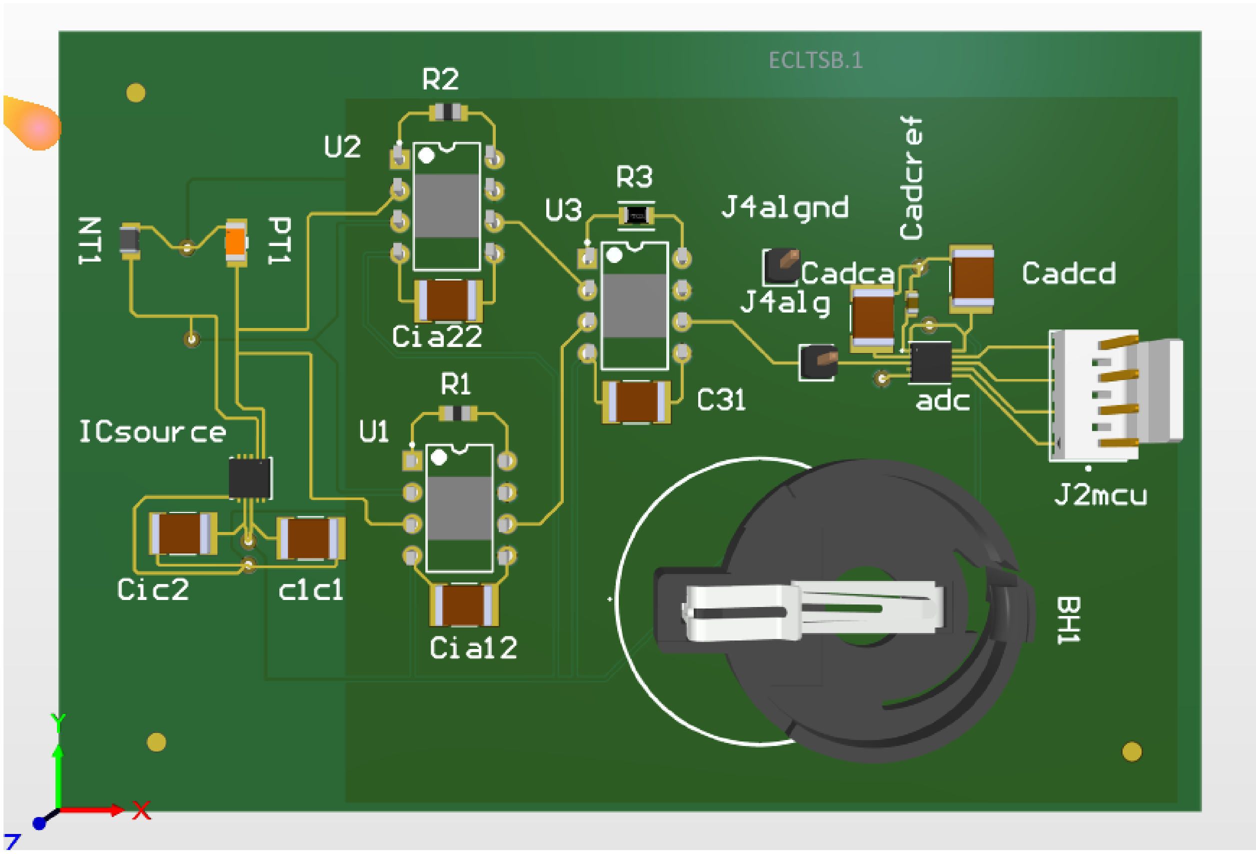
LAYERS

LAYER.1

The first layer or the top most layer is the layer where most of the component assembly is done. All components are mounted on this layer. In design window the connection / routing done on this layer is depicted by red colour traces and as yellow traces in 3D view as shown in the figures below :



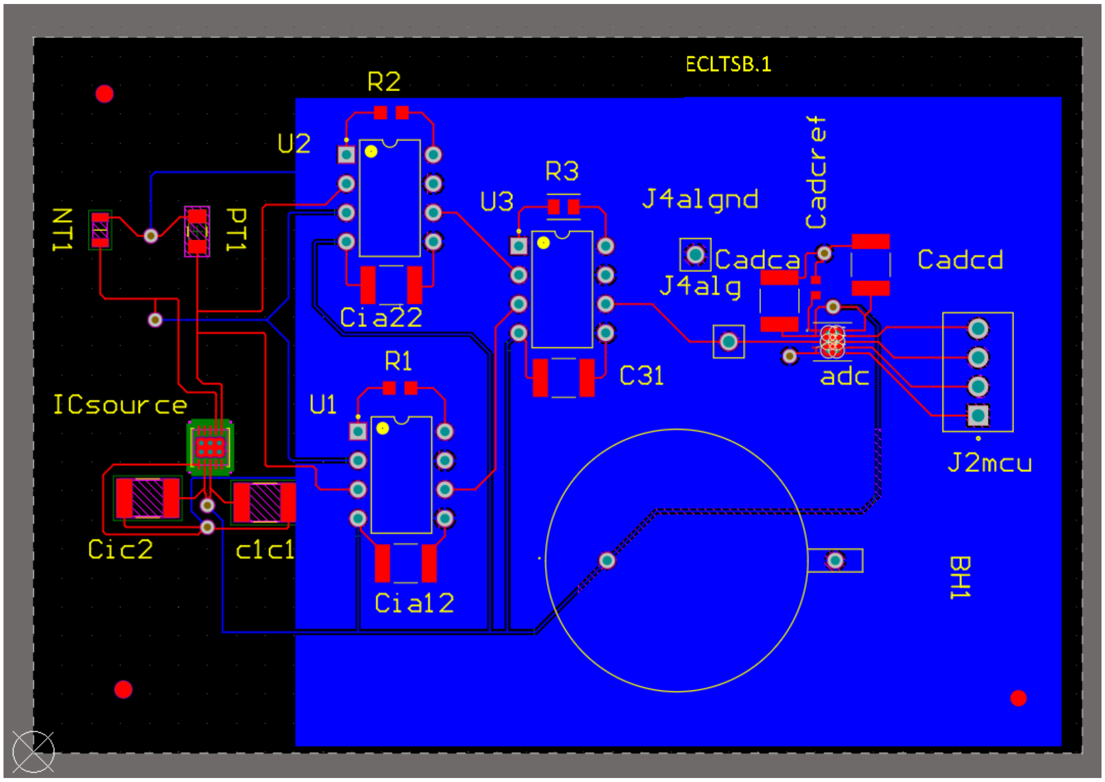
A- DESIGN VIEW OF PCB WITH TOP LAYER/ 1ST LAYER HIGHLIGHTED



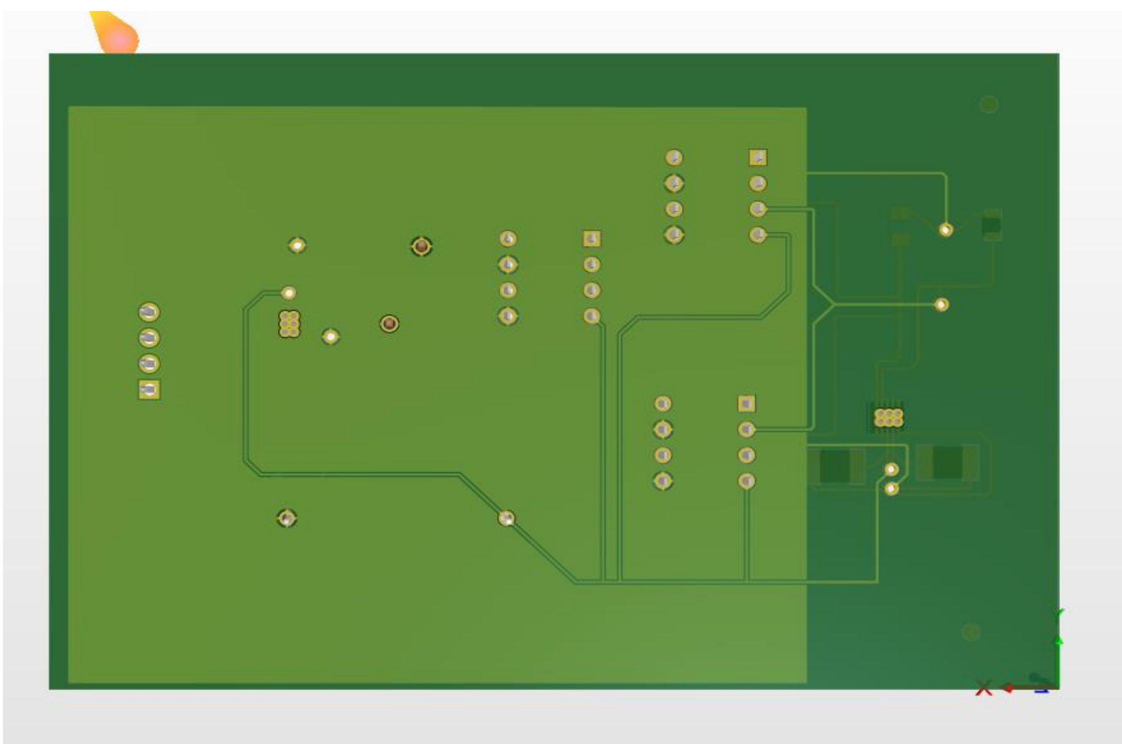
B- 3-D VIEW OF THE BOARD TOP LAYER

LAYER.2

This layer mainly constituted of the ground polygon pour and connections to the battery holder pins which would act as the supply, we have utilised various pads and vias to route connections between layers therefore reducing overcrowding in between the connections. The connections can be seen as traces in the colour blue along with the pour in the design mode whereas light green in 3D mode as shown in the figures below :

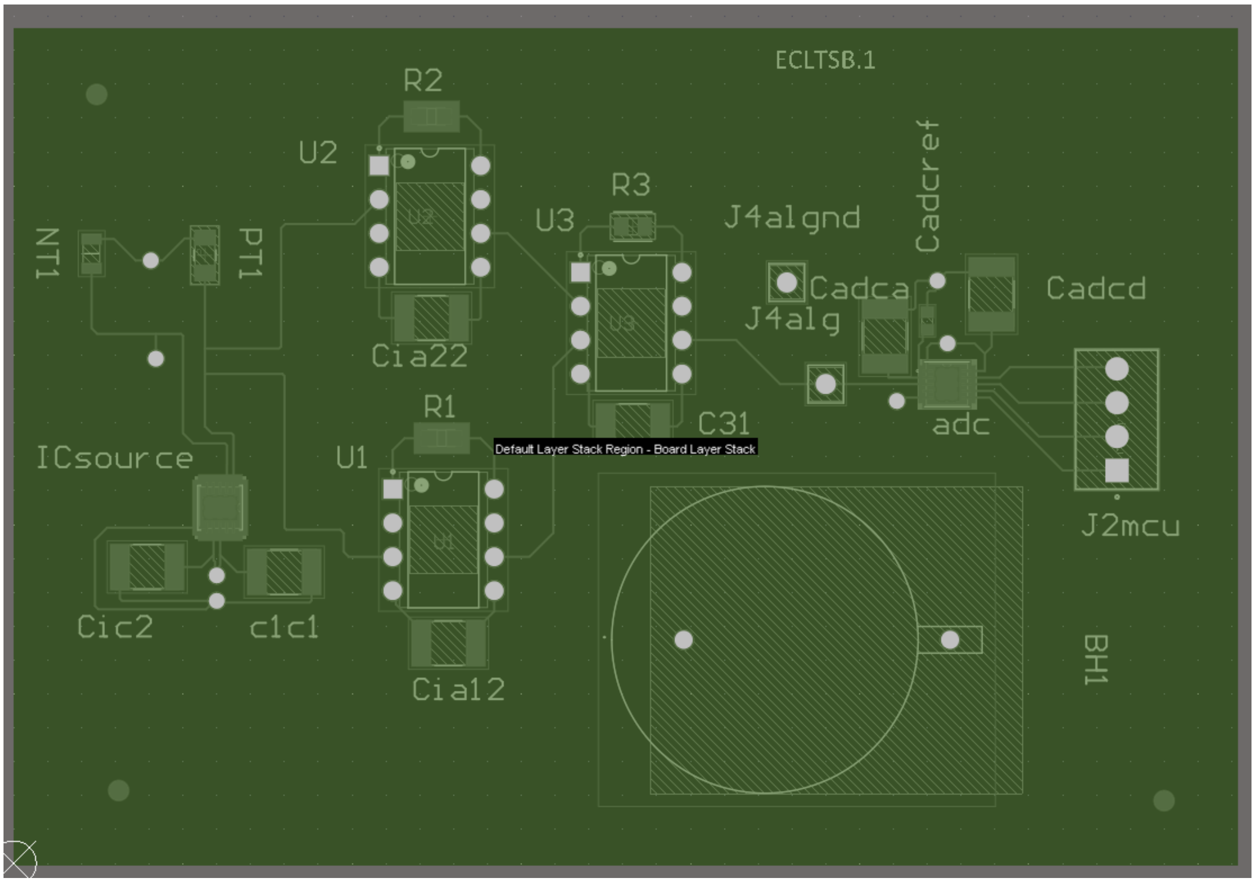


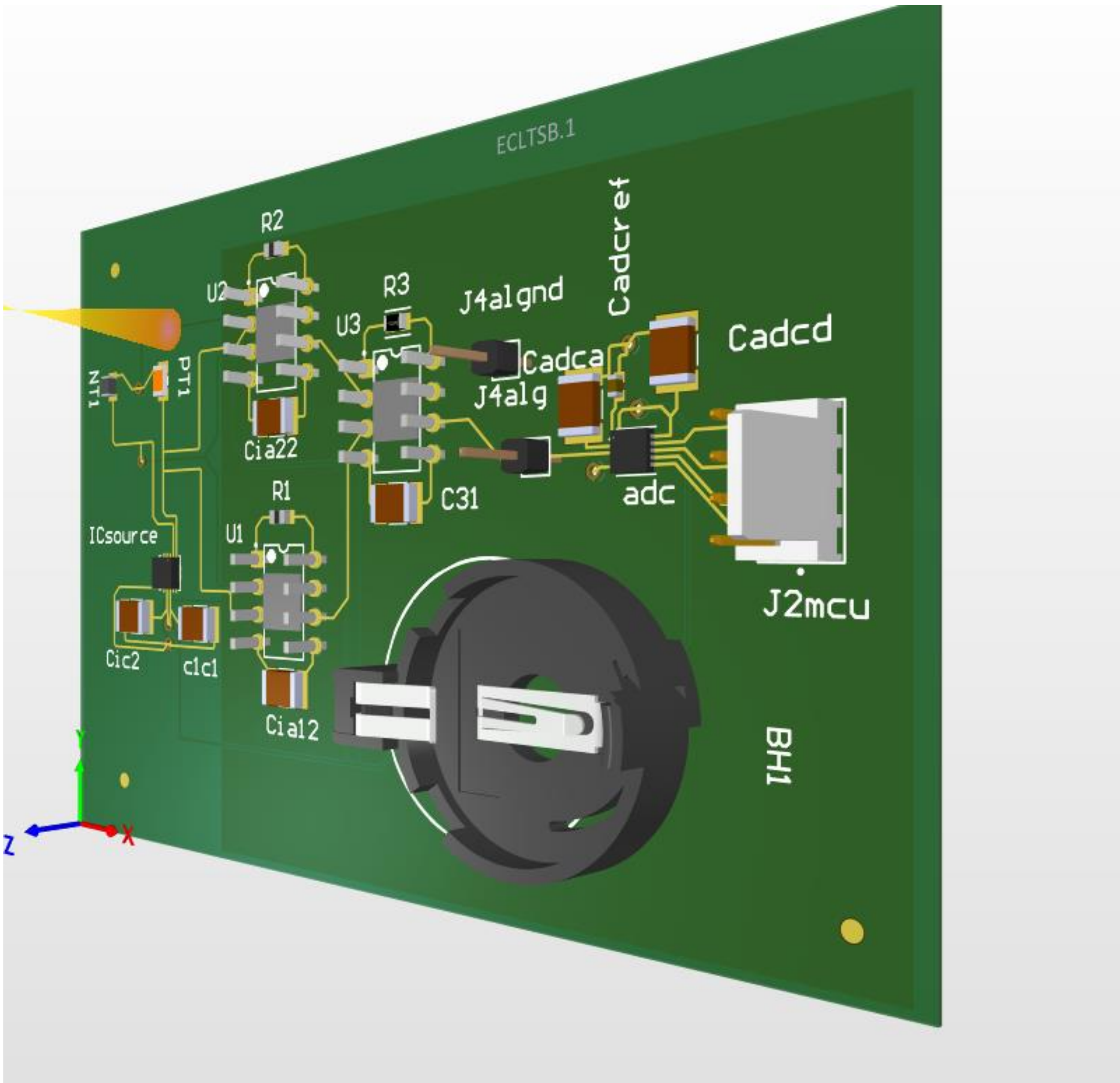
A. DESIGN VIEW OF THE BOARD WITH LAYER 2 HIGHLIGHTED



B. 3D VIEW OF THE BACKSIDE/ 2ND LAYER OF THE PCB

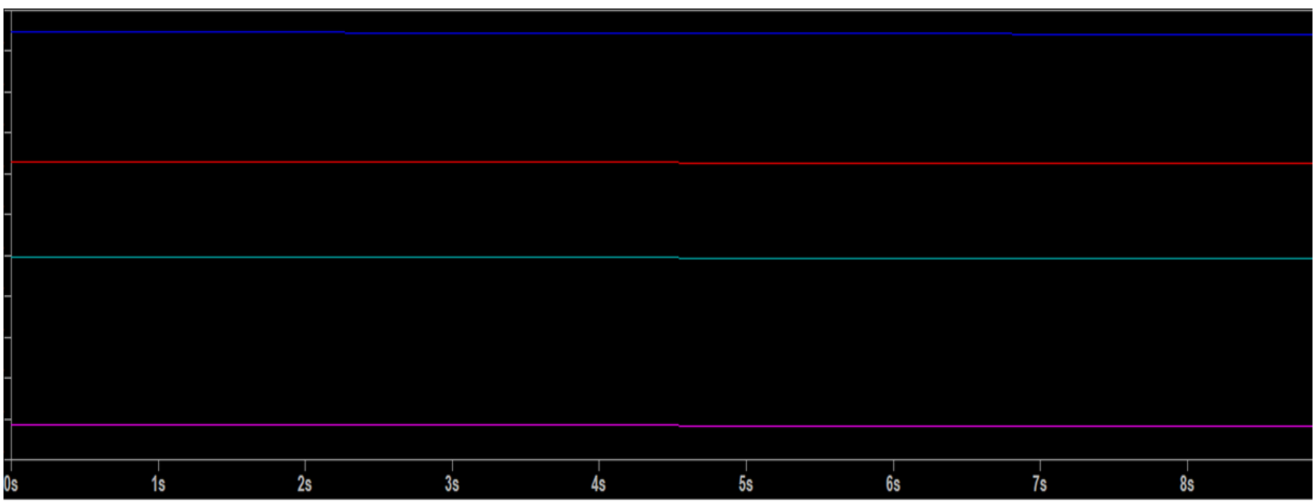
FURTHER IMAGES OF THE BOARD DESIGN




ECLTSB.1
R2
U2
R3
U3
J4algnd
Cadcref
Cadcd
Cadca
J4alg
NT1
PT1
Cia22
C31
adc
R1
ICsource
U1
J2mcu
Cic2
cic1
Cia12
BH1

SIMULATION SNAP

This snapshot depicts the various voltages at various temperatures set at 0.0005 degrees difference while running a transient simulation of current at 100uA



CONCLUSION

In conclusion one could argue that why should we develop a board based temperature sensor design when IC designs are highly accurate and much better suited , Our design tends to compete with that claim while adding value due to its modular nature and low cost of operation, assembly and fabrication without any compromise on immaculate performance and exceptional resolution . Therefore it is appropriate to state that this board with great connectivity and such high resolution can be suitably used for the mentioned hypothesis and investigate whether it’s the truth or a fallacy .

BIBLIOGRAPHY

Links to all datasheets are as follows :

1.https://www.ti.com/lit/ds/symlink/ina128.pdf?ts=1660668810701&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FINA128

2. https://www.mouser.com/datasheet/2/418/5/ENG_DS_1773272_M-1588495.pdf

3. https://www.yageo.com/upload/media/product/productsearch/datasheet/rchip/PYu-RT_1-to-0.01_RoHS_L_12.pdf

4. https://www.mouser.in/datasheet/2/427/tfpt-1762890.pdf

5. https://www.ti.com/lit/ds/symlink/ref200.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1660685366750&ref_url=https%253A%252F%252Fwww.mouser.in%252F

6. https://4donline.ihs.com/images/VipMasterIC/IC/TAIY/TAIY-S-A0002471898/TAIY-S-A0002471898-1.pdf?hkey=6D3A4C79FDBF58556ACFDE234799DDF0

7. https://datasheets.kyocera-avx.com/X7RDielectric.pdf

8. https://www.keyelco.com/userAssets/file/M65p5.pdf

9.https://www.ti.com/lit/ds/symlink/ads8881.pdf?ts=1660685096424&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FADS8881%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Dasc-null-null-GPN_EN-cpc-pf-google-wwe%2526utm_content%253DADS8881%2526ds_k%253DADS8881%2526DCM%253Dyes%2526gclid%253DCj0KCQjwgO2XBhCaARIsANrW2X0pJSGf8Y1kzzVJ2OonhNeiOaBamIDFVw_YlYXC8-5wOqQGsNyGqYkaAibaEALw_wcB%2526gclsrc%253Daw.ds

10. https://www.vishay.com/docs/29044/ntcs0805e3t.pdf

Summer Internship Report

Driver-Drowsiness Detection model using Convolutional Neural Network

Rahul Hembrom
Roll No. – BTECH/10213/19
Branch – CSE – 'A'

Advisor: Dr. Jafar Saniie

Graduate Assistant: Mr. Xinrui Yu

Summer 2022

ACKNOWLEDGEMENT

The internship opportunity I had at Armour R&D, Illinois Tech, Chicago was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me though this internship period.

I would like to express gratitude to my advisor **Dr. Jafar Saniie** who gave me the opportunity to work under his able guidance. I would also like to thank my Graduate Assistant **Mr. Xinrui Yu** for helping me throughout the internship. I also take this opportunity to express my deepest thanks the Head of Department, **Professor Sandip Dutta**, for giving me this privilege.

Lastly, I would like to thank : Birla Institute of Technology, Mesra; Birla Institute of Technology Mesra Alumni Association – North America (BITMAA-NA) for funding me for Immersive Summer Research Experience - 2022 program.

I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future, Sincerely,

Rahul Hembrom
Date: 17/08/2022

CERTIFICATE OF PARTICIPATION

ILLINOIS TECH | Armour College of Engineering

Upon the recommendation of the faculty of the
Armour College of Engineering
Of Illinois Institute of Technology

Rahul Hembrom
Is recognized as achieving

Summer Undergraduate Engineering
Research Immersion Program
Certificate of Participation

With all the rights, privileges and honors thereunto appertaining
Awarded at Chicago, in the State of Illinois of the United States of America
July 15, 2022

Dean, Kevin W. Cassel

Armour R&D

PREFACE

OBJECTIVE

The goal of our research is to develop a real-time model, which would successfully be able to detect if the driver of the vehicle has or is going to fall asleep and also the frequency of the repetitive sleeping behavior of the driver. Particularly, the study has the following sub-objectives:

a) To develop a model which would surpass the accuracy of previously developed drowsiness detection models.

b) To develop a model which extracts important facial as well as postural features.

c) To generate a mathematical correlation between the extracted features.

d) To develop a proper classification method.

The result of this research will be valuable to the industry practitioners as well as related software providers in developing better safety tools in the automobile sector.

OUTCOMES

a) Provides basic idea of different real life research processes, problems, trouble shooting, decision making and preventive maintenance techniques and professional culture of industry, work ethics and attitudes in industry.

b) Helps to understand the importance of the application of modern tools for the industrial automation / up-gradation / scale-up.

c) Develops conceptual theory and the regular class room discussion and its application in real-life industrial problem resolution.

d) Gives primary idea to analyse Case studies based on real life application.

Table of Contents

ABSTRACT

This project aims to build a real-time drowsiness detection model with the help of a Convolution Neural Network that can identify key attributes of drowsiness in the driver and triggers an alert when the driver is going to fall asleep. Computers and their algorithms have come a long way in successfully detecting the facial features of a human being, moreover, the fields of Computer Vision, Machine Learning, and Artificial Intelligence are growing rapidly. This paper aims to leverage the tools in these fields to make a Driver-Drowsiness detection model. The number of creators creating and adding pre-trained ML models to load and use from the Internet has skyrocketed. We plan on using an existing pre-trained YOLOv5s model (a lightweight model) on our dataset to make our system, from the YOLOv5 repository on GitHub. YOLOv5 is a family of object detection architectures and models pre-trained on the open-source COCO dataset containing around 80 different classes and represents Ultralytics open-source research into future vision AI methods. We are using the PyTorch ML framework to load the YOLO model and make detections along with OpenCV to access the webcam and render feeds.

INTRODUCTION

Drowsy driving refers to any instance of operating a vehicle while fatigued or sleepy. Even if the driver has not consumed alcohol, studies have shown the effects of driving while sleep-deprived are similar to those of drunk driving. These effects include impaired attention and coordination, slower reaction time, and poor judgment. It's believed that going 24 hours without sleep is comparable to having a blood-alcohol content of 0.10%, which exceeds the legal limit for driving in all 50 states.

According to the National Safety Council (NSC), drowsy driving accounts for over 100,000 crashes, 71,000 injuries, and 1,550 fatalities in the United States only. In another survey conducted by the AAA, 27% of drivers report driving while being so tired they have difficulty keeping their eyes open. In a rapidly developing world, more and more people will have access to private vehicles, and the accidents relating to drowsy driving are sure to go up. The existing models are extremely expensive and only work with high-end cameras and cars, or those which are affordable have low accuracy and are not robust. Hence it is imperative to develop an accurate model that detects the earliest signs of fatigue and drowsiness in the driver to alert him/her in time.

RELATED WORKS

1. Zhong G. December 14, 2019. Drowsiness Detection with Machine Learning. Towards Data Science

Objective: to detect extreme and visible cases of drowsiness along with softer signals of drowsiness as well.

Dataset: Real-Life Drowsiness Dataset from the University of Texas at Arlington.

Dataset Description: 30 hours of videos of 60 unique participants. Each video is 10 minutes long with around 240 frames being extracted from each video, i.e around 10560 frames to create the entire dataset.

Methodology: The paper used a 1-D CNN which extracted the EAR(Eye Aspect Ratio) and MAR(Mouth Aspect Ratio) as features to classify the object.

Accuracy: K-Nearest Neighbor (kNN, k = 25), Accuracy = 77.21% (BEST)
Naive Bayes, Accuracy = 57.75% (WORST)

2. Real-Time Driver Drowsiness Detection using Computer Vision, Mahek J., Bhavya B., Sowmyarani C. N., International Journal of Engineering and Advanced Technology, Volume-11 Issue-1

Objective: Detect a driver's drowsiness based on eyelid movement and yawning and reliably give appropriate voice alerts in real time.

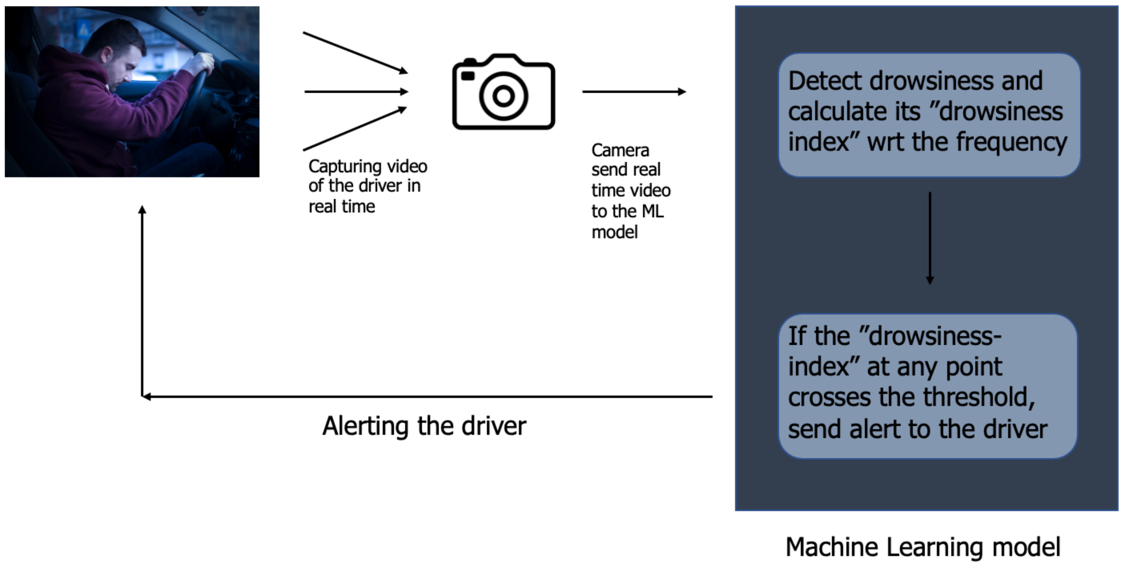
Dataset: iBUG-300w Dataset

Dataset Description: Open-Source dataset consisting of 300 indoor and as many outdoor images of human faces with different facial expressions.

Methodology: The face is localized using facial landmark detection. Then, shape prediction methods are used to detect important features by OpenCV built-in HAAR cascades, which are Pre-trained with EAR and MAR as the features.

Accuracy: As the model was trained on images the accuracy is over 95% possibly suggesting overfitting of data.

MODEL ROADMAP:



TECHNICHAL DESCRIPTIONS

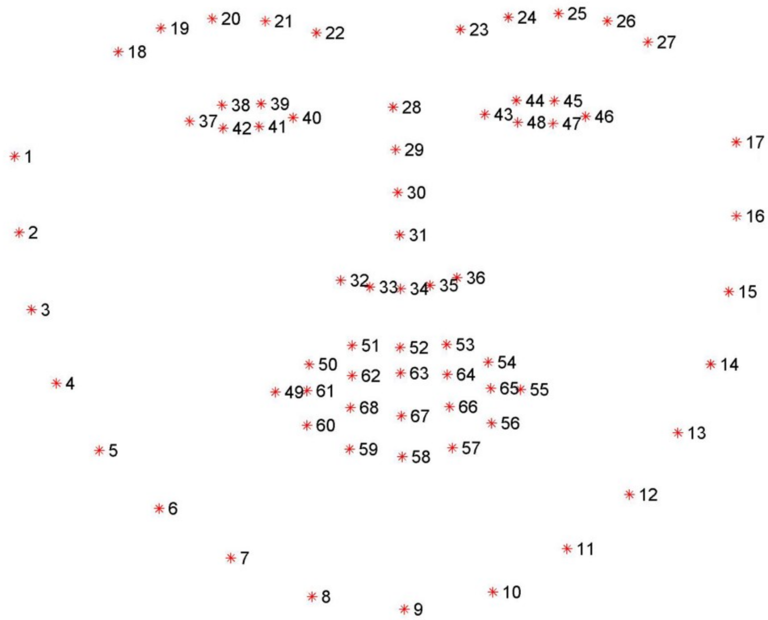
A. PROTOTYPE *(appendix A)*

The first version of the base prototype was our first attempt at creating a functioning Drowsiness detection model.

Component Required:

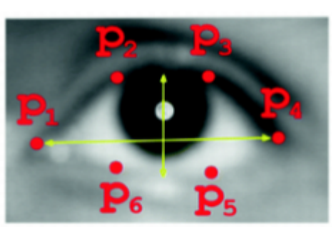
1. Python (Programming Language)
2. OpenCV (Image Processing Library)
3. Dlib (Extracting Facial Features)
4. Nvidia Jetson Nano
5. Microsoft Lifecam HD-3000

- **Dlib** is an open source suite of applications and libraries written in C++. It offers a wide range of functionality across a number of machine learning sectors, including classification and regression, numerical algorithms such as quadratic program solvers, an array of image processing tools, and diverse networking functionality, among many other facets.



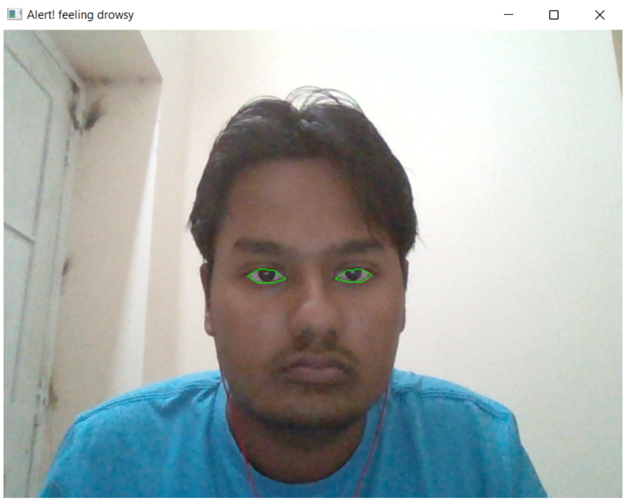
- Dlib first detects the region of the face, then searches facial landmarks within the facial region to increase efficiency, it uses Histogram of Oriented Gradients (HOG) for Object Detection with a linear classifier, an image pyramid, and a sliding window detection scheme to detect faces in an image. Once the region of the face is determined, facial landmarks will be detected using One Millisecond Face Alignment with an Ensemble of Regression Trees.

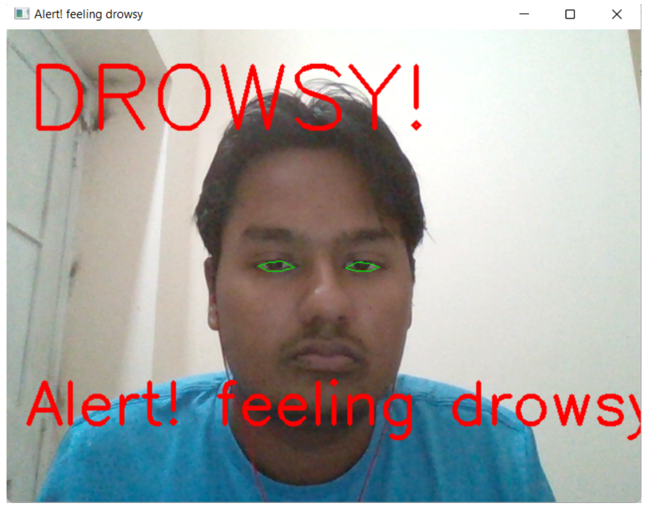
- The prototype model consists of a **Nvdia Jetson Nano**, along with a monitor and USB webcam. The model calculates EAR(Eye Aspect Ratio) of the Left and Right Eyes separately by using the point extracted by the Dlib shown above. EAR, as the name suggests, is the ratio of the length of the eyes to the width of the eyes. The length of the eyes is calculated by averaging over two distinct vertical lines across the eyes as illustrated in the figure below.



$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

- Then the program averages the EAR values of the two eyes and checks it against the threshold value (See in Appendix A). The threshold value was manually set through trial and error, and when the value crosses the threshold. The alarm is set off.





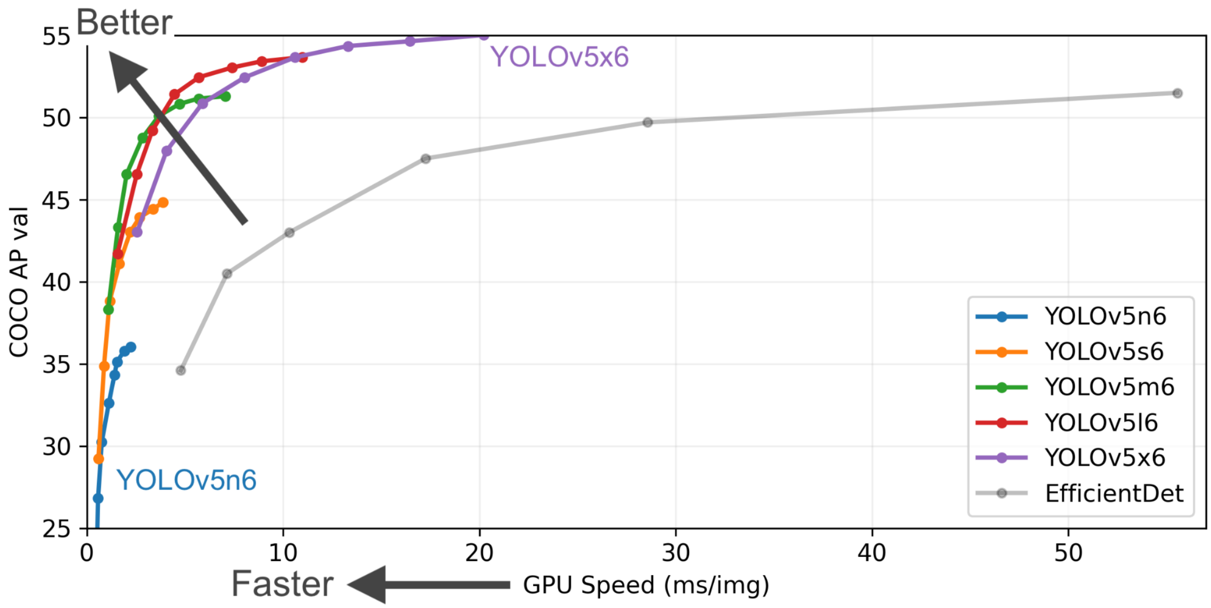
B. WORKING MODEL *(appendix B)*

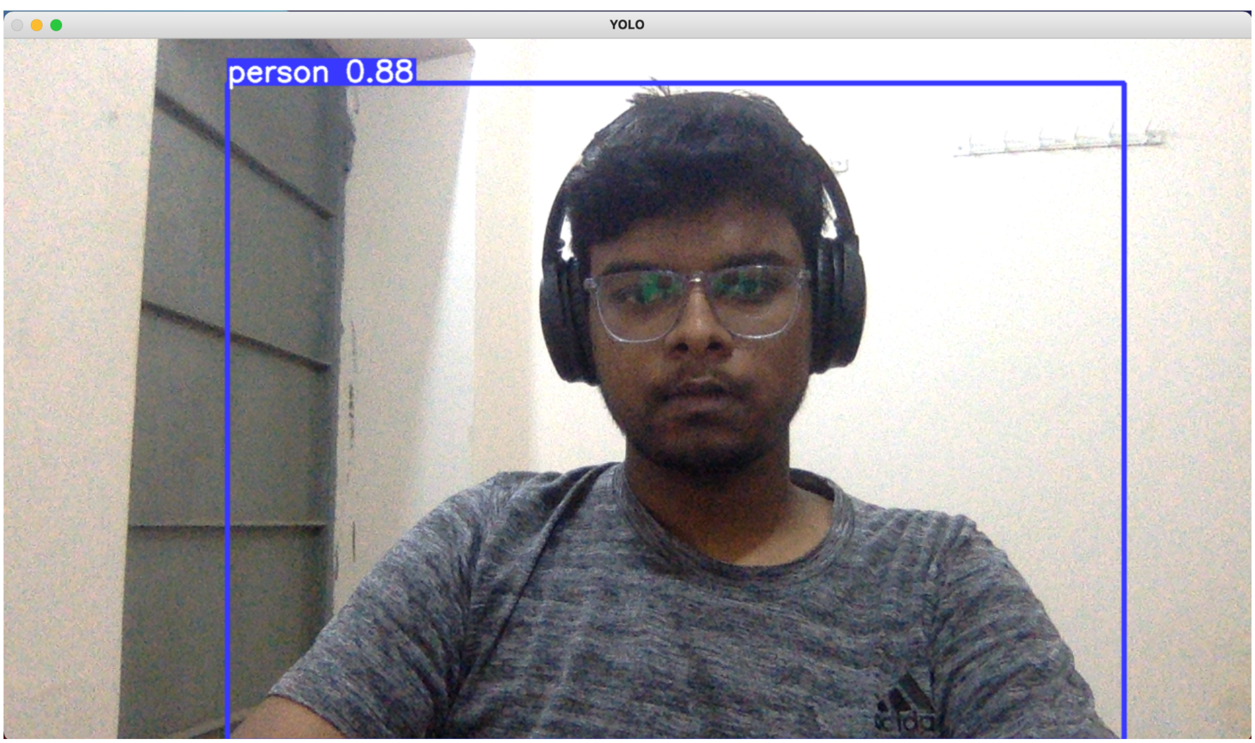
The final proposed version of the Drowsiness detection model.

Components Required:

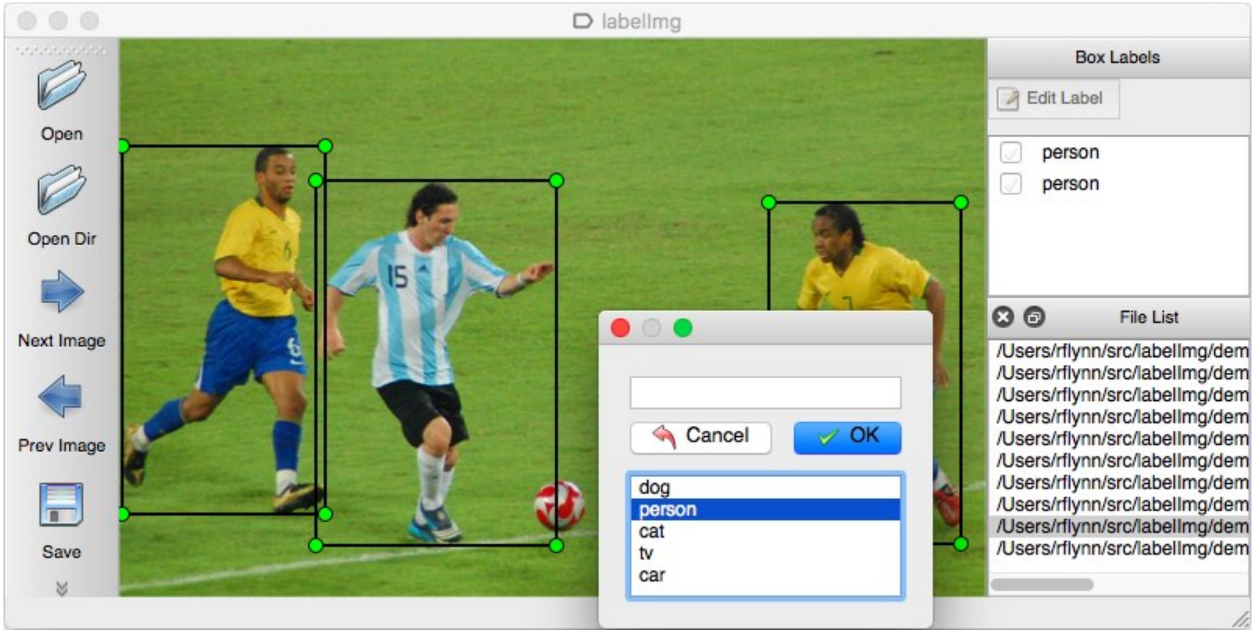
1. Python (Prgramming Language)
2. YOLOv5 (Image extracting and library)
3. LabelImg (A tool to convert images into vectors)
4. PyTorch (Machine Learning Library)
5. WebCamera (Capturing Image)

- **YOLOv5** is a family of object detection architectures and models pretrained on the COCO dataset, and represents Ultralytics open-source research into future vision AI methods, incorporating lessons learned and best practices evolved over thousands of hours of research and development. YOLOv5 consists of a pretrained CNN Model which helps in accurate detection of several objects which consists of human face as well. Using this model improved the detection of faces and its features drastically. It uses a sigmoid linear unit (SiLU) as an activation function and an architecture with 19 convolution layers and 5 Max Pooling layers. Once the YOLOv5 CNN model was imported, the model was then tweaked by training the CNN such that it could detect the drowsiness amongst the person as well.

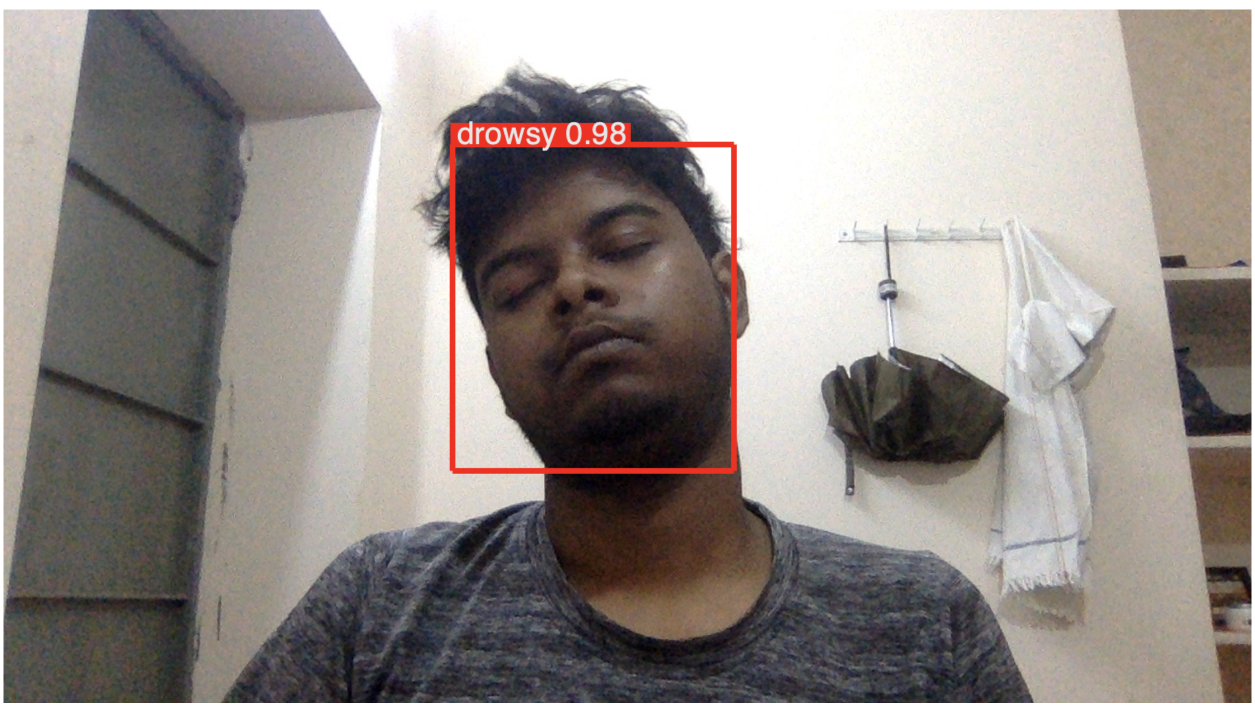




- **LabelImg** is a github repository which consists of a python script which inputs the image in jpeg format and then converts it into a numeric array for the training dataset. In this case several frames were captured using a python script both in optimum light and as well as in dark, which then through the LabelImg were first cropped into the areas of focus (eyes, mouth, face, etc.), then converted into the format of array of arrays numbers, which is the default training input in YOLO Models along with the labels 'awake' or 'drowsy'.



- After the retrieval of input datasets the imported CNN Model was trained with the new dataset with 320 image variations, 16 batches and 500 epochs, and the program written would run a python script consisting of the model which would detect the facial expression of the person frame by and display the current state of the of the person, either awake or drowsy.



CONCLUSION

In conclusion our proposed system takes the frames from video feed of the driver and sends those frames to the YOLOv5 CNN model, which has been already pre-trained with custom images of different users. As the model detect that the driver is feeling drowsy, the system immediately triggers an alert to wake the driver. The alert is usually an alarm sound made by the device or any wirelessly triggered electronic device of the driver.

Model Accuracy Achieved = ~87.6%

FUTURE PROSPECTS

The model we have created works on the images of the participants which is feed to the ML model. As currently our database only consists of around 160 images (80 per user) which has led to the model being overfitted. We need to add more participants or load more video feed from online datasets to improve our accuracy.

REFERENCES

Drowsy driving statistics and facts 2022: https://www.bankrate.com/insurance/car/drowsy-driving-statistics/

Drowsiness Detection with Machine Learning: https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a

Real-Time Driver Drowsiness Detection using Computer Vision: https://www.ijeat.org/wp-content/uploads/papers/v11i1/A31591011121.pdf

Drowsiness Detector (Blink Detection) OpenCV Project Tutorial - Python and Dlib - with code https://www.youtube.com/watch?v=OCJSJ-anywc&t=305s

Deep Drowsiness Detection using YOLO, Pytorch and Python By Nicholas Renotte https://www.youtube.com/watch?v=tFNJGim3FXw

Driver Drowsiness Detection System with OpenCV & Keras https://data-flair.training/blogs/python-project-driver-drowsiness-detection-system/

APPENDIX A

```
import cv2
import dlib
from scipy.spatial import distance
# print(cv2.__version__)
# print(dlib.__version__)

def calculate_EAR(eye):
    A = distance.euclidean(eye[1], eye[5])
    B = distance.euclidean(eye[2], eye[4])
    C = distance.euclidean(eye[0], eye[3])
    ear_aspect_ratio = (A+B)/(2.0*C)
    return ear_aspect_ratio

cap = cv2.VideoCapture(0)
hog_face_detector = dlib.get_frontal_face_detector()
dlib_facelandmark =
dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

while True:
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = hog_face_detector(gray)
    for face in faces:

        face_landmarks = dlib_facelandmark(gray, face)
        leftEye = []
        rightEye = []

        for n in range(36,42):
            x = face_landmarks.part(n).x
            y = face_landmarks.part(n).y
            leftEye.append((x,y))
            next_point = n+1
            if n == 41:
                next_point = 36
            x2 = face_landmarks.part(next_point).x
            y2 = face_landmarks.part(next_point).y
            cv2.line(frame,(x,y),(x2,y2),(0,255,0),1)

        for n in range(42,48):
            x = face_landmarks.part(n).x
```

```
            y = face_landmarks.part(n).y
            rightEye.append((x,y))
            next_point = n+1
            if n == 47:
                next_point = 42
            x2 = face_landmarks.part(next_point).x
            y2 = face_landmarks.part(next_point).y
            cv2.line(frame,(x,y),(x2,y2),(0,255,0),1)

        left_ear = calculate_EAR(leftEye)
        right_ear = calculate_EAR(rightEye)

        EAR = (left_ear+right_ear)/2
        EAR = round(EAR,2)
        if EAR<0.26:
            cv2.putText(frame,"DROWSY!",(20,100),
                cv2.FONT_HERSHEY_SIMPLEX,3,(0,0,255),4)
            cv2.putText(frame,"Alert! feeling drowsy",(20,400),
                cv2.FONT_HERSHEY_SIMPLEX,2,(0,0,255),4)
            print("Drowsy")
        print(EAR)

    cv2.imshow("Alert! feeling drowsy", frame)

    key = cv2.waitKey(1)
    if key == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

APPENDIX B

1. Installing and Importing Dependencies

```
!pip install torch==1.8.1+cu111 torchvision==0.9.1+cu111 torchaudio===0.8.1 -f htt
```

```
In[]:
!git clone https://github.com/ultralytics/yolov5
!cd yolov5 & pip install -r requirements.txt
```

```
In[]:
import torch
from matplotlib import pyplot as plt
import numpy as np
import cv2
```

2. Load Model

```
In[]:
model = torch.hub.load('ultralytics/yolov5', 'yolov5s')
```

```
In[]:
#sigmoid linear unit
model
```

```
Out[]:
AutoShape(
  (model): DetectMultiBackend(
    (model): Model(
      (model): Sequential(
        (0): Conv(
          (conv): Conv2d(3, 32, kernel_size=(6, 6), stride=(2, 2), padding=(2,
2))
          (act): SiLU(inplace=True)
        )
        (1): Conv(
          (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1))
          (act): SiLU(inplace=True)
        )
        (2): C3(
          (cv1): Conv(
            (conv): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
```

```
          )
          (cv2): Conv(
            (conv): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv3): Conv(
            (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(32, 32, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
        (3): Conv(
          (conv): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
          (act): SiLU(inplace=True)
        )
        (4): C3(
          (cv1): Conv(
            (conv): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv3): Conv(
            (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
```

```
                (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
            (1): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
        (5): Conv(
          (conv): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
          (act): SiLU(inplace=True)
        )
        (6): C3(
          (cv1): Conv(
            (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv3): Conv(
            (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
```

```
            (1): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
            (2): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
        (7): Conv(
          (conv): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
          (act): SiLU(inplace=True)
        )
        (8): C3(
          (cv1): Conv(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv3): Conv(
            (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
```

```
            (cv2): Conv(
              (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
              (act): SiLU(inplace=True)
            )
          )
        )
      )
      (9): SPPF(
        (cv1): Conv(
          (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (cv2): Conv(
          (conv): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (m): MaxPool2d(kernel_size=5, stride=1, padding=2, dilation=1,
ceil_mode=False)
      )
      (10): Conv(
        (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
      )
      (11): Upsample(scale_factor=2.0, mode=nearest)
      (12): Concat()
      (13): C3(
        (cv1): Conv(
          (conv): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (cv2): Conv(
          (conv): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (cv3): Conv(
          (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (m): Sequential(
          (0): Bottleneck(
            (cv1): Conv(
              (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
              (act): SiLU(inplace=True)
            )
            (cv2): Conv(
              (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
```

```
              (act): SiLU(inplace=True)
            )
          )
        )
      )
      (14): Conv(
        (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
        (act): SiLU(inplace=True)
      )
      (15): Upsample(scale_factor=2.0, mode=nearest)
      (16): Concat()
      (17): C3(
        (cv1): Conv(
          (conv): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (cv2): Conv(
          (conv): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (cv3): Conv(
          (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (m): Sequential(
          (0): Bottleneck(
            (cv1): Conv(
              (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
              (act): SiLU(inplace=True)
            )
            (cv2): Conv(
              (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
              (act): SiLU(inplace=True)
            )
          )
        )
      )
      (18): Conv(
        (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
        (act): SiLU(inplace=True)
      )
      (19): Concat()
      (20): C3(
        (cv1): Conv(
          (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
```

```
          )
          (cv2): Conv(
            (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv3): Conv(
            (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
        (21): Conv(
          (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
          (act): SiLU(inplace=True)
        )
        (22): Concat()
        (23): C3(
          (cv1): Conv(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv3): Conv(
            (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
```

```
              (cv2): Conv(
                (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
        (24): Detect(
          (m): ModuleList(
            (0): Conv2d(128, 255, kernel_size=(1, 1), stride=(1, 1))
            (1): Conv2d(256, 255, kernel_size=(1, 1), stride=(1, 1))
            (2): Conv2d(512, 255, kernel_size=(1, 1), stride=(1, 1))
          )
        )
      )
    )
  )
)
```

3. Making Detections with Images

```
In [ ]:
# save test_image in the same directory to detect images in the image
img = 'test_image.jpg'

In [ ]:
results = model(img)
results.print()

In [ ]:
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()

In [ ]:
results.render()
```

4. Real Time Detections

```
In [ ]:
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()

    # Make detections
    results = model(frame)

    cv2.imshow('YOLO', np.squeeze(results.render()))

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

5. Training from scratch

```
In [ ]:
import uuid   # Unique identifier
import os
import time

In [ ]:
IMAGES_PATH = os.path.join('data', 'images') #/data/images
labels = ['awake', 'drowsy']
number_imgs = 5

In [ ]:
cap = cv2.VideoCapture(0)
# Loop through labels
for label in labels:
    print('Collecting images for {}'.format(label))
    time.sleep(5)

    # Loop through image range
    for img_num in range(number_imgs):
        print('Collecting images for {}, image number {}'.format(label,
img_num))

        # Webcam feed
        ret, frame = cap.read()
```

```
        # Naming out image path
        imgname = os.path.join(IMAGES_PATH, label+'.'+str(uuid.uuid1())+'.jpg')

        # Writes out image to file
        cv2.imwrite(imgname, frame)

        # Render to the screen
        cv2.imshow('Image Collection', frame)

        # 2 second delay between captures
        time.sleep(2)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
cap.release()
cv2.destroyAllWindows()
```

In [ ]:

```
print(os.path.join(IMAGES_PATH, labels[0]+'.'+str(uuid.uuid1())+'.jpg'))
```

In [ ]:

```
for label in labels:
    print('Collecting images for {}'.format(label))
    for img_num in range(number_imgs):
        print('Collecting images for {}, image number {}'.format(label,
img_num))
        imgname = os.path.join(IMAGES_PATH, label+'.'+str(uuid.uuid1())+'.jpg')
        print(imgname)
```

In [ ]:

```
!git clone https://github.com/tzutalin/labelImg
```

In [ ]:

```
!pip install pyqt5 lxml --upgrade
!cd labelImg && pyrcc5 -o libs/resources.py resources.qrc
```

In [ ]:

```
!cd yolov5 && python train.py --img 320 --batch 16 --epochs 500 --data
dataset.yml --weights yolov5s.pt --workers 2
```

6. Load Custom Model

In [ ]:

```
model = torch.hub.load('ultralytics/yolov5', 'custom',
path='yolov5/runs/train/exp15/weights/last.pt', force_reload=True)
```

In [ ]:

```
img = os.path.join('data', 'images', 'awake.c9a24d48-e1f6-11eb-bbef-5cf3709bbcc6.jpg')

In [ ]:
results = model(img)

In [ ]:
results.print()

In [ ]:
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
In [ ]:
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()

    # Make detections
    results = model(frame)

    cv2.imshow('YOLO', np.squeeze(results.render()))

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

ENGR 498-07 Research in Artificial Intelligence and Deep Learning

Driver-Drowsiness Detection model using Convolutional Neural Network

Team Members:

Rishav Chakraborty

Rahul Hembrom

Advisor: Dr. Jafar Saniie

Graduate Assistant: Mr. Xinrui Yu

Summer 2022

ABSTRACT

This project aims to build a real-time drowsiness detection model with the help of a Convolution Neural Network that can identify key attributes of drowsiness in the driver and triggers an alert when the driver is going to fall asleep. Computers and their algorithms have come a long way in successfully detecting the facial features of a human being, moreover, the fields of Computer Vision, Machine Learning, and Artificial Intelligence are growing rapidly. This paper aims to leverage the tools in these fields to make a Driver-Drowsiness detection model. The number of creators creating and adding pre-trained ML models to load and use from the Internet has skyrocketed. We plan on using an existing pre-trained YOLOv5s model (a lightweight model) on our dataset to make our system, from the YOLOv5 repository on GitHub. YOLOv5 is a family of object detection architectures and models pre-trained on the open-source COCO dataset containing around 80 different classes and represents Ultralytics open-source research into future vision AI methods. We are using the PyTorch ML framework to load the YOLO model and make detections along with OpenCV to access the webcam and render feeds.

INTRODUCTION

Drowsy driving refers to any instance of operating a vehicle while fatigued or sleepy. Even if the driver has not consumed alcohol, studies have shown the effects of driving while sleep-deprived are similar to those of drunk driving. These effects include impaired attention and coordination, slower reaction time, and poor judgment. It's believed that going 24 hours without sleep is comparable to having a blood-alcohol content of 0.10%, which exceeds the legal limit for driving in all 50 states.

According to the National Safety Council (NSC), drowsy driving accounts for over 100,000 crashes, 71,000 injuries, and 1,550 fatalities in the United States only. In another survey conducted by the AAA, 27% of drivers report driving while being so tired they have difficulty keeping their eyes open. In a rapidly developing world, more and more people will have access to private vehicles, and the accidents relating to drowsy driving are sure to go up. The existing models are extremely expensive and only work with high-end cameras and cars, or those which are affordable have low accuracy and are not robust. Hence it is imperative to develop an accurate model that detects the earliest signs of fatigue and drowsiness in the driver to alert him/her in time.

RELATED WORKS

1. Zhong G. December 14, 2019. Drowsiness Detection with Machine Learning. Towards Data Science

Objective: to detect extreme and visible cases of drowsiness along with softer signals of drowsiness as well.

Dataset: Real-Life Drowsiness Dataset from the University of Texas at Arlington.

Dataset Description: 30 hours of videos of 60 unique participants. Each video is 10 minutes long with around 240 frames being extracted from each video, i.e around 10560 frames to create the entire dataset.

Methodology: The paper used a 1-D CNN which extracted the EAR(Eye Aspect Ratio) and MAR(Mouth Aspect Ratio) as features to classify the object.

Accuracy: K-Nearest Neighbor (kNN, k = 25), Accuracy = 77.21% (BEST)
Naive Bayes, Accuracy = 57.75% (WORST)

2. Real-Time Driver Drowsiness Detection using Computer Vision, Mahek J., Bhavya B., Sowmyarani C. N., International Journal of Engineering and Advanced Technology, Volume-11 Issue-1

Objective: Detect a driver's drowsiness based on eyelid movement and yawning and reliably give appropriate voice alerts in real time.

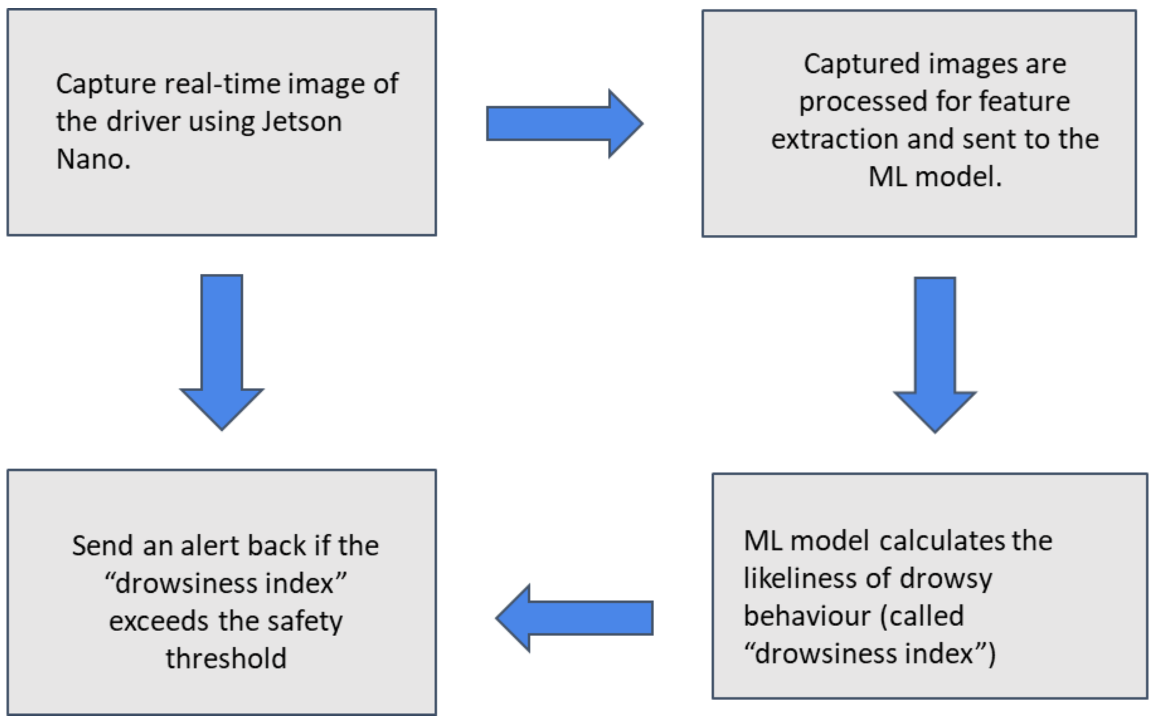
Dataset: iBUG-300w Dataset

Dataset Description: Open-Source dataset consisting of 300 indoor and as many outdoor images of human faces with different facial expressions.

Methodology: The face is localized using facial landmark detection. Then, shape prediction methods are used to detect important features by OpenCV built-in HAAR cascades, which are Pre-trained with EAR and MAR as the features.

Accuracy: As the model was trained on images the accuracy is over 95% possibly suggesting overfitting of data.

TECHNICHAL DESCRIPTIONS



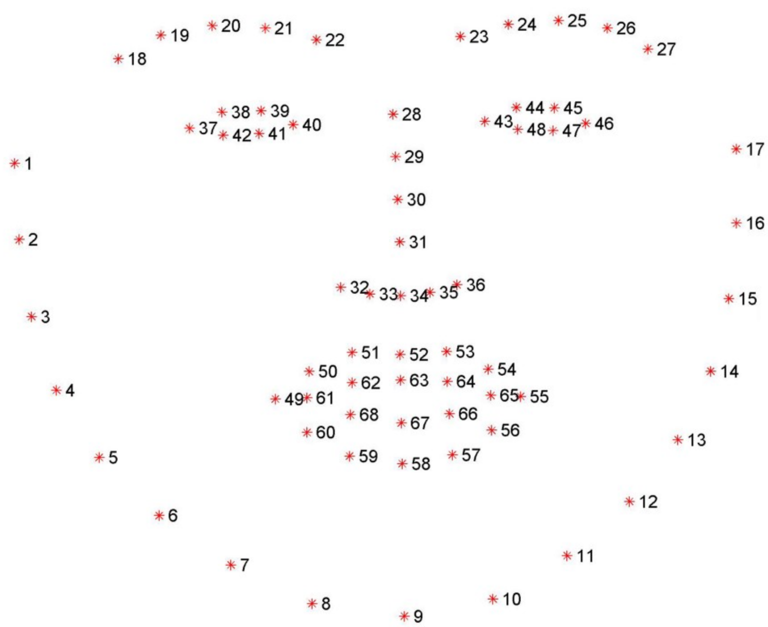
PROTOTYPE

The first version of the base prototype was our first attempt at creating a functioning Drowsiness detection model.

Component Required:

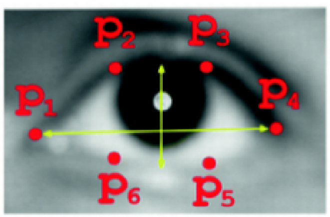
1. Python (Programming Language)
2. OpenCV (Image Processing Library)
3. Dlib (Extracting Facial Features)
4. Nvidia Jetson Nano
5. Microsoft Lifecam HD-3000

Dlib is an open source suite of applications and libraries written in C++. It offers a wide range of functionality across a number of machine learning sectors, including classification and regression, numerical algorithms such as quadratic program solvers, an array of image processing tools, and diverse networking functionality, among many other facets.



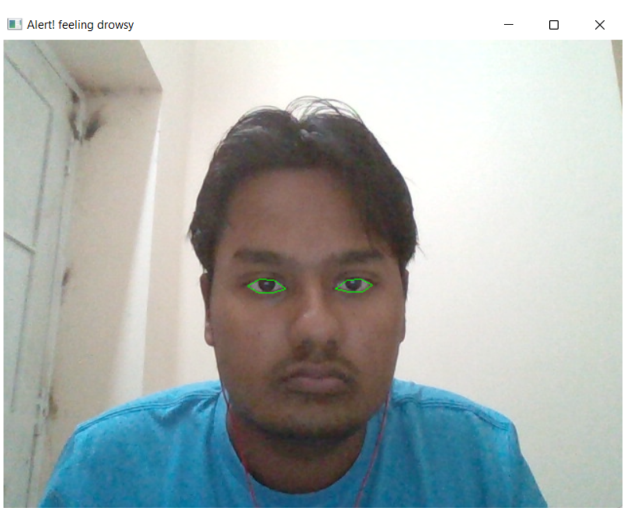
Dlib first detects the region of the face, then searches facial landmarks within the facial region to increase efficiency, it uses Histogram of Oriented Gradients (HOG) for Object Detection with a linear classifier, an image pyramid, and a sliding window detection scheme to detect faces in an image. Once the region of the face is determined, facial landmarks will be detected using One Millisecond Face Alignment with an Ensemble of Regression Trees.

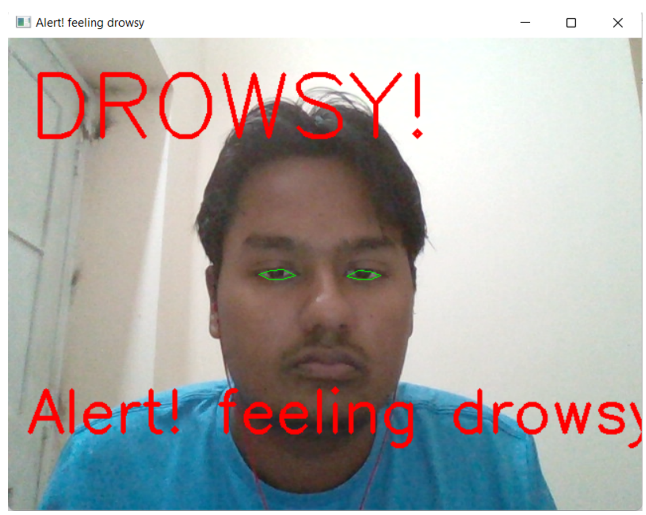
The prototype model consists of a Nvdia Jetson Nano, along with a monitor and USB webcam. The model calculates EAR(Eye Aspect Ratio) of the Left and Right Eyes separately by using the point extracted by the Dlib shown above. EAR, as the name suggests, is the ratio of the length of the eyes to the width of the eyes. The length of the eyes is calculated by averaging over two distinct vertical lines across the eyes as illustrated in the figure below.



$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Then the program averages the EAR values of the two eyes and checks it against the threshold value (See in Appendix A). The threshold value was manually set through trial and error, and when the value crosses the threshold. The alarm is set off.



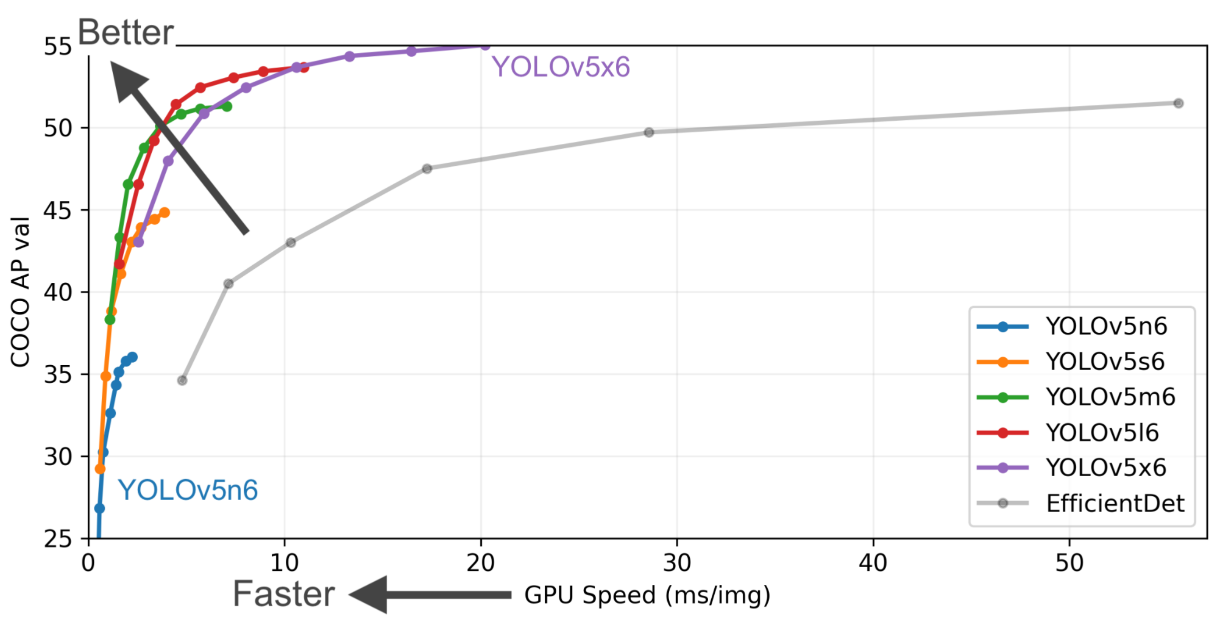


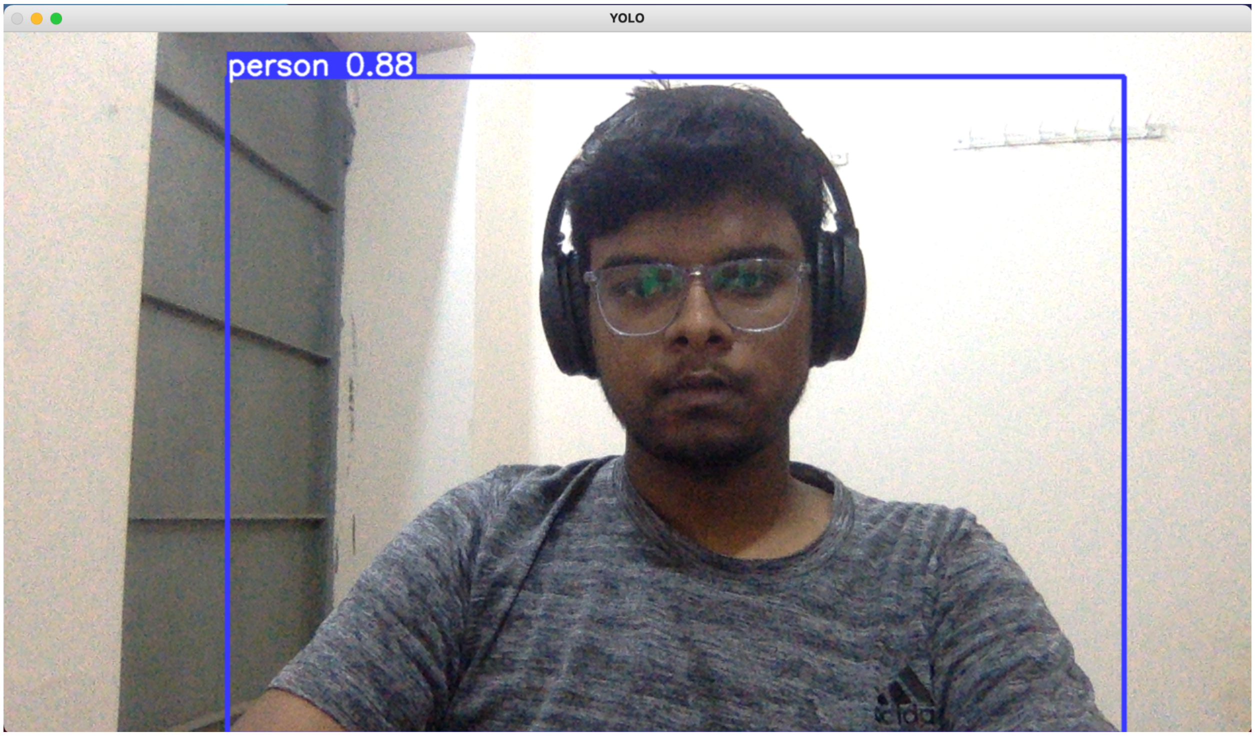
WORKING MODEL

Following are the components required for the Drowsiness Detection Model:

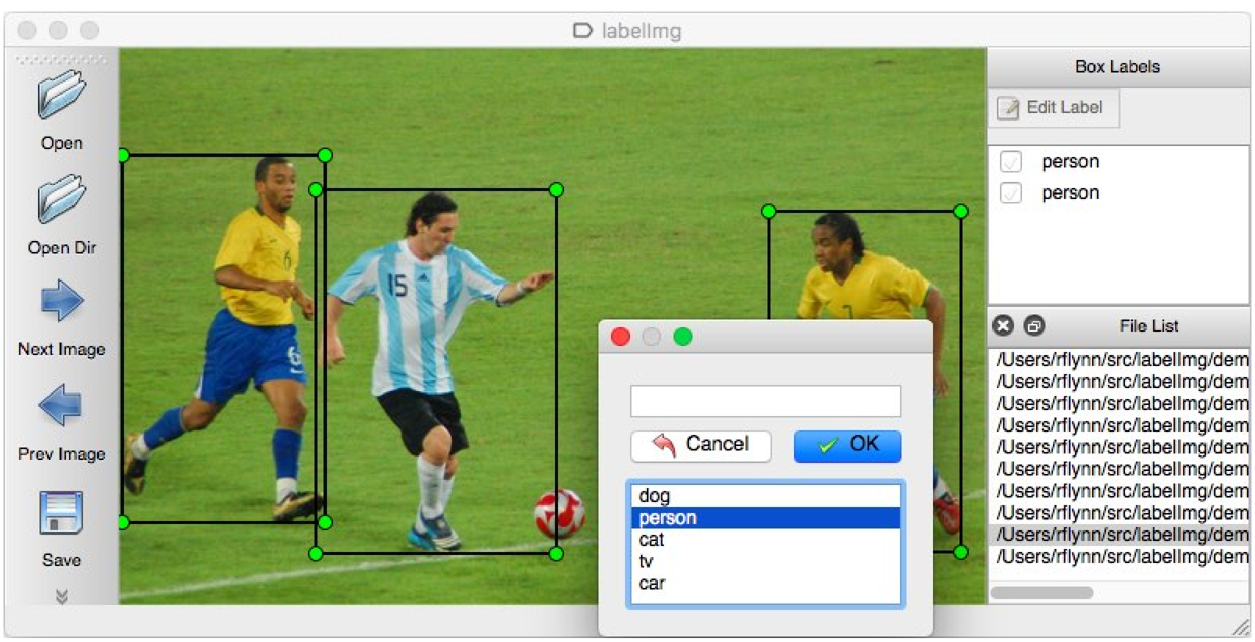
1. Python (Prgramming Language)
2. YOLOv5 (Image extracting and library)
3. LabelImg (A tool to convert images into vectors)
4. PyTorch (Machine Learning Library)
5. WebCamera (Capturing Image)

YOLOv5 is a family of object detection architectures and models pretrained on the COCO dataset, and represents Ultralytics open-source research into future vision AI methods, incorporating lessons learned and best practices evolved over thousands of hours of research and development. YOLOv5 consists of a pretrained CNN Model which helps in accurate detection of several objects which consists of human face aswell. Using this model improved the detection of faces and its features drastically. It uses a sigmoid linear unit (SiLU) as an activation function and an architecture with 19 convolution layers and 5 Max Pooling layers. Once the YOLOv5 CNN model was imported, the model was then tweaked by training the CNN such that it could detect the drowsiness amongst the person aswell.

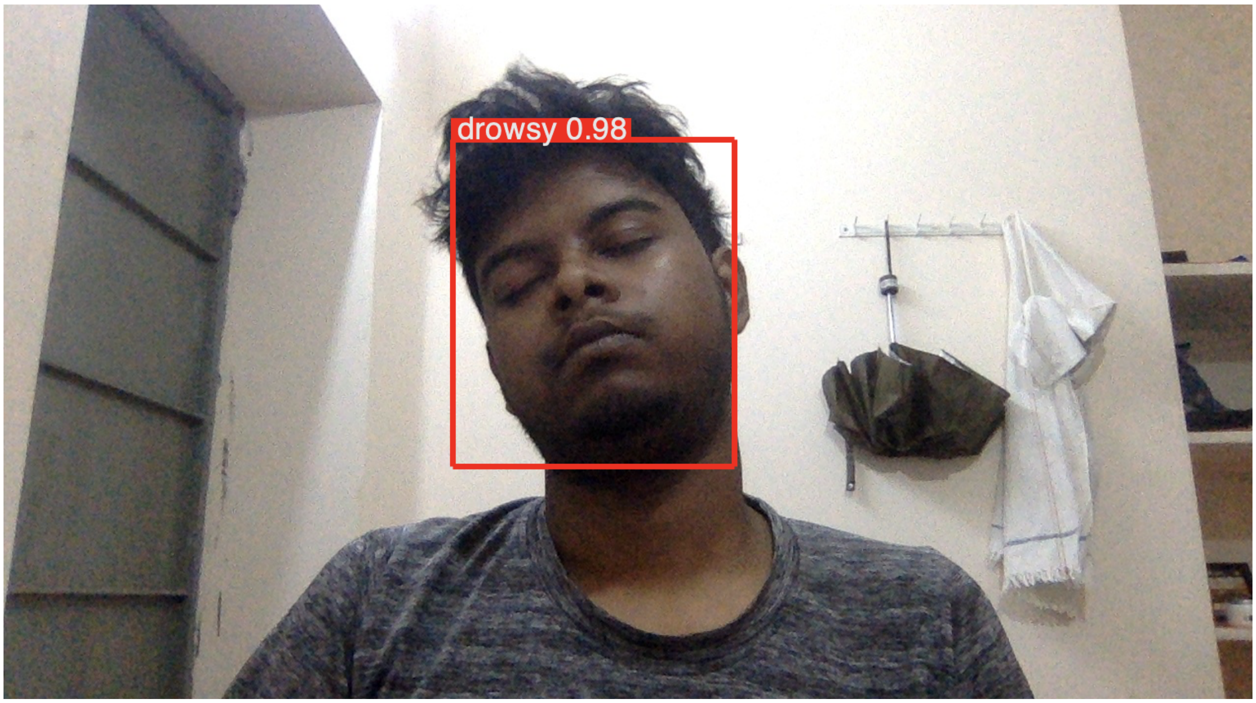




LabelImg is a github repository which consists of a python script which inputs the image in jpeg format and then converts it into a numeric array for the training dataset. In this case several frames were captured using a python script both in optimum light and as well as in dark, which then through the LabelImg were first cropped into the areas of focus (eyes, mouth, face, etc.), then converted into the format of array of arrays numbers, which is the default training input in YOLO Models along with the labels 'awake' or 'drowsy'.



After the retrieval of input datasets the imported CNN Model was trained with the new dataset with 320 image variations, 16 batches and 500 epochs, and the program written would run a python script consisting of the model which would detect the the facial expression of the person frame by and display the current state of the of the person, either awake or drowsy.



CONCLUSION

In conclusion our proposed system takes the frames from video feed of the driver and sends thoose frames to the YOLOv5 CNN model, which has been already pre-trained with custom images of different users. As the model detect that the driver is feeling drowsy, the system immediately triggers an alert to wake the driver. The alert is usually an alarm sound made by the device or any wirelessly triggered electronic device of the driver.

FUTURE PROSPECTS

The model we have created works on the images of the participants which is feed to the ML model. As currently our database only consists of around 160 images (80 per user) which has led to the model being ovefited. We need to add more participants or load more video feed from online datasets to improve our accuracy.

REFERENCES

Drowsy driving statistics and facts 2022:
https://www.bankrate.com/insurance/car/drowsy-driving-statistics/

Drowsiness Detection with Machine Learning:
https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a

Real-Time Driver Drowsiness Detection using Computer Vision:
https://www.ijeat.org/wp-content/uploads/papers/v11i1/A31591011121.pdf

Drowsiness Detector (Blink Detection) OpenCV Project Tutorial - Python and Dlib - with code
https://www.youtube.com/watch?v=OCJSJ-anywc&t=305s

Deep Drowsiness Detection using YOLO, Pytorch and Python By Nicholas Renotte
https://www.youtube.com/watch?v=tFNJGim3FXw

Driver Drowsiness Detection System with OpenCV & Keras
https://data-flair.training/blogs/python-project-driver-drowsiness-detection-system/

APPENDIX A

```python
import cv2
import dlib
from scipy.spatial import distance
# print(cv2.__version__)
# print(dlib.__version__)

def calculate_EAR(eye):
    A = distance.euclidean(eye[1], eye[5])
    B = distance.euclidean(eye[2], eye[4])
    C = distance.euclidean(eye[0], eye[3])
    ear_aspect_ratio = (A+B)/(2.0*C)
    return ear_aspect_ratio

cap = cv2.VideoCapture(0)
hog_face_detector = dlib.get_frontal_face_detector()
dlib_facelandmark =
dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

while True:
    _, frame = cap.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = hog_face_detector(gray)
```

```
for face in faces:

    face_landmarks = dlib_facelandmark(gray, face)
    leftEye = []
    rightEye = []

    for n in range(36,42):
        x = face_landmarks.part(n).x
        y = face_landmarks.part(n).y
        leftEye.append((x,y))
        next_point = n+1
        if n == 41:
            next_point = 36
        x2 = face_landmarks.part(next_point).x
        y2 = face_landmarks.part(next_point).y
        cv2.line(frame,(x,y),(x2,y2),(0,255,0),1)

    for n in range(42,48):
        x = face_landmarks.part(n).x
        y = face_landmarks.part(n).y
        rightEye.append((x,y))
        next_point = n+1
        if n == 47:
            next_point = 42
        x2 = face_landmarks.part(next_point).x
        y2 = face_landmarks.part(next_point).y
        cv2.line(frame,(x,y),(x2,y2),(0,255,0),1)

    left_ear = calculate_EAR(leftEye)
    right_ear = calculate_EAR(rightEye)

    EAR = (left_ear+right_ear)/2
    EAR = round(EAR,2)
    if EAR<0.26:
        cv2.putText(frame,"DROWSY!",(20,100),
            cv2.FONT_HERSHEY_SIMPLEX,3,(0,0,255),4)
        cv2.putText(frame,"Alert! feeling drowsy",(20,400),
            cv2.FONT_HERSHEY_SIMPLEX,2,(0,0,255),4)
        print("Drowsy")
    print(EAR)

cv2.imshow("Alert! feeling drowsy", frame)
```

```
    key = cv2.waitKey(1)
    if key == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

APPENDIX B

1. Installing and Importing Dependencies

```
!pip install torch==1.8.1+cu111 torchvision==0.9.1+cu111 torchaudio===0.8.1 -f htt
```

```
In[]:
!git clone https://github.com/ultralytics/yolov5
!cd yolov5 & pip install -r requirements.txt
```

```
In[]:
import torch
from matplotlib import pyplot as plt
import numpy as np
import cv2
```

2. Load Model

```
In[]:
model = torch.hub.load('ultralytics/yolov5', 'yolov5s')
```

```
In[]:
#sigmoid linear unit
model
```

```
Out[]:
AutoShape(
  (model): DetectMultiBackend(
    (model): Model(
      (model): Sequential(
        (0): Conv(
          (conv): Conv2d(3, 32, kernel_size=(6, 6), stride=(2, 2), padding=(2,
2))
          (act): SiLU(inplace=True)
```

```
        )
        (1): Conv(
          (conv): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1,
1))
          (act): SiLU(inplace=True)
        )
        (2): C3(
          (cv1): Conv(
            (conv): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(64, 32, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv3): Conv(
            (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(32, 32, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
        (3): Conv(
          (conv): Conv2d(64, 128, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
          (act): SiLU(inplace=True)
        )
        (4): C3(
          (cv1): Conv(
            (conv): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(128, 64, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv3): Conv(
```

```
          (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (m): Sequential(
          (0): Bottleneck(
            (cv1): Conv(
              (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
              (act): SiLU(inplace=True)
            )
            (cv2): Conv(
              (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
              (act): SiLU(inplace=True)
            )
          )
          (1): Bottleneck(
            (cv1): Conv(
              (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
              (act): SiLU(inplace=True)
            )
            (cv2): Conv(
              (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
              (act): SiLU(inplace=True)
            )
          )
        )
      )
      (5): Conv(
        (conv): Conv2d(128, 256, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
        (act): SiLU(inplace=True)
      )
      (6): C3(
        (cv1): Conv(
          (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (cv2): Conv(
          (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (cv3): Conv(
          (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (m): Sequential(
          (0): Bottleneck(
```

```
              (cv1): Conv(
                (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
            (1): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
            (2): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
        (7): Conv(
          (conv): Conv2d(256, 512, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
          (act): SiLU(inplace=True)
        )
        (8): C3(
          (cv1): Conv(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
```

```
          (cv3): Conv(
            (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
        (9): SPPF(
          (cv1): Conv(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(1024, 512, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): MaxPool2d(kernel_size=5, stride=1, padding=2, dilation=1,
ceil_mode=False)
        )
        (10): Conv(
          (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (11): Upsample(scale_factor=2.0, mode=nearest)
        (12): Concat()
        (13): C3(
          (cv1): Conv(
            (conv): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(512, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv3): Conv(
            (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
```

```
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
        (14): Conv(
          (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
          (act): SiLU(inplace=True)
        )
        (15): Upsample(scale_factor=2.0, mode=nearest)
        (16): Concat()
        (17): C3(
          (cv1): Conv(
            (conv): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(256, 64, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv3): Conv(
            (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
```

```
        (18): Conv(
          (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
          (act): SiLU(inplace=True)
        )
        (19): Concat()
        (20): C3(
          (cv1): Conv(
            (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(256, 128, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv3): Conv(
            (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(128, 128, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
        (21): Conv(
          (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(2, 2),
padding=(1, 1))
          (act): SiLU(inplace=True)
        )
        (22): Concat()
        (23): C3(
          (cv1): Conv(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (cv2): Conv(
            (conv): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
```

```
          (cv3): Conv(
            (conv): Conv2d(512, 512, kernel_size=(1, 1), stride=(1, 1))
            (act): SiLU(inplace=True)
          )
          (m): Sequential(
            (0): Bottleneck(
              (cv1): Conv(
                (conv): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
                (act): SiLU(inplace=True)
              )
              (cv2): Conv(
                (conv): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1),
padding=(1, 1))
                (act): SiLU(inplace=True)
              )
            )
          )
        )
        (24): Detect(
          (m): ModuleList(
            (0): Conv2d(128, 255, kernel_size=(1, 1), stride=(1, 1))
            (1): Conv2d(256, 255, kernel_size=(1, 1), stride=(1, 1))
            (2): Conv2d(512, 255, kernel_size=(1, 1), stride=(1, 1))
          )
        )
      )
    )
  )
)
```

3. Making Detections with Images

In [ ]:
```
# save test_image in the same directory to detect images in the image
img = 'test_image.jpg'
```

In [ ]:
```
results = model(img)
results.print()
```

In [ ]:
```
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
```

In [ ]:
```
results.render()
```

4. Real Time Detections

In [ ]:
```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()

    # Make detections
    results = model(frame)

    cv2.imshow('YOLO', np.squeeze(results.render()))

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

5. Training from scratch

In [ ]:
```
import uuid   # Unique identifier
import os
import time
```

In [ ]:
```
IMAGES_PATH = os.path.join('data', 'images') #/data/images
labels = ['awake', 'drowsy']
number_imgs = 5
```

In [ ]:
```
cap = cv2.VideoCapture(0)
# Loop through labels
```

```
for label in labels:
    print('Collecting images for {}'.format(label))
    time.sleep(5)

    # Loop through image range
    for img_num in range(number_imgs):
        print('Collecting images for {}, image number {}'.format(label,
img_num))

        # Webcam feed
        ret, frame = cap.read()

        # Naming out image path
        imgname = os.path.join(IMAGES_PATH, label+'.'+str(uuid.uuid1())+'.jpg')

        # Writes out image to file
        cv2.imwrite(imgname, frame)

        # Render to the screen
        cv2.imshow('Image Collection', frame)

        # 2 second delay between captures
        time.sleep(2)

        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
cap.release()
cv2.destroyAllWindows()
```

In [ ]:

```
print(os.path.join(IMAGES_PATH, labels[0]+'.'+str(uuid.uuid1())+'.jpg'))
```

In [ ]:

```
for label in labels:
    print('Collecting images for {}'.format(label))
    for img_num in range(number_imgs):
        print('Collecting images for {}, image number {}'.format(label,
img_num))
        imgname = os.path.join(IMAGES_PATH, label+'.'+str(uuid.uuid1())+'.jpg')
        print(imgname)
```

In [ ]:

```
!git clone https://github.com/tzutalin/labelImg
```

In [ ]:

```
!pip install pyqt5 lxml --upgrade
!cd labelImg && pyrcc5 -o libs/resources.py resources.qrc
```

In [ ]:
```
!cd yolov5 && python train.py --img 320 --batch 16 --epochs 500 --data dataset.yml --weights yolov5s.pt --workers 2
```

6. Load Custom Model

In [ ]:
```
model = torch.hub.load('ultralytics/yolov5', 'custom', path='yolov5/runs/train/exp15/weights/last.pt', force_reload=True)
```

In [ ]:
```
img = os.path.join('data', 'images', 'awake.c9a24d48-e1f6-11eb-bbef-5cf3709bbcc6.jpg')
```

In [ ]:
```
results = model(img)
```

In [ ]:
```
results.print()
```

In [ ]:
```
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
```
In [ ]:
```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()

    # Make detections
    results = model(frame)

    cv2.imshow('YOLO', np.squeeze(results.render()))

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Automatic Segregation of Bottles

Sakshat Jain, Aswin Sankaranarayanan
Carnegie Mellon University (ISRE 2022)
Summer internship report

Abstract: Out of the entire waste produced in India, 25% are dry waste components that can be recycled. In the process of recycling there are many steps and one such step is segregation of waste on the basis of material type and shape. The process of segregation between materials is very time consuming. In this paper, we have focused on how to reduce this time. We have taken a very specific subset of how to segregate plastic bottles and glass bottles since they can be confused while segregating because both have almost the same reflection properties. In this paper, we combat the aforementioned problem by using a deep learning method (object detection) to segregate between the two types of bottles. Our main contribution is to segregate between these two new categories of bottles which were not present earlier using supervised and semi-supervised learning.

1. Introduction

India generates around 62 million tons of waste with an average annual growth rate of 4%. It is also found that currently, India generates 70 million metrics of municipal solid wastes(MSW). Out of it, only 20% is recycled and the rest ends up in landfills and oceans affecting humans, and marine life, along with destroying the environment [1]. The situation is alarming and needs everyone to start taking immediate action to deal with this waste problem and necessitates a solid waste management system in place. This recyclable waste, dumped into landfills due to a lack of proper collection and infrastructure, can be reused as raw material. If it is properly segregated and processed further, it can be a highly lucrative source of revenue.

Waste can be broadly classified into two types: biodegradable and non-biodegradable. Biodegradable waste decomposes easily and it can be dumped into landfills. But non-biodegradable waste, like plastic and glass, takes millions of years to decompose and so it is preferable to recycle or reuse them. Plastic, in particular, has become one of the most pressing environmental issues that we face today. India generates roughly 3.5 million tonnes of plastic waste annually and the per capita plastic waste generation has nearly doubled over the last five years [2]. India is also generating around 1 million tonnes of glass waste [3]. Recycling of plastic and glass is one method for reducing environmental impact and resource depletion.

Not all non-biodegradable waste can be recycled, for this, we have to first do segregation between recyclable and non recyclable waste. After this segregation is done on the basis of material type like plastic, glass, iron, copper, etc. The process of segregation between materials is very time consuming because it is done manually by humans. And this makes the whole process of recycling much slower. To make this process faster we proposed some ideas in this research paper.

This paper focuses on how to make the process of segregation fast and automatic with the help of computer vision and deep learning. We have taken a very specific subset of how to segregate plastic bottles and glass bottles. We have chosen this subset because these two bottles can be confused very easily since they have very similar interaction with light. In the paper we have used some object detection techniques to segregate between plastic bottles and glass bottles.

1.1 Prior Work

Discussing the prior works which helped us to build our segregation network. Many of the initial works consist of bottles as a single category and their dataset didn't focus on the problem of segregation. Since their dataset didn’t consist of bottles regarding our problem, we created our own dataset which focuses on segregation problems. To make our Segregation Network we have used some prior works in the field of object detection. We have built our models on the detectron2 framework [4] using faster-rcnn [5] and retinanet [6] as our base networks.

1.1.1 Detectron2

Detectron2 [4] is Facebook AI Research's next generation library that provides state-of-the-art detection and segmentation algorithms. It supports a number of computer vision research projects. Detectron2 consists of a model zoo from which we have added the two state-of-the-art algorithms faster-rcnn and retinanet into our own network. We have chosen this framework since it was very easy to train and test different types of state-of-the-art models. They have a very easy to use and customizable library of functions.

1.1.2 Faster-RCNN

Faster-RCNN [5] is a state-of-the-art two stage object detection framework based on deep convolutional networks, which includes a Region Proposal Network (RPN) and an object detection network. Both networks are trained for sharing convolutional layers for fast testing. We have used this model as our base model. This model is trained on coco dataset which consists of only one category of bottle , but since we required two categories of bottle we created our own dataset and used this faster rcnn network to segregate between the bottles. We have used this model since it has an AP greater than its competitor and is faster than its previous generations.

1.1.3 Retinanet

RetinaNet [6] is one of the best one-stage object detection models that has proven to work well with dense and small scale objects. RetinaNet has been formed by making two improvements over existing single stage object detection models - Feature Pyramid Networks (FPN) and focal loss. We have used this model since it can handle class imbalance because it uses focal loss instead of cross-entropy Loss. And it has also faster testing time then faster rcnn.

1.2 Contributions

In this work, we propose a segregation network, which focuses on segregating the plastic bottle and glass bottle. Our main contributions are as follows:

- We have created the dataset from scratch consisting of RGB images of small, medium and large plastic and glass bottles, labeled using labelme software. Created the coco annotations of all training images.

- Our segregation network based on faster-rcnn has given average precision (AP) of 47.5% and AP50 (AP with IOU=0.5) of 69.2%.

- Created a semi-supervised network on the top of our segregation network which labels the testing images in the run time and adds it to the training network. This model has increased our AP to 51.8% and AP50 to 72.5%.

This process of segregation can extend to the ground level by deploying robots with this computer vision technique, so as to make this process automatic. We can extend this segregation method in the future to other categories as well, such as plastic bags, steel bottles, etc.

1.3 Limitations

Although Segregation Network provides us a decent accuracy, our method has some limitations.

- First, due to the fact that we are using a learning based algorithm, we are restricted to segregate in the image intensity space where physical factors are not measured.
- Second, since we have limited time, we were only able to annotate some images, the accuracy can be increased further if we were able to annotate more images.

2. Background and Setup

The goal of this paper is to provide a method to segregate between plastic bottles and glass bottles. Specifically, the model will take an image as an input and our goal is to give an output image containing bounding boxes for different categories of bottles.

2.1 Challenges

First, we face the challenge of creating the dataset, since we have to create it from scratch, so it consumes a lot of our time in getting the right dataset to work with. We first tried fitting our model to small datasets, so to make sure that we can continue with that data. In this process many times we modified our dataset.

Second, sometimes it was very difficult for us to label the images, since both types of bottles have almost the same transparency effects. So, to tackle this the model has to focus more on reflection and refraction properties of both the bottles to segregate between them.

2.2 Dataset

We have curated a dataset of real world images of plastic bottles and glass bottles. We have captured some images from our mobile camera but those images were not enough so then we crawled some images from the web.

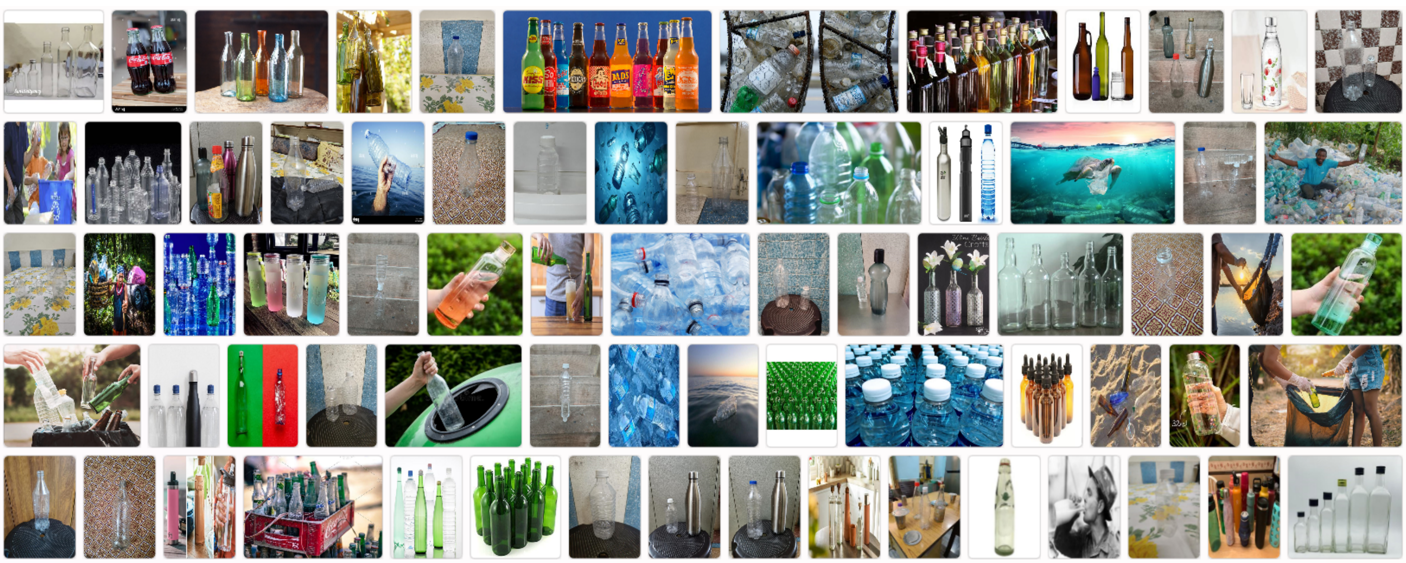


Figure 1: Training dataset images

Images contain bottles of various sizes, colors and transparency. Images contain bottles with various backgrounds like garbage, ocean, humans. etc. We have applied various augmentation to the dataset like cropping , rotation, random brightness, flip, etc.

We labeled the images using labelme software, which helped us to generate annotations in the form of coco dataset. We created two categories namely “plastic_b”(for plastic bottles) and “glass_b”(for glass bottles). The Labeling process was the most time consuming since we have to create the bounding boxes with max precision and accuracy.

We divided the dataset into 3 parts: train, validation and test. In train and validation sets we have images with their annotations and in the test set we have no annotations. In the train set we have 194 images and have 543 bounding boxes for plastic bottles and 543 bounding boxes for glass bottles. In the validation set we have 43 images with 123 bounding boxes for plastic bottles and 124 bounding boxes for glass bottles.

3. Methods

We have proposed some methods which help to segregate between plastic bottles and glass bottles.

3.1 Segregation Model based on Faster RCNN

Segregation model is a custom object detection model which uses faster-rcnn as its network and segregates between the two

- Network Input: As mentioned above we used images with two categories (plastic_b, glass_b). Now before giving images to the network we applied some augmentations like cropping , rotation, random brightness, flip, etc. The images were then given to the resnet [7] (backbone) network to convert the images into feature maps.
- Network Architecture: We have built a custom object detection model using faster-rcnn. faster-rcnn is a 2 stage detector. In this we have used the backbone of the network as Resnet. The backbone is used to generate feature maps. Then we have an RPN layer that is used to pool features from the backbone feature map. Then a fast-rcnn network is used as a detector for the model.
- Training: Our implementation of the model is in python using the Detectron2 framework. Training until convergence ends around 100 epochs with a learning rate of 8e-5. We run all of our experiments on 2 NVIDIA TITAN Xp GPUs.
    - The input image is first passed through the backbone CNN to get the feature map.
    - Next, the bounding box proposals from the RPN are used to pool features from the backbone feature map. This is done by the ROI pooling layer. The ROI pooling layer, in essence, works by
      a) Taking the region corresponding to a proposal from the backbone feature map.
      b) Dividing this region into a fixed number of sub-windows.
      c) Performing max-pooling over these sub-windows to give a fixed size output.
    - The output from the ROI pooling layer passes through 2 fully connected layers. After passing them through two fully connected layers, the features are fed into the sibling classification and regression branches.
    - The features are passed through a softmax layer to get the classification scores — the probability of a proposal belonging to each class. The regression layer coefficients are used to improve the predicted bounding boxes.
    - Loss Functions: Faster RCNN uses three loss functions:
      1) RPN Loss [8]: Loss function of Regional Proposal Network is the sum of classification (cls) and regression (reg) loss. The classification loss is the entropy loss on whether it's a foreground or background. The regression loss is the difference between the regression of the foreground box and that of the ground truth box. Equation which is used in the RPN network:

      $$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*)$$

      2) Cross Entropy Loss [9]: Categorical Cross-Entropy loss is traditionally used in classification tasks. As the name implies, the basis of this is Entropy. In statistics, entropy refers to the disorder of the system. It quantifies the degree of uncertainty in the model’s predicted value for the variable. The sum of the entropies of all the probability estimates is the cross entropy. This loss is used as classification loss to fine tune the classes.

$$Entropy = -p_i log_b(p_i)$$

$$CrossEntropy = -\sum_{i=1}^{i=n} Y_i log_b(p_i)$$

3) Smooth L1 Loss [10]: Creates a criterion that uses a squared term if the absolute element-wise error falls below beta and an L1 term otherwise. It is less sensitive to outliers than MSE Loss and in some cases prevents exploding gradients; this is used by regressor for bounding box calculation.
For a batch of size N, the unreduced loss can be described as:

$$l(x, y) = L = \{l_1, ..., l_N\}^T$$

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases}$$

If reduction is not none, then:

$$\ell(x, y) = \begin{cases} \text{mean}(L), & \text{if reduction} = \text{'mean'}; \\ \text{sum}(L), & \text{if reduction} = \text{'sum'}. \end{cases}$$

- Output: We were able to segregate between both the bottles with an AP50 of 70.2.

Input Image ***Detections***

Figure 2: Shows detection of bottles made by Segregation model

- Results: First we tested our model using a smaller dataset of 80 training images in which we have 210 instances of plastic bottles and 205 instances of glass bottles, and 20 validation images which contain 43 instances of plastic bottles and 40 instances of glass bottles. After training for 100 epochs we get the following results:

AP	AP50	AP75	AP_S	AP_M	AP_L
44.2	65.5	50.1	12.1	32.4	51.9

Table 1: Evaluation results for custom faster-rcnn model with smaller dataset

After this, since we were getting good results so we then run the model on the full dataset, and after training for 100 epochs we get the following results:

AP	AP50	AP75	AP_S	AP_M	AP_L
47.5	69.2	49.8	20.0	35.2	58.4

Table 2: Evaluation results for custom faster-rcnn model with full dataset

Now, looking at the above two results we can conclude that if we increase our dataset, we can definitely increase the performance of our model. But due to time constraints we cannot label more images, so to solve this limitation we tried to build a semi-supervised detection model which we discussed in (3.3) section.

3.2 Model with faster detection rate

The detection time per image of the above model is around 170 milliseconds. Now to increase the detection speed we tried to build the custom detection model over Retinanet. RetinaNet has been formed by making two improvements over existing single stage object detection models :

Feature Pyramid Networks (FPN) and Focal Loss.

- Network Architecture: We have built a custom object detection model using Retinanet. There are four major components of a RetinaNet model architecture:

  a) Bottom-up Pathway - The backbone network (ResNet) which calculates the feature maps at different scales, irrespective of the input image size or the backbone.
  b) Top-down pathway and Lateral connections - The top down pathway up samples the spatially coarser feature maps from higher pyramid levels, and the lateral connections merge the top-down layers and the bottom-up layers with the same spatial size.
  c) Classification subnetwork - It predicts the probability of an object being present at each spatial location for each anchor box and object class.
  d) Regression subnetwork - It regresses the offset for the bounding boxes from the anchor boxes for each ground-truth object.

- Focal Loss [9]: it is an enhancement over Cross-Entropy Loss (CE) and is introduced to handle the class imbalance problem with single-stage object detection models. Single Stage models suffer from an extreme foreground-background class imbalance problem due to dense sampling of anchor boxes (possible object locations). In RetinaNet, at each pyramid layer there can be thousands of anchor boxes. Only a few will be assigned to a ground-truth object while the vast majority will be background class. These easy examples (detections with high probabilities) although resulting in small loss values can collectively overwhelm the model. Focal Loss reduces the loss contribution from easy examples and increases the importance of correcting misclassified examples

$$FL(p_t) = -(1-p_t)^{\gamma} log(p_t)$$

● Output: We were able to segregate between both the bottles with an AP50 of 52.2, this is less than the above model but detection speed is faster than the above model which was the main motivation while building this model. The detection speed increased to 61 ms per image.

● Results: We have chosen AP as our evaluation metric.

AP	AP50	AP75	AP_S	AP_M	AP_L
40.7	64.4	42.8	20.7	34.6	44.7

Table3: Evaluation results for custom retinanet model

Segregation Model ***Custom Retinanet Model***

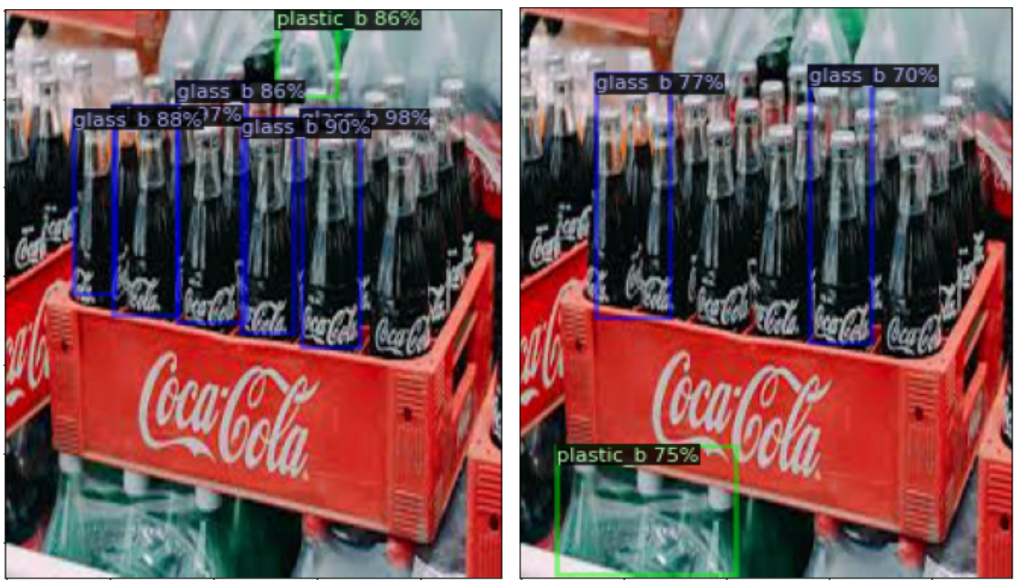


Figure 3: Comparing the outputs of above 2 models

3.3 Semi Supervised Network

Now to overcome the limitation of the dataset, we tried to build a semi-supervised model. First model (segregation network [3.1]) is the base of this semi-supervised technique.

● Training: Our implementation of the model is in python using the Detectron2 framework. Training until convergence ends around 500 epochs with a learning rate of 8e-5.. We run all of our experiments on 2 NVIDIA TITAN Xp GPUs.

1. In this we have created a data sink, which has 1000 unlabelled images. We use this data sink during our training. We can make this data sink autonomous by adding more images automatically.
2. Training first starts with running the first model (segregation network [3.1]) for 200 epochs.
3. After this the model makes detections for 50 images randomly selected from the data sink.
4. Once the detections are generated for the 50 images, the model creates annotations for these detections. After this the images with their annotations are added to the training dataset.
5. And that’s how the model is able to increase the dataset. Model then again trains for another 50 epochs.
6. Now the model will follow the steps from c to e until all the images from the data sink are added to the training dataset.

- Output: We were able to increase the AP50 metric to 73.5. This shows that we were able to successfully add more images to our dataset while training.
- Results: We have chosen AP as our evaluation metric

AP	AP50	AP75	AP_S	AP_M	AP_L
51.8	72.5	55.1	20.0	37.9	63.2

Table 4: Evaluation results for Semi Supervised Model

- Conclusions: In the above model we were able to increase the precision but there is a limitation to it. Since we were adding new images in the dataset while training the model, we were not removing any false negatives which can be generated by our running model , which then can lead to making these false negatives as true positives for the next iterations of the model.

  Now to remove these limitations we tried to implement a “Human in the Loop” model.So basically after the above model has detected bounding boxes for 50 data sink images , we will manually remove all the false negatives from the images and only after this the model will add these images and there annotations to the training datasets. So using this model we will be able to decrease false negatives in the model.

4. Results

For comparing different models we have chosen AP as our evaluation metric. AP is a way to summarize the precision-recall curve into a single value representing the average of all precisions. The AP is calculated according to the below equation. Using a loop that goes through all precisions/recalls, the difference between the current and next recalls is calculated and then multiplied by the current precision. In other words, the AP is the weighted sum of precisions at each threshold where the weight is the increase in recall.

$$AP = \sum_{k=0}^{k=n-1} [Recalls(k) - Recalls(k + 1)] * Precisions(k)$$

$$Recalls(n) = 0, Precisions(n) = 1$$

$$n = Number\ of\ thresholds.$$

In short, AP compares the ground-truth bounding box to the detected box and returns a score. The higher the score, the more accurate the model is in its detections.

Comparing the results of all three models in the below table.

Methods	AP	AP50	AP75	AP_S	AP_M	AP_L	FPS
Segregation Network	47.5	69.2	49.8	20.0	35.2	58.4	6
Custom Retinanet	40.7	64.4	42.8	20.7	34.6	44.7	17
Semi Supervised	51.8	72.5	55.1	20.0	37.9	63.2	6

Table 5: Comparison of different methods on the same dataset. FPS was calculated on the same machine

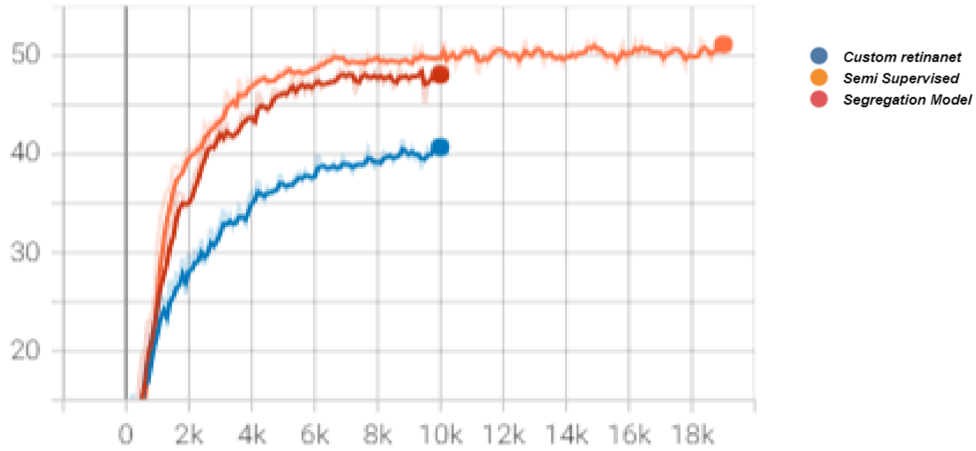


Figure 4: Comparison of AP of all 3 methods

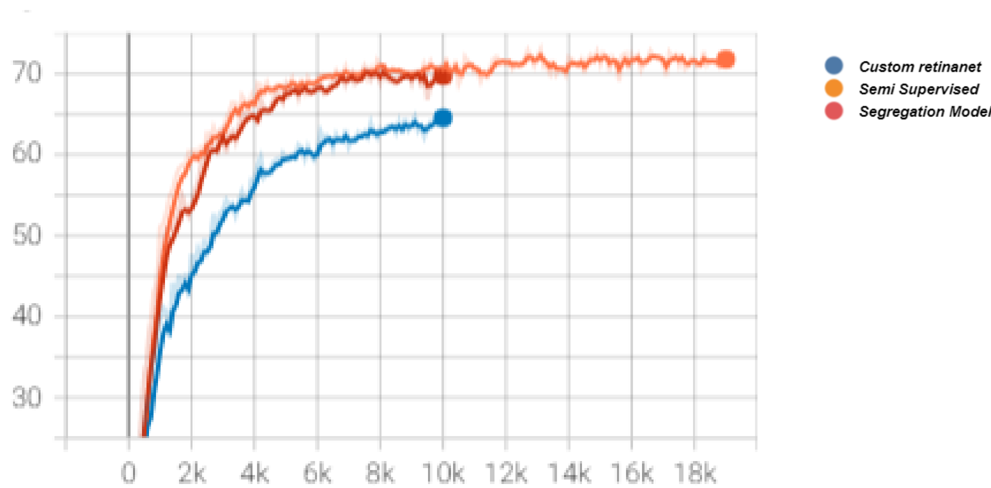


Figure 5: Comparison of AP50 of all 3 methods

5. Discussions

In this work, we presented three methods to make the process of segregation of plastic bottles and glass bottles fast and automatic. To make this work on ground, we have to deploy robots installed with cameras, hands and our object detection model. These robots can scan the garbage using the camera and can detect different types of bottles based on our model and using hands they can segregate them. This will make this process automatic and independent of humans and thereby making it fast.

In the future we can increase the capabilities of our model by training it on more categories of waste like plastic bags, steel bottles,etc. Which will make this whole process of segregation very fast and it will be no more a bottleneck in the whole process of recycling.

The accuracy of the semi-supervised model can be increased in the future, if we can somehow delete the false negatives which were produced in the intermediate steps of the model. “Human in the loop” can be one such method which can remove those false negatives. In the future we can deduce more techniques to solve this problem.

6. References

[1] https://timesofindia.indiatimes.com/blogs/voices/recycling-waste-can-generate-crores-in-revenue-in-india/
[2]https://economictimes.indiatimes.com/industry/environment/india-generates-3-5-million-tonnes-plastic-waste-annually-environment-minister/articleshow/90668558.cms?from=mdr
[3] https://www.sciencedirect.com/science/article/abs/pii/S0921344915300082
[4] https://ai.facebook.com/tools/detectron2/
[5] https://arxiv.org/abs/1506.01497
[6] https://arxiv.org/abs/1708.02002
[7] https://arxiv.org/abs/1512.03385
[8] https://towardsdatascience.com/review-faster-r-cnn-object-detection-f5685cb30202
[9] https://towardsdatascience.com/focal-loss-a-better-alternative-for-cross-entropy-1d073d92d075
[10] https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html
[11] https://blog.paperspace.com/mean-average-precision/

IMMERSIVE SUMMER RESEARCH EXPERIENCE - ISRE 2022 (CARNEGIE MELLON UNIVERSITY)

SMART DOOR USING FACIAL VERIFICATION

SETIKA RAJ
with
Prof. Aswin Sankaranarayanan

Abstract: *Security is now a prime concern for any individual in modern days. The aim of this project is to make every home secure and build modern, easily-used smart door lock systems. This smart door lock system is user friendly, convenient, allowing users to unlock their home door in a better secure way. The term "face recognition and detection" is like an ocean of research and innovation with the applications of image analysis and algorithm based understanding which can be called as computer vision. Facial recognition involves the detection and identification of the image. It uses an image capturing technique in the system. The camera catches the facial picture and compares it with the image which is stored in the database. If the picture is matched with the database the gate will open. This prototype system works on a face verification algorithm where a threshold is defined to verify if the detected face matches with a given face or not. In order to get an accurate and clear picture of an intruder, the Haar classifier method is used for face detection. In the process, different pretrained models are used to achieve the most accurate result. Our investigations show that the use of deep neural models provide extremely high verification rates while keeping false alarms to a minimum.*

1. <u>Introduction</u>

Face recognition is a biometric solution designed for the purpose of recognizing a human face without any physical contact required. It is an advanced feature in digital locks where people can easily unlock the door using their faces. Smart door-lock systems based on this technology provide access to who can come inside and we can get an alert message for an unknown face. With the advancing need of strengthening security systems, face recognition provides an easy , contactless and accurate way out. The motivation of this project is to use the face detection and face recognition techniques to develop an advanced and robust algorithm that can provide home security more efficiently.

Performing the face verification process, i.e, if the given face is similar to an already known face in the dataset, the conventional method used is the knn algorithm. The k in kNN represents the number of the nearest neighbors we used to classify new data points.Euclidean or Cosine distance are used to calculate the nearest neighbor.

The receiver operating characteristic (ROC) curve is frequently used for evaluating the performance of binary classification algorithms. It provides a graphical representation of a classifier's performance, rather than a single value like most other metrics. The ROC curve here is produced by calculating and plotting the false alarm (unknown face detected as correct) vs misdetect (a known face not detected) graph on varying thresholds. The plot produces an exponentially decreasing graph. The lesser the area under the graph, the more accurate is the algorithm used. Through the plot, the optimum threshold is obtained at which the rate of misdetect and false alarm is minimum and the verification is then performed, if the similarity parameter is below the threshold the test face is identified to be correct. To test in real-time, open-cv is used and the person's face before the webcam is tested.

2. <u>Methods</u>

Face verification is the task of comparing a candidate face to another, and verifying whether it is a match. It is a one-to-one mapping: we check if this person is the correct one. Here, my face is used for obtaining results. The ROC plot helps in setting the optimum boundary to verify if the given face is me or not.

To start with , training and negative datasets are created which contains face images of size 100*100*3 as detected by the opencv-haarcascade method of mine and other people in my locality respectively. Another set of test dataset is created consisting of my images to calculate similarity metrics and perform a comparison. To perform the similarity, euclidean and cosine distances are used.To start with a reference, KNN algorithm is used. Each image is vectorised initially. Euclidean and cosine distances of each test dataset image is calculated from each image of training and negative datasets respectively. The average of distances for each test image from training and negative datasets respectively is computed. Varying thresholds over a range , ROC curves(false alarm vs misdetect) are plotted. The plot gives an optimum threshold which can be further used for the verification process.

For pretrained models, the same algorithm is used but instead of directly vectorising the image matrix,it involves calculating a face embedding for a new given face and comparing the embedding to the embedding for the face known to the system. A face embedding is a vector that represents the features extracted from the face. This can then be compared with the vectors generated for other faces. Some pretrained models used here are :Deepface,Arcface,Facenet,VGGFace. *On observation,VGGFace is found to give the most accurate results.*

3. <u>Results</u>

The ROC curves for different algorithms used are as below:-

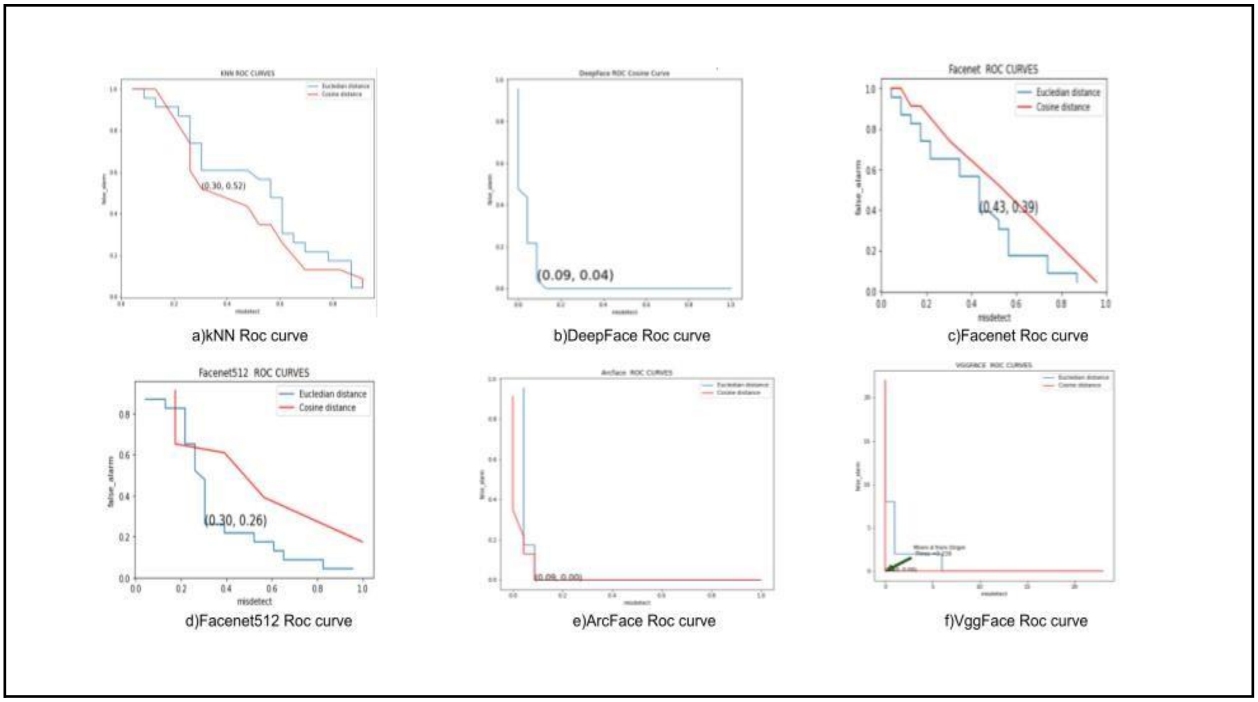


Fig 1. ROC PLOTS FOR DIFFERENT PRE-TRAINED MODELS. Each plot also shows a comparison between euclidean and cosine curves.

As observed from the ROC curves:

- **VGGFACE** gives the best result.
- Cosine ROC curves give a lesser area of convergence(more accurate).

4. <u>Discussion</u>

We highlight some of the key implications of the results presented in Section 3.

In face recognition literature, it is common to use a class of techniques called Convolutional neural networks (CNN), which themselves fall under the broad umbrella of so called artificial neural networks. A CNN has one or more convolution layers and are used mainly for image processing, classification. CNNs develop an internal representation of a two- dimensional image which allows the model to learn position and scale of faces in an image.

An image is nothing but a 2-dimensional array. Before training an image, we need to process the dataset, i.e, converting each image into a NumPy array.

Neural networks are like layers where each layer of the neural network contains nodes which calculates some values based on characteristics or weights. A typical neural network has a collection of specialized layers—for example, convolution, max pooling and flattening, which we briefly discuss next.

- *Convolution layer* is a fundamental mathematical operation that is highly useful to detect features of an image. In this layer we pass kernel. i.e., n*n matrix over the image pixel. Kernel has values in each cell. It processed with the original image helps to produce some characteristics which help to identify images of the same object while predicting.

- *Max Pooling* operation involves sliding a 2- dimensional filter over each channel of features map and extract maximum features from the image. Pooling layer reduces the number of parameters to learn and amount of computation to perform.
- *Flattening* operation is performed when we get multidimensional output and want to convert it into a single long continuous linear vector.The flattened matrix is fed as input to the fully connected layer.

Once the image is convolved, pooled and flattened, the result is a vector. This vector acts as the input synapse; the input layer is weight adjusted and put into an activation function. Every single neuron has a connection to every single neuron in the next layer. The output is then compared with true values and the error generated is back-propagated, i.e., the weights are re-adjusted and all the processes are repeated. This is done until the error is reduced or we get correct output.The process of adjusting these weights to get correct output is called training.

Why does VGGFACE give the most accurate result among all?

VGGFace model is used for face verification by calculating the embeddings of the face image and comparing it with the embeddings calculated for another face by calculating the similarity measures such as cosine distance or Euclidean distance and determining the face match by comparing the distance to a predefined threshold. It consists of 11 layers in which eight convolutional layers and three fully connected layers. The model was trained with a vast dataset containing 2.6 M face images of the same number of individuals. The architecture of VGGFace contains 38 layers, and the input image size for this model is 224 × 224. This work aims to evaluate its accuracy for face recognition in challenging scenarios like discriminating between siblings or similar-looking faces.

4. Challenges and Future Directions

Here, the recognition is limited to only one face ,i.e, my face. The algorithm checks if the face before the camera is mine or not. But as we extend to more people and larger datasets , the threshold method of verification becomes tedious and time-consuming. Instead we can use transfer learning which is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

CNN models for image classification focus on two parts:

- **Feature extraction:** to find the features in the image
- **Classification:** to use the various features extracted in the previous part and classify the image to the desired classes.

In transfer learning, we retain the feature extraction part of a trained model and only retrain the classifier part.

One major disadvantage of facial recognition is that the technology can be misused and its inability to differentiate between 2D and 3D images. This can be misused as biometric spoofing. The above technique does not check if the face is real or still so any intruder can use a still photo of the person and try to break in.

Apple has launched a facial-recognition technology on the iPhone X in 2017**. Apple's FaceID** cannot be spoofed by 2D photos.Face ID uses a **"TrueDepth camera system"**, which consists of sensors, cameras, and a dot projector at the top of the iPhone display in the notch to create a detailed 3D map of the face.Each time we glance at phone, the TrueDepth camera system will detect face with a flood illuminator, even in the dark.An infrared camera will then take an image, and a dot projector will project out over 30,000 invisible infrared dots. This system uses the infrared image and the infrared dots, and pushes them through neural networks to create a mathematical model of our face.

“TrueDepth camera system” technology if used in smart doors can help in eliminating chances of biometric spoofing and can help this technology head towards more accurate and secure authentication.

5. References

https://sefiks.com/2018/09/03/face-recognition-with-facenet-in-keras/
https://sefiks.com/2018/08/06/deep-face-recognition-with-keras/
https://gist.github.com/daniellerch/f61f1de3b0b26430c9b1bf2e087fa43e(vggface model code)
https://machinelearningmastery.com/how-to-perform-face-recognition-with-vggface2-convolutional-neural-network-in-keras/

Entrepreneurial development in Assistive Technology

INTERNSHIP REPORT
BY
SHASHWAT CHATURVEDI
BTECH/10703/19
DONE AT
ILLINOIS INSTITUTE OF TECHNOLOGY
CHICAGO

What is Assistive Technology?

- It is the technology developed to assist, rehabilitate and help people with disability to adapt with the surroundings.
- An assistive product is any external product (including devices, equipment, instruments or software), especially produced or generally available, the primary purpose of which is to maintain or improve an individual’s functioning and independence, and thereby promote their well-being.
- For a fact we know that every human in the world is likely to use assistive technology in their lives, as they get older or people with disabilities, they need help and we can help them live with independence with the help of assistive technology.

Accessibility

- Are services, facilities and information accessible?
- Is access equitable, regardless of factors such as gender, age, reason for need, socioeconomic groups, location?

Affordability

- Are products and services affordable for all users and their families?
- Are travel costs considered?
- Have financial barriers been identified and addressed for vulnerable groups?

Adaptability

- Are products and services adapted to the needs of individuals?
- Are services responsive to people's changing needs and goals over time?

Availability

- Are products and services available in sufficient quantities to serve the number of people in need?
- Is there an adequate range of products available?
- Are products and services provided close to where people live?

Acceptability

- Are efficiency, reliability, simplicity, safety and aesthetics considered when designing and providing products and related services?
- Are users able to exercise choice and control over decisions regarding their products and services?
- Are products and services appropriate, considering factors such as age, gender and culture?

Quality

- Do products meet standards including strength, durability, performance, safety and comfort?
- Do services meet guidelines, including staff training requirements?
- Are users involved in assessing quality of products and services?

Source: Joint position paper on the provision of mobility devices in less-resourced settings: a step towards implementation of the Convention on the Rights of Persons with Disabilities related to personal mobility. Geneva: World Health Organization; 2011 *(2)*.

Why do we care?

- Mobility issues are the most common issue around the
- world where the people might need a mobility device to do daily chores.
- This might be needed by people with chronic diseases, people getting old who are losing their motor skills.
- But even with the mobility devices, people with disabilities face difficulties as there is unavailability of ramps.
- Especially in places like cars special modifications are done but they are very expensive.


90-41
TO SEAT
90-81

What is our goal?

Here it is evident that affordability and availability are the issues when it comes to the ramps and the modifications done in a car to attach a ramp.

So the goal here was to make a ramp that is affordable and is easy to carry and adjustable to any height.

Why this?

- From our personal observations and citations from UNICEF report on assistive technology, it is observed that still at many places people with mobility issues cannot move independently, for example;
- In India, trains are the main mean of inter city transportation and it is not possible to board a train for a person on wheelchair without help.
- Every person in this world has a right to live independently hence we are designing a solution to end this dependence.

What do we propose?

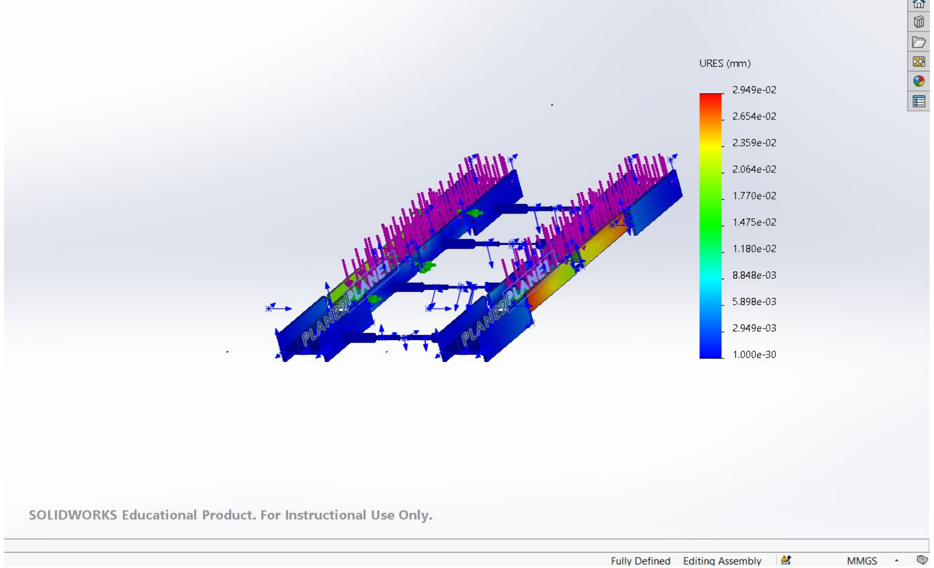
- ❖ A ramp shaped like a ladder which folds up like a briefcase.
- ❖ This reduces the weight of the ramp and decreases the material usage hence can be made at cheaper rate than a regular ramp- **AFFORDABILITY**
- ❖ The ramp can fold into a briefcase hence can be accessible everywhere as it is easy to carry- **ACCESSIBILITY AND AVAILABILITY**
- ❖ They are extendable to different length and width to suit different places like a simple curb to a steep train door to a minivan door.- **ADAPTABILITY**

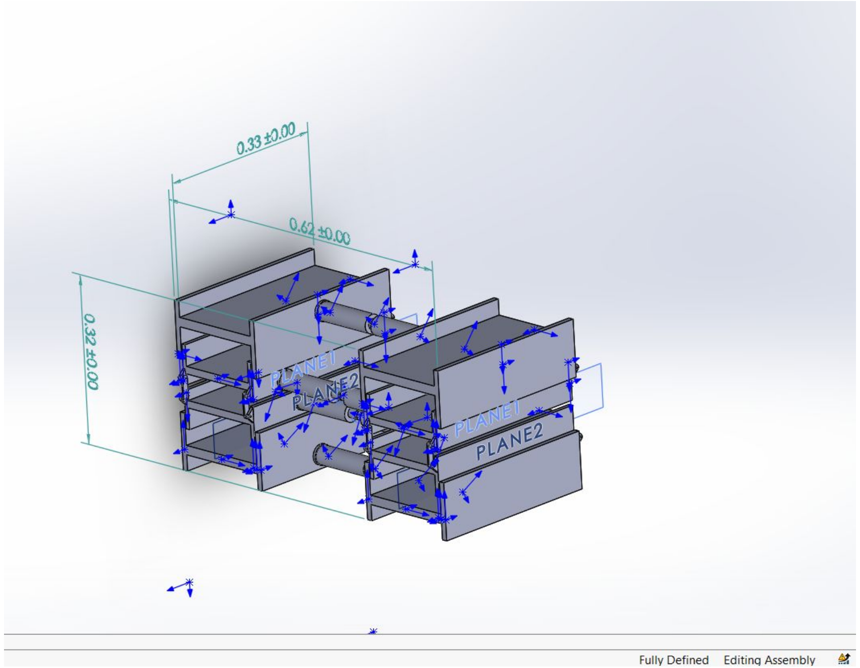
Parameters set for the design:

- The height for the ramp was chosen to be the Chrysler Town and Country minivan.

- While the motorized wheelchair was chosen for design of the ramp, dimensions decided after comparing various wheelchairs.

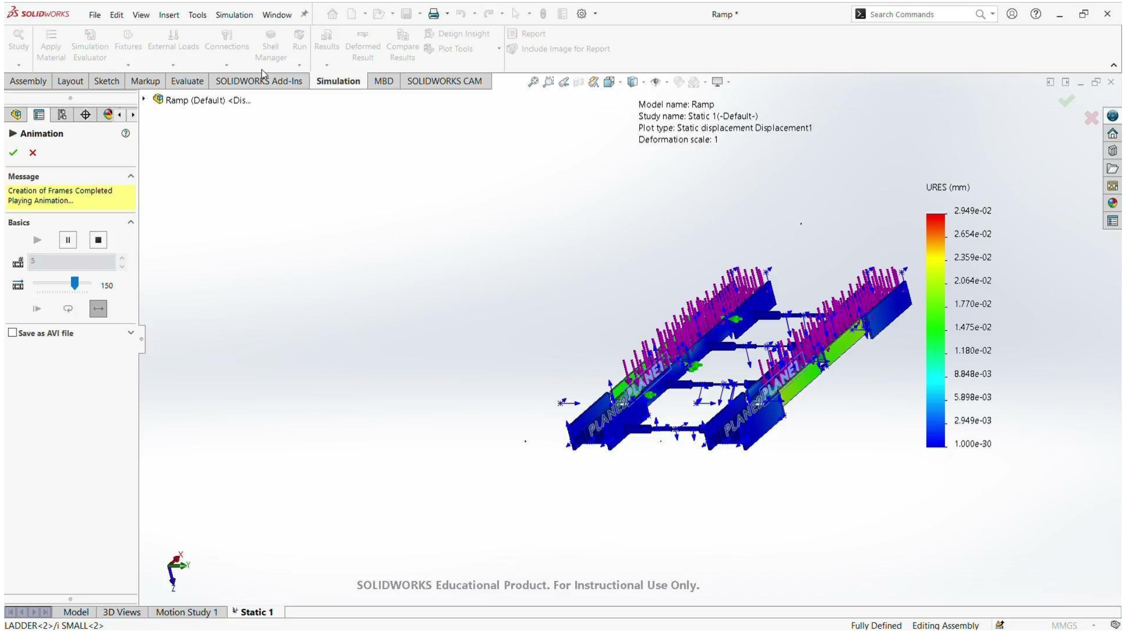
The design


URES (mm)
2.949e-02
2.654e-02
2.359e-02
2.064e-02
1.770e-02
1.475e-02
1.180e-02
8.848e-03
5.898e-03
2.949e-03
1.000e-30
PLANE1
PLANE2
SOLIDWORKS Educational Product. For Instructional Use Only.
Fully Defined
Editing Assembly
MMGS


0.33 ±0.00
0.62 ±0.00
0.32 ±0.00
PLANE1
PLANE2
Fully Defined
Editing Assembly

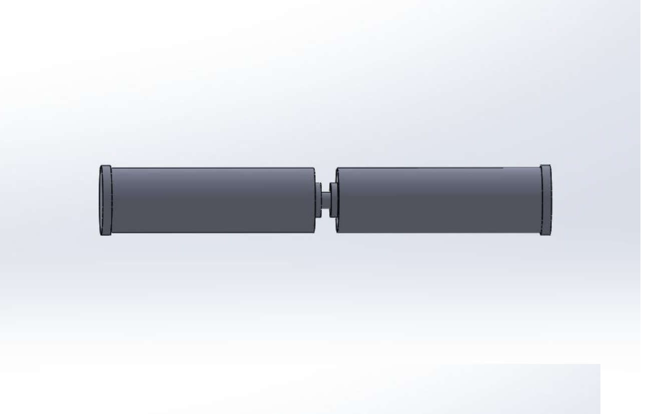
Analysis

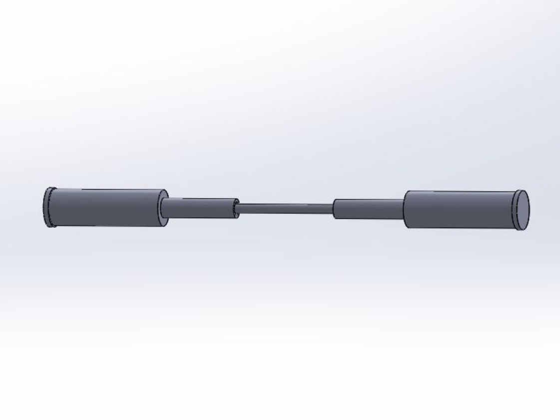
Design was analysed in Solidworks by applying a force of 150 kg as the wheelchair weighs 30kg and taking the weight of a person to be 120kg

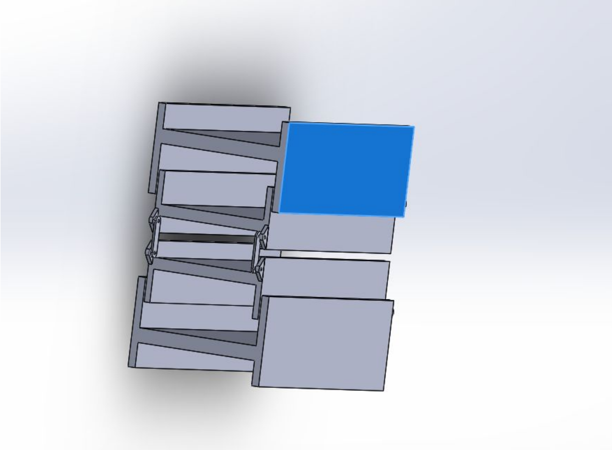


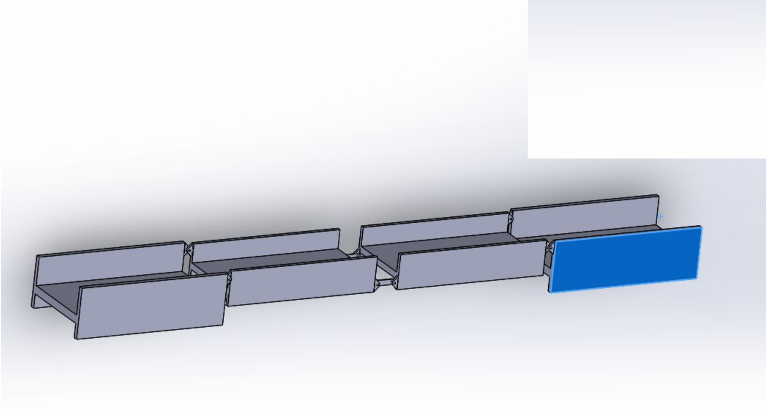
Design Specification

- Weight: 10kg
- Folded Dimensions: 0.33m*0.62m*0.32m
- Material Used: Al 6016 T6
- Displacement on applied force: 0.02mm
- Factor of Safety: 48









What ahead?

- The first design and prototype can be tested and the be further iterated seeing the software results and manual testing.
- Right now the displacement is very little while factor of safety is very high so there is huge scope for weight reduction in further designs.
- The whole process of unfolding-folding of ramp can be made automated after testing the ramp mechanically amongst a test group.

Thank You

HIGH RESOLUTION TEMPERATURE SENSOR DESIGN FOR DATA ACQUISITION IN INVESTIGATION OF CORRELATION BETWEEN NEUROLOGICAL ACTIVITY AND MINUTE BODY SURFACE TEMPERATURE CHANGE
ECLTSB.1

A Report on Summer Research Internship submitted in partial fulfilment of the Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL & ELECTRONICS ENGINEERING

BY

SHASHWAT JHA (BTECH/10005/19)

Under the guidance of
Dr. Aatmesh Shrivastava

Energy Efficient Circuits and Systems Lab College of Engineering Northeastern University Boston MA. US

And the supervision of

Dr. S. SHIVA KUMAR (Assistant professor)

EEE Department BIT MESRA

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
BIRLA INSTITUTE OF TECHNOLOGY MESRA,
RANCHI - 835215. (INDIA)

STUDENT DECLARATION

It is hereby declared that the **"Internship report"** submitted by **SHASHWAT JHA (BTECH/10005/19)** is work done by him and submitted during 2019 – 2023 academic year, in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY in ELECTRICAL AND ELECTRONICS ENGINEERING,

at Energy Efficient Circuits and Systems Lab College of Engineering, Northeastern University

Northeastern University Interdisciplinary Science & Engineering Complex
805 Columbus Ave, Boston, MA 02120, United States

College Internship Coordinator - Dr. S. SHIVA KUMAR (Assistant professor, EEE)
Head of the Department Department of EEE -Professor T. Ghose

ACKNOWLEDGEMENT

I would like to express my deep gratitude to our project guide and supervisor, **Dr. Aatmesh Shrivastava and Dr. S.Shiva Kumar** under whose guidance I was able to pursue this endeavour .

I am very grateful to **Dr. Pankaj Mishra**, Birla Institue of Technology for extending all faculties and giving valuable suggestions at all times for pursuing this project.

I would also like to thank **Prof. T Ghose, HOD Electrical and Electronics, Birla Institute of Technology Mesra.**

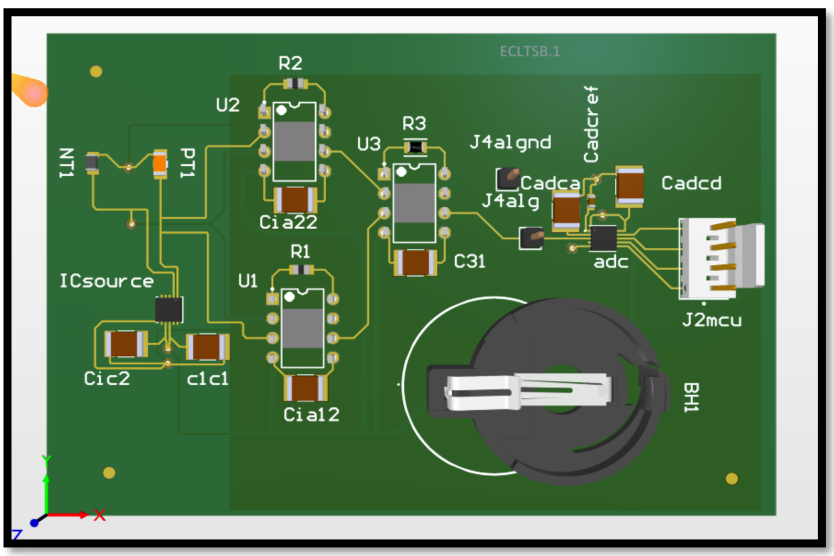
It feels impossible to express how grateful I am to the extent that it overwhelms me to realise the sheer support and positivity which was bestowed leading to the environment conducive for the success of this project

TABLE OF CONTENTS

INTRODUCTION

This report is drafted with the intention to outline my contributions in the development and design of *high resolution temperature sensor design for data acquisition in investigation of correlation between neurological activity and minute body surface temperature change*. During my internship at **Energy Efficient Circuits and Systems Lab College of Engineering, Northeastern University** Interdisciplinary Science & Engineering Complex 805 Columbus Ave, Boston, MA 02120, United States.

The proposed design achieved a staggering resolution of 0.0005 degree of temperature change while only utilising a 17-bit analog to digital conversion. The compact design of the sensor board makes it ideal for using it in biomedical applications.



ECLTSB.1

ABOUT THE LABORATORY

The EECS LAB at Northeastern University, College of Engineering is led by Prof. Aatmesh Shrivastava, performing research focussed on self-powered and ultra-low power circuits and system for Internet-of-things (IoTs). We are currently working to solve the power issue, in order to make IoT devices easily deployable in our environment. We are also focussing on analog computing hardware for AI applications, ultra-low power bio-medical and neural circuits, system modelling, and high-reliability circuits and system design. With 20 U.S granted patents and and multiple awards as well as grants from prestigious organisations like NSF, This lab is a powerhouse of innovation in analog circuits

ABOUT THE PROJECT

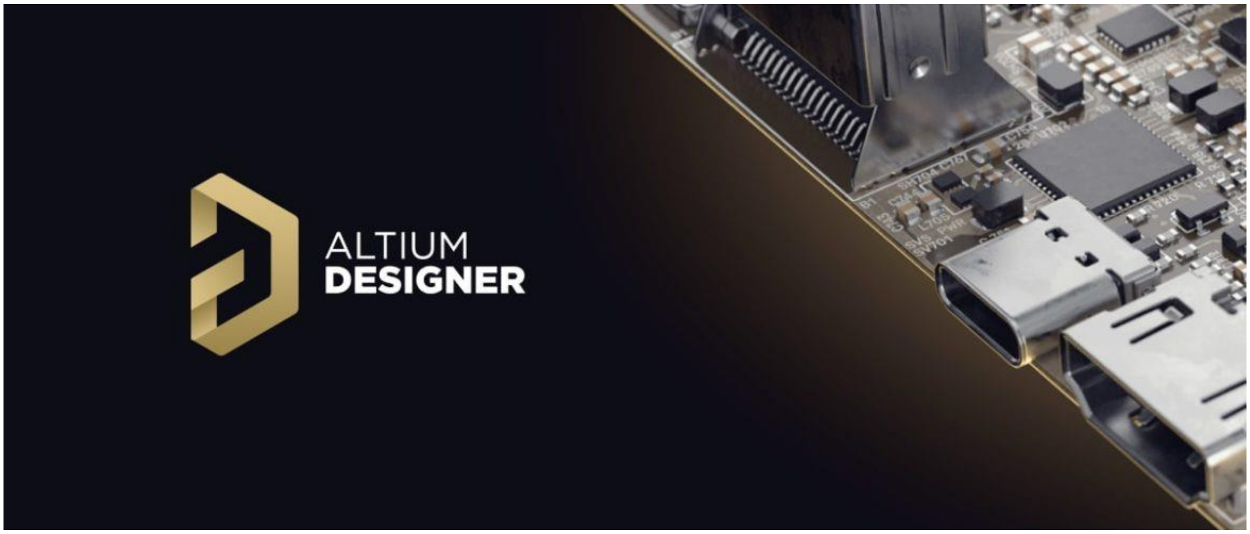
The temperature sensor design was based on the properties of thermistors. In which we utilised the caliber of both positive and negative temperature coefficient thermistors to create a difference in voltage whenever the temperature changed. Then this voltage difference was amplified to a level where the 17^{th} bit of 18-bit ADC with 1LSB error range was enough to assign a value till the 0.0005 degree change in temperature. the ability of the board design is of high utility since the it works proficiently even with a small 3V coin cell and approximately 100 uA of current.

SOFTWARE TOOLS UTILISED

1. LT SPICE

LTspice is a SPICE-based analog electronic circuit simulator computer software, produced by semiconductor manufacturer Analog Devices (originally by Linear Technology). It is the most widely distributed and used SPICE software in the industry. Though it is freeware, LTspice is not artificially restricted to limit its capabilities (no feature limits, no node limits, no component limits, no subcircuit limits). It ships with a library of SPICE models from Analog Devices, Linear Technology, Maxim Integrated, and 3rd party sources too. In our case it was utilised for simulating the thermistor-based circuit including the amplification in a transient manner with step parametric temperature change.

2. ALTIUM DESIGNER



Altium Designer is a PCB and electronic design automation software package for printed circuit boards. It is developed by Australian software company Altium Limited.was originally launched in 2005 by Altium, known at the time as Protel Systems Pty Ltd. It has roots back to 1985 when the company launched the DOS-based PCB design tool known as Protel PCB (which later emerged into Autotrax and Easytrax). Originally it was sold only in Australia. Protel PCB was marketed internationally by HST Technology since 1986.The product became available in the United States, Canada, and Mexico beginning in 1986, marketed by San Diego-based ACCEL Technologies, Inc. under the name Tango PCB.In 1987, Protel launched the circuit diagram editor Protel Schematic for DOS.

In 1991, Protel released Advanced Schematic and Advanced PCB 1.0 for Windows (1991–1993), followed by Advanced Schematic/PCB 2.x (1993–1995) and 3.x (1995–1998). In 1998, Protel 98 consolidated all components, including Advanced Schematic and Advanced PCB, into a single environment. Protel 99 in 1999 introduced the first integrated 3D visualization of the PCB assembly. It was followed by Protel 99 SE in 2000. Protel DXP was issued in 2003, Protel 2004 in 2004, Altium Designer 6.0 in 2005. Altium Designer version 6.8 from 2007 was the first to offer 3D visualization and clearance checking of PCBs directly within the PCB editor.

HARDWARE COMPONENTS UTILISED FOR THE DESIGN

1. **NTC Thermistor**
2. **TFPT Thermistor**
3. **INA128 Instrumantation amplifier**
4. **Decoupling capacitors**
5. **Gain resistors**
6. **Connecting head**
7. **Current source IC**
8. **Battery holder**

**Please note that the datasheets for all these components are mentioned in the BIBLIOGRAPHY containing all the requisite and relevant information

BILL OF MATERIALS – This sheet comprises of all the requisite information about the hardware components required for assembly and fabrication including supplier as well as manufacturing details.

BOM:

Line #	Name	Description	Designator	Quantity	Manufacturer 1	Manufacturer Part Number 1	Supplier 1	Supplier Part Number 1
	ADS8881IDRCR	anlg to digi con	adc	1	TEXAS INSTRUMEN	ADS8887IDRCT	digikey	296-37265-2-ND
	1066	cell holder	BH1	1	keystone	CH25-2032LF	digi-key	36-1066-ND
	18125C105JAT2A	Ceramic Capac	c1c1, C31, Cadca	7	KYOCERA AVX	18125C105JAT2A	digikey	478-7984-2-ND
	LMK107BJ106MALTD	Capacitor	Cadcref	1	taiyo yuden	LMK107BJ106MALTD	digikey	587-2562-2-ND
	REF200AU_2K5	Integrated Circ	ICsource	1	TEXAS INSTRUMEN	REF200AU/2K5	digikey	296-27726-2-ND
	22-29-2041	CONN HEADER	J2mcu	1	Molex	22-29-2041	Arrow Electro	22-29-2041
	61300111121	THT Vertical Pi	J4alg, J4algnd	2	Wurth Electronics	61300111121	Mouser	710-61300111121
	NTCS0805E3222FMT	Thermistor	NT1	1	VISHAY	NTCS0805E3222FMT	digikey	BC2569TR-ND
	TFPT1206L5601FV	TFPT	PT1	1	VISHAY	TFPT1206L5601FV	MOUSER	71-TFPT1206L5601FV
	RT0805BRD0756KL	RESISTOR	R1, R2	2	YAGEO	RT0805BRD0756KL	digikey	YAG1933TR-ND
	RP73PF2A12K4BTDF	Resistor	R3	1	te connectivity	1-2176093-3	digikey	A110648TR-ND
	TI-INA1XX-8	Instrumentatio	U1, U2, U3	3	TEXAS INSTRUMEN	INA128P	digikey	INA128P-ND

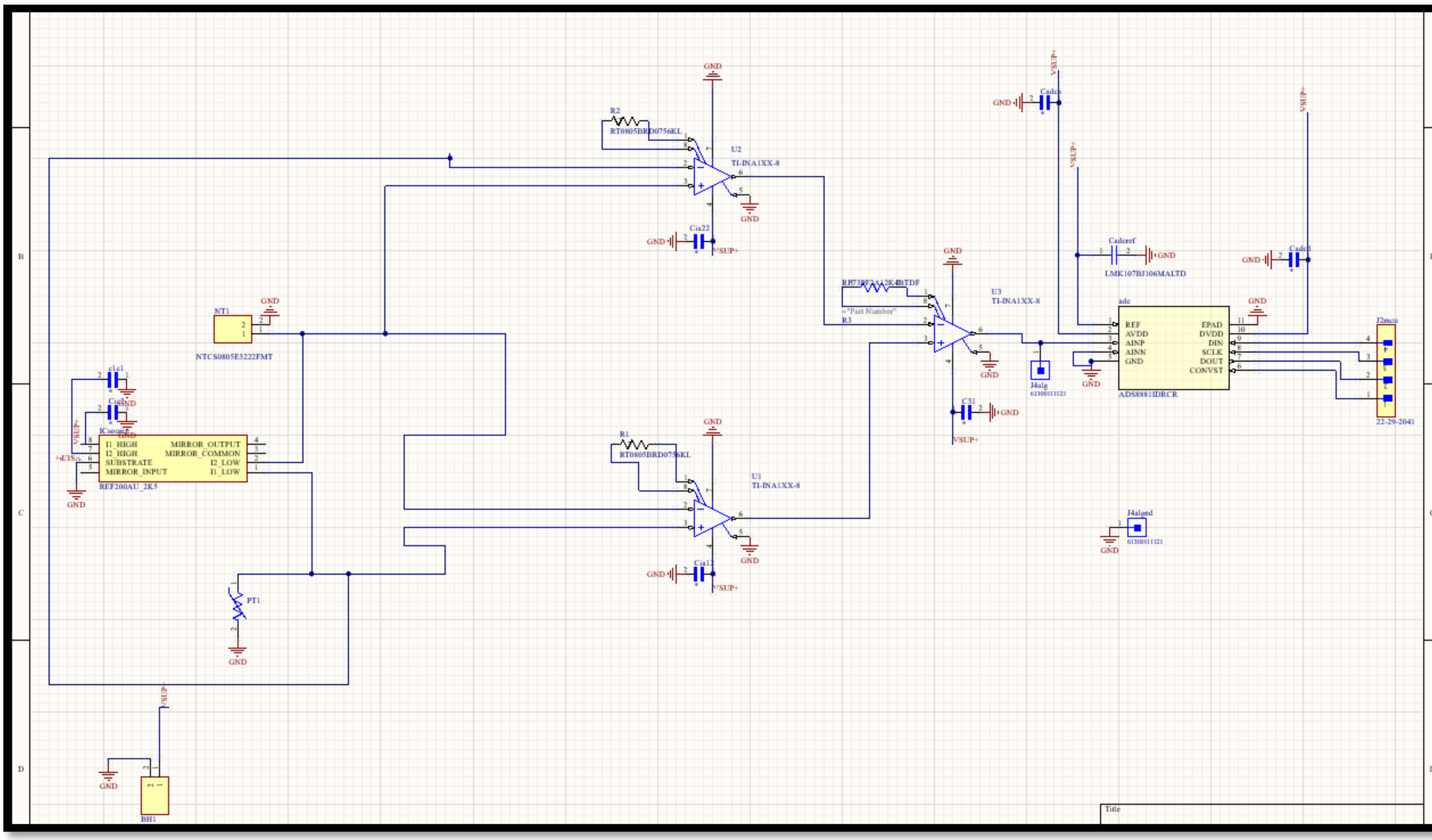
THE CIRCUIT

The circuit intends to make use of both increasing as well as decreasing temperature change and hence we utilise both a NTC as well as PTC thermistor to maximise the voltage difference between the points where we tap in the voltage from. Moreover, our design consists of 3 instrumentation amplifiers connected in opposing polarities to the thermistors with respect to each other, which in turn helps our temperature sensor to have a larger temperature sensing range even at extremely low temperatures without compromising the integrity of the signal/ differential voltage.

This analog acquisition unit is then driven into an 18bit – ADC with given error range of 1LSB making it effectively 17-bit. Considering all the factors, all the components in the circuit are decoupled using capacitors mentioned in their advised applications.

The current is drawn from a current source IC which operates on the same supply as rest of the components on the board i.e 3Volts which is supplied by a coin cell .

THE CIRCUIT SCHEMATIC



THE PCB DESIGN

The PCB was designed using Altium designer wherein the number of layers was restricted to 2. This itself contributed immensely in making the board ergonomic.

The components were placed ton the board such that the thermistors are a little away from the other components since otherwise this would've caused unnecessary deviation in the temperature measurements.

Specific design rules were followed to reduced the cost of fabrication for the board and maintaining the integrity while minimizing heat dissipation as well as current leakage.

In the the adc is connected to a header with the requisite output as well as SPI pins.

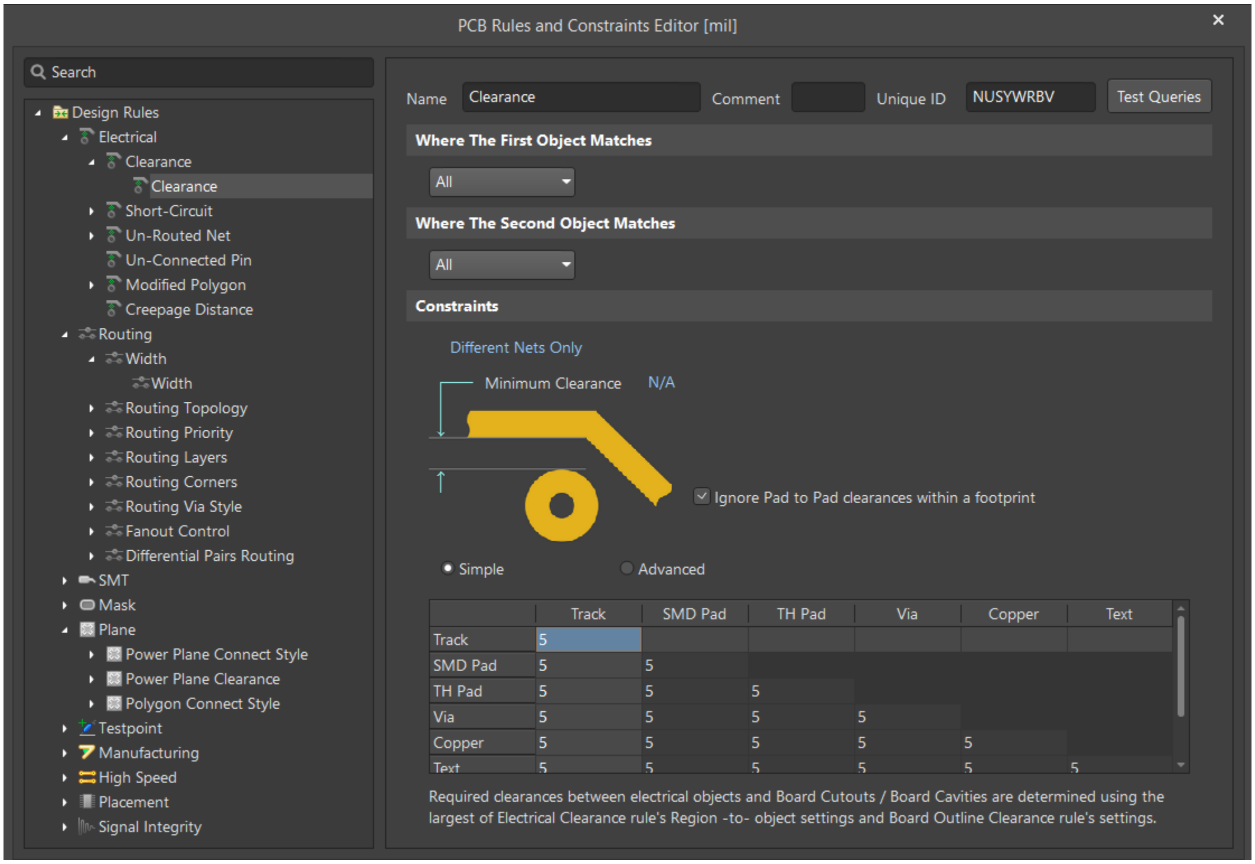
INDEX OF DESIGNATORS FOR CIRCUIT COMPONENTS

Description	Designator
analog to digital converter	adc
cell holder	BH1
Ceramic Capacitor X7R 1812 (4532 Metric)1uF	c1c1, C31, Cadca, Cadcd, Cia12, Cia22, Cic2
Capacitor 10uF	Cadcref
Integrated Circuit	ICsource
CONN HEADER VERT 4POS 2.54MM	J2mcu
THT Vertical Pin Header WR-PHD, Pitch 2.54 mm, Single Row, 1 pins	J4alg, J4algnd
Thermistor	NT1
TFPT	PT1
RESISTOR for IA gain 1st stage	R1, R2
Resistor for IA gain 2nd stage	R3
Instrumentation amplifiers	U1, U2, U3

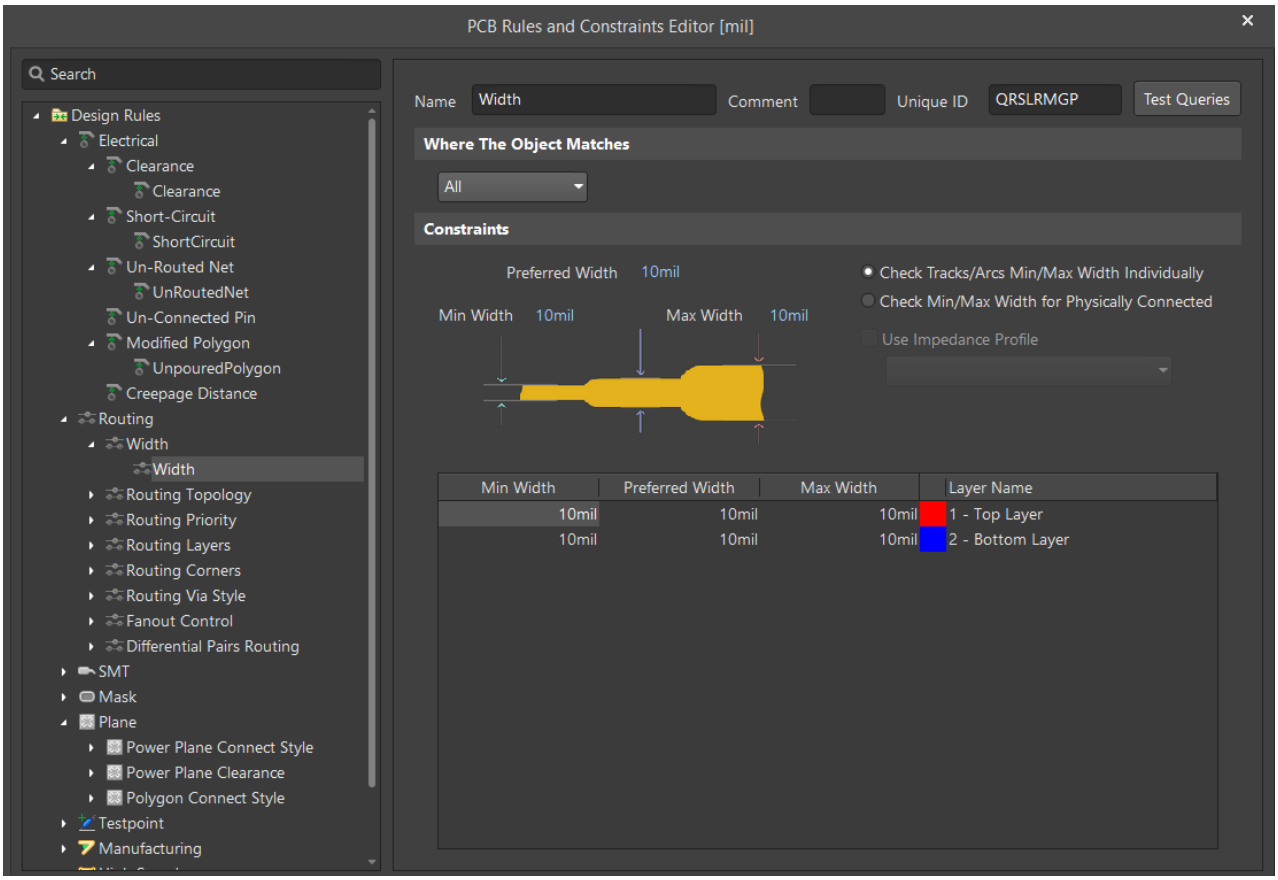
DESIGN RULES

***ALL BOARD MEASUREMENTS OF LENGTH ARE IN MILS**

1. CLEARANCE



2. ROUTING



***** the rules as evident in the images were set as following respectively:**

Clearance- 5 mils

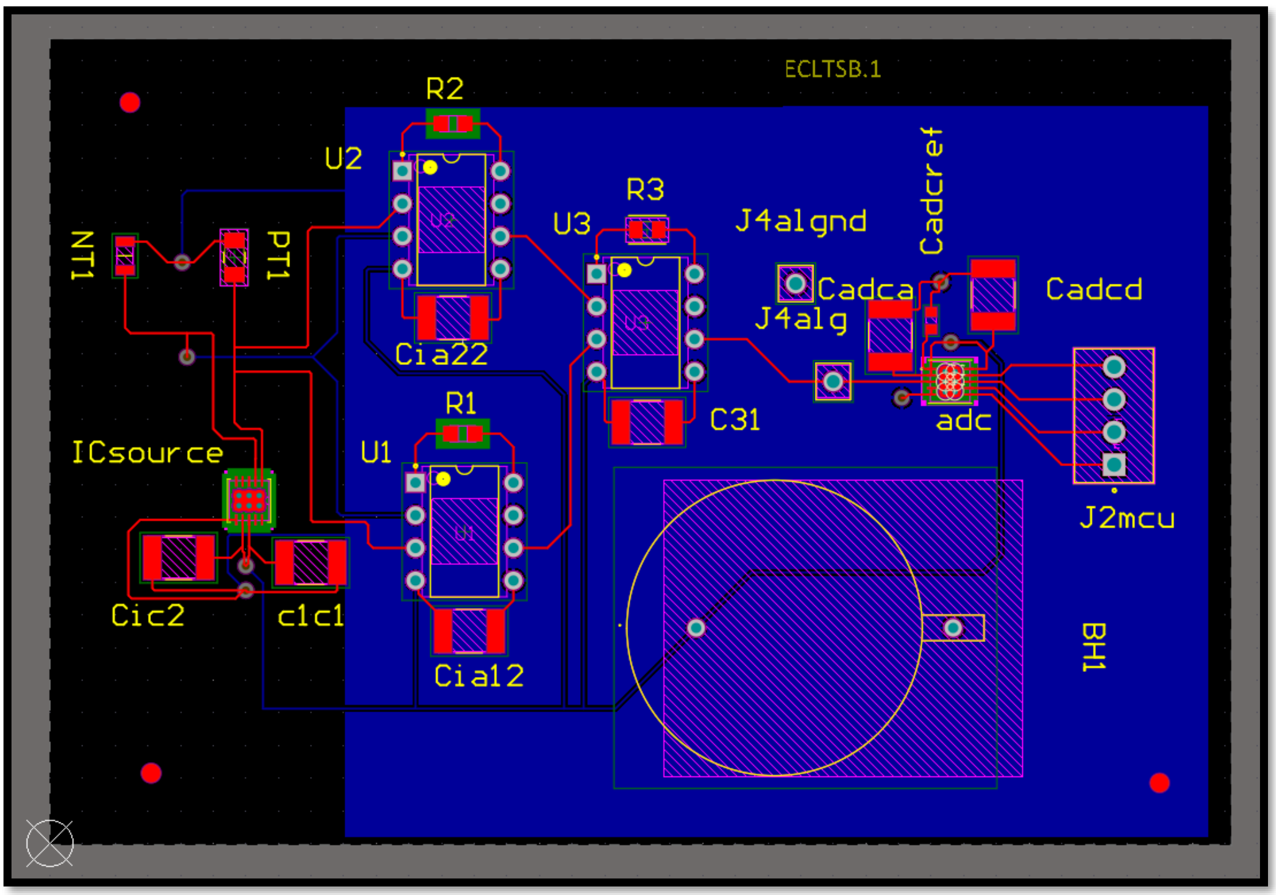
Routing – 10 mils

These were the values indicated by the fabricating facility and therefore we proceeded with these.

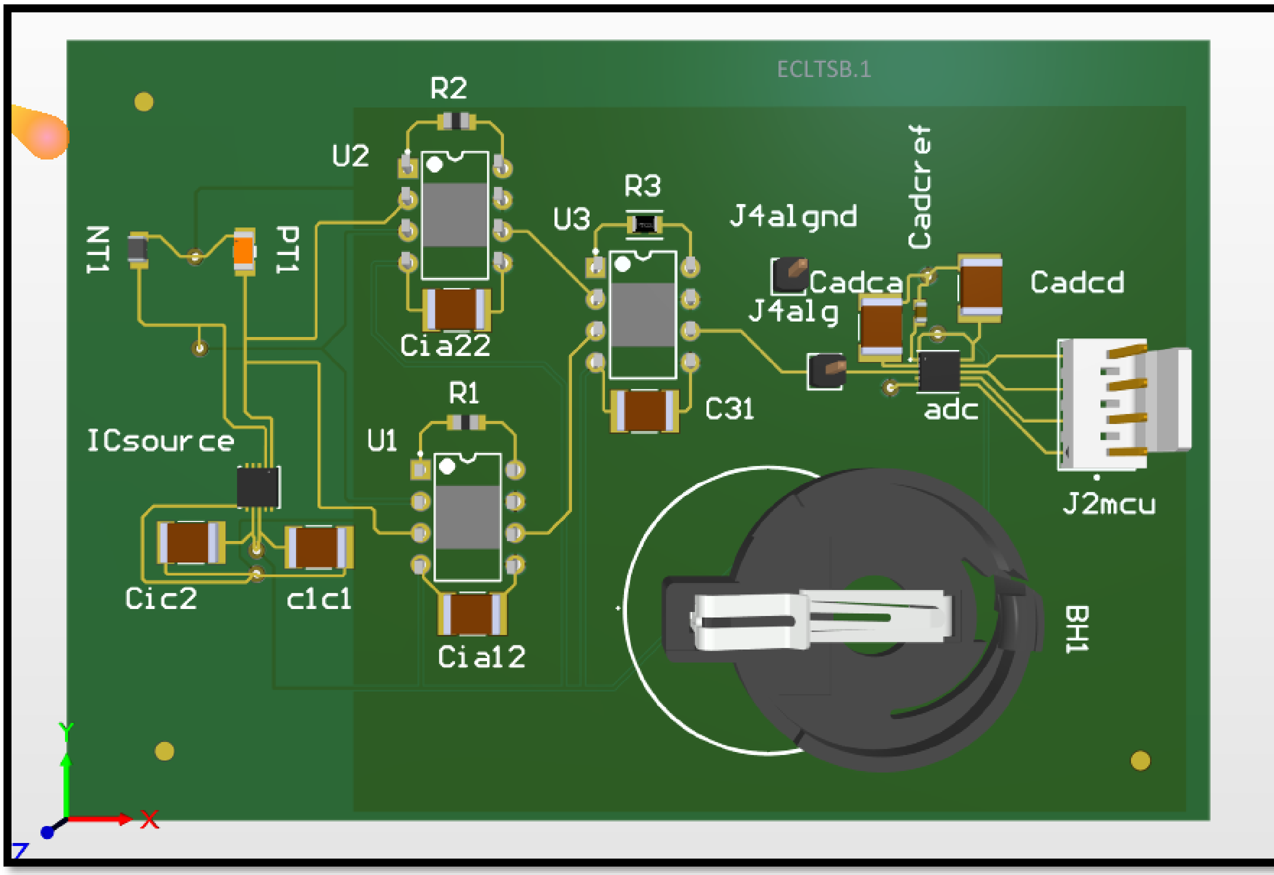
LAYERS

LAYER.1

The first layer or the top most layer is the layer where most of the component assembly is done. All components are mounted on this layer. In design window the connection / routing done on this layer is depicted by red colour traces and as yellow traces in 3D view as shown in the figures below :



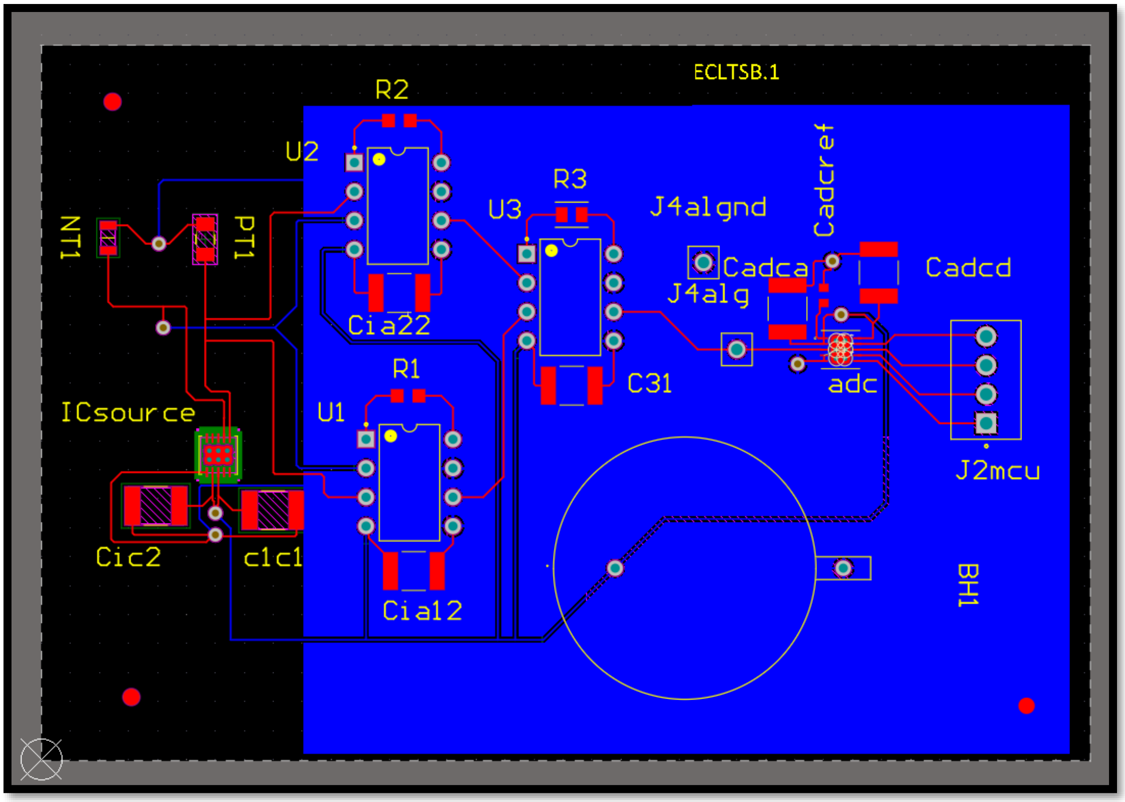
A- DESIGN VIEW OF PCB WITH TOP LAYER/ 1ST LAYER HIGHLIGHTED



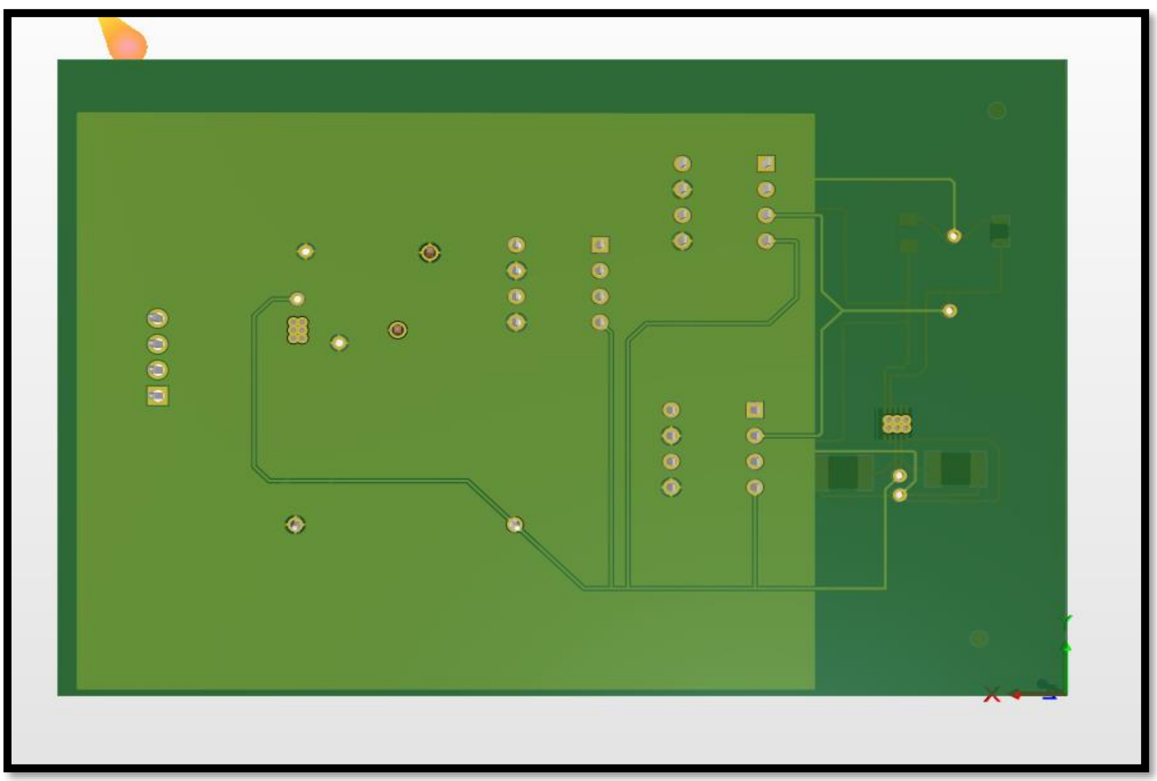
B- 3-D VIEW OF THE BOARD TOP LAYER

LAYER.2

This layer mainly constituted of the ground polygon pour and connections to the battery holder pins which would act as the supply, we have utilised various pads and vias to route connections between layers therefore reducing overcrowding in between the connections. The connections can be seen as traces in the colour blue along with the pour in the design mode whereas light green in 3D mode as shown in the figures below :

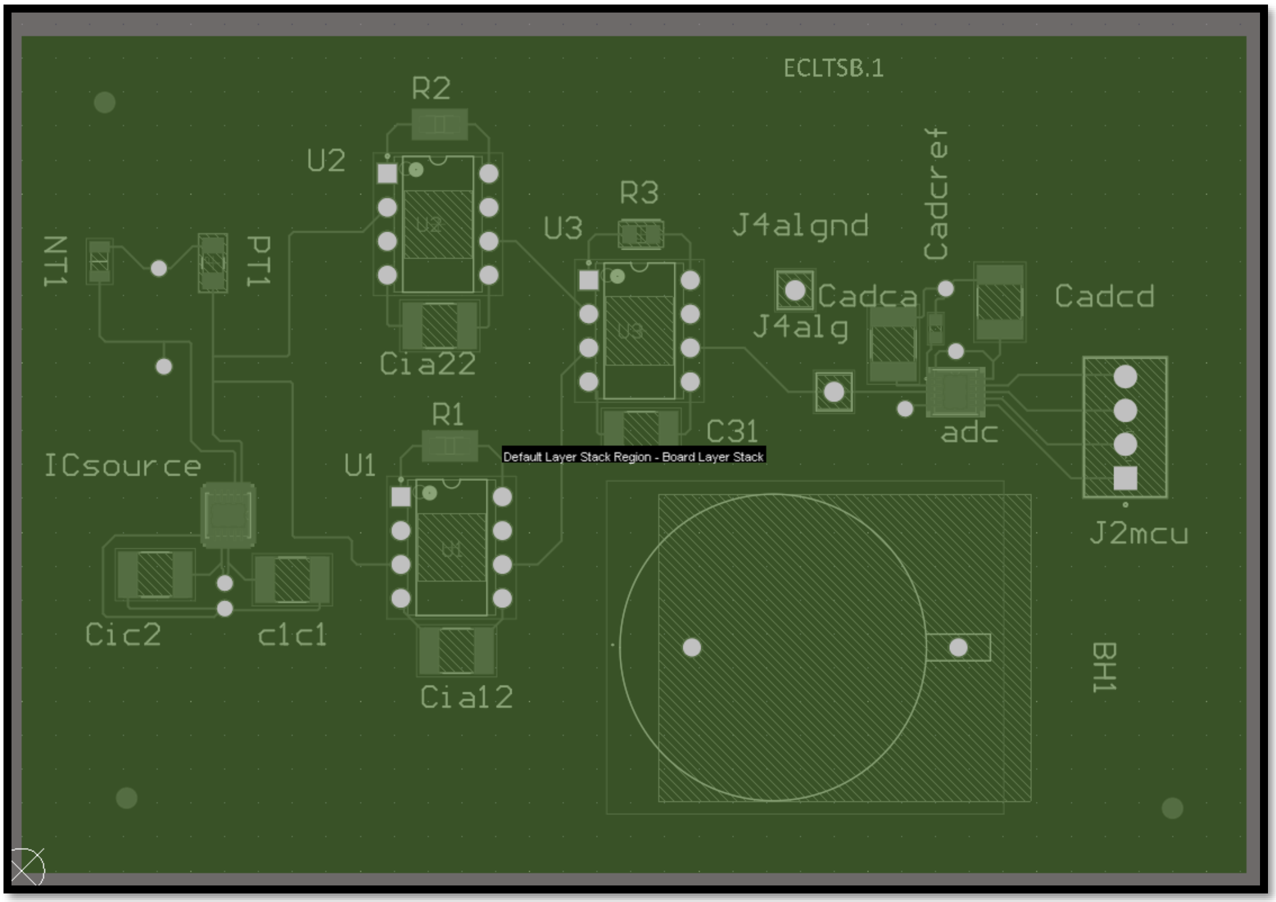


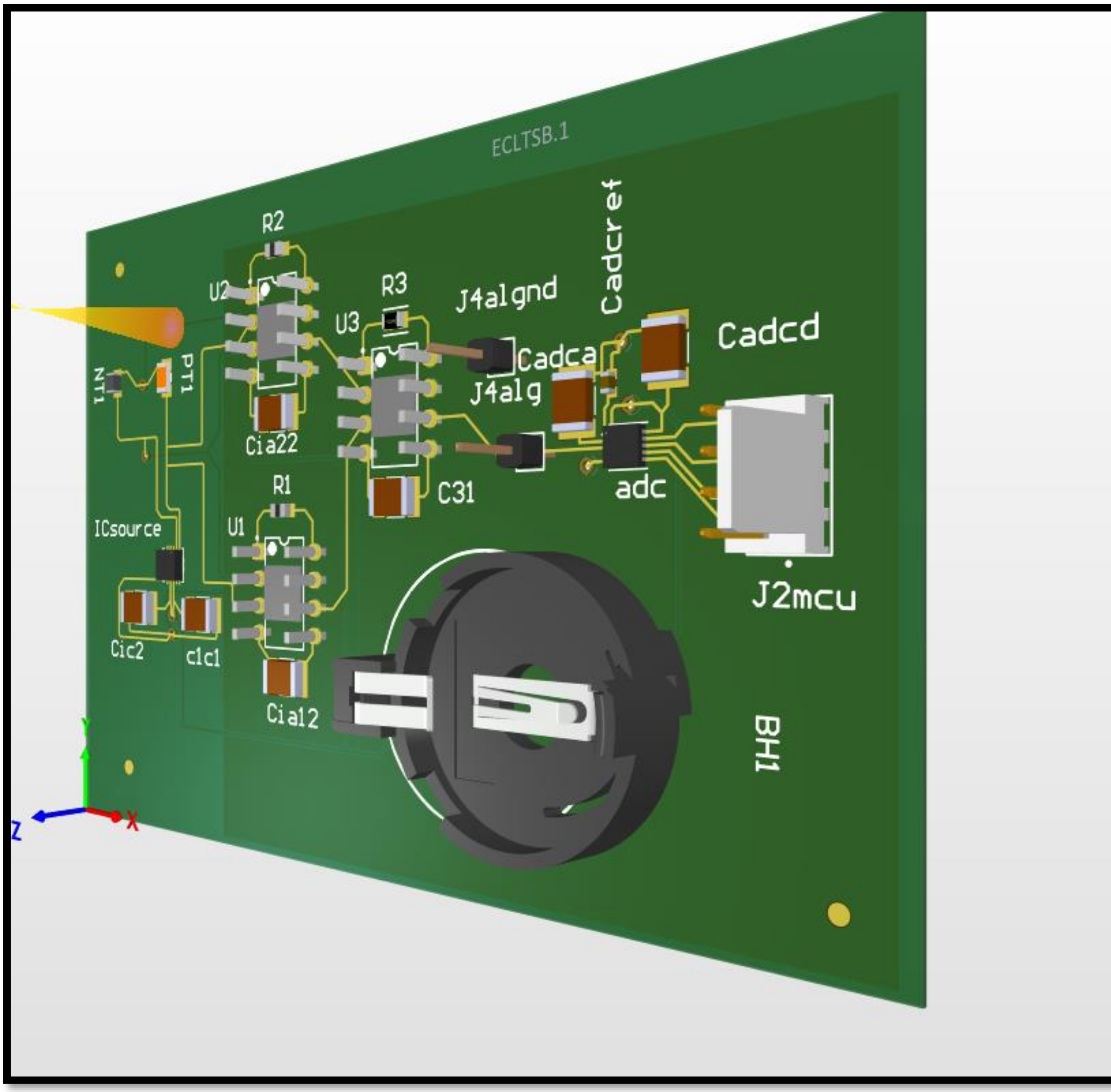
A. DESIGN VIEW OF THE BOARD WITH LAYER 2 HIGHLIGHTED



B. 3D VIEW OF THE BACKSIDE/ 2ND LAYER OF THE PCB

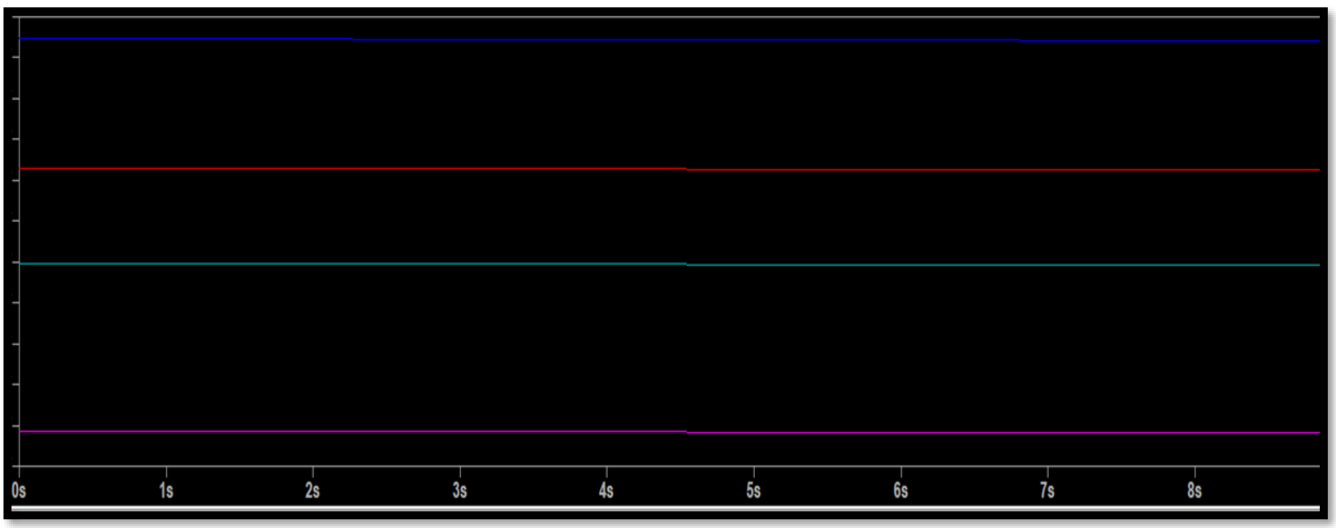
FURTHER IMAGES OF THE BOARD DESIGN




ECLTSB.1
R2
U2
R3
U3
J4algnd
Cadcref
Cadcd
Cadca
J4alg
NT1
PT1
Cia22
C31
adc
R1
ICsource
U1
J2mcu
Cic2
c1c1
Cia12
BH1
Y
Z
X

SIMULATION SNAP

This snapshot depicts the various voltages at various temperatures set at 0.0005 degrees difference while running a transient simulation of current at 100uA



CONCLUSION

In conclusion one could argue that why should we develop a board based temperature sensor design when IC designs are highly accurate and much better suited , Our design tends to compete with that claim while adding value due to its modular nature and low cost of operation, assembly and fabrication without any compromise on immaculate performance and exceptional resolution . Therefore it is appropriate to state that this board with great connectivity and such high resolution can be suitably used for the mentioned hypothesis and investigate whether it’s the truth or a fallacy .

BIBLIOGRAPHY

References

[1] K.S. Szajda, C.G. Sodini, H.F. Bowman, IEEE J. Sol State Circuits vol. 31 p. 1308-1313

[2] G. Kell, F. Winkler, M. Gunther, T. Morgenstern, K.

Wittmann, G. Otto, G. Gorler, Acta Astronautica 43

vol. 7-8, p. 385-395, 1998

[3] http://www.aslltd.co.uk/f250.htm

[4] D. Hayoz, Y. Tardy, B. Rutschmann, J.P. Mignot, H.

Achakri,F Feihl, J. J. Meister, B. Waeber, H.R. Brunner

Am. J.Physio. Heart Circulat. Physio. 264, 2080 (1993),

[5] G.L. Brengelmann “Body Surface Temperature: Mani festation of Complex Anatomy and Physiology of Cu taneous Vasculature” Proceedings of the 22th Annual

International Conference, p. 1927

[6] N. S. Latman, P. Hans, L. N. Nicholson, S. D. Zint, K.

Lewis, A. Shirey Biomed. Instrum. Technol. 35, 259

(2001)

[7] P. Dupuis, C. Eugène “A high Resolution Micropower

Acquisition System for Passive Sensors” IMEKO 1993

p.309

[8] Dupuis, P.; Eugene, Ch.; David, B. “Thermal charac terisation of human skin using differential thermome try” Instrumentation and Measurement Technology

Conference, 1996. IMTC-96. Proceedings. 'Quality

Measurements: The Indispensable Bridge between The ory and Reality'., IEEE , Volume: 2 , 1996 p. 1458

[9] K. Mabuchi, T. Chinzei, Y. Nasu, H. Genno, I. Fuji masa « Use of Thermal Rhythmography for the Evalua tion of Autonomic Nervous Function » Engineering in

Medicine and Biology Society, 1995. IEEE 17th An nual Conference , Vol. 2 , 20-23 Sept. 1995 pp: 1703 -

1704

[10] K. Kondo, N. Kakuta, T. Chinzei, Y. Nasu, T. Suzuki

“Termal Rhytmography – Topograms of the Spectral

Análisis of Fluctuations in Skin Temperatura” Engi neering in Medicine and Biology Society, 2001. Pro ceedings of the 23rd Annual International Conference

of the IEEE , Vol. 3 , 25-28 Oct. 2001, p: 2812 -2815

[11] H. Kataoka, H. Kano, H. Yoshida, A. Saijo, M. Ya suda, M. Osumi “Development of a Skin Temperature

Measuring System for Non-Contact Stress Evaluation”

Proceedings of the 20th Annual International Confer ence of the IEEE , Volume: 2 , 29 Oct.-1 Nov.1998

Page(s): 940 -943 vol.2

[12] O. Casas, J .Martínez, R. Pallàs-Areny “Monitoriza ción Mecánica de la Frecuencia Cardiaca. Resultados

Preliminares” Proc. XX Congreso Anual Sociedad Es pañola de Ingeniería Biomédica” Zaragoza 2002 p. 357

[13] http://www.oregongadgets.com

[14] http://www.betatherm.com

[15] H. Kataoka, H. Kano, H. Yoshida, A. Saijo, M. Ya suda, M. Osumi “Development of a Skin Temperature

Measuring System for Non-Contact Stress Evaluation”

Proc. 20th Annual International IEEE Conference on

Engineering in Medicine and Biology Society, 1998.

Vol. 2 , 29 Oct.1998 Page(s): 940 -943

[16] K. A. A. Makinwa. Smart Temperature Sensor Survey. [Online]. Avail able: http://ei.ewi.tudelft.nl/docs/TSensor_survey.xls

[17] H. C. de Graaff and M. T. M. Huybers, “1/f noise in polycrystalline

silicon resistors,” J. Appl. Phys., vol. 54, no. 5, pp. 2504–2507, 1983.

[18] E. Vereshchagina, R. A. M. Wolters, and J. G. E. Gardeniers, “The devel opment of titanium silicide–boron-doped polysilicon resistive tempera ture sensors,” J. Micromech. Microeng., vol. 21, no. 10, p. 105022, 2011.

[19] P. Li, J. Peng, and D. W. Zhang, “The study on metal-insulator metal capacitor performance improvement,” in Proc. ICSICT, Oct. 2014,

pp. 1–3.

[20] K. Nam, S.-M. Lee, D. K. Su, and B. A. Wooley, “A low-voltage

low-power sigma-delta modulator for broadband analog-to-digital con version,” IEEE J. Solid-State Circuits, vol. 40, no. 9, pp. 1855–1864, Sep. 2005.

[21] R. K. Waits, “Silicide resistors for integrated circuits,” Proc. IEEE,

vol. 59, no. 10, pp. 1425–1429, Oct. 1971.

[22] S. M. Kashmiri, S. Xia, and K. A. A. Makinwa, “A temperature-to digital converter based on an optimized electrothermal filter,” IEEE J.

Solid-State Circuits, vol. 44, no. 7, pp. 2026–2035, Jul. 2009.

[23] S. Pan, Y. Luo, S. H. Shalmany, and K. A. A. Makinwa, “A resistor based temperature sensor with a 0.13 pJ·K2 resolution FOM,” in IEEE

Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers, Feb. 2017,

pp. 158–159.

[24] A. Hastings, The Art of Analog Layout. Englewood Cliffs, NJ, USA:

Prentice-Hall, 2001

Links to all datasheets are as follows :

1.https://www.ti.com/lit/ds/symlink/ina128.pdf?ts=1660668810701&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FINA128

2. https://www.mouser.com/datasheet/2/418/5/ENG_DS_1773272_M-1588495.pdf

3. https://www.yageo.com/upload/media/product/productsearch/datasheet/rchip/PYu-RT_1-to-0.01_RoHS_L_12.pdf

4. https://www.mouser.in/datasheet/2/427/tfpt-1762890.pdf

5. https://www.ti.com/lit/ds/symlink/ref200.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1660685366750&ref_url=https%253A%252F%252Fwww.mouser.in%252F

6. https://4donline.ihs.com/images/VipMasterIC/IC/TAIY/TAIY-S-A0002471898/TAIY-S-A0002471898-1.pdf?hkey=6D3A4C79FDBF58556ACFDE234799DDF0

7. https://datasheets.kyocera-avx.com/X7RDielectric.pdf

8. https://www.keyelco.com/userAssets/file/M65p5.pdf

9.https://www.ti.com/lit/ds/symlink/ads8881.pdf?ts=1660685096424&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FADS8881%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Dasc-null-null-GPN_EN-cpc-pf-google-wwe%2526utm_content%253DADS8881%2526ds_k%253DADS8881%2526DCM%253Dyes%2526gclid%253DCj0KCQjwgO2XBhCaARIsANrW2X0pJSGf8Y1kzzVJ2OonhNeiOaBamIDFVw_YlYXC8-5wOqQGsNyGqYkaAibaEALw_wcB%2526gclsrc%253Daw.ds

10. https://www.vishay.com/docs/29044/ntcs0805e3t.pdf

MODELING UNSTEADY AERODYNAMIC SYSTEM

SHUBHANKAR SINHA
ROLL NO. – BTECH/10453/19
MENTOR: SCOTT DAWSON

FINAL GROUP REPORT

Introduction

The main goal of this research work is to minimize the computational complexity and cost that goes into representing various fluid mechanics and aerodynamic systems in a low dimensional form in order to understand and analyse them better.

Methodology

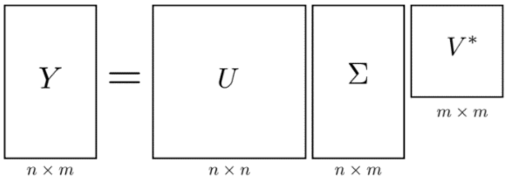
The datasets that we are dealing with are mostly functions of both space and time. In order to understand pertinent features of such datasets as well as to fulfil our goal of building a reduced-complexity model, our first step is to separate the variables using Singular Value Decomposition (SVD) and then using various curve fitting tools in MATLAB and PYTHON we create an analytical function that fits our dataset.

The decomposition methods presented in this report are founded on the singular value decompositions of matrices or operators.

Singular Value Decomposition

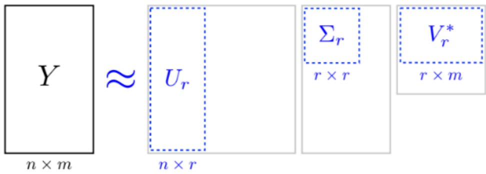
$$y(x,t) = \sum_{j=1}^{m} u_j(x) a_j(t)$$

In order to achieve this decomposition, we use SVD. The main idea of this is to break the matrix containing spatial and time coordinates into two separate vectors i.e.,



U and ***V*** are unitary matrices and $\sum$ is diagonal matrix with decreasing and nonnegative diagonal entries.

We truncate the above matrices in order to eliminate noise and we get the following,



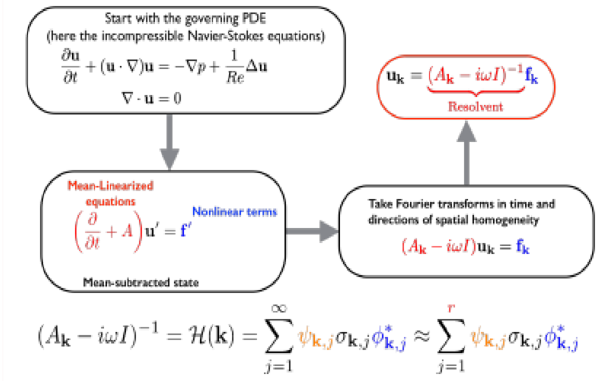
We then solve this expression and obtain

$$Y_r = U_r \Sigma_r V_r^* = \sum_{j=1}^{r} \sigma_j \boldsymbol{u}_j \boldsymbol{v}_j^*$$

Here uj and vj are vectors that are the j-th columns of Ur and Vr.

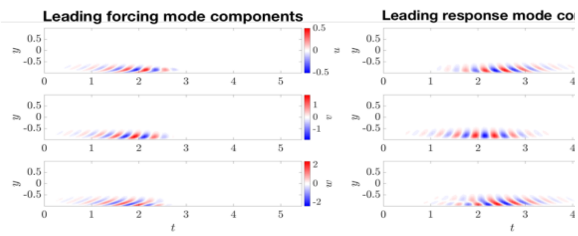
We use predicted coherent structures from a method called resolvent analysis, which looks at a singular value decomposition of a linear operator obtained from the governing Navier-Stokes equations

FINAL GROUP REPORT



The leading left and right singular vectors give the optimal response and forcing modes of the system.

Sample space-time modes in turbulent channel flow



Curve Fitting

The absolute and real components of u_j *and* v_j matrices are fitted using gaussian functions with the help of **curve_fit** function in Python and MATLAB.

The general form of the gaussian function assumed for fitting the components is,

$$\psi(y,t) = c\exp\left[i a_y y - b_y (y - y_0)^2\right]\exp\left[i a_t t - b_t (t -\right.$$

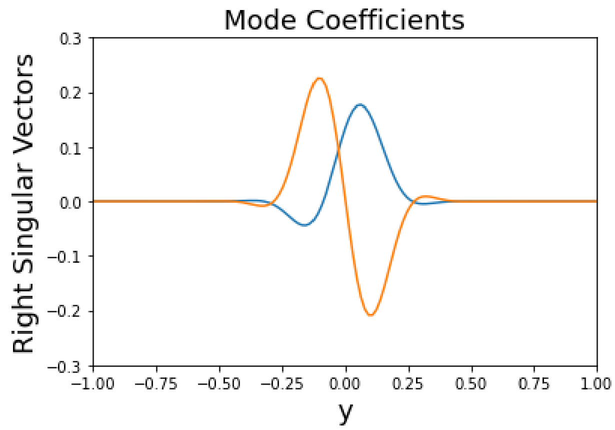
The unknown parameters of these functions are determined using curve fitting tools. For a better fit we enforce additional boundary conditions by adding additional image wavepackets.

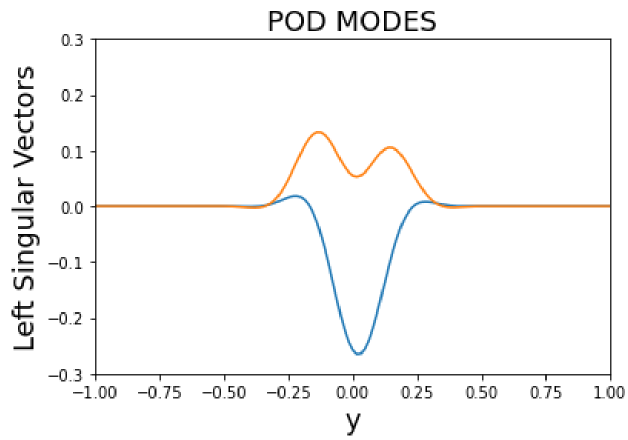
Results

Using the method discussed in the previous section, we have succeeded in creating analytical functions for various datasets. The results are shown and discussed below –

For TestMatrix –

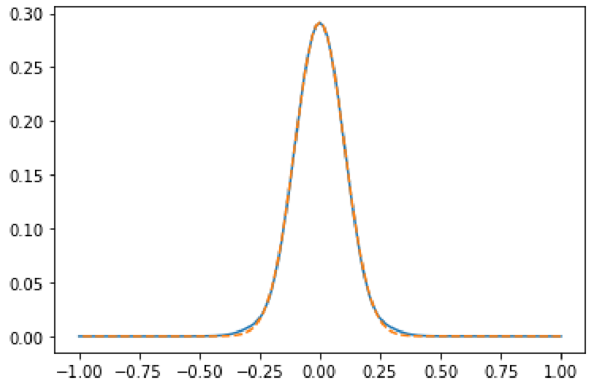
For our first dataset, we extracted the left and right singular vectors using built-in SVD tools in Matlab and python.



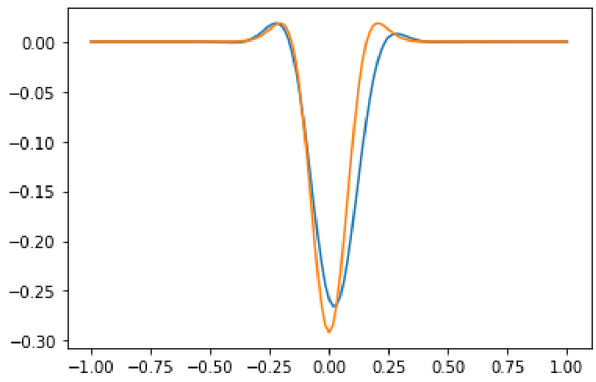


Now since we have separated space and time variables, we are in a position to fit them. Left Singular Vector is a function of space while Right Singular Vector is a function of time. For our first dataset, we fitted the absolute value of the above

vectors using one gaussian and the results were-

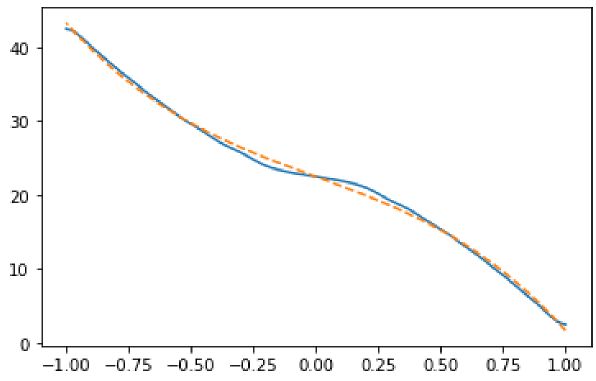


Fit of abs Left Singular Vector



Fit of real Left Singular Vector

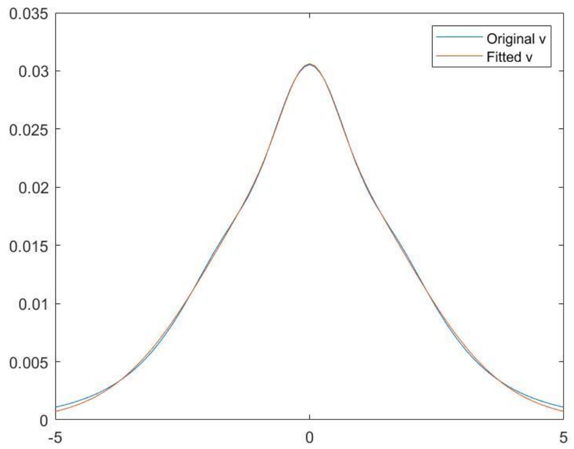
We also looked at the phase of the Left Singular Vector and fitted it.



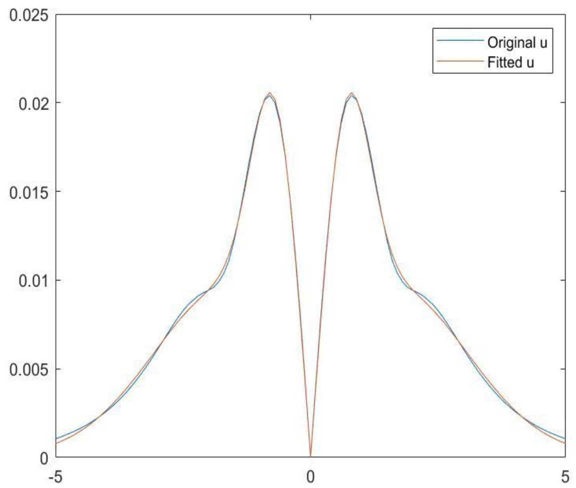
Now that we also have the phase, we can fit the real part of the vectors also. This framework has been used by us for the other datasets also.

For Flow over a Cylinder Data –

Here, we are going over some real world data and using the same analysis as before -

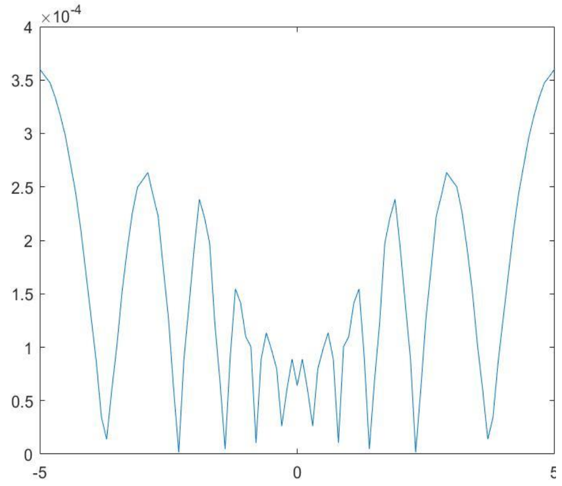


This fit was achieved using two gaussians, one for the inner region and the other accounting for the outer.

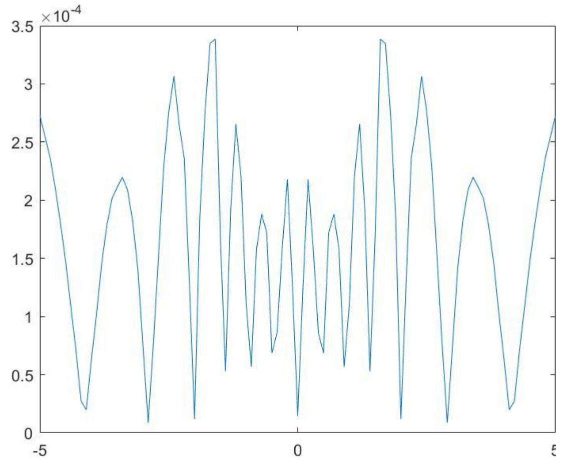


The above graph was obtained using 4 gaussians paired with a starting points guess as well as upper and lower bounds. These two graphs represent the u and v component of velocity of flow over a cylinder.

The error in both these cases is shown by the following graphs-



Error for v component of velocity

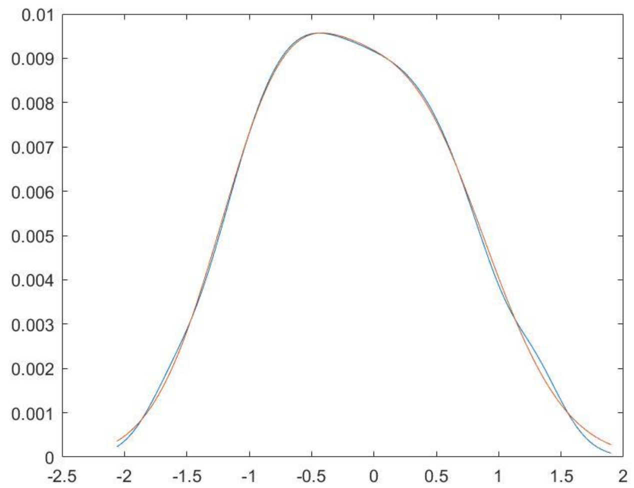


Error for u component of velocity

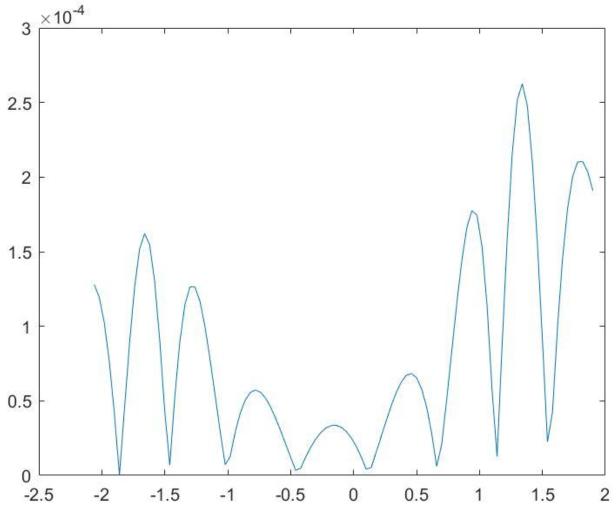
As we can clearly see, the error is negligible. That means our analytical function works and can be used for studying the dataset. Similarly, we were able to successfully create functions for other datasets also beginning with airfoil data.

For Flow over an Airfoil Data -

For this first plot, only 2 gaussians were required to achieve the necessary fit.



Velocity fitting graph



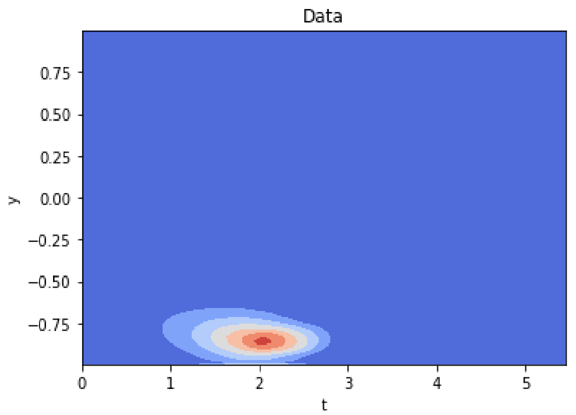
Error plot

As we can see, the error in this case is also negligible which is a good thing as our analysis has now worked on two sets of real world data. We will now look at some other data to test our theory even further.

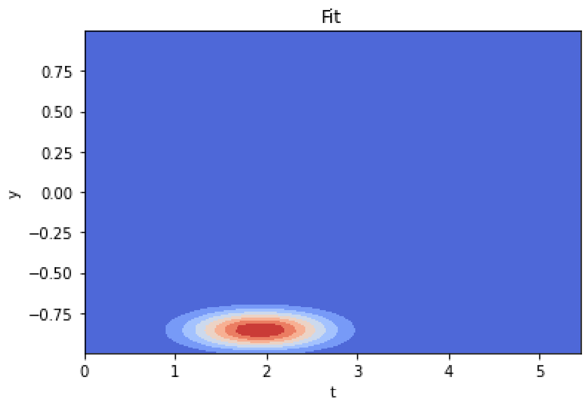
FINAL GROUP REPORT

For DataModes –

The below graph is a contour graph of the absolute value of PhiU. The second graph is obtained after finishing our fitting in time with one gaussian and two gaussian plus boundary conditions for space. As we can see, they match to a great extent.

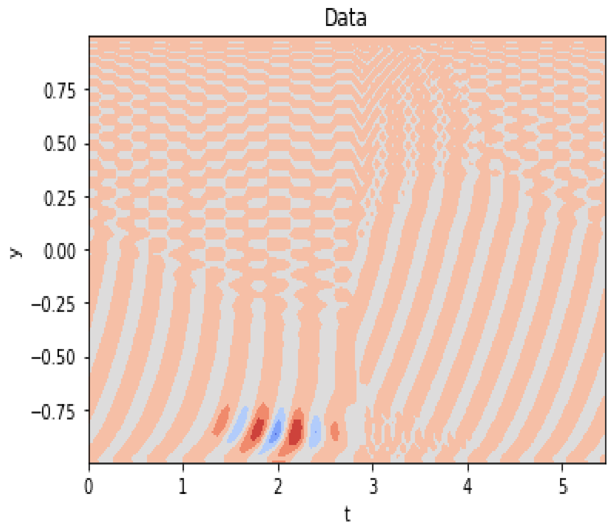


Original abs PhiU velocity

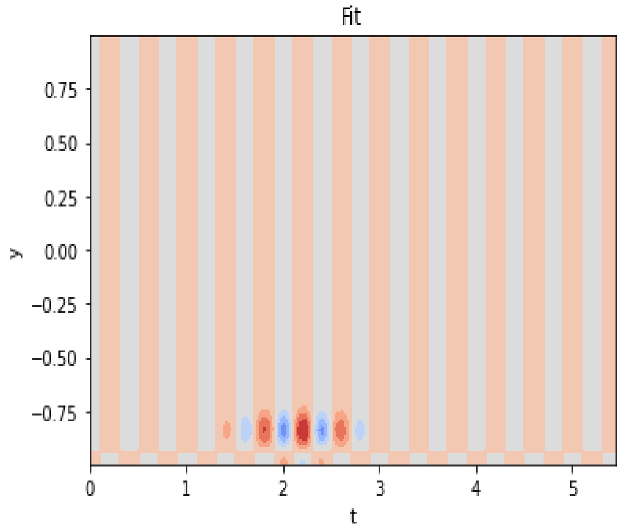


Fitted abs PhiU velocity

The next two graphs are of the real part of PhiU where we have extracted the phase and then fit a linear function to it using our fitting tools and combined it with our absolute valued function to get the fit of the real part. The graph matches the original in the x-direction but deviates a little in the y due to the phase.

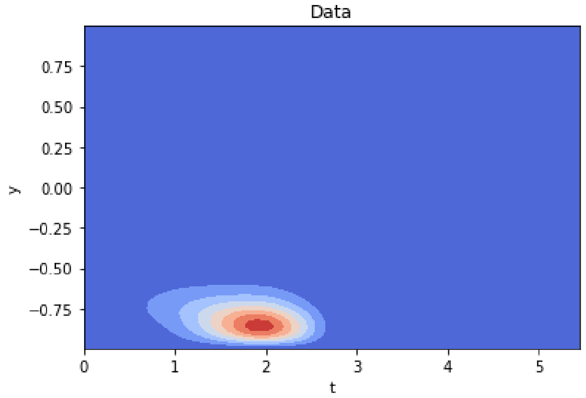


Original real PhiU velocity

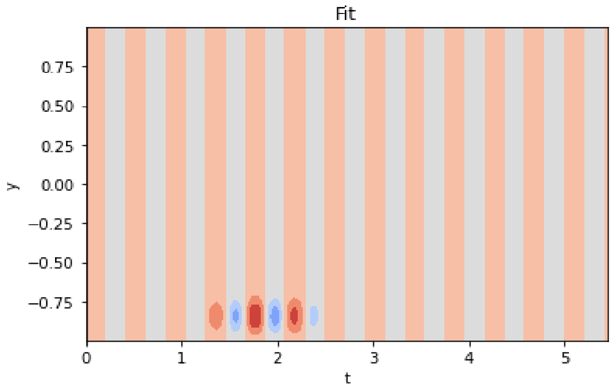


Fitted real PhiU velocity

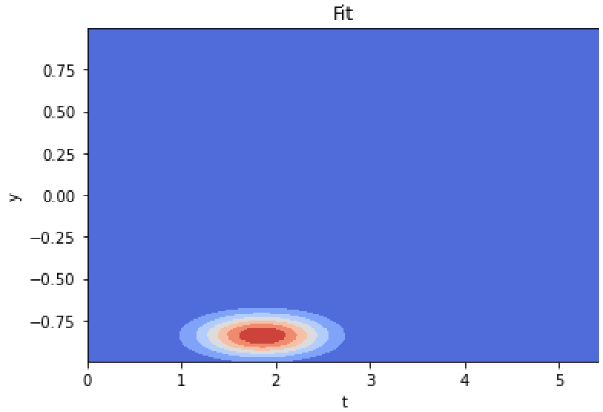
The same trend continues for PhiV, PsiU, PsiV and PsiW wherein we select that number of gaussian which reduces error and if necessary boundary conditions as shown by the following plots in the next few pages -



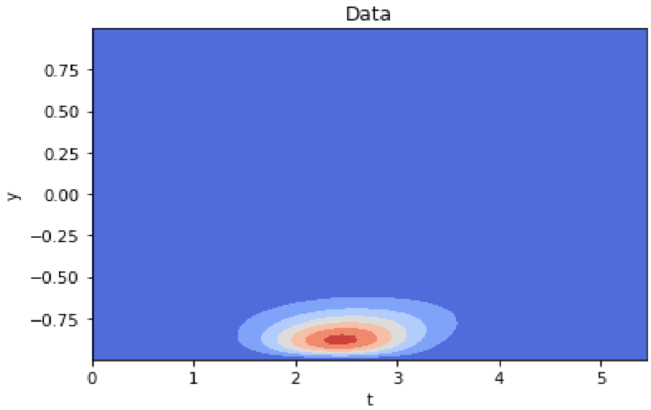
Original abs PhiV velocity



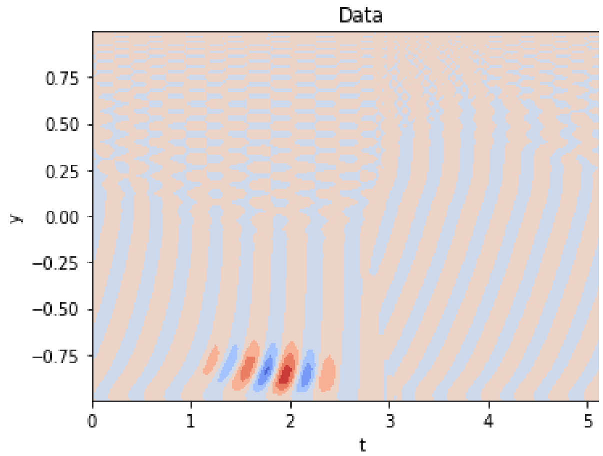
Fitted real PhiV velocity



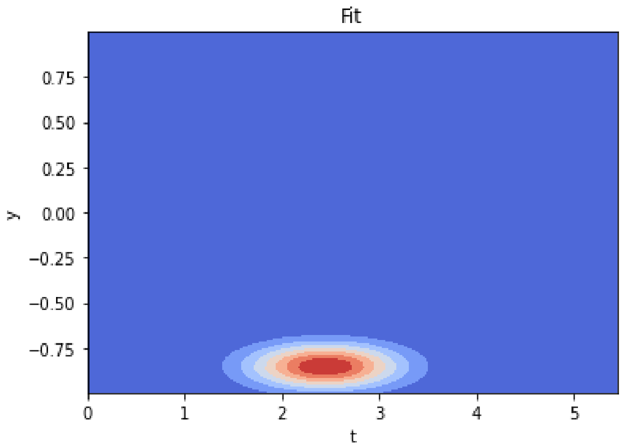
Fitted abs PhiV velocity



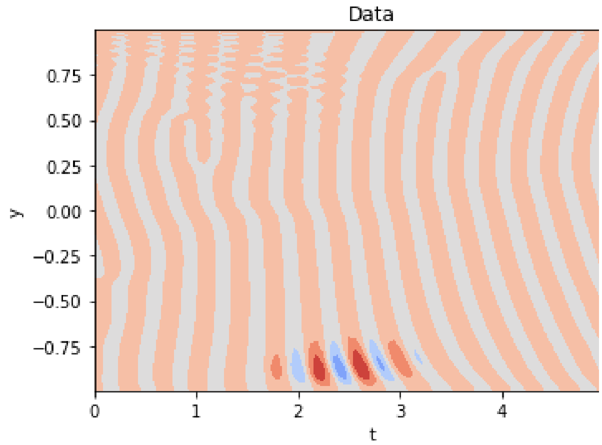
Original abs PsiU velocity



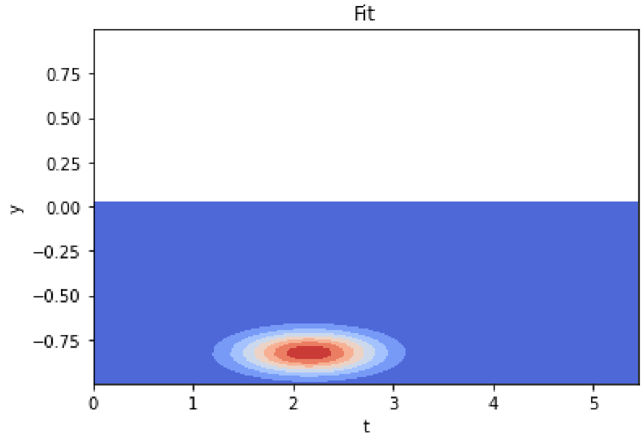
Original real PhiV velocity



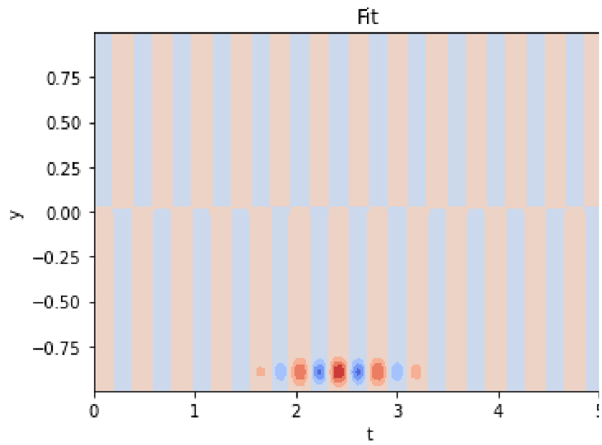
Fitted abs PsiU velocity



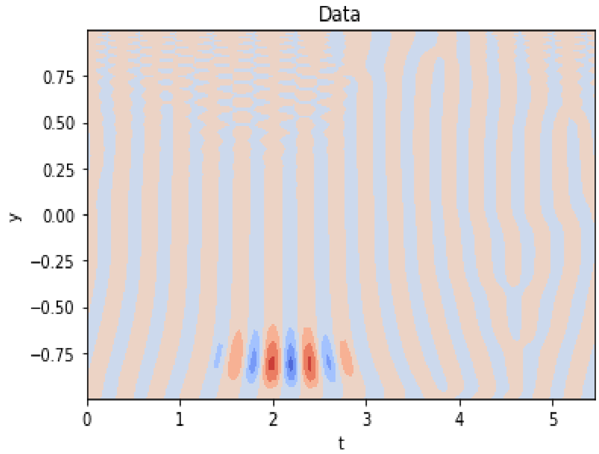
Original real PsiU velocity



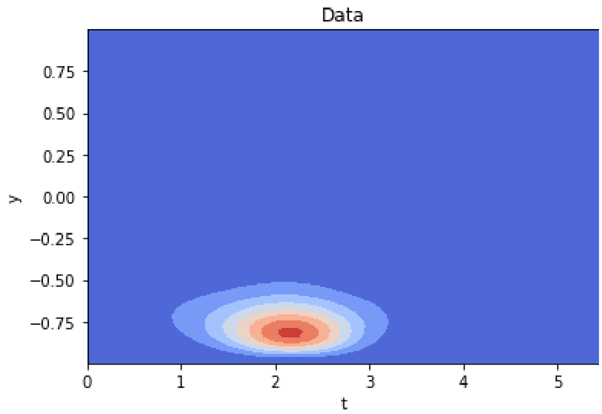
Fitted abs PsiV velocity



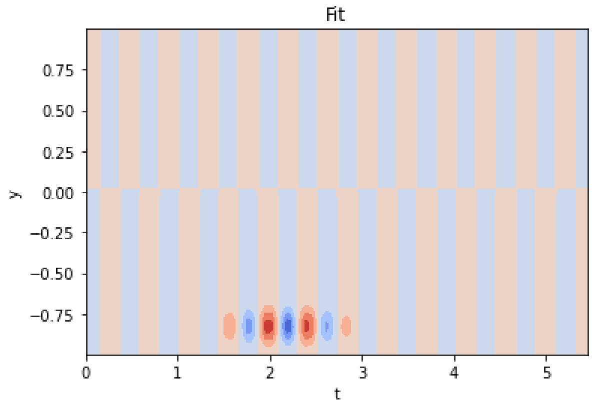
Fitted real PsiU velocity



Original real PsiV velocity

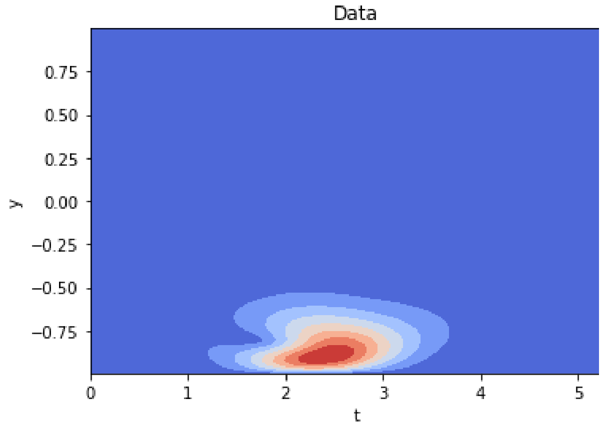


Original abs PsiV velocity

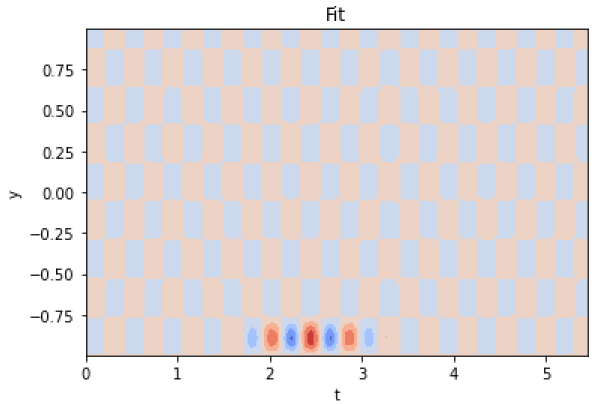


Fitted real PsiV velocity

FINAL GROUP REPORT



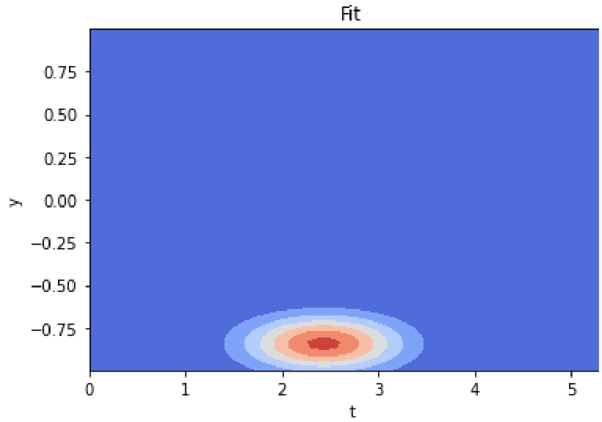
Original abs PsiW velocity



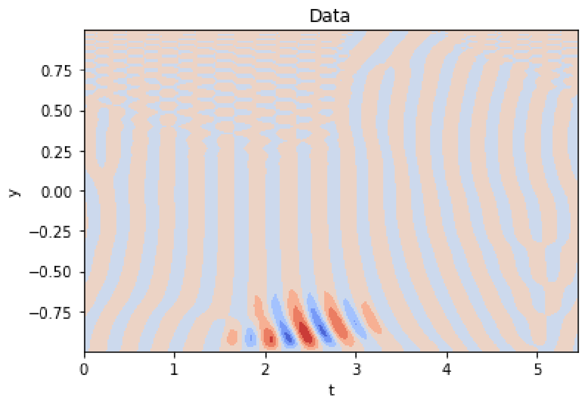
Fitted real PsiW velocity

For DataModesV3-

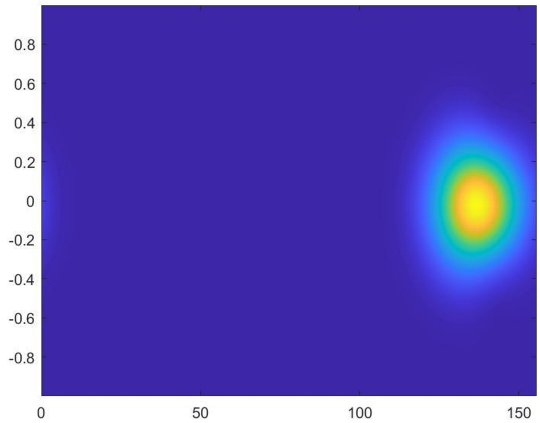
As evident by the name, DataModesV3 is an updated dataset of our previous DataModes. We have once again used our framework to create functions that represent the original data to a great extent as evident by the following plots-



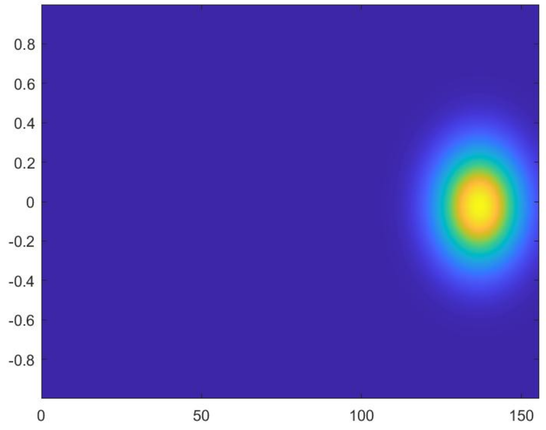
Fitted abs PsiW velocity



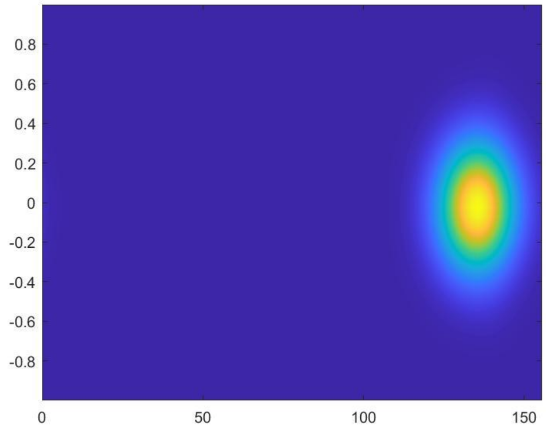
Original real PsiW velocity



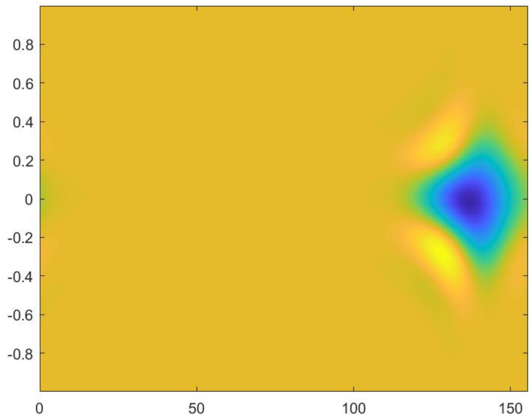
Original abs PhiU velocity



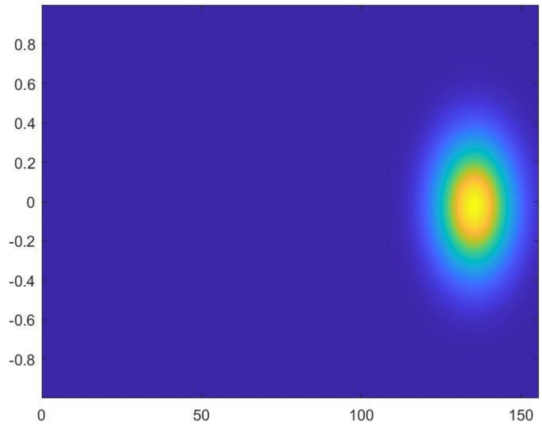
Fitted abs PhiU velocity



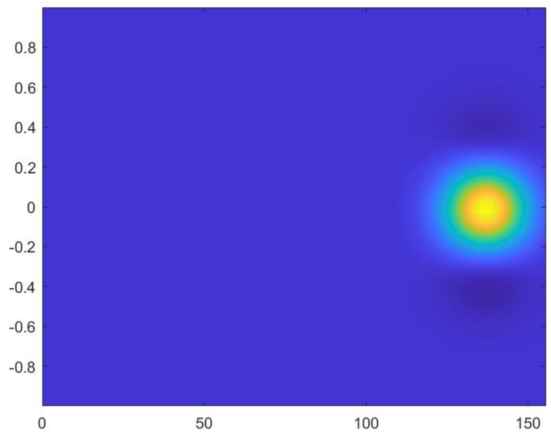
Original abs PhiV velocity



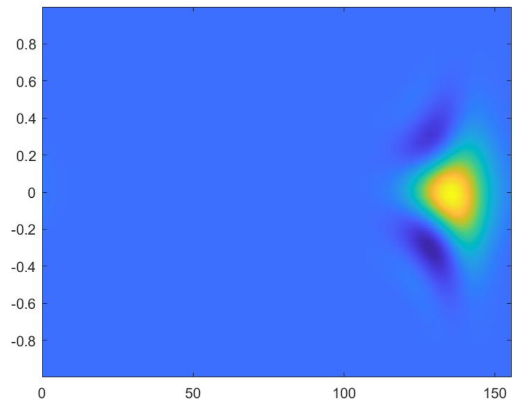
Original real PhiU velocity



Fitted abs PhiV velocity

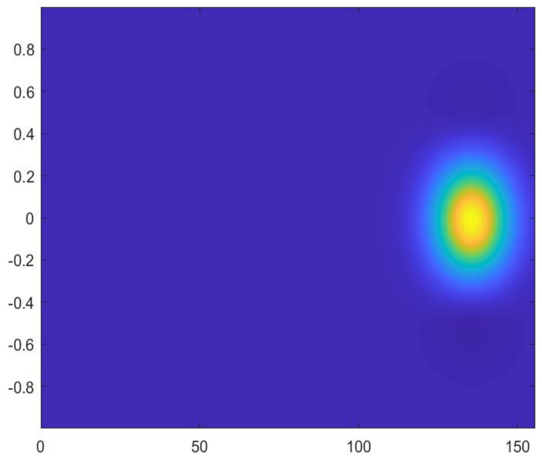


Fitted real PhiU velocity



Original real PhiV velocity

FINAL GROUP REPORT



Fitted real PhiV velocity

Conclusion –

After performing our analysis on a variety of data including both real world and theoretical and looking at the results that we have obtained, it is clear that our theory works and error is also minimal. But this theory does not provide us with a straight forward path as we have seen that in many cases, we had to select more than one gaussian sometimes even four and on top of that we also had to use boundary conditions and start point guesses sometimes in order to achieve a better fit. But our analysis also has its flaws as when it came to the real part of a function in DataModes, we were not able to get a great fit perhaps due to the phase not matching to a great extent or some other underlying issue. But apart from this, our error was less than 1% in flow over a cylinder, air-foil data, DataModesV3 and TestMatrix dataset.

Further Work –

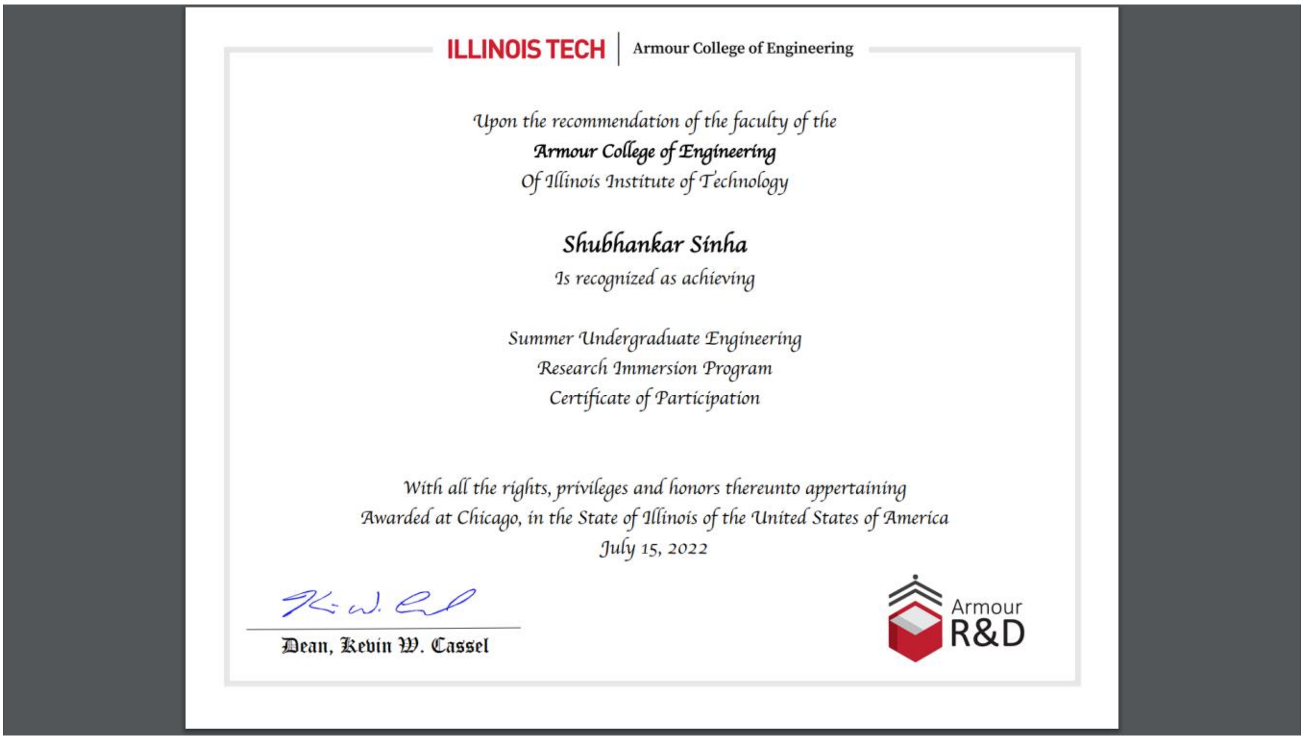
In our work, we have done our analysis mostly on datasets that resemble a gaussian function in some form or the other. Further work may be done to expand our theory to cover a wide range of functions and eliminate some of the issues we encountered in our present work.

Acknowledgement –

We would like to thank you sir for giving us the opportunity to do this research which allowed us to hone our skills in Matlab and python and we also got to know about the basics of research work for which we are eternally grateful. We would also like to wish you good health and success in your future work.

FINAL GROUP REPORT

CERTIFICATE-



ILLINOIS TECH | Armour College of Engineering

Upon the recommendation of the faculty of the
Armour College of Engineering
Of Illinois Institute of Technology

Shubhankar Sinha
Is recognized as achieving

Summer Undergraduate Engineering
Research Immersion Program
Certificate of Participation

With all the rights, privileges and honors thereunto appertaining
Awarded at Chicago, in the State of Illinois of the United States of America
July 15, 2022

Dean, Kevin W. Cassel

Armour R&D