**REWARD DRIVEN EMOTION DETECTION IN AUTISM**

**SPECTRUM DISORDER WITH ATTENTION MECHANISM**

**A Project-1 Report**

***Submitted by***

|  |  |
| --- | --- |
| Rahul Prasanth D | 2020506070 |
| Bala Natesh R M | 2020506018 |
| Sanjay G | 2020506080 |

***Under the supervision of***

Dr. J. Dhalia Sweetlin

*In partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

*in*

**INFORMATION TECHNOLOGY**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**MADRAS INSTITUTE OF TECHNOLOGY CAMPUS**

**ANNA UNIVERSITY, CHENNAI – 600044**

AUGUST 2023

**ANNA UNIVERSITY: CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this mini-project report titled “**REWARD DRIVEN EMOTION DETECTION IN AUTISM SPECTRUM DISORDER WITH ATTENTION MECHANISM” is** the bonafide work of Rahul Prasanth D (2020506070), Bala Natesh R M (2020506018) and Sanjay G (2020506080) who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **Signature** | **Signature** |
| **Dr. M. R. Sumalatha** | **Dr. J. Dhalia Sweetlin** |
| **HEAD OF THE DEPARTMENT** | **SUPERVISOR** |
| Professor | Associate Professor |
| Department of Information Technology | Department of Information Technology |
| MIT Campus, Anna University  Chennai – 600044 | MIT Campus, Anna University  Chennai – 600044 |

**ACKNOWLEDGEMENT**

It is essential to mention the names of the people, whose guidance and encouragement made us accomplish this project.

We express our thankfulness to our project supervisor **Dr. J. Dhalia Sweetlin**, Associate Professor of the Department, Department of Information Technology, MIT Campus, for providing invaluable support and assistance with encouragement which aided to complete this project.

We are thankful to the panel members **Dr. P. Anantha Kumar,** and **Dr. M. Hemalatha,** Department of Information Technology, MIT Campus for their invaluable feedback in reviews.

Our sincere thanks to **Dr. M. R. Sumalatha**, Head of the Department of Information Technology, MIT Campus for catering all our needs giving out limitless support throughout the project phase.

We express our gratitude and sincere thanks to our respected Dean of MIT Campus, **Dr. J. Prakash**, for providing excellent computing facilities throughout the project.

**RAHUL PRASANTH D 2020506070**

**BALA NATESH R M 2020506018**

**SANJAY G 2020506080**

**ABSTRACT**

Autism Spectrum Disorder (ASD) poses unique challenges in emotional expression and understanding, significantly impacting communication and social interactions in affected children. Emotion recognition in ASD individuals is particularly challenging due to variations in expressive behaviours. This project aims to address these challenges by developing a specialized system for comprehensive emotion recognition in autistic children. The proposed system utilizes a multimodal approach, incorporating facial expressions and their body movements to provide an elaborate view of emotional states. To overcome challenges related to occluded faces, Generative Adversarial Networks (GANs) are employed to reconstruct complete face from the occluded faces. Landmark detection techniques and the attention mechanism along with the reward function enhance emotion prediction accuracy. The system aims to provide valuable insights into the emotional experiences of autistic children, aiding educators, therapists, and caregivers in offering tailored support and interventions. The proposed work improves communication and emotional well-being in autistic children.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER**  **NO** | **TITLE** | | **PAGE NO** |
|  | **ABSTRACT**  **LIST OF TABLES**  **LIST OF FIGURES**  **LIST OF ABBREVIATIONS** | iv  viii  ix  x | |
| **1** | **INTRODUCTION** |  | |
|  | 1.1 Overview | 1 | |
|  | 1.2 Research Challenges | 1 | |
|  | 1.3 Objective  1.4 Scope of the Project  1.5 Contribution | 2  2  2 | |
|  | 1.6 Generative Adversarial Network | 3 | |
|  | 1.7 Attention Mechanism  1.8 Reward Function  1.9 Organization of the Thesis | 3  4  4 | |
| **2** | **LITERATURE SURVEY**  2.1 Facial dynamics of Autistic Children  2.2 Feed Forward Deep CNN  2.3 Discriminative Few Shot Learning  2.4 Facial Action Unit Detection  2.5 MultiLabel, MultiDomain Learning  2.6 Deep Cascade Guidance Learning  2.7 Adaptive Attention and Relation  2.8 Summary of Literature Survey | 8  8  9  9  10  11  11  11 | |
| **3** | **SYSTEM ARCHITECTURE AND DESIGN** |  | |
|  | 3.1 System Architecture | 13 | |
|  | 3.2 Face Reconstruction | 14 | |
|  | 3.3 LSTM | 14 | |
|  | 3.3.1 Attention Mechanism | 15 | |
|  | 3.3.2 Reward Function | 15 | |
| **4** | **ALGORITHM DEVELOPMENT AND** **IMPLEMENTATION**  4.1 Data Preprocessing | 17 | |
|  | 4.2 Landmark extraction  4.3 LSTM Model Implementation  4.4 Attention mechanism  4.5 Reward Functions | 17  18  18  18 | |
|  |  |  | |
| **5** | **RESULTS AND DISCUSSIONS** | 21 | |
|  | 5.1 Preprocessed data | 21 | |
|  | 5.2 Landmark points position  5.3 Reward Functions  5.4 Performance Comparison between Reward function | 22  23  23 | |
| **6** | **CONCLUSION AND FUTURE WORK** |  | |
|  | 6.1 Conclusion  6.2 Future Work  **REFERENCES** | 24  25  26 | |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TITLE** | **PAGE NO** |
| 4.1  4.2 | State Value Table  Action Value Table  viii | 19  20 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | P**AGE NO** |
| 1.1  3.1  5.1  5.2  5.3  5.4 | Reinforcement Learning  System Architecture  Server to Display Delay  Initial Display Delay  Delay Comparison  Sample Frames | 4  12  22  22  23  23 |
|  |  |  |
|  | ix |  |

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
|  |  |  |

**CHAPTER 1**

**INTRODUCTION**

**1.1 OVERVIEW**

Understanding and expressing emotions can be particularly challenging for children with Autism Spectrum Disorder (ASD). Traditional methods of recognizing emotions often miss the complex cues that are essential for effective communication in this population. This project addresses this gap by introducing a specialized system designed to decode the emotions of autistic children in a comprehensive way. This work goes beyond just facial expressions, encompassing body movements to provide a elaborate view of emotional states. Autistic children often communicate emotions differently, and by considering these diverse modalities, this system aims to create a more accurate and sensitive understanding. The proposed work acknowledges the everyday scenarios where faces may be partially hidden, and by using advanced techniques like Generative Adversarial Networks (GANs) to fill in the gaps and reconstruct complete facial expressions. The system is designed to be a bridge, helping educators, therapists, and caregivers connect more effectively with these children. By tapping into where and when emotions happen on their faces, coupled with specific facial expressions and body language, tools can be created that offers practical and meaningful support.

**1.2 RESEARCH CHALLENGES**

Developing the emotion recognition system in autistic children have intricate challenges that demand innovative solutions. The main challenge is the issue of facial occlusion, where traditional methods often struggle to precisely capture facial expressions when faces are partially hidden. Another critical challenge lies in the diverse and complex nature of emotional expression among autistic children. Conventional emotion recognition models, trained on datasets predominantly composed of neurotypical individuals, may not adequately capture the unique ways in which autistic children convey emotions. Autistic individuals often exhibit a broad spectrum of expressive behaviours, and the lack of diversity in training data can lead to models that struggle to generalize across this spectrum. Moreover, the scarcity of labelled data specific to autistic children poses a significant hurdle. The complex nature of emotions requires a substantial amount of diverse and accurately labelled data for effective model training. Acquiring such data is challenging due to privacy concerns, ethical considerations, and the need for expertise in labelling the complex emotional expressions.

**1.3 OBJECTIVE**

The primary objective of this system is to develop a specialized emotion recognition system tuned to the complex expressive behaviours of autistic children. By exploring the facial expressions and body movements, the goal is to construct a comprehensive tool that works beyond the traditional recognition methods. In addition to this, the system aims to address the challenge of occluded faces by incorporating methods for face reconstruction when certain parts are hidden from view. It aspires to construct a bridge connecting educators, therapists, and caregivers to the emotional experiences of autistic children

**1.4 SCOPE OF THE PROJECT**

The scope of this project encompasses the development of a specialized emotion recognition system tailored to the complex expressive behaviours of autistic children. The project will focus on exploring facial expressions and body movements as key modalities for emotion recognition. A significant aspect of the project's scope involves addressing the challenge of occluded faces by incorporating methods for face reconstruction when parts of the face are hidden. However, it's important to note that the project concentrates specifically on enhancing emotion recognition to improve communication and support.

**1.5 CONTRIBUTION**

The project contributes by developing a specialized emotion recognition system for autistic children, incorporating a reward mechanism to enhance performance. Addressing occluded faces and focusing on facial expressions and body movements, the system provides valuable insights to educators, therapists, and caregivers, ultimately improving understanding and support for the emotional experiences of autistic children.

**1.6 GENERATIVE ADVERSARIAL NETWORK**

Generative Adversarial Networks (GANs) operate through a dynamic interplay between two neural networks: a generator and a discriminator. The generator creates synthetic data, attempting to replicate realistic facial expressions, while the discriminator evaluates whether the generated faces are genuine or artificial. In the context of this project, when facial expressions are partially obscured, the generator, driven by the GAN, learns to reconstruct complete faces by filling in missing features. The adversarial training process refines both the generator's ability to create lifelike faces and the discriminator's capacity to distinguish between real and generated faces. This iterative process ensures that the GAN-enhanced system becomes skilled at reconstructing complete facial expressions, effectively addressing the challenge of occluded faces in the emotion recognition system for autistic children.

**1.7 ATTENTION MECHANISM**

The attention mechanism this system to focus on specific facial expressions and body movements which is crucial for emotion recognition. This targeted focus is particularly vital in scenarios where certain features are obscured or when dealing with the diverse ways in which autistic children express emotions. By incorporating an attention mechanism, the system's sensitivity and precision will be increased, providing a elaborate understanding of the unique emotional experiences of autistic children. This attention mechanism significantly elevates the effectiveness of emotion recognition systems in the context of autism.

**1.8 REWARD FUNCTION**

The proposed work introduces a reward function during which acts as a guide for the model. Unlike traditional approaches, the reward function plays a transformative role by encouraging the model's ability to correctly identify and interpret emotions, helps the model to prioritize precision. By positively reinforcing correct identifications, the system not only refines its understanding of the diverse emotional expressions inherent to ASD but also adapts dynamically to complex cues. This reward mechanism essentially guides the model towards more reliable and complex emotion detection, contributing to an enhanced and refined system tailored to the complexities of Autism Spectrum Disorder.

**1.9 ORGANIZATION OF THESIS**

The rest of the thesis is organized as follows. Chapter 2 presents the literature survey on emotion detection methodologies. Chapter 3 models the architecture and system design, outlining the architecture and design of the proposed system. Chapter 4 explains about the implementation details, explaining the specifications and environment. The results achieved are presented in Chapter 5. Chapter 6 presents the conclusion and some possible avenues for future research on the topic.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 FACIAL DYNAMICS OF AUTISTIC CHILDREN**

Toddlers were shown developmentally-appropriate and engaging movies presented on a smart tablet whose, frontal camera was used to record their faces, providing the opportunity for the automatic analysis via CV (Krishnappa Babu et al., 2023). The facial landmarks’ dynamics of the toddlers were studied specifically. The children’s facial dynamics were exploited from the eyebrows and mouth regions using multiscale entropy (MSE) analysis to study the complexity of such facial landmarks’ dynamics.

Distinctive landmarks’ dynamics were captured in children with ASD, characterized by a significantly higher level of complexity in both the eyebrows and mouth regions when compared to typically-developing children. In Cross-Validation using Decision Tree model the accuracy for each video shown is found to be Video1=77.5%, Video2=74.3%, Video3=73.8%, Video4=67.2%. The study sample has a limited number of ASD participants and did not have sufficient power to determine the impact of demographic characteristics on the results.

**2.2 FEED FORWARD DEEP CNN**

The proposed Face Detection Convolutional Neural Network (FDCNN) model exhibits a structured architecture tailored for emotion recognition in images derived from video frames (R Santoshkumar et al., 2019). With an input size of 150x150x3, representing the RGB channels, the model incorporates three convolutional layers, each applying 3x3 filters. The initial layer employs 32 filters, leading to 32 stacked feature maps, followed by max pooling to reduce spatial dimensions. Subsequent layers involve 64 and 128 filters, with corresponding max pooling operations, resulting in feature maps of dimensions (75x75x32), (37x37x64), and (18x18x128), respectively. The model culminates in fully connected layers, featuring 512 hidden units and an output layer with 15 neurons, aligning with the number of emotion classes. Training and validation sets are generated by converting input videos into frames, facilitating the model's training and evaluation.

This architecture conforms to the standard convolutional neural network paradigm for image classification tasks, utilizing convolutional layers to capture hierarchical features and max pooling for spatial downsampling. The fully connected layers contribute to the extraction of high-level representations before producing emotion predictions. The model's design, combining convolutional and fully connected layers, underscores its effectiveness in discerning complex patterns within facial expressions, making it well-suited for applications in emotion recognition from video frames.

**2.3 DISCRIMINATIVE FEW SHOT LEARNING**

The Few-Shot Learning (FSL) approach, incorporates distribution calibration and adaptive posterior learning (Zhang et al., 2023). The FSL system, when combined with the fusion of feature levels from each scene, achieves an impressive accuracy of 91.72% on the Caltech ADOS video data. The scene-level fusion, reveals insights into the unequal distribution of diagnostic information across different scenes and asserting that ASD is a complex condition requiring nuanced phenotyping beyond conventional classification categories. The model begins by extracting spatio-temporal features from the video, employing a combination of K-SVD with Marginal Fisher Analysis (MFA) to derive more discriminative representations. The scene-level feature fusion strategy, requires manually splitting entire hour-long videos into 15 separate scenes by time markers and extracting facial-dynamics features of each scene.

**2.4 FACIAL ACTION UNIT DETECTION**

This approach for the detection and analysis of facial expressions, particularly focuses on the intricate Facial Action Units (FAUs) associated with muscle activations (Jacob et al., 2021). Recognizing the pivotal role of facial expressions in conveying nonverbal information, underscoring the nuanced nature of expressions, ranging from universally understood to individualized, necessitating the utilization of the Facial Action Coding System (FACS). The task of FAU detection is systematically addressed as a multi-label binary classification problem, taking into account the varying degrees of FAU activation.

The model incorporates cutting-edge attention-based techniques, incorporating separate attention maps for each action unit, and harnesses multi-task learning to leverage inherent relationships among tasks. The framework of the model comprises feature extraction, attention learning, and multi-task modules, featuring innovative loss functions for both feature discrimination and multi-label classification. The approach, capable of end-to-end training, attains state-of-the-art performance on public datasets, achieving a notable F1-score of 61.5. A thorough evaluation that includes ablative studies to assess the impact of design choices, tackles challenges posed by class imbalance and variations in head pose and expression within the dataset.

**2.5 MULTILABEL, MULTIDOMAIN LEARNING**

A datasetwise selective sigmoid cross-entropy loss function is formalized to simultaneously train a multitask, multilabel and multidomain model (Pons et al., 2022). VGG-16 and Resnet-50 serve as the basis for training models for emotion recognition and AU detection. Dedicated individual networks are trained for each task and dataset separately. The soft-max cross-entropy loss function is used for emotion recognition tasks, while the sigmoid cross-entropy function is employed for AU detection due to the multilabel nature of the latter task. The accuracy was found to be 54.3% for Single RestNet-50 and 84.2% for SJMT RestNet-50. This method addresses one of the challenges with discrete emotion recognition in the wild, which is the lack of large public labelled datasets.

**2.6 DEEP CASCADE GUIDANCE LEARNING**

The proposed method involves a three-stage guidance learning scheme—occlusion detection, face parsing, and face reconstruction. The first two stages are trained on both synthesized and real data domains, enabling domain-agnostic guidance for the subsequent reconstruction stage and effectively mitigating the prevalent domain gap issue. By disentangling input information into domain-agnostic and appearance inputs, the cascade guidance learning model significantly reduces reliance on domain-sensitive appearance details, resulting in a substantial enhancement in the performance of face reconstruction on real-world images. Two more reference modules based on masked attention models are used that demonstrate both effectiveness and efficiency in inpainting occluded facial parts.

**2.7 ADAPTIVE ATTENTION AND RELATION**

The adaptive attention regression network, integrated with local attention predefinition and global attention learning, captures both predefined dependencies by landmarks in strongly correlated regions and facial globally distributed dependencies in weakly correlated regions (Shao et al., 2023). An adaptive spatio-temporal graph convolutional network simultaneously reasons the specific pattern of each AU, the inter-dependencies among AUs, as well as the temporal correlations. Extensive experiments on benchmark datasets show that the approach achieves comparable performance in both constrained scenarios and unconstrained scenarios, and can accurately learn the regional correlation distribution of each AU. The Adaptive Attention Regression (AAR) method achieved an average F1-frame score of approximately 63.8 on the BP4D benchmark. The AAR network was tested on input images with misalignment errors and occlusions. If input images are severely misaligned AAR fails to precisely capture AU Region Of Interests (ROIs). The AAR does not explicitly process misalignment errors, such as explicitly learning rotation-invariant and scale-invariant features.

**2.8 SUMMARY OF THE LITERATURE SURVEY**

**CHAPTER 3**

**SYSTEM ARCHITECTURE AND DESIGN**

* 1. **SYSTEM ARCHITECTURE**

The proposed project architecture is designed as a robust pipeline for the prediction of emotions of Autism Spectrum Disorder (ASD) children. The process initiates with Autism Videos serving as the primary dataset, and subsequent Frame Extraction facilitates the breakdown of video content into individual frames to enable granular analysis. Preprocessing steps ensure the frames are suitably prepared for subsequent analysis, including resizing and normalization procedures. The critical Face Occlusion Detection component determines whether facial features are partially obscured, prompting Face Reconstruction using Generative Adversarial Networks (GANs) when needed. In scenarios where faces are unoccluded, Face Landmark Extraction captures crucial facial features, while Body Landmark Extraction concurrently gathers key body landmark points which includes the hands, legs and so on. The architecture further integrates an LSTM (Long Short-Term Memory) layer, a recurrent neural network designed to comprehend temporal dependencies and patterns in the sequence of frames, enhancing the system's ability to understand dynamic facial and body expressions over time. Attention Mapping and the Reward Function are seamlessly incorporated into the LSTM layer, providing dual functionality to enhance focus on critical features and instill positive reinforcement during training. The Fully Connected Layer is connected to the Attention Layer, suggesting that the fully connected layer is incorporating information that has been selectively attended to by the attention mechanism. The reward during loss implies that, during training, the model is rewarded for making correct predictions or recognizing important patterns, enhancing its ability to learn and improve over successive iterations. In the final layer of the network architecture, the model outputs probabilities for six distinct emotion classes: anger, happy, surprise, neutral, fear, and sad. This output layer represents a comprehensive classification scenario for emotion recognition, assigning a probability distribution across these six classes for each input. Practical interpretation involves considering the class with the highest probability as the predicted emotion for a given input. This probabilistic approach provides a better understanding of the model's confidence in its predictions, allowing for a more detailed assessment of uncertainty or ambiguity in emotion recognition.

**Figure 3.1** – System Architecture

**3.2 FACE RECONSTRUCTION**

**3.3 LSTM**

**3.3.1 ATTENTION MECHANISM**

**3.3.2 REWARD FUNCTION**

**CHAPTER 4**

**ALGORITHM DEVELOPMENT AND IMPLEMENTATION**

**4.1 MACHINE LEARNING BASED APPROACH**

Machine Learning based approach is used in improving the reliability and performance of live stream by reducing the delay. Since the implementation is done in real time, both Supervised Learning and Semi-supervised Learning cannot be used since they don’t interact with the environment. Hence Reinforcement learning is used which helps in finding an optimal approach in a real time scenario. Further, Q-Learning approach which is based on Off-Policy Learning is implemented such that it helps in improving the action taken based on a reward in the current condition to minimise the delay. To select the best value from the Q-Table, Selection Policy is used.

**4.2 ALGORITHM**

1. Start

2. Initialize the Q-matrix consisting of state (s) and action (a) values, learning rate (α) and discount factor (γ)

3. Repeat for each state:

3.1 Observe current state (s) based on buffer availability.

3.2 Perform action (a) for the current state using selection policy.

//Selection policy function below

3.3 Compute the reward *(rwd)* value.

3.4 Update the value of reward (*rwd*) with the calculated value.

3.5 Compute new state (s') by measuring the buffer availability status.

3.6 Perform new action (a') and update the value of (s') using selection

policy.

3.7 Update the newly computed values in the Q-Table.

𝑄’(𝑠,𝑎)→𝑄(𝑠,𝑎) + 𝛼.[ *rwd* + 𝛾.𝑚𝑎𝑥 𝑄(𝑠’,𝑎’) − 𝑄(𝑠,𝑎)]

3.8 Update s' value (s 🡨 s')

3.9 Update new action (a 🡨 a')

3.10 The newly updated s and a values are sent to server.

4. End

**4.3 SELECTION POLICY**

1. Start

2. Initialize max <- 0, action <- 0, flag <- 0

3. For a = 1 to n:

3.1 Initialize Q(s,a)

4. For i=1 to n:

4.1 if Q(s,a) >= max:

4.1.1 max 🡨 Q(s,a)

4.1.2 action 🡨 i

5. End

**4.4 PROPOSED SYSTEM IMPLEMENTATION**

The system architecture proposed encompasses a client-server model. A schematic diagram of the client-server model along with its main functions. It represents the individual functions of the client and server and how their functionalities connect each other.

**4.4.1 Server**

The media locator URL of the server is initially set either to d-show API, that accesses the camera to capture the live video from the host system. The feed from the camera or from the stored video is encoded. Segments and Manifest Files are created accordingly.

**4.4.2 Client**

At the client side, the Media Player Description file plays an important role. The Media Player Description (MPD) file created by the server has the details of the segments regarding the live feed. The client gets the information about the stream such as the bitrate, frame rate of the stream and also the segment details. The media player is modified to handle streaming data. As the MPD file gets updated from time to time, the player should check the contents from the source buffer.

**4.4.3 State**

The buffer occupancy values are mapped to the state and each current state is shown in the Table 4.1.

**Table 4.1** – State Table

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Buffer Occupancy Value (In percentage)** | **State** |
| 1 | 0 – 10 | 1 |
| 2 | 10 – 20 | 2 |
| 3 | 20 – 30 | 3 |
| 4 | 30 – 40 | 4 |
| 5 | 40 – 50 | 5 |
| 6 | 50 – 60 | 6 |
| 7 | 60 – 70 | 7 |
| 8 | 70 – 80 | 8 |
| 9 | 80 – 90 | 9 |
| 10 | 90 – 100 | 10 |

**4.4.4 Action**

The action values mapped to the segment sizes accordingly and are shown in the Table 4.2.

**Table 4.2** – Action Table

|  |  |
| --- | --- |
| **Action** | **Segment Size (in seconds)** |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |

**4.5 IMPLEMENTATION ENVIRONMENT**

JavaScript programming environment is used for implementation purpose, since it runs on all browsers and is platform independent. For playing the media, Shaka player is used. Shaka is an open-source DASH player from Google. The server and the client both support MPEG – DASH. The client and server were implemented in Windows 10 (64 bit operating system). The client and server were connected to the 4G Mobile Hotspot network. The streaming was implemented on top of the HTTP/2.0 protocol with TCP as its underlying transport layer protocol.

**CHAPTER 5**

**RESULTS AND DISCUSSIONS**

**5.1 IMPLEMENTATION ENVIRONMENT**

A client-server environment for the dynamic adaptive streaming over HTTP is created. The client requests the server to send the video after getting connected via HTTP port. The details of the live feed get stored in the media player description file and the file is passed to the client. At the client side, the segments which are sent by the server are appended to the buffer and the player plays the contents from the buffer.

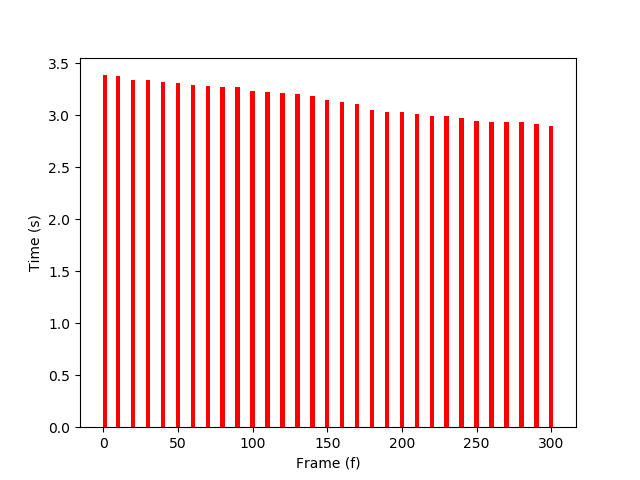
The proposed system is implemented and tested using an ACT broadband connection at the server end and an Airtel 4G Hotspot at the client end. The connection is established using TCP and it is supported on top of HTTP/2.0. The server uses NGROK tool along with d-show API to establish a connection with the client. The client receives the frames using Shaka Player and plays it for display.

**5.2 SERVER TO DISPLAY DELAY MEASUREMENT**

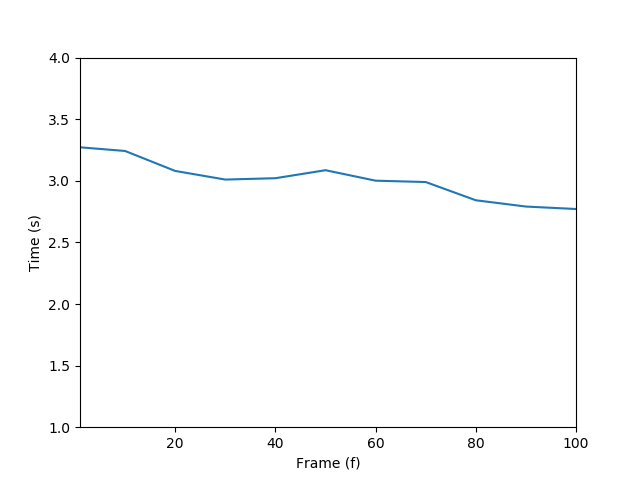
The Server to Display delay is a metric that determines the latency between a frame getting generated at the server side, pushed to stream and the frame viewed by the user at the client side. The Server to Display delay is calculated during the live streaming experiments and is plotted in Figure 5.1.

**5.3 INITIAL DELAY MEASUREMENT**

The initial server to display delay depends upon the segment size with which the live stream is initiated. The delay is a cumulative value of the segment size in seconds and the segment propagation time at the network to the client. The initial server to display delay for various initial segment size is shown in Figure 5.2.



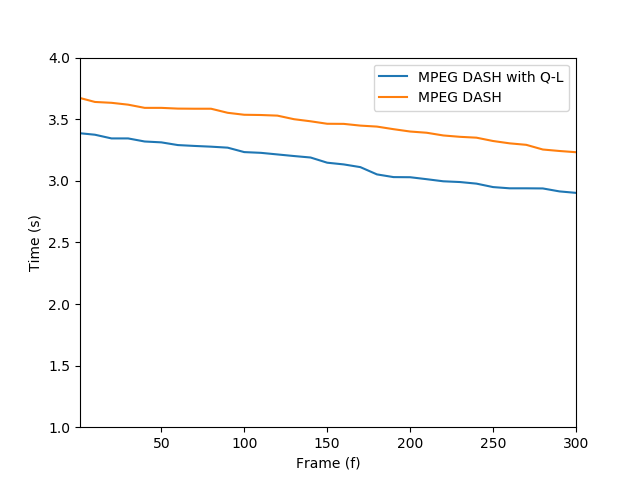
**Figure 5.1** – Server to Display Delay



**Figure 5.2** – Initial Display Delay

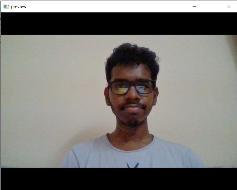
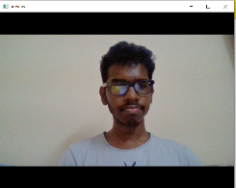
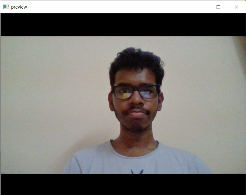
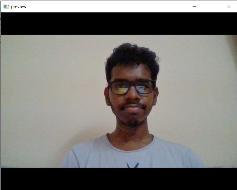
**5.4 DELAY COMPARISON**

The delay for the transfer of live stream was computed using a general MPEG-DASH approach and also with the implementation of Q-Learning. It shows that Q-Learning helps in reducing the delay further compared to a generic approach of DASH. This applies to two factors: the adaptive size of segments and buffer filling at client end. The observed delay for the two different approaches is plotted in Figure 5.3.



**Figure 5.3** – Delay Comparison

Few frame sequences of original video from the server side while experimenting with live streaming is depicted in Figure 5.4.



**Figure 5.4 –** Some frames captured during live streaming.

**CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

**6.1 CONCLUSION**

A live video streaming was implemented in a 4G wireless network on top of HTTP/2.0. The various challenges faced due to the changing link conditions are studied. Reduction of the delay is a requirement for improving the quality of the video and the overall quality of experience. As HTTP/2.0 is the protocol used here, the number of round trips are reduced and the delay reduction depends on the segment size of the video chunk which is pushed upon the request from client. The proposed system identifies the current buffer occupancy status at the client side. On a sudden bandwidth drop, or an abrupt change in the network, the buffer occupancy status gets greatly varied. The video chunks are sent in accordance with the status of the buffer at the client.

The buffer at client end may also get segments from the server very quickly that can lead to rapid pooling of segments at client side or slow down the segment arrival rate due to network congestions that could lead to the draining of available segments at the client side. The adaptation engine in the client side, determines the segment size with which the server must stream the chunks of video data to the client at regular intervals, so that there is neither any freezing occurring while playing the video nor the buffer faces any overflow or underflow. The Q-Learning works for streaming applications, but to achieve a better quality of experience for interactive applications, the streaming latency needs to be further reduced. The proposed method with experimental results could help in standardizing the future live streaming services and in delivering a minimum delay setup.

**6.2 FUTURE WORK**

The proposed system was implemented in one way video communication, but it can be extended to support two-way streaming. The system can also be structured in such a way that it enables the server to serve more than one client at a time. Furthermore, the streaming system need to provide adaptive support to both audio and video data streaming simultaneously at a very low latency. The system level implementation could be based on the ITU-T recommendation P.1203.1 which defines the MOS computation in a standard procedure.

**REFERENCES**

1. Bedogni, L., Di Felice, M. and Bononi, L., (2017), “Dynamic segment size selection in HTTP based adaptive video streaming”. In proceedings of the 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2nd May’ 17, pp. 665-670.

2. Bekcheol Jang, Myeonghwi kim, Gaspard Harerimana (2018), “Q-Learning Algorithms: A Comprehensive Classification and Applications”, In proceedings of the 2018 IEEE Conference on Computer Communication, 23rd October’ 18, pp. 294-296.

3. Huang, T.Y., Johari, R., McKeown, N., Trunnell, M. and Watson, M. (2015), “A buffer-based approach to rate adaptation: Evidence from a large video streaming service”, ACM SIGCOMM Computer Communication Review, 44(4), pp.187-198.

4. Huysegems, R., Van Der Hooft, J., Bostoen, T., Rondao Alface, P., Petrangeli, S., Wauters, T. and De Turck, F. (2015), “HTTP/2-based methods to improve the live experience of adaptive streaming”. In Proceedings of the 23rd ACM international conference on Multimedia, 28th October’ 15, pp. 541-550.

5. Hooft J.V.D., S. Petrangeli, T. Wauters, R. Huysegems, T. Bostoen, and F.D. Turck (2017), “An HTTP/2 push-based approach for low-latency live streaming with super-short segments”, Journal of Network and Systems Management, March’ 17, vol. 26, no. 1, pp. 51-78.

6. Le, H.T., Nguyen, T., Ngoc, N.P., Pham, A.T. and Thang, T.C., (2018), HTTP/2 Push-Based Low-Delay Live Streaming Over Mobile Networks With Stream Termination”, IEEE Transactions on Circuits and Systems for Video Technology, 28(9), pp.2423-2427.

7. Lohmar, T., Einarsson, T., Fröjdh, P., Gabin, F. and Kampmann, M., (2011), “Dynamic adaptive HTTP streaming of live content”. In proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, June’ 11, pp. 1-8.

8. Mohd Ramli, H., Sandrasegaran, K., Basukala, R., Patachaianand, R. and Afrin, S., (2011), “Video Streaming Performance Under Well-Known Packet Scheduling Algorithms”, International Journal of Wirless and Mobile Networks.

9. Nguyen, D.V., Le, H.T., Nam, P.N., Pham, A.T. and Thang, T.C., (2016), “Adaptation method for video streaming over HTTP/2”, IEICE communications Express, 5(3), pp.69-73.

10. Shuai, Y. and Herfet, T., (2018), “Towards reduced latency in adaptive live streaming”. In proceedings of the 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), 13th January’ 18, pp. 1-4.

11. Sobhani, A., Yassine, A. and Shirmohammadi, S., (2017), “A video bitrate adaptation and prediction mechanism for HTTP adaptive streaming”, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 13(2), p.18.

12. Stockhammer, T., (2011), “Dynamic adaptive streaming over HTTP: standards and design principles”. In Proceedings of the second annual ACM conference on Multimedia systems, 13th February’ 11, pp. 133-144.

13. Swaminathan, V. and Wei, S., (2011), “Low latency live video streaming using HTTP chunked encoding”. In proceedings of IEEE 13th International Workshop on Multimedia Signal Processing, 18th October’ 11, pp. 1-6.

14. Thang, T.C., Le, H.T., Pham, A.T. and Ro, Y.M., (2014), “An evaluation of bitrate adaptation methods for HTTP live streaming”, IEEE Journal on Selected Areas in Communications, 32(4), pp.693-705.

15. Ur Rahman, W. and Chung, K., (2015), “Buffer-based adaptive bitrate algorithm for streaming over HTTP”, KSII Transactions on Internet and Information Systems (TIIS), 9(11), pp.4585-4603.

16. Van Der Hooft, J., Petrangeli, S., Wauters, T., Huysegems, R., Bostoen, T. and De Turck, F., (2018), “An HTTP/2 push-based approach for low-latency live streaming with super-short segments”, Journal of Network and Systems Management, 26(1), pp.51-78.

17. Wang, Y.H., Li, T.H.S. and Lin, C.J., (2013), “Backward Q-learning: The combination of Sarsa algorithm and Q-learning”, Engineering Applications of Artificial Intelligence, 26(9), pp.2184-2193.

18. Wei, S. and Swaminathan, V., (2014), “Cost effective video streaming using server push over HTTP 2.0”. In proceedings of IEEE 16th International Workshop on Multimedia Signal Processing (MMSP), 16th September’ 14, pp. 1-5.

19. Xu, Z., Zhang, X. and Guo, Z., (2018), “QoE-driven Adaptive K-Push for HTTP/2 Live Streaming”, IEEE Transactions on Circuits and Systems for Video Technology.

20. Yun, D. and Chung, K., (2016), “Dynamic segment duration control for live streaming over HTTP”. In proceedings of the International Conference on Information Networking (ICOIN), 14th January’ 16, pp. 206-210.

21. FFmpeg - <https://www.FFmpeg.org>

22. NGROK - Secure introspectable tunnels to localhost <https://ngrok.com/>

23. NGINX – Web server for web serving [https://www.nginx.com/](https://www.nginx.com/%20)

24. Speedtest by Ookla - <https://www.speedtest.net/>