JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

Lost frame Detection and Recovery Model in a Multiview Video Transmission over Wireless Sensor Network using Neural Network

By

Adane Tadesse Gebryu

This thesis is submitted to School of Graduate Studies of Jimma University in partial fulfilment of the requirements for the degree of
Master of Science
in
Computer Engineering

October 2021
Jimma, Ethiopia

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

Lost frame Detection and Recovery Model in a Multiview Video Transmission over Wireless Sensor Network using Neural Network

By

Adane Tadesse Gebryu

Advisor: Dr. Kinde Anlay

Co-Advisor: Mr. Fetulhak Abdurahman

Submission Date: October 2021

Declaration

I, Adane Tadesse declare that this thesis titled "Lost frame Detection and Recovery Model in a Multiview Video Transmission over Wireless Sensor Network using Neural Network" and work presented in my own, except I refer different works that are previously done. I also declare that this thesis has not been previously submitted by any researcher in this and other institutions

RESEARCH THESIS SUBMITTED BY

Adane Tadesse ____________ ____________

SIGNATURE DATE

APPROVED BY ADVISORS:

ADVISOR: ____________ ____________

SIGNATURE DATE

CO-ADVISOR: ____________ ____________-

SIGNATURE DATE

Approved by Faculty of Electrical and Computer Engineering Research Thesis Examination members

1. Dr. Million Meshesha Million Oct. 20, 21

SIGNATURE DATE

2. Engr. Kris C. Calpotura, MSME ____________ October 20,

SIGNATURE DATE

3. ____________ ____________ ____________

SIGNATURE DATE

Acknowledgements

First and above all, I express gratitude toward God, the almighty for all his blessing throughout my life and for granting me the capacity to continue effectively in my investigations. I am heartily appreciative to every one of the individuals who supported me in any regard towards the completion of my Master thesis. I owe my deepest appreciation to my advisor Dr. Kinde Anlay and Mr. Fethulak Abduhrenman who supported me with there well-explained illustrations empowered me to build up a solid understanding of the subject in both practical works as well as the documentation. It is an honor for me to recognize the Jimma University which offered me a scholarship during the entire time of my master studies. Last, however, by no means least, I would like to express my deep gratitude to my lovely parents and friends for their personal support. Thanks to them for always being my main source of motivations for success.

Abstract

Multiview Video (MVV) is one of the emerging technology in recent years. The concept of MVV is becoming very important, during the implementation of 3D systems to enhancing the viewing of high-resolution videos and images from different angles. Streaming of MVV over a wireless sensor network (WSN) is very susceptible to whole-frame losses due to wireless channel errors and low-resolution cameras are used as sensor nodes in WSN. Along this, different studies try to develop error concealment techniques for MVC.

In this thesis, we propose the identification of a lost frame method by using a Machine Learning (ML) models and a recovery algorithm for a lost frames of MVV in WSN by using Long Short Term Memory (LSTM) regression method. The detection method uses video and image quality assessment techniques to extract the features from the MVV frame sequences. The recovery method uses motion estimation and disparity estimation techniques to extract and select features for LSTM regression algorithm from MVV frame sequences.

The performance of the proposed methods was analyzed on different MVV sequences. The experimental results of the proposed detection method have scored **93.12** % accuracy to detect the lost frames in MVV sequences. And the proposed LSTM based recovery algorithm has the capability to improve the video quality of MVC at the decoder side. Compared with the recent methods the proposed method exceeded the average Peak Signal to Noise Ratio (PSNR) upto **2.47dB**. The complexity of the proposed method also acceptable. This study is expected to open up new perspectives on how to detect and restore the missing frames in MVV transmission at real-time.

Keyword: ***Multiview Video, Lost Frame Detection, Motion Vector, Disparty Vector, Lost Frame Recovery, Peak Signal to Noise Ratio***

Contents

List of Figures

List of Tables

Abbreviations and Acronyms

1D	**O**ne **D**imension
2D	**T**wo **D**imension
3D	**T**hree **D**imension
ADNN	**A**rtificial **D**eep **N**eural **N**etwork
ANN	**A**rtificial **N**eural **N**etwork
AUC	**A**rea **U**nder the **C**urve
AVC	**A**dvance **V**ideo **C**oding
Bi-ConvLSTM	**Bi**directional **Conv**olutional **L**ong **S**hort-**T**erm **M**emory
Bi-LSTM	**Bi**directional **L**ong **S**hort-**T**erm **M**emory
BMA	**B**lock **M**atching **A**lgorithm
Conv3D	3 **D**imensional **C**onvolution **N**eural **N**etwork
ConvLSTM	**Conv**olutional **L**ong **S**hort-**T**erm **M**emory
dB	**D**ecibel
DBMA	**D**epth based **B**oundary **M**atching **A**lgorithm
DCT	**D**iscrete **C**osine **T**ransform
DV	**D**isparity **V**ector
EC	**E**rror **C**oncealment
ER	**E**rror **R**esilience
FSIM	**F**eature **S**imilarity **I**ndex
FVV	**F**ree **V**iewpoint **V**ideo
GM	**G**radient **M**agnitude
GOP	**G**roup **O**f **P**ictures
GOV	**G**roup **O**f **V**iew

HVS	**H**uman **V**isual **S**ystem
IQM	**I**mage **Q**uality **M**etrics
LSTM	**L**ong **S**hort-**T**erm **M**emory
MOP	**M**atrix **O**f **P**ictures
MPC	**M**atching **P**el **C**ount
MPR	**M**ultilayer **P**erceptron **R**egression
MV	**M**otion **V**ector
MVC	**M**ultiview **V**ideo **C**oding
MVV	**M**ultiview **V**ideo
NSS	**N**atural **S**cene **S**tatistics
PC	**P**hase **C**ongruency
PSNR	**P**eak **S**ignal-to-**N**oise
RF	**R**andom **F**ield
RNN	**R**ecurrent **N**eural **N**etwork
SAD	**S**um of **A**bsolute **D**ifferences
SCSF	**S**patial **C**ontrast **S**ensitivity **F**unction
ROC	**R**eceiver **O**perating **C**haracteristic
SSIM	**S**tructural **S**imilarity **I**ndex
TV	**T**ele**V**ision
VIF	**V**isual **I**nformation **F**idelity
VQM	**V**ideo **Q**uality **M**etrics
WMSN	**W**ireless **M**ultimedia **S**ensor **N**etwork
WSN	**W**ireless **S**ensor **N**etwork

Chapter 1

Introduction

1.1 Background

WSN (Wireless Sensor Network) is a new technology for collecting data with autonomous task oriented sensors. This technology became more attractive because of its good features relatively inexpensive, it needs low energy to operate, reliability, responsiveness, and mobility [1]. It is used for intelligent transportation, environmental monitoring, video surveillance and monitoring space [2, 3]. For example, most used WSN measures scalar physical appearances like humidity, pressure, temperature, or the position of objects. Most applications of WSN has become at low-bandwidth and delay-tolerant. At this time the accessibility of inexpensive hardware used as a sensor, like microphones and cameras which are used to record multimedia data from situations has supported the evolution of Wireless Multimedia Sensor Networks (WMSNs) [2–4]. In WMSNs, networks of wireless integrated devices that permit retrieving video signal, audio signal streams, scalar sensor data, and still images from the physical environment [5]. A lot of audio and visual sensor nodes are wirelessly interconnected in a WMSN, and every sensor takes a video view regarding target occasions and sends it to a central base station or a data sink [6]. WMSN can be regarded as a MVV system [7].

MVV transmission System is a fast-growing multimedia technology that is active in both production and research societies [8]. Different multiple cameras simultaneously capture a scene from a different perspective and transmit over some transmission channel is called MVV transmission. The content transmission method for MVV is the ability to transmit various simultaneous views of the same scene in the same period [8, 9]. The view could be span depending on the number of cameras. Numbers cameras are allowed the end-users to perceive a more comprehensive view of the scene. A single video camera may not take the full scenes, hence by using various cameras to cover an appropriate scene and by eliminating redundancy in a scene to obtain a comprehensive view [10, 11].

MVV is that the crucial technology for many applications, like immersive teleconferencing, Free-Viewpoint TV (FVT), 3DTV, and Free-Viewpoint Video (FVV). The conventional video is a Two-Dimensional (2D) signal that still uses proactive channels to allow audiences to visualize the situation. The MVV, on the other hand, does provide users with a variety of situational videos by providing random perspectives on different scenes. Different video frames are taken by multiple cameras at a similar time, though in various locations, which are involved in the MVV [12–14]. We need to compress the multiview sequence effectively by considering the quality of the frames in video sequences since this method has serious constraints in information transmission applications such as broadcast, high technological multimedia systems, and alternative conventional technologies. To transmit the MVV over the transmission it desires the involvement of MVC [15, 16].

MVC is a significant technology to establish the MVV transmission scheme to realize multiview illustrations with a suitable design for a bandwidth of wireless channel and decoder resource usage. MVV is obtained by different kinds of several cameras, like 1D (One-Dimension) parallel, 2D (Two-Dimension) parallel, and different arc types of cameras. Views are encoded using the components of the encoder which is the first section of MVC and transmit over the wireless medium. The MVC decoder constructs the coded MVVs. Since various types of display methods are available to TV audiences, the 3DTV scheme can aid the Intermediate

View Reconstruction (IVR) process for MVV displays where the number of views to be viewed exceeds the number of encoded views [12, 17, 18].

To transmit MVVs over WSN the MVC encoder deploys in every wireless sensor simultaneously to encode various kinds of views and video frame sequences to reach higher compression gain by taking advantage of inter-view redundancy [19]. After compression is done then transmit the bitstreams over the wireless transmission channel to the MVC decoder. The MVC decoder decoding each video frame pixels. So packet losses in MVV transmission time directly affect the views in the decoder side of MVC the received videos with errors like lost some pixel values of frames or lost full-frame [20].

In this case there are several VQMs to specify the standard of videos in decoder side. There are two types of video quality measurements: objective VQMs and subjective VQMs [21]. Subjective metric is based on the experiments led on a group of people who are deciding the quality of the video by observing it. There are three types of objective metrics: No-Reference (NR), Reduced Reference (RR), and , Full Reference (FR). In FR technique original frame compared with distorted frame when the original frame is available as reference frame. In the RR method, it provides regarding structure or another characteristic of the actual image or video. The input in this method is the comparison between reduced information from the actual frame and information from the distorted frame. The NR approach would not need accessibility of the actual image or video, instead of relying on information from the bitstream or searching for data in the pixel domain [20–23].

A lost frame from a sequence of frames in a received video can be detected by using video and image quality metrics. As discussed early, during transmission MVV distorted or lost frames is available on decoder side. Therefore, techniques to control the impacts of transmission errors are highly desirable. Error Concealment (EC) is one of the approaches to struggle with these issues. This technique used to hide the errors in the received frame at the decoder side, enhancing visual consistency while eliminating the need to adjust the encoder or transmission channel. Based on this, different machine learning algorithms are more effective or suited for controlling such errors in video transmission.

Machine Learning has been more popular in recent years, owing to the enormous amount of data generated by applications, recent increases in computer power, and the development of improved algorithms[24]. ML was being utilized for a variety of applications at the time, including prediction, image recognition, speech recognition, medical diagnosis, and financial trading. With the introduction of ML, machines or objects are becoming more automated, needing less work and, as a result, requiring less physical control. ML is proven to be more efficient and trustworthy than statistical techniques in several instances [24, 25]. To date, a variety of approaches have been used to complete AI and ML operations, with several inventions and revises. However, a general classification of machine learning algorithms is Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, and Reinforcement Learning.

Deep learning is a type of machine learning that imitates the way humans gain certain types of knowledge. Deep learning is an important element of data science, which includes statistics and predictive modeling. It is extremely beneficial to data scientists who are tasked with collecting, analyzing and interpreting large amounts of data; deep learning makes this process faster and easier[24, 26]. In deep learning, a computer algorithm learns to perform classification tasks directly on complex data in the form of images, text, or sound. These algorithms can accomplish state-of-the-art (SOTA) accuracy, and even sometimes surpassing human-level performance.

DL, as a powerful model, has been proved effective in video transmission and video error concealment tasks. In many video frames recovery method, deep neural networks were used to predict the MV for video error concealment. To extract motion in video transmission, first designed two parallel networks to separately process the horizontal and vertical motion fields of the MV of the previous frame[26, 27]. Then, the combined output of the two networks was used to reconstruct the corrupted portion of the video frames. Both of the aforementioned two methods reconstructed corrupted images by utilizing the information from the adjacent frames of the video. However, reliable adjacent frame information is not available in some situations.

1.2 Statement of the Problem

In WSN, packet losses are expected because of the insufficient bandwidth, shadowing, flat-fading, and noise interference of wireless transmission. The compressed MVV coded sequences are a high chance to transmit errors. The Lossing of one bit appears to destroy full blocks, and this is probably the reason for the loss of sequential blocks. Consequently, certain packet losses in wireless communication reduce the quality of the MVV frame sequences at the decoder side.

In WSN, retransmit errors or lost packets is difficult due to delay, power, and bandwidth constraints of WSN on real-time MVV transmission. Error control methods such as EC and Error Resilience (ER) become effective in improving the MBs in the MVV communication system. Thus, it is necessary to avoid error propagation by recovering the corrupted MBs at the decoder into the utilization of appropriate pre-processing ER techniques and post-processing EC techniques.

Most of these EC algorithms require MVV depth information to recover lost blocks But in WSN good resolution cameras are does not available because of its cost is expensive. In [28] try to develop the first lost frame recovery method for MVV transmission over WSN. This method uses the previous, and succeeding frames based on time and left and right frames based on views to recover the lost frames. This method also uses block-based motion extraction technique for MV and DV extrapolation technique. The other lost frame recovery method [29] for MVV transmission over WSN is by combining the previous method [28] and MPR-algorithm. In this method uses the pixel method to extract the MV and DV which are used for feature extraction technique for MPR. These methods have similar limitations which is the performance or scored PSNR result is not satisfy and not used as input the recovered pixels or blocks for lost frames so why they not well performed.

The other main drawback of the two methods is the lost frame recovery time is high which means they take high time complexity to recover. And no lost frame detection type in any type of video transmission.

In this thesis, we concern on addressing ANN based full-lost frame detection and recovery at the decoder side of MVC. In this method we have develop new lost frame detection in MVV transmission method by using different VQM and IMQ for feature extraction purpose and ANN model for classification purpose. And also we have develop new lost frame recovery method by using new motion and disparity estimation technique for feature extraction purpose and also we have use RNN method to recover the lost blocks in MVV transmission. RNN algorithm is best for this recovery method than other methods because of it used for time-series data so video is time-series data why RNN is highly applicable in video transmission.

1.3 Research Questions

The following are the research questions for lost frame detection and recovery method in MVV transmission over WSN.

- How do lost frames detected in MVV transmission system over WSN ?
- What impacts have lost frame detection method compared with other VQM based on latency complexity ?
- How Lost frames are recovered in MVV transmission system over WSN ?

1.4 Objectives of the Research

1.4.1 General Objective

The general objective of the study is designing Lost frame Detection and Recovery model in a Multiview Video Transmission over Wireless Sensor Network using Neural Network

1.4.2 Specific Objectives

- Study and characterization of different video and image quality metrics to extract the features from MVV sequences to detect lost frames
- To apply ANN classification algorithm to detect lost frames in MVV based on extracted features
- To show the complexity effects of the developed lost detection method from other
- To model the new MV and DV estimation technique for feature extraction and selection purpose in a video sequence to the recovery method
- Applying LSTM method to recover the lost frame in MVV sequences based on MV and DV.
- Analyze the results and compare with other methods by testing different MVVs.

1.5 Methodology

Methodology is a means of achieving the goals of a research problems. Literature on the MVV transmission system, errors in MVV transmission time, different VQM, and frame concealing strategies in MVV can be found in books, journal articles, conference proceedings, and the Internet. In addition, prior and current research efforts on MVV transmission over WSN and various VQMs were evaluated in order to gain a better understanding of the best performing algorithms and methodologies in terms of empirical models and machine learning models. Since this study is meant to be a continuation of prior studies, it will need to be combined with machine learning models and other motion estimation and disparity estimation approaches in order to recover lost frames in MVV transmission via WSN. Therefore, in order to undertake and achieve the objective of this research the following methods and techniques are used.

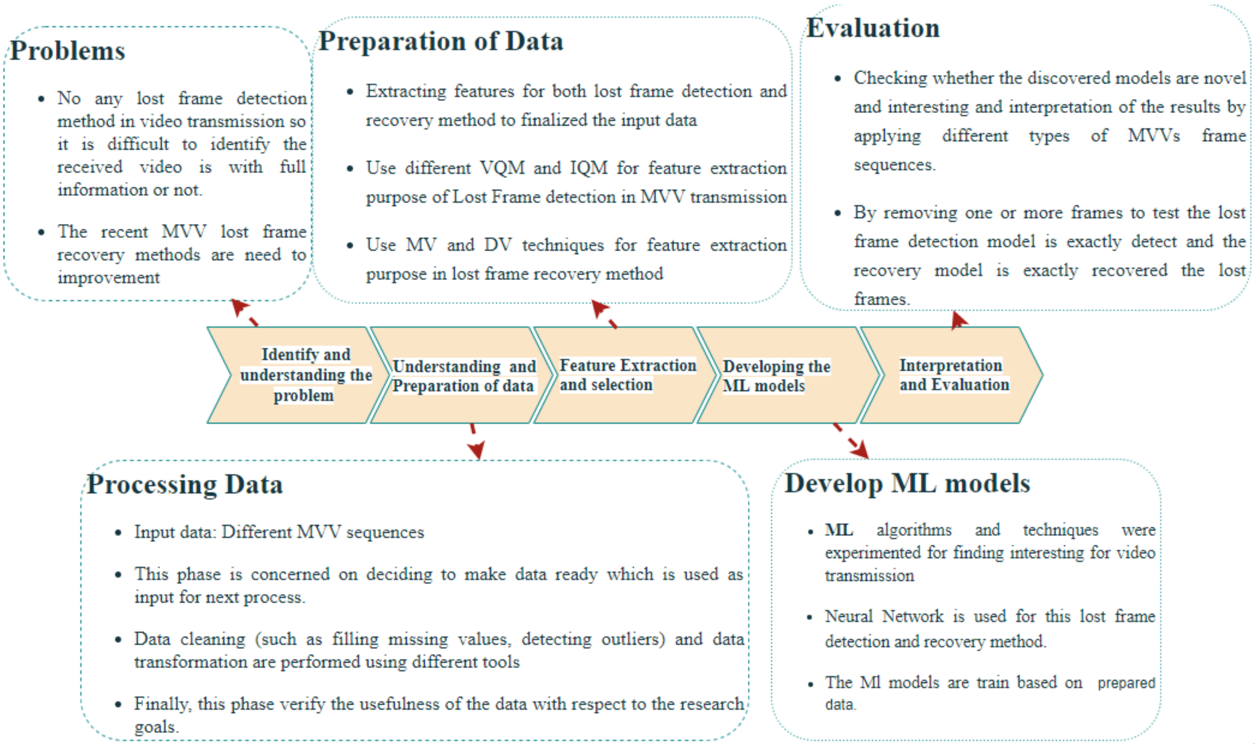
1.5.1 Research design

This study is based on an experimental study that use manipulation and controlled testing to better understand causal mechanisms. This form of study will produce conclusions that can be verified through observation or experiment [30]. Following steps of experimental research are data preparation, preprocessing, feature extraction, feature selection, selection of relevant machine learning algorithms, model building, and evaluation also shown in Figure 1.1.

1.5.2 Data Preparation

In this phase which includes collecting sample data and deciding which data, including format and size, are needed. The data used in this experiment are collected from IRCCyN/IVC [31], the MPEG resource [32], and Nagoya University [33]. These MVVs are available in free from those videos we have use different sequences (Ballet, Breakdance, Lovebird1, Newspaper, BookArrival, LeavingLaptop, DoorFlowers, AltMoabit, Dog, Champagnetower and Pantomime, Kendo, Balloons, Soccer, Akkakiyo, and champian) to test experiments for both detection and recovery methods.

After we collected the data, the subsequent task is extract the features from buffer of MV frames based on VQM and IQM without lost and with lost transmission to


Identify and understanding the problem
Understanding and Preparation of data
Feature Extraction and selection
Developing the ML models
Interpretation and Evaluation

Problems

- No any lost frame detection method in video transmission so it is difficult to identify the received video is with full information or not.
- The recent MVV lost frame recovery methods are need to improvement

Preparation of Data

- Extracting features for both lost frame detection and recovery method to finalized the input data
- Use different VQM and IQM for feature extraction purpose of Lost Frame detection in MVV transmission
- Use MV and DV techniques for feature extraction purpose in lost frame recovery method

Evaluation

- Checking whether the discovered models are novel and interesting and interpretation of the results by applying different types of MVVs frame sequences.
- By removing one or more frames to test the lost frame detection model is exactly detect and the recovery model is exactly recovered the lost frames.

Processing Data

- Input data: Different MVV sequences
- This phase is concerned on deciding to make data ready which is used as input for next process.
- Data cleaning (such as filling missing values, detecting outliers) and data transformation are performed using different tools
- Finally, this phase verify the usefulness of the data with respect to the research goals.

Develop ML models

- ML algorithms and techniques were experimented for finding interesting for video transmission
- Neural Network is used for this lost frame detection and recovery method.
- The Ml models are train based on prepared data.

FIGURE 1.1: Steps conducted in this Research [30]

prepare dataset for lost detection method. In lost recovery method, we develop a new motion estimation and disparity estimation technique to extrapolate MV and DV for feature extraction and selection purpose from 4 estimated frames 2-based on inter-view and other 2-based on intra-view.

1.5.3 Implementation tools selection

The tools used to develop this research are Python 3.6 using Pycharm editor, OpenCV library for video and image processing cases, NumPy and pandas libraries [34] are used for preprocessing and feature extraction purposes in lost frame detection and recovery methods. We use the TensorFlow and Keras libraries [35] to implement the machine learning and deep learning models.

1.5.4 Evaluation and Analysis

The developed model is evaluated for understanding the result, checking whether the discovered model is well detecting the lost frames in MVV transmission and it recover the lost frames in MVV transmission. The performance of the model is tested at different bandwidth to transmit MVV for lost frame detection and different type of MVVs which means fast or slow-motion MVVs [36] for lost frame recovery method. In this thesis, the comparison of the VQM, state of art method, recent (MPR-method) and the proposed system is presented.

1.6 Significance of Thesis

This paper will have many significance on improving of the quality of MVV transmission over WSN. The major contribution of this thesis are summarized as follows:

- To increase the reliability of the MVV transmission over WSN by identifying the loss content in video sequences
- To introduce new lost frame detection method at decoder side of MVC for feature researcher in image and video quality assessment area.
- To introduce the new BMA based MV and DV extrapolation technique of MVV transmission for feature researcher work on different MVC techniques area.
- To enhance the performance of reconstruction of lost frames in MVV transmitted over WSN at decoder side.
- By using this lost frame detection method can design new video and image quality metrics.
- By adding this lost frame recovery method can enhance the video quality in different applications like for video conference, TVs. transmission, and 3D videos.

- By using this developed new motion and disparity estimation techniques the researchers can develop new MVC method for MVV transmission.

1.7 Scope of the Study

From its superficial view, this work is bounded lost frame detection and prediction for MVV transmission over WSN at the decoder side only. For lost frame detection method, we assumed that the reference and the test frames are available at decoder side. So this method needs two consecutive MVV frames to identify lost frames between consecutive sequential video frames also it requires the minimum size of test MVV frames is 100 × 100. And the lost frame recovery method needs minimum three different cameras set at different angles. The over all method is implemented based on gray level video frames.

1.8 Thesis Outline

The rest of the report is organized as follows. Chapter 2 presents an introduction of MVC and different lost frame concealment techniques found in the literature. Different applications for these methods will be discussed and the comparison among these methods is evaluated. Chapter 3 presents the design of the lost frames detection method in MVV. This begun by asses different video and image quality metrics to extract different features from videos. After asses different VQM then applying the ANN classifier to detect the lost frame. Chapter 4 presents the Lost frames Recovery in MVV. This begun extracting MV and DV from MVV frame sequences and fed into LSTM the extracted candidate blocks. The simulation procedure, simulation model, test cases, and simulation results will be presented in Chapter 5. The work will then be concluded and future work or recommendations will be presented in Chapter 6.

Chapter 2

Literature Review

2.1 Conceptual Review

Due to the large amount of data to store and transmit MVV requires much higher bandwidth than traditional 2D videos. Hence, it needs efficient compression techniques to compress this vast amount of data concerning by reducing the redundancies in the temporal domain also by eliminating spatial redundancies. Efficient compression techniques are required to enhance the feasibility of these technologies. Depending on this, MVC is a basic technology to help an MVV transmission method realizing multiview representation with an efficient scheme in WSN. [12, 37, 38].

MVC is an extension of the single view video coding standard H.264/AVC, which is a method used to encode various video sequences that are concurrently obtained from multiple cameras located at different angles and positions [39]. The overall block diagram of MVC is shown in Figure 2.1. The MVC encoder receives temporally and combined video streams to creates one bitstream from every of the N camera scenes [12]. The MVC decoder received the bitstream, decodes, and generates the video signals for every of the N views. Every view can be used to synthesize and show the required types of output, like 3D video, FVV, a hybrid 3-D-FVV or maybe a standalone 2-D output [14].

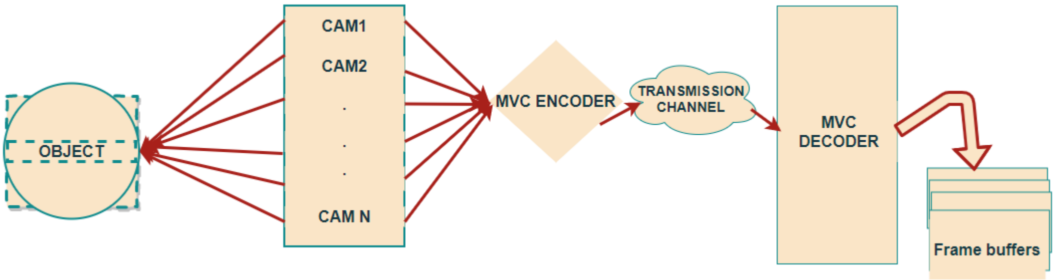


FIGURE 2.1: The MVC structure [12]

As the MVV camera captures analogous superclass 3D exhibitions, there appear essential correlations between video frames. The similarities are classified into two types inter-view similarities among adjacent camera scenes and temporal correlations between sequent temporal frames of every MVV sequence. This analysis agrees with the natural arrangement of MVV frames toward a Matrix of Pictures (MOP). Each column contains spatially adjacent views obtained at an equivalent period moment, and all row contains a temporally consecutive video frame of one image. Every view sequence is organized toward the row of its MOP, which differs from linear camera arrays in this case. The idea is to distinguish between inter-view correlation and temporal relationship only [40].

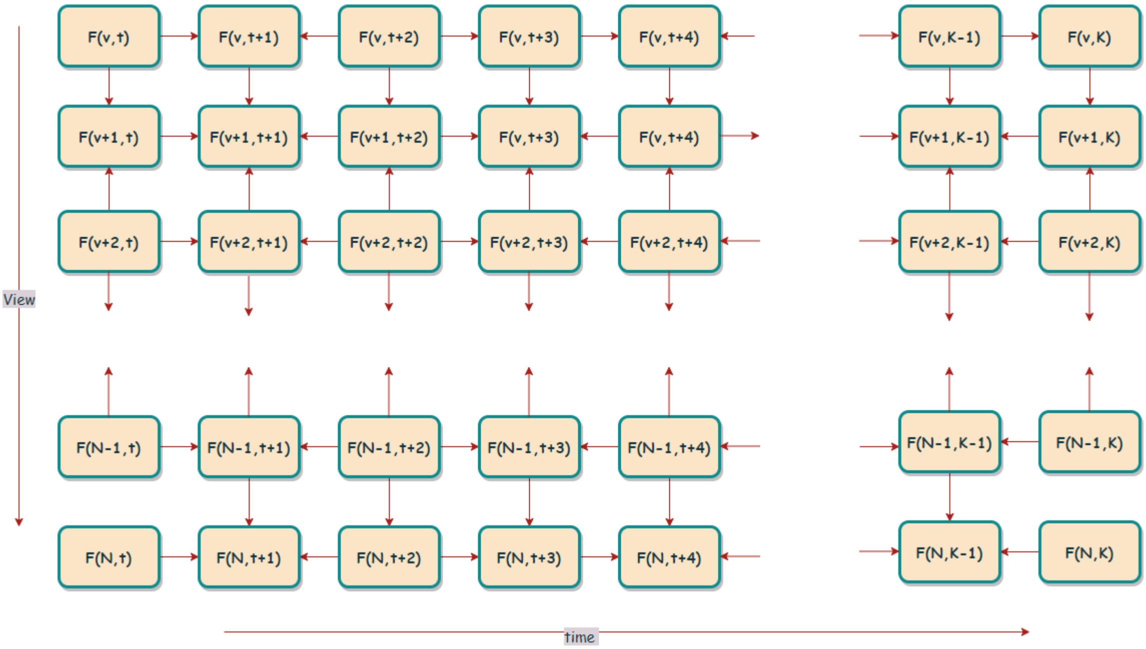


FIGURE 2.2: MOP for v=N image sequences, each of which consists of t=K temporally consecutive images [40].

Figure 2.2 shows an image matrix for v = N set of frames, every made up of $t = K$ temporally consecutive frames. t =K temporally sequential images form a temporal Group of Pictures (GOP), and $v = N$ views form a Group of Views (GOV). For instance, the primary view sequence pictures are represented by $v + n, v + n, ..$,N with $n = 0, 1, 2, ...$,N . MOPs with N×K pictures to analyze the compression performance of coding systems that process N×K pictures together. Joint compression has a goal to use all relationships between those pictures. There are many effects on compression performance of MVC depending on MOP size (N, K) [40, 41]. The coding way of the conventional techniques also provides the convenience of the dynamic size prediction method of H.264 to utilize the redundancies within succeeding frames in the spatial and temporal domains. The prediction technique in MVC consists of proper variable size motion estimation and the added disparity estimation way [42].

Problems arising during transmission, such as signal loss or interference, must be dealt with by the wireless communication system (bit insertion, deletion, or inversion). The major issue in MVV communication via wireless networks is to provide end-users with a satisfactory quality of experience (QoE)[43]. However, due to bandwidth limitations and the occurrence of channel defects, the wireless channel remains a difficult challenge. These issues, which include bandwidth fluctuation and transmission faults, are briefly discussed. One of the most essential elements in multimedia communications is bandwidth restriction. Bandwidth limitation is one of the most important factors in multimedia communications [44].

Video streaming using wireless networks with time varying conditions and resource limitations. However, unreliability, bandwidth fluctuations and high bit error rates of wireless channels can cause severe degradation to video quality. Packet loss and transmission error are significant obstacles to compressed video transmission across networks, resulting in packet losses owing to congestion. Because of these factors, information is lost on the decoder side of the MVC. As a result, some blocks in single frames are missed during transmission; as a result of these circumstances, lost frames occur during MVV transmission through WSN. Because of the losses, the receiver may not be able to receive all of the compressed video data, causing the visual quality to decrease. Traditionally, error control methods on the encoder and decoder sides of MVC have been used to solve this problem [44–46].

The Automatic Repeat Request (ARQ) and Forward Error Correction (FEC) methods are the most frequently utilized error control strategies in video data transmission. The primary disadvantage of the ARQ method is its significant latency, which makes it unsuitable for real-time applications. As a result, owing of its dependability in real-time applications, the usage of FEC has been frequently recommended. FEC, or forward error correction, is a kind of error control that use channel codes to mitigate the impact of channel faults in a wireless network. The encoder uses these FEC codes to secure the bit stream before transmitting it to the decoder, and the FEC codes are subsequently used to rectify errors in the bit stream once it is received. FEC methods are effective in mitigating random bit errors, but they underperform against longer-duration bursts. Furthermore, even when the channel is error-free, the FEC method incurs continual transmission cost. As a result, the video's coding efficiency may be harmed and lost [45–47]. As a result of these FEC reasons, researchers are attempting to enhance the performance of video transmission by using error hiding techniques at the MVC decoder side. A decoder should be able to cope with transmission faults in addition to design and implementation requirements, and one method a decoder can handle losses caused by transmission mistakes is through error hiding techniques.

In MVC, error hiding is a non-standard feature. A fast post-processing approach ensures error control in the decoder without adding to the bitrate or adding to the latency. A MVC error concealment decoder should be able to identify and hide transmission faults by interpolating missing MBs from successfully received nearby intra or inter MBs to reduce visual impairment in a frame. There are two types of error concealment techniques at the decoder side of MVC pixel- based and block based error concealment techniques. In pixel-based method, a motion vector is assigned to each pixel of the image, resulting in a dense motion field. It has the advantage to provide a precise description of the motion. However, from a video coding viewpoint, it entails a costly representation resulting in a large overhead for motion information [47, 48]. In this method, recover the missed frame by locating pixel by pixel so it performed high result than other methods. Nevertheless, all pixel based recovery methods takes high time latency than other methods. In block-based method, lost frames are divided into different blocks and assign different MVs for those lost blocks. Different researchers are proposed different methods for block-based method to recover the full or partially lost frame sequences in MVV.

2.1.1 ANN

The architecture of the ANN is focused on the characteristics of biological neurons. A normal brain can understand new information and respond to new and evolving conditions. The brain has the remarkable capacity to interpret incomplete and contradictory evidence and make its own decisions.

The brain consists of cells known as neurons. These cells (neurons) are interconnected to form the neural network or the brain. There are about 10 billion neurons and 60 trillion interactions between them [49].

A biological neuron is consists of cell bodies, axons, and dendrites, as seen in Figure 2.3. Other neurons send electrical pulses to the dendrite, which are then received by the cell body. The nuclei and other chemical structures used to provide the cell are found in the cell body, which is known as Soma. The axon transmits the signal from one neuron to the next. Synapse is a link between the dendrites of two neurons or a neuron and muscle cells. ANN is a replica of a normal neural

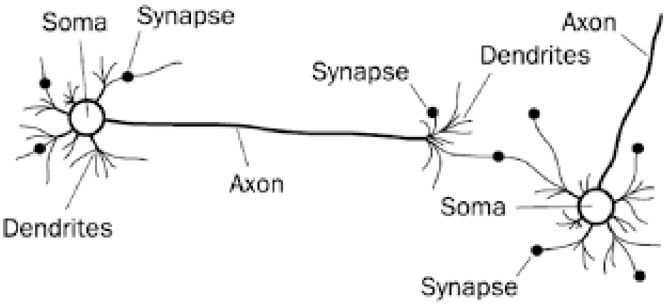


Figure 2.3: Biological neural network

network, in which artificial neurons are linked similarly to a brain network. Table 2.1 illustrates the biological and ANN comparison.

A neural networks fundamental processing element is a neuron. The main tasks associated with processing units are to acquire input from neighbors providing incoming activation, compute output, and sending output to their neighbors . Since several processing units can carry out their computations at the same time, such a system is essentially parallel. neural network has three processing units: IThe input processing units of a neural network receive input from external sources, calculate their activation level, compute their output as a function of their activation level, and transfer this information to the rest of the network. Output processing units are calculated and broadcast their output to external receivers until receiving information from the rest of the network or feed their output back to the network input layer for further processing. Hidden processing units that only acquire information from other processing units in the network and transmit their generated output to them [49].

Table 2.1: Similarities between biological neural networks and artificial neural networks

Bilogical Neural Network	ANN
Soma	Neuron
Dendrite	Input
Axon	Output
Synapse	Weight

In most instances, a neuron receives several inputs at the similar period. The amplitude of the input signal as registered by the artificial neuron is determined by the relative weight of each input. They are a measurement of the input's relation power. These abilities may be changed in response to different training sets, according to the topology of a network, or by learning.

The network inputs are mixed within the neuron using the net function. The weighted linear combinations is presented in 2.4. Weights are used to store the amount of data regarding the input that is needed to solve problems. Before being added to the summing block, each signal is compounded by a related weight $w_1, w_2, w_3, ..., w_n$. In addition, the artificial neuron has a bias term w_0, a threshold value T that must be met or expanded for the artificial neuron to generate a signal, a linear or nonlinear function f that operates on the generated signal net and an output y after this function are also included in the artificial neuron. In Figure 2.4, the bias neuron's input is considered to be 1.

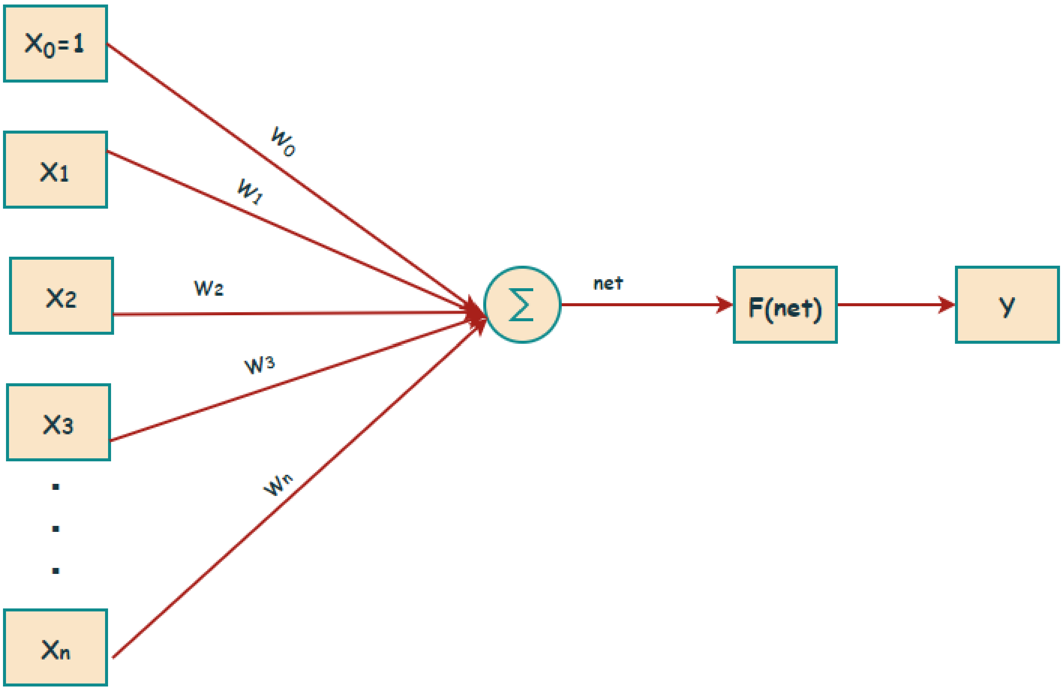


FIGURE 2.4: Basic neural model

The following relation describes the transfer function of the basic neuron model:

$$net = w_0 + \sum_{k=1}^{n} w_k x_k \tag{2.1}$$

$$y = f(net) \tag{2.2}$$

The linear or nonlinear activation function ensures that the neuron's response is constrained. Nonlinear functions are often used, based on the paradigm and algorithm used for training the network, to accomplish the benefits of multilayer nets relative to the restricted capacities of single-layer networks [50].

Table 2.2: Activation functions

Activation Functions	Formula
Linear	f(x)=x
Logistic (Sigmoid)	f(x)=$\frac{1}{1+e^{-x}}$
Hyperbolic Tangent	f(x)=$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
Sinousoidal	Weight

For each layer, you can define one specific activation function. There is no appropriate rule theoretically for defining the activation function of the various layers. Despite the fact that certain experiments use separate functions for the individual layers, others use the same mechanism for the input, hidden, and output layers. Table 2.2 summarizes the most widely used activation functions. Linear or straight-line functions are constrained because the output is proportional to the input. Linear functions aren't very useful. This was the issue in the early models of the network.

The logistic and symmetric logistic functions get the property of varying in the ranges$[0, 1]$ and $[-1, 1]$, respectively. The symmetric function grasps the dynamic characteristics of certain issues in a quite detailed measure, particularly in the input and hidden layers. The majority of observational evidence suggests that this feature is used in the hidden layer, although there is no clear theoretical justification for it [50].

The hyperbolic tangent function provides for reliable network adaptation throughout the hidden layers (particularly in three-layer networks), especially when the observer has selected a logistic or linear function as the output function.

At the asymptotes, a sigmoid or S-shaped curve reaches a minimum and maximum point. When the curve extends from 0 to 1, it's called a sigmoid, and when it ranges from -1 to 1, it's called a hyperbolic tangent. This function is particularly beneficial for using back-propagation algorithms trained in neural networks. Because it is easy to distinguish, and this can minimize the computation capacity for training.

The sinusoidal function is often used in analysis, and it is recommended that input and output be normalized in the range $[-1, 1]$.

2.1.1.1 ANN Architectures

The organization of neurons into layers and the patterns of connections within and between layers are called net architecture. A typical neural network is constructed by layers. A single-layered network has an input layer and an output layer of neurons. A multi-layer network contains hidden neurons in addition to one or more hidden layers. The ability of the network to derive higher-order statistics from input data is improved by adding more hidden neurons.

The signal flow, and the connection between neurons, will define the net type. Hence, ANNs have two architecture types, feed-forward networks, and Recurrent or Feed-Back Networks.

Figure 2.5, shows feed-forward networks, the signal flow is from the input to the output units, from the input nodes, through the hidden nodes (if any) and to the output nodes. The network is without cycles or loops. Three types of FeedForward networks are single layer perceptron, multi-layer perceptron (MLP), and radial basis function nets [51].

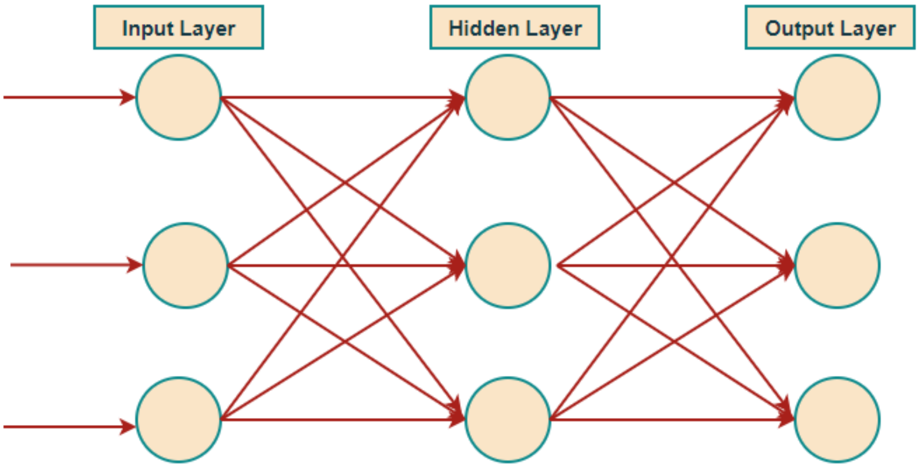


FIGURE 2.5: MLP

2.1.1.2 Training an ANN

A good set of weights is required to approximate a given target function. The problem is that altering one weight will possibly alter the function's output over the entire input space, so it's not as straightforward as using a grid-based approximation. One possible solution is to minimize an error function which measures just how bad an approximation is. The process of finding weights that minimize the error function is called training or learning by artificial intelligence researchers. Learning or training methods can be categorized as:

Supervised Training: With supervised learning, the ANN must be trained before it becomes useful. Training consists of providing data about input and output to the network. We also refer to this data as the training set. That is, the corresponding desired output set is also given for each set of inputs provided to the system [49].

Unsupervised Training: in this type of algorithm, the network is equipped among inputs in unsupervised training but it couldn't require outputs. The system will then have to determine which features it would use to group the data it receives. This is also known as adaptation or self-organization.

2.1.1.3 Training Algorithms

In ANNs, the Training algorithm refers to the process of altering the weights of connections between the nodes of a given network to find the error by comparing the network's output value with the target value and then reducing the difference (error) by altering the weights. For MLP, there are a number of training algorithms. Due to its popularity in terms of both simplicity and applicability, the backpropagation algorithm is among the most common methods for network training. [51].

The algorithm is divide into two sections: the training and recall phases. Each weights of the network being initialized randomly during the training phase. After that, the network's contribution is evaluated and compared to the target value. The output layer weights are adjusted depending on the network's training error. Likewise, the network loss is propagated backward as well as required to modify the previous layers' weights.

The error values are produced and transmitted to modify the network's weights, as seen in Figure 2.6. Just the feed-forward calculations are used to distribute weights from the training process in the recall phase. In the training and recall stage, the feed-forward algorithm has been applied [49, 52].

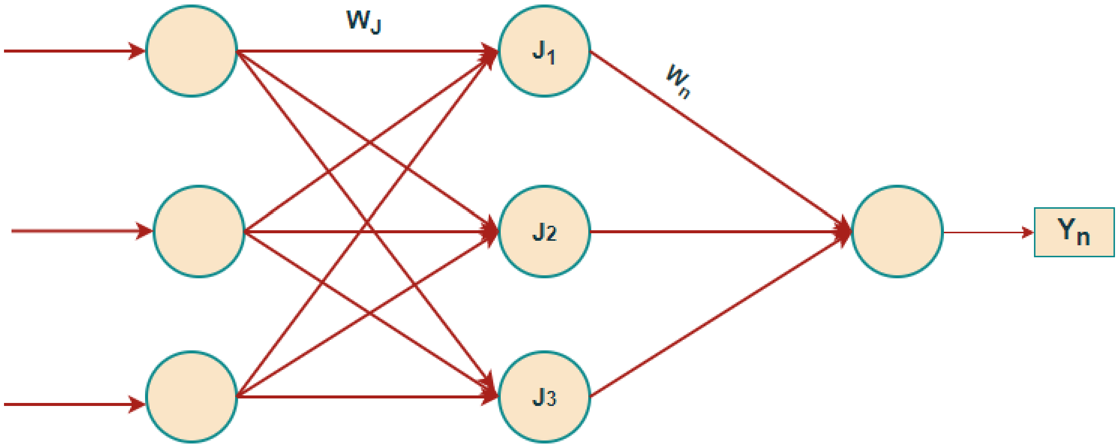


FIGURE 2.6: Back propagation algorithm

When the error value is below the designer's minimal fixed value, the training process will be completed. The Backpropagation algorithm has one disadvantage: the training process takes a long time [52].

During the recall phase, the network with the final weights decided during the training period will be used. As a result, for each input pattern in this phase, the output should be determined also using linear or nonlinear activation functions. An important advantage of this method is that in the recall step, it gives a very fast network [52].

2.1.2 RNN Algorithm

Another form of a neural network used for sequential labeled activities is the RNN. The cyclic connections that each unit can have distinguishes this form of neural network from others, in addition to those of any ANN. Its preference for using it in sequence labeling comes from its ability to connect the previous information with the present task. This gives the RNN memory-like capabilities, making it a good example for things like sequenced applications. Not similar to a basic MLP, which simply creates inputs into outputs, an RNN will potentially use all previous inputs for each output, creating an internal state separate from the network parameters [53].

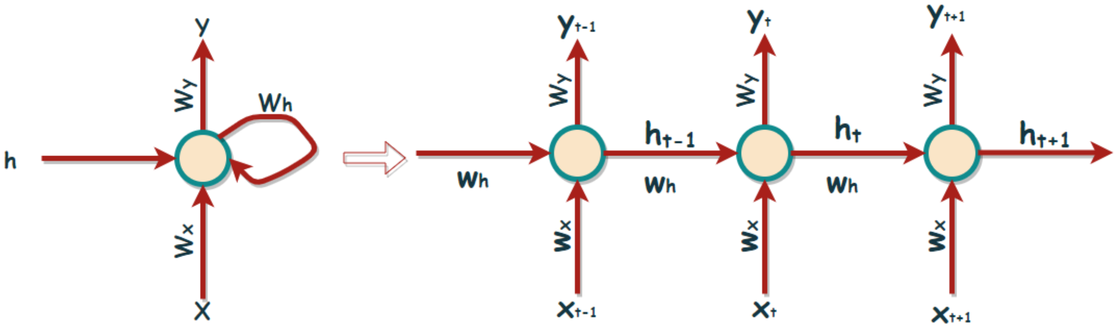


FIGURE 2.7: An RNN, and the computations involved in its forward calculation occurring in time [53].

We must first compact a series of input symbols $X = (x_1, x_2, ..., x_n))$ to a fixed-dimensional vector through recursion until we can construct an RNN. We can handle variable-length inputs and outputs by applying the vector. Suppose we have a vector h_{t-1} at step t that represents the past of all previous symbols before the current one. The RNN will generate a new vector, or internal state, h_t, that compresses all previous symbols $(x_1, x_2, ..., x_{t-1})$ and also the new symbol x_t by:

$$h_t = f(W_h h_{t-1} + W_x x_t + b) \tag{2.3}$$

$$y_t = f(w_y h_t) \tag{2.4}$$

where,The input weight matrix is W_x , the recurrent weight matrix is W_h, the hidden layer weight is W_y, and the bias vector is b.

An RNN is close to a conventional Neural Network in terms of training time [54]. This back-propagation technique is usually included, although with some variations. Since certain timing steps within the network share these variables, the gradient on every output is calculated by both the present and succeeding time steps' calculations. For example, to find the gradient at t = 4, we must back-propagate three steps and add the gradients. This technique does define as Backpropagation Through Time (BPTT).The tendency of the derivatives of the activation functions ($tanh(x), softmax(x)$) to converge is one of the challenges in the training phase [54, 55]. RNNs of this form Vanishing gradients are a common issue, particularly in networks that are supposed to model something. The use of Gated Recurrent Unit (GRU) or LSTM architectures [56] is a more recent common approach to the issue of the vanishing gradient. GRUs bag has two gates: reset and update, but LSTM's bag has three gates: input, output, and forget. Because there are fewer gates in GRU than in LSTM, it is less complicated. However, if the dataset is small, GRU is preferable; for bigger datasets, LSTM is preferred[57].

2.1.2.1 LSTM

As previously stated, the vanishing gradient effect limits conventional RNNs through acquiring long-term dependence. The LSTM network is an RNN architecture that can learn long-term dependencies. Hochreiter and Schmidhube [56] implied the first to present LSTM, and several other works followed in the years that followed. It is currently one of the most commonly used Deep Learning frameworks for sequential data processing. As Figure 2.8 shows the data flow of LSTM , In the present neuron, They have three inputs: preceding instant output, present instant input, and past instant cell state [57].

As shown in Figure 2.9, the top horizontal line in the repeating module of LSTM is its key part which is called a cell state. The cell state acts as a conveyor belt and continues through the entire chain with small linear interactions. Information can be let through or denied from passing through the cell state of LSTMs by using gates made from a sigmoid network layer and a point-wise multiplication operation. Gates carefully regulate information to either let through the cell state or reject it. The sigmoid layer of gates gives an output having a value of either zero or one, where zero prohibits the flow of information through the cell state and one lets every information to pass through the cell state [56].

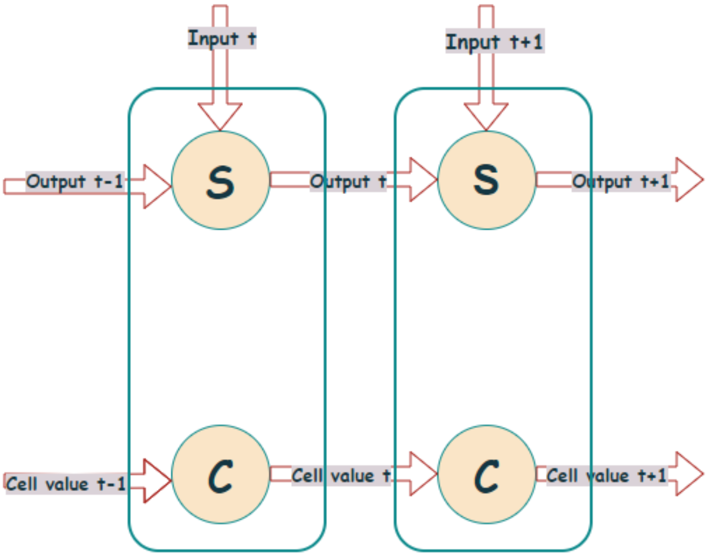


FIGURE 2.8: Data transmission flow of LSTM [57]

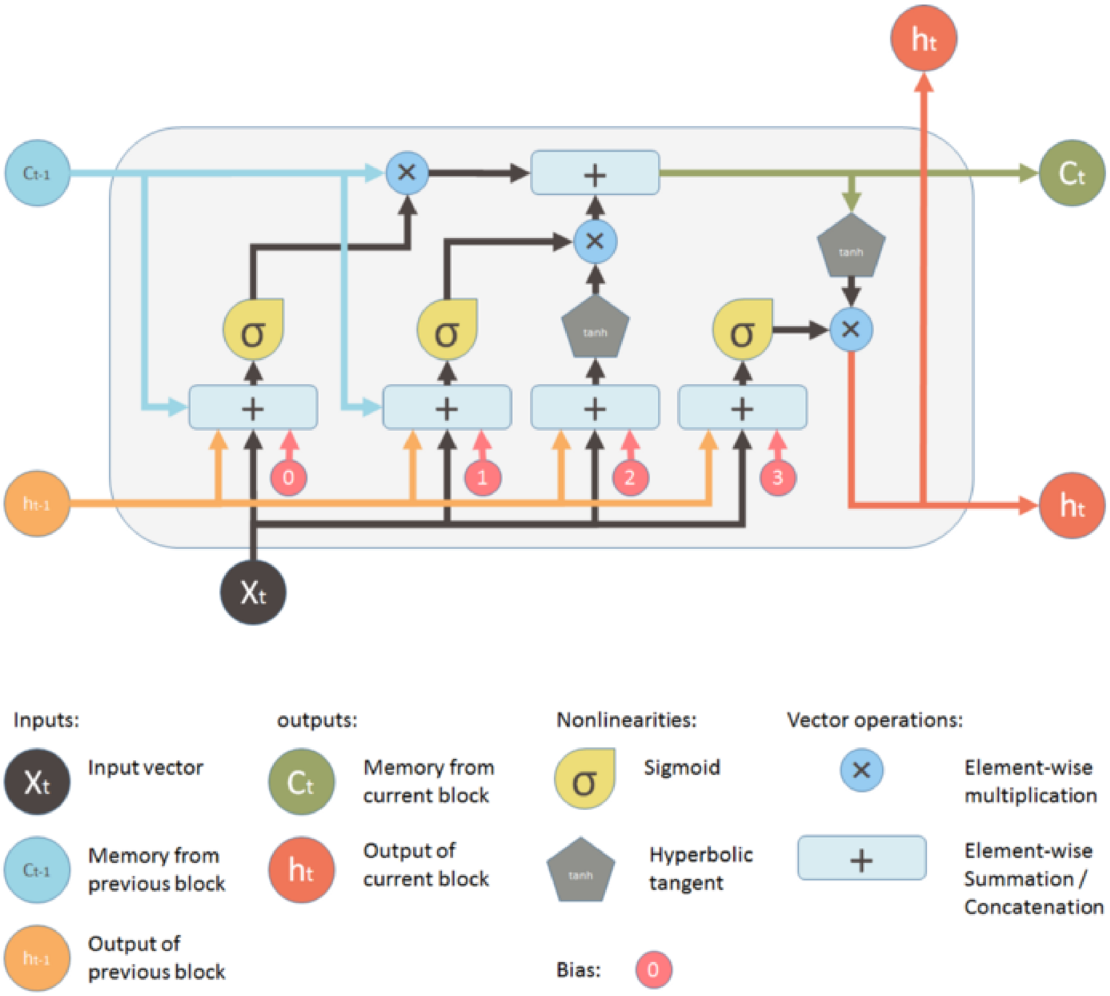


FIGURE 2.9: The LSTM neural network's architecture [57].

To measure the internal states, an LSTM network is made up of memory cells and gate units. The memory cells are organized by three gates: input gate I, forget gate f, and output gate O. These gates conceptually decide how much of a given vector can be considered at a particular time phase t. The input gate, for example, specifies how much of the present input would pass, the forget input specifies how much of the previous state should pass, and the output gate specifies how much of the internal state must be affected in subsequent time stages and layers. The gates and cell update and output are defined as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1}) \tag{2.5}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1}) \tag{2.6}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1}) \tag{2.7}$$

$$\overline{c_t} = tanh(W_{xc}x_t + W_{hc}h_{t-1}) \tag{2.8}$$

$$c_t = f_t \otimes c_{t-1} \oplus i_t \otimes \overline{c_t} \tag{2.9}$$

$$h_t = o_t \otimes tanh(c_t) \tag{2.10}$$

The multiplexer and summation operations are denoted by $\otimes$ and $\oplus$, respectively. The network parameters are expressed by W matrices. The LSTM networks have been strongly educated and those gates are good at dealing with exploding/vanishing gradients. The hyperbolic tangent *tanh* and the sigmoid $\sigma()$ are non-linearities, and h_t has been represent the hidden condition [56].

2.2 Related Works

In recent years, several literatures have proposed many EC techniques of MVV at decoder side of MVC .

Depth based Boundary Matching Algorithm (DBMA) [58] is a lost frame concealment technique that combines depth information with the Block Matching

Algorithm (BMA) to retrieve a corrupted Motion Vector (MVs) for 3D video transmission. Although the depth map represents the depth content of each pixel in a two-dimensional video[59], This recovery method was analyzed in simulation with different experiment in different papers. The results showed that the DBMA is very suitable and useful algorithm. After employing this method the decode 3D video quality can be highly enhanced after broadcasting over noisy channels. However, this method did not fully recover the lost frames that mean the PSNR of this method was low but compared to the previous methods it is good. Further more, DBMA method needs high resolution video frames to get the depth in formation [60, 61].

In work [62, 63], also proposed one method of EC at the decoder side of MVC. Investigate the depth of video frames to determine the lost MVs for corrupted macroblocks in 3D video transmission. Then use the impacts of the extracted depth map to recover precisely the lost motion vector for the corrupted blocks. This paper shows the recovery technique is very effective than other previous papers [58–61]. By testing simulation and experiments. The experiment was tested by implementing the algorithm based on H.264 video coding standard [64]. Furthermore, this paper emphasizes getting good performance than previous algorithms by changing the bandwidth of the transmission channel (5%-10%) packet losses. However, the recovery method of these papers is not enough because of its low PSNR value. Additionally, these methods did not address the effect of the algorithm on the low-resolution sensor and at low-bandwidth transmission time.

All the papers [65–68], have discussed frame concealment algorithms for a stereoscopic video sequence. Temporal and spatial concealment techniques depend on video analysis for stereoscopic video proposed on [65, 66]. In [67] discussed two concealment techniques, in the pixel and the block domain methods. Both techniques use the MVs and DVs of neighboring frames to recreate the macroblock of a missing frame. In all above papers concerning only in color intensities value of the pixel location in video frame signals. As a result, they are unsuitable for concealing depth frames in video plus depth (V+D) sequences [68], that have less texture and appear to have more immediate temporal variation.

In [69, 70], the Efficient Frame Concealment for Depth Image-Based 3-D Video transmission has been presented. These articles discuss a new lost frame recovery algorithm for 3-D video transmission, that is based on the conventional MV extrapolation method (MVE) for 2-D video. The depth map represents the depth value for each pixel in the 2-D video. This method also describes some of the relationships between the depth map and the two-dimensional recording. To recover the missing frame, the depth-based algorithm makes extensive use of relationships. They shows the result by using an experiment to simulate by applying the JM 15.0 H.264/AVC codec [64]. The experiment reveals that incorporating the Depth Image-Based 3-D Concealment algorithm improves the efficiency of existing methods dramatically. Still, this method also needs depth video frames to conceal errors. When the transmission channel is at low bandwidth and the cameras are low-resolution, this method does not improve performance.

All the above works required the depth information of the multi-view video to accomplish the algorithm. But in WSN good resolution cameras are does not available because of its cost is expensive. Then the above all articles did not improve the performance of the MVV [28, 71].

In [28], the authors proposed the first frame loss concealment method for MVV through WSN. Since it solves the issues of low-resolution sensors means without the necessity of depth information.The DV extrapolation process between the neighbor views of the missed frame (Inter-View) and the MV extraction technique between neighbor views of the lost frame are used in this method to retrieve missed frames (Intra-View). The EC method of this paper was analyzed by using experiments and simulation. The experiment used H.264/MVC reference software (JMVC 8.0) to implement the proposed algorithm [64]. The results of this approach show that it enhances performance at low bandwidth transmission when compared to previous methods. In this method, also the scored result is not satisfying the PSNR value compared to the original frame with its recovered frame signals.

MPR-based frame recovery technique in WSN is introduced on paper [29]. This method tries to improve the performance by applying the MPR algorithm to work

[28]. This reconstruction algorithm uses a pixel-based method which is to recover lost frames pixel by pixel. It generates candidate pixels for each pixel locations in a lost frame from inter-view frames (f(v-1,t) and f(v+1,t)) and intra-view frames (f(v,t-1) and f(v,t+1)). Select the candidate pixels by using MV and DV which is discussed in the previous work [28]. Then, each candidate frames prepare one generated pixel for each one-pixel position in the lost frame and reduce the size of candidate pixels to two from those four candidate pixels. After that, fed into the 2-input MPR model to estimate the pixel of a single-pixel location in the lost frame. This method achieves a good PSNR value during comparing the original frame signals to recovered frame signals than the other lost frame recovery algorithms. Still this MPR-method not efficient to recover lost frames and because it recovers the lost frames pixel by pixel, it has a higher computational complexity than other methods.

In this thesis, ANN-based lost frame detection, and RNN-based lost frame recovery method is introduced in MVV transmission over WMSN on the decoder side of MVC. And, MV and DV extrapolation methods are used for feature extraction purposes in the RNN recovery method. To enhance the performance of the frame recovery method by using correlation idea in the same frame implies use previous recovered blocks as input for the RNN [72, 73] model.

Chapter 3

Lost Frame Detection in MVV

The overall model of frame lost detection and recovery is shown in Figure 3.1. The algorithm is designed for the decoder side of MVC. In MVC the decoder receives a buffer of frames (f(v-1,t), f(v+1,t), f(v,t-1), f(v,t+1) and f(v, t)) based on different view and time sequence. Before reconstruction of the views in MVC the frames are passed through on algorithm. It has two main blocks: lost detection, it detects there is a lost frame between

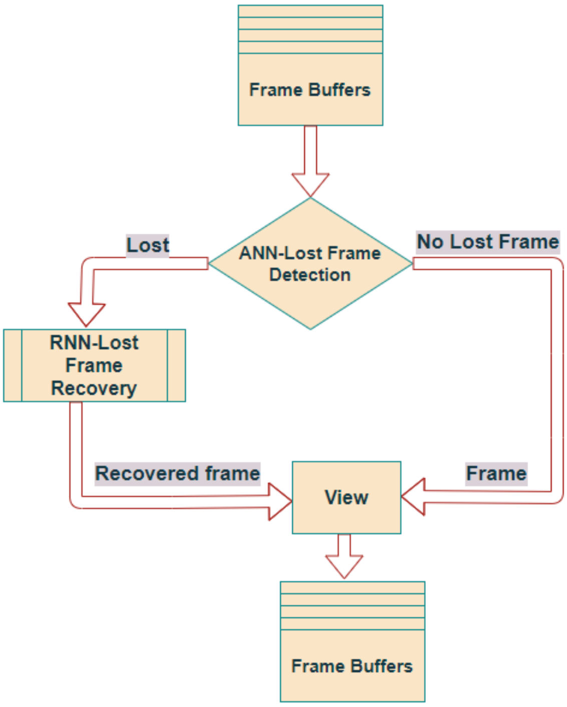


Figure 3.1: Flow chart of overall proposed model

a sequence of frames are compared with reference frame F(v,t-1). If there is frame lost between two frames then first block sent information to RNN based lost frame recovery block and then reconstruct the lost frames in this block.

3.1 Lost Frame Detection in MVC

The block diagram of the proposed ANN-based lost frame detection method is shown in Figure 3.2. This approach takes two consecutive frames as input, $F(v-1,t)$ and $F(v,t)$, and extracts features using various image and VQMs.

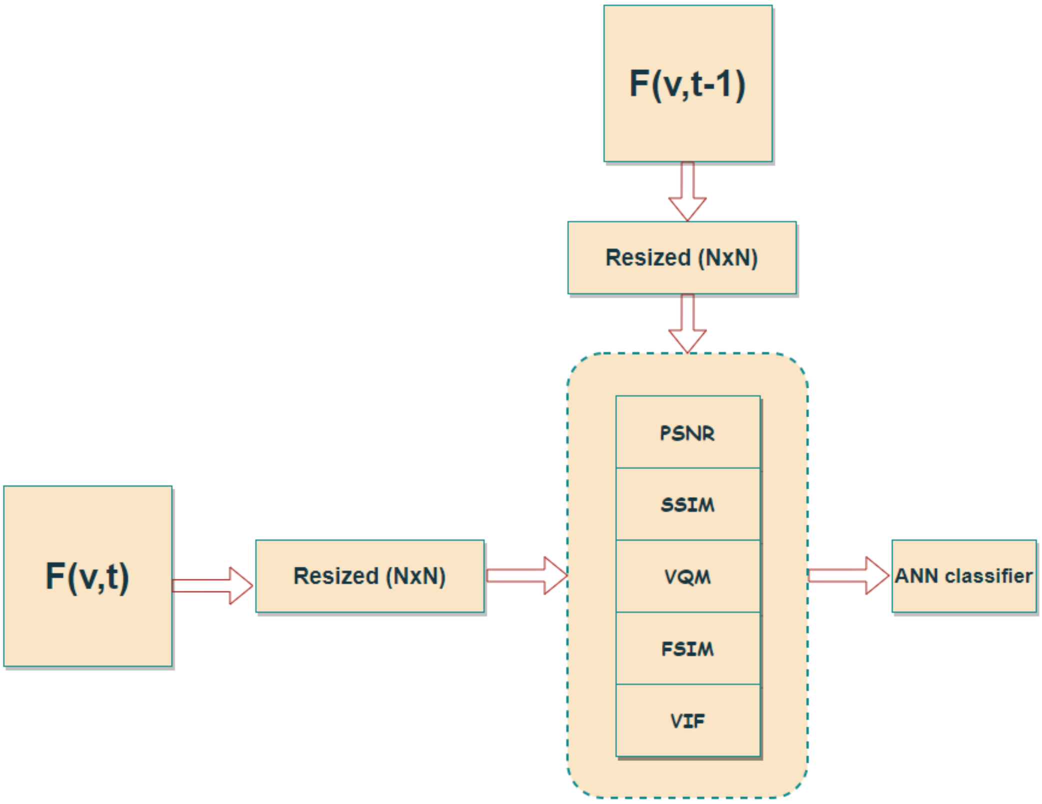


Figure 3.2: Lost frame detection in MVC

Then the extracted metrics between reference and test frame values are fed into ANN classifier. The ANN is used to detect the lost and normal frames from transmitted videos.

3.1.1 Feature Selection

This task aims to observe and choose the most useful features in a dataset. The features with low importance are eliminated. In this method, We evaluated different videos using five objective measurements which are **PSNR**, **SSIM**, **VQM**, **FSIM**, and **VIF**. These type VQMs detects distorted video frames in video compression and transmission time. The selection criteria for video metrics in this detection method are dependent on various features of FR objective video metrics [74], such as structural similarity, error sensitivity, Spatio-temporal, and information fidelity approaches. Then select one VQM from each approaches, to generate different features from transmitted videos.

3.1.2 PSNR

PSNR is expressed in a logarithmic decibel (dB) scale which is a proportion between two digital signals. By using this metric can measure the quality between the reference image and the corrupted image. The video or image signals are affected by noise after transmission in the wireless channel, compression, or, processing [74, 75]. Because several signals become extremely dynamic ranges (the proportion among these signals value highly changes the quality of images or video frames). Because of the non-linear response of the HVS, the video data, as well as the codec model, do not improve, the PSNR does not attain accurate perfection in the calculation of similarities of differences in the perception values. Nevertheless, it is a reliable quality assessment because it may be an indication of the accuracy of a video signal in lossy environments.

Video frame signal F(v,t) transmit to certain transmission channel and it compared to a reference video frame signal F(v,t-1). The row and column dimensions for both frames are N and M in this case. PSNR is described as the proportion of signal power to noise power as shown in:

$$PSNR(v,t) = 10log_2(\frac{(f_m^2)}{MSE(v,t)} \tag{3.1}$$

Where fm is maximum possible pixel value of the video frame, MSE stands for mean square error and is calculated as follows:

$$MSE(v,t) = \frac{1}{NxM}\sum_{j=0}^{M}\sum_{i=0}^{N}(F_{v,t}(i,j) - F_{v,t-1}(i,j))^2 \tag{3.2}$$

where $F_{v,t}(i,j)$ is the pixel value at location (i, j) of the source frame, and $F_{v,t-1}(i,j)$ is the pixel value in the reference frame at location (i, j) and f_m is maximum pixel value.

3.1.3 SSIM

SSIM is computed by using the presumption of the HVS, which use to obtain some structural data from the field of view. Hence, the difference of structural information between anchor (corrupted) and reference video frame signals in MVV could be an important approach to identify the distortions in video transmission. In [76] essential parts of SSIM are presented, where structural information is combined by comparing the basic HVS factors like luminance, contrast, and structure. Let F(v,t-1) and F(v,t) are two consecutive frame signals of at dimension (N,M). First, by applying a function of mean intensities $\mu_{f(v,t-1)}$ and $\mu_{f(v,t)}$ to extract the luminance difference of both frame:

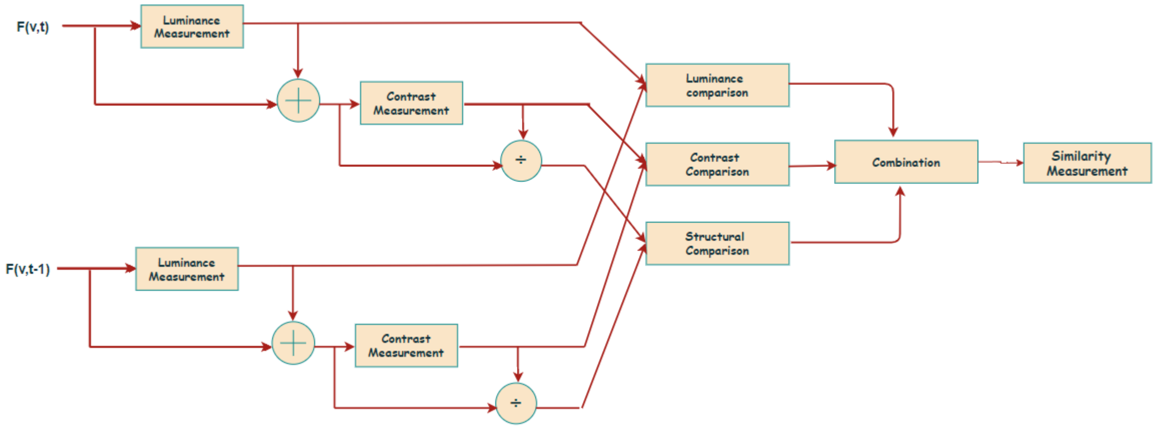


FIGURE 3.3: Diagram of SSIM measurement system [76]

$$l(f(v,t-1),f(v,t)) = \frac{\mu_{f(t-1,v)}\mu_{f(t,v)} + C_1}{\mu^2_{f(t-1,v)} + \mu^2_{f(t,v)} + C_1} \tag{3.3}$$

where C_1 denotes the constant involved to eliminate an uncertainty when $\mu^2_{f(v,t-1)}+\mu^2_{f(v,t)}$ is approximately zero. $C_1 = (K_1L)^2$, where L implies the video frames' dynamic scale and $K_1 << 1$. In a similar manner for the constants under contrast and structure functions are done. By using a function of standard deviations $\sigma_{f(v,t-1)}$ and $\sigma_{f(v,t)}$ to calculate the contrast difference of the video frames f(v,t-1) and f(v,t).

$$c(f(v,t-1),f(v,t)) = \frac{2\sigma_{f(v,t-1)}\sigma_{f(v,t)} + C_2}{\sigma^2_{f(v,t-1)} + \sigma^2_{f(v,t)} + C_2} \tag{3.4}$$

where C_2 is $(K_1L)^2$ and $K_2 << 1$. And also, the amount of the contrast change computed as $\Delta\sigma = \sigma_{f(v,t-1)} - \sigma_{f(v,t)}$, At high base contrast, this computation is less sensitive than at low base contrast, which leads to HVS's contrast masking behavior. Finally, to compute the correlation based on the structure comparison between the two MVV frame in the form:

$$s(f(v,t-1),f(v,t)) = \frac{\sigma_{f(v,t-1)f(v,t)} + C_3}{\sigma_{f(v,t-1)}\sigma_{f(v,t)} + C_3} \tag{3.5}$$

where C_3 is also a constant number it less than 1. Remark, that correlation coefficient among original video frame is the same as between candidate video frame signals (i.e ($f(v,t) - \mu_{f(v,t)}/\sigma_{f(v,t)}$)) and consequently could describe the structure well. By combining the above representations we can get the structural similarity in the form of:

$$SSIM(f(v,t),f(v,t-1)) = [l(f(v,t-1),f(v,t))]^{\alpha}[c(f(v,t-1),f(v,t))]^{\beta}[s(f(v,t-1),f(v,t))]^{\gamma} \tag{3.6}$$

where $\alpha, \beta, \gamma > 0$ variables that are used to adjust the relevant values of individual elements. For simplification in [12] they state $\alpha = \beta = \gamma = 1$ and $C3 = C_2/2$, which described the measure to

$$SSIM(f(v,t), f(v,t-1)) = \frac{(2\mu_{f(v,t)}\mu_{f(v,t-1)} + C1)(2\sigma_{f(v,t-1)f(v,t)} + C2)}{(\mu^2_{f(v,t)} + \mu^2_{f(v,t-1)} + C1)(\sigma^2_{f(v,t)} + \sigma^2_{f(v,t-1)} + C2)} \tag{3.7}$$

When using SSIM, it is better to do so locally. it involves to determine local statistics $\mu_{f(v,t)}$, $\sigma_{f(v,t)}$ and $\sigma_{f(v,t-1)f(v,t)}$ in minimum window size which is pixel-by-pixel moved around the whole video frame pixels and then averaging the results. The explanations for this approach are that different areas of video frames can vary significantly, and humans can only focus on a small area at a time. This technique can also be used to generate a spatially changing resolution map of the video frame to learn more about the distortion. Not only in the area of image processing but also in other fields, SSIM is one of the most widely applied measures. For instance, SSIM has been used in the prize freeware H.264codec x.264, as well as voice recognition, compression algorithms, and other applications. Even though SSIM outperforms MSE, also has its drawbacks. For instance, even though the content of the translated, scaled, or rotated images is the same as the reference images, the simple variant does not work well.

3.1.4 VQM

This video assessment technique is a Discrete Cosine Transform (DCT) based metrics [77]. The simple human perceptual visual field model will be applied to this metric. Both video and image metrics are built on the foundation of human spatial-temporal contrast sensitivity. This Model analyzes distortion of video frames in four steps show in Figure 3.4. The first step is find DCT for both reference and test frames that helps to separate the frame into parts (or spectral sub-bands) of differing importance (with respect to the frames signals visual quality). A frame signal is transferred from the spatial domain to the frequency domain using the DCT. For both frames the model performs DCT in equation (3.8) for R x R pixels blocks $B_i^{v,t}(x,y)$ of the reference frame F(v,t) and for blocks $B_i^{v,t-1}(x,y)$ of

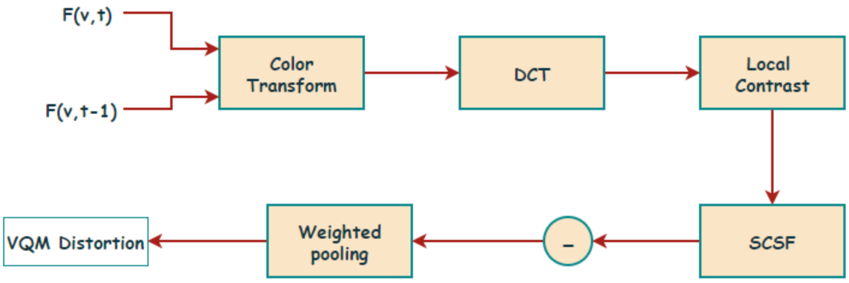


FIGURE 3.4: Overview of VQM [77]

the source frame F(v,t-1).

$$D(u,v) = \frac{1}{\sqrt{2R}} Z(u)Z(v) \sum_{x=0}^{R-1} \sum_{y=0}^{R-1} p(x,y) cos\left[\frac{(2x+1)u\pi}{2R}\right] cos\left[\frac{(2y+1)v\pi}{2R}\right] \tag{3.8}$$

$$Z(m) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if m=0} \\ 1 & m > 0 \end{cases} \tag{3.9}$$

where p(x,y) is the pixel values in single blocks of source and test frames and D(u,v) is the DCT value for both frames. Based on equation (3.8) we get DCT values for $B_i^{v,t}(x,y)$ and $B_i^{v,t-1}(x,y)$.

$$\begin{aligned} DCT_{B_i^{v,t}}(u,v) &= D(B_i^{v,t}(x,y)) \\ DCT_{B_i^{v,t-1}}(u,v) &= D(B_i^{v,t-1}(x,y)) \end{aligned} \tag{3.10}$$

The data is then converted to local contrast (LC). The proportion of DCT magnitude to DC magnitude for the corresponding block is defined as LC. The following formula is used to measure LC:

$$\begin{aligned} LC_i^{v,t}(u,v) &= \frac{DCT_{B_i^{v,t}}(u,v).(\frac{DC_i^{v,t}}{1024})^{0.65}}{DC_i^{v,t}} \\ LC_i^{v,t-1}(u,v) &= \frac{DCT_{B_i^{v,t-1}}(u,v).(\frac{DC_i^{v,t-1}}{1024})^{0.65}}{DC_i^{v,t-1}} \end{aligned} \tag{3.11}$$

Each block's DC portion is referred to as DC. The average DCT value for an 8-bit video frame is 1024. For suitable psychophysics data, 0.65 is the best parameter.

$$\begin{aligned} DC_i^{v,t} &= DCT_{B_i^{v,t}}(0,0) \\ DC_i^{v,t-1} &= DCT_{B_i^{v,t-1}}(0,0) \end{aligned} \tag{3.12}$$

By dividing each DCT coefficient according to its belong visual threshold, the effects are transformed to JND for all blocks $JND_i^{v,t}(u,v)$ and $JND_i^{v,t-1}(u,v)$. This is how the spatial component of the spatial contrast sensitivity function(SCSF) is implemented. We are using one SCSF matrix for static frames and another matrix for dynamic frames rather than using temporal filtering and individual SCSF independently. The computing and memory burden are minimized as a result of this. The MPEG standard quantization matrix in equation 3.13 could be use as the inverse of the static SCSF matrix (The contrast threshold matrix, that is the inverse of the contrast sensitivity matrix, is quantization matrix).

$$MAX = \begin{bmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 27 & 29 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{bmatrix} \tag{3.13}$$

$$\begin{aligned} JND_i^{v,t}(u,v) &= LC_i^{v,t}(u,v) * MAX^{-1} \\ JND_i^{v,t-1}(u,v) &= LC_i^{v,t-1}(u,v) * MAX^{-1} \end{aligned} \tag{3.14}$$

To account for SCSF's temporal features, every component in the static SCSF matrix is magnified by a power in the dynamic matrix. The frame rate of video sequences determines the power output. After that, the two sequences are subtracted.The result of the difference of video frame sequence is subjected to a Contrast Masking (CM) process, which is also dependent on the reference video sequence.

$$\begin{aligned} mean_{dist} &= \tfrac{1}{RxR} \textstyle\sum_{x=0}^{R-1} \sum_{x=0}^{R-1} |JND_i^{v,t}(u,v) - JND_i^{v,t-1}(u,v)| \\ max_{dist} &= max(|JND_i^{v,t}(u,v) - JND_i^{v,t-1}(u,v)|) \end{aligned} \tag{3.15}$$

Finally, the mask variations can be merged in a number of ways to represent perceptual error across several scales, and the combined error can be translated to Visual Quality (VQ). It is possible that a wide distortion in one area can reduce sensitivity to other a smaller distortion. In this case ,weighted maximum distortion into merged distortion becomes more better than combined distortion alone [77] . Then VQM can be determined as follows:

$$VQM = 1000mean_{dist} - 5max_{dist} \tag{3.16}$$

Numerous primitive psychophysics experiments can be used to determine the highest distortion weight parameter.

3.1.5 FSIM

Feature Similarity Index (FSIM) for FR video and image metrics is proposed by IQA it is based on HVS understands a video frames in the main consistent with its low-level options rather than high-level options. Two major features in FSIM that can be detail viewed in depth are Gradient Magnitude (GM) & Phase Congruency (PC). In FSIM, the primary function is the PC, which is a non-dimensional measurement, which used to show the impact of a local structure. Since PC is contrast variational and contrast information influences HVS' perception of image quality, the GM is used as a secondary function in FSIM. In defining image quality at the local level, PC and GM play significant role [78, 79].

PC: plays an important role in FSIM algorithm to extract main features of video frames.The odd-symmetric and even-symmetric wavelets at scale m, respectively, are denoted by I_m^e and I_m^o and they form a quadrature pair [78]. The convolution results of the input video frames $F^{(v,t)}$ with quadrature pairs of filters at position $F^{(v,t)}(x,y)$ on scale m will form a response vector that is the basic components $(e_m^{(v,t)}(x,y),\, O_m^{(v,t)}(x,y))$ to calculate PC.

$$\begin{aligned} e_m^{(v,t)}(x,y) &= F^{(v,t)}(x,y) * I_m^e \\ o_m^{(v,t)}(x,y) &= F^{(v,t)}(x,y) * I_m^o \end{aligned} \tag{3.17}$$

To find the PC of 2D grayscale images, first we applay 2D log-gabor filter in reference and source video frames. Due to the bandwidth restriction of a single Log-Gabor filter, 2D Log-Gabor filters can be applied to create a full filter bank in both the horizontal and radial directions. The 2D log-Gabor transition formula is as follows:

$$G_2(\omega,\theta_j) = exp\left(-\frac{(log(\omega/\omega_0))^2}{2\sigma_r^2}\right).exp\left(-\frac{(\theta-\theta_j)^2}{2\sigma_\theta^2}\right) \tag{3.18}$$

where σ_r regulates the filter's bandwidth and ω_0 is the filter's core frequency. θ_j is the log-Gabor filter's orientation angle, which is expressed in $\theta_j = J\pi/j$, $j = 0, 1, ..., J-1$, J has been the number of orientations, σ_θ which defines the filter's angular bandwidth. The PC of video frame $F^{(v,t)}(x,y)$ can be defined as the summation over θ_j and scale m.

$$PC^{(v,t)}(x,y) = \frac{\sum_j E_{\theta_j}^{(v,t)}(x,y)}{\epsilon + \sum_m \sum_j A_{m,\theta_j}^{(v,t)}(x,y)} \tag{3.19}$$

where $A_{m,\theta_j}^{(v,t)}(x,y) = \sqrt{e_{m,\theta_j}^{(v,t)}(x,y)^2 + o_{m,\theta_j}^{(v,t)}(x,y)^2}$ is the local amplitude on scale m and orientation θ_j, and $E_{\theta_j}^{(v,t)}(x,y) = \sqrt{F_{\theta_j}^{(v,t)}(x,y)^2 + H_{\theta_j}^{(v,t)}(x,y)^2}$ is local energy along orientation θ_j, where $F_{\theta_j}^{(v,t)}(x,y) = \sum_m e_{m,\theta_j}^{(v,t)}(x,y)$ and $H_{\theta_j}^{(v,t)}(x,y) = \sum_m o_{m,\theta_j}^{(v,t)}(x,y)$ and ϵ is a small positive constant number. $PC^{(v,t)}(x,y)$ is a real number within $(0,1]$ [80].

GM: To express gradient operators, different forms of convolution masks could be used [81]. Some of the most commonly identified gradient operators are Prewitt operators, Sobel operators, and Scharr operators. In comparison to Prewitt and Sobel operators, the Scharr operator could achieves better result. The partial derivatives $g_x^{(v,t)}(x,y)$ and $g_y^{(v,t)}(x,y)$ of the video frame $F^{(v,t)}(x,y)$ along horizontal and vertical direction using scharr gradient operator. The GM of $F^{(v,t)}(x,y)$ is then defined as

$$g^{(v,t)}(x,y) = \sqrt{(g_x^{(v,t)}(x,y))^2 + (g_y^{(v,t)}(x,y))^2} \tag{3.20}$$

After extracted the PC and GM feature maps of source and test video frames, we can compute the FSIM index for IQA. $PC^{(v,t)}(x,y)$ and $PC^{(v,t-1)}(x,y)$ are denotes the PC map extracted from source video frame($F^{(v,t-1)}(x,y)$) and test video frame($F^{(v,t)}(x,y)$) and $g^{(v,t)}(x,y)$ and $g^{(v,t-1)}(x,y)$ are the GM maps extracted from both frames. The next step is compute the GM similarity measure and PC similarity measure of test and source video frames[79]. The similarity measure for $PC^{(v,t)}(x,y)$ and $PC^{(v,t-1)}(x,y)$ is defined as:

$$S_{PC(x,y)} = \frac{2PC^{(v,t)}(x,y).PC^{(v,t-1)}(x,y) + T_1}{PC^{(v,t)}(x,y)^2 + PC^{(v,t-1)}(x,y)^2 + T_1} \tag{3.21}$$

and the similarity measure for $g^{(v,t)}(x,y)$ and $g^{(v,t-1)}(x,y)$ is defined as:

$$S_{g(x,y)} = \frac{2g^{(v,t)}(x,y).g^{(v,t-1)}(x,y) + T_2}{g^{(v,t)}(x,y)^2 + g^{(v,t-1)}(x,y)^2 + T_2} \tag{3.22}$$

T_1 and T_2 are positive numbers that increase $S_{PC(x,y)}$ stability and GM dynamic range, respectively. After computing the local similarity feature maps [79]. Select maximum PC value from both frames $\{PC_{max}(x,y) = max(PC^{(v,t)}(x,y), PC^{(v,t-1)}(x,y))\}$. Since the maximum PC value has a major effect on HVS when comparing the similarities of Test and Source video frames. Generally the FSIM index between $F^{(v,t)}(x,y))$ and $F^{(v,t-1)}(x,y))$ is defined as :

$$FSIM = \frac{\sum_x^r \sum_y^c S_{g(x,y)}.S_{PC(x,y)}.PC_{max}(x,y)}{\sum_x^r \sum_y^c PC_{max}(x,y)} \tag{3.23}$$

Where r and c are the dimensions of a video frames.

3.1.6 VIF

The information-theoretical criteria for frame fidelity estimation are called VIF, and it is a full-reference VQM. The VIF technique measures the data that can be derived from the brain in the source frame, as well as the omission of such information due to distortion, by using Natural Scene Statistics (NSS), HVS, and a video frame distortion process, as seen in Figure 3.5.

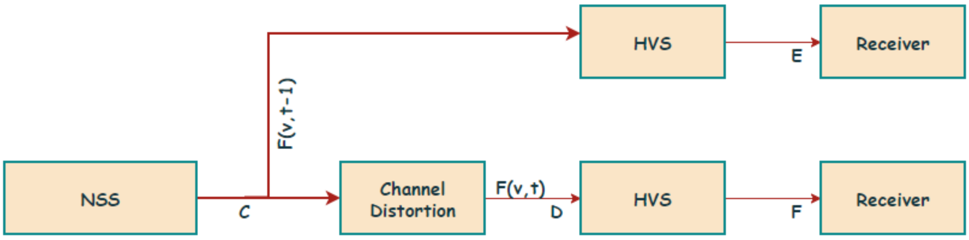


FIGURE 3.5: Diagram of mutual information [82]

The VIF is measured by considering two local similarity levels: mutual data among the HVS channel's input and output when there is no distortion channel present (known as source video frame) and mutual data among the distortion channel's input and output for the test video frame [82].

In the wavelet domain, the NSS model is modeled as the Gaussian Scale Mixtures (GSM) method. One subband of the wavelet decomposition of an image can be considered as a GSM random field (RF), $C = \{C_i : i\epsilon I\}$ where I stands for the RF's set of spatial indices. C is an RF that can be defined as the product of two independent RFs,

$$C = S.U = \{S_i.\vec{U_i} : i\epsilon I\} \tag{3.24}$$

where $U = \{\vec{U_i} : i\epsilon I\}$ is a Gaussian vector RF containing mean zero and co-variance C_u and $S = \{S_i : i\epsilon I\}$ is an RF with positive scalars, The vectors U_i and C_i have M dimensions. Model block I as the vector $\vec{C_i}$, which divides the subband coefficients into non-overlapping blocks of M coefficients each.

The main function of a distortion channel is to describe a generic distortion operator that distracts the statistics of video frames. The distortion model provides essential features while being both mathematically controllable and computationally easy. The signal attenuation as well as additive noise paradigm exist in the wavelet domain [74]. The model of the output of the distortion channel looks like:

$$D = gC + V = \{g_i\vec{C}_i + \vec{V}_i : i\epsilon I\} \tag{3.25}$$

where C is the RF from one of the source video frame signal's subbands ,$D = \{\vec{D}_i : i\epsilon I\}$ in the test video frame signal is the RF as from corresponding subband, $V = \{\vec{V}_i : i\epsilon I\}$ is a Gaussian noise RF with variance that is stationary additive zero-mean $C_v{\sigma_v}^2$ and $g = \{\vec{g}_i : i\epsilon I\}$ is a deterministic scalar gain field. The RF V is white and is unaffected by S or U. Slightly adjusting the field G to constrain it. On distorted frame forms, this model captures blur and additive noise.

In the VIF framework, the HVS model's intentions to measure the variance that perhaps the HVS introduces into the signal which passes through it.Inside the wavelet domain, it has been a zero-mean, stationary, and, additive white Gaussian noise construct. Thus, the HVS noise model is in the wavelet domain as stationary RFs $H = \{\vec{H}_i : i\epsilon I\}$ and $H^{'} = \{\vec{H'}_i : i\epsilon I\}$,$\vec{H}$ and $\vec{H}^{'}$ are zero-mean uncorrelated multivariate Gaussian with the same dimensionality as C_i. That is we use for reference frame and test frame respectively.

$$E = C + H \tag{3.26}$$

$$F = D + H^{'} \tag{3.27}$$

E and F denote the visual signal in the display including an HVS schema of the source and also test video frames in one subband, sequentially. U, S, and V are considered to not affect the RFs H and $H^{'}$. The covariance of H and $H^{'}$ as:

$$C_H = C_{H'} = \sigma_H^2 I \tag{3.28}$$

where σ_H^2 is a component of the HVS models.

The VIF criterial model could be derived from previous models (NSS, Distortion and HVS) for both frames. $I(\vec{C}, \vec{E}|s)$ and $I(\vec{C}, \vec{F}|s)$ is The mutual information of both reference and test frames. where s stands for the realization of S for particular reference frames. Let $\vec{C}^N = \{\vec{C}_1, \vec{C}_2, \vec{C}_3, .., \vec{C}_N\}$ is denotes N block vectors from C. For reference frame the mutual information is computing by using chain rule to get equation 3.29 .

$$I(\vec{C}^N; \vec{E}^N|\vec{S}^N) = \sum_{j}^{N}\sum_{i}^{N} I(\vec{C}_i; \vec{E}_j|\vec{C}^{i-1}, \vec{E}^{j-1}, s^N) \tag{3.29}$$

because of conditional independence of C and H given s, simplify to equation (3.30 and 3.31).

$$= \sum_{i}^{N} I(\vec{C}_i; \vec{E}_i|\vec{C}^{i-1}, \vec{E}^{i-1}, s^N) \tag{3.30}$$

$$= \sum_{i}^{N} I(\vec{C}_i; \vec{E}_i|s_i) \tag{3.31}$$

The relationship between mutual information I(X,Y) and entropy h(X,Y) is [83] :

$$I(X, Y) = h(Y) - h(Y|X) \tag{3.32}$$

Based on equation (3.32) drive equation (3.33)

$$= \sum_{i}^{N} h(\vec{C}_i + \vec{H}_i|s_i) - h(\vec{H}_i|s_i) \tag{3.33}$$

depending above equations finally mutual information for both reference equation (3.34) and test equation (3.35) frames are:

$$I(\vec{C}^N; \vec{E}^N|\vec{S}^N) = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{M} log_2(1 + \frac{s^2{}_n\lambda_k}{\sigma^2{}_H}) \tag{3.34}$$

$$I(\vec{C}^N; \vec{F}^N|\vec{S}^N) = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{M} log_2(1 + \frac{g^2{}_n s^2{}_n\lambda_k}{\sigma^2{}_V + \sigma^2{}_H}) \tag{3.35}$$

where λ_k are the eigenvalues of covariance (C_u). By using extracted information of reference and test frames have been measure the final VIF results from each subband. So the final drived formula of VIF is :

$$VIF = \sum_{j\epsilon subbands} \frac{I(\vec{C}^{N,j}; \vec{E}^{N,j} | \vec{s}^{N,j})}{I(\vec{C}^{N,j}; \vec{F}^{N,j} | \vec{s}^{N,j})} \tag{3.36}$$

where $\vec{C}^{N,j}$ represents N elements of RF C_j for j−th subband. The VIF is figured for a set of N x M wavelet coefficients that would represent either a complete subband of video frames or images, or a spatially localized region of subband coefficients. For the values of VIF belong to the interval [0, 1], where VIF = 1 if and only if the test frames is the copy of the reference frame and VIF = 0 completely all pixel value was lost because of the distortion [82].

3.1.7 Lost Detection Using ANN

In this thesis, ANN model is used to identifying there is a lost frame between in two consecutive video frames. To train a neural network we have used PSNR, FSIM, SISM, VQM and VIF metrics as inputs. As already mentioned in the previous topics, all the PSNR, FSIM, SISM, VQM and VIF fed into the neural network with respect to the corresponding values at lost frames occurred and in normal condition. And the proposed ADNN architecture is shown in Figure 3.6. It is composed of five layers, which are an input layer, 3 hidden layer, and an output layer.

Following the training of the neural network model, the next critical step is to validate the trained neural network. When feeding new data to the model, testing the ANN is critical to ensure that the trained network generalizes well and produces the optimal outputs. As a result, we Test our proposed model using three different techniques.

Plotting the predicted neural network outputs against the expected target values is one such technique. The slope of this line gives one an understanding of the training process when analyzing such a manner. The slope should ideally be one.

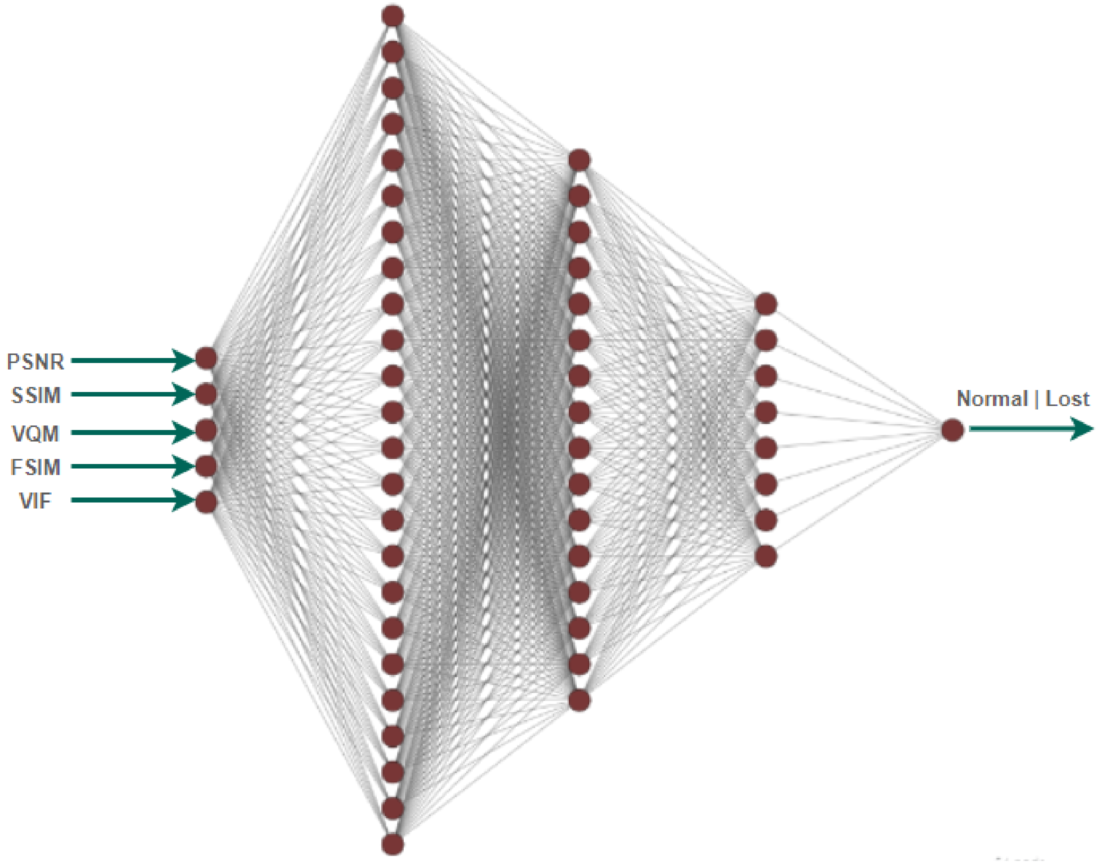


Figure 3.6: Proposed ANN Architecture.

Further to that, the outputs and targets correlation coefficient (m) measures how closely the ANN's outputs track the expected targets [51].

Drawing the confusion matrix and looking at the exact number of instances which have been classified positively by the neural network is another technique used to test the neural network. This number should ideally be 100, indicating that there was no ambiguity in the classification process. As a result, if the confusion matrix demonstrates very low positive classification scores, the neural network is expected to perform poorly [84].

The final and most apparent method of evaluating the neural network models is to send a whole new sample of data with known inputs and desired output and measure the classification percentage error in the prediction. The neural network has passed the test if the average percentage error in its performance is acceptable [85].

3.1.8 Datasets

The structure of the prepared dataset with sample data for lost frame detection is shown in Table 3.1. The first column indicates the MVV sequence type, the other 5 consecutive (2-6) columns are represent the inputs of ADNN and the last column is describe the expected output of ADNN. PSNR, SSIM, VIF, VQM, and FSIM are computed for Normal type frames using perfectly consecutive frames and for Lost type frames by removing one or more frames from successive frames.

Table 3.1: The structure of the prepared dataset of lost frame detection

Sequence	**SSIM**	**PSNR**	**VQM**	**VIF**	**FSIM**	**TYPE**
Break_Dance	0.985323985	34.96531393	1.020892043	0.803580597	0.855886363	Normal
Break_Dance	0.961481676	30.77012883	1.208237654	0.70883388	0.79545566	Lost
Break_Dance	0.949210271	29.38709502	1.486114105	0.699169203	0.784706316	Lost
Break_Dance	0.987572051	35.79621878	0.928708583	0.844684124	0.866675092	Normal
Akka_Kiyo	0.941879676	24.17375942	1.174796467	0.556863459	0.730315149	Normal
Akka_Kiyo	0.828073996	20.84370536	1.481660154	0.414304786	0.625926313	Lost
Champ_Tow	0.865567948	20.25323157	2.733695207	0.201023165	0.600472525	Lost
Champ_Tow	0.999965587	55.19962576	0.705373452	0.983202208	0.936798426	Normal
Book_Arrival	0.999847882	45.86260161	0.228102714	0.954325952	0.947303441	Normal
Book_Arrival	0.991221833	28.24983025	0.078707602	0.717020929	0.893399079	Lost

Table 3.2 shows the total prepared sample data from different MVV for training and testing purpose to detect a lost frame in MVV sequences.

Table 3.2: Dataset description of lost frame detection

Item	Descriptions
Total prepared data	21,000 records
Normal type data	10,500 records
Lost type data	10,500 records

Chapter 4

Lost frames Recovery in MVV

A block diagram of LSTM-based lost frame recovery for MVC is shown in Figure 4.1. The recovery algorithm is designed at decoder side of MVC. In this method take frame buffers as input such as (f(v,t-1) and f(v,t+1)) selected frames based on time and (f(v-1,t) and f(v+1,t)) selected frames based on view when assumed f(v,t) as a lost frame. After that, select candidate blocks by projecting block positions in lost frames to extract overlapping blocks from candidate frames.

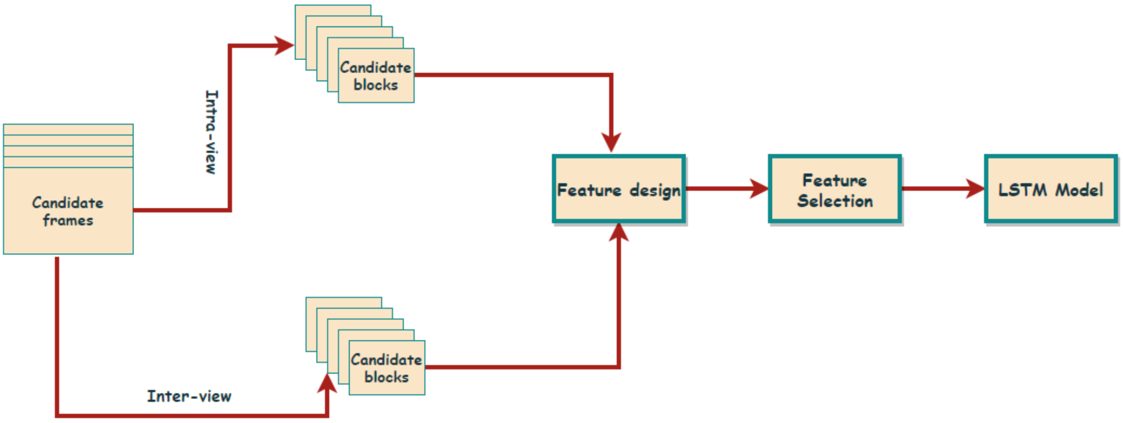


FIGURE 4.1: Block diagram of proposed lost frame recovery method

From candidate blocks we extract, design, and select features for the input of the LSTM model by choosing the best matched blocks using motion estimation and disparity estimation concepts.

4.1 Motion Estimation

Motion estimation is one of the most important computational methods in video processing. It is the process of determining MVs, which describe the transformation from one frame to other neighboring frames in video sequences. In many video applications, motion estimation has become a significant concern. It's a crucial element of video compression, as it helps to reduce bit rates and increase coding performance. Motion prediction is a method for improving video compression efficiency and reducing video processing time.

There are two motion estimation types these are forward and backward motion estimation. In backward motion estimation, the reference frame is the previous frame and the candidate frame is considered as the current frame which is used to compute MVs [86]. The inverse of backward motion estimation is forward motion estimation. Forward MVs are performed on a frame that occurs after the candidate frame in forward motion estimation. Backward motion estimation is influenced by forward motion estimation. There are two motion estimation techniques: pixel-based and block-based. Pixel-based Motion estimation techniques determine each pixels displacement independently and do not need motion information to be transmitted; instead, they recursively use luminance variations to find motion information. The optical flow method is a method for estimating MVs for each pixel in video frames. Since the threshold value varies from pixel to pixel, this method of motion measurement has the downside of being based on

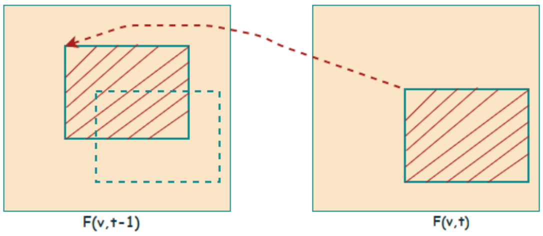


FIGURE 4.2: Forward motion estimation

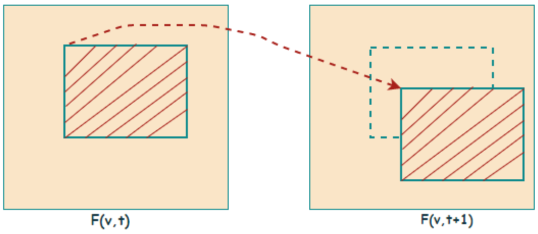


FIGURE 4.3: Backward motion estimation

the threshold. Block-based motion estimation is a faster and more efficient solution. Block-based motion prediction is the method of finding the right fit blocks by going block by block. [87].

4.1.1 Block Matching Motion Estimation

In this method, the MV is computed between the candidate and the source frame frames. Each candidate frame is divided into non-overlap macroblocks pieces. To determine an MV, these macroblocks are matched in the source frame with a corresponding block and its neighboring blocks. The best appropriate block matched between the candidate frame and also referenced frame matching criteria is used to compute the MV. For all macroblocks in a frame, the MV provides a motions estimate for the current frame. Motion is calculated in the present frame using the MV for all macroblocks that make up a frame [87]. The MV refers to the distance between the current candidate block and the best match block within a search window in the reference frame. The displacement in horizontal and vertical directions is represented by the MV. Specifically, around the x- and y-axes. In the block matching process, the actual video frame, $F(v,t)$ is first segmented into

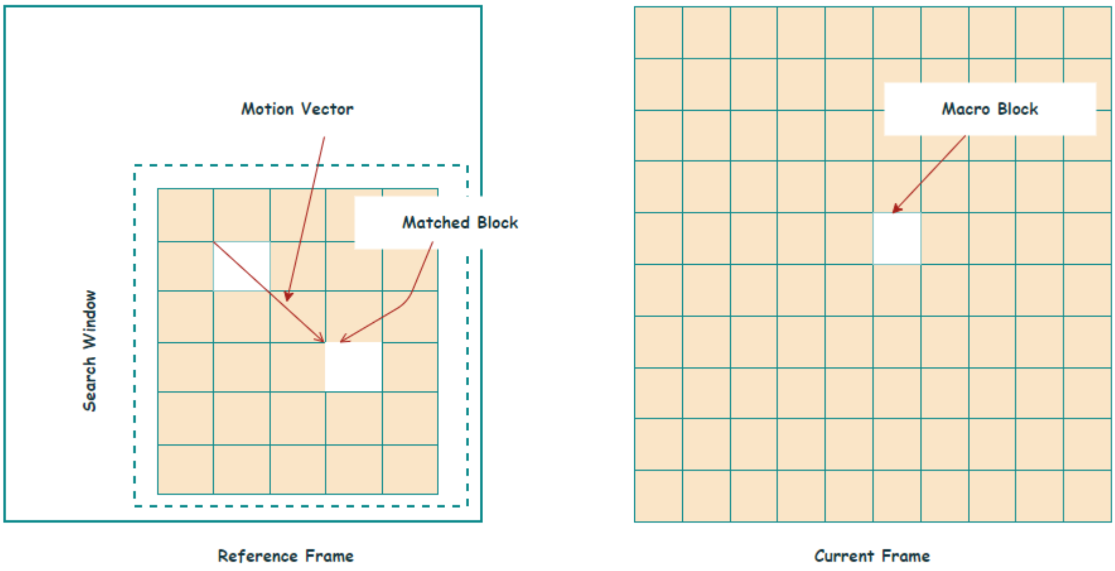


FIGURE 4.4: Using block based method extract MVs.

blocks of N×M pixel values. Let assumes that all pixels in divided block is undergo the same transnational movement. Therefore, all pixels in the same block have the same motion vector, $MV = MV(mv_x, mv_y)$. This MV is determined by finding the optimally matched block from a larger search area.

4.1.2 Matching Criteria for Motion Estimation

Interframe predictive coding is a technique for compressing video sequences by reducing the volume of spatial and temporal redundancy. The variation between the present and expected frames are coded and broadcast to wireless channel is done in general predictive coding. If there is motion in a sequence, a pixel on the corresponding part of the moving object is a better prediction for the current pixel, resulting in a lower predictive error and consequently a lower transmission bitrate. There are several requirements for determining the quality of a match. The MSE Criterion, SAD criterion, and, MPC Criterion are three common matching criteria for BMA motion estimation. [12, 86, 87].

- **MSE Criterion:** Considering $F^{v,t-1}$ is as the references frame and $F^{v,t}$ is as the candidate frame. MSE of a block of pixels computed in the reference frame is given by:

$$MSE(d_x, d_y) = \frac{1}{CxR} \sum_{i=x}^{R+x} \sum_{j=y}^{C+y} \left[F^{v,t}(i,j) - F^{v,t-1}(i+d_x, j+d_y)\right]^2 \tag{4.1}$$

Where the block size is RxC, d_x and d_y are the displacements from candidate block to estimate block in reference frame. x and y are the initial position of estimated block in reference frame. The MV, which is chosen from candidates and presents the MSE's minimum value, is calculated as follows:

$$MV(mv_x, mv_y) = (d_x, d_y) \tag{4.2}$$

Where (d_x, d_y) is displacement positions of minimum MSE value.

- **MPC Criterion:** The pixels of the candidate blocks are compared to the corresponding pixels in the source frame-block among displacement (d_1, d_2), and those which are less than a defined threshold value, i.e., approximately matched, are counted.

$$count(d_x{+}i, d_y{+}j) = \begin{cases} 1, & where\ \ |F^{v,t}(i,j) - F^{v,t-1}(i+d_x, j+d_y)| \leq Tsh \\ 0, & otherwise \end{cases} \tag{4.3}$$

Where Tsh is the predefined trashold value. Then MPC of the candidate block is:

$$MPC(d_x, d_y) = \sum_{i=x}^{R+x} \sum_{j=y}^{C+y} count(d_x + i, d_y + j) \tag{4.4}$$

$$MV(mv_x, mv_y) = (d_x, d_y) \tag{4.5}$$

Where (d_x, d_y) is displacement positions of maximum MPC value block.

- **SAD Criterion :** The SAD already tends to make the error values positive, but rather than adding the squared differences, it sums the absolute differences. The SAD at displacement (d_x, d_y) is described as follows:

$$SAD(d_x, d_y) = \frac{1}{CxR} \sum_{i=x}^{R+x} \sum_{j=y}^{C+y} \left[|F^{v,t}(i,j) - F^{v,t-1}(i+d_x, j+d_y)| \right] \tag{4.6}$$

The MVs are calculated in the same way as the MSE are calculated.

$$MV(mv_x, mv_y) = (d_x, d_y) \tag{4.7}$$

Where (d_x, d_y) is displacement positions of minimum SAD value.

Block Size

The other significant parameter of the BMA is the size of blocks. It performs higher prediction accuracy when the block size is smaller. This is because of several reasons. The impression of the accuracy problems is decreased as the block is small in size. For a smaller block size, the chance of several objects going towards various opposite directions is reduced. In comparison, a reduced block size allows for a more linear transnational approach to non-transnational motion. While the reduced size of blocks implies there will be more blocks (and thus more MVs) across the frame, this improved estimation accuracy comes at the cost of more motion raised. Although many video coding techniques apply macroblock sizes of 16×16, 16×8, 8×16, 8×8, 4×8, 8×4, and 4×4.

Search Range

The other important parameter in BMA is the searching area. The maximum allowable motion distance is d_m, also defined as the searching area, which seems to have a significant effects mostly on the computational latency and accuracy of the BMA's predictions. With a limited d_m, fast-moving regions are poorly compensated, resulting in low prediction accuracy. A broad searching area has been increased prediction accuracy while increasing computational complexity (because there are $(2d_m+1)^2$ candidate blocks to be matched in the searching area). Longer MVs and, as a result, a small improvement in motion overhead can be accomplished with a greater dm [87]. For low-bit-rate applications, a maximum permitted distance of d_m = +15 or -15 pixels is usually sufficient.

4.2 Disparity Estimation

Disparity estimation is used to reduce redundancy between different views in MVC. It is a process of determining a DV between two frames having a different viewpoint from a current viewpoint and calculating a DV of a current viewpoint frame using the determined DVs and a certain translation parameter. To estimate the DV between in two consecutive views, block based estimation technique is popular used similarly to motion estimation [88].

Consequently, analogous to motion estimation, disparity estimation has two modes: forward and backward estimation modes [12, 88]. The forward disparity estimation method denotes a changes determined from the isochronal left viewpoint images, while the backward disparity estimation method denotes a change calculated from the correct viewpoint images that directly follows the isochronal right view picture.

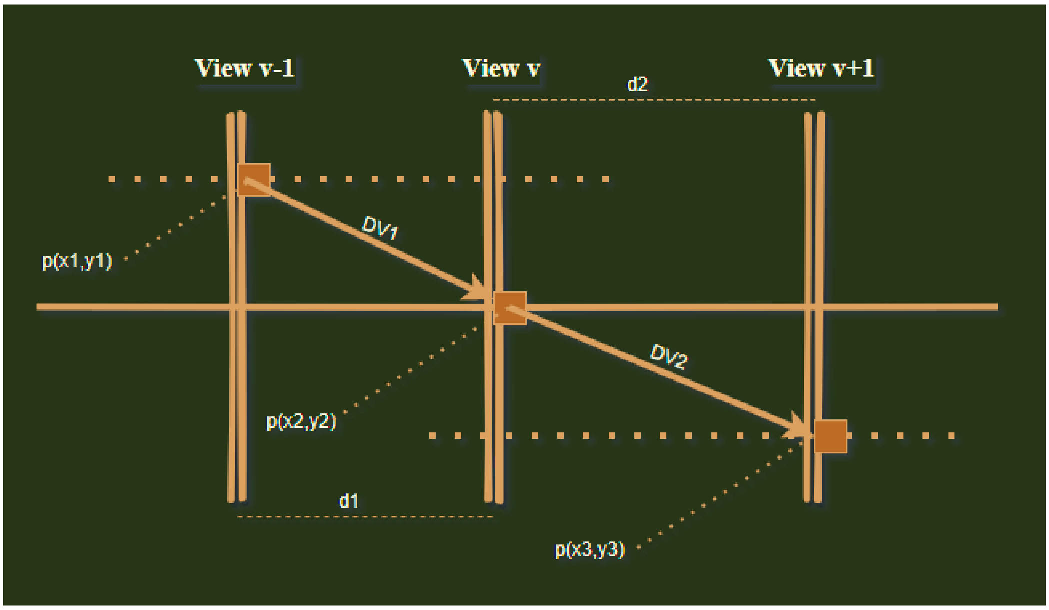


FIGURE 4.5: Block based disparity estimation

BMA disparity estimation method is shown in Figure 4.5. DV1 and DV2 are indicates the estimated DV of the left view (v-1) and the right view (v+1) from current view (v). In block matching disparity estimation process, the current view is dividing into non-overlap blocks based on its size. After divided the view into macro-blocks, compare the block to a corresponding blocks in adjacent left and right views. So to compute DV by selecting the block based on matching criteria's (MSE, SAD, and MPC) like motion estimation [87]. d_1 and d_2 indicates the distance from the left and right view cameras to current view camera. The DV selected from two view candidate is:

$$\begin{aligned} DV1(dv_x, dv_y) &= DV(x_1 - x_2, y_1 - y_2) \\ DV2(dv_x, dv_y) &= DV(x_2 - x_3, y_2 - y_3) \end{aligned} \tag{4.8}$$

4.3 Using MV and DV Feature Extraction Method

In this section, we describe the detail of feature extraction and selection method. In this case, we use to extract features from input MV videos frame buffers based on motion estimation and disparity estimation that are describes on section 4.1 and 4.2. In Figure 4.6 shows, f(v,t) is the lost frame that is between f(v-1,t) and f(v+1,t) depending on camera view. And also it is between f(v,t-1) and f(v,t+1) subject on time sequence. To construct the lost frame f(v,t) we use block based concealment method.So, in this technique, first divide the lost frame into lost Macroblocks (MBs) and then reconstruct each lost block using a recovery method. After that, connect the recovered blocks to reconstruct the lost frame. we use 4×4 block because it is the smallest unit for motion estimation and disparity estimation in H.264/MVC.

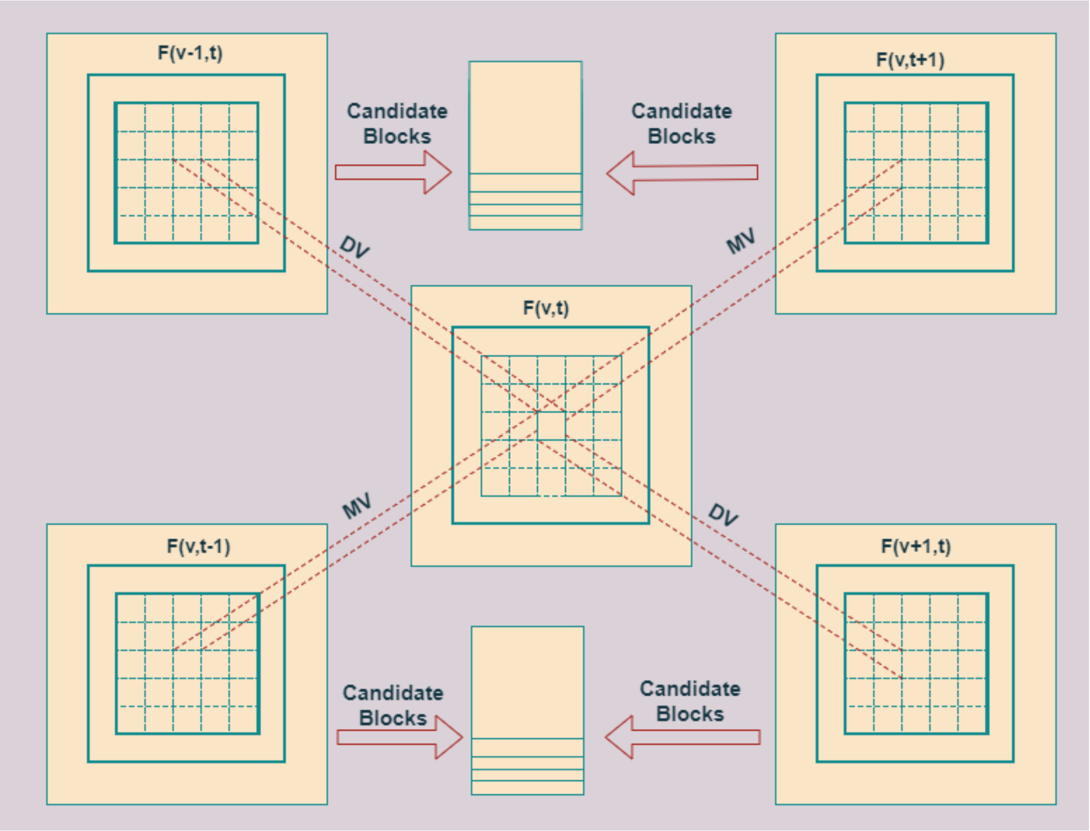


Figure 4.6: Proposed MV and DV extrapolation method from four candidate MVV Frames.

As previous mentioned f (v, t) is the lost frame, and f (v, t-1), f (v, t+1) represents its preceding and subsequent frames in period at the view v, respectively. The missed blocks at f (v, t) is represents by $blk^0_{v,t}, blk^1_{v,t}, blk^2_{v,t}, blk^3_{v,t}, ..., blk^{T_b-1}_{v,t}, blk^{T_b}_{v,t}$ where number of blocks in lost frame $T_b = \frac{N}{4} \times \frac{M}{4}$, N and M are the dimensions of lost frame.

Let $MV_1, MV_2, MV_3, \ldots, MV_{K-1}, MV_K$ are the estimated MVs for f (v, t-1) and f (v, t+1). $DV_1, DV_2, DV_3, \ldots, DV_{K-1}, DV_K$ are estimated DVs for f (v-1, t) and f (v+1, t). In this case K is the number of candidate blocks in single frame. The first step is finding features (MVs and DVs) of the lost frame by begin extrapolating blocks in the searching area of f (v,t-1),f(v,t+1), f(v-1,t) and f(v+1,t). To find the MVs and DVs in MVC is the search range main factor as we discussed in previous topic so in our case we use all blocks as candidate blocks which are overlapped to lost block shown in Algorithm 1. Depending on this to get $2 \times K$ best blocks which are inputs for LSTM model depending on Intra-vew by starting extract the overlapping blocks.

Algorithm 1 An algorithm for Extraction of overlapped blocks

```
Require: $F(v,t-1), F(v,t+1), F(v-1,t), \&F(v+1,t)$
Ensure: 1 $Blk_{v,t-1} \& Blk_{v,t+1}$
Ensure: 2 $MV_{v,t-1} \& MV_{v,t+1}$
 1: Start
 2: $BLK_{v,t-1} \leftarrow BLK^0_{v,t-1}, BLK^1_{v,t-1}, ..., BLK^{T_b}_{v,t-1}$   ▷ blocks in F(v,t-1)
 3: $MV_{v,t-1} \leftarrow MV^0_{v,t-1}, MV^1_{v,t-1}, ..., MV^{T_b}_{v,t-1}$   ▷ motion vectors in F(v,t-1)
 4: $i \leftarrow 0, j \leftarrow 0$
 5: $Blk_{v,t-1} \leftarrow blk^0_{v,t-1}, ..., blk^p_{v,t-1}$   ▷ expected overlapped blocks in F(v,t-1)
 6: $MV_{v,t-1} \leftarrow mv^0_{v,t-1}(x,y), ..., mv^p_{v,t-1}(x,y)$   ▷ expected MV overlap in F(v,t-1)
 7: while $i \leq T_b$ do
 8:     if $BLK^i_{v,t-1}$ is overlapped then
 9:         $blk^j_{v,t-1} \leftarrow BLK^i_{v,t-1}$
10:         $mv^j_{v,t-1}(x,y) \leftarrow MV^i_{v,t-1}$
11:         $j \leftarrow j+1$
12:     end if
13:     $i \leftarrow i+1$
14: end while
15: Repeat step 1 to 10 to extract overlap blocks and MVs in F(v,t+1)
16: $Blk_{v,t+1} \leftarrow blk^0_{v,t+1}, ..., blk^p_{v,t+1}$   ▷ overlapped blocks in F(v,t+1)
17: $MV_{v,t+1} \leftarrow mv^0_{v,t+1}(x,y), ..., mv^p_{v,t+1}(x,y)$   ▷ MVs of overlap in F(v,t+1)
18: End
```

The overlapped blocks projected from $f(v,t-1)$ and $f(v,t+1)$ with there respective MVs.

$$Blk^{i'}_{v,t-1} = \left[blk^{1}_{v,t-1}, blk^{2}_{v,t-1}, blk^{3}_{v,t-1}, ..., blk^{p-1}_{v,t-1}, blk^{p}_{v,t-1}\right] \tag{4.9}$$

$$Blk^{i'}_{v,t+1} = \left[blk^{1}_{v,t+1}, blk^{2}_{v,t+1}, blk^{3}_{v,t+1}, ..., blk^{p-1}_{v,t+1}, blk^{p}_{v,t+1}\right] \tag{4.10}$$

$$\begin{aligned} MV_{v,t-1} = \big[&(mvx^{1}_{v,t-1}, mvy^{1}_{v,t-1}), (mvx^{2}_{v,t-1}, mvy^{2}_{v,t-1}), (mvx^{3}_{v,t-1}, mvy^{3}_{v,t-1}) \\ &, ..., (mvx^{p-1}_{v,t-1}, mvy^{p-1}_{v,t-1}), (mvx^{p}_{v,t-1}, mvy^{p}_{v,t-1}\big] \end{aligned} \tag{4.11}$$

$$\begin{aligned} MV_{v,t+1} = \big[&(mvx^{1}_{v,t+1}, mvy^{1}_{v,t+1}), (mvx^{2}_{v,t+1}, mvy^{2}_{v,t+1}), (mvx^{3}_{v,t+1}, mvy^{3}_{v,t+1}) \\ &, ..., (mvx^{p-1}_{v,t+1}, mvy^{p-1}_{v,t+1}), (mvx^{p}_{v,t+1}, mvy^{p}_{v,t+1}\big] \end{aligned} \tag{4.12}$$

Where $p > k$ is the number of overlapped blocks projected from $f(v,t-1)$ and $f(v,t+1)$, then $p = (2n-1)^2$, n is the dimension of block in our case $n = 4$. After the extraction of overlapped blocks with there respective MVs from lost block then find the correlation between $Blk^{i'}_{v,t-1}$ and $Blk^{i'}_{v,t+1}$ using SAD matching criterion. To find the correlation between overlapped blocks we use $f(v,t-1)$ as a reference frame for Intra-View shown in Algorithm 2. Based on this in below expression 4.15 and 4.14 is defined the SAD with respect MVs of each overlap blocks.

$$\begin{aligned} SAD^{1}_{v,t-1} &= \left[SAD(blk^{1}_{v,t-1}, blk^{1}_{v,t+1}), SAD(blk^{1}_{v,t-1}, blk^{2}_{v,t+1}), ..., SAD(blk^{1}_{v,t-1}, blk^{p}_{v,t+1})\right] \\ SAD^{2}_{v,t-1} &= \left[SAD(blk^{2}_{v,t-1}, blk^{1}_{v,t+1}), SAD(blk^{2}_{v,t-1}, blk^{2}_{v,t+1}), ..., SAD(blk^{2}_{v,t-1}, blk^{p}_{v,t+1})\right] \\ . \qquad & \qquad . \qquad\qquad . \qquad\qquad . \\ . \qquad & \qquad . \qquad\qquad . \qquad\qquad . \\ . \qquad & \qquad . \qquad\qquad . \qquad\qquad . \\ SAD^{p-1}_{v,t-1} &= \left[SAD(blk^{p-1}_{v,t-1}, blk^{1}_{v,t+1}), SAD(blk^{p-1}_{v,t-1}, blk^{2}_{v,t+1}), ..., SAD(blk^{p-1}_{v,t-1}, blk^{p}_{v,t+1})\right] \\ SAD^{p}_{v,t-1} &= \left[SAD(blk^{p}_{v,t-1}, blk^{1}_{v,t+1}), SAD(blk^{p}_{v,t-1}, blk^{2}_{v,t+1}), ..., SAD(blk^{p}_{v,t-1}, blk^{p}_{v,t+1})\right] \end{aligned} \tag{4.13}$$

$$
\begin{aligned}
MV_{v,t-1}^{1} = \big[&(mvx_{v,t-1}^{1} - mvx_{v,t+1}^{1}, mvy_{v,t-1}^{1} - mvy_{v,t+1}^{1}), (mvx_{v,t-1}^{1} - mvx_{v,t+1}^{2},\\
&mvy_{v,t-1}^{1} - mvy_{v,t+1}^{2}), ..., (mvx_{v,t-1}^{1} - mvx_{v,t+1}^{p}, mvy_{v,t-1}^{1} - mvy_{v,t+1}^{p})\big]\\
MV_{v,t-1}^{2} = \big[&(mvx_{v,t-1}^{2} - mvx_{v,t+1}^{1}, mvy_{v,t-1}^{2} - mvy_{v,t+1}^{1}), (mvx_{v,t-1}^{2} - mvx_{v,t+1}^{2},\\
&mvy_{v,t-1}^{2} - mvy_{v,t+1}^{2}), ..., (mvx_{v,t-1}^{2} - mvx_{v,t+1}^{p}, mvy_{v,t-1}^{2} - mvy_{v,t+1}^{p})\big]\\
. \quad & . \quad . \quad .\\
. \quad & . \quad . \quad .\\
. \quad & . \quad . \quad .\\
MV_{v,t-1}^{p-1} = \big[&(mvx_{v,t-1}^{p-1} - mvx_{v,t+1}^{1}, mvy_{v,t-1}^{p-1} - mvy_{v,t+1}^{1}), (mvx_{v,t-1}^{p-1} - mvx_{v,t+1}^{2},\\
&mvy_{v,t-1}^{p-1} - mvy_{v,t+1}^{2}), ..., (mvx_{v,t-1}^{p-1} - mvx_{v,t+1}^{p}, mvy_{v,t-1}^{p-1} - mvy_{v,t+1}^{p})\big]\\
MV_{v,t-1}^{p} = \big[&(mvx_{v,t-1}^{p} - mvx_{v,t+1}^{1}, mvy_{v,t-1}^{p} - mvy_{v,t+1}^{1}), (mvx_{v,t-1}^{p} - mvx_{v,t+1}^{2},\\
&mvy_{v,t-1}^{p} - mvy_{v,t+1}^{2}), ..., (mvx_{v,t-1}^{p} - mvx_{v,t+1}^{p}, mvy_{v,t-1}^{p} - mvy_{v,t+1}^{p})\big]
\end{aligned}
\tag{4.14}
$$

By using equation 4.15 we get the correlation of each overlapped blocks in $f(v, t-1)$ with all candidate blocks in $f(v, t+1)$. And from equation 4.14 we can get there respect distance between candidate blocks in two frames. After this, select

Algorithm 2 An algorithm for Extraction of motion vectors

```
Require: $Blk_{v,t-1}, Blk_{v,t+1}, MV_{v,t-1} \& MV_{v,t+1}$
Ensure: 1 $MV_0, MV_1, MV_2, ...., MV_{K-1}, MV_K$
Start
$SAD \leftarrow [SAD^0, SAD^1, SAD^2, ..., SAD^p]$    ▷ expected SAD
$SAD^i \leftarrow [SAD^i_0, SAD^i_1, SAD^i_2, .., SAD^i_p]$    ▷ each SAD have p SAD arrays
$MV \leftarrow MV^0, MV^1, MV^2 ..., MV^p$    ▷ respective motion vectors
$MV^i \leftarrow MV^i_0, MV^i_1, MV^i_2 ..., MV^i_p$    ▷ motion vectors for each SAD
$MV^{cnd} \leftarrow mv_0, mv_1, mv_2, mv_3 ..., mv_{p \times K}$    ▷ Candidate MVs
$i \leftarrow 0$
while $i \leq p$ do
    $j \leftarrow 0$
    while $j \leq p$ do
        $SAD^i_j \leftarrow SAD(blk^i_{v,t-1}, blk^j_{v,t+1})$
        $MV^i_j \leftarrow diff(mv^i_{v,t-1}, mv^j_{v,t+1})$
        $j \leftarrow j + 1$
    end while
    $i \leftarrow i + 1$
end while
$SAD^i \leftarrow sort(SAD^i)$    ▷ set by there increasing order of SAD
$MV^i \leftarrow sort(MV^i)$    ▷ set based on SAD values
$i \leftarrow 0, k \leftarrow 0$
while $i \leq p$ do
    $j \leftarrow 0$
    while $j \leq K$ do
        $mv_k \leftarrow MV^i_j$
        $j \leftarrow j + 1, k \leftarrow k + 1$
    end while
    $i \leftarrow i + 1$
end while
$MV^{cnd} \leftarrow sort(MV^{cnd})$    ▷ the sort is based on occurrence of motion vectors
$i \leftarrow 0$
while $i \leq K$ do
    $MV_i \leftarrow mv_i$
    $i \leftarrow i + 1$
end while
End
```

K MVs from each candidate blocks by comparing there SAD value.

$$\begin{aligned}
&MV_1^{slc} = \left[MV_1^1, MV_1^2, MV_1^3, MV_1^4, ..., MV_1^{K-1}, MV_1^K\right] \\
&SAD_1^{min} = \left[SAD_1^1 < SAD_1^2 < SAD_1^3 < SAD_1^4 < ... < SAD_1^{K-1} < SAD_1^K\right] \\
\\
&MV_2^{slc} = \left[MV_2^1, MV_1^2, MV_2^3, MV_1^4, ..., MV_2^{K-1}, MV_2^K\right] \\
&SAD_2^{min} = \left[SAD_2^1 < SAD_2^2 < SAD_2^3 < SAD_2^4 < ... < SAD_2^{K-1} < SAD_2^K\right] \\
&. \qquad\qquad . \qquad\qquad\qquad\qquad . \qquad\qquad\qquad\qquad\qquad . \\
&. \qquad\qquad . \qquad\qquad\qquad\qquad . \qquad\qquad\qquad\qquad\qquad . \\
&. \qquad\qquad . \qquad\qquad\qquad\qquad . \qquad\qquad\qquad\qquad\qquad . \\
&MV_{p-1}^{slc} = \left[MV_{p-1}^1, MV_{p-1}^2, MV_{p-1}^3, MV_{p-1}^4, ..., MV_{p-1}^{K-1}, MV_{p-1}^K\right] \\
&SAD_{p-1}^{min} = \left[SAD_{p-1}^1 < SAD_{p-1}^2 < SAD_{p-1}^3 < SAD_{p-1}^4 < ... < SAD_{p-1}^{K-1} < SAD_{p-1}^K\right] \\
\\
&MV_p^{slc} = \left[MV_p^1, MV_p^2, MV_p^3, MV_p^4, ..., MV_p^{K-1}, MV_p^K\right] \\
&SAD_p^{min} = \left[SAD_p^1 < SAD_p^2 < SAD_p^3 < SAD_p^4 < ... < SAD_p^{K-1} < SAD_p^K\right]
\end{aligned} \tag{4.15}$$

Where MV_i^{slc} is stores the i^{th} selected MVs based on SAD_i^{min}. By combine all MVs from $MV_1^{slc}, MV_2^{slc}, MV_3^{slc}, ..., MV_p^{slc}$ we can minimize the candidate blocks to Kxp from pxp.

$$MV^{cnd} = \left[MV_1^1, ..., MV_1^K, MV_2^1, ..., MV_2^K, MV_3^1, ..., MV_3^K, ..., MV_p^1, ..., MV_p^K\right] \tag{4.16}$$

In equation 4.18 MV^{cnd} stores all candidate MVs overlapped blocks. After combine all MVs set the occurrence of MV in MV^{cnd} by comparing each other.

$$occ_i^j = \begin{cases} occ_i^j + 1, & where\ \ if(MVx_i^j == MVx_{i+1}^j)\&\&(MVy_i^j == MVy_{i+1}^j), .. \\ occ_i^j, & otherwise \end{cases} \tag{4.17}$$

Where $i = 1, 2, 3, ...p$, $j = 1, 2, 3, ...K$ and occ_j^i is the occurrence of MV_i^j in MV^{cnd}. By sorting the candidate MVs in decreasing order of occ_i^j. The sequence of the MVs can be expressed as:

$$\begin{aligned} &MV^{cnd} = \left[MV_1^1, MV_2^1, MV_3^1, ..., MV_i^j, ..., \right] \\ &occ = \left[occ_1^1 > occ_2^1 > occ_3^1 > ... > occ_i^j > ....\right] \end{aligned} \tag{4.18}$$

After sorting the MVs we select K MVs as estimated MVs from MV^{cnd} which are there occurrence value is better than from other candidates and so the estimated MVs are.

$$\begin{aligned} &MVs = [MV_1, MV_2, MV_3, ..., MV_{K-1}, MV_K] \\ &occ = [occ_1 > occ_2 > occ_3 > ... > occ_{K-1} > occ_K] \end{aligned} \tag{4.19}$$

Then using $MV_1, MV_2, MV_3, ..., MV_{K-1}, MV_K$ extract blocks from $f(v, t-1)$ and $f(v, t+1)$ as feature or input for LSTM. Depending inter-view disparity estimation $2xK$ best blocks which are inputs for LSTM model projected from $f(v-1, t)$ and $f(v+1, t)$ with there respective DVs $DV_1, DV_2, DV_3, ..., DV_{K-1}, DV_K$. So to get these estimated DVs we use similar block matching method of Intra-view motion estimation technique in equation 4.9 to 4.19.

In similar manner, to extract the DVs from $f(v-1, t)$ and $f(v+1, t)$ we start from define the overlapped blocks from both frames. After this, we find the correlation between two inter-view frames by using SAD concept and we take $f(v-1, t)$ as reference frame. So depending on there SAD values select K DV from candidate blocks of two inter-view frames and after select the DVs by count the occurrence of each DVs by compare each other we select K most frequent DVs from candidate blocks.

$$\begin{aligned} &DVs = [DV_1, DV_2, DV_3, ..., DV_{K-1}, DV_K] \\ &occ = [occ_1 > occ_2 > occ_3 > ... > occ_{K-1} > occ_K] \end{aligned} \tag{4.20}$$

And also using $DV_1, DV_2, DV_3, ..., DV_{K-1}, DV_K$ extract blocks from $f(v-1, t)$ and $f(v+1, t)$ as feature or input for LSTM.

In this thesis, each pixel which are in extracted blocks from $f(v, t-1), f(v, t+1), f(v-1, t)$ and $f(v+1, t)$ represents features or inputs of LSTM model to reconstruct single block in lost frame $f(v, t)$. Figure 4.7 shows the lost block recovery method from selected blocks using LSTM algorithm. $blk_{v,t-1}^{MV1}, blk_{v,t-1}^{MV1}, ..., blk_{v,t-1}^{MVK}$ represents blocks extracted from frame $f(v,t-1)$,$blk_{v,t+1}^{MV1}, blk_{v,t+1}^{MV1}, ..., blk_{v,t+1}^{MVK}$ represents blocks extracted from frame $f(v,t+1)$ both frames are prepare their candidate block based on selected MVs. and also $blk_{v-1,t}^{DV1}, blk_{v-1,t}^{DV1}, ..., blk_{v-1,t}^{DVK}$ represents blocks extracted from frame $f(v-1,t)$,$blk_{v-1,t}^{DV1}, blk_{v-1,t}^{DV1}, ..., blk_{v-1,t}^{DVK}$ represents blocks extracted from frame $f(v-1,t)$ both frames are prepare their candidate block based on selected DVs.

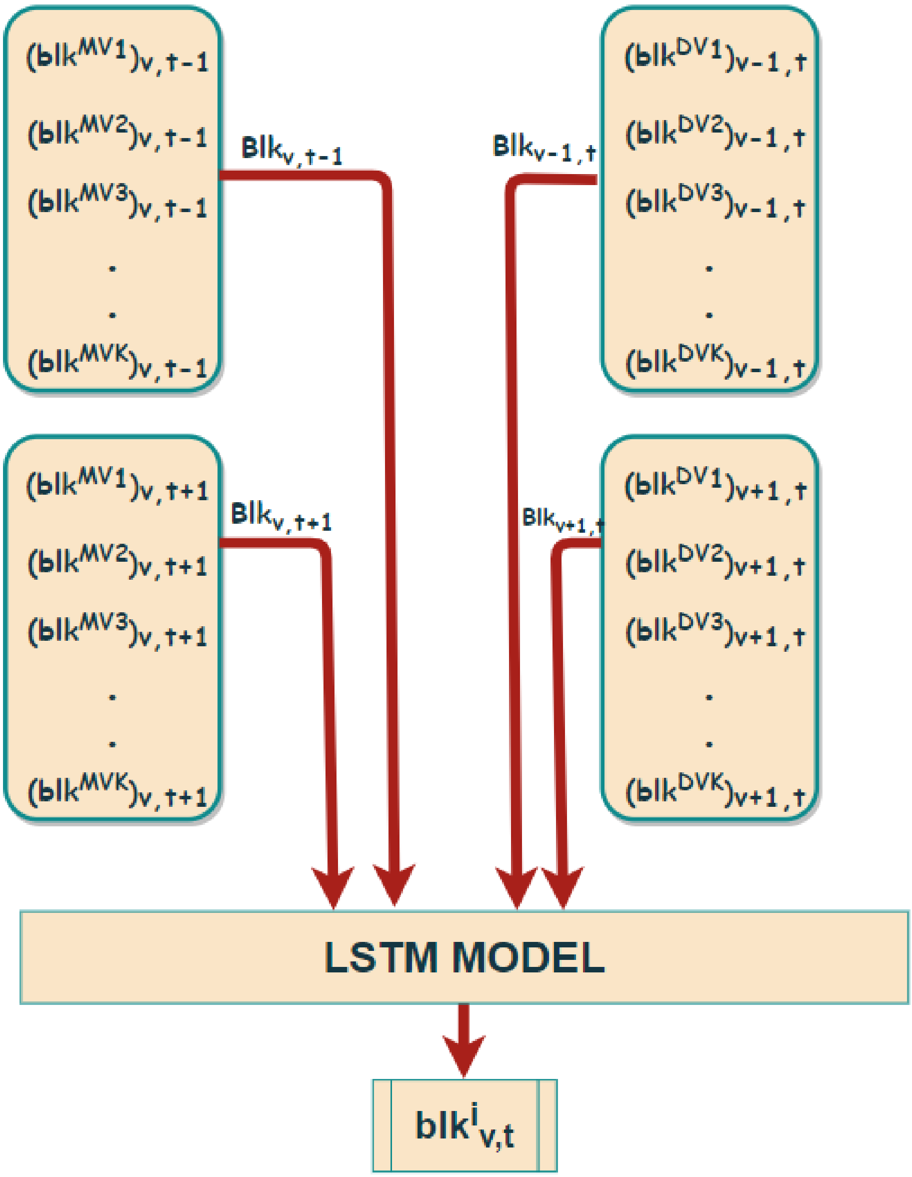


FIGURE 4.7: A proposed LSTM based lost frame recovery in MVV transmission

$$\{Blk_{v,t-1}, Blk_{v,t+1}, Blk_{v-1,t}, Blk_{v-1,t}\} \xrightarrow{\text{LSTM}} \{blk_{v,t}\} \tag{4.21}$$

Where,

$$\begin{aligned} Blk_{v,t-1} &= \left[blk^{MV1}_{v,t-1}, blk^{MV2}_{v,t-1}, ..., blk^{MVK}_{v,t-1}\right] \\ Blk_{v,t+1} &= \left[blk^{MV1}_{v,t+1}, blk^{MV2}_{v,t+1}, ..., blk^{MVK}_{v,t+1}\right] \\ Blk_{v-1,t} &= \left[blk^{DV1}_{v-1,t}, blk^{DV2}_{v-1,t}, ..., blk^{DVK}_{v-1,t}\right] \\ Blk_{v+1,t} &= \left[blk^{DV1}_{v+1,t}, blk^{DV2}_{v+1,t}, ..., blk^{DVK}_{v+1,t}\right] \end{aligned} \tag{4.22}$$

After extracting the candidate blocks combine the blocks and fed them into the LSTM model. The final inputs of the LSTM algorithm are defined in equation 4.23.

$$\left\{x^i_1, x^i_2, x^i_3, x^i_4, ..., , x^i_{T-1}, x^i_T\right\} \xrightarrow{\text{LSTM}} \left\{y^i_1, y^i_2, y^i_3, ..., y^i_{15}, y^i_{16}\right\} \tag{4.23}$$

Where, $x_1 - x_T$ are the pixel value from candidate blocks. $T = 4KX16$, because of we select K-blocks from each candidate frames (f(v,t-1),f(v,t+1),f(v-1,t) and f(v+1,t)) then we have $4K$ total candidate blocks and each blocks have 16 pixels ($size = 4x4$). Therefore the total inputs are $4KX16$ pixels for LSTM model. $y_1 - y_{16}$ are the predicted pixel values of lost block in frame $f(v,t)$. In this case, $K = 4$ was score good results.

The structure of the prepared data for lost frame recovery is shown in Table 4.1. The first column does not input for LSTM it used to to manage the extracted data, the other (x_0-x_{255}) columns are represent the inputs of LSTM which are extracted pixel values from candidate blocks, and the last 16 (Y_0-Y_{15}) columns are represent the actual pixel value of the lost block.

Table 4.1: The structure of the prepared dataset of lost frame recovery

Block No	$\mathbf{x}_0$	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	**.**	**.**	$\mathbf{x}_{254}$	$\mathbf{x}_{255}$	$\mathbf{Y}_0$	$\mathbf{Y}_1$			$\mathbf{Y}_{14}$	$\mathbf{Y}_{15}$
Block0	146	168	186	176			154	184	139	164			165	175
Block1	128	134	151	165			197	216	137	133			154	164
Block2	179	198	197	140			222	219	181	194			200	151
Block3	100	108	120	140			85	44	98	112			120	145
Block4	173	192	173	187			46	60	172	192			173	187
Block5	209	218	219	218			157	178	212	216			219	218
Block6	212	183	99	47			181	168	212	176			89	48
Block7	39	34	36	43			133	127	36	36			39	42
Block8	79	77	65	56			71	58	80	80			75	79
Block9	44	30	18	14			20	11	30	42			14	10
Block10	10	14	21	44			10	27	14	21			22	19
Block11	62	69	64	49			69	64	49	51			54	69
Block12	49	43	41	49			53	61	53	61			52	59
Block13	69	77	78	77			82	84	82	84			82	85
Block14	79	73	66	58			78	68	78	68			84	86

4.4 ConvLSTM-Autoencoder Based Lost Frame Recovery Method

In this section, we present another new lost frame recovery technique which is ConvLSTM-Autoencoder [89] Based Lost Frame Recover. This method consists of three basic components: encoder layer, decoder layer, and Average layer. This approach reconstruct lost frames by using precedent and succeeding frame sequences based on time, as well as the left and right frame sequences based on views. This model is built on the sequence to one architecture idea, which implies that to generate a single lost frame a series of frames based on time and cameras is required. Figure 4.8 shows the model for this method, which has four inputs from different positions: f(v,t-4),f(v,t-3),f(v,t-2), and f(v, t-1) from previous time sequence of frames, f(v,t+4),f(v,t+3),f(v,t+2), and f(v, t+1) from succeeding time sequence of frames, f(v-4,t),f(v-3,t) ,f(v-2,t), and f(v-1, t) from left view

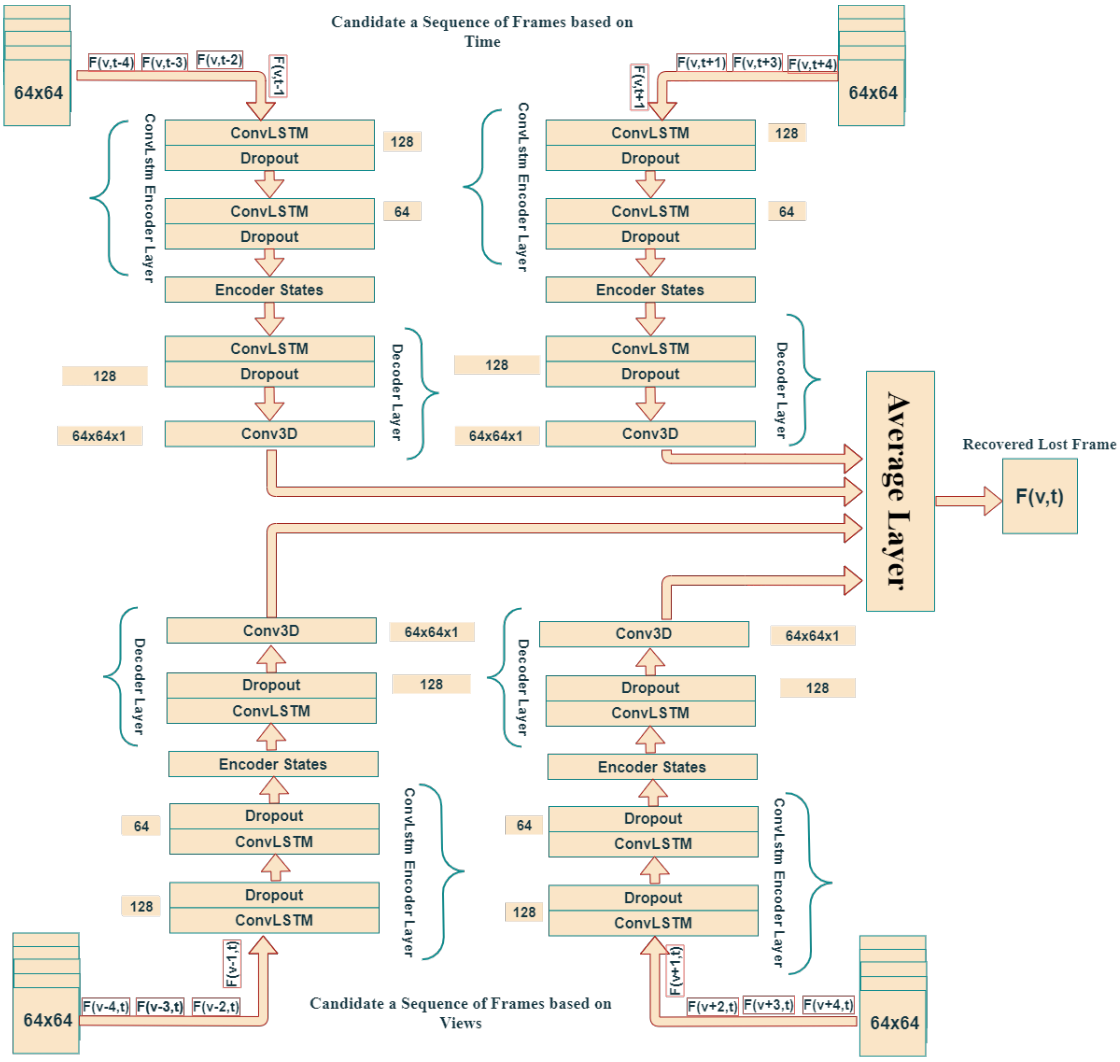


FIGURE 4.8: ConvLSTM autoencoder based lost frame recovery model

sequence of frames, and f(v+4, t),f(v+3,t) ,f(v+2,t), and f(v+1, t) from right view sequence when, f(v, t) is the lost frame in MVV. The encoder layer in this model implemented by using a convolutional LSTM (ConvLSTM) algorithm. ConvLSTM model uses the LSTM layer instead of FC (Fully Connected) layer operators with the convolutional operator [90]. The input-to-hidden and hidden-to-hidden connections are formed using convolution operators. The encoder part of the model has two ConvLSTM layers, with 128 and 64 filters, respectively, with the (5x5) kernel size. To extract the temporal and spatial features in a sequence of MVV frames, the ConvLSTM layers are stacked in a layered structure. After that store, those vectors to encoder state and pass to the decoder layer. The decoder layer consist of one ConvLSTM layer which has 128 filters with(5x5) kernel size and one 3D-CNN layer which has 1 filter with (3,3,3) kernel size. The 3D-CNN used to transform these feature maps from ConvLSTM into actual predictions similar to lost frame size in our case (64x64x1). The average layer takes four distinct vectors with sizes of (64x64x1) as inputs, which are the outputs of the 3D-CNN layer at different positions and times. Finally, it uses these candidate vectors to predict the lost frame.

And we prepare dataset from different sequence types to train this autoencoder model so the structure and sample of prepared data is shown in Table 4.2.

TABLE 4.2: Structure of the dataset to train and test for autoencoder model

Data Num	**Features**	**Frame Sequence Inputs (F(v,t))**				**Output**
Data 1	**Previous Time**	F(4,0)	F(4,1)	F(4,2)	F(4,3)	**F(4,4)**
	Succeeding Time	F(4,8)	F(4,7)	F(4,6)	F(4,5)	
	Left Views	F(0,4))	F(1,4)	F(2,4)	F(3,4)	
	Right Views	F(8,4)	F(7,4)	F(6,4)	F(5,4)	
Data 2	**Previous Time**	F(4,1)	F(4,2)	F(4,3)	F(4,4)	**F(4,5)**
	Succeeding Time	F(4,9)	F(4,8)	F(4,7)	F(4,6)	
	Left Views	F(0,5))	F(1,5)	F(2,5)	F(3,5)	
	Right Views	F(8,5)	F(7,5)	F(6,5)	F(5,5)	
Data 3	**Previous Time**	F(4,2)	F(4,3)	F(4,4)	F(4,5)	**F(4,6)**
	Succeeding Time	F(4,10)	F(4,9)	F(4,8)	F(4,7)	
	Left Views	F(0,6))	F(1,6)	F(2,6)	F(3,6)	
	Right Views	F(8,6)	F(7,6)	F(6,6)	F(5,6)	
...	**...**	...	...	...	...	**...**
	...	...	...	...	...	
Data N	**Previous Time**	F(10,146)	F(10,147)	F(10,148)	F(10,149)	**F(10,150)**
	Succeeding Time	F(10,154)	F(10,153)	F(10,152)	F(10,151)	
	Left Views	F(6,150)	F(7,150)	F(8,150)	F(9,150)	
	Right Views	F(14,150)	F(13,150)	F(12,150)	F(11,150)	

Chapter 5

Result and Discussion

This chapter discusses the experiments that have conducted on both detection and recovery, and the results of these experiments compare with other methods. The experiments are made in Python. And we use Keras library for ANN and LSTM, and we also use pandas library to manipulate the data which are input for deep learning.

5.1 Result and Discussion of Lost Frame Detection

In this section, we discusses the experiment results of lost frame detection and also compare the result to other video and image quality assessment methods. Each input video frame is resized to 100×100 resolution.

5.1.1 ANN Training

In the ANN model training of the proposed work, we prepare normal and lost sample datasets. By finding the features (PSNR, SSIM, VIF, VQM, and FSIM) between two consecutive video frames $((f(v, t-1)$ and $f(v, t))$ or $(f(v, t)$ and

$f(v,t+1))$) we have been prepared normal dataset. And also by removing one or more frames from video sequences and by find the features between those frame (($f(v,t-1)$ and $f(v,t+1)$) , ($f(v,t)$ and $f(v,t+2)$) or ($f(v,t-1)$ and $f(v,t+2)$)) we can prepare lost datasets. The dataset was split randomly into 80% for training and 20% for testing. We use 20% data from training dataset for validation purpose.

The training set of examples has been used to learn the network classification of sample data into class of lost frame or normal (no lost) in video transmission. The validation set was used to manage the learning process, such as calculating regularization parameters and optimizing neural network architectures. The test set was created specifically for the purpose of evaluating the output of a fully specified classifier (neural network).

Hyper-parameters are crucial for machine learning algorithms because they directly regulate the behaviors of training algorithms and have a major impact on model performance. The hyperparameters used to develop lost frame detection model are shown in Table 5.1.

Table 5.1: Used hyperparameters in the model

Hyperparameter	**Value**
Batch Size	32
Activation Function	Sigmoid
Optimizer	Adam
Dropout	0.2

In ANN, deciding on a number of neurons in a hidden layer is a difficult task. Below Table 5.2 describes the performance of different architectures combination 1-7 respectively. Among the other architectures, the [24x16x8] architecture (24 nodes in the input layer, 16 nodes in the hidden layer, and 8 nodes in the output layer) does have best training accuracy performance.

TABLE 5.2: Best architecture performances for classification

Combination	Layer 1	Layer 2	Layer 3	Accuracy
1	12	8	4	84.65
2	15	10	5	87.84
3	16	12	8	90.03
4	18	14	10	91.45
5	20	15	10	92.12
6	**24**	**16**	**8**	**93.11**
7	22	18	14	91.76

5.1.2 Test Results

The ANN training has been reaches good result at epochs 500s. By using the selected ANN architecture (combination 6, [24,16,8]) in Table 5.2. The results indicate **93.11%** classification accuracy. The training efficiency graph of the neural network [24,16,8], which is achieved the best result among the other ANN architecture, is shown in Figure 5.1.

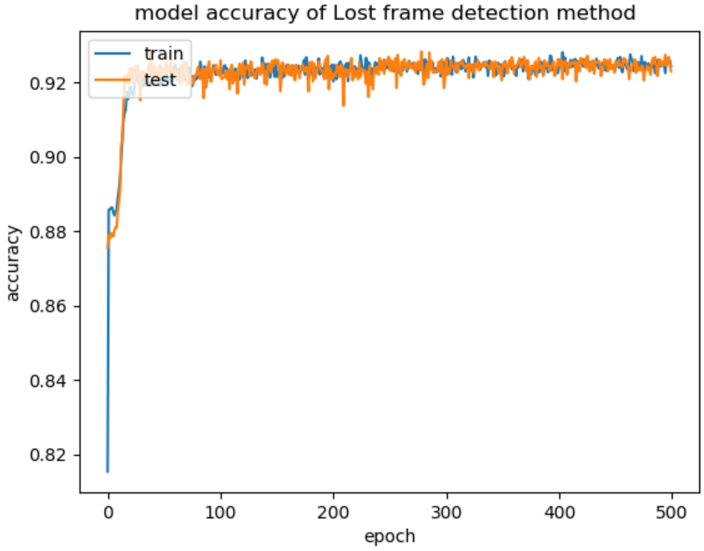


FIGURE 5.1: Accuracy versus different epoch of ANN based lost frame detection.

Figure 5.2 shows the epoch-by-epoch cross-entropy loss for the train (blue) and test (orange) datasets. The training process completed well, as seen in both Figures 5.1 and 5.2.

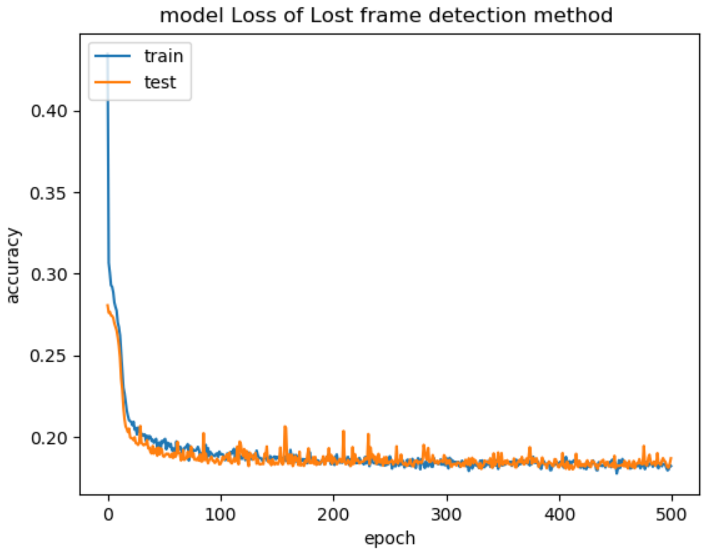


Figure 5.2: Loss versus different epoch of ANN based lost frame detection.

In the previous test results, the ANN was used 20% data for the test set and 80% use for the training set. To judge, the ANN accuracy acquired from a single set does not provide a complete result of the ANN model performance. When extremely varied accuracies are observed after testing a model with distinct test sets, this is referred to as variance. The k-fold cross-validation methodology is implemented in this experiment to optimize the approach used to assess the ANN. This methodology has the advantages of using all samples for both training and test, with each sample being used only once for a test.

We use k=10 in this experiment, which is a frequent choice in many papers. We divided the original dataset into ten subgroups based on this. A single subset was chosen for testing from the ten subsets. The training sets were made up of the remaining 9 subsets. By changing the test and training sets, the cross-validation process was repeated ten times. The parameter and hyperparameters of the model were used again. Figure 5.3 shows the results of the 10-folds cross-validation experiments. The average accuracy of the 10-folds has 92.83%. The variation of the accuracy of all folds also calculated by using the variance concept then it scores 0.000025.

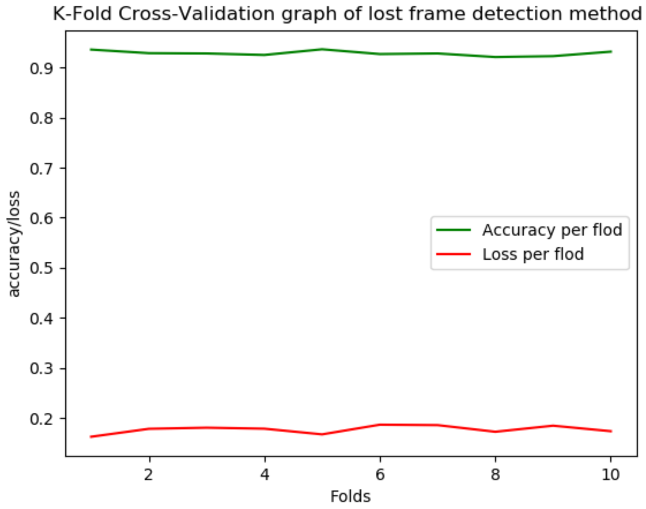


FIGURE 5.3: K-fold cross-validation accuracies and loss for fitting the training Set.

The classification confusion matrix in Figure 5.4 is used to calculate the different types of errors that occur during the training phase, which is the second way of measuring the neural network's performance. To generate the confusion matrix, data is fed into the ANN model. The comparison between the desired and predicted classification groups is stored in the confusion matrix.

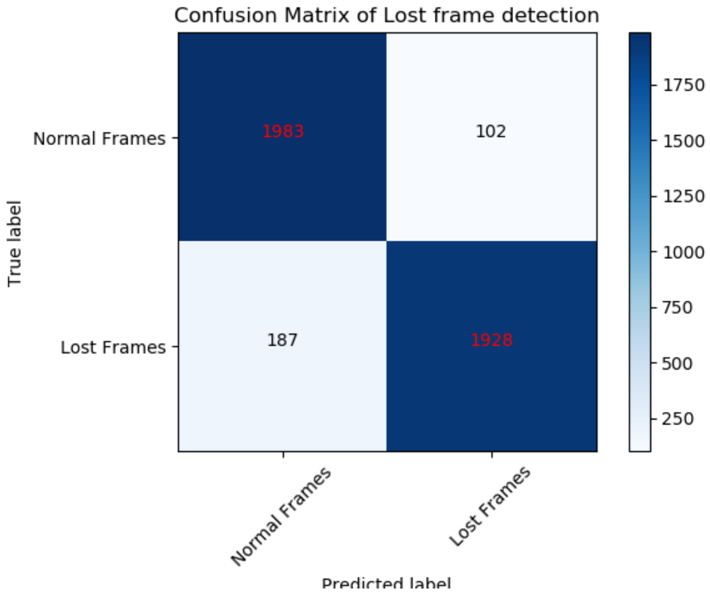


FIGURE 5.4: Confusion matrix of ANN based lost frame detection.

The diagonal cells are seen in blue in the case of successful classification of a targeted class case, indicating the number of cases correctly classified by the ANN to classify feature condition. Whether or not there have been any lost frames. Based on this from 2085 Normal frames 1983 (95.10%) frames detected exactly Normal and from 2115 lost frames 1928 (91.15%) frames detected exactly lost. And also diagonal cells in white, indicates the number of cases that have been wrongly classified by the ANN, was not identified the condition of features. Based on this from 2085 Normal frames 102 (4.89%) frames detected as Lost frame and from 2115 lost frames 187 (8.85%) frames detected as Normal frames. The overall number of cases that were correctly categorized is seen in blue in the matrix, and inversely in white.

The other useful testing way for the binary classification method is ROC (Receiver Operating Characteristic) curve approach. It is a graph that shows how well a classification model performs across all classification thresholds. The x-axis indicates False Positive Rate (FPR) and the y-axis indicates True Positive Rate (TPR). The Area Under the Curve (AUC) is a summary of the ROC curve that measures a classifiers ability to distinguish between classes. Figure 5.5 shows the ROC graph of the lost frame detection. In this case the AUC value is 0.98. This suggests an 98 % chance that the model distinguish a normal frames from lost frames in video

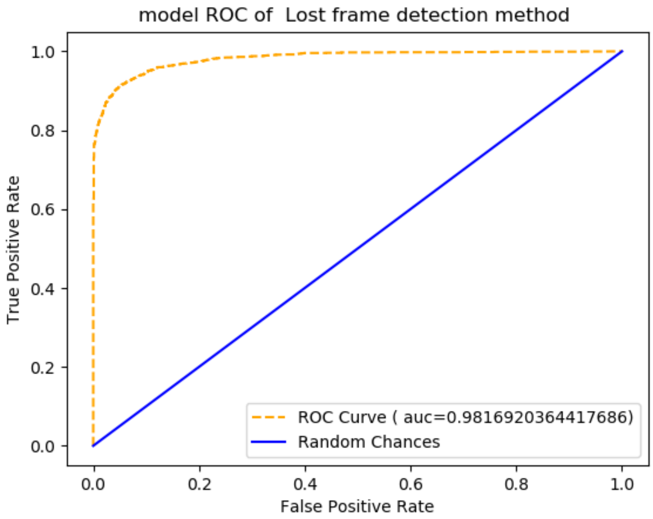


Figure 5.5: ROC curve of ANN based lost frame detection method

frame sequences.

The proposed ABNN model achieved the **93.12%** classification accuracy rate; this indicates the distributed ANN algorithm's output efficiency in reducing the degree of uncertainty management in decision-making.

5.1.3 SVM Training

To train and test SVM we were used similar train and test data of ANN method. The feature extraction method have done by using VQM and IMQ techniques it also similar to ANN. The RBF kernel with gamma and C values of 1 and 1001, respectively, is used to create the SVM model. The grid search method is implemented to optimizing the SVM hyper-parameters. In this SVM model the validation accuracy is **92.71 %** and it performs **91.52%** to detect the lost frames in MVV transmission.

5.1.4 Comparison Results

In this section we aim to verify the performance of the proposed detection method compare with other methods. We compare the results with SVM and other VQM methods.

5.1.4.1 Compare ANN and SVM

The accuracy, precision, recall, and F1 score values of the ANN model were compared with the SVM model, shown in Table 5.3. The results show that the ANN model outperforms than SVM approach.

Table 5.3: SVM Vs ANN

Algorithm	Accuracy	Recall	Precision	F1 Score
ANN	0.931	0.913	0.951	0.932
SVM	0.915	0.881	0.958	0.918

5.1.4.2 Comparing with other Metrics

In this section we aim to compare the performance of the proposed detection method with other VQMs. To compare detection accuracy of proposed method with other VQMs, the test set 4200 sample data was used which have 2085 Normal actual data and 2115 Lost actual data. The test data have been includes five different video sequences types at different bandwidth $< 128Kbs, 128 - 250Kbs, 250 - 750Kbs, 750 - 1250kbs, > 1250Kbs.$

To compare the proposed method to other video metrics we use threshold(T_{sh}) values of objective video metric with subjective video metrics. Table 5.4 presents the mapping of objective video metrics to subjective video metrics. subjective user-perceived video quality is commonly stated using a 5-point MOS (Mean Opinion Score) scale (i.e., excellent, good, fair, poor, bad). So for PSNR, SSIM, VIF, and FSIM the MOS is expressed depending on calculated value $\geqslant T_{sh}$ but for VQM calculated value $\leqslant T_{sh}$.

Table 5.4: Mapping T_{sh} value of video metric based on subjective video metrics

Bandwidth(kb/s)	**MOS**	**PSNR**	**SSIM**	**VQM**	**VIF**	**FSIM**
>1250	Excellent	36	0.94	2	0.56	0.95
>750	Good	29	0.85	1.5	0.4	0.9
>250	Fair	24	0.76	1	0.27	0.85
>128	Bad	22	0.7	0.8	0.16	0.8
<64	Very Bad	20	0.62	0.7	0.12	0.75

Based on the mapping T_{sh} value we analyze the experiment results of the video metrics and proposed metrics. Figure 5.6 shows the classification accuracy percentage comparison of our proposed method respect to PSNR, SSIM, VQM, VIF and FSIM at different video transmission network scenario (very annoying, annoying, slightly annoying, perceptible, and imperceptible network)[91]. Then, Figure 5.6 shows the experiment results indicate PSNR 62.2%, SSIM 57.6%, VQM 51.2%, VIF 54.2%, FSIM 57.6%, and proposed method 93.12% scores average accuracy to classify lost frame or normal frame in video transmission.

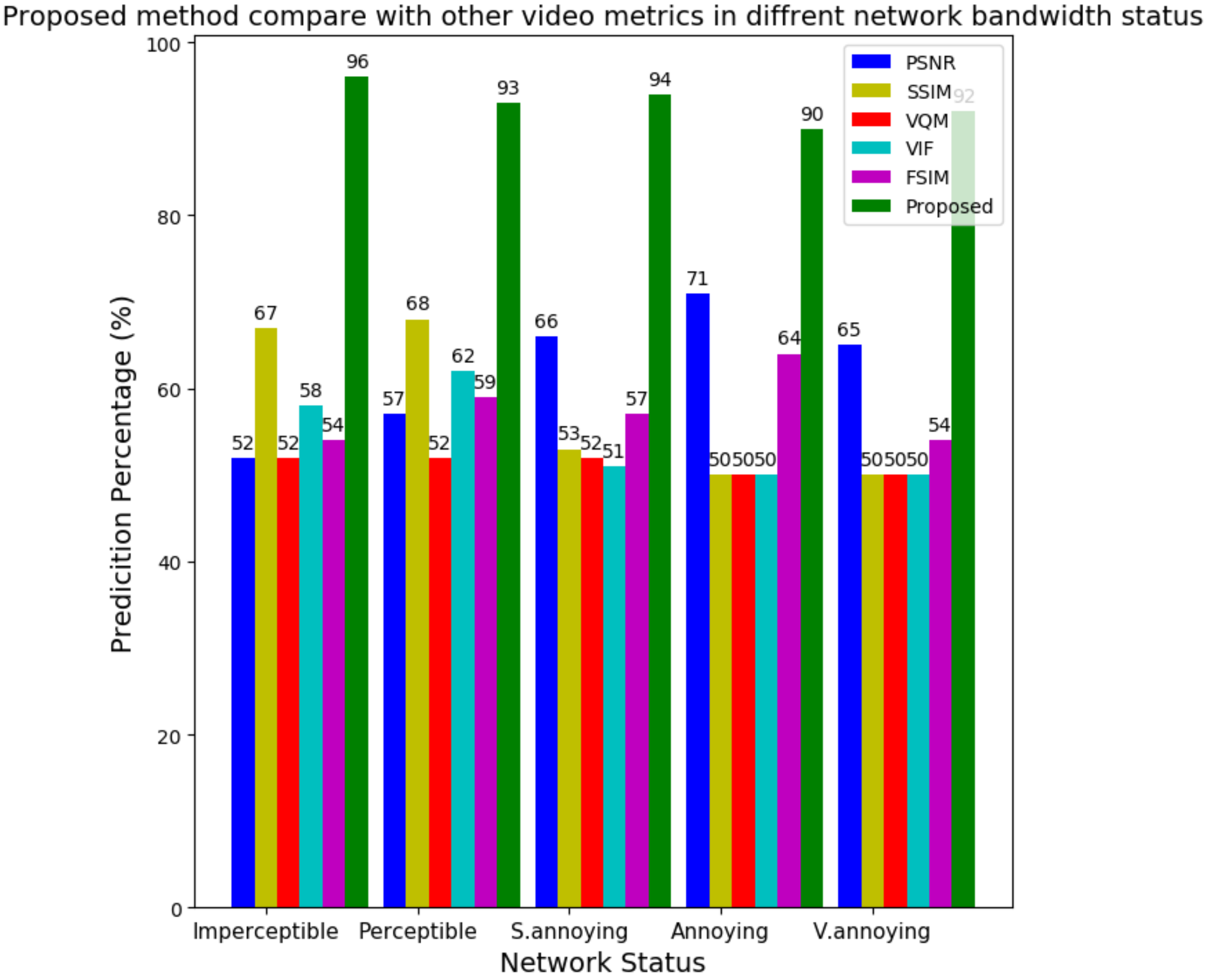


FIGURE 5.6: Comparison of VQMs with The proposed method

As the experiment indicates the proposed model achieved the highest lost frame detection accuracy than other VQMs. It achieves this result because of two basic reasons.

- These objective video metrics have assessed the videos based on only specific characteristics (The intensity difference between consecutive frames, Structural difference between two frames, frame fidelity measurement-based on NSS and spatiotemporal). In this thesis, ANN model adapts all characteristics of the video raw frames have been lost or normal frame by taking those VQMs as parameters to get the property of video.

- There are no standard T_{sh} values for objective video metrics to detect the lost frames in video transmission. We use the mapping value of objective video metrics to subjective metrics as T_{sh} values. Because of this, other video metrics achieve lower classification accuracy than our method.

This lost frame detection model has sometimes been classified wrongly which means it says lost frames when the transmission is without lost frames or no lost frames but the transmission has lost frames. so why this misclassification has happened in the detection model, because the prepared data was from different types of MVV like very fast motion videos and very slow-motion videos. In very fast-motion videos the frame sequences are almost similar which means no difference so in this case, detecting the lost frames in such like videos are difficult and in very slow-motion videos the gap between two consecutive sequence frames is high so in this case, to identify the normal frames it is difficult.

5.1.5 Complexity Analysis

Figure 5.7 presents the effect of increasing the number of pixels in a single video frame versus the execution time of VQMs and the Proposed method.

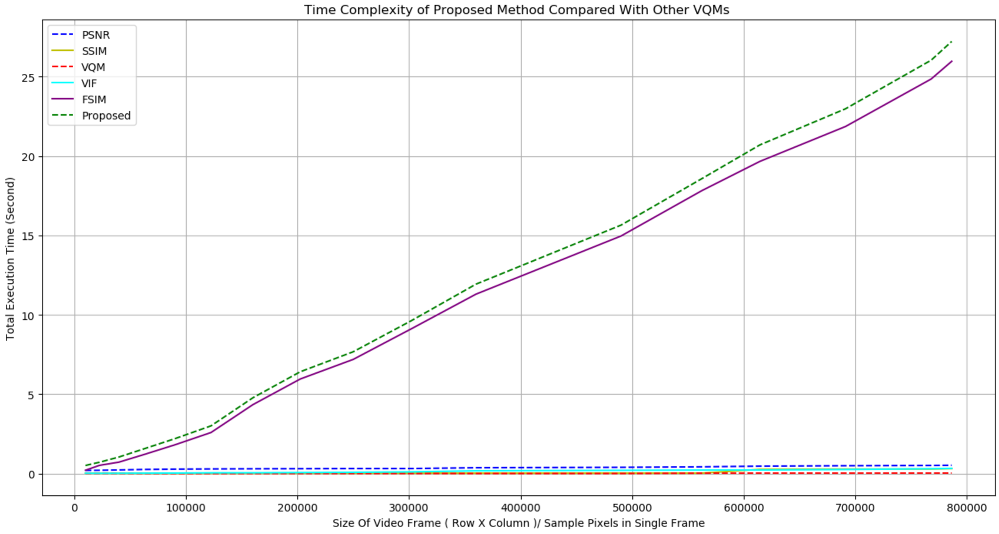


Figure 5.7: Time complexity comparison of VQMs with the proposed method

As a result, shows that the detection time is directly proportional to the number of pixel for both VQMs and the proposed method. We downsized the video frames to 100×100 as described in section 5.1. In this scenario, the proposed detection methods complexity was acceptable or almost comparable to other approaches.

5.2 Result and Discussion of RNN based lost frame Recovery

In this section, we discussed the results of proposed lost frame recovery method and we compare the performance of the proposed method against other recovery methods.

5.2.1 LSTM Training

The input data is the extracted candidate pixels features from MVV buffer frames data. The training and testing data were derived from the input dataset. we use 70% of the data to train the LSTM, and 30% has been used to test the model. The dataset is prepared from all type of MVV sequences, which are list in section 3.1.8. The LSTM model consists of three LSTM layers each layer consists 50 hidden neurons and 256 input pixels with 16 outputs. A total number of $1,440,000$ sample data were generated from different video sequences. The LSTM was trained using $1,008,000$ data from the prepared dataset, and it has been correctly predicted the next $432,000$ test pixels with an RMSE of $1.4e^{-5}$. The model was fit using the Adam optimizer and trained for 400 epochs with RMSE as the loss function.

5.2.2 Test Results

The Figure 5.8 and 5.9 shows the comparison of the recovered and the original pixel signals of the test frames from selected video sequences. The sample tested pixels were above 400, 000 pixels, which are prepared from different type of video sequences. In the graph, x-axis represents the number of tested sample pixels and y-axis shows the respected intensity value of sample pixels. The recovered intensity value represented by green in the graph and the red part indicates the original intensity value of the sample tested pixels.

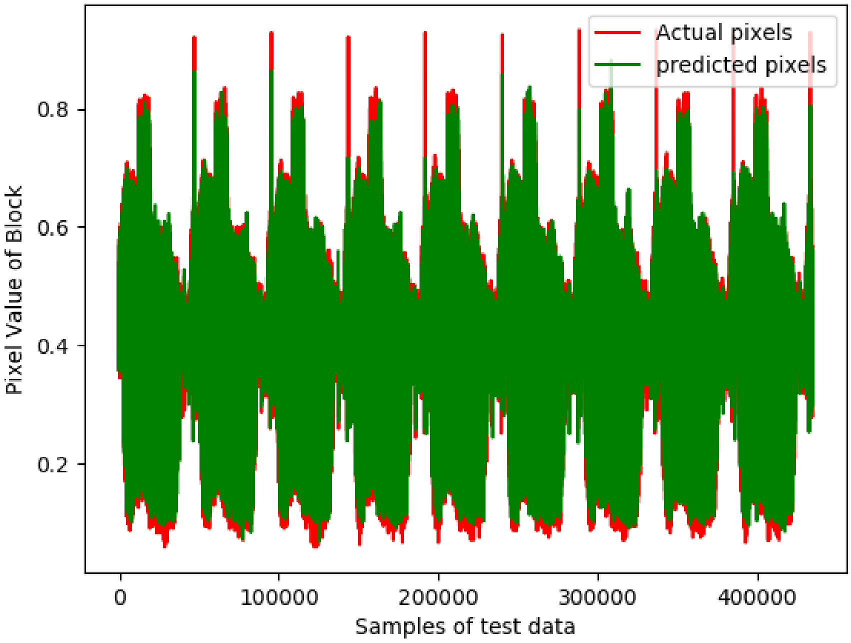


FIGURE 5.8: Sample recovered pixel with expected pixel

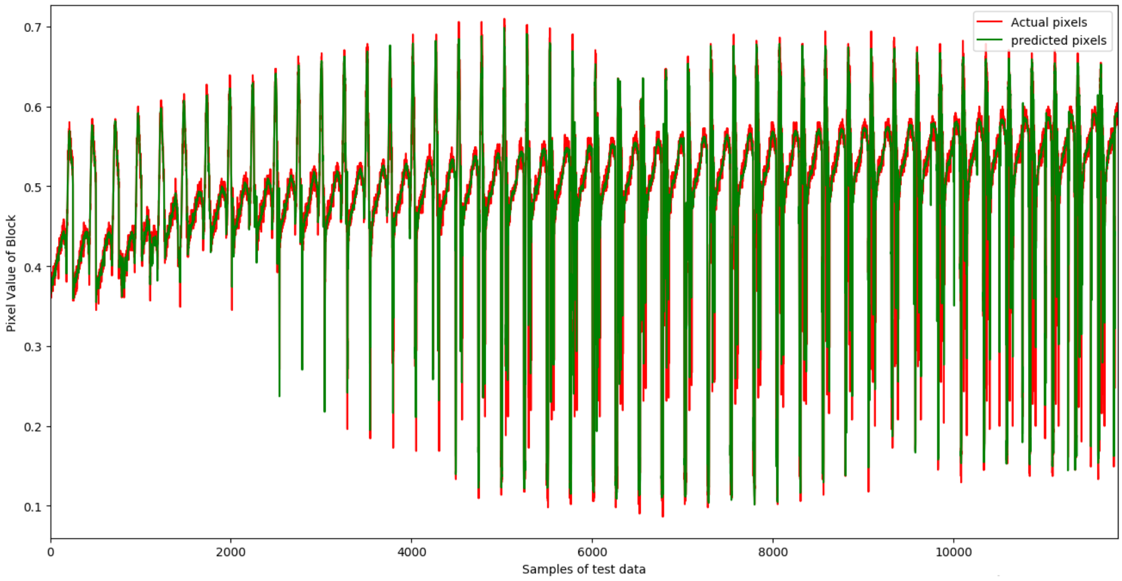
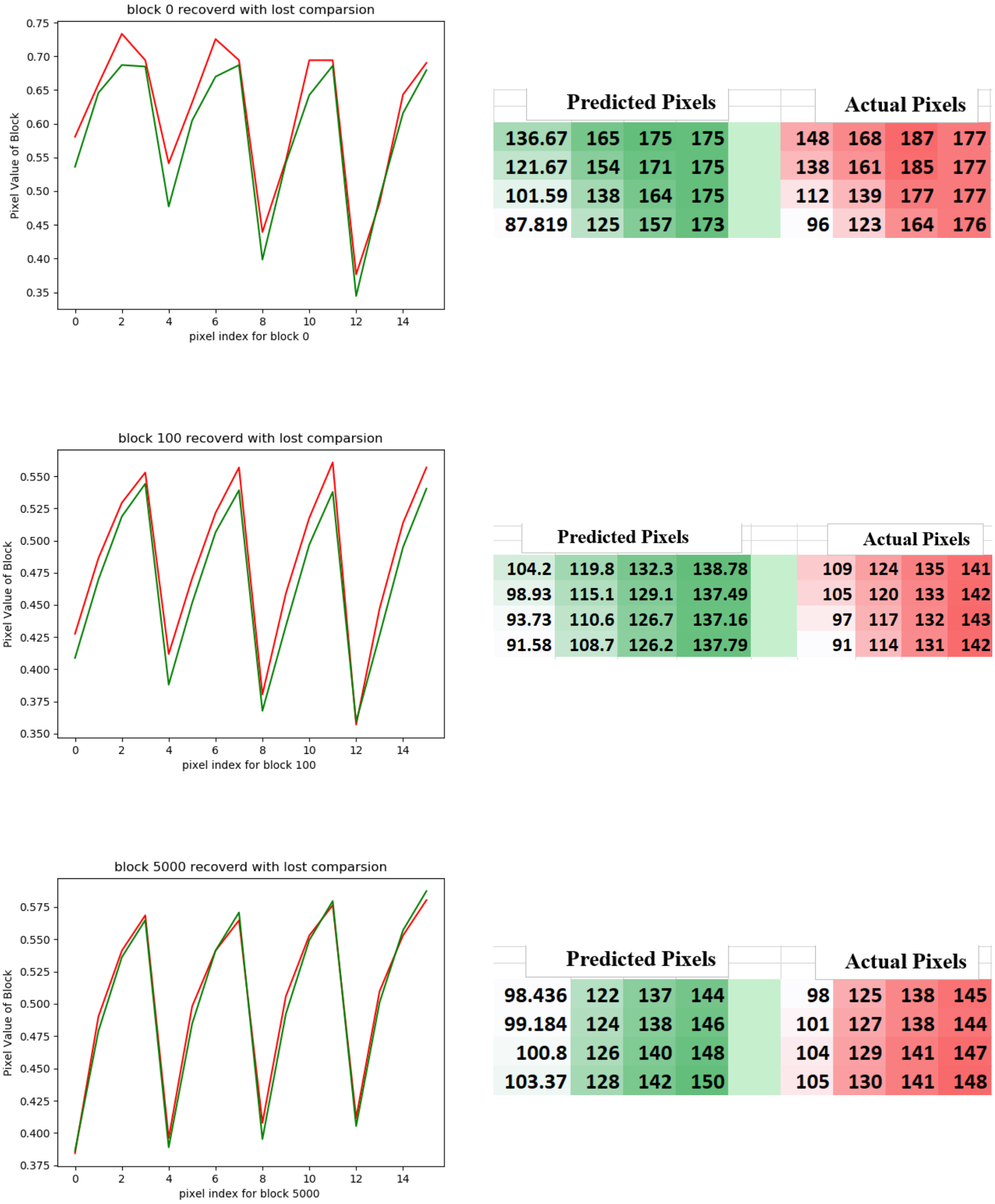


FIGURE 5.9: Zoomed sample recovered pixel with expected pixel

Figure 5.10 shows the comparison of actual and predicted pixel values of single blocks in lost frame of Akko&Kayo 4^{th} sequence. Then MSE values of block0, block100, and block5000 are 92.76, 19.85, and 3.808 respectively.



Predicted Pixels					Actual Pixels			
136.67	165	175	175		148	168	187	177
121.67	154	171	175		138	161	185	177
101.59	138	164	175		112	139	177	177
87.819	125	157	173		96	123	164	176

Predicted Pixels					Actual Pixels			
104.2	119.8	132.3	138.78		109	124	135	141
98.93	115.1	129.1	137.49		105	120	133	142
93.73	110.6	126.7	137.16		97	117	132	143
91.58	108.7	126.2	137.79		91	114	131	142

Predicted Pixels					Actual Pixels			
98.436	122	137	144		98	125	138	145
99.184	124	138	146		101	127	138	144
100.8	126	140	148		104	129	141	147
103.37	128	142	150		105	130	141	148

FIGURE 5.10: Sample recovered blocks of lost 4^{th} frame in Akko&Kayo sequences

Based on the MSE value the blocks set in first column of frames are poorly predicted result than others. And also in Figure 5.11 presents the comparison of original and predicted all blocks (18644 blocks) in Akko&Kayo 4^{th} sequence.

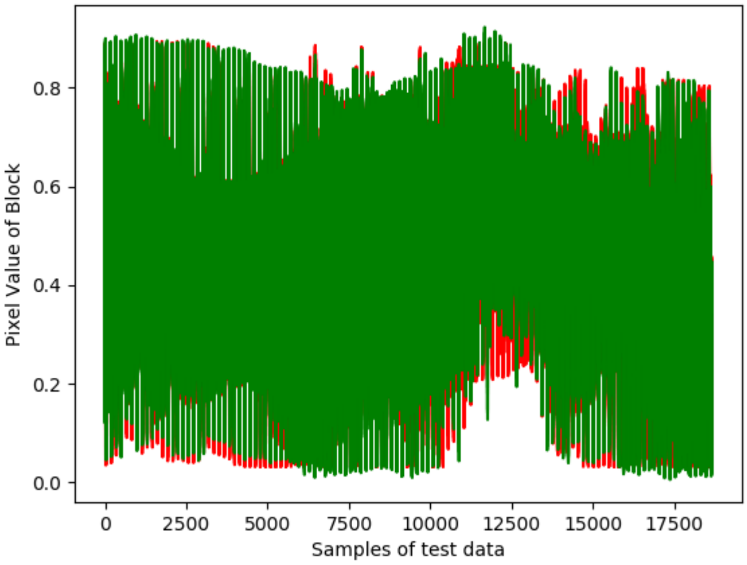


FIGURE 5.11: Sample tested lost frame and recovered 4^{th} frame of Akko&Kayo sequence

Table 5.5 shows the MSE test value of the proposed work for different video sequences. Most of the sequences scores low MSE value. We test the video frames by removing one frame from the sequence of frames then we try to reconstruct the lost frame by using our proposed work. After recover we compare the original video frame to recovered video frame by finding the MSE between two frames.

TABLE 5.5: The scored MSE values of the predicted and the original frame of different test sequences

Sequence type	**Size of Frame**	**MSE**
Akko&Kayo	640X480	17.26
Balloons	1024X768	8.75
Book_Arrival	1024X768	38.56
Break_dance	1024X768	14.498
Bullet	1280X960	16.75
Dog	1280X960	12.63
Lovebird	1024X768	101.34
Newspaper	1024X768	25.56
Rena	640X480	22.72
Soccer_arc	1920X1080	23.74

.

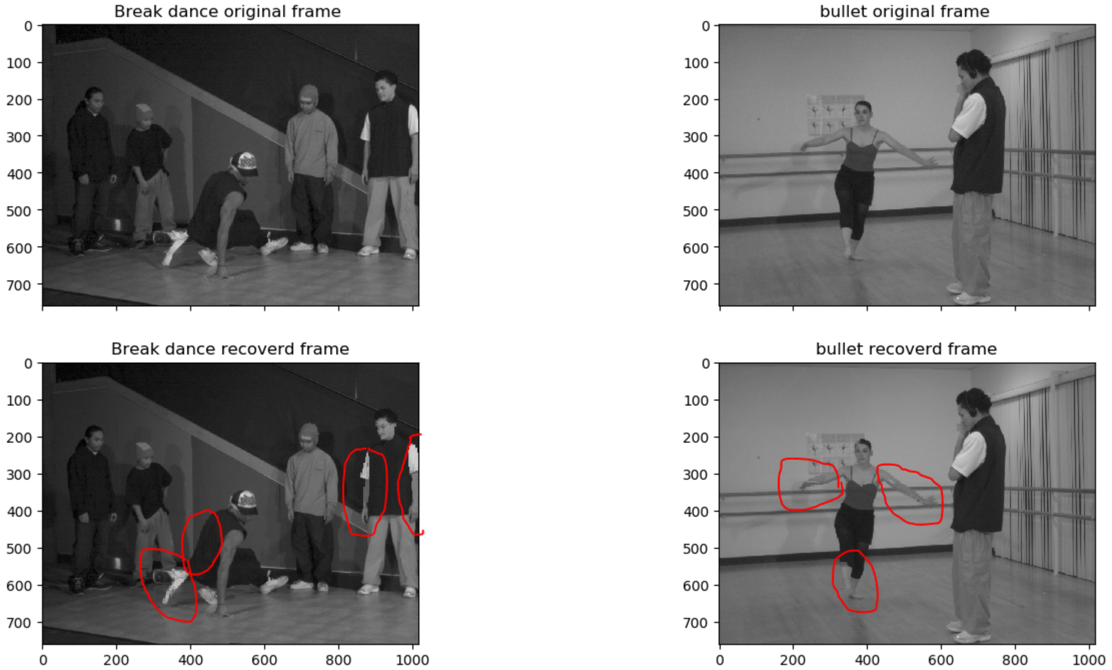


FIGURE 5.12: Visual comparison of performance between predicted and original frames of bullet and breakdance

Figure 5.12 also shows the visual comparison of the recovered frame and the original frame of break dance and bullet sequences. then the image shows that the proposed method provides good visual result than other recovery method.

5.2.3 Comparison Results

In this section, we evaluate the performance of different deep learning models (LSTM, Bi-LSTM, GRU, and ConvLSTM Autoencoder) as well as other lost frame recovery methods.

5.2.3.1 Comparing with LSTM, Bi-LSTM, and GRU

To train and test Bi-LSTM and GRU we were used similar train and test data of LSTM method. The feature extraction method have done by using MV and DV technique it also similar to LSTM. The Bi-LSTM model consists of three Bidirectional LSTM layers each layer consists 100 hidden neurons and 256 input pixels with 16 outputs and The GRU model also consists of three gru layers each layer have 256 hidden neurons and 256 input pixels with 16 outputs. And other training parameters (epoch=400, batch size=64, optimizer=adam) are used for all methods. Figure 5.13 shows the comparison results of predicted frame by using LSTM, Bi-LSTM, and GRU to actual (expected) frame intensity value of each pixel location in lost frame.

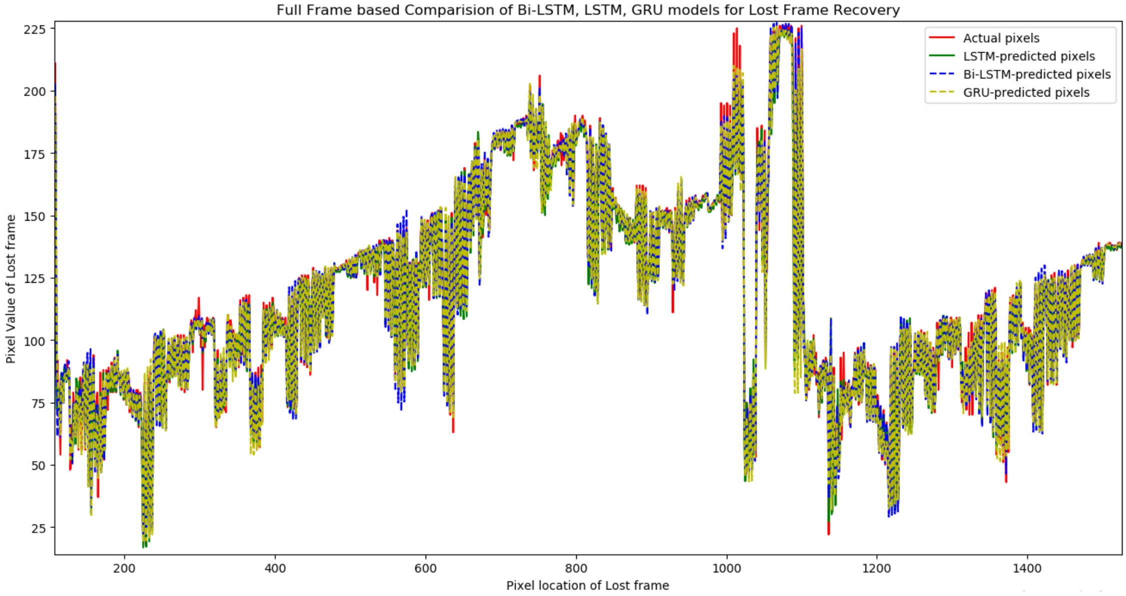


Figure 5.13: Graphical comparison of predicted frame using GRU, Bi-LSTM, and LSTM with original frame

The comparsion MSE value of predicted frame using LSTM, Bi-LSTM, and GRU to actual frame are shown in Table 5.6. The experiment results are recorded from different MVV frame sequences after that we take the average MSE values.

Table 5.6: The comparison result of LSTM, GRU, and Bi-LSTM

Method	**MSE**	**PSNR**
LSTM	16.47	35.97
Bi-LSTM	19.30	35.27
GRU	20.036	35.11

As the result indicates the frame sequence recovered using LSTM algorithm are more correlated to original sequences than other(Bi-LSTM and GRU) methods.

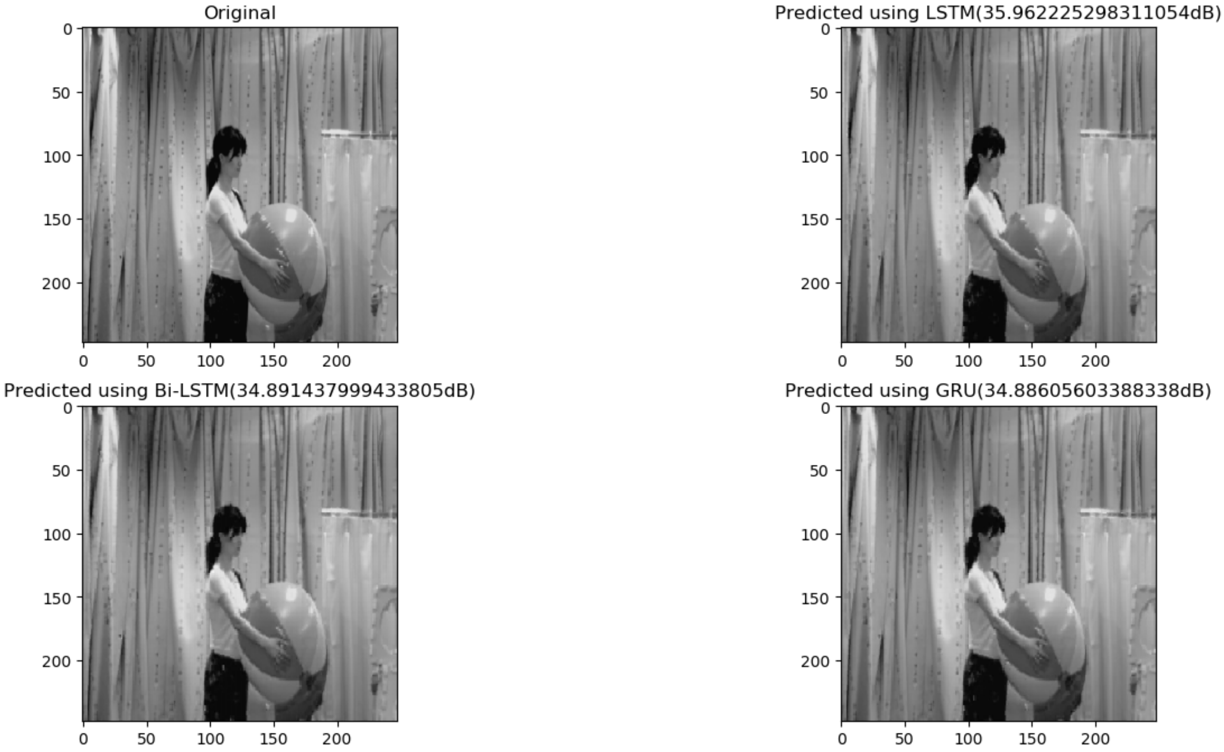

Original
Predicted using LSTM(35.962225298311054dB)
Predicted using Bi-LSTM(34.891437999433805dB)
Predicted using GRU(34.88605603388338dB)
0
50
100
150
200

FIGURE 5.14: Visual comparison of lost frame reconstruct using LSTM,Bi-LSTM, and GRU

The visual comparsion of the original frame to recovered frame by using LSTM, Bi-LSTM, and GRU method is shown in Figure 5.14.

5.2.3.2 Comparing with Autoencoder based Lost Frame Recovery Method

In this section, we show that experimental results of ConvLSTM and Bi-ConvLSTM Autoencoder-based lost frame recovery methods. We prepared 4000 data from ten different MVV sequences. 80% of the data used for training, while the remaining 20% of the data used for testing . Like the ConvLSTM Autoencoder model which is discussed in Section 4.4, the Bi-ConvLSTM Autoencoder model has three fundamental components Encoder, Decoder, and Average layer. Bi-ConvLSTM Autoencoder uses the Bi-ConvLSTM layers instead of ConvLSTM layers to build the encoder and decoder part of the model. The other components and methods of Bi-ConvLSTM Autoencode are similar to ConvLSTM Autoencoder. We used batch size = 32, optimizer=adam, epochs=100, and sigmoid as activation functions to train this model. Figure 5.15 presents the experimental results of ConvLSTM and Bi-ConvLSTM Autoencoder-based lost frame recovery.

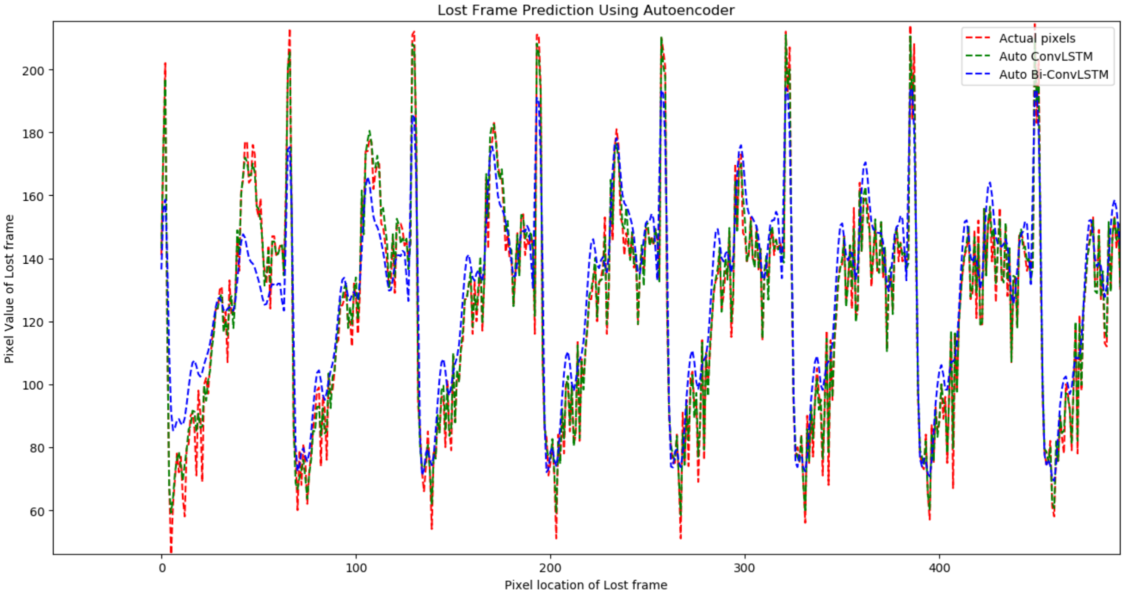


FIGURE 5.15: Graphical comparison of predicted frame using Bi-ConvLSTM ,and ConvLSTM autoencoder with original frame

The intensity value predicted by the ConvLstm Autoencoder model is more comparable to original pixel values than the Bi-ConvLSTM Autoencoder, as seen in the graph. The recovered frames using the ConvLSTM and Bi-ConvLSTM Autoencoder models, as well as the original frame, are shown in Figure 5.16. In comparison to the original frame, the ConvLSTM and Bi-ConvLSTM autoencoder models achieve 31.1dB and 29.23dB, respectively.

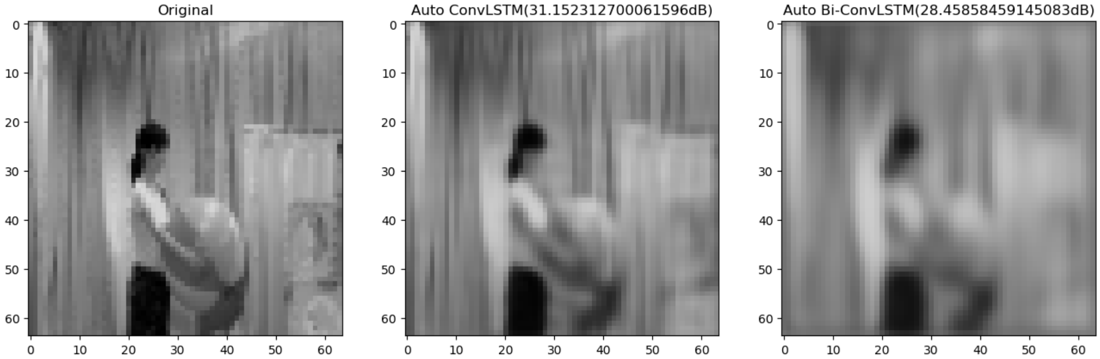

Original
Auto ConvLSTM(31.152312700061596dB)
Auto Bi-ConvLSTM(28.45858459145083dB)
0
10
20
30
40
50
60

FIGURE 5.16: Visual comparison of lost frame reconstruct using Bi-ConvLSTM ,and ConvLSTM autoencoder

5.2.3.3 Comparing other Recovery Method

In this section we compare the performance of the proposed work to other frame concealment methods using different video sequences results.

TABLE 5.7: Comparison of different lost frame recovery methods with different MVV sequences

Sequances	**State of art method** [28]	**Deep learning method** [29]	**Proposed work**
Akko&Kayo	29.73	30.13	35.76
Balloons	32.65	37.81	38.71
Book_Arrival	30.42	34.23	32.27
Break_dance	25.67	27.31	36.52
Bullet	31.00	32.64	35.89
Dog	28.31	36.78	37.11
Lovebird	29.26	30.65	28.68
Newspaper	30.12	31.32	34.055
Rena	29.56	31.57	34.566
Soccer_arc	27.49	30.76	34.37

As can be see in the Table 5.7 , the proposed work obtaining significant gain than to state of art and MPR based lost recovery methods. In terms of average PSNR, the state of the art method for lost frame recovery scores **29.42 dB**, the recent MPR based lost frame recovery method scores **32.32 dB**, and the proposed lost frame recovery method scores **34.79 dB**, then the proposed lost frame recovery method **5.37 dB** and **2.47 dB** difference when compared to the state of the art method and MPR based recover algorithm, respectively.

As the experiment indicates the proposed method achieves high PSNR value than other recovery methods because of the motion and disparity estimation technique is based on SAD we get the correlations of blocks in different frames. The other reason is by using RNN we use the previous recovered block as input to recover the next block in the lost frame.

Finally, in this work we contribute new lost frame detection method 93.12 % detection accuracy with low latency complexity. And we also contribute new MV and DV extraction techniques by using mathematical model for different MVC in MVV transmission. We also enhance the lost frame recovery performance by using different DL algorithms.

Chapter 6

Conclusion and Recommendation

This chapter is organized in two sections. In the first section we give a brief summary of the conclusion of our work. In the second section we give recommendations for future work.

6.1 Conclusion

This proposed lost frame detection and recovery method in MVV transmission used to enhance the video quality of WMSN (low bandwidth network, low resolution camera) at decoder side of MVC.

In this thesis, we develop a new lost frame detection method. from this proposed work result we conclude that:

- By extract different features (properties) between two consecutive frames we achieves well lost frame detection method.
- By applying ANN algorithm and by testing different video frames with lost and without lost it achieves **93.12**% detection accuracy.
- The computational latencies were also measured and this detection method was low computational latencies than other VQM at high resolution videos.

The other task of this thesis was to recover the lost frames of MVV. To develop this method the following steps were implemented :

- From four estimated frames successfully we extract candidate MV and DVs by using new ME and DE technique.
- Based on candidate MV and DVs we were extract blocks from estimated frames and by using these blocks as input to LSTM, Bi-LSTM, and GRU algorithms we recover single blocks in lost frame. By combine those recovered blocks we can reconstruct the lost frame in MVV sequences.
- By using ConvLSTM and Bi-ConvLSTM autoencoder algorithm we can recover the full-frame without extract MV and DV.

The proposed method LSTM based lost frame recovery method was achieves **2.47 dB** average outperforms than the MPR based method. In addition the time complexity was also measured to be faster than the other methods because of the MPR method was recover the lost frames pixel by pixel the proposed method was using block by block. This lost frame detection method have limitations in very-fast and very-slow motion videos with lost or without lost frames in MVV transmission. The lost frame recovery method have some limitations to recover the lost blocks at the border of the lost frames.

6.2 Recommendation

Although we have shown good results can be achieved in both lost frame detection and recovery methods at decoder side of MVC. For further improvement we have the following recommendations for detection and recovery methods.

- By extract, the motion between two consecutive frames (f(v,t-1) and f(v,t)) and add this extracted motion as one feature of the ML model we recommended that design new computer-based subjective video metrics and improve the lost frame detection accuracy method with low complexity.

- To reduce the complexity of lost frame recovery in MVV we recommended that use a deep learning-based motion and disparity estimation technique than other empirical methods.
- Still the lost frame recover needs improvement based on the performance so we recommended that increase the number of feature in RNN algorithms to enhance the scored PSNR in MVV transmission through WSN.

References

[1] Singh AP, Luhach AK, Gao XZ, Kumar S, Roy DS. Evolution of wireless sensor network design from technology centric to user centric: an architectural perspective. International Journal of Distributed Sensor Networks. 2020 Aug;16(8):1550147720949138.

[2] Akyildiz IF, Melodia T, Chowdhury KR. A survey on wireless multimedia sensor networks. Computer networks. 2007 Mar 14;51(4):921-60.

[3] Almalkawi IT, Guerrero Zapata M, Al-Karaki JN, Morillo-Pozo J. Wireless multimedia sensor networks: current trends and future directions. Sensors. 2010 Jul;10(7):6662-717.

[4] Sharif A, Potdar V, Chang E. Wireless multimedia sensor network technology: A survey. In2009 7th IEEE International Conference on Industrial Informatics 2009 Jun 23 (pp. 606-613). IEEE.

[5] Melodia T, Akyildiz IF. Research challenges for wireless multimedia sensor networks. InDistributed video sensor networks 2011 (pp. 233-246). Springer, London.

[6] Abbas, N., Yu, F. and Fan, Y., 2018. Intelligent video surveillance platform for wireless multimedia sensor networks. Applied Sciences, 8(3), p.348.

[7] Colonnese S, Cuomo F, Melodia T. An empirical model of multiview video coding efficiency for wireless multimedia sensor networks. IEEE transactions on multimedia. 2013 Jun 27;15(8):1800-14.

[8] Ibrahim AB, Sadka AH. Error resilience and concealment for multiview video coding. In2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting 2014 Jun 25 (pp. 1-5). IEEE.

[9] Zhang X, Toni L, Frossard P, Zhao Y, Lin C. Adaptive streaming in interactive multiview video systems. IEEE Transactions on Circuits and Systems for Video Technology. 2018 Mar 26;29(4):1130-44.

[10] San X, Cai H, Lou JG, Li J. Multiview image coding based on geometric prediction. IEEE Transactions on Circuits and Systems for Video Technology. 2007 Oct 29;17(11):1536-48.

[11] Kubota, A., Smolic, A., Magnor, M., Tanimoto, M., Chen, T. and Zhang, C., 2007. Multiview imaging and 3DTV. IEEE signal processing magazine, 24(6), pp.10-21.

[12] Ho YS, Oh KJ. Overview of multi-view video coding. In2007 14th International Workshop on Systems, Signals and Image Processing and 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services 2007 Jun 27 (pp. 5-12). IEEE.

[13] Akin A, Capoccia R, Narinx J, Masur J, Schmid A, Leblebici Y. Real-time free viewpoint synthesis using three-camera disparity estimation hardware. In2015 IEEE International Symposium on Circuits and Systems (ISCAS) 2015 May 24 (pp. 2525-2528). IEEE.

[14] Carballeira P, Carmona C, Díaz C, Berjón D, Quesada JC, Moran F, Doblado C, Arnaldo S, del Mar Martin M, Garcia N. FVV Live: A real-time free-viewpoint video system with consumer electronics hardware. IEEE Transactions on Multimedia. 2021 May 14.

[15] Müller K, Schwarz H, Marpe D, Bartnik C, Bosse S, Brust H, Hinz T, Lakshman H, Merkle P, Rhee FH, Tech G. 3D high-efficiency video coding for multi-view video and depth data. IEEE transactions on image processing. 2013 May 23;22(9):3366-78.

[16] Nasri SA. Multiview coding and compression for 3D video (Doctoral dissertation, Université Badji Mokhtar-Annaba (Algérie)).

[17] Zamarin M, Milani S, Zanuttigh P, Cortelazzo GM. A novel multi-view image coding scheme based on view-warping and 3D-DCT. Journal of Visual Communication and Image Representation. 2010 Jul 1;21(5-6):462-73.

[18] Kwon JS, Hwang WY, Choi CY, Chang EY, Hur NH, Kim JW, Kim MB. Multi-View Video Processing: IVR, Graphics Composition, and Viewer. Journal of Broadcast Engineering. 2007;12(4):333-41.

[19] Zhou Y, Xiang W, Wang G. Frame loss concealment for multiview video transmission over wireless multimedia sensor networks. IEEE Sensors Journal. 2014 Oct 31;15(3):1892-901.

[20] Barkowsky M, Wang K, Cousseau R, Brunnström K, Olsson R, Le Callet P. Subjective quality assessment of error concealment strategies for 3DTV in the presence of asymmetric transmission errors. In2010 18th International Packet Video Workshop 2010 Dec 13 (pp. 193-200). IEEE.

[21] Akramullah S. Video Quality Metrics. InDigital Video Concepts, Methods, and Metrics 2014 (pp. 101-160). Apress, Berkeley, CA.

[22] Seshadrinathan K, Soundararajan R, Bovik AC, Cormack LK. Study of subjective and objective quality assessment of video. IEEE transactions on Image Processing. 2010 Feb 2;19(6):1427-41.

[23] Chikkerur S, Sundaram V, Reisslein M, Karam LJ. Objective video quality assessment methods: A classification, review, and performance comparison. IEEE transactions on broadcasting. 2011 Feb 10;57(2):165-82.

[24] Schmidt J, Marques MR, Botti S, Marques MA. Recent advances and applications of machine learning in solid-state materials science. npj Computational Materials. 2019 Aug 8;5(1):1-36.

[25] Alpaydin E. Introduction to machine learning. MIT press; 2020 Mar 17.

[26] Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E. Deep learning applications and challenges in big data analytics. Journal of big data. 2015 Dec;2(1):1-21.

[27] LeCun Y, Bengio Y, Hinton G. Deep learning. nature. 2015 May;521(7553):436-44.

[28] Zhou Y, Xiang W, Wang G. Frame loss concealment for multiview video transmission over wireless multimedia sensor networks. IEEE Sensors Journal. 2014 Oct 31;15(3):1892-901.

[29] Lin TL, Tseng HW, Wen Y, Lai FW, Lin CH, Wang CJ. Reconstruction algorithm for lost frame of multiview videos in wireless multimedia sensor network based on deep learning multilayer perceptron regression. IEEE Sensors Journal. 2018 Aug 17;18(23):9792-801.

[30] Eiben AE, Jelasity M. A critical note on experimental research methodology in EC. InProceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600) 2002 May 12 (Vol. 1, pp. 582-587). IEEE.

[31] Barkowsky M, Pinson M, Pépion R, Le Callet P. Analysis of freely available dataset for HDTV including coding and transmission distortions. InFifth International Workshop on Video Processing and Quality Metrics (VPQM) 2010 Jan 13.

[32] Sp.cs.tut.fi. 2021. MOBILE.3DTV - Stereo-video database. [ONLINE] Available at: http://sp.cs.tut.fi/mobile3dtv/stereo-video/. [Accessed 27 February 2021].

[33] Fujii.nuee.nagoya-u.ac.jp. 2021. Fujii Lab's sequences Download lists. [ONLINE] Available at: https://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/. [Accessed 27 February 2021].

[34] McKinney W. Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. " O'Reilly Media, Inc."; 2012 Oct 8.

[35] Brownlee J. Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras. Machine Learning Mastery; 2016 May 13.

[36] Zeng H, Wang X, Cai C, Chen J, Zhang Y. Fast multiview video coding using adaptive prediction structure and hierarchical mode decision. IEEE Transactions on Circuits and Systems for Video Technology. 2014 Mar 6;24(9):1566-78.

[37] Chen Y, Wang YK, Ugur K, Hannuksela MM, Lainema J, Gabbouj M. The emerging MVC standard for 3D video services. EURASIP Journal on Advances in Signal Processing. 2008 Dec;2009:1-3.

[38] Chen Y, Hannuksela MM, Suzuki T, Hattori S. Overview of the MVC+ D 3D video coding standard. Journal of Visual Communication and Image Representation. 2014 May 1;25(4):679-88.

[39] Vetro A, Wiegand T, Sullivan GJ. Overview of the stereo and multiview video coding extensions of the H. 264/MPEG-4 AVC standard. Proceedings of the IEEE. 2011 Jan 31;99(4):626-42.

[40] Rahimunnisha S, SudhaVani G. Multi-View Video Coding Algorithms/Techniques: A Comprehensive Study.

[41] Krebs J, Delingette H, Ayache N, Mansi T. Learning a Generative Motion Model from Image Sequences based on a Latent Motion Matrix. IEEE Transactions on Medical Imaging. 2021 Feb 2;40(5):1405-16.

[42] Yang L, Yu K, Li J, Li S. An effective variable block-size early termination algorithm for H. 264 video coding. IEEE Transactions on Circuits and Systems for Video Technology. 2005 May 31;15(6):784-8.

[43] Umar A. 3D multiple description coding for error resilience over wireless networks (Doctoral dissertation, Brunel University School of Engineering and Design PhD Theses).

[44] Feamster N, Balakrishnan H. Packet loss recovery for streaming video. In12th International Packet Video Workshop 2002 Apr 24 (pp. 9-16). PA: Pittsburgh.

[45] Guo X, Lu Y, Wu F, Zhao D, Gao W. Wyner–Ziv-based multiview video coding. IEEE Transactions on circuits and systems for video technology. 2008 May 28;18(6):713-24.

[46] Naderi MY, Rabiee HR, Khansari M, Salehi M. Error control for multimedia communications in wireless sensor networks: A comparative performance analysis. Ad Hoc Networks. 2012 Aug 1;10(6):1028-42.

[47] Wang Y, Zhu QF. Error control and concealment for video communication: A review. Proceedings of the IEEE. 1998 May;86(5):974-97.

[48] Micallef BW, Debono CJ, Farrugia RA. Error concealment techniques for H. 264/MVC encoded sequences.

[49] Walczak S. Artificial neural networks. InAdvanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-Computer Interaction 2019 (pp. 40-53). IGI Global.

[50] Sharma S, Sharma S. Activation functions in neural networks. Towards Data Science. 2017 Sep;6(12):310-6.

[51] Jain AK, Mao J, Mohiuddin KM. Artificial neural networks: A tutorial. Computer. 1996 Mar;29(3):31-44.

[52] Cavallaro L, Bagdasar O, De Meo P, Fiumara G, Liotta A. Artificial neural networks training acceleration through network science strategies. Soft Computing. 2020 Dec;24(23):17787-95.

[53] Jozefowicz R, Zaremba W, Sutskever I. An empirical exploration of recurrent network architectures. InInternational conference on machine learning 2015 Jun 1 (pp. 2342-2350). PMLR.

[54] Salehinejad H, Sankar S, Barfett J, Colak E, Valaee S. Recent advances in recurrent neural networks. arXiv preprint arXiv:1801.01078. 2017 Dec 29.

[55] Gilardoni G. Recurrent neural network models for financial distress prediction.

[56] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15;9(8):1735-80.

[57] Li Y. Prediction of energy consumption: Variable regression or time series? A case in China. Energy Science & Engineering. 2019 Dec;7(6):2510-8.

[58] Yang M, Yang Y, Cosman P. Depth-assisted error concealment for intra frame slices in 3D video. In2012 19th IEEE International Conference on Image Processing 2012 Sep 1 (pp. 1281-1284). IEEE.

[59] Liu Y, Wang J, Zhang H. Depth image-based temporal error concealment for 3-D video transmission. IEEE Transactions on Circuits and Systems for Video Technology. 2009 Nov 3;20(4):600-4.

[60] Yan B. A novel H. 264 based motion vector recovery method for 3D video transmission. IEEE Transactions on Consumer Electronics. 2007 Nov;53(4):1546-52.

[61] Tai SC, Wang CC, Hong CS, Luo YC. An effiicient full frame algorithm for object-based error concealment in 3D depth-based video. Multimedia tools and applications. 2016 Aug;75(16):9927-47.

[62] Yan B, Zhou J. Efficient frame concealment for depth image-based 3-D video transmission. IEEE Transactions on Multimedia. 2012 Feb 7;14(3):936-41.

[63] Yang M, Lan X, Zheng N, Cosman P. Depth-assisted temporal error concealment for intra frame slices in 3-D video. IEEE transactions on broadcasting. 2014 May 29;60(2):385-93.

[64] Dominguez HD, Villegas OO, Sanchez VG, Casas ED, Rao KR. The H. 264 video coding standard. IEEE Potentials. 2014 Mar 4;33(2):32-8.

[65] Marins HR, Estrela VV. On the use of motion vectors for 2D and 3D error concealment in H. 264/AVC video. InResearch Anthology on Recent Trends,

Tools, and Implications of Computer Programming 2021 (pp. 765-787). IGI Global.

[66] Chung TY, Sull S, Kim CS. Frame loss concealment for stereoscopic video based on inter-view similarity of motion and intensity difference. In2010 IEEE International Conference on Image Processing 2010 Sep 26 (pp. 441-444). IEEE.

[67] Chung TY, Sull S, Kim CS. Frame loss concealment for stereoscopic video based on inter-view similarity of motion and intensity difference. In2010 IEEE International Conference on Image Processing 2010 Sep 26 (pp. 441-444). IEEE.

[68] Hewage CT, Worrall S, Dogan S, Kondoz AM. Frame concealment algorithm for stereoscopic video using motion vector sharing. In2008 IEEE international conference on multimedia and expo 2008 Jun 23 (pp. 485-488). IEEE.

[69] Yan B, Zhou J. Efficient frame concealment for depth image-based 3-D video transmission. IEEE Transactions on Multimedia. 2012 Feb 7;14(3):936-41.

[70] Liu W, Ma L, Qiu B, Cui M, Ding J. An efficient depth map preprocessing method based on structure-aided domain transform smoothing for 3D view generation. PloS one. 2017 Apr 13;12(4):e0175910.

[71] Misra S, Reisslein M, Xue G. A survey of multimedia streaming in wireless sensor networks. IEEE communications surveys & tutorials. 2008 Oct 10;10(4):18-39.

[72] Radhika S, Anitha K, Sabitha R. Applications of Deep Learning: A Review. Annals of the Romanian Society for Cell Biology. 2021 Jan 31:1927-34.

[73] Cui Q, Wu S, Liu Q, Zhong W, Wang L. MV-RNN: A multi-view recurrent neural network for sequential recommendation. IEEE Transactions on Knowledge and Data Engineering. 2018 Nov 14;32(2):317-31.

[74] Akramullah S. Video Quality Metrics. InDigital Video Concepts, Methods, and Metrics 2014 (pp. 101-160). Apress, Berkeley, CA.

[75] Instruments N. Peak signal-to-noise ratio as an image quality metric.

[76] Wang Z, Bovik AC, Sheikh HR. Structural similarity based image quality assessment. InDigital Video image quality and perceptual coding 2017 Dec 19 (pp. 225-242). CRC Press.

[77] Oguz SH, Faibish A, Faibish S, Cotter G. Objective image quality metrics for dct-based video compression. In36th SMPTE Annual Advanced Motion Imaging Conference 2002 Feb 7 (pp. 1-13). SMPTE.

[78] Yang G, Li D, Lu F, Liao Y, Yang W. RVSIM: a feature similarity method for full-reference image quality assessment. EURASIP Journal on Image and Video Processing. 2018 Dec;2018(1):1-5.

[79] Zhang L, Zhang L, Mou X, Zhang D. FSIM: A feature similarity index for image quality assessment. IEEE transactions on Image Processing. 2011 Jan 31;20(8):2378-86.

[80] Zhang L, Zhang L, Zhang D, Guo Z. Phase congruency induced local features for finger-knuckle-print recognition. Pattern Recognition. 2012 Jul 1;45(7):2522-31.

[81] Liu L, Hua Y, Zhao Q, Huang H, Bovik AC. Blind image quality assessment by relative gradient statistics and adaboosting neural network. Signal Processing: Image Communication. 2016 Jan 1;40:1-5.

[82] Sheikh HR, Bovik AC. Image information and visual quality. IEEE Transactions on image processing. 2006 Jan 16;15(2):430-44.

[83] Learned-Miller EG. Entropy and mutual information. Department of Computer Science, University of Massachusetts, Amherst. 2013 Sep 16.

[84] Luque A, Carrasco A, Martín A, de las Heras A. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition. 2019 Jul 1;91:216-31.

[85] Alaloul WS, Qureshi AH. Data processing using artificial neural network. InDynamic Data Assimilation-Beating the Uncertainties 2020 May 26. IntechOpen.

[86] Metkar S, Talbar S. Motion estimation techniques for digital video coding. New York: Springer; 2013 Mar 15.

[87] Jakubowski M, Pastuszak G. Block-based motion estimation algorithms—a survey. Opto-Electronics Review. 2013 Mar 1;21(1):86-102.

[88] Pan Z, Zhang Y, Kwong S. Efficient motion and disparity estimation optimization for low complexity multiview video coding. IEEE Transactions on Broadcasting. 2015 Apr 27;61(2):166-76.

[89] Nazerfard E, Atashgahy Z, Nadali A. Abnormal Activity Detection for the Elderly People using ConvLSTM Autoencoder.

[90] Xu Y, Gao L, Tian K, Zhou S, Sun H. Non-local convlstm for video compression artifact reduction. InProceedings of the IEEE/CVF International Conference on Computer Vision 2019 (pp. 7043-7052).

[91] Matulin M, Mrvelj Š. Modelling user quality of experience from objective and subjective data sets using fuzzy logic. Multimedia Systems. 2018 Nov;24(6):645-67.