Efficient Classification of Motion in Video Data by Using Deep Neural Networks

Author

Muntaha Irfan

FALL 2018-MS-18(CE) 00000278499

MS-18 (CE)

Supervisor

Dr. Farhan Hussain

DEPARTMENT OF COMPUTER ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

DEC, 2021

Efficient Classification of Motion in Video Data by using Deep Neural Networks

Author

Muntaha Irfan

FALL 2018-MS-18(CE) 00000278499

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Computer Engineering

Thesis Supervisor

Dr. Farhan Hussain

Thesis Supervisor's Signature: _____

DEPARTMENT OF COMPUTER ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

DEC, 2021

DECLARATION

I hereby certify that I have created this thesis titled as "*Efficient Classification of Motion in Video Data by Using Deep Neural Network*" completely on the basis of my personal efforts under the supervision of Dr. Farhan Hussain. This work has not been presented elsewhere for assessment. The material that has been used from other sources has been properly acknowledged / referred. No part of the work submitted in this thesis was published in assistance of any other degree or any institute of learning or university.

_____

Signature of Student

Muntaha Irfan

FALL 2018-MS-18(CE) 00000278499

LANGUAGE CORRECTNESS CERTIFICATE

This thesis has been checked by an English expert and is error-free in terms of syntax, typing, grammatical structure and spelling. The thesis is also formatted in accordance with the University's guidelines for MS thesis work.

_____

Signature of Student

FALL 2018 MS-18(CE)00000278499

_____

Signature of Supervisor

Dr. Farhan Hussain

COPYRIGHT STATEMENT

ACKNOWLEDGEMENTS

All praise and glory to ALMIGHTY ALLAH (the most glorified, the most merciful) Who gave me the courage, patience, knowledge, and ability to carry out this work and to complete it satisfactorily. Without His blessings I wouldn't have achieved this accomplishment. Whoever guided me throughout the course of my research was Allah's will. So, no one is worthy of admiration except Allah. Undoubtedly, HE eased my way and gave me an opportunity to use my knowledge in best way and encouraged me to carry out this work

I would like to express my sincere appreciation to my thesis supervisor, ***Dr. Farhan Hussain*** for boosting my morale and for his vital guidance, motivation, dedication and appreciation throughout this work. My deepest gratitude goes to him for showing endless faith in me. I am blessed to have such a co-operative advisor and kind mentor for my research. I am much esteemed to have worked under his supervision.

Along with my advisor, I am also highly obliged and grateful to my entire thesis committee members: ***Dr. Arslan Shaukat*** and ***Dr. Ali Hassan*** for their cooperation, valuable feedback, suggestions, and guidance in this research work.

My acknowledgement would remain incomplete unless I thanked the most important source of my solidity, my family. I am also grateful to my parents, who brought me up when I was unable to walk, and they have actively supported me in every part of life. I am also thankful to my loving sisters and brothers, who have been by my side in every situation. And at last, I'd like to appreciate all of my friends and people who have helped me throughout my research

In dedication

*To my **father Irfan Saleem**:*

for encouraging and supporting me to achieve this daunting task.

*To my **mother Bushra Irfan**:*

for making me who I am

ABSTRACT

Video has become more popular in many applications in recent years due to increased storage capacity, more advanced network architectures, as well as easy access to digital cameras, especially in mobile phones. Classification of the type of motion in a video sequence is an area targeted by many researchers for the purpose of traffic control, video scene classification, event prediction, sport analysis, management of web videos etc. There are several conventional and unconventional techniques for motion classification in videos but due to the advent of sophisticated algorithms and high computational capabilities deep learning architectures are utilized for almost every image/video processing task including motion classification. Deep learning methodology is more reliable and effective than other approaches. Training a deep learning architecture for motion classification requires that all of the frames (pixel by pixel) are fed to the network along with their corresponding label and once the network learns the classification task, we can use it for inference purpose. However, this method requires a lot of memory and computational resources as large amount of data (all the frames in a video) needs to be processed by the architecture. We aim to reduce the amount of data to be processed by the deep learning architecture for motion classification task this subsequently results in low memory requirements and reduced computational complexity. At the same time, we strive for maintaining the classification accuracy. A video is a sequence of individual frames hence there exists a lot of temporal redundancy between consecutive frames. This redundancy can be exploited by traditional motion estimation which gives us awareness about the motion information in a video sequence. If instead of inputting the standard video frames to the deep learning architectures, we feed them the motion information so that our architectures have to process much less amount of information for the motion classification task. In our work the motion information in a video sequence is retrieved by using the three-step search which is a block matching algorithm. This algorithm gives us the motion vectors which contain the motion information in a video sequence and hence we train our network on these motion vectors instead of the standard frames to achieve motion classification task. Experimental results show that by employing our proposed method the motion classification task can be carried out by processing much less amount of information while maintain good accuracies.

Keywords: Block Matching Motion Estimation, Deep Neural Network, Motion Vector, Motion Estimation

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1.
INTRODUCTION

This section provides a detail introduction about the important concepts related to our research, the current problem, and an overview of our solution. It is arranged into five sub sections. Background study is described in **Section 1.1**, Problem statement of research is presented in **Section 1.2**, Section **1.3** explains the applications related to our research. In **Section 1.4** we have discussed Aims and Objectives and structure of thesis is described in **Section 1.5.**

1.1 Background study

Video Classification is the task of producing a label that is relevant to the video given its frames while Video Motion Classification is the task of producing a label that is relevant to the motion in videos given their frames. Many examples related to video motion classification include classification of sports such as baseball, football etc. or classification of a traffic such as speedy, or jammed or slow etc. One of the key inspirations, which fascinates researchers to work in video classification, is the huge domain of its applications in surveillance videos, human computer interaction, robotics, video games for player characters, human activity recognition, automated databases of videos and management of web videos. Deep learning for video classification is an emerging and vibrant field. The video classification has extremely evolved by deep neural networks, and they present outstanding performance in terms of video analysis. They have been greatly employed for image recognition challenges and achieved state-of-the-art results in video processing, detection, retrieval, recognition and segmentation. DNN equally learns feature relationships and class relationships. It simultaneously carries out video classification within the similar framework by exploiting the learned relationships. Now-a-days, technology has revolutionized the world and aimed at automating processes. It has reduced the manual work and has provided ways for benefiting mankind in every field of life. In the past years, it has tried to automate things by enabling systems to follow a set of commands and perform according to the instructions programmed by a programmer. Lately a concept of artificial intelligence is originated whose purpose is to mimic the learning

capabilities of human brain. Technology vendors are trying to create intelligent systems that are able to learn, modify and possibly perform autonomously rather than executing the instructions already defined by a programmer. Deep Neural Networks are major breakthrough in machine learning till now. They have been clearly applied in many real-world applications such as autonomous vehicles, science, games and art. Deep learning has led to groundbreaking innovations in different disciplines such as computer vision, Nature Language Processing (NLP), and speech processing. Motivated by the great success of the deep learning processes in analyzing image, audio and text data, considerable attempts are just being dedicated to the design of deep nets for video classification. Among the many practical needs, classifying videos (or video clips) is useful in many applications. In this work, we are targeting Video Motion Classification Using Deep Neural Networks. Specifically, we are making our system efficient by utilizing the motion information for training our deep learning architectures instead of pixel frames. The naïve video motion classification method by deep learning models involves feeding all the pixels in a frame and all the frames in the video to the network and then the corresponding label with the highest probability is chosen. This involves both high processing power as well as processing of high volumes of data. With video we can usually presume that successive frames in a video are associated with respect to their semantic contents. If we can get benefit of the temporal nature of videos, we can enhance our real video motion classification results and computational intensity. Our aim is to classify motion in video data by using motion vectors and Deep Neural Networks. The idea of extracting motion vectors from video frames was taken from data compression technique [3]. Image data in video frames is largely consistent across motion trajectories. It means that the scene content does not change substantially from one frame to another. To reduce redundant information from image sequences, one must first estimate motion in video frames so that motion trajectories may be processed. If we utilize these motion trajectories for motion classification via deep learning networks, it will result in reduced computational complexity as well as low processing power for huge amount of data.

1.2 Problem Statement

Mostly motion classification in videos is implemented by exploiting all the frames in a video pixel-wise. The classification of each frame is performed individually, and the video is labeled according to the majority voting of individual frame labels. There is very little or no work regarding processing of motion vectors by deep neural networks for video motion classification. By using motion vectors of a video, we can minimize the number of computational requirements to determine the motion information in a video sequence. The deep learning architectures have to deal with less amount of information by utilizing the motion vectors. Hence, we can retrieve accurate motion classification information in a video sequence by processing motion vectors instead of complete video frames. Deep learning has been successful in processing image, speech, and text data during the last decade. We aim to use deep neural networks for processing the motion vectors to retrieve accurate motion information in a video sequence.

1.3 Applications

Motion classification is important in many research areas with several applications in the field of computer vision and image processing. Several applications include online checking of assembly process, surveillance videos, robotics, human computer interaction, sports analysis, video games for player characters, human activity recognition, automated databases of videos, automatic categorizing, searching, indexing, segmentation, and retrieval of videos, gesture control and management of web videos. Live streaming prediction, violence detection, character recognition, traffic control, social media analysis, emotion analysis, movie review, violence detection from video of real time game, video scene classification, event prediction, animation movie video classification, sport player action recognition, twitter video classification, stock market prediction and movie video trailer classification are also the applications of video classification.

1.4 Aims and Objectives

Aims and objectives of the research are as follows:

- To find out the motion vectors in a video sequence by an efficient three step search method
- To utilize these motion vectors to classify the corresponding motion in video sequences with the help of Deep Neural Network.

1.5 Structure of Thesis

This work is structured as follows:

Chapter 1: offers a brief introduction containing the background study, problem statement, applications, aims and objectives and structure of thesis.

Chapter 2: covers the detailed introduction about Motion Estimation, Block Matching Motion Estimation, types of block Matching Motion Estimation and Evaluation criteria.

Chapter 3: provides the detailed literature review highlighting the work done in the Domain of Video Motion Detection, Block matching motion estimation and Deep Neural Network. It presents the detailed systematic review of current techniques that can be used for classifying motion in video data and research already done in this area. It discusses different types of methods used for Block Matching Motion Estimation.

Chapter 4: covers the details of proposed methodology used for identification and solving the problem in hand. It explains the complete methodology adopted to solve the problem highlighted in problem statement and to achieve the objectives stated previously.

Chapter 5: All the experimental results are discussed in detail with all desired figures and tables. It presents the detailed implementation of our framework and architecture.

Chapter 6: concludes the research and recommends a future work that can be done in order to further extend this research.

CHAPTER 2.
MOTION ESTIMATION AND MOTION VECTORS

Motion is regarded as valuable information source in many video sequences. Camera movements and 3D-video scenes causes motion of moving objects. The apparent motion captures the spatiotemporal variations in pixel intensities that result in subsequent images of video frames. Motion estimation approaches analyze the visual content to extract this information. In the fields related to image sequence analysis, video communication, and computer vision, efficient and accurate estimation of motion is critical [6]. Motion estimation methods in picture sequence analysis and computer vision aim to accurately and authentically replicate the movements in the scene. Efficient estimation is pivotal especially in video coding applications like streaming media, TV broadcasting, digital movies, videoconferencing, Blu-ray, and DVD for compression purposes. Motion estimation algorithms are used to reduce the amount of storage space needed for data retention [7]. It lowers the transmission requirements of streaming digital video sequences. Motion vectors are used in compression for video frames like H.264 to reduce transmission and storage requirements. When encoding video, Motion Vectors are calculated by matching same blocks between two consecutive frames during the motion estimation process. Motion vectors are utilized to store the video information once it has been encoded [8]. The primary goal of motion estimation is to enhance compression efficiency rather than to locate the 'actual' motion in the image sequences. The current research trend is to use block matching motion estimation to extract motion vectors and to classify them using DNN. To increase coding efficiency, video classification takes advantage of the high correlation between subsequent frames which is achieved through estimation and compensation techniques. Many strategies for improving the computational complexity of ME algorithms have been proposed. The strategies can be block matching algorithms, parametric model, optical flow, and pel recursive methods [9]. BMA appears to be the most common method among these. It is an effective and simple approach both in software and hardware implementations. This motion estimation can be conducted pixel-by-pixel, region-by-region, or block-by-block.

Pixel-based: In pixel-based approach each and every pixel of image is assigned a value of motion vector for creating dense motion vector field. Motion vectors provide a detailed description of motion in pixel-based method [10]. However, in video processing applications it requires a lot of computational complexities to process motion information. Pixel-based has the drawback of relying on a threshold value where changes appear from pixel to pixel.

Region-based: Region-based model is employed to a portion of the image that has a coherent motion pattern. Moving things in the scene must be identified in this situation. In larger computations, region-based computations have issues.

Block-based: In block-based approach, which is a special case of region-based approach, the image is divided into blocks. For smaller blocks, the hypothesis that the block is traveling coherently is possibly correct. Block partitioning has another benefit that it doesn't need any extra details to indicate the region's shape. Block matching algorithms are primarily motion estimating systems. These are straightforward, robust, and commonly utilized to compress videos and to classify for identification of best match [11]. The goal of block matching is to reduce a measure of dissimilarity between two frames. Furthermore, these methods will be resistant to noise, which is an added advantage as compared to the other methods like global optimization and pel-recursive algorithms that provide dense motion fields. The following is the model to obtain motion vectors

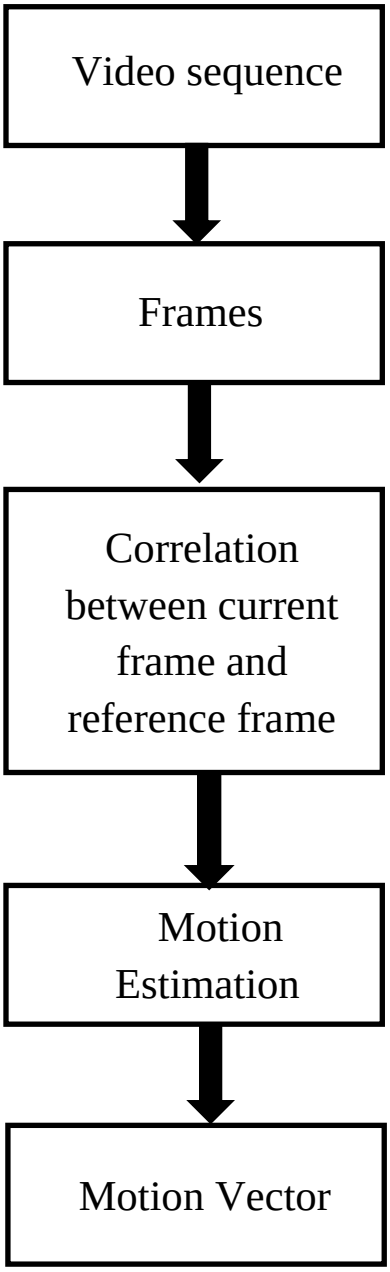

Figure 2-1: Flowchart of Motion Estimation

2.1 Block Matching Motion Estimation

Block-matching for motion estimation (BMME), a popular and successful method for reducing temporal redundancy, has been implemented in several video processing and video coding techniques, including H.261, H.265, and H.264. It is also employed in various motion-compensated video coding methodologies [12]. As a result, a quick yet precise search technique that is block-based, is extremely popular to achieve a significant reduction in processing time while keeping good, reconstructed image quality. Motion estimation using block matching (BM) plays a key role in motion detection, encoding, and video compression. An approach for identifying motion in a video sequence is known as block matching. Because it requires less calculation and is less complicated than other algorithms, the Block Matching Algorithm is an excellent choice. The high degree of similarity between each neighboring pixel is the most important component of block matching motion estimation [13]. As a result, applying a motion vector to a group of pixels is more beneficial than assigning one to a single pixel. Block determination, search method, and matching criteria are three basic components of BMA. The first component of BMA is block determination. It is responsible for explicitly describing the point in the reference frame from where the search starts, as well as the position and size of the block in the present frame. The second component is the search method, which establishes the target in the reference frame which is sought for suitable blocks. The third component is the matching criteria which is used for finding the best match between two blocks [9]. For matching the blocks, the reference frame is broken up into ‘macro blocks. Then the macro blocks of two frames (current and reference) are compared. After that a motion vector is constructed that denotes the motion of a macro block from one frame to another. P pixels all around the relevant macro block are included in the search region for finding the best match. For larger motions, a larger p (search parameter) is necessary, and as the p grows larger (search parameter), the motion estimating procedure becomes computationally more costly. In block matching technique we divide the present frame into blocks (NxN pixels) that are not overlapped with each other. We find the best match block within the search window of previous frame. Search for best-matched block is done to minimize the sum of absolute differences (SAD) [14]. The Motion Vector (MV) is the difference in position between two blocks: a block

in the present frame and the best-matched block in the preceding frame (Fig. 1). There are up to seven different block size modes (BSM) to choose from: 16 x 16, 4 x 4, 8 x 4, 8 x 8, 8 x 16,

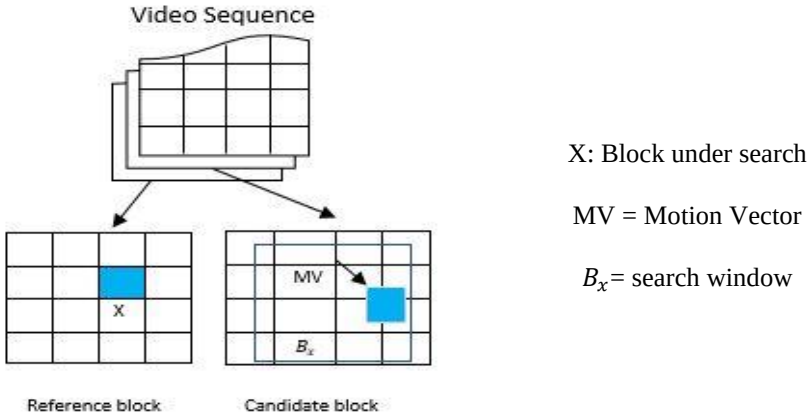

Figure 2-2: Representation of Block Matching Motion Estimation

4 x 8, and 16 x 8. A motion vector is not required for each pixel of image rather a single motion vector is sufficient for each block of pixels. In general, block matching implies locating a candidate block in a reference frame. That candidate block must match well with the current block within a search area. This technique can be thought of as an optimization problem. A reference and a candidate block are compared pixel wise utilizing evaluation criteria (distance) that will be discussed later.

2.1.1 Search Window and Search Strategy

Within an appropriate collection of candidate vectors, the motion vector is chosen. This set is known as the search window, and it indicates the area that will be searched for finding the block that matches well with previous frame. The search window is mainly the rectangular region at the center of the block of the reference image in the most usual situation. Figure 2.3 shows an illustration of search window [8]. The complexity and accuracy of the motion estimating algorithm are significantly impacted by the search window structure. As a result, a motion estimation algorithm's search window and related search strategy selections are crucial.

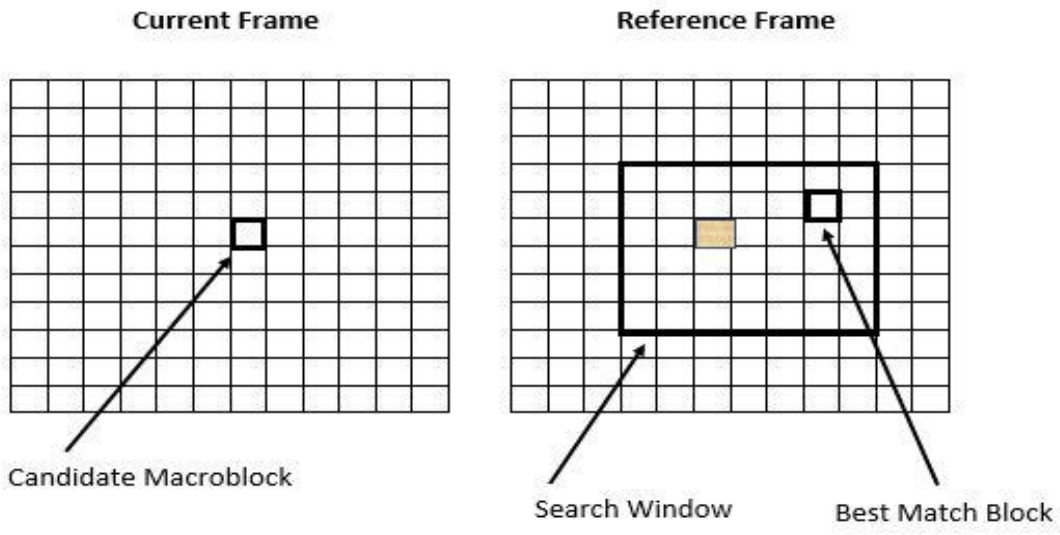

Figure 2-3: Representation of search window

2.2 Types of Block Matching Motion Estimation

There are several Block matching motion estimation methods including Full Search or Exhaustive search, three-step search, four-step search, diamond search, Simple and efficient search, Adaptive Rood Pattern search [15] etc. The idea is to come up with the motion vectors in the most efficient manner possible. The optical flow method can also be adopted to detect the moving object. This method computes the optical flow field and then grouping of the optical flow distribution properties is done. Optical flow, in contrast to previous approaches, may be used to identify moving objects in backgrounds that are not static and provide movement information completely [16]. However, it is inappropriate for real-time demanding scenarios due to some limitations such as very high computational complexity, high sensitivity to noise, and poor performance in noisy surroundings. The following are some approaches for estimating block matching motion:

2.2.1 Full Search or Exhaustive Search:

ES algorithm is also termed as Full Search (raster scan) is the simplest block matching approach available, but it has a high computational cost. This algorithm guarantees finding the motion vector accurately. It achieves this accuracy by computing SAD values. This algorithm undergoes an exhaustive search across all feasible blocks within the search window. It, then, comes up with the optimal solution in terms of prediction quality. The FS algorithm does an exhaustive search of the search area's available points. The cost function is calculated by this algorithm at each potential location within the search area [17]. As a consequence, among all block matching algorithms, it finds the match with best similarity and produces the highest PSNR. One benefit of full search method is that it can determine the most precise motion vector. It is the simplest yet offers the best estimation method with the least amount of matching error. The best match is found with the maximum similarity or lowest dissimilarity, and strong searching accuracy providing the optimal search is achieved. It is, however, inappropriate for real-time video coding as it takes a longer time for computations [18].

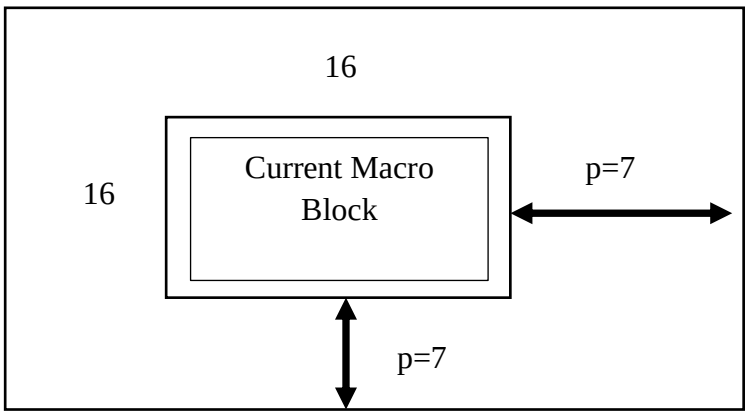

Figure 2-4: Block Matching a macro block of side 16 pixels

2.2.2 Three Step Search:

The goal of three-step search algorithm, as the name implies, is to achieve uniform complexity, which regardless of the motion activity, corresponds to 3 iterations. It uses rectangular search patterns of various sizes. This algorithm is both reliable and easy [9]. It uses the coarse-to-fine search approach and outputs motion vectors. The steps of the algorithm begin with specifying the window size search area that best match. At suitable step size distance, plots eight points around the central point and chooses between them for comparisons. Next, if the minimum cost point, i.e., BDM, is identified at any of the nine sites in the previous step, the step size is divided, and the center is changed to that position for the third step. In the third step, we repeat the previous step until the window's step size is less than one. The TSS needs to use 25 search/checkpoints [8]. With only 25 criteria computations, the TSS algorithm can discover displacements of up to ±7 pixels in both directions (at the previous step, the value of J has been calculated for the center of the search window, nine for the first step, and eight for both second and third steps).

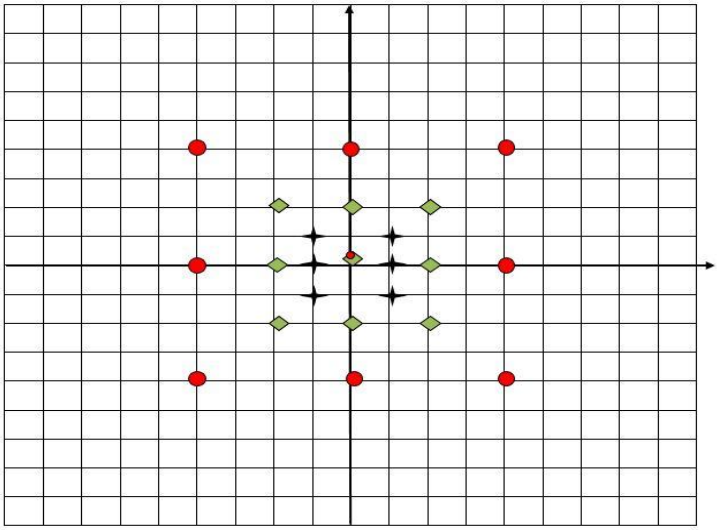

Figure 2-5: TSS search model

2.2.3 Four Step Search:

In the first step of the four-step search, a 5x5 window is used. Unlike the NTSS, it does not use 9x9 window. FSS employs the model of search shown in fig. 2.6 and fig 2.7 in the next two steps.

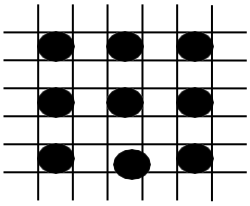

Figure 2-6: A model of search 1st step

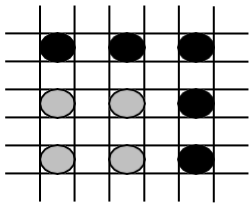

Figure 2-7: A model of search 2nd/3rd step

FSS performs the same as TSS for the fourth step. However, the number of checking points don't exceed 27

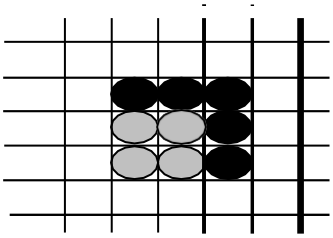

Figure 2-8: Four step search model in FSS

2.2.4 Diamond Search:

DS algorithms employ two different patterns: large diamond shape search pattern (LDSP). This pattern comprises nine checking points, eight of which are plotted around the central point. The other pattern is the small diamond shape search pattern (SDSP) [14]. This pattern consists of five check points. The algorithm works as follows: firstly, LDSP is plotted on the search region, and all the nine checking points of LDSP are checked. Move to step 3 if central point appears with the minimum cost, otherwise, move to step 2. If the minimum BDM point in the subsequent step is attained at a place other than the center of the LDSP, build a new LDSP pattern in step 2 [17]. Proceed to step 3 if the middle position point appears to be with the minimal cost; otherwise, repeat step 2. If the minimal cost point is located in the center of the

LDSP pattern, switch to the SDSP pattern in step 3. Step 3 yields the lowest cost point, which is then used to calculate MV. In the best-case scenario, DS requires 13 searches/checkpoints. In terms of MSE, the DS algorithm outperformed 4SS and block-based gradient descent search (BBGDS), however, it required more searching/checking locations. The search models are demonstrated.

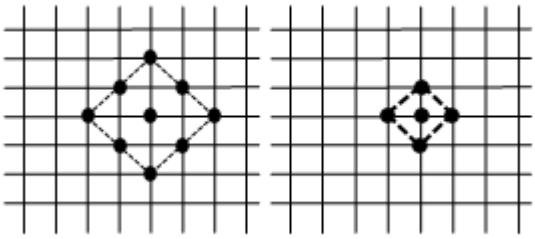

Figure 2-9: Search model in large and small diamond pattern

2.2.5 Adaptive Rood Pattern Search (ARPS)

In this technique a frame's general motion is coherent most of the time. It emphasizes that the macro blocks that surround each macro block usually travel in the same direction. That's why it is highly likely that neighboring macro blocks will have comparable motion vectors most of the time [19].

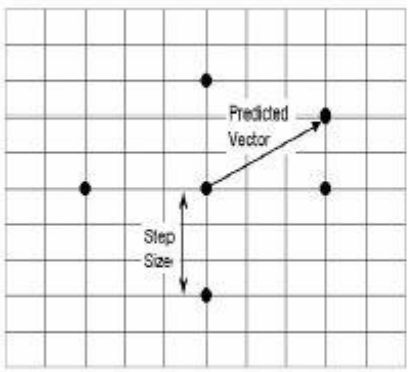

Figure 2-10: The predicted motion vector is (3,2) and the step size is 3

2.3 Evaluation Criteria

Distortion metrics are used to measure the homogeneity of the present macroblock with predicted reference block [12]. Visual quality observation is difficult to reliably assess because it is mostly subjective, and fidelity evaluations might differ greatly due to individual perceptions of content matter. When portions of an image are evaluated at the pixel level for image matching, block matching is used to increase efficiency. The purpose of image matching is to figure out how similar two images or sections of images are. The similarity measure, often known as the correlation measure, is an important part of the matching process [20]. A proficient method in block matching is to find the minimum difference or matching error rather than the maximum similarity or correlation. The employment of sophisticated search algorithms to reduce computation time is an important feature of block matching. The output of a cost function is used to match macro block with one another. In the entire motion estimation process, different distortion metrics are applied to determine the best fit for a specified macro block [21]. Literature discusses several approaches to compare the reference and a candidate block. The mean absolute error (MAE), mean squared error (MSE), the sum of squared differences (SSD), the sum of absolute differences (SAD), and PSNR (peak signal to noise ratio) are the measures for determining algorithm's efficiency [3].

The formulas are given below:

$$\text{MAE} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} |B(i,j) - A(i,j)| \qquad 2.1$$

$$\text{MSE} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (B(i,j) - A(i,j))^2 \qquad 2.2$$

$$\text{SSD} = \sum_{i=1}^{N} \sum_{j=1}^{N} (B(i,j) - A(i,j))^2 \qquad 2.3$$

$$\text{SAD} = \sum_{i=1}^{N} \sum_{j=1}^{N} |B(i,j) - A(i,j)| \qquad 2.4$$

$$\text{PSNR} = 10\ log_{10} \frac{(peak\ to\ peak\ value\ of\ data)^2}{MSE} \qquad 2.5$$

Where N is the block length in pixels, B (i, j) is the value of the pixel in the candidate block at position (i,j) and A (i,j) is the value of the pixel in the reference block at position (i, j). Difference between these evaluation criteria is known as computational complexity. PSNR is the most widely used criterion because it is simple and for making calculations quickly. When motion vectors computed using the BM method are applied, this value shows the reconstruction quality. Another commonly used matching metric is the sum of absolute differences (SAD)[11]. SAD is considered as time-consuming operation in the BM process. SAD is a popular criterion for BM algorithms. Eq. (2.6) considers a template block at position (*x*, *y*) in the current frame and a candidate block at position ($x + u\hat{}, y + v\hat{}$) in the preceding frame I_{t-1}.

$$\text{SAD}\ (u\hat{}, v\hat{}) = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} |g_t\ (x+i,y+j) - g_{t-1}\ (x+ u\hat{}+i, y+ v\hat{}+j)| \qquad 2.6$$

CHAPTER 3.
LITERATURE REVIEW

3.1 Overview

Over the past decade, researchers have presented various hand-crafted and deep nets-based methods for video classification. Conventional video classification systems are based on extraction of local features, which have been generally advanced in both detection and description. Local features can be closely sampled or chosen by maximizing specific saliency functions. Until 2012, the majority of image identification, classification, and segmentation challenges were solved by obtaining hand-made features and employing particular algorithms to these challenges. If we needed to detect a number plate on a motorcycle, we first segment the image by finding corners and straight lines, and eventually reducing the image. Finally, we obtained a region that look a lot like the geometry of bike's number plate. In other words, we were looking for certain capabilities that can resolve a particular problem. Histogram of Oriented Gradients method is a common technique used for hand-crafted features in order to classify motion in video data. The primary concept underlying this descriptor is that the edge directions can be used to characterize the form and structure of local objects. In the past few years, video classification has found a major paradigm shift, which involved shifting from hand-designed features to deep network methodologies that learn features and categorize end-to-end. Deep learning algorithms were introduced in 2012, resulting in new approaches to video categorization and related challenges [1]. Recently, with the availability of large-scale video datasets and mass computation power of GPUs, deep neural networks have achieved remarkable advances in the field of video classification. Over the years a variety of problems like multimedia event recounting, surveillance event detection, action search, and many more have been proposed. A large family of these research is related to video classification. Many improvements of video classification are inspired by the advances in image domain. The innovation on image classification [20] also revives the interest in deep neural networks for video classification. Because of its large feature representation characteristics, deep learning-based techniques may accurately find hidden patterns in visual data. Moreover, it also

necessitates a large amount of data for training and a lot of computing capacity to process it. We were able to strike a compromise between the system's complexity and video categorization accuracy in this project. Our method is computationally efficient as it focuses on just the motion vectors of video frames instead of complete video sequences. In the following we focus our discussion on works related to block matching motion estimation and classification methods that are based on DNN.

3.2 Block Matching Motion Estimation

Numerous methods have been studied for detecting motions in a video series: background subtraction, frame difference, and optical flow method [7]. These methods have a number of flaws, including results that are inaccurate. They are having difficulty identifying optimal solutions due to the huge search space. They also take a long time to run, therefore they can't be employed in real-time applications. Many academics have previously used block matching motion estimation to categorize motion in video data by processing individual pixels. This procedure requires a significant amount of computational time. Researchers have undertaken a number of initiatives to address this issue, which are summarized in this section in chronological order. Erik Cuevas1 et al [7] proposed a Block matching technique for motion estimation process. Their technique was based on Artificial Bee Colony for reducing search locations in Block Matching method. The calculation of search positions in this algorithm is significantly decreased by using a fitness computation procedure. The procedure shows when it is feasible to compute other search sites. Razali Yaakob et al. [17] have proposed four alternative Block Matching Algorithms for Motion Estimation. Furthermore, they determined the best block matching technique using Peak signal to noise ratio. Sonam T. Khawase et al [8] have proposed different block matching algorithms. Their technique was based on shapes and patterns, in addition to block matching principles for motion estimation. Mr. P. Vijaykumar et al [5] proposed Latest Applications, Innovations and trends in Motion Estimation. Their research examines recently discovered motion estimating applications that are causing a revolution in the world of technology by providing more precise and efficient solutions to difficult problems. They have a long list of applications for motion estimation. Just a few

examples include motion estimation in biology, space science, cardiac surgery, user authentication, and respiratory analysis. Dr. Anil Kokaram et al [24] pioneered Block Matching Motion Estimation, Gradient Based Motion Estimation, and Optical Flow. Thier research introduces about motion estimation, forms of motion estimation, gradient-based techniques, block-based motion estimation, and Weiner estimates in a short presentation. Abir Jaafar Hussain et al. [6] suggested a number of quick, block-based matching algorithms to overcome concerns in the motion estimating process and to take use of assumptions made about distortion distribution behaviour. Through a literature assessment of their supporting papers, their investigation has analysed a number of such approaches, allowing for the establishment of comparison analysis based on various performance metrics. For global motion estimation, Syed Shuja Hussain et al [25] devised a robust video stabilization technique. They invented an algorithm that eliminates shakiness in a raw YUV video and resulting in YUV video stabilization. They presented parallel processing approaches for motion vector estimation, and the results were enhanced by using the proposed methodology to reduce the program's operating time. Junggi Lee et al [4] have introduced a compact but effective deep neural architecture known as BlockNet. This architecture extracts rich features from two input photos. It then calculates coarse-to-fine block motion using a pyramidal arrangement. In the block matching technique, K.Laidi et al. [9] used and compared various search algorithms. Among these algorithms are the three-step search algorithm, four-step search algorithm, new three-step search algorithm, and minimal search algorithm etc. The performance of an algorithm is a trade-off between the search time computations and PSNR. Weisheng Li et al [27] used statistical data from motion vectors to solve challenges in three steps. First step is frame preprocessing. This method utilizes Mode reduction method for removing redundant motion vectors due to camera movements. The second process is known as intra frame processing in which K-means clustering is utilized for segmentation of moving objects. The third process is inter-frame processing in which tracking object is assigned the same label in subsequent frames by comparing their positioning information. The tracking object's halting is represented by a copying rule. For occlusion problems, the motion vector's direction and velocity information are used. Daoud Boumazouza et al. [14] proposed the performance of the block matching method utilizing the Bees' Algorithm. In this algorithm efficient exploration of search space is

carried out. It uses two new approaches which are intensification and diversification. Josue Hernandez et al [13] introduced a motion detection technique based on the estimation of movement vectors, which is then filtered to gain more information about real motion in a picture. The proposed system's correctness was demonstrated by experimental findings. The proposed system estimates motion vectors from an input video frame, which are subsequently filtered to eliminate distortion caused by noise and poor illumination. The movement trace is then computed using the estimated motion vectors to evaluate if it is a relevant or irrelevant motion. Finally, if the movement is significant, it is tracked until the object has outside the restricted zone. Takanori Yokoyama et al. [12] proposed method that produces noise-free local motion vectors as well as high-quality global motion vectors for use in object detection, tracking, and other applications. This method is also applicable to videos shot with a moving camera. In many studies, they used real-life videos for demonstrating efficiency of proposed algorithm.

3.3 Classification of video data via deep learning

Several deep learning architectures like Convolution Neural Networks and Artificial Neural Networks have shown exceptional performance for many years in various computer vision applications. It is believed that they could be the ideal solution for many videos categorization challenges. Furthermore, rapid advancement of Graphics Processing Units (GPUs) has empowered the innovation of deep Neural Network based systems [28]. Deep learning is a novel approach in which neural networks are gradually deepened by adding tens or hundreds of layers. Prior to 2012, there was a lot of work done in computer vision with multi neural network layers, but the outcomes were disappointing. A CNN comprised of deeper layer model was used to recognize characters. However, it wasn't until 2012 that the true potential of CNNs were revealed, thanks to Alex Krizhevsky's AlexNet proposal. These strategies were initially employed during the Imagenet Competition, where GPUs were used to speed the revolutionary deep multi-layer neural network techniques. Since then, latest, and enhanced hardware has emerged. This paves the way for larger CNNs, improving classification precision and creating deep network a cost-effective scientific tool. The rise of CNNs has benefited computer vision

research groups focusing on video motion classification, and current evaluations have revealed that improved and more accurate outcomes are possible [28]. In general, naive movement models built on human posture analysis have been used to classify videos in the past. Other previous methods for estimating motion in video data have relied on Kalman Filters (KF). Due to the inability to appropriately handle video data, the majority of current approaches provided unsatisfactory outcomes. After reviewing current techniques, we can conclude that significant progress has been made in video motion classification systems in recent years. More research is needed for delivering of good classification accuracy, increased performance, and ease of incorporation in recent security surveillance techniques. Alex Dominguez et al. [22] have presented a technique that is based on CNN model. Their method utilizes existing techniques for pedestrian detection for generating sum of subtracted frames. The frames are given as input to the improved versions of CNN architecture like GoogleNet, ResNet and AlexNet. Furthermore, in order to obtain accurate results, they designed a latest data set for this purpose and examined the impact of neural network training. Can Yang and Gyz Gidófalv et al. [26] established a CNN model for classification of movement patterns in trajectories. They developed a new feature descriptor for trajectory data. The method identifies locations in space where the majority of trajectories follow a specific movement pattern, resulting in a regional dominant movement pattern. This movement data can be utilized for different purposes such as detecting hotspots, checking pedestrian location, and reviewing animal behavior. The pioneer work of Karpathy et al. [19] trained 2D-CNN on various forms of stacked video frames from Sports-1M. However, these deep networks are somewhat lesser to the shallow model based on the best hand-engineered features. Simonyan et al. [36] created the two-stream networks with two 2D-CNNs on spatial and temporal streams. This technique takes benefit of the large-scale ImageNet [20] dataset for pre-training and substantially lowers the complication to model dynamic motions through optical flow. Ji et al. [4], for example, provided a simple implementation of video categorization utilizing deep networks employing 3D convolutional networks. They have used kernels based on 3D convolution architectures to capture both temporal and spatial information from video frames. They also argued that their method could capture motion and optical flow from data. [23] proposes a CNN multi-resolution architecture for collecting local spatiotemporal data in the time domain. Their method was tested on a new

dataset of YouTube videos (1 million) that consists of 487 classes. The authors have reduced the training complexity by using CNN's multi resolution architecture. They improved the recognition level for huge datasets to 63.9 percent. The recognition rate on UCF101 dataset was only 63.3 percent which is still insufficient for such a critical task of video categorization. [1] proposes a two-stream CNN architecture which captures temporal and spatial information and shows intense optical flow between frames. They combined two datasets to increase the quantity of data for training the Convolution Neural Network.

CHAPTER 4.
PROPOSED METHODOLOGY

This chapter presents the suggested methodology for efficient classification of motion in video data using Deep Neural Network approach. We have used deep learning techniques for classification of motion in video data because we feel they could be the practical solution to video classification issues. Furthermore, quick evolution and outstanding performance of Graphics Processing Units (GPUs) has supported the creation of deep learning-based systems. Machine learning and image processing methodologies have been altered in many ways. In this research we have presented a motion classification algorithm using motion fields and Deep Neural Network. The motion field is created using block matching algorithm. The magnitude as well as direction of motion vectors extracted from video frames are given as input to Deep Neural Networks. In our research, we plan to combine Block matching motion estimation technique and Deep Neural Network. Block matching motion estimation algorithm will provide the values of motion vector for each frame of video sequence. From the values of motion vector, we will calculate magnitude and direction associated with each motion vector. Magnitude and direction values will be provided as input to Deep Neural Network. There will be a training phase in which Deep Neural Network will learn about the gesture information contained in the motion vectors of each frame. Gesture can be of different types like accident, call, doctor, help etc. In the testing phase we will provide the motion vectors from a sequence in order to classify the motion in them accurately. We have used Artificial Neural Network and Convolution Neural Network for classifying gesture in a video sequence. The proposed methodology is shown in fig 4.1

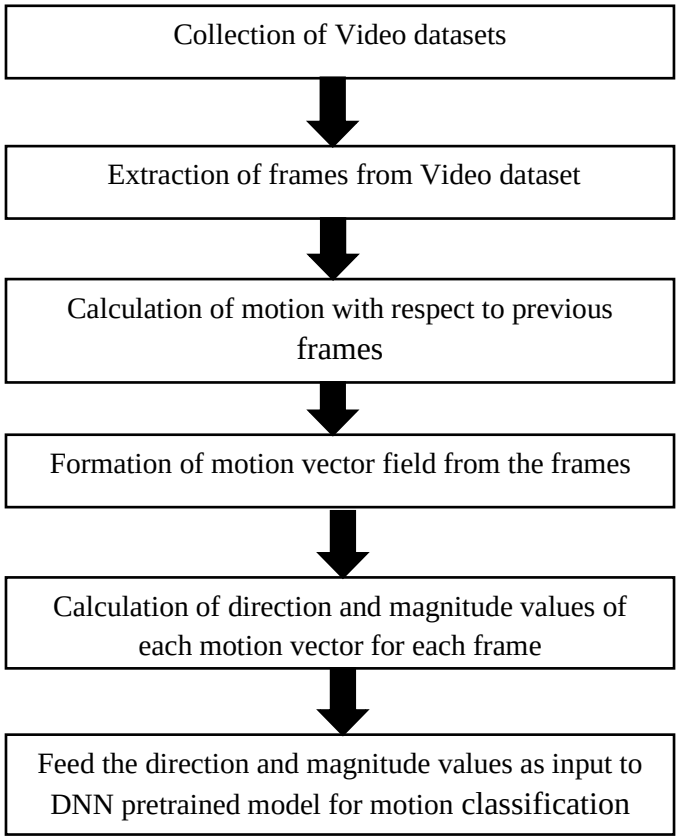

Figure 4-1: Flowchart showing proposed methodology

4.1 Artificial Neural Network

A neural network is built up of simple units or nodes which are connected together. The functionality of neural network is similar to neurons that are present in animal's brain. The weights or interunit connection strengths have the ability to process input data. The network of neurons may have single node or large collection of nodes. The nodes are connected to other nodes in the form of a net [30]. One example of network is depicted in Figure 4.2. The nodes are represented by circles with weights attached on all connections. The nodes are arranged in the form of layers. The signal from nodes originates and passes via several other nodes before reaching the final output. This structure is known as *feedforward* structure. For example, if the input consists of handwritten light and dark patterns, the output layer will have 26 nodes each

node representing one letter for the alphabet. This would be accomplished by assigning one output node to each class. Artificial neural networks (ANNs) are quite recent tools that can be used for many computational tasks. ANNs can solve many complex real-world challenges. The beauty of ANNs comes from their extraordinary information managing characteristics. The main capabilities of ANNs are high parallelism, noise tolerance and generalization. Artificial Neural Networks has turn out to be the most important architecture in today's research. Machine learning use many statistical and mathematical approaches for increasing the capability of computers to recognize the data content. ANNs deals with complex forms in large quantity of data. This requires a deep learning algorithmics in a systematic manner. There are many types of procedures that are required during the implementation process of ANN [31]. Three types of such procedures are supervised learning, unsupervised learning or reinforcement learning. The network pattern of neurons can perform various decision-making tasks. The formation of an ANN was based on input, process and output nodes. As a result, ANN performs as a complex mathematical tool to produce an ideal result for any datasets. Neurons should be tested in two ways to complete a network cycle i.e., feed-forward and backward algorithms [30]. Backward algorithms, also known as back propagation, have been widely discussed and studied by many researchers in the past and present. In the word 'artificial neural network,' the term network means the interconnection of neurons arranged in various levels. A three-layered architecture is used as an illustration. The first layer comprises of input neurons. The data from input neurons is transmitted to the second layer of neurons. The second layer of neurons then passes data to the third layer of neurons which is actually the final output. More layers of neurons will be present in more complicated architectures of ANN. The synapses store "weights," which are used to change data in computation [31]. An ANN is commonly defined by three types of parameters: 1. The pattern of connectivity among several neurons 2. The process in which weights are changed 3. The activation function for converting a neuron's weighted input to activation of its output. The Artificial Neural Network model is depicted in the diagram below.

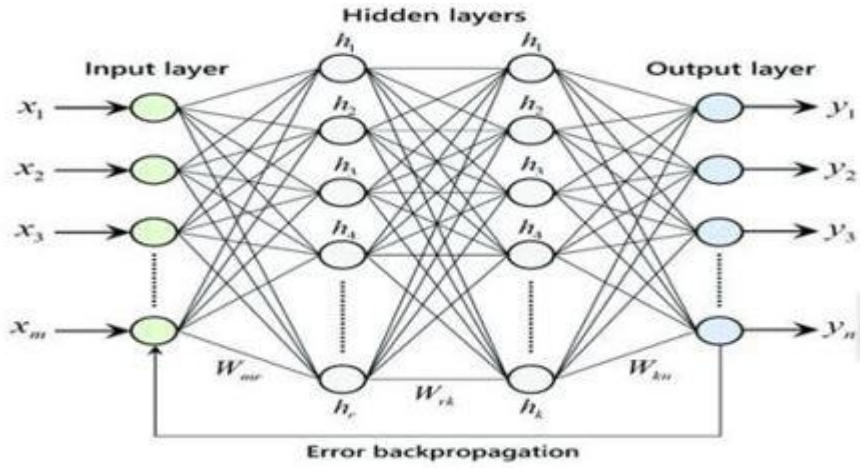

Figure 4-2: Artificial Neural Network model

The working of neural networks depends upon the ways in which the neurons are grouped together. This clustering of neurons is arranged in a person's mind so that information may process in an interactive, and self-organizing manner. These neurons are capable to form unrestricted interconnections. But the man-made network is entirely different from animal's neurons. Currently, neural networks are the grouping of neurons. This grouping take place in the form of layers that are connected with one another. Connectivity of neurons is actually the art of engineers for resolving genuine world problems.

4.2 Convolution Neural Network

Recently convolutional neural networks have completely dominated the machine vision space. Deep learning has become one of the trendiest issues in artificial intelligence (AI) today because of neural networks' capability and influence. CNNs were invented by Yann Lecon of New York University, who is also the director of Facebook's AI project. CNNs are type of deep neural network that was created using biologically inspired models [33]. Researchers discovered that mammals and humans use a layered architecture of neurons in the brain to visually interpret their surroundings. Engineers were then inspired to create comparable pattern recognition systems, which is how these nets evolved.

4.2.1 Resemblance with MLP

The CNN concept is similar to that of a multilayer perceptron (MLP). As demonstrated in Figure 4.3, cutting a few connections and sharing MLP weights results in a single CNN layer. The graphic shows that the configuration of an artificial neural network (ANN) is equivalent to a 2D convolution operation, with weights acting as filters (also called masks or kernels) [33]. The number of parameters involved in a convolutional network is greatly reduced when biases and weights are shared.

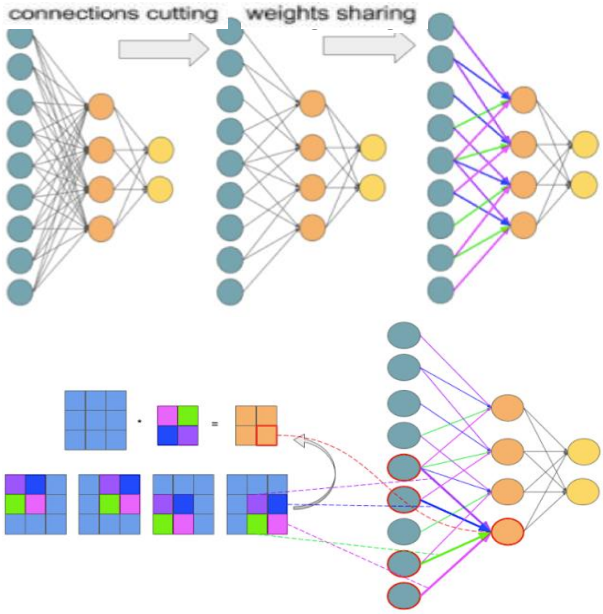

Figure 4-3: CNN resemblance with MLP

4.2.2 Architecture

An input, numerous hidden layers, and an output layer make up a CNN. Convolutional layers, activation layers, pooling layers, and fully connected (FC) layers are common hidden layers. The first three layers are applied one after the other to aid in feature learning, while the last FC layer is employed to aid in categorization. Following layers are the main building blocks of a typical CNN:

4.2.3 Convolutional layers

The convolutional layer is made up of a number of distinct kernels or filters, each of which is convolved with the image separately (Figure 4.4). Convolution is achieved by applying a filter over the entire image and taking the dot product between portions of the image and the filter along the way. All of the filters are initialized at random, and these are the parameters that the network will learn later. The first layers look for simple patterns like lines and corners [34]. As we progress through the convolutional layers, the filters perform dot products on the previous convolutional layers' input. As a result, they are using the excised parts or edges to create larger pieces.

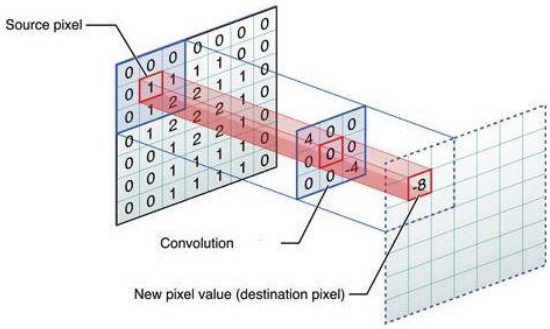

Figure 4-4: Representation of Convolution operation

4.2.4 ReLU (Rectified Linear Units) layers:

ReLU is a nonlinear activation layer. It is applied directly after each convolutional layer. ReLU layers reduce the time it takes to train a network (due to their computational efficiency) without sacrificing accuracy. Anther function of ReLU layer is to solve the problem of vanishing gradient. To all the values in the input volume, ReLU applies the activation function f(x) = max (0, x). To put it another way, the layer simply sets all negative numbers to zero. It improves the model's nonlinear properties and the entire network's nonlinear properties without affecting the convolutional layer's receptive fields.

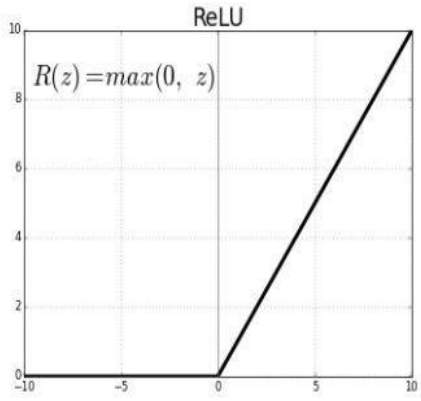

Figure 4-5: Graph showing ReLU function.

4.2.5 Pooling layers

The aim of pooling layer is to decrease the number of computations and parameters of the network. There are many non-linear functions that can perform pooling operations such as L2-norm pooling, average pooling but the most common is max pooling. Max pooling divides the image into a set of non-overlapping chunks. It selects the maximum value for each chunk. Thus, it reduces the spatial dimension of the input data. There are two main purposes of pooling layer. Firstly, it reduces the computational cost by decreasing 75% amount of parameters [35]. The second purpose is to control the problem of overfitting. The issue of overfitting happens when model is so tuned for the training samples that it can't generalize for the validation data. Figure 4.6 shows an example of how max and average pooling is done.

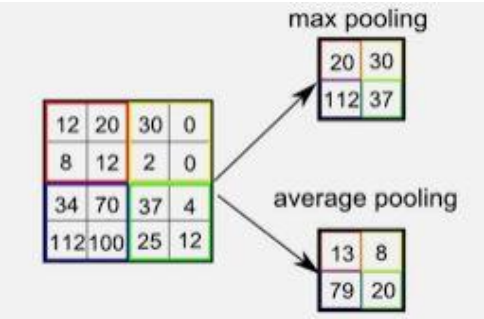

Figure 4-6: Max and average pooling example

4.2.6 Fully connected layers

The convolutional, ReLU and pooling layers are applied repeatedly. The function of FC layers is to achieve high level reasoning in CNN model. The output image from the convolutional layers has high-level features in it. FC layer adds non-linear combinations of those features. Combinations of features is much better than all the features from convolutional layers. Neurons in an FC layer have full connections with all activations of previous layers.

4.3 Motion Classification in videos via ANN and CNN

Video classification using Deep Neural Network have some basic steps and those steps should be done in sequential order. First step is collection of video dataset. Second step is to extract frames from video dataset. In the third step we have to extract motion vectors from video frames using three step search which is a block matching motion estimation method. Motion vectors are extracted from two consecutive frames i.e., from frame 1 and frame 2 similarly from frame 2 and frame 3 and so on up to the number of total frames contained in a video sequence. In the fourth step, after extracting the values of motion vector, we have to calculate magnitude and direction values associated with each motion vector. In the fifth step, we have to input magnitude and direction values of motion vectors to Artificial Neural Network and find testing and training accuracies for video class prediction. In the sixth step, we have to input magnitude and direction values of each motion vector to Convolution Neural Network and again find training and testing accuracies as well as class labels. In the end we will compare results associated with magnitude and direction values of motion vectors with just frames of video sequences. Following figure illustrate the step-by-step methodology for extracting motion vectors using block matching motion estimation and feeding the magnitude and direction values to Deep Neural Network for video class prediction.

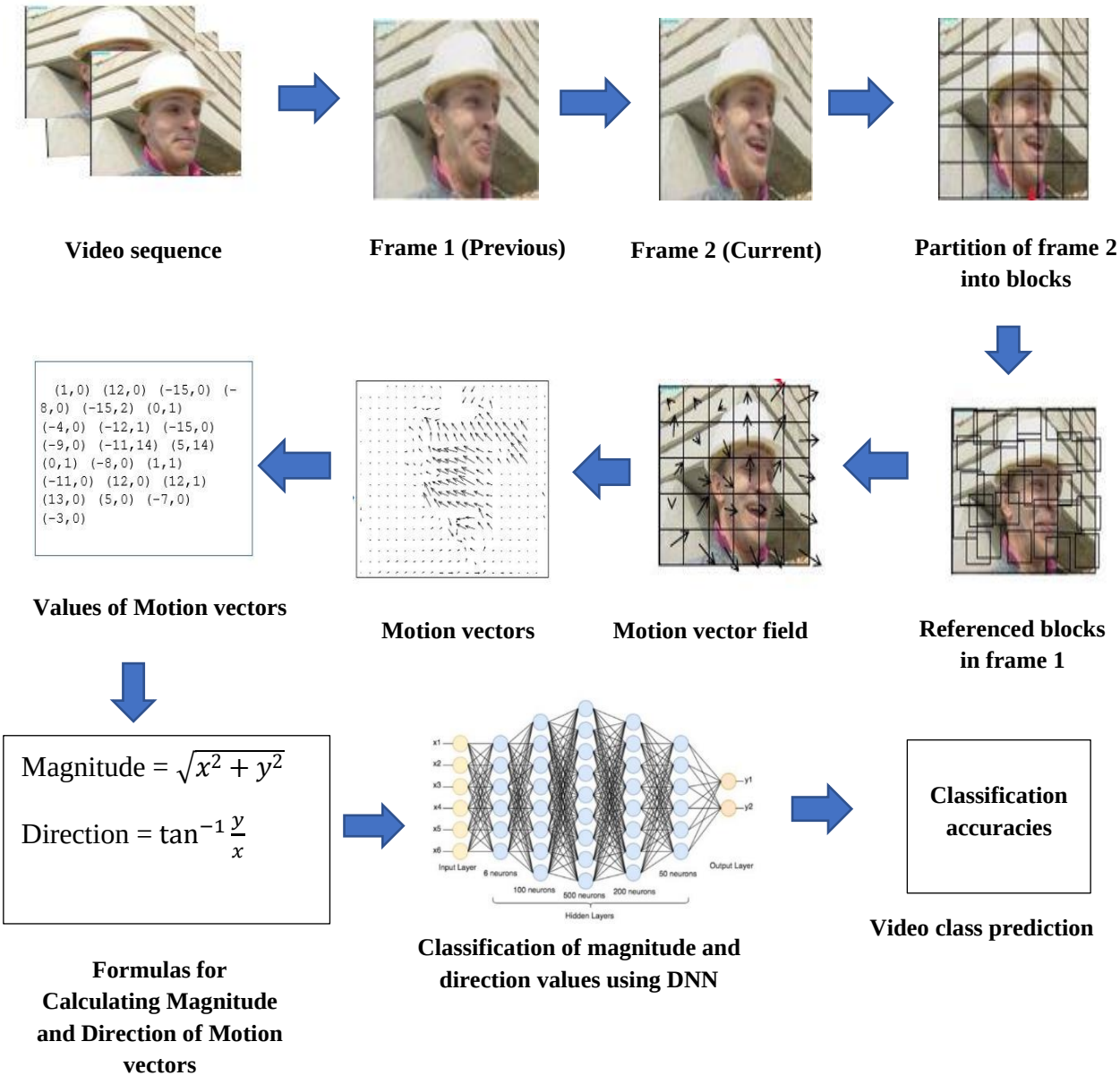

Figure 4-7: Step by step representation of proposed methodology

CHAPTER 5.
EXPERIMENTAL RESULTS

5.1 Databases

Experimental results are proposed in this section. The results are discussed using a benchmark Video Dataset of Hand Gestures. The gesture dataset represents Sign Language Words that can be used in Emergency Situations [36]. This dataset was obtained from the Mendeley website.

5.1.1 Description

The dataset consists of hand gesture video files for 8 different classes. The dataset represents Sign language (ISL) which are accident, lose, thief, call, hot, help, pain and doctor [36]. The data is useful for the researchers who wants to work on vision-based automatic sign language recognition and hand gesture recognition. Sign language recognition helps the deaf by bridging the communication gap that exists between them and the rest of society. The aim of sign language communication is to improve sign language recognition [36]. All the gestures are dynamic hand gestures except the gesture for doctor. The participants for data collection were standing behind a black colored board. The digital camera used for shooting videos was a Sony cyber shot with 20.1 mega pixels resolution. Twenty-six people have participated for video collection task. There were 12 men and 14 women whose age were ranging from 22 to 26 years. Each participant had performed the gesture twice. All the gestures were performed under normal lighting conditions with the camera set at a fixed distance. The dataset is divided into two folders, one containing the original raw video sequences and the other containing the cropped and down sampled video sequences.

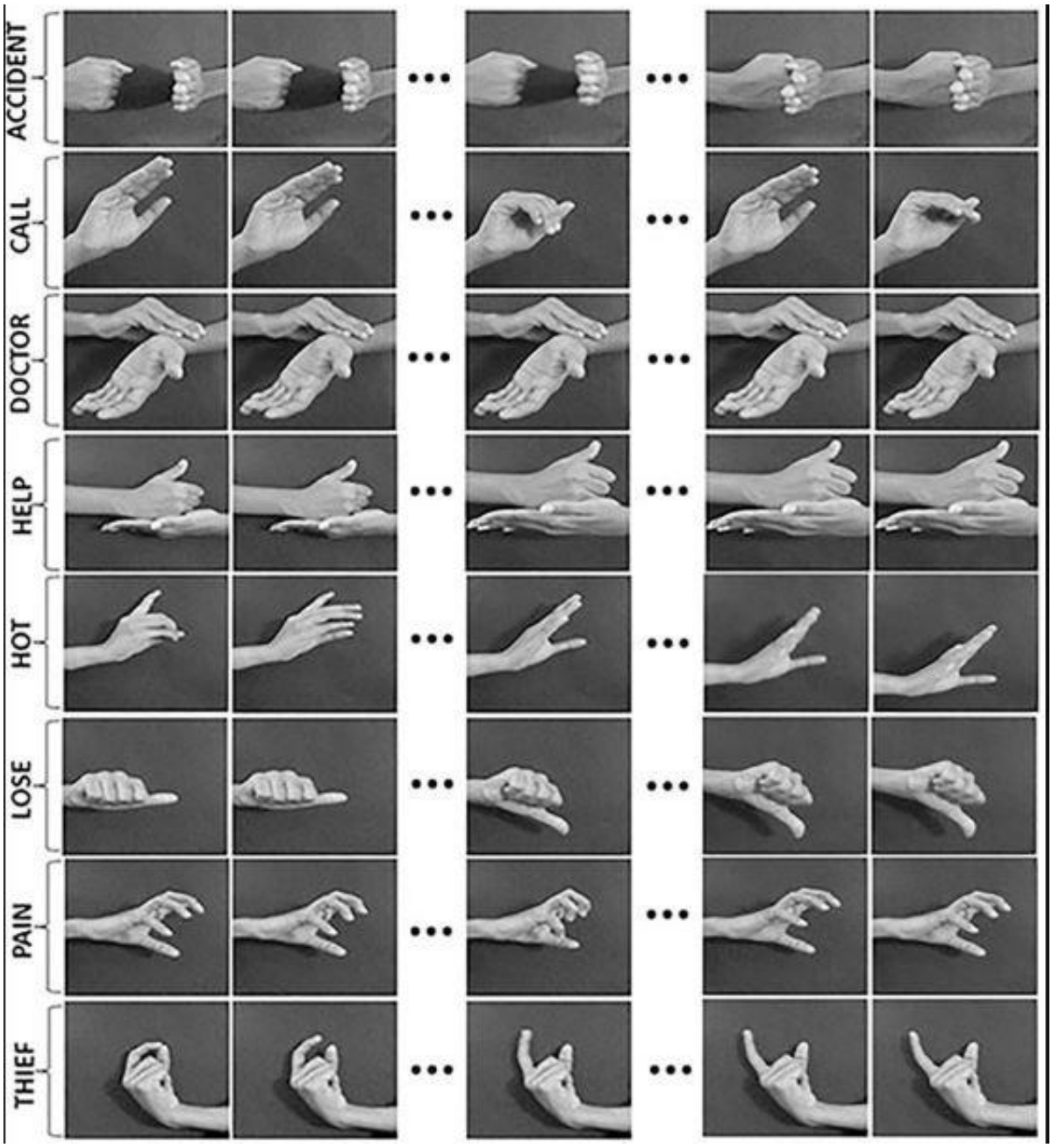

Figure 5-1: The key frame sequences of the hand gestures of the ISL words

Table 5.1: Specifications of dataset

Sr No	Parameters	Specifications
1	Subject	Pattern Recognition
2	Specific subject area	Automatic sign language recognition
3	Type of data	Videos
4	How data were acquired	The participants performing hand gestures were standing behind a black colored board. The digital camera used for shooting videos was a Sony cyber shot with 20.1 mega pixels resolution
5	Data format	Videos were divided into two sets. First set contains video sequences in original form while second set contains videos in crop form. Pixel size was 500 x 600 after down sampling the frames.
6	Parameters for data collection	Camera was kept at a fixed distance. Participants were gathered from different parts of India. Their hand and skin tones were also different.

5.2 Experimental Setup

In this section, we will describe experimental setup that has been employed to demonstrate the efficacy of our algorithm during testing and validation. We ran various simulations to ensure accuracy. We used three-step search block matching algorithms to extract motion vectors, and Artificial Neural Network and Convolution Neural Network for classifying motion vectors. As test materials, we used eight different video sequences of hand gestures in AVI format. Accident, lose, thief, help, hot, call, pain, and doctor are the hand gestures. Each video sequence contains an average of 70 video frames, which represents a varied range of motion and can

improve the accuracy of the simulation results. The following are the test conditions: The experiments were carried out on Windows Operating System, and the software used for experimentations are MATLAB and Python; some parameter settings are shown in the table.

Table 5.2: Specifications of video frames

Sr No	Parameters	Values
1	Input File	Video file of hand gestures (8 classes)
2	Average Frames Extracted	70
3	Frame rate	30.0
4	Source width	601
5	Source height	501
6	Images AVI	4:2:0
7	Block size	35x35
8	Maximum Displacement search	[7 7]
9	Match Criteria between blocks	Mean square error
10	Search Method	Three step searches

5.3 Experiment Analysis for Artificial Neural Network

Five layered Neural Network is used in our experiments. These layers are input layer, 2 hidden layers, and an output layer. We then have calculated the motion vector for each image frame in such a way that first the algorithm calculates the motion vector of frame 1 and frame 2 then it calculates the motion vectors of frame 2 and frame 3 then for frame 3 and frame 4 and so on up to the total number of video frames for each video sequence. We have then arranged the values of motion vectors in a csv file and calculated magnitude and direction values for each motion vector. The csv file contained magnitude and direction of motion vectors as well as class label. Label contained 8 classes of accident, doctor, call, help, hot, pain, lose and thief. We have inputted the .csv file to Artificial Neural Network. Table shows the proposed ANN architecture used for classification.

Table 5.3: Specification of ANN model

Sr No	Parameters	Specifications
1	Model	Sequential
2	No of layers	5
3	Learning rate	0.001
4	Loss	Categorical Cross entropy
5	Batch size	32
6	Epochs	100
7	Optimizer	Adam

5.4 Experiment Analysis for Convolution Neural Network

Following layers and parameters are included in our proposed CNN architecture. It has proved to be the best architecture for our classification problem.

Input layer: The input layer of CNN receives input data and generates output that is used to feed convolution layers. In our case, inputs are magnitudes and directions of motion vectors. The parameters that are defining the dimensions of motion vectors are (15 x 15).

Convolution layers: The function of convolution layer is to convolve the 2-D magnitude and direction values with a set of learnable filters. Each filter produces one feature map in the output. This model has two convolution layers [33]. The size of receptive fields (kernels) is 5x5. The zero padding has been set to zero. Stride has been set to one. The first convolution layer convolves motion vector values with 32 filters, and the second convolution layer convolves motion vector values with 64 filters. The standard deviation was set to 0.0001.

Pooling layers: The function of pooling layer is to sample the spatial dimension of the input. The pooling layer is added after each convolution layer. They are all configured to use 3x3 receptive fields (spatial extent). The stride has been set to 2. The first pooling layer is in charge of most common max operation on the receptive field. The second layer employs average pooling.

ReLU layers: ReLU activation function is a non-linear operator. For convenience we have treated it as a layer. This model contains three ReLU layers. The ReLU layer calculates the neuron's output for each input value x. The parameters identify whether the negative part should be leaked by multiplying it by the slope value (0.01 or so). When this parameter is not set, the activation is simply thresholded at 0, which is equal to the basic ReLU function f(x) = max (0, x) [34].

Fully connected layers: These layers consider the input and output as a simple vector. Two inner product layers are contained in this model. The fully connected output layer is the last layer with softmax as activation function. The number of fully connected output layer is determined by the number of classes in the classification problem. Our classification problem contains eight output filters for eight classes. Table summarizes the parameters of the CNN layers.

Table 5.4: Specifications of CNN model

Sr NO	Parameters	Layer 1	Layer 2	Layer 3	Layer 4
1	Type	CONV + $POOL_{max}$	CONV + $POOL_{avg}$	FC	FC
2	Channels	32	64	64	8
3	Filter Size	5x5	5x5	--	--
4	Convolution Stride	1x1	1x1	--	--
5	Pooling size	2x2	2x2	--	--
6	Pooling stride	2x2	2x2	--	--
7	Padding size	0	0	--	--

5.5 Performance Measures

We used the Confusion matrix as a performance measure for ANN and CNN-based gesture data classification in a video sequence. A Confusion matrix is actually an M x M matrix. It is utilized to evaluate classification model's performance. The factor M is the number of target classes. The function of matrix is to compare the genuine target values with the machine learning model's predictions. We can estimate the performance of classification model using confusion matrix. We can visualize the performance of a prediction model in a tabular form. Each entry in a confusion matrix signifies the number of predictions made by the model. It shows that either the classes were correctly classified or not. It gives us a very simple and effective performance metrics for our model. The confusion matrix is comprised of four basic characteristics. It defines the classifier's measurement metrics. These four basic characteristics are:

1. TP (True Positive): It denotes the number of predictions accurately predicted as positive by the classifier.
2. TN (True Negative): It denotes the number of predictions accurately predicted as negative by the classifier
3. FP (False Positive): It represents the number of times the classifier predicts the negative class as positive.
4. FN (False Negative): It is the number of predictions in which the classifier predicts the positive class as negative.

Most commonly used performance measures for confusion matrix are listed below. The performance of an algorithm is based on accuracy, precision, F1 score and recall. The performance measures are calculated using the TP, TN, FP, and FN values that were defined previously.

Accuracy: It represents the complete accuracy of classification model. It is the percentage of overall samples accurately classified by the classifier. The accuracy calculation (AC) is used to calculate the efficacy of the system. The following formula is used for accuracy calculation:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

Misclassification Rate: It shows the proportion of predictions that were not correct. It is also known as a Classification Error. We can compute it using the following formula:

$$MIS\ rate = \frac{FP + FN}{TP + TN + FP + FN} \tag{5.2}$$

Recall: It represents the total positive samples that were correctly classified as positive. True Positive Rate (TPR), Sensitivity, and Probability of Detection are other terms for it. The recall is determined by dividing the percentage of correctly identified positive inputs by the number of positive inputs. It is also referred to as specificity. It is calculated using the following equation:

$$SEN = TPR = \frac{TP}{TP + FN} \tag{5.3}$$

Precision: It shows the percentage of entire negative samples correctly predicted as negative by the classifier. It is also referred to as the True Negative Rate (TNR). It is also referred to as sensitivity. It is calculated using the following equation:

$$SPE = TNR = \frac{TN}{TN + FP} \tag{5.4}$$

F1 score or F measure: It incorporates recall and precision into a single metric. It is the harmonic mean of precision and recall in mathematics. It also serves as a measure of the test's precision. Its maximum value is 1 and minimum value is 0.

$$F1\ score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \tag{5.5}$$

5.6 Results

The results show the accuracy of magnitude and direction values of motion vectors using ANN and CNN architectures. The results were also evaluated using video frames of hand gesture dataset.

5.6.1 Classification Results of ANN

The classification methodology was evaluated on magnitude and direction of motion vectors obtained from hand gestures of eight classes like call, accident, help, lose, doctor, hot, thief and pain. According to machine learning protocol, the dataset is divided into 80 % and 20 % training and testing set respectively. The training data is inputted in mini batches of 32, with a learning rate of 0.001 for cost minimization. The iterations used were one hundred to learn the data sequence patterns. The experiments were evaluated on combined magnitude and direction values of motion vectors for calculation of training and testing accuracies with class labels. The training and testing accuracy trend of some iteration (or data fold) of ANN with magnitude and directions values is plotted and they showed the ANN model accuracy and model loss respectively. ANN model achieved 86% training and 84% testing accuracy when tested with magnitude and direction values of motion vectors.

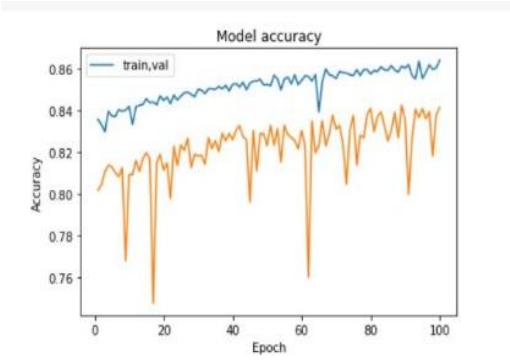

Figure 5-2: ANN model accuracy for magnitude and direction values of motion vector

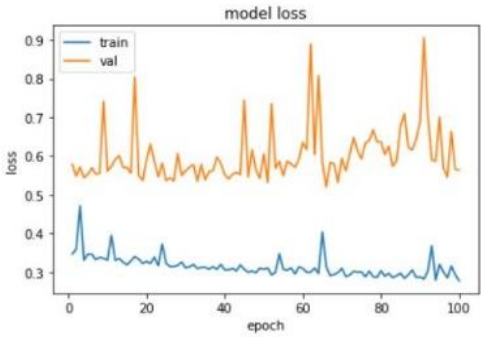

Figure 5-3: ANN model loss for magnitude and direction values of motion vector

Classification performance of deep learning ANN model on the magnitude and direction values of motion vectors of 8 ISL words.

	precision	recall	f1-score	support
0	0.93	0.94	0.94	8222
1	0.94	0.95	0.95	8512
2	0.90	0.96	0.93	8434
3	0.98	0.92	0.95	8526
4	0.96	0.94	0.95	8311
5	0.62	0.99	0.76	13527
6	0.93	0.91	0.92	8290
7	0.90	0.92	0.95	8106
accuracy			0.84	71928
macro avg	0.83	0.83	0.80	71928
weighted avg	0.82	0.84	0.80	71928

Figure 5-4: Values of performance measures against magnitude and direction of motion vectors

Confusion matrix of ANN showing the correctly predicted 8 classes on principal diagonal using magnitude and direction values of motion vectors

```
Accident [[ 7744   116   133     6    10    61   137    15]
   Call  [   98  8057   120     8    74    53    92    10]
 Doctor  [  125    37  8056    22    60    58    74     2]
   Help  [   36    36   261  7877   146    47   123     0]
    Hot  [   51   108    98    61  7822    73    97     1]
   Lose  [    5    15    59     0     1 13364     5    78]
   Pain  [  275   156   187    29    35    69  7537     2]
  Thief  [    6    14    29     0     0    62     8  7987 ]]
```

Figure 5-5: Confusion matrix results for magnitude and direction values of motion vectors

5.7 Classification results of CNN

The classification methodology was evaluated on magnitude as well as direction of motion vectors obtained from hand gestures of eight classes like call, help, doctor, pain, lose, hot and accident According to machine learning protocol, the dataset is divided into 80% and 20% training and testing set respectively. The training data is inputted in mini batches of 64, with a learning rate of 0.001 for cost minimization. The iterations used were one hundred to learn the data sequence patterns. Table 5.5 shows the classification results of CNN. Column 2 depicts the input file which was given as input to Convolution Neural Network. The experiments were evaluated on concatenated magnitude and direction values of motion vectors for calculation of training and testing accuracies. Because we're interacting with motion vectors of frames, the basic point is that these motion vectors possess sufficient information to train a model if an appropriate set of motion vectors is extracted from each frame of the video sequence. So, in order to compare the CNN results of magnitude and direction values of motion vectors we have also carried out experiments with frames of video sequences of 8 gestures. Frames of each video sequences are given as input to CNN in order to find training and testing accuracies.

Table 5.5: CNN model accuracy with magnitude and direction values of motion vectors

Sr no	Input	Training Accuracy	Testing Accuracy
1	Magnitude and Direction (Combined)	94%	90%
2	Frames of video sequences	100%	99%

The training and testing accuracy trend of some iteration (or data fold) of CNN is plotted and they showed the CNN model accuracy and model loss respectively. CNN model achieved 90% testing accuracy when tested with magnitude and direction values of motion vectors. The receiver operating characteristic (ROC) curve illustrates the accuracy of CNN model with magnitude and direction values of motion vectors.

.

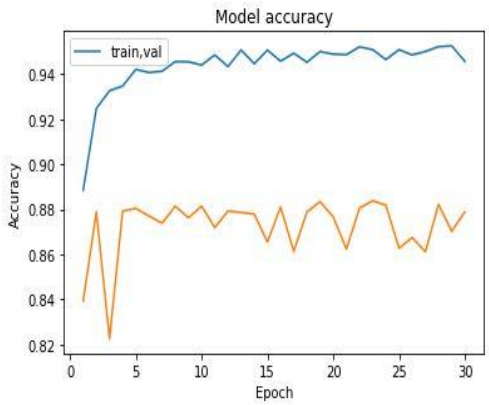

Figure 5-6: CNN model accuracy for magnitude and direction values of motion vector

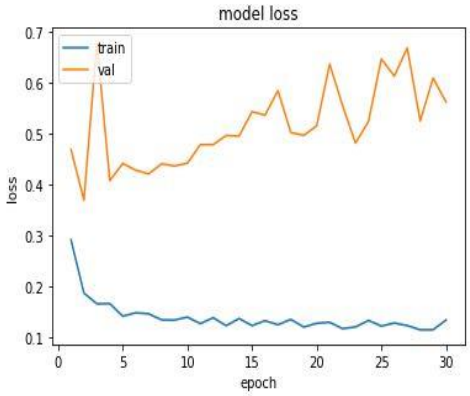

Figure 5-7: CNN model loss for magnitude and direction values of motion vector

The classification performances of all the approaches have also been estimated by applying the metrics for recall, precision and F-score values. These values correspond to each gesture class as shown in the following figures

	precision	recall	f1-score	support
0	0.71	0.95	0.81	612
1	0.92	0.82	0.87	662
2	1.00	0.95	0.97	653
3	0.97	0.93	0.95	624
4	0.86	0.84	0.85	666
5	0.90	0.88	0.89	585
6	0.92	0.87	0.89	632
7	0.98	0.95	0.96	601
accuracy			0.90	5035
macro avg	0.91	0.90	0.90	5035
weighted avg	0.91	0.90	0.90	5035

Figure 5-8: Values of performance measures against magnitude and direction of motion vectors

Confusion matrix of CNN showing the correctly predicted 8 classes on principal diagonal using magnitude and direction values of motion vectors.

Accident	[[581 0 2 11 7 8 3 0]
Call	[32 545 0 0 33 16 31 5]
Doctor	[27 1 623 1 0 1 0 0]
Help	[38 0 0 578 5 1 1 1]
Hot	[44 20 0 4 557 26 13 2]
Lose	[34 3 0 3 24 515 2 4]
Pain	[33 24 0 1 20 5 549 0]
Thief	[29 2 0 0 1 1 0 568]]

Figure 5-9: Confusion matrix result for magnitude and direction values of motion vectors

Experiments were also carried out on entire frames of video sequences in order to compare the results with magnitude and direction values of motion vectors. The training and testing accuracy trend of some iteration (or data fold) of CNN is plotted and they showed the CNN model accuracy and model loss respectively. CNN model achieved 99% testing accuracy when tested with just frames of video sequences

The receiver operating characteristic (ROC) curve illustrates the accuracy of classifying frames of 8 video sequences.

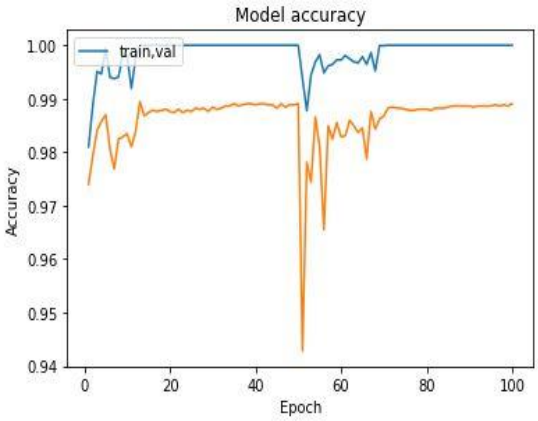

Figure 5-10: CNN model accuracy for video frame classification

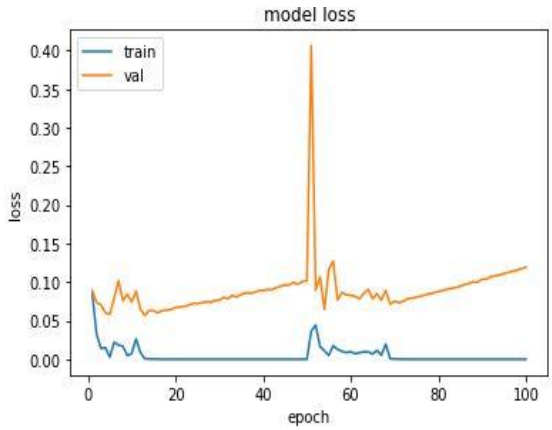

Figure 5-11: CNN model loss for video frame classification

The classification performances of all the approaches have also been estimated by applying the metrics for recall, precision and F-score values. These values correspond to each gesture class as shown in the following figures

	precision	recall	f1-score	support
0	1.00	0.99	0.99	640
1	0.98	0.96	0.97	626
2	1.00	1.00	1.00	616
3	1.00	0.99	1.00	617
4	0.97	0.99	0.98	622
5	0.98	0.99	0.98	562
6	0.99	0.99	0.99	573
7	0.99	1.00	0.99	585
accuracy			0.99	4841
macro avg	0.99	0.99	0.99	4841
weighted avg	0.99	0.99	0.99	4841

Figure 5-12: Values of performance measures for video frame classification

Confusion matrix of CNN showing the correctly predicted 8 classes on principal diagonal using frames of video sequences.

```
Accident [633   0   0   1   2   3   0   1]
Call     [  0 603   0   0   5   7   7   4]
Doctor   [  0   0 616   0   0   0   0   0]
Help     [  1   0   0 613   3   0   0   0]
Hot      [  0   5   0   0 616   1   0   0]
Lose     [  0   1   0   0   5 555   0   1]
Pain     [  0   4   0   0   2   0 567   0]
Thief    [  0   0   0   0   0   0   0 585]]
```

Figure 5-13: Confusion matrix result for video frame of each gesture

5.8 Comparisons

Comparison of results showed that although the accuracy is much better when classification is performed by giving frames as an input to CNN, but it requires high computational complexity

while processing each pixel of video frames rather than just magnitude and direction values of motion vectors. Our classification approach is fundamentally different from other methods and produces significantly better results as it requires less computational time, and less memory requirements. Comparison is also performed with other research on video classification. The researchers have used different CNN models and datasets for video classification task. Following table illustrates the performance of different deep learning models on different datasets

Table 5.6: Different DNN techniques for video classification

S. No	Paper	Dataset	Technique	Score/Accuracy
1	Pedestrian Movement Direction Recognition using Convolution Neural Network	Daimler dataset	AlexNet, Google LeNet and ResNet	79%
2	Multilayer and Multimodal fusion of Deep Neural Networks for video classification	UCF101 And HMDB51	2D CNN	91.6% and 61.8%
3	Exploring inter-feature and inter-class Relationships with Deep Neural Networks for video Classification	Hollywood2, Columbia Consumer Videos,	Deep Neural Network	65.7%
4	BlockNet: A Deep Neural Network for Block-Based Motion Estimation using Representative matching	Flying Chairs dataset	BlockNet	70%
5	Action Recognition in Video Sequences using	UCF101, HMDB51	LSTM and CNN	75%

	Deep Bi-Directional LSTM with CNN features	And Action YouTube		
6	A Novel key frame extraction method for video classification using Deep Neural Network	KTH and UCF-101	ConvLstm VGG-16	67.39%
7	Large scale video classification with Convolution Neural Network	UCF-101 and sports 1M dataset	CNN	65.4%
8	Learning Deep trajectory descriptor for action recognition in videos using Deep Neural Network	KTH and UCF-50	DNN	95.6% on KTH and 92.4 on UCF-50
9	Compressed Domain video classification with deep neural network	UCF-101 and HMDB-51	3D CNN	77.5% on UCF and 49.5% on HMDB
10	Hand gestures for emergency situations: A video dataset based on words from Indian Sign Language	Hand gesture dataset with 8 classes	GoogleNet and LSTM, SVM	90% with SVM And 96.25% with Google Net

CHAPTER 6.
CONCLUSION & FUTURE WORK

6.1 Conclusion

We have proposed an efficient classification of motion in video data by using deep neural network. This method reduces the computation time for video classification using the idea of motion vectors. First motion vectors were extracted from whole video frames and their magnitude and direction values were calculated. By using just magnitude and direction values of motion vectors we have classified motion in video data by using deep neural network. This is an efficient way of classifying motion in video data as it requires less computational complexity, memory requirements and helps to reduce the redundant information in video frames. Experimental findings on popular hand gesture benchmarks for emergency situations have revealed that our method works well than other alternative techniques. When comparison is performed with traditional machine learning models trained on the same dataset, our classification results on hand gesture dataset revealed enhanced accuracy obtained by ANN and CNN. Furthermore, in terms of model training, the proposed method is comparable to, if not faster than, conventional methods, which is essential for large-scale applications. We have also compared the results of our proposed methodology with the accuracy achieved by inputting just frames of video sequence to DNN. Comparisons of results showed that although the DNN classification accuracy with frames of video data is higher than the magnitude and direction values of motion vectors, but our methodology gives a significant reduction in terms of computational time and cost when compared to the frames of video data. In particular, we have evaluated our model on the Sign language dataset for emergency situation and showed that our method is computationally less expensive while giving an approximately similar performance as of video frames.

6.2 Contribution

- Efficient classification of motion in video data by inputting magnitude and direction values of motion vectors to ANN.
- Efficient classification of motion in video data by inputting magnitude and direction values of motion vectors to CNN.

- Review & comparison of recent techniques for efficient video classification task.

6.3 Future Work

Future work can explore different DNN architectures for classification of video data such as Google net, VGG 16 and VGG 19 etc. Also, strategies to extract motion vectors for improving the accuracy can be discovered. Our methodology can be tested on other benchmark datasets including UCF-101, YouTube videos and HMDB51. To overcome the limitations of the video classification is the trends and opportunity for the researcher. To classify longer video, to recognize multiple action in video, to find correlation among different videos, classification of multiple objects action in the video and live steaming game video prediction are also trending and future work in video classification using DNN.

REFERENCES

[1] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018, doi: 10.1109/ACCESS.2017.2778011.

[2] S. Acharjee, N. Dey, D. Biswas, P. Das, and S. S. Chaudhuri, "A novel Block Matching Algorithmic Approach with smaller block size for motion vector estimation in video compression," in *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA)*, Nov. 2012, pp. 668–672. doi: 10.1109/ISDA.2012.6416617.

[3] S. Immanuel, D. Bala, and A. George, "A Study on Block Matching Algorithms for Motion Estimation," *Int. J. Comput. Sci. Eng.*, vol. 3, Jan. 2011.

[4] J. Lee, K. Kong, G. Bae, and W.-J. Song, "BlockNet: A Deep Neural Network for Block-Based Motion Estimation Using Representative Matching," *Symmetry*, vol. 12, no. 5, p. 840, 2020.

[5] M. P. Vijaykumar, A. Kumar, and S. Bhatia, "Latest Trends, Applications and Innovations in Motion Estimation Research."

[6] "Block Matching Algorithms for Motion Estimation – A Comparison Study | SpringerLink." https://link.springer.com/chapter/10.1007/978-3-319-04960-1_32 (accessed Nov. 13, 2021).

[7] E. Cuevas, D. Zaldívar, M. Pérez-Cisneros, H. Sossa, and V. Osuna, “Block matching algorithm for motion estimation based on Artificial Bee Colony (ABC),” *Appl. Soft Comput.*, vol. 13, no. 6, pp. 3047–3059, Jun. 2013, doi: 10.1016/j.asoc.2012.09.020.

[8] S. T. Khawase, S. D. Kamble, N. V. Thakur, and A. S. Patharkar, “An Overview of Block Matching Algorithms for Motion Vector Estimation,” Jun. 2017, pp. 217–222. doi: 10.15439/2017R85.

[9] K. Laidi, M. A. Bailiche, and M. Mehenni, “Comparative Study of Block Matching Techniques Used in Video Images Motions Estimation,” in *2007 5th International Symposium on Image and Signal Processing and Analysis*, Sep. 2007, pp. 29–34. doi: 10.1109/ISPA.2007.4383659.

[10] S. Safie, A. A. Samah, G. Sulong, H. A. Majid, R. Muhammad, and H. Hasan, “Block matching algorithm for moving object detection in video forensic,” in *2017 6th ICT International Student Project Conference (ICT-ISPC)*, May 2017, pp. 1–5. doi: 10.1109/ICT-ISPC.2017.8075330.

[11] M. S. Sri, B. Rajendra Naik, and K. Jayasankar, “Object Tracking using Motion Estimation based on Block Matching Algorithm,” in *2020 International Conference on Inventive Computation Technologies (ICICT)*, Feb. 2020, pp. 519–522. doi: 10.1109/ICICT48043.2020.9112511.

[12] T. Yokoyama, T. Iwasaki, and T. Watanabe, “Motion Vector Based Moving Object Detection and Tracking in the MPEG Compressed Domain,” in *2009 Seventh International Workshop on Content-Based Multimedia Indexing*, Jun. 2009, pp. 201–206. doi: 10.1109/CBMI.2009.33.

[13] J. Hernandez, H. Morita, M. Nakano-Miytake, and H. Perez-Meana, “Movement Detection and Tracking Using Video Frames,” in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, Berlin, Heidelberg, 2009, pp. 1054–1061. doi: 10.1007/978-3-642-10268-4_123.

[14] D. Boumazouza, Y. Sefouane, M. Djeddi, B. Khelouat, and K. Benatchba, “Bees for block matching,” in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, Nov. 2013, pp. 2390–2394. doi: 10.1109/IECON.2013.6699505.

[15] “A Comparison of Different Block Matching Algorithms for Motion Estimation - ScienceDirect.” https://www.sciencedirect.com/science/article/pii/S2212017313003356 (accessed Nov. 13, 2021).

[16] M. Chriqui and P. Sinha, “Survey of motion estimation techniques for video compression,” in *Low-Light-Level and Real-Time Imaging Systems, Components, and Applications*, Feb. 2003, vol. 4796, pp. 218–226. doi: 10.1117/12.452142.

[17] “A Comparison of Different Block Matching Algorithms for Motion Estimation - ScienceDirect.”https://www.sciencedirect.com/science/article/pii/S2212017313003356 (accessed Dec. 07, 2021).

[18] D. Chandradevi and M. Sundaresan, “Exhaustive block matching algorithm to estimate disparity between stereo images,” in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, Mar. 2016, pp. 3876–3881.

[19] V. Bafna and M. M. Mushrif, “Motion Estimation Algorithm in Video Coding,” in *Knowledge-Based Intelligent Information and Engineering Systems*, Berlin, Heidelberg, 2007, pp. 485–492. doi: 10.1007/978-3-540-74819-9_60.

[20] “Multiple block-size search algorithm for fast block motion estimation | IEEE Conference Publication | IEEE Xplore.” https://ieeexplore.ieee.org/abstract/document/5397582?casa_token=vFEqP1iaNRoAAAAA:ntg7_QPN5TzmHBwlHB8pyIGn9njuZ3HLmysXxfjgCju3Y2ujCgt92XkOqjwTUIjtYaS-XJu8wQ (accessed Dec. 07, 2021).

[21] “Full article: Detecting regional dominant movement patterns in trajectory data with a convolutional neural network.” https://www.tandfonline.com/doi/full/10.1080/13658816.2019.1700510 (accessed Nov. 13, 2021).

[22] A. Dominguez-Sanchez, M. Cazorla, and S. Orts-Escolano, “Pedestrian Movement Direction Recognition Using Convolutional Neural Networks,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3540–3548, Dec. 2017, doi: 10.1109/TITS.2017.2726140.

[23] “Multilayer and Multimodal Fusion of Deep Neural Networks for Video Classification | Proceedings of the 24th ACM international conference on Multimedia.” https://dl.acm.org/doi/abs/10.1145/2964284.2964297?casa_token=2l0Zr9rOFT8AAAAA:bXrmi54YiUTvgaPLfkQuP4CEOWu9ys3cuwy7VPEUPaxEZOkiP-asRCEHT4Dp_MFlWR1ukCmHn3g-AQ (accessed Dec. 07, 2021).

[24] “download.pdf.” Accessed: Dec. 07, 2021. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.224.2723

[25] M. N. Syed Shuja Hussain, “A ROBUST VIDEO STABILIZATION ALGORITHMS FOR GLOBAL MOTION ESTIMATION USING BLOCK MATCHING,” *Int. Trans. J. Eng.*, vol. Management, p. 11A06S: 111, 2020, doi: 10.14456/ITJEMAST.2020.119.

[26] C. Yang and G. Gidófalvi, “Detecting regional dominant movement patterns in trajectory data with a convolutional neural network,” *Int. J. Geogr. Inf. Sci.*, vol. 34, no. 5, pp. 996–1021, May 2020, doi: 10.1080/13658816.2019.1700510.

[27] W. Li and D. Powers, “Multiple Object Tracking Using Motion Vectors from Compressed Video,” in *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Nov. 2017, pp. 1–5. doi: 10.1109/DICTA.2017.8227469.

[28] Z. Wu, Y.-G. Jiang, J. Wang, J. Pu, and X. Xue, “Exploring Inter-feature and Inter-class Relationships with Deep Neural Networks for Video Classification,” in *Proceedings of the 22nd ACM international conference on Multimedia*, New York, NY, USA, Nov. 2014, pp. 167–176. doi: 10.1145/2647868.2654931.

[29] “Few-Example Video Event Retrieval using Tag Propagation | Proceedings of International Conference on Multimedia Retrieval.” https://dl.acm.org/doi/abs/10.1145/2578726.2578793 (accessed Nov. 14, 2021).

[30] A. D. Dongare, R. R. Kharde, and A. D. Kachare, “Introduction to Artificial Neural Network.”

[31] “Artificial Neural Network | SpringerLink.” https://link.springer.com/chapter/10.1007/978-1-4615-0377-4_5 (accessed Dec. 07, 2021).

[32] Y. Yuan, W. Xu, X. Yuan, L. Yan, and M. M. Deris, “Bio-Inspired Neural Networks for Block Based Motion Estimation,” *J. Algorithms Comput. Technol.*, vol. 8, no. 4, pp. 471–482, Dec. 2014, doi: 10.1260/1748-3018.8.4.471.

[33] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, “Deep convolution neural network for image recognition,” *Ecol. Inform.*, vol. 48, pp. 257–268, Nov. 2018, doi: 10.1016/j.ecoinf.2018.10.002.

[34] J. Lee, K. Kong, G. Bae, and W.-J. Song, “BlockNet: A Deep Neural Network for Block-Based Motion Estimation Using Representative Matching,” *Symmetry*, vol. 12, no. 5, Art. no. 5, May 2020, doi: 10.3390/sym12050840.

[35] S. Abu-El-Haija *et al.*, “YouTube-8M: A Large-Scale Video Classification Benchmark,” *ArXiv160908675 Cs*, Sep. 2016, Accessed: Dec. 07, 2021. [Online]. Available: http://arxiv.org/abs/1609.08675

[36] “Hand gestures for emergency situations: A video dataset based on words from Indian sign language - ScienceDirect.” https://www.sciencedirect.com/science/article/pii/S2352340920309100 (accessed Dec. 07, 2021).