**GESTURE RECOGNITION SYSTEM FOR CONVERTING SIGN LANGUAGE TO TEXT**

****

*A project report(Phase-1) submitted to*

*Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal*

*Toward partial fulfilment of*

*the degree of*

*Bachelor of Engineering*

*in*

*Computer Science & Engineering*

**Guided by:- Submitted by:-**

Dr. Urjita Thakar Aishwarya Gurjar(0801CE151010)

Professor, Divas Jain (0801CS151027)

Department of Computer Science & Engineering Mansi Agrawal(0801CS151043)

Romil Jain(0801CS151062)

Siddhant Gupta(0801CS151079)

**Department of Computer Science & Engineering**

**Shri Govindram Seksaria Institute of Technology & Science**

**Indore (M.P.)**

**2018-2019**

# **Shri Govindram Seksaria Institute of Technology & Science, Indore (M.P.)**



**CERTIFICATE**

This is to certify that the project report(Phase-1) entitled “GESTURE RECOGNITION SYSTEM FOR CONVERTING SIGN LANGUAGE TO TEXT AND AUDIO”, submitted by Aishwarya Gurjar, Divas Jain, Mansi Agrawal, Romil Jain, Siddhant Gupta students of final year B.E. (Computer Science and Engineering) in the year 2018-2019 of this institute, is a satisfactory account of their Project work (Phase-1) based on syllabus.

**Internal Examiner External Examiner**

**ACKNOWLEDGEMENT**

We express our profound sense of gratitude to **Dr. Urjita Thakar**, Professor, Computer Engineering and our project guide, who supported us to do the project work. Her continuous support and motivation helped us to bring forth our best. Having such guidance and expert device in this field has been an amazing experience and one of the main reason for the success of this phase of the project. We are grateful to her for providing necessary facilities to carry out our project work.

We would like to give our warm expression of thanks to **Dr. R.K. Saxena** (Director) and **Prof. D. A. Mehta**, (Senior Professor of Department) for providing all facilities and academic environment during the course of study.

We sincerely wish to express, our gratefulness to all the faculty and staff members of Computer engineering department who extended their cooperation at all the times and have contributed to the successful completion of the project

The successful completion of the project is not an individual effort. It is an outcome of the cumulative effort of a number of people, each having their own importance to the objective.

We would like to show our gratitude towards our **Parents** for their support, motivation and lessons.

With a blend of gratitude, pleasure and great satisfaction we convey our indebtedness to all those who have directly contributed to the successful completion of the project work.

**Regards,**

Aishwarya Gurjar

Divas Jain

Mansi Agrawal

Romil Jain

Siddhant Gupta

**CONTENTS**

**Chapter No. Title Page No.**

Abstract

Chapter 1 INTRODUCTION

1.1 Preface 1

1.2 Need of the Project 1

1.3 Problem Statement 2

1.4 Objectives 2

1.5 Proposed Approach 2

1.6 Organisation of Report 3

Chapter 2 LITERATURE REVIEW

2.1 Related Work 4

2.1.1 Implemented Algorithms 5

2.2 Background Study 7

2.2.1 Hand Gesture Recognition 7

2.2.2 Challenges of hand 8

Gesture Recognition

2.2.3 Application of Hand 9

Gesture Recognition

2.2.4 Convolutional Neural Network 10

2.2.5 Recurrent Neural Network 15

2.2.6 Long Short Term Memory 16

Chapter 3 ANALYSIS

3.1Existing System 20

3.2 Detailed Problem Statement 21

3.3 Requirement Analysis 22

3.3.1 Functional Requirement 22

3.3.2 Non-Functional Requirement 23

3.4 Resource Requirement 24

3.4.1 Hardware Requirement 24

3.4.2 Software Requirement 25

3.5 Feasibility Study 27

3.6 Use-Case Diagram 29

3.7 System Flow Diagram 30

Chapter 4 DESIGN

4.1 System Architecture 31

4.2 Video Preprocessing 32

4.3 Processing frames in 3D CNN 32

4.4 Processing frames in RNN 33

4.5 Using Softmax 33

4.6 Mean Prediction 33

Conclusion 34

References 35

**List of Figures**

|  |  |  |
| --- | --- | --- |
| Figure No | Title | Page No |
| 2.1 | Image of Immersion CyberGrasp sensor | 3 |
| 2.2 | Flowchart of vision based recognition system | 4 |
| 2.3 | Image of VPL Data Glove | 5 |
| 2.4 | Architecture of Nvidia Approach | 10 |
| 2.5 | System model of Nvidia | 11 |
| 3.1 | Proposed Data Format | 15 |
| 3.2 | Use-Case Diagram | 17 |
| 3.3 | System Flow Diagram | 18 |
| 4.1 | Architecture of Proposed approach | 19 |
| 4.2 | Architecture of CNN | 20 |
| 4.3 | Convolution Edge Detection | 21-22 |
| 4.4 | Convolutions on RGB Images | 22 |
| 4.5 | Retention of important/dominant features | 23 |
| 4.6 | Model of RNN | 25 |
| 4.7 | Model of LSTM | 27 |
| 4.8 | Figure of forget gate operation | 27 |
| 4.9 | Figure of update gate operation | 28 |
| 4.10 | Figure of output gate operation (Part 1) | 28 |
| 4.11 | Figure of output gate operation (Part 2) | 29 |

**Abstract**

A real-time sign language translator is an important milestone in facilitating communication between the deaf community and the general public. Being able to recognize sign language is an interesting machine learning problem while simultaneously being extremely useful for deaf people to interact with people who don’t know how to understand American Sign Language. The Hand Gesture Recognition System is highly desired due to its ability to overcome the barrier between deaf and hearing people. At present, a robust recognition system is still unavailable in the real world due to numerous obstacles such as external hardware requirements. Additionally, as we know, Hand Gesture recognition has emerged as one of the most important research areas in the field of human computer interaction (HCI).

**CHAPTER 1**

**INTRODUCTION**

This chapter gives a basic outline of the project describing the problem statement, the objectives the need of the project, a brief description of the solution approach and details about the organization of the project report.

**1.1. Preface:**

The only way the speech and hearing impaired (i.e dumb and deaf) people can communicate is by sign language. Sign language can be understood as a kind of gesture. Gestures are naturally used by many peoples and are especially major and natural way of interaction for deaf/blind people. The main problem of this way of communication is, people who cannot understand sign language and gestures cannot communicate with these people or vice versa. The project aims to bridge the gap between the speech and hearing impaired people and the normal people. The basic idea of this project is to make a gesture recognition system through which dumb people can significantly communicate with all other people using their normal gestures. Gesture recognition is to recognise specific human gestures and process them. Building gesture recognition system means building computer keen to understand human language, thus building a human computer interaction. Hand gesture recognition system has been applied for different applications on different domains such as Sign Language Recognition, Robot Control, Graphic Editor, Virtual Environment, Number Recognition, Television, 3D Modeling. The project uses image processing system to identify, especially American Sign Language(ASL) used by the deaf people to communicate and converts it into text so that normal people can understand. The system will use RGB camera rather than expensive 3D cameras and therefore cost effective.

**1.2. Need of the project:**

Hand gesture recognition systems are becoming important in various user interfaces. Various computer vision algorithms have employed color and depth cameras for hand gesture recognition, but a robust classification of gestures from different subjects performed under widely varying lighting conditions is still challenging. Nvidia has proposed an algorithm for hand gesture recognition using depth and intensity values using 3D convolutional neural networks. It has no time sequencing i.e. it considers only spatial features and does not consider the temporal features. Softmax, an activation function classifies the output of 3D CNN into different categories of gestures and assigns them probability values. The Nvidia’s approach has an overhead of dividing the video into chunks which need to be fed in CNN. Therefore, focus is to build a system on vision-based framework, that can be developed to allow users to interact with computers through hand gestures at a lower cost and without any additional hardware requirements. The proposed system focuses on implementing hand gesture recognition and overcoming limitations of Nvidia’s system by including Recurrent Neural Network layers after the CNN layers, in order to specify the time sequence of the features extracted from the CNN part.

**1.3. Problem statement:**

Gesture Recognition System for converting sign language to text using video analysis.

**1.4. Objective:**

To build a real time gesture classification system that can detect sign symbols using a real time gesture based system developed to identify gestures. To build a system simple, easy and user friendly without using any special hardwares, where all the computations will occur on single PC or workstation.

**1.5. Proposed Approach:**

The proposed system intend to use the complete video, then converting the video into frames. These frames are given as input into the 3D CNN which extracts features from different frames. These extracted features are then fed into N different RNN which are used to learn the time sequence of the gestures.

The system is composed of five major part :

1. Video Preprocessing
2. Processing frames in the 3D CNN
3. Processing the Frames in RNN
4. Using Softmax
5. Mean Prediction for the best result

The system will detect, recognize and interpret the hand gestures through computer vision. The system will use RGB camera rather than expensive 3D cameras and sensors which are being used.

A frame from the webcam is captured. The image is segmented into two parts, both of them are manipulated simultaneously before analysing the resultant data, skin pixels and moving patterns are detected. A new image is created containing the location of the center of the moving hand. Ten latest consecutive frames are tracked continuously, in each frame the centers of the moving hands are detected. Pattern Recognition: through the user's hands motion, the features are compared with those stored in the database, the maximum likelihood correspondence is chosen.

**1.6. Organisation of Report:**

The main body of the report is preceded by detail contents , including the list of figures, followed by units used in the report. This is followed by executive summary giving briefly the scope and objectives of the study, importance of the topic, methodologies, strategies and algorithms used to solve the problem.

**CHAPTER-2**

**Literature Review And Background Study**

In this chapter, the literature work related to hand gesture recognition are studied. The previously implemented systems, along with their limitations are included in chronological order. In addition, the machine learning concepts required to implement the system are described in detail.

**2.1 Related Work**

Research has been limited to Literature Review small scale systems able of recognizing a minimal subset of a full sign language. Christopher Lee and Yangsheng Xu [1] developed a glove-based gesture recognition system that was able to recognize 14 of the letters from the hand alphabet, learn new gestures and able to update the model of each gesture in the system in online mode, with a rate of 10Hz. Over the years advanced glove devices have been designed such as the Sayre Glove, Dexterous Hand Master and PowerGlove [2]. The most successful commercially available glove is by far the VPL DataGlove as shown in figure 2.3.



Figure 2.3 Image of VPL Data Glove

It was developed by Zimmerman [3] during the 1970’s. It is based upon patented optical fiber sensors along the back of the fingers. Star-ner and Pentland developed a glove-environment system capable of recognizing 40 signs from the American Sign Language (ASL) with a rate of 5Hz. Hyeon-Kyu Lee and Jin H. Kim [4] presented work on real-time hand-gesture recognition using HMM (Hidden Markov Model) . Kjeldsen and Kendersi [5] devised a technique for doing skin-tone segmentation in HSV space, based on the premise that skin tone in images occupies a connected volume in HSV space. They further developed a system which used a backpropagation neural network to recognize gestures from the segmented hand images. Etsuko Ueda and Yoshio Matsumoto [6] presented a novel technique a hand-pose estimation that can be used for vision-based human interfaces, in this method, the hand regions are extracted from multiple images obtained by a multi viewpoint camera system, and constructing the “voxel Model.” Hand pose is estimated. Chan Wah Ng, Surendra Ranganath[7] presented a hand gesture recognition system, they used image furrier descriptor as their prime feature and classified with the help of RBF network . Their system’s overall performance was 90.9%. Claudia Nölker and Helge Ritter [8] presented a hand gesture recognition model based on recognition of finger tips, in their approach they find full identification of all finger joint angles and based on that a 3D model of hand is prepared and using neural network.

The author Zhang Yuye et.al. [9] System use AT89C51 and CAN BUS controller which leads to complicated design and cost of the system more because of CAN BUS controller. Also in this case power requirement will be more in case of AT89C51.

The author Cai Bai-gen et.al.[11] designed a vehicle detection system based on magneto-resistive sensor is composed by wireless traffic information collection nodes which are set on two sides of road to detect vehicle signal. The magneto-resistive sensor is expensive and maintenance cost of the system will be more if the system fails.

The author Manoj Kanta Mainali et.al.[10] proposed a genetic algorithm approach to estimate the traffic volume in road sections without the traffic information of road sections. This method estimates the unknown traffic volume using only the known traffic volumes.

**2.1.1 Implemented Algorithms**

The most commonly used hand gesture recognition algorithms include HMM (Hidden Markov Model) model based on statistics [12], algorithm based on genetic algorithm [13] and artificial neural network [14]. Statistics based HMM takes advantage of causal relationship between visual features obtained from prior knowledge to deal with the inherent problem of uncertainty in video processing, not only can build the probabilistic modeling of dependencies among different features corresponding to multiple random variables in every moment, but also consider the transition probability between every moment, which can well reflect the timing relationships between features. However, it needs to maintain a certain size of the sample library and ask for large computational quantity, even though the larger the size of the sample library, the closer to the actual situation and the higher the accuracy of hand gesture recognition will be, moreover, this method also needs data smoothing technology to enlarge the value of small probability. The genetic algorithm is used to discretize images, control the discrete points and then convert image recognition problems into combinatorial optimization problems of a series of discrete points. But it is not able to take use of the feedback information from network in time, in addition, this method is troubled with slow search speed, large training sample and long training time. Artificial neural network has a large number of simple processing units (neurons) which are widely connected to form a complex information processing network, it mimics the function of information processing, storage and retrieval of human brain neural system in a certain extent and level, its processing efficiency is high with small samples can be satisfied, but the training process needs the participation of human, and the accuracy of recognition is influenced by the subjective factors.

Many techniques on HOG (Histogram of Gradient) like [15] have been proposed in the past which employ edge and gradient based descriptors for hand gesture recognition. But they are only able to detect hand gestures in a simple background and are liable to fail when the background is cluttered. L. G. Zhang, J. Q. Wu [16] extracted the edge pixels of hand gesture, took use of the idea of model-based matching using Hausdorff distance to realize the recognition of Chinese alphabet, the method proposed had advantages of small computation and strong adaptability but disadvantages of ignoring the situation of rotation, scaling and skin color interfere. The H. B. Ren, G. Y. Xu [17] put up with a method based on hand characteristic curves, the result of combination of color, motion and edge information, this method can reduce the dependence on hand segmentation, but the computation is too complex, and the real-time performance is poor.

In addition, there are also many other research works that successfully harvest some certain achievement of hand gesture recognition. Y. Cui, D.L. Swets and J.J. Weng [18] used the maximum difference feature to classify the gestures after the segmentation of MDF (The Most Discriminating Feature) space, the algorithm can adapt to the occasion with complex background. In [19] an inductive learning system was introduced, this system could extract rules from DNF (Disjunctive Normal Form), and it obtained the recognition rate of 94%. The elastic curve matching method introduced by J. Triesch, C. V. D. Malsburg [20] was less independent on segmentation, and could reach the recognition rate of 85% in a complex background. Bjorn et al. [21] used color and motion features to detect and track the hand, combined the method of template matching and nearest neighbor classifier to recognize hand gesture. In [22] the authors divided the hand into 21 different regions and train a SVM classifier to model the joint distribution of these regions for various hand gestures so as to classify the gestures.

**2.2 Background Study**

**2.2.1 Hand Gesture Recognition**

Hand gesture recognition is generally divided into static gesture recognition and dynamic gesture recognition, static gesture recognition is the recognition of hand shape, read out the meaning of hand expression, and dynamic hand gesture recognition is the recognition of hand motion trajectory in space, and then perform the corresponding operation based on obtained trajectory parameters, such as for the playing courseware on the projection, hand gestured can be used to flip up and down, pause, start, etc. The traditional gesture recognition was through the use of wearable technology, allowing users to do some hand gestures with special data gloves on, the data gloves would transfer user’s gestures and location information to the computer and help it comprehend the gestures and behaviors of uses. Figure 2.1 shows a multi-function virtual reality device composed of many sensors on the glove called Immersion CyberGrasp. Through the software mapping, the virtual objects can be shifted, clutched and rotated by the glove with the ability of “reach into the computer”.

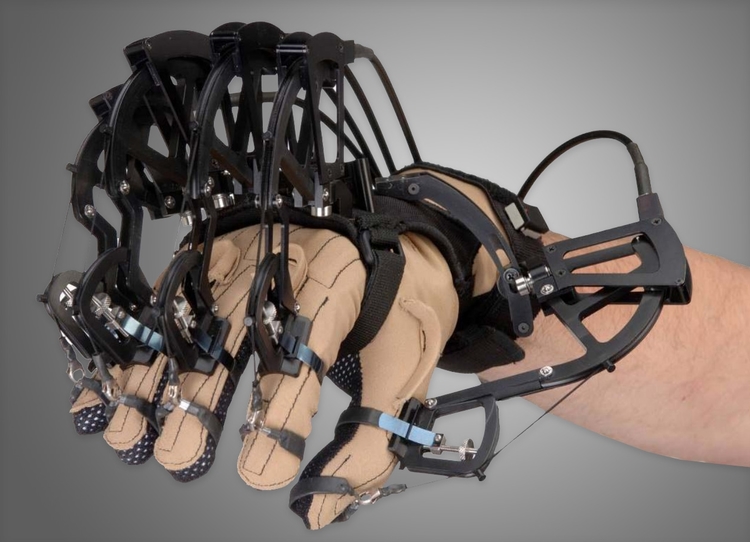
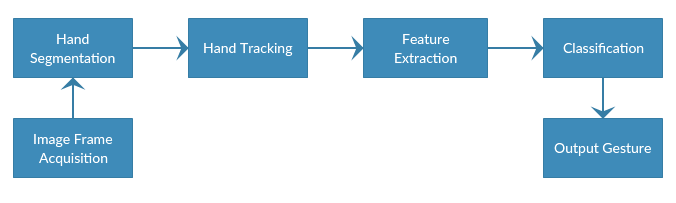


Figure 2.1 Image of Immersion CyberGrasp sensor

The glove can transmits hand gesture to the computer in real time accurately, and then receives feedback from the virtual environment to the operator. It provides users with a direct and universal human-computer interaction mode with advantages of high accuracy, simple data and fast processing speed, etc., but because of the shortcomings of expensive equipment, inconvenient operation, and not suitable for long-distance control, this kind of interaction model is hard to get promotion.



**Figure 2.2** Flowchart of vision based recognition system

Vision based gesture recognition as shown in figure 2.2 takes use of the camera to capture hand gestures to system, and after image preprocessing such as detection and segmentation to extract features of extracted image sequence to understand and describe its behavior. When one or more cameras obtain the video stream of user gestures, the system will monitor whether there are hand gestures contained in the data stream according to the interactive mode of gesture, if there are, separates them. Then choose appropriate method to detect and extract features, and choose appropriate classifier to recognize the gesture in current image. The greatest advantage of vision based hand gesture recognition is that the input is simple with lower dependence on equipment, and it is in line with the people’s daily interaction, therefore, vision based hand gesture recognition is bound to be the new pursuit of human-computer interaction.

**2.2.2 Challenges of Hand Gesture Recognition**

At present, although the study of hand gesture recognition has made great progress and achieved high recognition rate in different areas, it is still facing many challenges, such as: extraction of invariant features, transition model between gestures, minimal sign language recognition units, automatic segmentation of recognition units, recognition approach with scalability about vocabularies, auxiliary information, signer independent and mixed gestures recognition, etc. Thereinto, the static gesture recognition based on vision is the current trend of hand gesture recognition and mainly has the following two aspects of technical difficulties:

1. Difficulties in target detection

Target detection is to capture the target from the image stream in a complex background, which is to extract the object of interest. In vision based hand gesture recognition methods, it is always a difficult problem to divide the human hand area and other background areas in the image, this is mainly due to the variety of background and unforeseen environment factors.

2. Difficulties in target recognition

Hand gesture recognition is to explain the high-level implications contained according to the posture and changing process of hand. In view of the following characteristics of hand gesture, the key technology of hand gesture recognition is to extract the geometric invariant features.

1) Hand is an elastic object, there may big differences exist between same gestures and high similarity between different gestures. Human hand has more than 20 degrees of freedom, its movement is very flexible and complex. Therefore, same gestures made by different people may vary, and gestures made by the same person at different time or place may also different.

2) Hand has a lot of redundant information, since the key part of hand gesture recognition is to identify finger features, so palm feature is one of the redundant information.

3) The position of hand refers to the projection of hand from three-dimensional space to two-dimensional, so the projection direction is really related.

4) It is easy to produce shadows due to the non-smooth surface of hand. Because of these characteristics of hand, the above two problems have not been solved well yet, so it is necessary to add some restrictions when doing hand gesture recognition.

**2.2.3 Applications of Hand Gesture Recognition**

Hand gestures recognition system has been applied for different applications on different domains, as mentioned in [23][25] including; sign language translation, virtual environments, smart surveillance, robot control, medical systems etc. overview of some hand gesture application are sare listed below[23][24].

* Sign Language Recognition:Since the sign language is used for interpreting and explanations of a certain subject during theconversation, it has received special attention [23]. A lot of systems have been proposed to recognize gestures using different types of sign languages [24]. For example [24] recognized American Sign Language ASL using boundary histogram, MLP neural network and dynamic programming matching. [29] recognized Japanese sign language JSL using Recurrent Neural Network, 42 alphabet and 10 words. [30] recognized Arabic Sign language ASL using two different types of Neural Network, Partially and Fully Recurrent neural Network.
* Robot Control:Controlling the robot using gestures considered as one of the interesting applications in this field[33]. [26] proposed a system that uses the numbering to count the five fingers for controlling a robot using hand pose signs. The orders are given to the robot to perform a particular task [26],where each sign has a specific meaning and represents different function for example, “one”means “move forward”, “five” means “stop”, and so on.
* Graphic Editor Control:Graphic editor control system requires the hand gesture to be tracked and located as a preprocessing operation [23]. [31] used 12 dynamic gestures for drawing and editing graphic system. Shapes for drawing are; triangle, rectangle , circle, arc, horizontal and vertical line for drawing, and commands for editing graphic system are; copy, delete, move, swap, undo, andclose [31].
* Virtual Environments ( VEs):One of the popular applications in gesture recognition system is virtual environments VEs,especially for communication media systems [25]. [27] provided 3D pointing gesture recognition for natural human computer Interaction HCI in a real-time from binocular views. The proposed system is accurate and independent of user characteristics and environmental changes [27].
* Numbers Recognition: Another recent application of hand gesture is recognizing numbers. [32] proposed an automatic system thatvision-basedate and recognize a meaningful gesture from hand motion of Arabic numbers from 0 to 9 in a real time system using HMM.
* Television Control:Hand postures and gestures are used for controlling the Television device [25]. In [28] a set of hand gesture are used to control the TV activities, such as turning the TV on and off, increasing and decreasing the volume, muting the sound, and changing the channel using open and close hand [28].

3D Modeling To build 3D modeling, a determination of hand shapes are needed to create, built and view 3D shape of the hand [25]. Some systems built the 2D and 3D objects using hand silhouette [25]. 3D hand modeling can be used for this purpose also which still a promising field of research [25].

**2.2.4 Convolutional Neural Network**

CNN is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery. They have wide applications in image and video recognition, recommender systems and natural language processing.

CNNs, like neural networks, are made up of neurons with learnable weights and biases. Each neuron receives several inputs, takes a weighte+d sum over them, pass it through an activation function and responds with an output.The Fundamental block of CNN is convolution operation.It is a layer based architecture.

**CNN Architecture**

Convolutional Neural Networks have a different architecture than regular Neural Networks. Regular Neural Networks transform an input by putting it through a series of hidden layers. Every layer is made up of a set of neurons, where each layer is fully connected to all neurons in the layer before. Finally, there is a last fully-connected layer,the output layer  that represent the predictions.

Convolutional Neural Networks are a bit different. First of all, the layers are organised in 3 dimensions: width, height and depth. Further, the neurons in one layer do not connect to all the neurons in the next layer but only to a small region of it. Lastly, the final output will be reduced to a single vector of probability scores, organized along the depth dimension.CNN has two components:

**1) The Hidden layers/Feature extraction part**

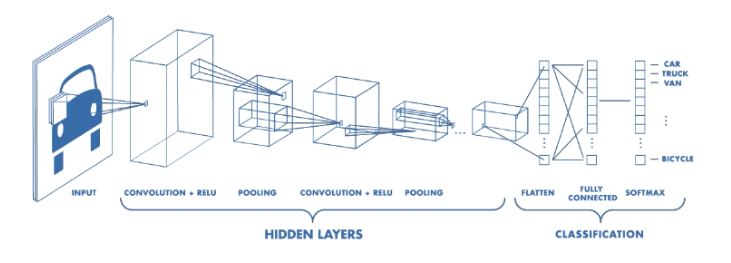
In this part, the network will perform a series of convolutions and pooling operations during which the features are detected.In case of a object like this is the part where the network would [recognise its stripes, two ears, and four legs](https://distill.pub/2018/building-blocks/).This part comprises of two types of layers:

* Convolution Layer
* Pooling Layer

**2) The Classification part**

The fully connected layers will serve as a classifier on top of these extracted features. They will assign a probability for the object on the image being what the algorithm predicts it is.This part comprises of one layer:

* Fully Connected layer

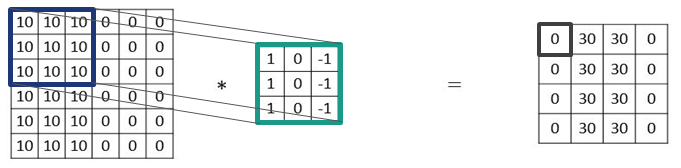
****

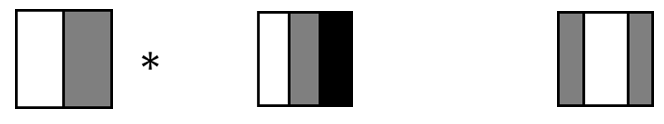
**Figure 2.4 Architecture of a CNN**

**1. Convolution Layer**

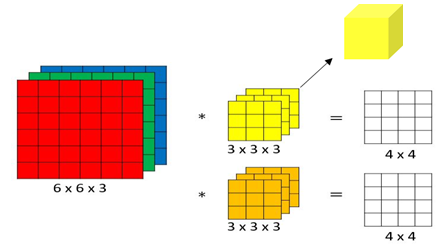
Convolution is one of the main building blocks of a CNN. The term [convolution](http://timdettmers.com/2015/03/26/convolution-deep-learning/) refers to the mathematical combination of two functions to produce a third function. It merges two sets of information.

In the case of a CNN, the convolution is performed on the input data with the use of a filter or kernel (these terms are used interchangeably) to then produce a feature map.The convolution is then executed by sliding the filter over the input. At every location, a matrix multiplication is performed and sums the result onto the feature map.

****

****

**Figure 2.5 Convolution (Edge Detection)**

****

**Figure 2.6 Convolutions on RGB images**

### **2. Pooling Layer**

### A pooling layer is another building block of a CNN. Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network and hence it results in speed gain of the algorithm. Pooling layer operates on each feature map independently. The most common approach used in pooling is **max pooling.**

### *.*

**Figure 2.7 Retention of important/dominant features**

**3.Fully Connected Layer**

After the convolution and pooling layers, the classification part consists of a few fully connected layers.Neurons in a fully connected layer have full connections to all the activations in the previous layer. This part is in principle the same as a regular Neural Network.These fully connected layers can only accept 1 Dimensional data.The feature vector is unrolled from a 3D volume into a 1D vector and finally fed into a softmax function which is used for classification.[36]

**Features of Convolution operation:**

Parameter sharing: A feature detector (such as a vertical edge detector) that’s useful in one part of the image is probably useful in another part of the image.

Sparsity of connections: In each layer, each output value depends only on a small number of inputs.

**Why 3D Convolutional Neural Network**

* 2D CNN only captures spatial features across each depth(for RGB images).
* 3D CNN captures the time sequence of the frames in the video that represents a particular gesture.
* In other words, the temporal aspect of the data is incorporated as the third dimension in the 3D CNN

**Difference between a CNN and a Regular Neural Network**

* Able to learn more relevant features(very good feature extractors)
* They consider the context/shared information in the small neighborhoods i.e. they capture local information. This future is very important in many applications such as image, video, text, and speech processing/mining as the neighboring inputs (eg pixels, frames, words, etc) usually carry related information.
* It supports weight sharing and hence reduce the number of units in the network.This means, there are fewer parameters to learn which reduces the chance of overfitting as the model would be less complex than a fully connected network.
* Reduces time and space complexity

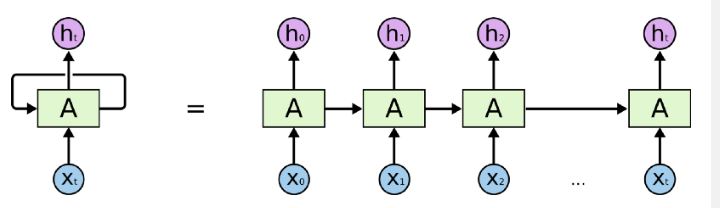
**How are the problems tackled ?**

1. The RNN has helped in finding the time sequence that was a problem in the previous model.
2. The video is now not divided into chunks in the 3D CNN phase, so the overall video can be considered a single gesture and the prediction can be much better.

**Why do we Need RNN?**

The problem with other models is that they perform poorly when given a sequence of data. An example of sequence data is an audio or a video gesture clip which contains a sequence of spoken words or an action. Another example would be a video of a gesture which contains a number of frames. Feedforward networks and CNN take a fixed length as input, but, when you look at the number of frames in a videos, not all are of the same length. You could overcome this issue by padding all the inputs to a fixed size. However, they would still perform worse than an RNN because those conventional models do not understand the context of the given input. This is where the major difference between sequence models and feedforward models lies. Given a video, when looking at a frame, sequence models try to derive relations from the previous frame in the same video. This is similar to how humans think as well. When we are reading a sentence, we don’t start from scratch every time we encounter a new word.[36]

### **2.2.5 Recurrent Neural Network(RNN)**

The recurrent neural network is represented as shown in the below figure. Each node at a time step takes an input from the previous node and this can be represented using a feedback loop. We can unfurl this feedback loop and represent it as shown in the figure below. At each time step, we take an input x\_i and a\_i-1(output of the previous node) and perform computation on it and produce an output h\_i. This output is taken and given to the next node. This process continues until all the time steps are evaluated.

**Figure 2.8** Model of RNN

**Problems with Simple RNN Network**

Vanishing Gradient Problem

Sometimes, we only need to look at recent information to perform the present task.

* “The cat eat and *was* full”
* It is pretty obvious that the next word will be .
* “The cat ate the food and went for a walk in the garden because it (*was/were)* full.”

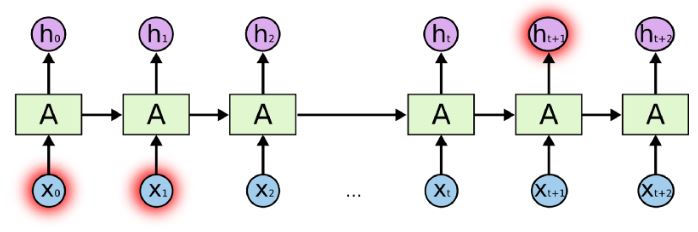
Recent information suggests that the next word is probably the name of a language, but which language?

Inorder to solve the of Vanishing Gradient problem a new type of RNN is needed.[39]

**2.2.6 Long-Short Term Memory**

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems.

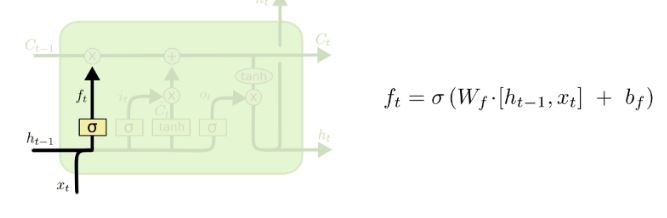
The disadvantage with RNN is that as the time steps increase, it fails to derive context from time steps which are much far behind.

****

**Figure 2.9** Model of LSTM

The structure of an LSTM network remains the same as an RNN, whereas the repeating module does more operations. Enhancing the repeating module enables the LSTM network to remember long-term dependencies. Let’s try to break down each operation which helps the network remember better.

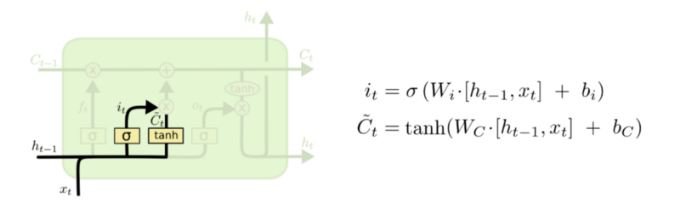
#### **Forget gate operation**

****

**Figure 4.8** Figure of forget gate operation

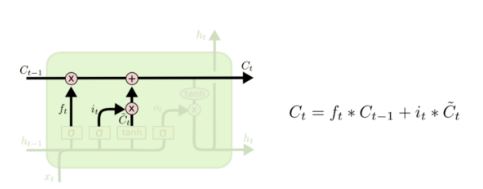
We take the input from current time step and the learned representation from previous time step and concatenate them. We pass the concatenated value into a sigmoid function which outputs a value(f\_t) between 0 and 1. We do an element-wise multiplication between f\_t and c\_t-1. If a value is 0, then it is eliminated from c\_t-1, if the value is 1, then it is completely let through. Therefore, this operation is also called “Forget gate operation”.

#### **Update gate operation**

**Figure 2.10** Figure of update gate operation

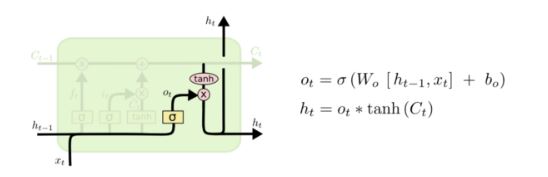
The above figure represents the “Update gate operation”. We concatenate values from current time step and the learned representation from previous time step. By passing the concatenated values through a tanh function we generate candidate values and by passing it through a sigmoid function we choose which values to be selected from the candidates. The chosen candidate values are updated to c\_t-1.

#### **Output gate operation**

****

**Figure 2.11** Figure of output gate operation (Part 1)

We concatenate values from current time step and the learned representation from previous time step and pass it through a sigmoid function to choose which values we are going to use as the output. We take the cell state and apply a tanh function and do an element-wise operation which lets through only the selected outputs.[38]



**Figure 2.12** Figure of output gate operation (Part 2)

Now, this is a lot of operations to be done in a single cell. When using a bigger network, the training time would significantly increase compared to an RNN. There is an alternative to LSTM if want to reduce your training time but also use a network that remembers long term dependencies.

**Chapter 3**

**Analysis**

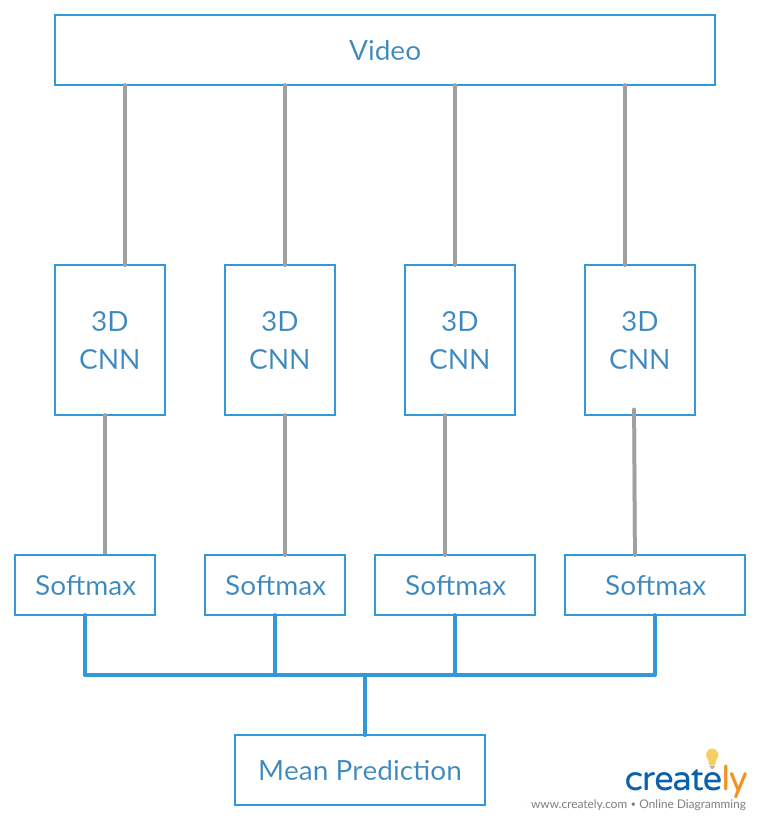
In this chapter, the existing system has been analysed and includes the functional and non-functional requirements of the proposed system. The software tools and hardware components required to operate the system are specified along with the feasibility of the system.

**3.1 Existing Systems**

Hand gesture recognition systems are becoming important in various user interfaces. Various computer vision algorithms have employed color and depth cameras for hand gesture recognition, but a robust classification of gestures from different subjects performed under widely varying lighting conditions is still challenging.

**3.1.2 Nvidia Approach**

Nvidia has proposed an algorithm for hand gesture recognition using depth and intensity values using 3D convolutional neural networks. Nvidia introduced a hand gesture recognition system that utilizes depth and intensity channels with 3D convolutional neural networks.[1]

****

**Figure 3.1** Architecture of Nvidia Approach

**Initial Tasks**

Major challenges involved in the above approach are breaking the clips into frames and changing the resolution of frames to the desired resolution.

**Limitations of Nvidia’s Approach**

It has no time sequencing i.e. it considers only spatial features and does not consider the temporal features. Softmax is classifying the chunks individually which are then being averaged to get the final gesture. It has an overhead of dividing the video into chunks to be fed in CNN.

**3.2 Detailed Problem Statement**

Human gestures have long been an important way of communication, adding emphasis to voice messages or even being a complete message by itself. Such human gestures could be used to improve the human-machine interface and also for specially abled people to communicate with others. A vision-based framework can be developed to allow users to interact with computers through human gestures at a lower cost and without any additional hardware requirements. This study focuses on understanding such human gesture recognition, typically hand gesture recognition generally involves various stages like video acquisition, background subtraction, feature extraction, and gesture recognition. In the proposed approach, the long-term recurrent convolutional network is used to classify the video sequences of hand gestures. In the standard long-term recurrent convolutional network-based action classifier, multiple frames sampled from the video sequence are given as an input to the network, to perform classification. However, the use of multiple frames increases the computational complexity, apart from reducing the classification accuracy of the classifier. To address these issues the system extracts a fewer representative frame from the video sequence and inputting them to the long-term recurrent convolutional network.

**3.3 Requirement Analysis**

**3.3.1 Functional Requirement**

The system would incorporate the following functional requirements :

**1) Recording the gesture in form of video**

The software will use an RGB Camera to record the gesture in the form of videos. The system will capture particularly the hand movements and the facial expressions.

**2) Segmentation of video into frames**

The software will divide the captured video of the gesture into the number of frames( approximately 80 frames for a video of 3 seconds ). The relevant frames having a symbolic difference from the reference frame will be accounted for the next phase.

**3) Translating input**

The system will translate the high-quality frames(1024 x 2048 pixel) into the lower-quality( 100 x 100 pixel) frame by lowering the resolution of each frame.

**4) Extracting features**

The goal of feature extraction is to find the most discriminating information in the recorded images. Feature extraction operates on two-dimensional image arrays but produces a list of descriptions or a “feature vector”. The feature vector will be sent in the next phase.

**5) Recognition of Gesture**

The system will use the feature vectors for the recognition of the gesture. After recognizing the gesture, the system will convert the gestures into textual form and will display as an output to the user.

**3.3.2 Non Functional Requirement**

The system will incorporate the following non-functional requirements :

**1) Efficiency in Computation**

This software will minimize the use of Central Processing Unit (CPU), Graphics Processing Unit (GPU) and memory resources on the operating system. When the system is training, the software will utilize less than 80% of the system’s CPU since almost all computations are performed on GPU and less than 100 megabytes of system memory.

**2) Extensibility**

The software will be extensible to support future developments and add-ons to the software. The gesture control module shall be at least 50% extensible to allow new gesture recognition features to be added to the system.

**3) Portability**

The software will be 100% portable to all operating platforms that support Python 3. Therefore, this software should not depend on the other system configuration.

**4) Performance**

This software will minimize the number of calculations needed to perform image processing and hand gesture detection. Each captured video frame shall be processed within 350 milliseconds to achieve 12 frames per second performance.

**5) Reliability**

The software will be operable in all lighting conditions. Regardless of the brightness level in the user’s operating environment, the program will always detect a user’s gesture if they are performed right in front of the RGB camera.

**6) Usability**

This software will be easy to use for all users with minimal instructions. 100% of the

languages on the graphical user interface (GUI) will be intuitive and understandable by the user.

7) **Accuracy**

This software will be modification of Nvidia’s Hand Gesture Recognition System which has an accuracy of 71.5% over a dataset of 1500 videos. The proposed system will have target accuracy of 75%.

**3.4 Resource Requirement**

**3.4.1 Dataset**

The system will work on “The American Sign Language Lexicon Video Dataset”. The number and type of signs included in the dataset are similar in scale and scope to the set of lexical entries in existing English-to-ASL dictionaries [2, 3, 4, 5]. The dataset has already at least one video example per sign from a native signer, for almost all of the 3,000 signs contained in the Gallaudet Dictionary of American Sign Language [5].

In regular ASL discourse, signers sometimes use fingerspelling, i.e., spell out English words (typically proper nouns, technical terms, or borrowings) using the letters of the manual alphabet. With the exception of some commonly used signs composed of such letters, frequently referred to as “loan signs,” fingerspelled items would not normally be included in an ASL dictionary [2, 3, 4, 5], and will not be included in the lexicon dataset.

**3.4.1 Hardware Requirements**

The software does not require special equipment except for a personal computer (PC) and an RGB camera. The specification of the RGB camera and the PC is mentioned below :

1. **RGB Camera**
   1. Image format should be analogous to an aerial film image at a format of 23 cm x 15 cm, scanned at 15 μm.
   2. Image data format should be JPEG; TIFF with options for 8, 12 and 16 bits, scan-line, stripped or tiled.
   3. Image storage format in level 2 should be full resolution panchromatic with separate color channels at color resolution. Color at level 3 should be full resolution R, G, B, Near-IR channels, planar or pixel-interleaved.
   4. Color (multi-spectral capability) : 4 channels -- RGB & NIR
   5. Color image size : 4810 \* 3140 pixels
   6. Color physical pixel size : 7.2 μm [6]
2. **GPU: Nvidia GeForce 940m 2 GB dedicated** 
   1. The CPU of the computer should have at least two cores and a GPU with dedicated RAM to handle the enormous amount of calculations needed for the image processing unit.
   2. GeForce 940M is Maxwell-based and uses GM108 chip.
   3. Memory bandwidth: 14.40 GB/sec
   4. Maximum frequency: 1176 MHz
   5. Power consumption: 33 W
   6. Number of cores: 384
   7. Memory bus: 64-bit

**3.3.2 Software Requirements**

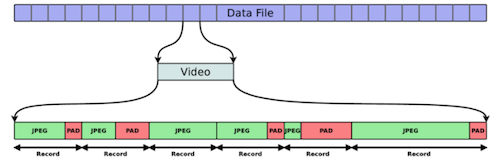
* **Prefered Data Format**

In the case of Image datasets, the image data set is usually small enough to fit in file system cache. During training, the initial epoch would be a little bit slower because of the data reading from the disk but the next iterations will not be the bottleneck.

But in the case of Video datasets or Video Recognition, video data is itself large comparative to image data and when thousands of videos are there, this will not fit into file system cache. So in each iteration, the system has to read data over and over again and this would become the bottleneck.

The system refers to a data format in which-

* All the frames of a video are concatenated as a JPEG image into a single data-file and save the offsets and metadata in a secondary meta-file.
* Using JPEG compression that reduces the file size significantly
* An important feature of the format is that multiple videos can be stored in a single chunk.
* Since all videos and all frames stored sequentially, it circumvents many delays caused by seeking from magnetic storage.

**Figure 3.1 Data Format**

* **Software Tools Used :**

1. **vPython 3.x**

In the VPython 3.x environment, VPython programs run on a local server, using standard Python, and output is sent to a browser, where the GlowScript graphics library is used to display the 3D animation. In the GlowScript environment, VPython programs are compiled in the browser itself by the RapydScript Python-to-JavaScript compiler, and the program is run in the browser.

The GlowScript libraries are based on WebGL and use GPU (Graphics Processing Unit) hardware. Use of the GPU makes possible much more sophisticated 3D graphics than the CPU-based OpenGL graphics library used in Classic VPython can provide. Thus the system will make efficient use of GPU.[7]

1. **PyTorch: 0.4.0**

PyTorch uses an imperative/eager paradigm. That is each line of code required to defines a component of that graph. We can independently perform computations on these components itself, even before your graph is built completely. This is called “**define-by-run**” methodology.

Studies have successfully shown that PyTorch outperforms all other major deep learning libraries in training a Long Short-Term Memory (LSTM) network by having the lowest median time per epoch.The APIs for data loading is well designed in PyTorch. The interfaces are specified in a dataset, a sampler, and a data loader.On comparing the tools for data loading in TensorFlow (readers, queues, etc.), PyTorch’s data loading modules pretty easy to use. Also, PyTorch is seamless while building a neural network, so there is no need to rely on third-party high-level libraries like keras.[8]

Preferable conda installation for Pytorch - <https://pytorch.org/>

1. **Python Library**
   1. **Torchvision:** The torchvision package consists of popular datasets, model architectures, and common image transformations for computer vision which is required by the system for making transformations in the image.
   2. **Matplotlib:** Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits.
   3. **PIL**: The Python Imaging Library (PIL) adds image processing capabilities to the Python interpreter. This library supports many file formats and provides powerful image processing and graphics capabilities.
   4. **OpenCV**- OpenCV is an extensive vision library implemented in C++ and Python, with other language wrappers. The library provides implementations for many different computer vision algorithms, creating an API that is rather simple to use and that can yield powerful results with little code.

**3.5 Feasibility Study**

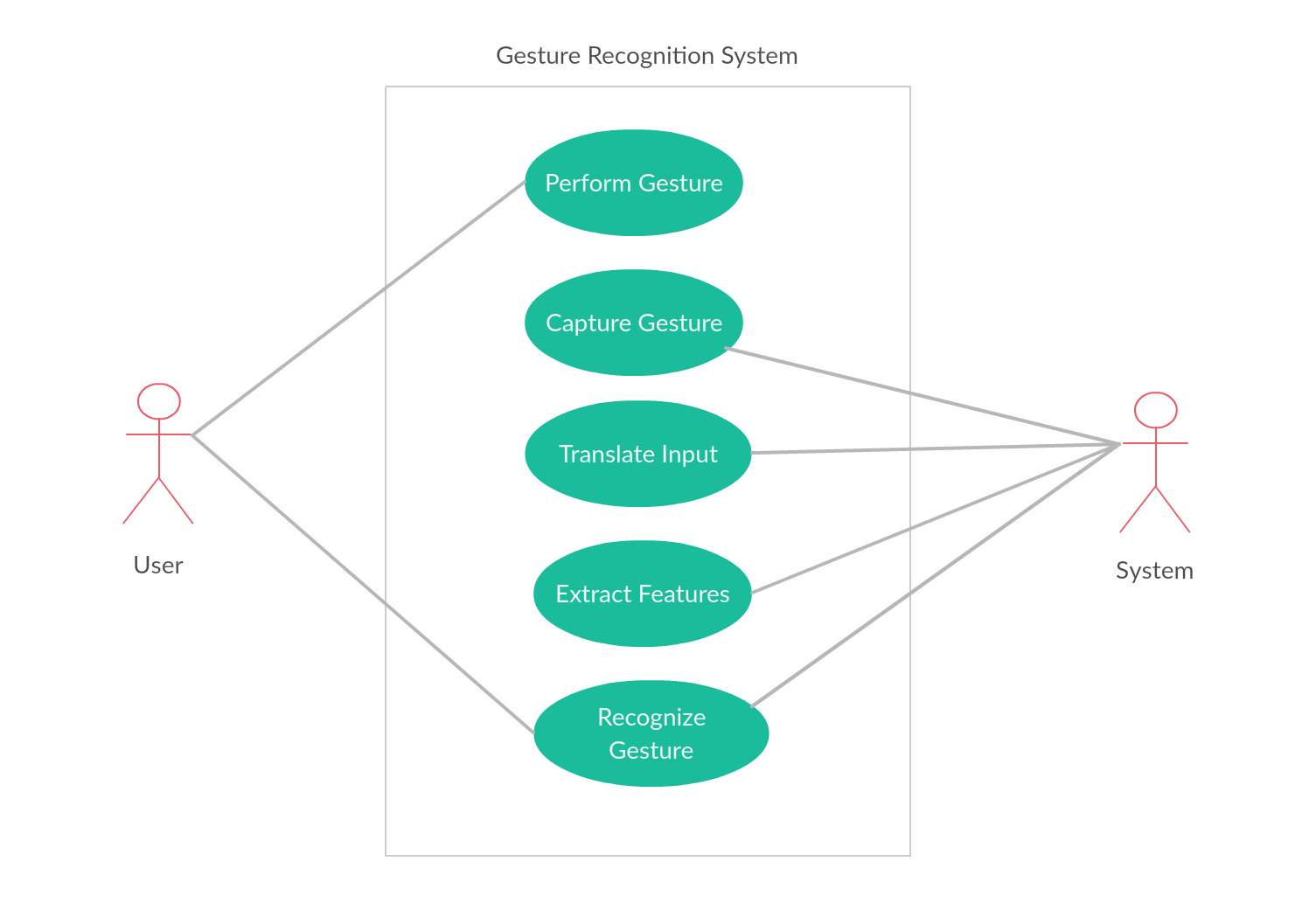
**3.5.1 Technical Feasibility**

1. Computational Feasibility:
   1. The time required for framing a video=4.633311748504639 seconds
   2. Total hours to process video: 77 hours which is a one-time investment.
   3. LSTM Recurrent Neural Network with n different branches
   4. The data format we have used speeds up the HDD access up to 9 to 16 times.
   5. The total amount of data we have 100\*121\*12\*100\*100\*3=4356000000 pixels
   6. The training of one epoch takes around 1 Hour on Nvidia GTX 1080
2. Result Feasibility:
   1. As the similar type of system has been developed by Nvidia on 1500 videos according to the research paper published in 2016.
   2. This System is an extension of the research paper and will have a greater accuracy than Nvidia system which was 75%.

**3.5.2 Economical Feasibility**

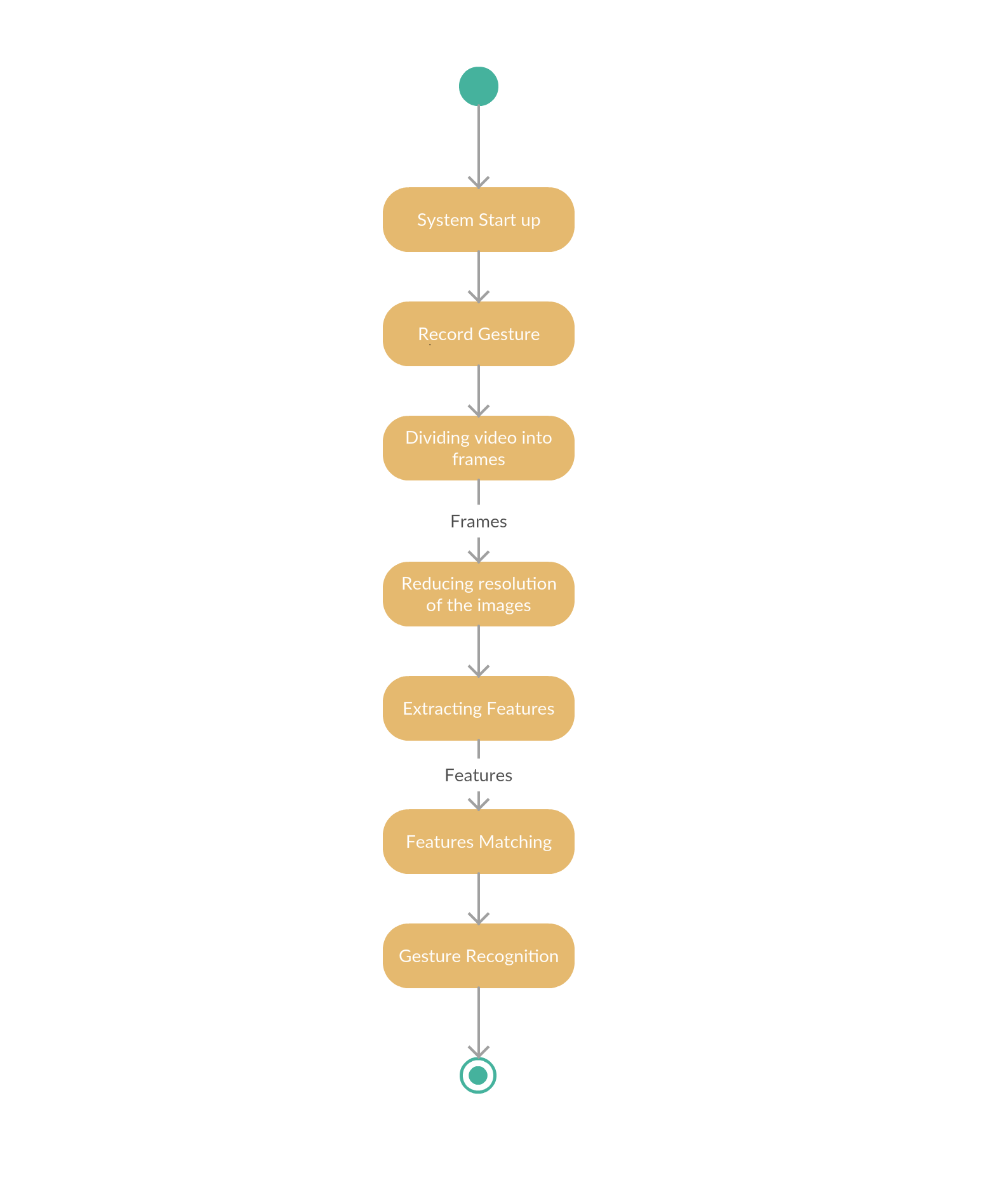
1. The cost factors for RGB Camera and Personal Computers are well known. The proposed system will reduce the cost for additional hardware support required for recognizing the gesture(such as Microsoft Kinect).
2. A single module reduces the installation cost as there is no need to individually support and maintain multiple modules to meet the requirements.

**3.6 Use Case Diagram**

****

**Figure 3.2** Use-Case Diagram

**3.7 System Flow Diagram**

****

**Figure 3.3** System Flow Diagram

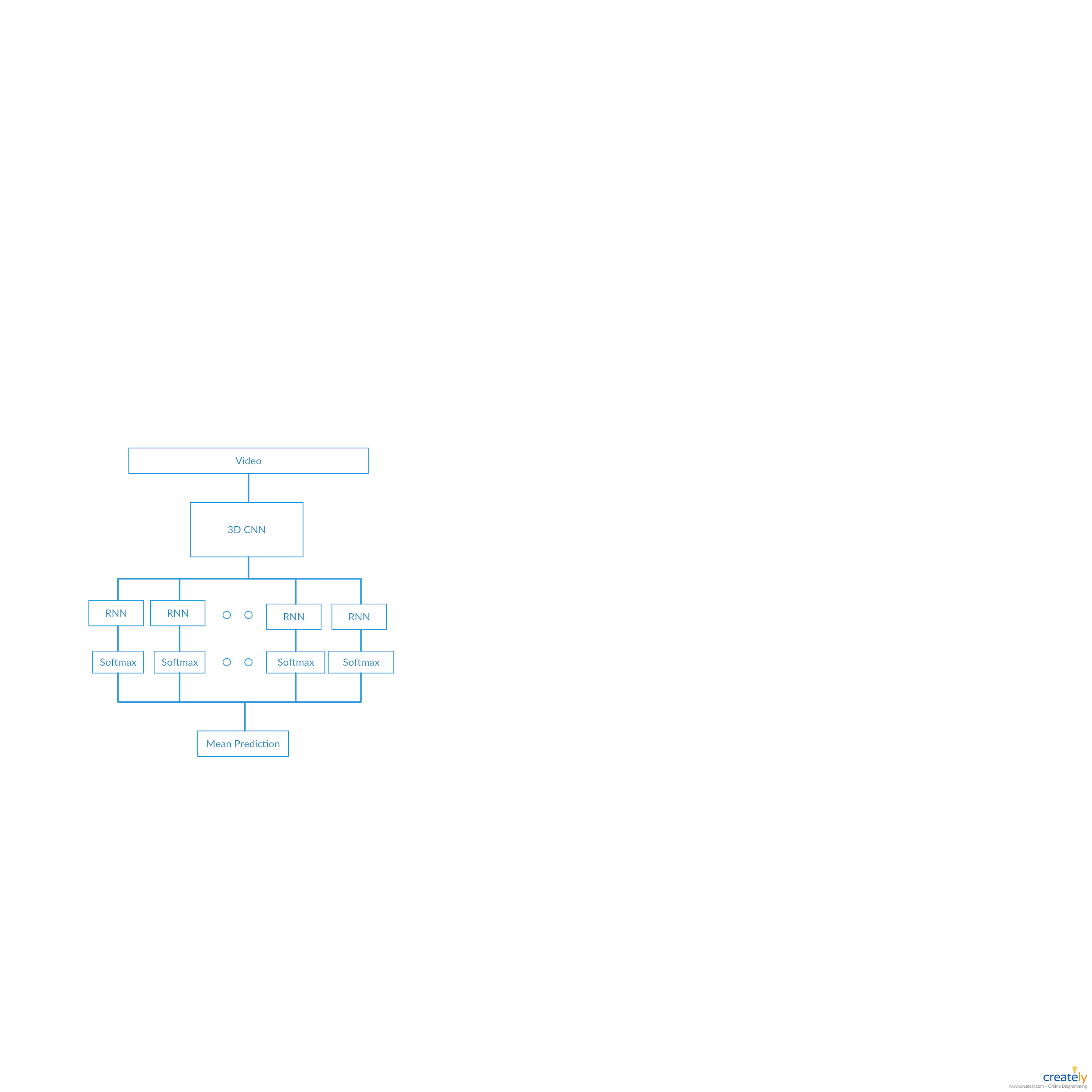
**Chapter 4**

**Design**

So far the discussion was about the current working models in the field of gesture recognition and the technological advancements in the field. Based on the analysis of current models, the proposed system intend to overcome their limitations. In this chapter, the system components and their working is explained.

**4.1 System Architecture**

The proposed system intend to use the complete video, then converting the video into frames. These frames are given as input into the 3D CNN which extracts features from different frames. These extracted features are then fed into N different RNN which are used to learn the time sequence of the gestures.

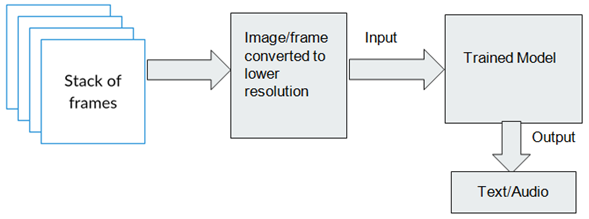
****

**Figure 4.1** Architecture of Proposed approach

The system is composed of five different major part :

1. Video Preprocessing
2. Processing frames in the 3D CNN
3. Processing the Frames in RNN
4. Using Softmax
5. Mean Prediction for the best result

**4.2 Video Preprocessing**



**Figure 4.2:** System Model

Major challenge involved in the above approach is breaking the clips into frames and changing the resolution of frames to the desired resolution so that they can be fed to the model directly without any further preprocessing. This is important step because otherwise, the processing and training will require an enormous amount of memory and processing power.

**4.3 Processing frames in the 3D CNN**

In this part, the network will perform a series of convolution operations on the output frames of the above part and then input features are detected. Convolution is executed by sliding the filter over the input. At every location, a matrix multiplication is performed and sums the result onto the feature map.

Finally the fully connected layers will serve as a classifier on top of these extracted features. They will assign a probability for the object on the image being what the algorithm predicts it is.

**4.4 Processing Frames in RNN**

The frames from the 3D CNN will detect features but the sequence of these frames in which the action was performed is also very important. To detect the sequence it is important to pass these frames through an RNN network which has the capability to find time sequence in videos. Before passing the frames through RNN, that were processed by CNN. The frames are divided into N different blocks. This N block goes into N different RNN network. This is done in order to get a better prediction.

The RNN we will be using is an LSTM (Long Short-Term Memory Network). As the video is composed of about 40 frames it is important to remember the output of the 1st frame when the processing of the 40th frame is going on and this will also help us in better predicting the result.

In order to do so, LSMT has a function that will estimate the usefulness of the result of any frames that have been processed by it and will keep feeding the result to the next RNN block.

The mathematical explanation of the LSTM is given under Background Study (Chapter 2).

**4.5 Using Softmax**

The output of the previous section is provided as input to the softmax function. Softmax is a type of activation function used in classification. We are using softmax as it can classific more than one class at a time and provide the probability for each of the gestures that are being recognised by the system .So in this system there are 121 different gestures that are being predicted.so Softmax activation function will generate a unit vector or in order words a 1x121 matrix than will contain different probability for the action that are present in the classification.

**4.6 Mean Prediction**

The outputs produced by the softmax function are then averaged to get the overall result. This is done by considering the probabilities assigned by the softmax function to each of the gestures detected and then predicting the gesture which has the largest probability.

**Conclusion**

**Work done in Phase-1**

In the first phase, related literature work has been studied and the system requirements have been gathered. The existing systems have been studied and their approaches are analysed. Accordingly, the report incorporates the proposed modifications and the design of the system.

**Work needed to be done in Phase-2**

In the secondphase of project, the following things are planned :

* Data cleaning : Dividing videos into frames, reducing the resolutions of frames.
* Implementation : Developing the architecture, manipulating different types of hyperparameters in order to get a better model.
* Training the model on the training dataset.
* Designing test cases.
* Testing and Evaluation.
* Deploying the software.

**References**

[1] Christopher Lee and Yangsheng Xu, “Online, interactive learning of gestures for human robot interfaces” Carnegie Mellon University, The Robotics Institute, Pittsburgh, Pennsylvania, USA, 1996 .

[2] Richard Watson, “Gesture recognition techniques”, Technical report, Trinity College, Department of Computer Science, Dublin, July, Technical Report No. TCD-CS-93-11, 1993.

[3] Thomas G. Zimmerman , Jaron Lanier , Chuck Blanchard , Steve Bryson , Young Harvill, “A hand gesture interface device”, SIGCHI/GI Proceedings, conference on Human factors in computing systems and graphics interface, p.189-192, April 05- 09, , Toronto, Ontario, Canada, 1987 .

[4] Hyeon-Kyu Lee and Jin H. Kim,” An HMM-Based Threshold Model Approach for Gesture Recognition” IEEE transactions on pattern analysis and machine intelligence, vol. 21, no. 10, october 1999 .

[5] Rick Kjeldsen and John Kender,“Finding skin in colour images”, In Proc. IEEE Int. Conf. on autom. Face and Gesture Recognition, pages 3 12-3 17, 1996 .

[6] Etsuko Ueda, Yoshio Matsumoto, Masakazu Imai, Tsukasa Ogasawara. “Hand Pose Estimation for Vision based Human Interface”, IEEE Transactions on Industrial Electronics, Vol. 50, No. 4, pp. 676–684,2003.

[7] Chan Wah Ng, Surendra Ranganath, “Real-time gesture recognition system and application”, Image Vision Comput, 20(13-14): 993-1007 ,2002.

[8] Claudia Nölker and Helge Ritter, “Visual Recognition of Continuous Hand Postures”, IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 31, no. 1, February 2001.

[9] Zhang Yuye & Yan Weisheng;“Research of Traffic Signal Light Intelligent Control System Based On Microcontroller”, First International Workshop on Education Technology and Computer Science,pp301- 303. 2009.

[10] Manoj Kanta Mainali & Shingo Mabu ;“Evolutionary Approach for the Traffic Volume Estimation of Road Sections”, pp100- 105, IEEE. 2010.

[11] Ms promila sinhmar; “ Intelligent Traffic Light and Density Control using Ir Sensors and Microcontroller”; International Journal of Advanced Technology & Engineering Research; ISSN NO: 2250-3536; VOLUME 2, ISSUE 2, MARCH 2012.

[12] X. Y. Wang, G. Z. Dai, X. W. Zhang, Recognition of Complex Dynamic Gesture Based on HMM-FNN Model, Journal of Software, 19 (2008), 2302-2312. https://doi.org/10.3724/sp.j.1001.2008.02302

[13] X. H. Wang, H. J. Bao, Gesture Recognition Based on Adaptive Genetic Algorithm, Journal of Computer-aided Design & Computer Graphics, 19 (2007), 1056-1062.

[14] F. F. Zeng, Y. Wang, G. J. Huang, Fuzzy Neural Network Based Hand Gesture Recognition, Mini-Micro Computer Systems, 21 (2000), 706-709.

[15] H. J. Lee, J. H. Chung, Hand Gesture Recognition Using Orientation Histogram, Proceedings of the IEEE Region 10 Conference, 2 (1999), 1355-1358. https://doi.org/10.1109/tencon.1999.818681

[16] L. G. Zhang, J. Q. Wu, Hand Gesture Recognition Based on Hausdorff Distance, Journal of Image and Graphics: A, 7 (2002), 1056-1062.

[17] H. B. Ren, G. Y. Xu, Hand Gesture Recognition Based on Characteristic Curves, Journal of Software, 13 (2002), 987-993.

[18] Y. Cui, D.L. Swets and J.J. Weng, Learning-based Hand Sign Recognition Using SHOSLIF-M, Proceeding of 5th International Conference on Computer Vision, (1995), 631-636. https://doi.org/10.1109/iccv.1995.466879

[19] F. Quek, T. Mysliwiec, M. Zhao, Finger Mouse: a Free-hand Pointing Interface, IEEE International Conference on Automatic Face and Gesture Recognition-FGR, (1995), 372-377.

[20] J. Triesch, C. V. D. Malsburg, A System for Person-independent Hand Posture Recognition against Complex Backgrounds, IEEE Transaction on Pattern Analysis and Machine Intelligence, 23 (2001), 1449-1453. https://doi.org/10.1109/34.977568

[21] B. Stenger, Template Based Hand Pose Recognition Using Multiple Cues, Lecture Notes in Computer Science, Vol. 3852, Springer-Verlag Berlin Heidelberg, 2006, 551-560. https://doi.org/10.1007/11612704\_55

[22] C. Keskin, F. Kirac and Y. E. Kara, Lale Akarun, Real Time Hand Pose Estimation Using Depth Sensors, In Proceedings of the IEEE International Conference on Computer Vision Workshops, (2011), 1228–1234. <https://doi.org/10.1109/iccvw.2011.6130391>

[23] S. Mitra, and T. Acharya. (2007). “Gesture Recognition: A Survey” IEEE Transactions on systems,Man and Cybernetics, Part C: Applications and reviews, vol. 37 (3), pp. 311- 324, doi:10.1109/TSMCC.2007.893280.

[24] Simei G. Wysoski, Marcus V. Lamar, Susumu Kuroyanagi, Akira Iwata, (2002). “A Rotation Invariant Approach On Static-Gesture Recognition Using Boundary Histograms And Neura

[25] Joseph J. LaViola Jr., (1999). “A Survey of Hand Posture and Gesture Recognition Techniques and Technology”, Master Thesis, Science and Technology Center for Computer Graphics and Scientific Visualization, USA

[26] Malima, A., Özgür, E., Çetin, M. (2006). “A Fast Algorithm for Vision-Based Hand Gesture Recognition For Robot Control”, IEEE 14th conference on Signal Processing and Communications.

[27] Guan, Y., Zheng, .M. (2008). “Real-time 3D pointing gesture recognition for natural HCI. IEEE Proceedings of the 7th World Congress on Intelligent Control and Automation WCICA 2008, doi:10.1109/WCICA.2008.4593304

[28] Freeman, W. T., Weissman, C. D. (1995). ” Television Control by Hand Gestures”. IEEE International Workshop on Automatic Face and Gesture Recognition.

[29] Kouichi M., Hitomi T. (1999) “Gesture Recognition using Recurrent Neural Networks” ACM conference on Human factors in computing systems: Reaching through technology (CHI '91), pp.237-242. doi: 10.1145/108844.108900

[30] Manar Maraqa, Raed Abu-Zaiter. (2008). “Recognition of Arabic Sign Language (ArSL) Using Recurrent Neural Networks,” IEEE First International Conference on the Applications of Digital Information and Web Technologies, (ICADIWT), pp. 478-48. doi:10.1109/ICADIWT.2008.4664396

[31] Min B., Yoon, H., Soh, J., Yangc, Y., & Ejima, T. (1997). “Hand Gesture Recognition Using Hidden Markov Models”. IEEE International Conference on computational cybernetics and simulation. Vol. 5, Doi: 10.1109/ICSMC.1997.637364

[32] Mahmoud E., Ayoub A., J¨org A., and Bernd M., (2008). “Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition”, World Academy of Science, Engineering, and Technology 41

[33] Xingyan Li. (2003). “Gesture Recognition Based on Fuzzy C-Means Clustering Algorithm”, Department of Computer Science. The University of Tennessee Knoxville.

[35]Stan Sclaroff, Carol Neidle, Vassilis Athitsos, ”The American Sign Language Lexicon Video Dataset”.(2008) IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops

[36][Daphne Cornelisse](https://medium.freecodecamp.org/@daphn3cor?source=post_header_lockup), “An intuitive guide to Convolutional Neural Networks”

Neuroscience Student at Erasmus University College, Instructor Bletchley AI Bootcamp

[37] [Sambit Mahapatra](https://towardsdatascience.com/@sambit9238?source=post_header_lockup), ”Medium post Why Deep Learning over Traditional Machine Learning?”, <https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063>

## [38]Christopher Olah, “Understanding LSTM Networks” <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

# [39][Monik Pamecha](https://becominghuman.ai/@monikkinom?source=post_header_lockup), ”A noob’s guide to implementing RNN-LSTM using Tensorflow”

# <https://becominghuman.ai/a-noobs-guide-to-implementing-rnn-lstm-using-tensorflow-1907a5bbb1fa>

[40]Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz, ”Hand Gesture Recognition with 3D Convolutional Neural Networks”, NVIDIA, Santa Clara, California, USA, 2015

[41] E. Costello. Random House American Sign Language Dictionary. Random House, New York, 1994.

[42] M. L. A. Sternberg. American Sign Language Dictionary Unabridged. Collins, 1998.[20] R. A. Tennant and M. G. Brown. The American sign language Handshape Dictionary. Gallaudet U. Press, Washington, DC, Washington, DC

[43] R. A. Tennant and M. G. Brown. The American Sign Language Handshape Dictionary. Gallaudet U. Press, Washington, DC, Washington, DC.

[44] C. Valli, editor. The Gallaudet Dictionary of American Sign Language. Gallaudet U. Press, Washington, DC,2006.

[45] Vexcel Imaging GmbH, a Microsoft company Anzengrubergasse 8, “Introduction to UltraCamx”, <https://www.kasurveys.com/documents/ULTRACAM-Specs-UCX.pdf>

[46]Ruth Chabay, Bruce Sherwood, John Coady, Danny Caballero, Matt Craig, Steve Spicklemire, Aaron Titus(2016), “The Evolution of VPython”, <http://vpython.org/contents/announcements/evolution.html>

[47]Faisan Shaikh, “An Introduction to PyTorch – A Simple yet Powerful Deep Learning Library”, <https://www.analyticsvidhya.com/blog/2018/02/pytorch-tutorial/>