3. PROPOSED WORK

This chapter discusses the work carried out for lip reading in Gujarati language. First, it discusses the existing frame selection algorithm from the videos. Then, it gives a detailed description of the work that has been carried out in the form of a large dataset of Gujarati language alphabets and proposed ViLiDEx algorithm for Lip Extraction from the video frames.

3.1 EXISTING METHOD OF FRAME SELECTION

In a field of research, Automatic Lip Reading (ALR) can be defined as *the process of recognizing the utterances by analysing the lip, tongue and teeth movements of a speaker from a given video recording without any acoustic input [Zhou, 2013]*.

As discussed earlier, the ALR process involves the following steps:

1. Lip Detection and Extraction
2. Feature Extraction
3. Feature Transformation
4. Classification.

The first step, Lip Detection and Extraction includes raw data collection, Data pre-processing and Dataset Creation. After collecting raw data, the most important task is data pre-processing which includes audio and video signal separation, key frame selection and ROI extraction.

3.1.1 Data Preprocessing Method for Video Frames

In ALR, for alphabet recognition, videos are captured at 25 fps or 50 fps. The length of the video for each alphabet may differ, as the length pronunciation of each alphabet and utterance time for different speakers may vary. Researchers have tried to remove redundant information from all the original frames to extract key frames and cropped lip area. The following method is performed for extracting lip area:

1. The entire video is divided into 10 equal parts, and a random frame from each part is selected as a key frame.
2. OpenCV library has been used to load images and convert them into a three-dimensional matrix [18]. Then, the facial landmark detection is carried out using

Dlib toolkit [19]. It takes the face images as input, and the returns face structure consists of different landmarks for each specific face attribute. For lip part, seven key points of the mouth have been identified with numbers 49, 51, 53, 55, 57, 58, 59.

3. The mouth images are segmented, and the redundant information are removed. The centre position of the mouth is calculated based on the coordinate points of the image boundary, denoted as (X_0, Y_0). The width and height of the lip image are represented by W and H, respectively, L_1 and L_2 represent the left and right, upper and lower dividing lines surrounding the mouth, respectively. The following formula has been used to calculate the bounding box of the mouth:

$$L_1 = X_0 \pm W_2 \ldots \ldots \ldots (1)$$

$$L_2 = Y_0 \pm H_2 \ldots \ldots \ldots (2)$$

3.1.2 Limitations of the method

In this work, the key frames are randomly selected from each of the partitions. This strategy is good for digit recognition, as the digits are recognized using words from the starting position of mouth. The following figure shows lip movements for digits 0-4. Digits are recognized from the starting position of mouth. Figure 4 shows lip movements for digits 0, 1, 2, and 4. Here it is observed that initial 4-5 frames are sufficient to recognize a particular digit.

Figure 1. Lip sequence for English Digits

In the case of alphabet recognition, it is difficult to recognize an alphabet by initial frame sequences. The following figure shows lip sequence of guttural class alphabets. The alphabets of the same class have similar articulation effect and require a greater number of frames for recognition. Hence, twenty frames that have been considered as utterance style may differ for different speakers, total number of frames per one alphabet may differ for each speaker.

Following figure 2 shows the initial lip sequence of guttural class alphabets. It is difficult to recognize an alphabet from the initial lip sequence. If frame selection is random, it might miss some key frames which could be useful for alphabet classification. Hence, to avoid this randomness, ViLiDEx, a modulo based frame number selection for frame removal, algorithm is proposed.

Figure 2. Initial Lip sequence of Gujarati alphabet (Guttural class: ka, kha, ga, gha)

3.2 THE PROPOSED WORK – GJVARNA DATASET AND VILIDEX ALGORITHM

3.2.1 GJVarna Dataset Creation

There are a total 36 consonants and 12 vowels in Gujarati language. Figure 2 shows consonants classified into 5 sub classes named Guttural, Palatal, Retroflex, Dental, and Labial.

Figure 3. Gujarati Alphabets and classification (courtesy: https://www.languagesgulper.com/eng/Gujarati.html)

As discussed in literature study chapter, no work has been carried out for Lip-Reading in Gujarati language. Hence, for this research work, the first contribution is to create a huge dataset of Gujarati alphabets. This has been created and named GJVarna. The second contribution is the frame removal algorithm, ViLiDEx. The frame removal algorithm has been tested for five sub classes, Guttural (ક, ખ, ગ, ઘ), Palatal (ચ, છ, જ, ઝ), Retroflex (ટ, ઠ, ડ, ઢ, ણ), Dental (ત, થ, દ, ધ, ન), and Labial (પ, ફ, બ, ભ, મ). The following subsections discuss the research carried out in detail.

GJVarna is a 2D image dataset with depth. It includes 34 consonants of Gujarati language. The following steps are performed to create the GJVarna dataset.

1. **Raw Data Creation**
    a. Video recording using Nikon D 5300 camera with 1920X1080 FHD resolutions and 50 frames/second.
    b. Recording was performed in a room with regular white light. (See the following figure 1)

Figure 4 Room set up for recording

c. Speakers are family members, friends and students who know Gujarati language.
d. Speakers are not trained in advance for Gujarati alphabet, so that they can utter the words the way they usually do. However, it was ensured that their mother tongue is Gujarati.
e. Total 30 speakers took part in recording for dataset creation, including 17 females and 13 males. From these 30 speakers, 5 speakers have been repeated and recorded with different get up and time. Figure 12 shows some of the speakers of GJVarna dataset.

Figure 5 Some of the speakers

f. 3 shots have been recorded for each speaker.

g. One continuous video, speaking 34 consonants, is recorded for 28 speakers. (eg. Ka, kha, ga,gha …ksha, gna).

h. 4 speakers were uttering each alphabet three times. (ક, ક, ક, ખ, ખ, ખ, … જ્ઞ, જ્ઞ, જ્ઞ). This recording was done to check if there is a difference in articulation of alphabet.

i. Among these 30 speakers, 3 speakers were removed due to blurry videos, 2 speakers were removed due to not proper articulation of alphabets.

j. Also, 2 speakers from 5 speakers, who were generating the same lip movement for the alphabet.

k. Finally, the dataset containing 28 speakers, 19 for training and 9 for testing, were taken.

l. The dataset created for these 28 speakers with total frames 15, 20. Figure 5 shows a few data of GJVarna dataset for total frames 20.

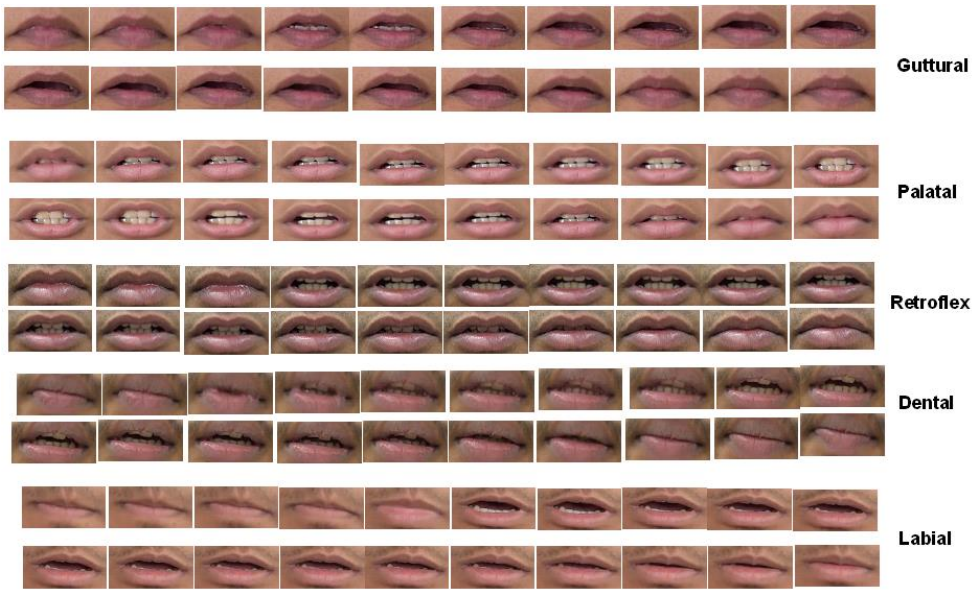

Figure 6 Dataset sample for Guttural, Palatal, Retroflex, Dental, and Labial class for 20 frames

Figure 7 15 frames dataset for 'ક', 'ર', and 'થ' consonant

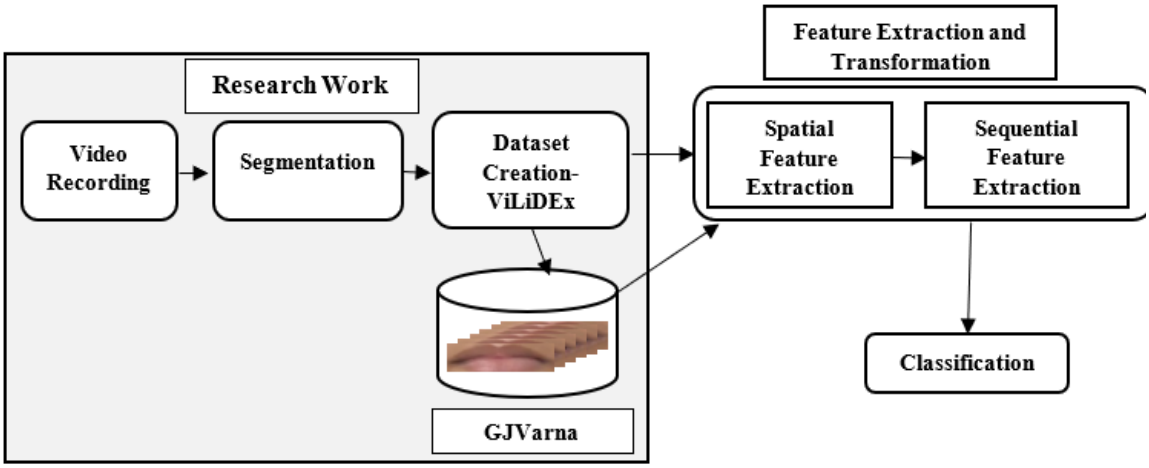

Figure 8 Proposed research work for ALR Process

2. **Data Pre-processing using Data Segmentation**
    a. All continuous video files are segmented into individual clips, each for a single alphabet, using “Movies and Tv” application on Windows 10 operating system.
    b. Alphabets wise each segment is saved as a separate video file.
3. **Data Creation using ViLiDEx**
    a. The *ViLiDEx* algorithm is applied to each video file for alphabet for key frame selection and lip area (ROI) extraction.
    b. 20 frames per alphabet are saved for each speaker in GJVarna dataset.

3.2.2 ViLiDEx Algorithm for Extra Frame Removal

Alphabet utterances for different speakers may vary, so for uniform frame size, ViLiDEx algorithm is designed based on Facial landmark pre-trained model of Dlib. Facial landmarks using Dlib give a total of 68 landmarks of face, among them landmarks from 49-68 which are for lip area are cut down and given as an input for next level (see Figure 8). These points are extracted and used as an input for the next level. In alphabet utterance, key frames are distributed throughout the video, specifically for alphabets of same class. Total number of frames may also vary for different alphabet and speaker. Hence, random selection of frames in existing algorithms may miss some key frames. To overcome this problem, ViLiDEx has been developed.

Figure 9 Face landmark points and ROI Selection

The ViLiDEx algorithm takes, for each alphabet, a video as an input. Then, it counts the

total number of frames, detects lip area and extracts it from each frame and finally saves the new frame. If the total number of frames is more than the limit (15/20), the extra frames will be removed. Frame numbers divided by following numbers (2, 3, 5, 7, 11, 17… up to Total number of frames) will be removed.

This algorithm calculates total frames of input video of alphabet. If total frames are multiple of 20 (20*1, 40(20*2), 60(20*3), 80(20*4) … and so on), then frame number divisible by multiplicand (1, 2, 3, 4…) will be kept and others will be discarded as extra frames. If total frames are not multiple of 20 then Frame difference will be calculated. Prime numbers and total numbers divided by these prime numbers up to total frames are listed. Prime numbers whose count is equal to frame difference will be searched and frame numbers divisible by these prime numbers will be discarded (See Table 1). For the remaining 20 frames, using Face landmark points 49-68, lip area will be extracted and stored. Time complexity of this algorithm is *O(m*n*p)*, where *m*n* is the resolution of image in the frame and *p* is total number of frames. Steps of ViLiDEx algorithm are as follows.

ViLiDEx Algorithm

1. *Read input video.*
2. *Count Total number of Frames.*
3. *Calculate Frame difference = Total Frames- 20*
4. *If frame difference = 0*

    Density = 'E'

    Divisor =1

    Else if Frame difference % 20 = 0

    Density = 'M'

    Divisor = int (Total Frames / 20)

    Else

    Density='S'

    List Prime numbers from 3 to Total Frames

    Count total numbers (1 to Total Frames) divisible by each prime number listed above

    Search for the counts whose total is equal to frame difference
    Corresponding numbers in list of primes are List of Divisors for Extra frames

5. *Set the path to store dataset*
6. *For each captured frame from an input video*
   If Density = 'E'
   Select each frame
   Else if Density = 'M'
   Select the frames whose frame numbers are divisible by Divisor and discard others
   Else if Density = 'L'

   Select the frames whose frame numbers are not divisible by divisor and discard which are divisible

   Else
   Select the frames whose number is divisible by List of Divisors and discard other frames
7. *For each frame selected in step 6*
   i. *Convert the image to grayscale*
   ii. *Using face landmark points, detect the ROI*
   iii. *Crop ROI and count edges using canny edge detector algorithm*
   iv. *Resize the cropped ROI*
   v. *Apply image sharpening on cropped ROI if edges are in specific range*
   vi. *Save the ROI*
8. *Close input video*

Total Frames	**Frame Difference**	**Divisor /List of Primes**	**Density**	**Prime Nos Needed**	**Count total numbers divisible by each prime**
20	0	1	'E' for Equal	-	-
40	20	40/2=2	'M' for Multiple of 20	-	-
30	10	[3]	'L' for in List of Primes	[**3**, 5, 7, 11, 13, 17, 19, 23, 29]	[**10**, 6, 4, 2, 2, 1, 1, 1, 1]
39	19	[3, 5, 23]	'S' for search in List of Primes	[**3**, **5**, 7, 11, 13**,** 17, 19, **23**, 29, 31, 37]	[**13**, **7**, 5, 3, 3, 2, 2, **1**, 1, 1, 1] 13 + (7 – 2)+1(remove frame no 15 and 30 common for 3 and 5) = 19
38	18	[3, 5, 23]	'S' for search in List of Primes	[**3**, **5,** 7, 11, 13, 17, 19, **23**, 29, 31, 37]	[**12**, **7,** 5, 3, 3, 2, 2, **1**, 1, 1, 1] 12+(7-2)+1

Table 1. Working of ViLiDEx algorithm

3.3 STEPS OF ALPHABET RECOGNITION USING MOBILENET

For alphabet classification, CNN-LSTM model has been used. CNN identifies lip shape and then each of the output of CNN becomes transform sequence. LSTM learns the pattern of the sequence that contains the change of lip shape. Here, trained CNN model MobileNet has been used for training and testing.

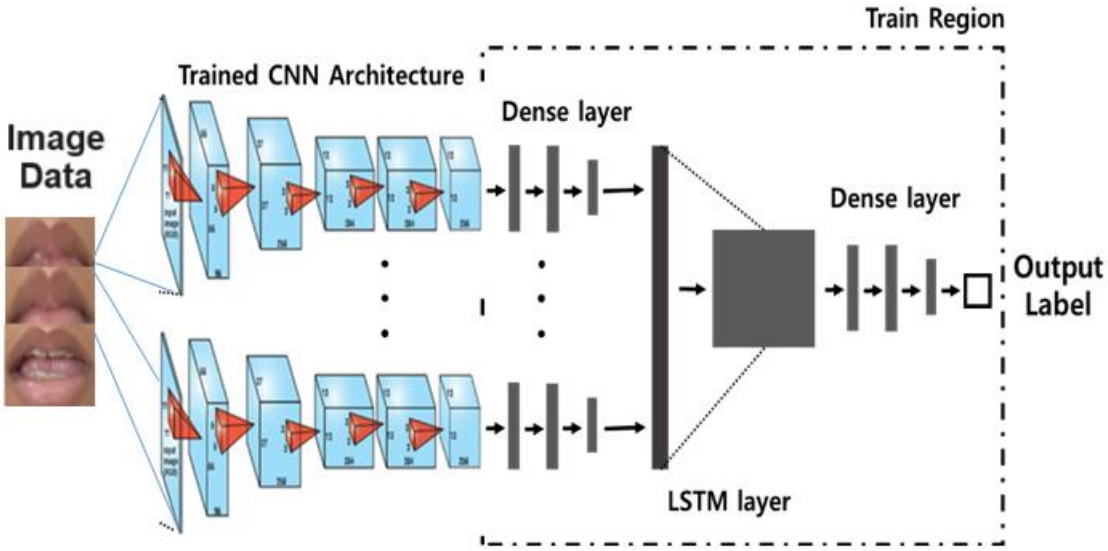

Figure 10 CNN LSTM Model for Classification

CNN-LSTM model shown in figure 17 is the simple architecture for classification of Gujarati alphabets. After transforming the lip videos into images as an input to pre-trained CNN architecture. Output of CNN passes through dense layer where images are transformed and sent to LSTM layer. Output of LSTM layer is again passed through another dense layer and finally output labels are received by SoftMax activation function. Dotted line region is train region.

MobileNet is a compact model, and users can resize within regular range, as a result MobileNet is selected. Label wise counted data is piled in timestamps of 20 and 15 frames. This image list is transformed into Numpy array for Keras input. Then data is shuffled, and batch is made for training.

In this research, two datasets, one with 15 frames and other with 20 frames are used for recognition. Two different learning rates (0.0001 and 0.0002), two values of validation ratio (0.2 and 0.25) for different epoch values (5, 10, 15, 20, 25, 30, 35, and 40) are used. Gujarati alphabets are classified in 5 classes (1 for Guttural, 2 for Palatal, 3 for Retroflex,

4 for Dental, and 5 for Labial). From 34 consonants, 15 consonants are used, 3 per class. Steps for training and testing using Mobile Net are given below. Results and observations are discussed in the next chapter.

3.3.1 Training of GJVarna Dataset Using Mobile Net:

1. Set Parameters like timestamp, labels, learning rate, batch size, epochs etc.
2. Create an empty dataset and load training dataset
3. Build a model
4. Compile the model
5. Train the model
6. Save the model

3.3.2 Testing of GJVarna Dataset Using Mobile Net:

1. Load the saved model
2. Load testing dataset
3. Test dataset
4. Print result

SUMMARY

This chapter explains the existing method of frame selection and its limitations. ViLiDEx algorithm and its working for frame removal. The raw data collection and dataset creation, speakers, dataset samples, and CNN-LSTM model for recognition has also been discussed.