International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 6, Issue 3, May- June 2017 ISSN 2278-6856

# Audio and Video Speech Processing

Reshma S Hebbar[1], K. Suchetha Shenoy[2], Deepika Anchan[3], Deepa S. Kini[4] , Imraz[5]

[1,2,3,4,5] Dept. of Computer Science and Engineering, Shri Madhwa Vadiraja Institute of Technology and Management, Vishwothama Nagar, Bantakal, Udupi, Karnataka 574115, India

## Abstract

***Audio and video speech processing has been a trending research area in the recent years. Conversion of audio cues and video cues into text is really helpful for people having speech and hearing impairments. In this paper, we present the method of converting audio and video frames into the text which represents the corresponding spoken word. We have used a k-NN classifier to classify different speakers' utterances, MFCC algorithm for audio classification as well as content-based retrieval and Support Vector Machine, a supervised learning method which generates input-output mapping functions from a set of labeled training data.***

**Keywords:** AVSP, MFCC, SVM, k-NN classifier.

## 1. INTRODUCTION

Speech is an important means for conveying the information among human beings. This project AVSP (Audio and Video Speech Processing) works as 2 modules. This project allows the user to decide whether to convert his/her video or an audio file into text. Initially 12 words such as hi, hello, sorry, thank you, welcome, good morning, good afternoon, good evening, good night, where are you, what are you doing, what is your name are recorded individually in audio as well as video formats and stored in a single folder. Based on the user's choice, the resulting text will be displayed in the GUI.

The project AVSP has been developed solely on the MATLAB R2013a. The main reason is the availability of the image processing toolbox, which provides some of the important features such as segmentation, feature extraction, image transformation functions and so on.

Feature extraction is the backbone process of project AVSP. We have used k-NN (k-Nearest Neighbor) to analyze the data for classification analysis. It is mainly used for classification of image frames in visual speech recognition. MFCCs are used for audio feature extraction in the project AVSP.

## 2. APPROACH

In the first module, our application can detect visual speech recognition, using MATLAB as the platform for face feature detection, mouth segmentation, pre-processing, thresholding and recognition. Extracted visual cues from the video are matched with the trained data using the k-NN classification algorithm to give the output as textual data which can be read by the normal or people and understand what the speech impaired person is trying to convey.

In the second module, human voice in the .wav format is converted into digital signal form to produce digital data representing each level of signal at every discrete time step. The digitalized speech samples are then processed using MFCC (Mel Frequency Cepstral Co-efficient) to produce voice features. After that, the coefficient of voice features can go through SVM (Support Vector Machine) classification to select the pattern that matches the database and input frame in order to minimize the resulting error between them and provide the result. The MFCC technique is implemented using MATLAB. At different levels, different operations are performed on the input signal such as pre-emphasis, framing, windowing, mel-cepstrum analysis and recognition (matching) of the spoken word. Followed by classification using the SVM classifier which is used to test the loaded audio and then generate the textual data as the result. Thus, a hearing impaired person can understand what a normal person is trying to convey.

## 3. METHODOLOGY

The overall procedure block diagram is shown in the figure 1, where user is given two options to choose from, i.e. whether he wants to convert visual cues to text or audio cues to test based on his/her requirement.

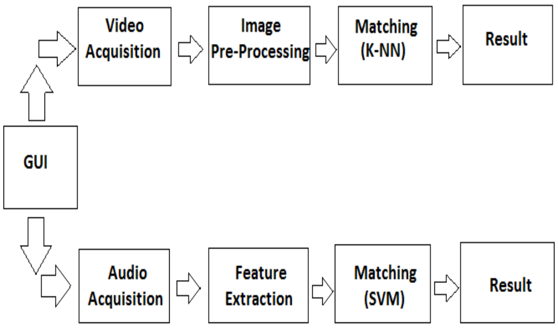


**Figure 1** Overall Process Diagram

### 3.1 Visual Cue processing

**Image Acquisition:**

Before any video or image processing can commence an image must be captured by a camera and converted into a manageable entity. This is process is known as image acquisition. The image acquisition process consists of 3 steps: energy reflected from the object of interest, an optical system which focuses on the energy and finally a sensor which measures the amount of energy.

First of all, a camera or a webcam is used to capture a video of a person whose face is right in front of a camera or computer. In the video, the person is supposed to utter syllable as slow as possible but continuously without any sound. A few samples of a video should be captured to get the best one whereby the utterance of the person without any sound can be seen clearly through the mouth movements with the best brightness of the background. Face detection is a computer technology that determines the locations and sizes of human faces in digital image. It only detects facial features of humans and ignores other things that may be either living things or non-living things, such as animals, buildings, trees and vehicles. After the face detection is implemented on the selected frames, fundamental but crucial face features such as the eyes, nose, mouth and the next few elements are needed to be focused on. Among the facial features which are detected, the mouth is the Region of Interest (ROI) [1].

**Image Pre-processing**:
Image pre-processing is manipulation of data which are in the form of an image through several techniques. The 3 basic operations that can be performed on it are image enhancement, restoration and compression. Enhancement of an image occurs when an image is modified so that the information it contains is clearer. Another basic operation is image restoration which also aims at improving an image. The knowledge of how the image was formed is to retrieve the ideal image. Compression is a way of representation of an image by fewer numbers and minimization of the degradation of the information contained in the image simultaneously [1].

**Image Thresholding:**
Threshold selection is one of the most important processes in image processing. It is a widely used technique for image segmentation. It is based on the assumption that object and the background can be distinguished. The pre-processing also defines a compact representation of the pattern. Binarization process converts a gray scale image into a binary image. The first step in processing is to binarize the numeral images so that the character image is converted into binary image (black and white) i.e. having pixel values 0 and 1. Generally the scanned image is in its true color (RGB image) and this has to be converted into a binary image, based on threshold value.

**Matching:**
This is the final step of visual cue processing the project AVSP. We have used the k-NN algorithm. The k-nearest neighbor is a pattern recognition algorithm used for classification. In this, the input consists of k closest training examples in the feature space. It is used to match the existing trained data to the data being tested to provide the textual data. Classification algorithm consists of 2 phases:

- **Training**: In this phase, the training examples are vectors in a multidimensional feature space, each with a class label and the corresponding values stored in the (255*255) matrix. Feature vectors and class labels of training are given as input to form a baseline for testing where the trained data is stored in a .mat format. Samples have to be fed in again and again to achieve the accuracy which helps in distinguishing between the accurate and noise filled data.
- **Testing**: In this phase, the loaded data are compared with the trained data, based on the minimum distance; weights are associated with each frame. Thus, the frames which have 90% closeness with the trained data are considered accurate and the respective class label will be the output text displayed in the text box.

### 3.2 Audio Cue Processing

**Audio Acquisition:**
For any processing of audio signal to take place, first and foremost we need to record the audio using a microphone. In the PC, the sample audios are recorded and are stored in a .mat file. The first step in speech processing is to convert the analog representation into a digital signal. This process of analog to digital conversion has 2 steps: Sampling and quantization. A signal is sampled by measuring its amplitude and sampling rate at a particular time. The sampling rate is the number of samples taken per second. The amplitude measurements are usually stored either as 8-bit (values from -128 to 127) or 16-bit (values from -32768 to 32767) integers. This process of representing real valued numbers as integers is called quantization. Before applying feature extraction, the signal must be free of noise and therefore the spectral subtraction method is applied to get rid of unwanted noise. In this method, an average signal spectrum and an average noise spectrum are estimated in the parts of the recording and are subtracted from each other so that the average SNR (Signal-to-Noise Ratio) is improved.

**Audio Feature Extraction and processing:**
MFCC is based on human hearing perceptions which cannot perceive frequencies over 1KHz. In other words, in MFCC is based on known variation of the human ear's critical bandwidth with frequency. There are seven basic steps to be followed to obtain the co-efficient:

- **Pre-emphasis**: This step processes the passing of signal through a filter which emphasizes higher frequencies. This process will increase the energy of signal at higher frequency.

$$Y[n] = X[n] - 0.95X[n-1] \qquad (1)$$

  Let us consider a=0.95, which makes 95% of any one sample is presumed to originate from previous sample.
- **Framing**: The process of segmenting the speech samples obtained from analog to digital conversion (ADC) into a small frame with the length within the

range of 20 to 40 msec. The voice signal is divided into frames of N samples, with next frames separated by M samples (M<N). Normally, M=100, N=256.

- **Hamming Windowing**: It is used as window shape by considering the next block in feature extraction processing chain and integrates all the closest frequency lines.

N=number of samples in each frame
Y[n] = Output signal
X[n] = Input signal,
W[n] = Hamming window
The result of windowing signal is shown below:

$$Y[n] = X[n] \times W[n] \qquad (2)$$

- **Fast Fourier Transform**: To convert each frame of N samples from time domain into frequency domain. Each frame having $N_m$ samples are converted into frequency domain. Fourier transformation is a fast algorithm to apply Discrete Fourier Transform (DFT), on the given set of $N_m$ samples shown below:

$$D_k = \Sigma \ D_m e^{-j2\pi km/N}{}_m \qquad (3)$$

Where k = 0, 1, 2 … $N_{m-1}$

- **Mel Filter Bank Processing**: The frequencies range in FFT spectrum is very wide and voice signal does not follow the linear scale. It gives a set of triangular filters that are used to compute a weighted sum of filter spectral components so that the output of process approximates to a Mel scale [3].

$$fmel = 2595 \times Log10[1+(\frac{flin}{700})] \qquad (4)$$

- **Discrete Cosine Transform**: This is the process to convert the log Mel spectrum into time domain using Discrete Cosine Transform (DCT). The result of the conversion is called Mel-scale Frequency Cepstrum Coefficient. The set of coefficient is called acoustic vectors. Therefore, each input utterance is transformed into a sequence of acoustic vector.

$$C_n = \frac{2}{N`(\sum_{i=1}^{N_f} Xkcos(ki \times 2\Pi/N` \times n)} \qquad (5)$$

Where $1 \leq n \leq p$

- **Delta Energy and Delta Spectrum**: The voice signal and the frames changes, such as the slope of a formant at its transitions. Therefore, there is a need to add features related to the change in cepstral features over time. 13 delta or velocity features (12 cepstral features plus energy), and 39 features a double delta or acceleration feature are added. The energy in a frame for a signal x in a window from the time sample t1 to time sample t2, is represented at the equation below:

$$Energy = \sum X^2[t] \qquad (6)$$

Each of the 13 delta features represents the change between frames corresponding cepstral or energy feature, while each of the 39 double delta features represents the change between frames in the corresponding delta features.

$$d(t) = \frac{(c(t+1) - c(t-1))}{2} \qquad (7)$$

**Matching:**

This is the final step of audio cue processing. We have used the SVM classifier to achieve the matching of the audio frames used for training with that of the audio frames created by the loaded audio. SVM classification uses different planes in space to divide data points using planes. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories or classes are divided by a hyper plane that maximizes the margin between different classes. This is due to the fact if the separating plane has the largest distance to the nearest training data points of any class, it lowers the generalization error of the overall classifier. The test points or query points are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

Classification consists of two steps: training and testing

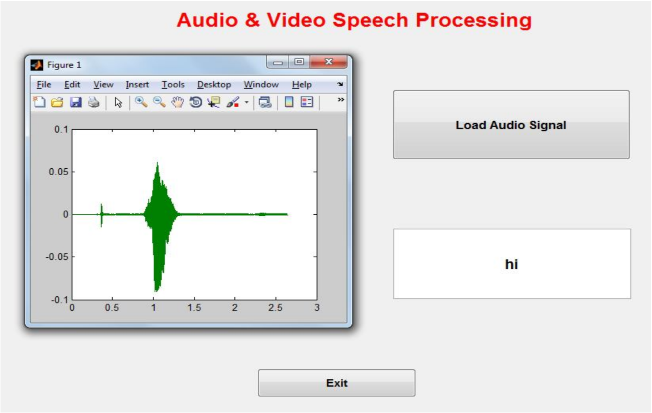
- In the training phase, SVM receives some feature patterns as input. These patterns are the extracted speech features represented by N feature parameters that can be seen as points in N-dimensional space. Then the classifying machine becomes able to find the labels of new vectors by comparing them with those used in the training phase.
- In the testing phase, SVM constructs a hyper plane which can be used for classification. A good separation is achieved by the hyper plane that has largest distance to the nearest training- data point of any class. Given a set of training examples, each marked as belonging to one or the other categories, an SVM training algorithm builds a model that assigns frames of the loaded audio signal into one of the categories previously defined. The categories are divided by a clear gap and based on the gap the new examples are classified and fed into the TextConversion.m script wherein the label value is assigned to the results based on the accuracy obtained due to the matching/classification done by the SVM.

## 4. RESULTS AND DISCUSSION

In both visual and audio speech to text conversion system, 12 video files such as Hi, Hello, Sorry, Thank you, Welcome, Good Morning, Good Afternoon, Good Evening, Good Night, Where are you?, What are you doing?, What is your name? are fed individually and the resulting texts are displayed as shown in Figure 2 and Figure 3 respectively.

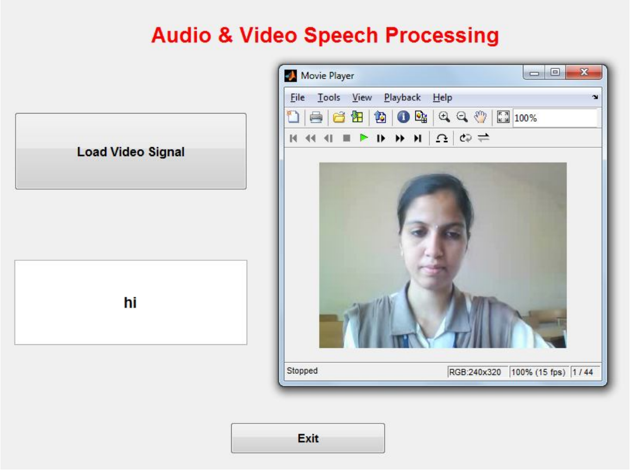
## 5.ACKNOWLEDGEMENT

The authors would like to thank the people who are both directly and indirectly involved in helping us develop this project. We would also like to thank our mentors who have given their valuable feedback and suggestions.


Audio & Video Speech Processing
Figure 1
File Edit View Insert Tools Desktop Window Help
0.1
0.05
0
-0.05
-0.1
0
0.5
1
1.5
2
2.5
3
Load Audio Signal
Exit

hi

**Figure 2** Audio waveform & the resulting text – Hi



hi

**Figure 3** Video player: Hi & the resulting text

## References

[1] Lai Pei Mei: “Interpretation of Alphabets by Images of Lip Movement For Native Language”, 2014.

[2] Chadawan Ittichaichareon, Siwat Suksri and Thaweesak Yingthawornsuk: “Speech Recognition using MFCC”, International Conference on Computer Graphics, Simulation and Modeling (ICGSM’2012) July 28-29, 2012.

[3] Lindasalwa Muda, Mumtaj Begam and I. Elamvazuthi: “Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques”, Journal of Computing, Volume 2, Issue 3, March 2010.

[4] Siddarth Deokar: K_Nearest_Neighbour_Algorithm, 2009.

[5] Nuzhat Atiqua Nafis and Md. Safaet Hossain: “Speech to Text Conversion in Real Time”, International Journal of Innovation and Scientific Research ISSN 2351-8014 Vol. 17 No. 2 Aug, 2015.