Gender Voice Recognition Using Auto-correlation and Pitch Frequency

Abdallah Kharouf
1183328

Ahmaide Al Awawdah
1190823

Raji Abdallah
118095

***Abstract*—Sounds processing techniques are becoming a very important application worldwide, and that is due to the huge development of Artificial Intelligence (AI) techniques and algorithms. Machine Learning also played very critical role in sounds processing in all forms, starting from detection, recognition or identification. This paper proposes an algorithm that recognizes the gender voices (male or female), based on the pitch frequency of the voice after driving the extracted data to a pre-defined threshold comparator. The pitch frequency is calculated using voice framing analysis using Auto-correlation. This paper uses MATLAB and Python Libraries which provides built-in methods for voice analysis, and processing. In this paper, the flow would contain an overview of the research, procedure of processing, result presentation, and result verification.**

■ **INTRODUCTION** Many applications recently in modern companies, and research labs require an accurate algorithms and implementations in fields of detection, classification, segmentation, and recognition. The huge revolution in data led to a breakthrough in developing new methods for processing in most of computer applications. Voice processing is a state-of-the-art in recent decades, due to the promising future for new electronic circuits (i.e, IC's and computer chips) in electronic devices. To fetch the idea simpler; take mobile sensors as an example. These devices contain many hardware circuits that are responsible to recognize the voice, the background, the filtering, and the general recognition.

This project is to analyze given voice, to implement a satisfying algorithm to recognize the voice if it belongs to male speaker or female speaker. The problem seems simple, and it actually is. The approach to implement such algorithm, auto-correlation in digital signal processing would be the main technology for our recognition. There would be many implementations to achieve the goal that we are looking forward to compromise. Machine learning algorithms would be also an option,

Date of publication 14 June 2022; date of current version 13 June 2022

by using a binary classifier. But, this paper introduces the auto-correlation and pitch-frequency estimation as main approach to achieve the recognition goal through MATLAB and Python.

Auto-Correlation Overview

”Autocorrelation” is used to compare a signal with a time-delayed version of itself. If a signal is periodic, then the signal will be perfectly correlated with a version of itself if the time-delay is an integer number of periods. Along with related experiments, has implicated autocorrelation as a potentially important part of signal processing in voice recognition and human hearing.
To talk more mathematically, the autocorrelation corresponding to a delay time is calculated by:

- finding the value of the signal at a time t
- finding the value of the signal at a time t + delta
- multiplying those two values together
- repeating the process for all possible times, t, and then
- computing the average of all those products.

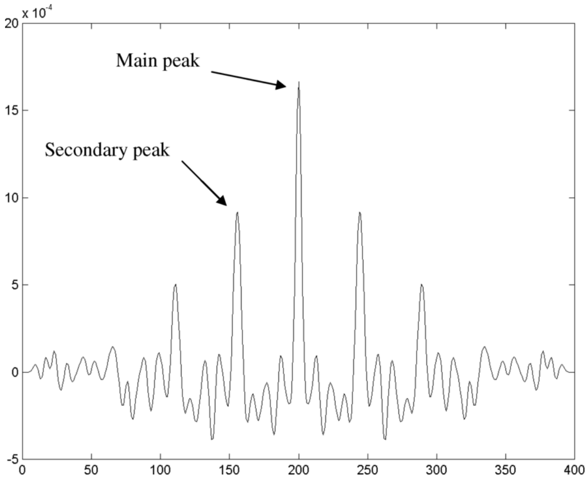


Figure 1. autocorrelation sample

Autocorrelation of Human Voice

Human Voice is considered very popular voice sample to be analyzed, and processed for many applications and softwares. As this paper is specialized to analyze and reconize the speaker’s gender, so autocorrelation would play very critical role in voice processing. Pitch Frequency is used in compatible way with autocorrelation to classify the frequencies in a voice. Autocorrelation is used to estimate pitch frequency, and will be demonstrated in the next subsection.

Pitch Frequency Estimation

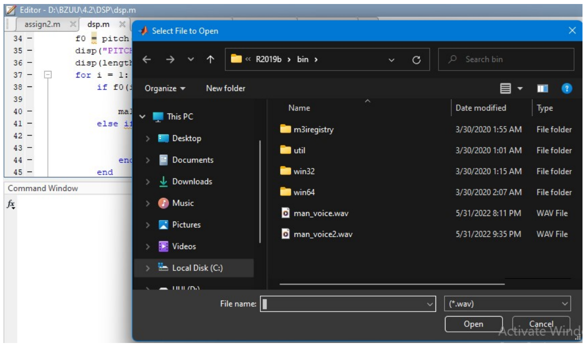
Pitch frequency or fundamental frequency of a signal forms very important information about the signal. In our application, we depend on pitch frequency to be applied on a threshold comparator to classify if the speaker is a male or a female. But, the approach in our paper is to estimate the pitch frequency using auto-correlation. More details will be introduced in approach and procedure section.

Approach and Procedure

As mentioned, we are tending to use autocorrelation to estimate pitch frequency at the first stages, in order to inject the produced value into threshold comparator to detect if the speaker is a male or a female.
In this stage, we will talk more technical. MATLAB and Python provide many pre-built functions for voice processing. At final stage, we will use a simple verification algorithm to verify our results using MATLAB. Our application is hybrid, we will use Python and MATLAB in compatible way to integrate results through processing to verification.

Audio File Reading

The format of audio file can be varied between .mp3 or .wav extensions. Two types of files are valid to be processed. The reading of the file will be user-interactive by giving possibility to user to load any local file in machine as follows.



```
        f0 = pitch
        disp("PITCH
        disp(length
        for i = 1:
            if f0(i

                mal
            else i


                end
            end
```

Name	Date modified	Type
m3iregistry	3/30/2020 1:55 AM	File folder
util	3/30/2020 1:01 AM	File folder
win32	3/30/2020 1:15 AM	File folder
win64	3/30/2020 2:07 AM	File folder
man_voice.wav	5/31/2022 8:11 PM	WAV File
man_voice2.wav	5/31/2022 9:35 PM	WAV File

File name:
(*.wav)

Figure 2. Select File window

After selecting desired file to be processed. The continuous time signal would be shown in order to have a scan on the signal which will be splitted and

processed.
The code below shows the first step or reading the file and displaying the corresponding signal.

```
[file,path] = uigetfile('*.wav');  % ...
    getting the file
if isequal(file,0)
   disp('You did not select any file ...
       - operation terminated');
   [file,path] = uigetfile('*.wav');
else
    f = fullfile(path,file); % ...
        assigning file to path
    disp(['You selected: ',f]);
    % read wav file
    [y,Fs] = audioread(f);
    sound(f); % lestining the audio ...
        loaded
    N = length(y); % sample length
    slength = N/Fs; % total time ...
        span of audio signal
    disp(slength)
    t = linspace(0, N/Fs, N);
    plot(t, y); % pplots the audio
    disp(Fs)
```

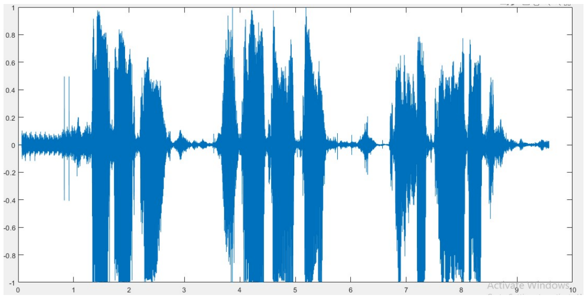


Figure 3. Voice of Signal Sample

As voice is heard, and displayed. We are now moving to the first step of the processing of the voice which we loaded.

Sampling Frequency and Framing

Simple short-time pitch frequency (or fundamental frequency F0) which can be estimated from the short frames (20-30ms) using auto-correlation method as mentioned and introduced. For our model, we are tending to split the audio file into N frames, with t seconds overlapping. This step is to minimize number of computation of autocorrelation of the signal. Hence, we deal with individual frames alone and then to combine fetched data in order to retrieve the pitch frequency by set of proposed solutions that will be suggested.
The code below shows how framing is performed

```
  % do framing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   f_d = 0.2;
   f_size = round(f_d * Fs);
   n = length(y);
   n_f = floor(n/f_size);  %no. of ...
       frames
   temp = 0;

   % iterate through all frames
   for i = 1 : n_f
    frames(i,:) = y(temp + 1 : temp ...
        + f_size);
    temp = temp + f_size;

    [lags{i},acf] = ...
        xcorr(frames(i,:),frames(i,:));
   % taking random samples to scan
     if i==10
        [acf,lags{i}] = ...
            xcorr(frames(i,:),frames(i,:))
        stem(lags{i},acf);
     end
     if i==10
         figure;
         autocorr(frames(i,:));
     end

   lags{i} = lags{i}(2:end); % ...
       removing first element 1 in ...
       each lag
   end % for i= 1 : n_f
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

As seen in the above code segment, framing is done using sampling frequency with pre-defined value as needed to be frame size, and overlapping period among each frame. Also as seen, we used the pre-built autocorrelation function in matlab. We can use either xcorr() function or autocorr(), but xcorr() does not subtract the mean from the input, while autocorr() does.
To make sure that all the frames have been processes, we make a simple check as follows:

```
   if length(lags) == n_f
      disp("TRUE completion")
   else
      fprintf('Lost %d frames',n_f ...
          - length(lags))
  end
```

Proposed Solutions

In fact, there are many possibilities to apply to detect the best estimated pitch frequency, in order to determine later whether the voice belongs to a man or to a woman. In this section, we will propose number of approaches how to deal with autocorrelation values that we have found.

Counter-Based Approach

After getting the values of the auto-correlation of each frame in the dynamically-generated frames, which are basically represent max period which is correspondent to the first max peak, after elimination of the first detected value. Expecting that we have N frames and each one contains max peak which can perform the conjugate of it to get list of frequencies at last.

Counter method is to iterate through all the generated frequencies in all frames and to set a counter incriminating based on this logic. The results will be analyzed and displayed later in results section.

```
for i = 1: length(f0)
    if f0(i) > MIN_MALE_THRESHOLD & ...
        f0(i) < MAX_MALE_THRESHOLD
        male_c = male_c + 1;
    else if f0(i) > ...
        MIN_FEMALE_THRESHOLD & f0(i) ...
        < MAX_FEMALE_THRESHOLD
        female_c = female_c + 1;
    end
    end
end
if male_c > female_c
    disp("MALE Voice")
else disp("FEMALE Voice");
end
```

Mean-based Solution

Another approach to have satisfying indication about the speaker of the voice based on the autocorrelation technique, is to pre-process the autocorrelation values of all the frames that have been under operation. Also, the technique can be use a hash map, to store each value to corresponding key (i.e, tag). Where the key represents the number of the frame (i.e, array index), and the value is the autocorrelation values that has been generated previously in autocorrelation step.

```
max_vals=[]
for i=1 : length(lags)
    max_vals = ...
        [max_vals,max(lags{i})];
end
max_in_max=[]
min_in_max=[]
for i=1 :length(max_vals)
    if(max_vals(i) > ...
        MIN_FEMALE_THRESHOLD)
         max_in_max = ...
             [max_in_max,max_vals(i)]
    end
    else if max_vals(i) > ...
        MIN_MALE_THRESHOLD &
            max_vals(i) < ...
                MAX_MALE_THRESHOLD
            min_in_max = ...
                [min_in_max,max_vals(i)]
        end
        last_result = [max_in_max ...
            min_in_max]
        disp("PITCH Frequency :")
        disp(1/mean(last_result))
end
```

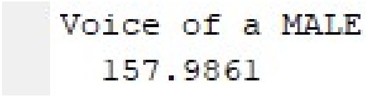
Results Analysis and Test Cases

As has been mentioned, we have seen the flow of the processing to identify the gender of the speaker through set of operations that are applied consequently. For now, we will apply some records from the data set where it was collected through free internet sound providers.

In this section, we will use Python also with Matlab to compare some results, and finally to **verify** our results using Matlab.

Test Case 1 - Man's Voice

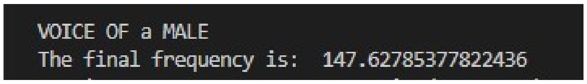
In this level, we are ready to test our model through number of records to analyze the results, and to see if the model is accurate enough to be utilized later. Starting to test a man's voice, as follows:



```
Voice of a MALE
  157.9861
```

Figure 4. Result of man voice - counter base algorithm

Let's now try the same voice on the second method we have (i.e, the mean-based method).



```
VOICE OF a MALE
The final frequency is:  147.62785377822436
```

Figure 5. Result of man voice - mean-based on Python

Test Case 2 - Woman's Voice

Now, let's try a voice of a woman in both approaches, in order to detect if there are some differences. Please not that verification step is done after this stage using **pitch()** function in Matlab.

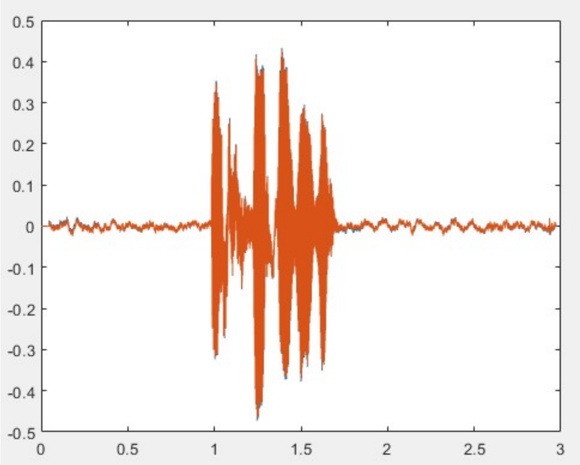
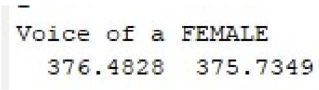
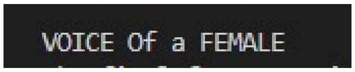


Figure 6. Signal of Woman's voice



```
-
Voice of a FEMALE
  376.4828   375.7349
```

Figure 7. Counter-based algorithm



```
VOICE Of a FEMALE
```

Figure 8. Mean-based algorithm

Results Verification and Comparison

At last step, we will utilize **pitch()** function in Matlab to verify our results, and then to display the information in tabular form.

	Counter-based	Mean-based	pitch()
no. test cases	6	6	6
successful recognition	5	4	6
fail recognition	1	2	0
accuracy*	83%	67%	100%

*: accuracy is calculated based on the 6 test cases that have been injected into the model. It may be different when injecting large numbers of cases

CONCLUSION

As a conclusion, autocorrelation technique in intergation with framing and proposed measurement and threshold comparison for voice gender identification seem promising for normal and fast applications. As in result verification and comparison, we declare that we have developed differnet methods in order to find the near-optimal algorithm, and we noticed the variety of test cases we have injected. At last, we could say that all algorithms gave somehow satisfying results, on regardless of the variety of the voices, and the effects in each voice.

REFERENCES

1. Pala, Mahesh. (2016). A New Human Voice Recognition System. AJSAT. 5. 23-30.
2. Elasmi-Ksibi, Rim Besbes, Hichem Lopez-Valcarce, Roberto Cherif, Sofiane. (2010). Frequency estimation of real-valued single-tone in colored noise using multiple autocorrelation lags. Signal Processing. 90. 2303-2307. 10.1016/j.sigpro.2010.01.025.
3. Magimai-Doss, Mathew Stephenson, Todd Bourlard, Herve. (2003). Using pitch frequency information in speech recognition.