Mimic Human Speech in Bahasa Indonesia Using Speech Recognition and Speech Synthesis

Valens Prabagita Ivan Susilo
Department of Computer Science
President University
Cikarang, Bekasi, 17550, Indonesia
prabagita12@gmail.com

Abstract

Everyday people use speech recognition and speech synthesis unconsciously. The technologies help them with their activities. With each technology can produce any kinds software related to speech. Combine both of technologies can produce many more. One of the combinations is mimic human speech. This research will discuss about Speech Recognition that use Convolutional Neural Network to as machine learning model and Speech Synthesis that use Concatenative Synthesis with syllables as speech unit. The purpose of this research is to develop application to collect, train, and mimic speech in Bahasa Indonesia. User can participate record their speech. The collected speech will be train to be used in the application to recognize the speech. After the collected speech is trained, User can mimic their speech by identify or recognize the speech and generate or synthesis the speech. The application to collect and mimic speech develops in website application.

1 Introduction

“Ok Google, play some music”. “Siri, what should I eat for lunch?”. Everyday people use their artificial assistance to boost their activities. People very like to use it because they just asked to their device and then in seconds, the wish is granted. It seems like, people are talking to the computer. The truth is, speech recognition takes big role with the help of machine learning. Google Assistance, Apple Siri, Microsoft Cortana, Amazon Alexa, and others have thousands of speech data to be analysed with the machine learning and they easily add data by collecting people speech from the assistance with permission.

If speech recognition is the process to get data by analysed speech, the opposite of speech recognition is speech synthesis, the process to produce artificial speech. Therefore, speech recognition is known as speech-to-text and speech synthesis is known as text-to-speech. “Hey Cortana, read my email” command make artifical assistance generate speech from the email text. With each technology can produce any kinds software related to speech. Combine both of technologies can produce many more. One of the combinations is mimic human speech.

This research aims to develop application which can be used to collect speech data, train machine learning model with collected data and mimic speech in Bahasa Indonesia. The application to collect and mimic speech develops in website application. The application can recognize the speech and generate speech from text.

2 Limitation

The limitations of this application are as following:

- There are 9 selected syllables to be used in the application, a, i, na, ma, mu, di, ri, and ku.
- Recorded speech in 1 second, with sample rate 16000 and mono sound.
- Speech recognition data is taken from recorded speech in Bahasa Indonesia.
- Speech synthesis data is taken from saved speech, result from speech recognition.
- Application is developed as website application.

3 Method

The approach used to achieve this research objectives are using techniques from speech synthesis concatenative synthesis, speech recognition Mel Frequency Cepstral Coefficients (MFCC), and machine learning Convolutional Neural Network (CNN).

3.1 Concatenative Synthesis

Concatenative synthesis connecting pre-recorded natural utterances is probably the easiest way to produce intelligible and natural sounding synthetic speech. One of the most important aspects in concatenative synthesis is to find correct unit length.

The selection is usually a trade-off between longer and shorter units. With longer units, high naturalness, less concatenation points and good control of coarticulation are achieved, but the number of required units and memory is increased. With shorter units, less memory is needed, but the sample collecting and labelling procedures become more difficult and complex [Hande, 2014].

In present systems units used are usually words, syllables, demisyllables, phonemes, diphones, and sometimes even triphones. As there aren’t found any exact amount of Bahasa Indonesia phonemes, syllables can be the best options for the speech unit.

3.2 MFCC

MFCC is one of the most commonly used feature extraction method in speech recognition introduced by Davis and Mermelstein in the 1980's [practicalcrypthography.com, 2012].

3.2.1 Framing and Windowing

A step can be done before framing and windowing is to apply a pre-emphasis filter on the signal to amplify the high frequencies. The pre-emphasis filter can be applied to a signal $\boldsymbol{x}$ using the first order filter in the following equation where typical values for the filter coefficient ($\boldsymbol{\alpha}$) are 0.95 or 0.97 [haythamfayek.com, 2016]:

$$\boldsymbol{y}(\boldsymbol{t}) = \boldsymbol{x}(\boldsymbol{t}) - \boldsymbol{\alpha x}(\boldsymbol{t} - \mathbf{1})$$

Framing is done because of an audio signal is constantly changing, so to simplify things, assuming that on short time scales the audio signal doesn't change. Typically, signal is framing into 20-40ms frames (25ms is standard). Frame stripe typically is 10ms, which allows some overlap to the frames. If the speech file does not divide into an even number of frames, pad it with zeros so that it does.

After slicing the signal into frames, apply a window function such as the Hamming window to each frame can be done. Hamming window has the following form where, $\mathbf{0} \leq \boldsymbol{n} \leq \boldsymbol{N}-\mathbf{1}$, $\boldsymbol{N}$ is the window length:

$$\boldsymbol{w}[\boldsymbol{n}] = \mathbf{0.54} - \mathbf{0.46}\,\boldsymbol{cos}\left(\frac{\mathbf{2}\boldsymbol{\pi n}}{\boldsymbol{N}-\mathbf{1}}\right)$$

3.2.3 Discrete Fourier Transform and Power Spectrum

Compute each window with Discrete Fourier Transform (DFT) or Fast Fourier Transform (FFT) can be followed with compute power spectrum (periodogram). Periodogram use the following equation where, $\boldsymbol{x_i}$ is the $\boldsymbol{i^{th}}$ frame of signal $\boldsymbol{x}$ and $\boldsymbol{N}$ is FFT size as a power of two greater than or equal to the number of samples in a single window length:

$$\boldsymbol{P} = \frac{|\boldsymbol{FFT}(\boldsymbol{x_i})|^2}{\boldsymbol{N}}$$

3.2.4 Mel Filterbank

The periodogram spectral estimate still contains a lot of information not required for speech recognition. Take clumps of periodogram bins and sum them up to get an idea of how much energy exists in various frequency regions. This is performed by mel filterbank.

Computing mel filterbank is applying triangular filters, typically 20 – 40 (26 or 40 is standard) filters, on a mel-scale to the power spectrum to extract frequency bands. The formula to convert between Hertz ($\boldsymbol{f}$) and Mel ($\boldsymbol{m}$) using the following equations:

$$\boldsymbol{m} = \mathbf{2595}\ \mathbf{log_{10}}\left(\mathbf{1} + \frac{\boldsymbol{f}}{\mathbf{700}}\right)$$

$$\boldsymbol{f} = \mathbf{700}\left(\mathbf{10}^{\frac{\boldsymbol{m}}{\mathbf{2595}}} - \mathbf{1}\right)$$

Each filter in the filterbank is triangular having a response of 1 at the center frequency and decrease linearly towards 0 till it reaches the center frequencies of the two adjacent filters where the response is 0. Good values are to start filter from 300Hz for the lower and up to 8000Hz for the upper frequency. The filterbank can be modelled by the following equation:

$$\boldsymbol{H_m}(\boldsymbol{k}) = \begin{cases} \mathbf{0}, & \boldsymbol{k} < \boldsymbol{f}(\boldsymbol{m}-\mathbf{1}) \\ \frac{\boldsymbol{k}-\boldsymbol{f}(\boldsymbol{m}-\mathbf{1})}{\boldsymbol{f}(\boldsymbol{m})-\boldsymbol{f}(\boldsymbol{m}-\mathbf{1})}, & \boldsymbol{f}(\boldsymbol{m}-\mathbf{1}) \leq \boldsymbol{k} < \boldsymbol{f}(\boldsymbol{m}) \\ \mathbf{1}, & \boldsymbol{k} = \boldsymbol{f}(\boldsymbol{m}) \\ \frac{\boldsymbol{f}(\boldsymbol{m}+\mathbf{1})-\boldsymbol{k}}{\boldsymbol{f}(\boldsymbol{m}+\mathbf{1})-\boldsymbol{f}(\boldsymbol{m})}, & \boldsymbol{f}(\boldsymbol{m}) < \boldsymbol{k} \leq \boldsymbol{f}(\boldsymbol{m}+\mathbf{1}) \\ \mathbf{0}, & \boldsymbol{k} > \boldsymbol{f}(\boldsymbol{m}+\mathbf{1}) \end{cases}$$

3.2.5 Logarithm

Once compute the mel filterbank, the next is simply take the logarithm of them. Generally, to double the perceived volume of a sound it needs to put 8 times as much energy into it.

3.2.6 Discrete Cosine Transform

The final step is to compute the Discrete Cosine Transform (DCT). The DCT decorrelates the energies which means diagonal covariance matrices can be used to model the features. But only 12 of the DCT coefficients are kept. This is because the higher DCT coefficients represent fast changes in the filterbank energies and it turns out that these fast changes actually degrade speech recognition performance, so dropping them will get a small improvement.

3.3 Convolutional Neural Network

Convolutional neural network (CNN or ConvNet) is one of known variants neural network model to recognized image [wikipedia.org, 2018]. The model is designed to recognize an object no matter what surface the object is on. The model doesn't have to re-learn the idea of child for every possible surface it could appear on [Geitgey, 2016].

3.3.1 Convolution Layer

Convolution layer is the layer to feed the pre-processing image or another output into small neural network. The small neural network treating every image or output equally. It will mark if something interesting appears as the model learning.

3.3.2 Max-pooling Layer

Max-pooling or down sampling is the layer to reducing the output by finding maximum value in the output. The output broke down into equal pool size and stride or slide into entire output. Then, each pool is found the maximum value.

3.3.3 Fully-connected Layer

Fully-connected is the layer to high-level reasoning in the dense neural network to recognize the image.

4 Experimental Result

In order to evaluate the effectiveness of the proposed methods in the previous section within the application, experiments are done to ensure the application runs well.

As speech synthesis just concatenative, the evaluation focus on evaluate speech recognition. The dataset during the speech recognition evaluation is 10000 male and female speech data, each 500 on each syllable, on not noisy background and each 500 unknown sound. Corrected result shows from the more than 75% of model accuracy. The table also fill with accuracy with the syllable result along with it. Random or unknown condition is tested with silent condition, o, and mi syllables.

The scenarios have 4 testers. Male user that the records is trained by the application is called Tester 1. Female user that the records is trained by the application is called Tester 2. Male user that the record hasn't trained by the application called Tester 3. Female user that the record hasn't trained by the application called Tester 4. Every tester is test in each the following environment:

- Noisy background (Loud music or people chit-chat).

- Semi noisy background (Rain noise or sound from the other rooms).
- Not noisy background.

Table 1: Tester 1 results on noisy background.

No	Spoken Syllable	Noisy Background			Correct Result
		1	2	3	
1	a	52 *(a)*	100 *(a)*	100 *(unknown)*	1/3
2	i	62 *(na)*	85 *(ri)*	96 *(ma)*	0/3
3	na	100 *(unknown)*	86 *(kan)*	100 *(unknown)*	0/3
4	ma	100 *(a)*	84 *(unknown)*	100 *(a)*	0/3
5	mu	99 *(kan)*	71 *(ku)*	88 *(unknown)*	0/3
6	di	59 *(mu)*	100 *(ri)*	97 *(ri)*	0/3
7	ri	50 *(na)*	100 *(ri)*	92 *(a)*	1/3
8	ku	100 *(ku)*	85 *(a)*	100 *(kan)*	1/3
9	kan	72 *(unknown)*	61 *(kan)*	79 *(kan)*	3/3
10	Unknown 1 *(silent)*	99 *(unknown)*	100 *(a)*	99 *(unknown)*	2/3
11	Unknown 2 *(o)*	100 *(na)*	100 *(a)*	100 *(ku)*	0/3
12	Unknown 3 *(mi)*	100 *(a)*	51 *(kan)*	96 *(ma)*	1/3

Table 2: Tester 1 results on semi noisy background.

No	Spoken Syllable	Semi Noisy Background			Correct Result
		1	2	3	
1	a	100 *(a)*	100 *(a)*	100 *(a)*	3/3
2	i	100 *(i)*	36 *(ri)*	100 *(i)*	2/3
3	na	97 *(mu)*	48 *(ma)*	99 *(na)*	1/3
4	ma	49 *(i)*	98 *(ri)*	79 *(mu)*	0/3
5	mu	55 *(ri)*	99 *(ku)*	50 *(ri)*	0/3
6	di	57 *(i)*	100 *(di)*	100 *(i)*	1/3
7	ri	100 *(ri)*	75 *(ri)*	100 *(ri)*	3/3
8	ku	92 *(ku)*	68 *(ku)*	91 *(ku)*	2/3
9	kan	100 *(kan)*	94 *(na)*	100 *(kan)*	2/3
10	Unknown 1 *(silent)*	100 *(a)*	99 *(unknown)*	100 *(ma)*	2/3
11	Unknown 2 *(o)*	100 *(ku)*	89 *(ku)*	72 *(ku)*	1/3
12	Unknown 3 *(mi)*	100 *(ri)*	94 *(ri)*	100 *(ri)*	0/3

Table 3: Tester 1 results on not noisy background.

No	Spoken Syllable	Not Noisy Background			Correct Result
		1	2	3	
1	a	100 (a)	100 (a)	100 (a)	3/3
2	i	94 (i)	48 (kan)	94 (i)	2/3
3	na	80 (na)	99 (na)	100 (na)	3/3
4	ma	96 (di)	90 (i)	57 (di)	0/3
5	mu	100 (i)	86 (ku)	95 (i)	0/3
6	di	99 *(ri)*	100 *(i)*	80 *(ri)*	0/3
7	ri	100 *(ri)*	96 *(ri)*	91 *(ri)*	3/3
8	ku	85 *(ku)*	96 *(ku)*	100 *(ku)*	3/3
9	kan	92 *(kan)*	100 *(kan)*	100 *(kan)*	3/3
10	Unknown 1 *(silent)*	98 *(unknown)*	98 *(unknown)*	98 *(unknown)*	3/3
11	Unknown 2 *(o)*	100 *(ku)*	98 *(ku)*	65 *(ku)*	1/3
12	Unknown 3 *(mi)*	54 *(mu)*	100 *(ri)*	67 *(mu)*	2/3

Table 4: Tester 2 results on noisy background.

No	Spoken Syllable	Noisy Background			Correct Result
		1	2	3	
1	a	99 *(unknown)*	99 *(a)*	99 *(a)*	2/3
2	i	82 *(unknown)*	98 *(ma)*	99 *(ri)*	0/3
3	na	52 *(ri)*	99 *(ma)*	61 *(ma)*	0/3
4	ma	50 *(unknown)*	37 *(unknown)*	95 *(ri)*	0/3
5	mu	99 *(unknown)*	52 *(di)*	99 *(unknown)*	0/3
6	di	99 *(i)*	99 *(unknown)*	99 *(unknown)*	0/3
7	ri	86 *(i)*	20 *(kan)*	57 *(di)*	0/3
8	ku	99 *(ku)*	68 *(kan)*	90 *(unknown)*	1/3
9	kan	84 *(kan)*	99 *(kan)*	78 *(a)*	2/3
10	Unknown 1 *(silent)*	100 *(unknown)*	100 *(unknown)*	100 *(unknown)*	3/3
11	Unknown 2 *(o)*	99 *(ma)*	99 *(ku)*	99 *(ku)*	0/3
12	Unknown 3 *(mi)*	36 *(di)*	96 *(mu)*	99 *(di)*	1/3

Table 5: Tester 2 results on semi noisy background.

No	Spoken Syllable	Semi Noisy Background			Correct Result
		1	2	3	
1	a	100 *(a)*	100 *(unknown)*	99 *(a)*	2/3
2	i	92 *(ri)*	97 *(unknown)*	96 *(ri)*	0/3
3	na	83 *(ma)*	62 *(unknown)*	75 *(kan)*	0/3
4	ma	87 *(unknown)*	64 *(ma)*	90 *(na)*	2/3
5	mu	76 *(ma)*	78 *(ku)*	66 *(di)*	0/3
6	di	58 *(mu)*	48 *(mu)*	30 *(kan)*	0/3
7	ri	99 *(kan)*	89 *(kan)*	99 *(ri)*	1/3
8	ku	99 *(unknown)*	99 *(ku)*	88 *(unknown)*	1/3
9	kan	99 *(kan)*	65 *(kan)*	85 *(kan)*	2/3
10	Unknown 1 *(silent)*	100 *(unknown)*	79 *(a)*	99 *(ri)*	1/3
11	Unknown 2 *(o)*	99 *(a)*	82 *(ku)*	99 *(na)*	0/3
12	Unknown 3 *(mi)*	89 *(kan)*	35 *(ku)*	96 *(ri)*	1/3

Table 6: Tester 2 results on not noisy background.

No	Spoken Syllable	Not Noisy Background			Correct Result
		1	2	3	
1	a	88 *(a)*	99 *(a)*	99 *(a)*	3/3
2	i	99 *(i)*	99 *(i)*	52 *(i)*	2/3
3	na	97 *(a)*	54 *(kan)*	99 *(a)*	0/3
4	ma	99 *(na)*	99 *(na)*	98 *(na)*	0/3
5	mu	81 *(i)*	95 *(i)*	44 *(i)*	0/3
6	di	99 *(i)*	36 *(i)*	96 *(i)*	0/3
7	ri	47 *(ri)*	32 *(unknown)*	48 *(i)*	0/3
8	ku	98 *(ku)*	67 *(kan)*	99 *(ku)*	2/3
9	kan	84 *(na)*	85 *(na)*	87 *(na)*	0/3
10	Unknown 1 *(silent)*	99 *(unknown)*	99 *(unknown)*	100 *(unknown)*	3/3
11	Unknown 2 *(o)*	100 *(ku)*	99 *(ku)*	95 *(ku)*	0/3
12	Unknown 3 *(mi)*	67 *(ri)*	78 *(di)*	77 *(i)*	1/3

Table 7: Tester 3 results on noisy background.

No	Spoken Syllable	Noisy Background			Correct Result
		1	2	3	
1	a	100 *(a)*	99 *(a)*	100 *(a)*	3/3
2	i	71 *(ma)*	99 *(ri)*	99 *(ri)*	0/3
3	na	70 *(ku)*	99 *(a)*	57 *(unknown)*	0/3
4	ma	99 *(a)*	100 *(a)*	100 *(a)*	0/3
5	mu	74 *(mu)*	96 *(kan)*	74 *(mu)*	0/3
6	di	99 *(ri)*	96 *(ri)*	89 *(ri)*	0/3
7	ri	100 *(unknown)*	90 *(na)*	86 *(a)*	0/3
8	ku	98 *(kan)*	64 *(mu)*	99 *(a)*	0/3
9	kan	99 *(na)*	100 *(kan)*	98 *(kan)*	2/3
10	Unknown 1 *(silent)*	99 *(a)*	99 *(a)*	100 *(a)*	0/3
11	Unknown 2 *(o)*	92 *(na)*	84 *(na)*	61 *(mu)*	1/3
12	Unknown 3 *(mi)*	64 *(kan)*	53 *(na)*	99 *(ri)*	2/3

Table 8: Tester 3 results on semi noisy background.

No	Spoken Syllable	Semi Noisy Background			Correct Result
		1	2	3	
1	a	100 *(a)*	99 *(a)*	99 *(a)*	3/3
2	i	99 *(ri)*	90 *(di)*	92 *(di)*	0/3
3	na	100 *(kan)*	99 *(kna)*	92 *(ma)*	0/3
4	ma	99 *(a)*	100 *(kan)*	100 *(kan)*	0/3
5	mu	99 *(mu)*	99 *(mu)*	65 *(ku)*	2/3
6	di	99 *(ri)*	100 *(ri)*	99 *(ri)*	0/3
7	ri	95 *(kan)*	76 *(i)*	78 *(di)*	0/3
8	ku	99 *(a)*	96 *(a)*	100 *(a)*	0/3
9	kan	99 *(a)*	96 *(a)*	99 *(a)*	0/3
10	Unknown 1 *(silent)*	99 *(ri)*	73 *(di)*	60 *(unknown)*	2/3
11	Unknown 2 *(o)*	100 *(a)*	99 *(ku)*	100 *(a)*	0/3
12	Unknown 3 *(mi)*	99 *(na)*	99 *(ri)*	99 *(i)*	0/3

Table 9: Tester 3 results on not noisy background.

No	Spoken Syllable	Not Noisy Background			Correct Result
		1	2	3	
1	a	100 *(a)*	100 *(a)*	100 *(a)*	3/3
2	i	92 *(ri)*	95 *(ri)*	41 *(ku)*	0/3
3	na	99 *(na)*	74 *(ku)*	80 *(a)*	1/3
4	ma	59 *(a)*	84 *(ku)*	48 *(kan)*	0/3
5	mu	99 *(mu)*	99 *(ri)*	99 *(ri)*	1/3
6	di	100 *(ri)*	99 *(i)*	99 *(i)*	0/3
7	ri	99 *(ri)*	96 *(ri)*	94 *(i)*	2/3
8	ku	73 *(di)*	100 *(mu)*	99 *(mu)*	0/3
9	kan	100 *(kan)*	99 *(kan)*	96 *(kan)*	3/3
10	Unknown 1 *(silent)*	98 *(unknown)*	98 *(unknown)*	98 *(unknown)*	3/3
11	Unknown 2 *(o)*	99 *(ku)*	68 *(mu)*	100 *(ku)*	1/3
12	Unknown 3 *(mi)*	99 *(ri)*	93 *(i)*	100 *(ri)*	0/3

Table 10: Tester 4 results on noisy background.

No	Spoken Syllable	Noisy Background			Correct Result
		1	2	3	
1	a	100 *(unknown)*	99 *(unknown)*	94 *(ma)*	0/3
2	i	97 *(unknown)*	99 *(ri)*	99 *(ri)*	0/3
3	na	100 *(na)*	99 *(ri)*	99 *(ma)*	1/3
4	ma	93 *(na)*	99 *(ma)*	100 *(na)*	1/3
5	mu	55 *(ri)*	99 *(unknown)*	43 *(unknown)*	0/3
6	di	99 *(ri)*	54 *(i)*	99 *(ma)*	0/3
7	ri	97 *(ri)*	74 *(ri)*	100 *(ri)*	2/3
8	ku	68 *(ma)*	92 *(ku)*	52 *(52)*	1/3
9	kan	48 *(ku)*	22 *(ku)*	99 *(a)*	0/3
10	Unknown 1 *(silent)*	100 *(unknown)*	100 *(unknown)*	100 *(unknown)*	3/3
11	Unknown 2 *(o)*	100 *(a)*	100 *(a)*	100 *(a)*	0/3
12	Unknown 3 *(mi)*	55 *(mu)*	99 *(ri)*	94 *(i)*	1/3

Table 11: Tester 4 results on semi noisy background.

No	Spoken Syllable	Semi Noisy Background			Correct Result
		1	2	3	
1	a	73 *(a)*	91 *(unknown)*	99 *(unknown)*	0/3
2	i	81 *(ri)*	99 *(i)*	100 *(ri)*	1/3
3	na	100 *(ma)*	99 *(unknown)*	99 *(unknown)*	0/3
4	ma	99 *(ma)*	99 *(na)*	78 *(ma)*	2/3
5	mu	81 *(unknown)*	99 *(unknown)*	99 *(unknown)*	0/3
6	di	100 *(ri)*	99 *(i)*	99 *(i)*	0/3
7	ri	85 *(ri)*	99 *(ri)*	98 *(ri)*	3/3
8	ku	96 *(ku)*	89 *(ku)*	99 *(a)*	2/3
9	kan	100 *(a)*	99 *(a)*	64 *(ma)*	0/3
10	Unknown 1 *(silent)*	99 *(unknown)*	99 *(ri)*	100 *(unknown)*	2/3
11	Unknown 2 *(o)*	99 *(a)*	100 *(a)*	99 *(a)*	0/3
12	Unknown 3 *(mi)*	43 *(a)*	59 *(ri)*	99 *(unknown)*	3/3

Table 12: Tester 4 results on not noisy background.

No	Spoken Syllable	Not Noisy Background			Correct Result
		1	2	3	
1	a	99 *(unknown)*	75 *(a)*	100 *(a)*	2/3
2	i	88 *(i)*	99 *(i)*	99 *(ri)*	2/3
3	na	99 *(na)*	99 *(ma)*	99 *(ma)*	1/3
4	ma	55 *(ma)*	72 *(kan)*	99 *(na)*	0/3
5	mu	99 *(unknown)*	94 *(unknown)*	28 *(i)*	0/3
6	di	89 *(i)*	99 (ri)	99 *(ri)*	0/3
7	ri	99 *(ri)*	100 *(ri)*	99 *(ri)*	3/3
8	ku	39 *(kan)*	99 *(ku)*	58 *(mu)*	1/3
9	kan	100 *(a)*	99 *(a)*	100 *(a)*	3/3
10	Unknown 1 *(silent)*	40 *(unknown)*	24 *(unknown)*	60 *(unknown)*	3/3
11	Unknown 2 *(o)*	100 *(a)*	99 *(a)*	100 *(a)*	0/3
12	Unknown 3 *(mi)*	60 *(ri)*	100 *(i)*	99 *(ri)*	1/3

Although noisy and semi background on every tester not show a good result, it still can predict about 2 or 3 correct spoken syllables with high accuracy. Not noisy background isn't guaranteed all spoken syllables are correct even on tester 1 and 2. But, the result is better than noisy and semi background. This is because all the training data is recorded on not noisy background. Also, noise removal or reduction is not applied when extraction process is done before training begin.

Tester 1 and 2 at most moment have good result than tester 3 and 4. But, at some moment tester 3 and 4 have better result than tester 1 and 2. The most good result on tester 1 and 2 is due to the training data is all contain tester 1 and 2. When tester 3 and tester 4 have better result it can be cause by the background condition or the microphone ability to record the speech.

Syllable ma, mu, di, unknown o, and unknown mi have bad result on most time compare to others. The unknown o and mi are caused by the unknown training data is random and mostly background noise instead of focusing o and mi. The machine learning model can't predict syllable ma, mu, and di well. Most times they have low accuracy but correct prediction or they predict the relative syllable, in example ri or i is the result on di. This can be improved by adding more the training data as the training data is still relatively small. An optimum machine learning model can also improve the result on them and also other syllables as well.

5 Discussion

In this section, there are some discussion why the proposed methods are used towards this research. The important points are:

- Concatenative Synthesis for Speech Synthesis
- MFCC for Speech Recognition
- CNN for Machine Learning model

Concatenative is simply what mimic speech needs, the speech that taken from recognized speech is saved and when one wants to generate speech, it loads the speech and concatenative corresponding to inputted text.

Each step in MFCC process has its own reason. As can be conclude that MFCC aims to mimic the non-linear human ear perception of sound, by being more discriminative at lower frequencies and less discriminative at higher frequencies.

Two main reason using CNN. First, the concatenative use speech unit. Each of the speech unit is small part of sentences or word. CNN have been considered an optimum way to small-footprint keyword, speech unit, spotting than other machine learning approach

[Sainath & Parada, 2015]. Second, MFCC extraction process can be plot into spectogram. CNN is the most common way to solve or analyzed visual imagenery data, including spectogram.

6 Conclusion

There are several conclusions that can be obtained from this research. First, the application enables to recognize speech in Bahasa Indonesia speech from record audio well with CNN and MFCC approach. Second, the application enables to generate speech in Bahasa Indonesia speech from text well with concatenative synthesis approach. That means the speech recognition n and speech synthesis works well in order to mimic speech.

In the future, a further research in the speech recognition is improving machine learning model. When there is no right or wrong in modelling the machine learning model, there is always optimal model to get the best prediction and accuracy. In the speech synthesis by removing some of silence or unused part of the speech and also reducing background will make generated speech more fluently and good to hear. In Bahasa Indonesia, a research in determining Bahasa Indonesia phonemes can be a big improvement since the application use syllables as concatenative synthesis.

7 Acknowledgements

The author would like to thank Mr. Tjong Wan Sen as the thesis advisor for his advices and support throughout the development of this thesis. Also thank to others lecturer for their support, knowledge, and experience during the university life.

References

[Hande, 2014] Hande, S. S. (2014). A Review on Speech Synthesis an Artificial Voice Production. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 8.

[practicalcrypthography.com, 2012] practicalcryptography.com. (2012). *Mel Frequency Cepstral Coefficient (MFCC) tutorial*. Retrieved from Practical Cryptography: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/

[haythamfayek.com, 2016] haythamfayek.com. (2016, April 21). *Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between*. Retrieved from Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between: https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

[wikipedia.org, 2018] wikipedia.org. (2018, December 26). *Convolutional neural network*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Convolutional_neural_network

[Geitgey, 2016] Geitgey, A. (2016, Juny 14). *Machine Learning is Fun! Part 3: Deep Learning and Convolutional Neural Networks*. Retrieved from Medium: https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721

[Sainath & Parada, 2015] Sainath, T. N., & Parada, C. (2015). *Convolutional Neural Networks for Small-footprint Keyword Spotting.* New York, NY, U.S.A: Google, Inc.