

# ARTIFICIAL INTELLIGENCE SPEECH RECOGNITION WITH MORSE CODE



Student Name:

1. Almika Abu Ghiffari - 001202200158
2. Clement Delano Samosir - 001202200174
3. Indah Novianti Setyoningrum - 001202200164
4. Rafly Athalla Pasya - 001202200182
5. Satria Ibnu Pamungkas - 001202200071

INFORMATICS STUDY PROGRAM

FACULTY OF COMPUTER SCIENCE

SUBMITTED AS A PARTIAL MARK OF ARTIFICIAL INTELLIGENCE

2023-2

PRESIDENT UNIVERSITY

Jababeka Education Park, Jl. Ki Hajar Dewantara,

Kota Jababeka, Cikarang Baru, Bekasi 17550 – Indonesia

## INTRODUCTION

The application, which utilizes speech recognition AI, demonstrates the integration of AI technology. By incorporating the `speech\_recognition` library, the program highlights AI's ability to understand and interpret human speech, allowing users to interact with it through spoken commands.

The program also showcases AI's adaptability by processing recognized speech and executing contextually appropriate actions, as shown in its greeting and farewell responses. Furthermore, the application's ability to convert text into

Morse code and generate corresponding sounds illustrate AI's versatility in processing and transforming data across different modalities. In summary, this application's combination of AI components enables it to recognize, process, and respond to human speech, enhancing its usability and overall functionality

## METHOD

### The AI Algorithm Used in Our Project

The speech recognition functionality in my project relies on the Google Web Speech API, which utilizes advanced artificial intelligence techniques, particularly deep learning algorithms.

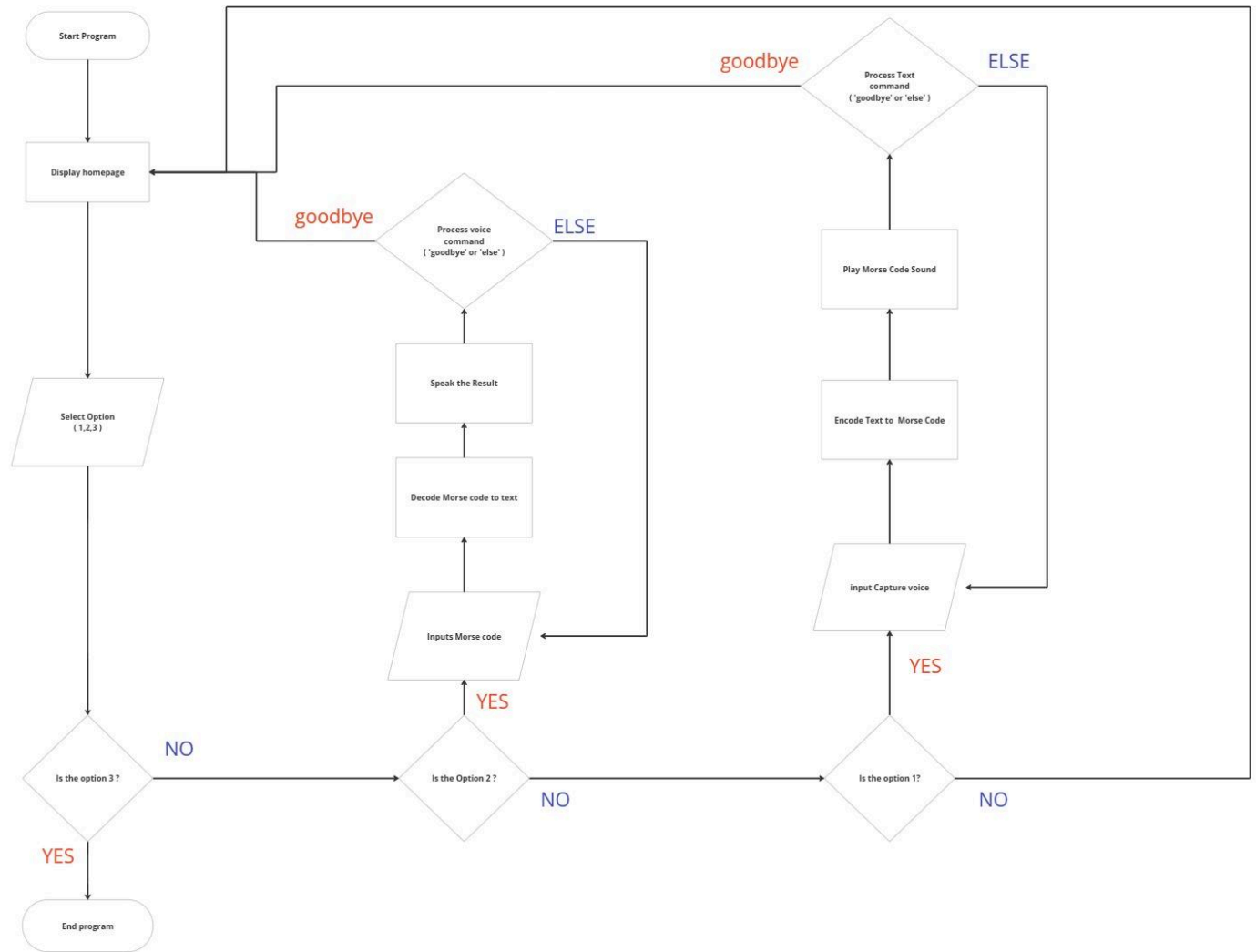
While the exact details of the algorithms used within the API are proprietary and not publicly disclosed, it's widely understood that deep neural networks (DNNs) play a significant role.

These deep learning algorithms are trained on large datasets of speech samples to learn patterns and relationships between audio features and corresponding text transcriptions. This enables the system to accurately transcribe spoken words into text, even in noisy environments or with various accents

- The Libraries Used:
  - ❖ `speech\_recognition`: Used for speech recognition and converting speech input into text. The speech recognition functionality in my project relies on the Google Web Speech API, which utilizes advanced artificial intelligence techniques, particularly deep learning algorithms.
  - ❖ `MorseCodePy`: Library providing functionalities to encode and decode text into Morse code, as well as play Morse code as sound. For encoding, each character in English is translated with a 'dot' and a 'dash' or sometimes just a single 'dot' or 'dash'. For decoding, the process is reversed
  - ❖ `pyttsx3`: Used for speech synthesis, converting text into speech. This library is typically a combination of text processing and speech synthesis techniques. It is a text-to-speech conversion library in Python.

- The Commands Function :
  - ❖ Function 'capture\_voice\_input()' : Utilizes the speech\_recognition module to capture speech input from the user via the microphone.
  - ❖ Function 'speak(text)' : Uses the pyttsx3 library to vocalize the given text using speech synthesis.
  - ❖ Function 'convert\_voice\_to\_text(audio)' : Employs 'speech\_recognition' to convert speech input into text.
  - ❖ Function 'encode\_to\_morse(text)' : Utilizes 'MorseCodePy' to convert text into Morse code.
  - ❖ Function 'process\_voice\_command'(text) : Uses the recognized text to generate a response or decide whether the program should terminate. Processes the recognized speech command converted into text.
  - ❖ Function 'morse\_to\_sound'(text) : Plays sound based on the given Morse code using 'MorseCodePy.'
  - ❖ Function 'decode\_morse\_input(cipherT)' : Decodes the given Morse code message into text using 'MorseCodePy.'
  - ❖ Function 'homepage()' : Prints the main menu of the program and prompts the user to select an option.
  - ❖ Function 'main()' : The main function of the program containing the main loop to execute the program. Processes user inputs and calls appropriate functions based on user's choice.

# FLOWCHART



## THE PROGRAM CODE

Line	Code
1.	import speech_recognition as sr
2.	from MorseCodePy import encode, play, decode
3.	import pyttsx3
4.	
5.	#pip install SpeechRecognition
6.	#pip install pyaudio
7.	#pip install MorseCodePy
8.	#pip install pyttsx3
9.	
10.	
11.	recognizer = sr.Recognizer()
12.	engine = pyttsx3.init()
13.	
14.	def capture_voice_input():
15.	with sr.Microphone() as source:
16.	print("Listening... (say 'goodbye' to quit)")
17.	audio = recognizer.listen(source)
18.	return audio
19.	
20.	def speak(text):
21.	engine.say(text)
22.	engine.runAndWait()
23.	
24.	def convert_voice_to_text(audio):
25.	#abbreviations source:
	<a href="https://www.yourdictionary.com/articles/morse-code-abbreviations">https://www.yourdictionary.com/articles/morse-code-abbreviations</a>
26.	abbreviations = {
27.	"i have no more to send" : "30",
28.	"best regards" : "73",
29.	"love and kisses" : "88",
30.	"all after" : "AA",
31.	"all before" : "AB",
32.	"amplitude modulation" : "AM",
33.	"break" : "BK",
34.	"break-in" : "BK",
35.	"all between" : "BN",
36.	"been" : "BN",

37.	"separation" : "BT",
38.	"before" : "B4",
39.	"yes, correct" : "C",
40.	"check" : "CK",
41.	"call" : "CL",
42.	"i am closing my station" : "CL",
43.	"calling any station" : "CQ",
44.	"see you" : "CU",
45.	"day" : "DA",
46.	"from" : "DE",
47.	"this is" : "DE",
48.	"down" : "DN",
49.	"dear" : "DR",
50.	"distance" : "DX",
51.	"element" : "EL",
52.	"and" : "ES",
53.	"excellent" : "FB",
54.	"frequency modulation" : "FM",
55.	"from" : "FM",
56.	"good afternoon" : "GA",
57.	"go ahead" : "GA",
58.	"good evening" : "GE",
59.	"going" : "GG",
60.	"good morning" : "GM",
61.	"good night" : "GN",
62.	"give" : "GV",
63.	"error in sending" : "HH",
64.	"the telegraph laugh" : "HI",
65.	"high" : "HI",
66.	"headquarters" : "HQ",
67.	"here" : "HR",
68.	"hear" : "HR",
69.	"have" : "HV",
70.	"how" : "HW",
71.	"how copy" : "HW",
72.	"leave" : "LV",
73.	"milliamperes" : "MA",
74.	"no" : "N",
75.	"negative" : "N",
76.	"incorrect" : "N",
77.	"nothing doing" : "ND",
78.	"no more" : "NM",

79.	"number" : "NR",
80.	"now" : "NW",
81.	"i resume transmission" : "NW",
82.	"old boy" : "OB",
83.	"old chap" : "OC",
84.	"old man" : "OM",
85.	"operator" : "OP",
86.	"oldtimer" : "OT",
87.	"old top" : "OT",
88.	"point" : "PT",
89.	"press" : "PX",
90.	"received as transmitted" : "R",
91.	"are" : "R",
92.	"decimal point" : "R",
93.	"ragchew" : "RC",
94.	"concerning; regarding" : "RE",
95.	"receive" : "RX",
96.	"receiver" : "RX",
97.	"sweepstakes" : "SS",
98.	"zero" : "T",
99.	"transmit" : "TR",
100.	"transmit or receive" : "T/R",
101.	"that" : "TT",
102.	"thank you" : "TU",
103.	"transmitter" : "TX",
104.	"you" : "U",
105.	"very" : "VY",
106.	"watts" : "W",
107.	"word after" : "WA",
108.	"word before" : "WB",
109.	"word" : "WD",
110.	"well" : "WL",
111.	"will" : "WL",
112.	"weather" : "WX",
113.	"young lady" : "YL",
114.	"year" : "YR",
115.	"about" : "ABT",
116.	"addressee" : "ADEE",
117.	"address" : "ADR",
118.	"again" : "AGN",
119.	"antenna" : "ANT",
120.	"broadcast interference" : "BCI",

121.	"broadcast listener" : "BCL",
122.	"be seeing you" : "BCNU",
123.	"better" : "BTR",
124.	"semi-automatic key" : "BUG",
125.	"confirm" : "CFM",
126.	"i confirm" : "CFM",
127.	"callbook" : "CLBK",
128.	"called" : "CLD",
129.	"calling" : "CLG",
130.	"can't" : "CNT",
131.	"conditions" : "CONDX",
132.	"circuit" : "CKT",
133.	"see you later" : "CUL",
134.	"come" : "CUM",
135.	"difference" : "DIFF",
136.	"delivered" : "DLVD",
137.	"for" : "FER",
138.	"ground" : "GND",
139.	"guess" : "GESS",
140.	"giving" : "GVG",
141.	"hope" : "HPE",
142.	"repeat" : "IMI",
143.	"say again" : "IMI",
144.	"info" : "INFO",
145.	"a poor operator" : "LID",
146.	"long" : "LNG",
147.	"later" : "LTR",
148.	"letter" : "LTR",
149.	"leaving" : "LVG",
150.	"typewriter" : "MILL",
151.	"message" : "MSG",
152.	"prefix to radiogram" : "MSG",
153.	"net control station" : "NCS",
154.	"nothing" : "NIL",
155.	"i have nothing for you" : "NIL",
156.	"operator" : "OPR",
157.	"preamble" : "PBL",
158.	"package" : "PKG",
159.	"please" : "PSE",
160.	"power" : "PWR",
161.	"received" : "RCD",
162.	"receiver" : "RCVR",



163.	"refer to" : "REF",
164.	"referring to" : "REF",
165.	"reference" : "REF",
166.	"radio frequency interference" : "RFI",
167.	"station equipment" : "RIG",
168.	"repeat" : "RPT",
169.	"report" : "RPT",
170.	"radio teletype" : "RTTY",
171.	"readability, strength, tone" : "RST",
172.	"self-addressed, stamped envelope" : "SASE",
173.	"said" : "SED",
174.	"says" : "SEZ",
175.	"signed" : "SGD",
176.	"signature" : "SIG",
177.	"signal" : "SIG",
178.	"operator's personal initials or nickname" : "SINE",
179.	"schedule" : "SKED",
180.	"signal for help needed" : "SOS",
181.	"sorry" : "SRI",
182.	"single sideband" : "SSB",
183.	"station" : "STN",
184.	"some" : "SUM",
185.	"service" : "SVC",
186.	"prefix to service message" : "SVC",
187.	"traffic" : "TFC",
188.	"tomorrow" : "TMW",
189.	"thanks" : "TNX",
190.	"tricks" : "TRIX",
191.	"that is" : "TTS",
192.	"television interference" : "TVI",
193.	"text" : "TXT",
194.	"your" : "UR",
195.	"yours" : "URS",
196.	"very fun business" : "VFB",
197.	"variable frequency oscillator" : "VFO",
198.	"words" : "WDS",
199.	"with" : "WID",
200.	"worked" : "WKD",
201.	"working" : "WKG",
202.	"words per minute" : "WPM",
203.	"word" : "WRD",
204.	"would" : "WUD",

```

205. "transceiver" : "XCVR",
206. "transmitter" : "XMTR",
207. "crystal" : "XTAL",
208. "wife" : "XYL",
209. "end of message" : "AR",
210. "stand by" : "AS",
211. "invite receiving station to transmit" : "BK",
212. "pause" : "BT",
213. "break for text" : "BT",
214. "beginning of message" : "KA",
215. "end of the transmission" : "KN",
216. "going off the air" : "CL",
217. "clear" : "CL",
218. "calling any amateur radio station" : "CQ",
219. "go, invite any station to transmit" : "K",
220. "go only, invite a specific station to transmit" : "KN",
221. "all received OK" : "R",
222. "end of contact" : "SK",
223. "understood" : "VE"
224. }
225.
226. try:
227.     text = recognizer.recognize_google(audio)
228.     print("You said:", text)
229.
230.     for phrase, abbreviation in abbreviations.items():
231.         if phrase.lower() in text.lower():
232.             text = text.replace(phrase.lower() + " ", abbreviation + " ")
233.
234.     print("Processed Text:", text)
235.     print()
236.
237. except sr.UnknownValueError:
238.     text = ""
239.     print("Sorry, I didn't understand that.")
240.     print()
241. except sr.RequestError as e:
242.     text = ""
243.     print("Error:", e)
244. return text
245.
246. def process_voice_command(text):

```

```
247.     if "hello" in text.lower():
248.         print("Hello! How can I help you?")
249.     elif "goodbye" in text.lower():
250.         print("Goodbye! Have a great day!")
251.         return True
252.     return False
253.
254. def encode_to_morse(text):
255.     if text:
256.         cipherText = encode(text, language='english')
257.         print("The Morse is : " + cipherText)
258.         print()
259.         return cipherText
260.
261. def morse_to_sound(text):
262.     try:
263.         play(text, delay=0.3, volume=0.7)
264.     except AttributeError:
265.         pass
266.
267. def decode_morse_input(cipherT):
268.     if cipherT:
269.         try:
270.             plainT = decode(cipherT, language='english', dot='.')
271.             print("\nThe Real Message is : " + plainT)
272.             speak("The real message is " + plainT)
273.             print()
274.         except TypeError:
275.             pass
276.     else:
277.         print("You didn't enter anything. Please enter a Morse code message.")
278.
279.
280. def homepage():
281.     print("-----")
282.     print("Welcome to the Morse Code Converter!")
283.     print("-----")
284.     print()
285.     print("1. Encode to Morse Code")
286.     print("2. Decode from Morse Code")
287.     print("3. Exit")
288.     print()
```

```

289. choice = input("Please enter your choice: [1/2/3]: ")
290. print()
291.
292. return choice
293.
294.
295. def main():
296.     while True:
297.         choice = homepage()
298.
299.         if choice == '1':
300.             print("\nYou choose encode to Morse Code.")
301.
302.             end_program = False
303.             while not end_program:
304.                 audio = capture_voice_input()
305.                 text = convert_voice_to_text(audio)
306.                 end_program = process_voice_command(text)
307.
308.                 if end_program:
309.                     break
310.
311.                 cipherT = encode_to_morse(text)
312.                 morse_to_sound(cipherT)
313.
314.             elif choice == '2':
315.                 print("\nYou choose to decode from Morse Code.")
316.                 print()
317.
318.                 while True:
319.                     print("\nexample input:")
320.                     print(".... . -.-. -.- --- --. / .-- --- .-. .-.. -.-")
321.                     cipherT = input('please enter the morse message that you want to
decode. (type end to quit)\n')
322.
323.                     if cipherT.lower() == "end":
324.                         print()
325.                         break
326.
327.                     decode_morse_input(cipherT)
328.
329.             elif choice == '3':

```

330.	print("\nExiting the program...")
331.	break
332.	
333.	else:
334.	print("\nInvalid choice! Please enter 1, 2, or 3.")
335.	
336.	
337.	if _name_ == "_main_":
338.	main()

## REFERENCE

- Morse code abbreviations:  
<https://www.yourdictionary.com/articles/morse-code-abbreviations>
- Morse code library and usage guide:  
<https://pypi.org/project/MorseCodePy/>
- Voice Recognition  
<https://reintech.io/blog/how-to-create-a-voice-recognition-system-with-python>

## LOG HOURS

Student Name	Activity	Log Hours
Almika Abu Ghiffari	Brainstroming/ Coding implementing the library	3 Hours
Clement Delano Samosir	Team Cordination and Writing project progress report	3 Hours
Indah Novianti Setyoningrum	Drafting flowchart and Writing project progress report	3 Hours
Rafly Athalla Pasya	Research on AI speech recognition with morse code and Writing project progress report	3 Hours
Satria Ibnu Pamungkas	Brainstorming/ Coding feature fuction	3 Hours