

		Głosowa łączność z komputerem		
Kierunek: Inżynieria Biomedyczna	Rok: V	Grupa: 1	Data wykonania: 18.04.2018	Data oddania: 19.04.2018
<b>Temat projektu:</b> Rozpoznawanie komend głosowych				
<b>Imię i nazwisko, nr. indeksu</b> Karolina Gajewska, 266673 Magdalena Hetmańska, 266684 Aneta Kalamaszek, 266689 Klaudia Kantor, 266691 Katarzyna Zawada, 266685				

## Cel projektu

Celem niniejszego projektu było utworzenie algorytmu przetwarzania i klasyfikacji sygnałów audio w celu rozpoznania wypowiedzianych komend głosowych na potrzeby obsługi “inteligentnego domu”.

## Rozwiązanie

### 1. Przygotowanie sygnałów

Pierwszym etapem projektu była rejestracja wybranych komend głosowych, które zostały wybrane na drodze “burzy mózgów” i uznane zostały za najbardziej przydatne w obsłudze domu. Każda z osób nagrała czterokrotnie ten sam zestaw komend rano, przed południem, po południu oraz wieczorem. Następnie w programie Audacity dokonano oznaczenia poszczególnych słów.

### 2. Otrzymanie wektora cech

Kolejnym etapem było podzielenie sygnału na pliki zawierające oddzielnie każde słowo zgodnie ze wcześniejszym oznaczeniem. Po otrzymaniu słów w oddzielnych plikach, następnym krokiem było wyznaczenie wektora cech dla każdego ze słów. W tym celu wykorzystano bibliotekę *librosa*, która dedykowana jest dla analizy sygnałów dźwiękowych.

Z biblioteki tej do wyznaczenia wektora cech sygnału wykorzystano funkcję *mfcc()*, która posiada zaimplementowany algorytm Mel-frequency Cepstral Coefficients (MFCCs). Do obliczenia odległości między otrzymanymi wektorami cech, wykorzystano algorytm Dynamic Time Warping (DTW), znajdujący się w bibliotece *dtw*.

### 3. Klasyfikacja

Ostatnim etapem było przeprowadzenie klasyfikacji słów na podstawie otrzymanych z DTW odległości. Działanie prostego klasyfikatora polega na tym, że sprawdza on do którego słowa z zestawu testowego klasyfikowany dźwięk ma najmniejszą odległość i to słowo jest przyporządkowywane. Gdy minimalna odległość między słowami przekracza 200, słowo nie zostaje sklasyfikowane. Całość kodu zawierająca przetwarzanie i klasyfikację, z dołączoną macierzą błędów znajduje się w pliku *classification\_with\_final\_matrix.ipynb*.

## Rezultaty

Zaimplementowane algorytmy zostały przetestowane na zarejestrowanych sygnałach. Wykorzystany algorytm MFCC+DTW pozwolił na uzyskanie zadowalających rezultatów. Testując algorytm na sygnałach osoby, na której był on uczony skuteczność wyniosła aż **91,28%**. Wynikową macierz błędów dla tego przypadku przedstawia rysunek 1. Sprawdzono także skuteczność algorytmu dla sygnałów nagranych przez inną osobę niż zbiór sygnałów uczących. W tym przypadku skuteczność algorytmu znacząco się pogorszyła i wyniosła jedynie **52.91%**. Wynikową macierz błędów przedstawia rysunek 2, całość macierzy dostępna jest w generowanym pliku Excel.

	KWIATKI	W	ZAPARZ	RADIO	DO	ZMIEN	ZAMKNIJ	DRZWI	KAWIE	KUCHNI	ZAGOTUJ	ROLETY	ALARM	SYPIALNI	WODE	STOPIEN	BRAM
KWIATKI	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
W	0	58.25	0	0	18.75	0	0	0	0	0	0	0	0	0	0	0	0
ZAPARZ	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RADIO	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	5
DO	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
ZMIEN	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
ZAMKNIJ	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
DRZWI	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
KAWIE	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
KUCHNI	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
ZAGOTUJ	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
ROLETY	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
ALARM	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	2
SYPIALNI	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
WODE	0	0	0	0	12.5	0	0	0	0	0	0	0	0	0	75	0	0
STOPIEN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
BRAM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10

Rysunek 1. Wynikowa macierz błędów algorytmu MFCC+DTW przy zbiorze uczącym i testowym od tej samej osoby.

	JEDEN	KANAL	KUCHNI	TELEWIZOR	ZAMKNIJ	ZAPAL	SYPIALNI	LAZIENCE	WLACZ	RADIO	KAWE	OGRZEWANIE	ALARM	ZWIEKSZ	STOPIEN
JEDEN	25	0	0	0	0	0	0	25	0	0	0	0	0	0	0
KANAL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KUCHNI	0	0	25	0	0	0	0	0	0	25	0	0	0	0	0
TELEWIZOR	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
ZAMKNIJ	0	0	0	0	50	0	0	0	0	25	0	0	0	0	0
ZAPAL	0	0	0	0	0	25	0	0	0	50	0	0	0	0	0
SYPIALNI	0	0	0	0	0	0	25	25	0	0	0	0	0	0	0
LAZIENCE	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0
WLACZ	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0
RADIO	0	0	0	0	0	0	0	0	0	50	0	0	0	0	25
KAWE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OGRZEWANIE	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
ALARM	0	25	0	0	0	0	0	0	12.5	25	0	0	0	0	0
ZWIEKSZ	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
STOPIEN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Rysunek 2. Wynikowa macierz błędów algorytmu MFCC+DTW przy różnym zbiorze uczącym i testowym.

Sprawdzono również skuteczność innych funkcji obliczających wektor cech, jednak nie przyniosły one pozytywnych rezultatów. Użycie funkcji *tonnetz()*, pozwoliło na uzyskanie skuteczności na poziomie 1,16 %, natomiast funkcja *spectral\_centroid()*, obniżyła skuteczność algorytmu do 6,4 %. Można więc z całą pewnością stwierdzić, że połączenie MFCC i DTW daje najlepsze rezultaty, co również potwierdzają liczne publikacje wykorzystujące te metody w algorytmach rozpoznania mowy.