

	<p align="center">Głosowa łączność z komputerem</p> <p>autor : Blanka Hasior nr indeksu: 266683</p>			
<p>Kierunek: Inżynieria Biomedyczna</p>	<p>Rok: V</p>	<p>Grupa: 1</p>	<p>Data wykonania: 31.08.2018</p>	<p>Data oddania: 01.09.2018</p>
<p>Temat projektu: Rozpoznawanie komend głosowych</p>				

Cel projektu

Celem projektu było stworzenie algorytmu do rozpoznawania wypowiedzianych komend głosowych na potrzeby inteligentnych domów, które stają się coraz bardziej popularne również w Polsce.

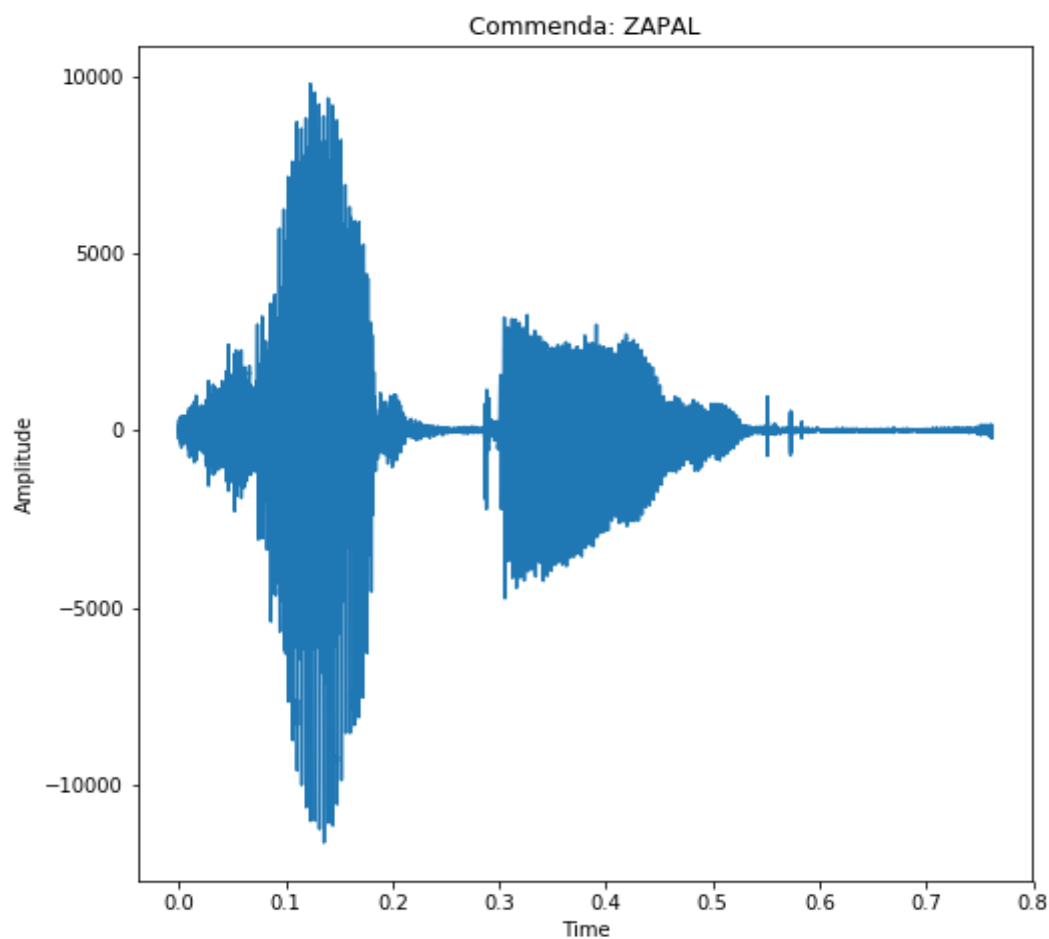
Etapy algorytmu

Algorytm został podzielony na dwie części : przetworzenie komend wypowiedzianych przez użytkownika inteligentnego domu oraz klasyfikacja komend za pomocy klasyfikatora. Algorytm został napisany za pomocą języka skryptowego python a wyświetlony w środowisku jupyter.

1. Nagrywanie sygnałów

Etap ten polegał na nagraniu poleceń najczęściej wydawanych w domach. Nagrania zostały nagrane o różnych pora dnia: rano, przed południem, po południu oraz wieczorem. W programie Audacity dokonano oznaczenia poszczególnych słów. Zestaw nagranych komend znajduje się poniżej:

1. Zapal światło w kuchni.
2. Otwórz drzwi do garażu
3. Włącz zmywarkę.
4. Wyłącz telewizor.
5. Podnieś rolety w sypialni.
6. Zamknij bramę.
7. Zwiększ ogrzewanie o jeden stopień.
8. Zakręć wodę w łazience.
9. Ustaw alarm.
10. Przycisz radio.
11. Zmień kanał.
12. Podleć kwiatki.
13. Zaparz kawę.
14. Wyłącz alarm w garażu.
15. Zagotuj wodę



Rys.1 Przedstawia komendę ‘Zapal’ wypowiedzianą rano z której będzie generowany wektor cech.

```
{'GARAZU': 8, 'SWIATLO': 4, 'STOPIEN': 4, 'OTWORZ': 4, 'KUCHNI': 4,
'RADIO': 4, 'ZMIEN': 4, 'PODLEJ': 4, 'OGRZEWANIE': 4, 'ZMYWARKE':
4, 'TELEWIZOR': 4, 'W': 16, 'SYPIALNI': 4, 'ROLETY': 4, 'ZAMKNIJ':
4, 'KANAL': 4, 'ZAPAL': 4, 'WYLACZ': 8, 'BRAME': 4, 'WODE': 8, 'DRZ
WI': 4, 'ALARM': 8, 'ZWIEKSZ': 4, 'WLACZ': 4, 'KAWE': 4, 'JEDEN':
4, 'DO': 4, 'PRZYCISZ': 4, 'ZAGOTUJ': 4, 'USTAW': 4, 'O': 4, 'PODNI
ES': 4, 'KWIATKI': 4, 'ZAKREC': 4, 'ZAPARZ': 4, 'LAZIENCE': 4}
```

Tab.1 Przedstawiająca ile razy powtarza się poszczególne słowa w komendach rannych, przed południem, po południu oraz wieczorem.

2. Wektora cech

Jednym z ważniejszych etapów algorytmu jest utworzenie wektora cech z każdego słowa znajdującego się w komendach. Zdecydowano się wykorzystać bibliotekę librosa, która odnosi się do analizy sygnałów dźwiękowych. Za wyodrębnienie wektora cech wykorzystano funkcję mfcc() w której zaimplementowany jest algorytm MFCCs, natomiast do obliczenia odległości między otrzymanymi wektorami ech wykorzystano algorytm DTW.

```
{ 'USTAW': array([-4.87952977e+02, -4.58362441e+02, -4.23583356e+02,
-4.05933201e+02, -3.99818058e+02, -3.68037632e+02,
-3.28656691e+02, -2.93514148e+02, -2.80598148e+02,
-3.03368134e+02, -3.44944699e+02, -3.88072770e+02,
-3.77204942e+02, -3.57934266e+02, -3.16702549e+02,
-2.63069691e+02, -2.36164611e+02, -2.49704832e+02,
-2.84771618e+02, -3.19815577e+02, -3.75623705e+02,
-4.58926697e+02, -4.88197214e+02, -4.94993561e+02,
-4.99169578e+02, -5.24308593e+02, -5.87283755e+02,
-6.54912708e+02, -6.83084827e+02, -6.81882029e+02,
-6.83993462e+02, -6.85843238e+02, -6.86354302e+02,
-6.86202276e+02, -6.86714225e+02, -6.84259903e+02,
-6.85952370e+02]),
```

Tab. 2 Utworzony wektor cech za pomocą MFCCs + DTW dla przykładowej komendy Ustaw.

3. Klasyfikacja

Ostatnim etapem było przeprowadzenie klasyfikacji słów. Działanie 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. K

	USTAW	KANAL	W	DO	ROLETY	O	JEDEN	PRZYCISZ	KAWĘ	RADIO
USTAW	2.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
KANAL	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
W	0.0	0.0	5.0	1.0	0.0	0.0	9.0	0.0	0.0	0.0
DO	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0
ROLETY	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	0.0
O	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0
JEDEN	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0
PRZYCISZ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0
KAWĘ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0
RADIO	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0
ZAKREC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DRZWI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
KWIATKI	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
SWIATŁO	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PODLEJ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ZAMKNIJ	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
WODE	0.0	1.0	0.0	1.0	0.0	0.0	4.0	0.0	0.0	0.0
PODNIES	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Tab.1 Macierz wynikowa klasyfikacji poszczególnych komend.

	USTAW	KANAL	W	DO	ROLETY	O	JEDEN	PRZYCISZ
USTAW	50.0	0.0	0.00	0.00	0.0	0.0	25.00	0.0
KANAL	0.0	50.0	0.00	0.00	0.0	0.0	0.00	0.0
W	0.0	0.0	31.25	6.25	0.0	0.0	56.25	0.0
DO	0.0	0.0	0.00	100.00	0.0	0.0	0.00	0.0
ROLETY	0.0	0.0	0.00	0.00	100.0	0.0	0.00	0.0
O	0.0	0.0	0.00	0.00	0.0	100.0	0.00	0.0
JEDEN	0.0	0.0	0.00	0.00	0.0	0.0	100.00	0.0
PRZYCISZ	0.0	0.0	0.00	0.00	0.0	0.0	0.00	100.0
KAWIE	0.0	0.0	0.00	0.00	0.0	0.0	0.00	0.0
RADIO	0.0	0.0	0.00	0.00	0.0	0.0	0.00	0.0
ZAKREC	0.0	0.0	0.00	0.00	0.0	0.0	0.00	0.0
DRZWI	0.0	0.0	0.00	0.00	0.0	0.0	0.00	0.0
KWIATKI	0.0	0.0	0.00	0.00	0.0	0.0	25.00	0.0
SWIATLO	0.0	0.0	0.00	0.00	0.0	0.0	0.00	0.0
PODLEJ	0.0	0.0	0.00	0.00	0.0	0.0	0.00	0.0
ZAMKNIJ	0.0	0.0	0.00	0.00	0.0	0.0	0.00	0.0
WODE	0.0	12.5	0.00	12.50	0.0	0.0	50.00	0.0
PODNIES	0.0	0.0	0.00	0.00	0.0	0.0	0.00	0.0

Tab.2 Macierz procentowa algorytmu klasyfikacji.

Rezultaty

Powstały algorytm pozwala na wykrycie komend dla większości słów około 56%. Należy zwrócić uwagę na wagę stworzonego algorytmu. Program poprawia jakość życia w inteligentnych domach, złe rozpoznanie komend nie spowoduje zagrożenia życia lub zdrowia człowieka, jedynie nie wykona określonych poleceń. W rezultacie wynik osiągniętego algorytmu można przyjąć jako wystarczający. Aby uzyskać lepszy wynik należało wyciągnąć lepszy wektor cech, poszukać bardziej skutecznego klasyfikatora. Poniżej przedstawiono kod realizujący procentowy stan algorytmu.

In [18]:

```
success = (171-error)/171*100
print('Success rate: %.2f percent'%(success))
```

Success rate: 55.56 percent

