

Projekt z SI

Dominik Lau, Mateusz Kowalczyk, Michał Tarnacki

11 kwietnia 2023

1 Wstęp

Celem projektu było określanie chwil na nagraniu, w których osoba bierze wdech i wydech. Dokonano oceny jakościowej za pomocą detekcji oddechu na żywo jak i ilościowej (przy wykorzystaniu dalej wymienionej metryk).

2 Teoria

2.1 DFT

W celu przejścia z dziedziny natężenia od czasu do dziedziny częstotliwości stosujemy dyskretną transformatę Fouriera. Funkcję natężenia w czasie przedstawiamy jako

$$I(t) = \sum_{\omega} I_{\omega}(t) \sin(\omega t)$$

stąd dla danego momentu w czasie jesteśmy w stanie stworzyć wektor natężeń

$$\mathbf{I} = [I_{f_1}, I_{f_2}, \dots, I_{f_n}]$$

W dalszych rozważaniach wykorzystujemy także średnią częstotliwość, którą liczymy ze wzoru

$$\bar{f} = \frac{\sum_f f I_f}{I}$$

Wykorzystujemy implementację transformaty z biblioteki *numpy*.

2.2 SVM

Pierwszym modelem, jakim dokonujemy klasyfikacji, jest SVM. Metoda ta polega na szukaniu optymalnych wag \mathbf{w}, b . Następnie będziemy klasyfikować według funkcji maszyny uczącej

$$f(x) = \text{sgn}(\mathbf{w}^T \mathbf{x} + b)$$

jeśli f zwraca 1 to traktujemy to jako jedną z binarnych decyzji (w naszym przypadku np. wdech) a jeśli zwraca -1 to drugą (czyli np. wydech). \mathbf{x} jest wektorem cech. Znalezienie optymalnych wag będzie polegało na minimalizacji funkcji kosztu

$$J(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{N} \sum_x \max\{0, 1 - y(\mathbf{w}^T \mathbf{x} + b)\}$$

gdzie C jest parametrem uregulowania. Funkcję będziemy minimalizować metodą spadku po gradientie. W projekcie użyliśmy własnej implementacji SVM-a.

2.3 Bramka szumów

Główna metoda odszumiania, z której korzystamy to bramka szumów. Pierw wyznaczany jest spektrogram sygnału i tworzony jest próg szumu dla każdej częstotliwości spektrogramu. Częstotliwości progowe służą do stworzenia maski, której potem używamy do usunięcia niechcianych dźwięków. Następnie ze spektrogramu tworzymy z powrotem natężenie $I(t)$. W projekcie wykorzystujemy gotową implementację bramki szumów z biblioteki *noisereduce*.

2.4 Filtr Savitzky-Golay

Filtru Savitzky-Golay używamy do wygładzenia funkcji. Polega on na dopasowywaniu zbioru sąsiadujących punktów do wielomianu niskiego stopnia za pomocą metody najmniejszych kwadratów. Te fragmenty wielomianów są potem łączone w wygładzoną funkcję. Korzystamy z gotowej implementacji z biblioteki *scipy*.

2.5 Ocena modeli

Dokładność (*accuracy*) definiujemy jako

$$accuracy = \frac{\#correct}{\#all}$$

czyli **ilości próbek** poprawnych do wszystkich.

2.6 Standaryzacja

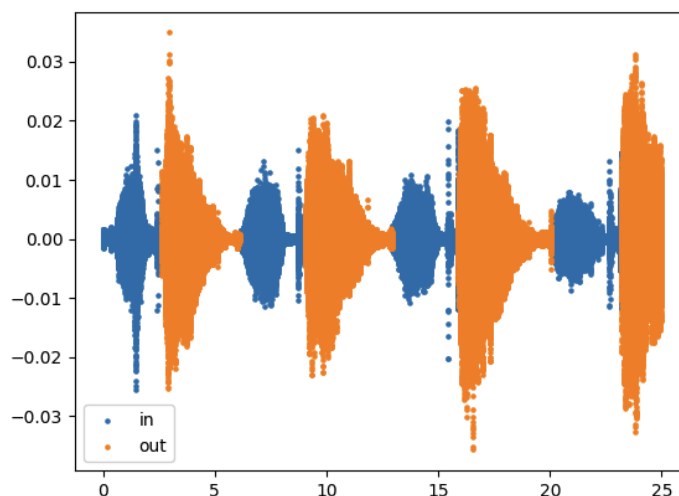
Dane wejściowe do modelu zawsze standaryzujemy, czyli

$$\tilde{X} = \frac{X - EX}{\sigma_X}$$

stosujemy gotową implementację z *scikit-learn*.

3 Nagrywanie danych oddechowych

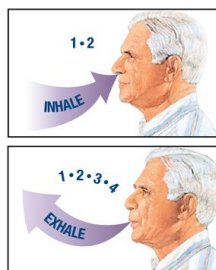
Żeby zapewnić dobre oznaczenie danych, etykietujemy je jeszcze w trakcie nagrywania dźwięku. Osoba nagrywana naciska przycisk, żeby zasignalizować, że przestała brać wdech i zaczyna wydychać powietrze lub na odwrót. Momenty przejścia z wdechu na wydech i w drugą stronę zapisywane są w pliku .csv, a dźwięk w pliku .wav. Częstotliwość próbkowania ustalamy na 44,1 kHz.



Rysunek 1: Przykładowe nagrane dane, in-wdech, out-wydech. Jak widać dane oznaczone na żywo są bardzo dokładne, być może nie bylibyśmy w stanie osiągnąć takiej dokładności oznaczając ręcznie (lub byłoby to bardzo żmudne).

4 Przyjęty model oddechu

Na początku przyjmujemy model "sportowego"oddychania, czyli wdech nosem i wydech ustami. Uproszczenie to polega na tym, że wydawane dźwięki są dosyć różne, dopiero potem sprawdzamy jak nasze podejście będzie się sprawować przy innych metodach oddychania.



Rysunek 2: Ilustracja przyjętego modelu, źródło

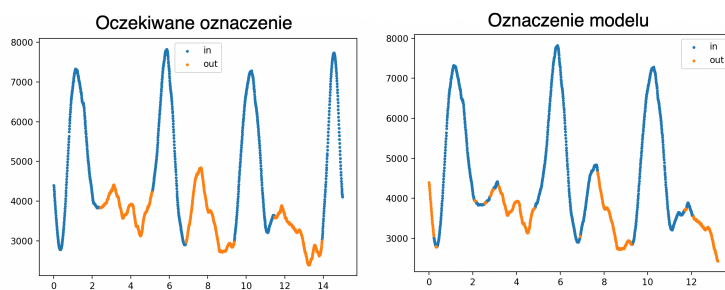
5 Średnia częstotliwość w czasie

5.1 Metoda

Pierwotnie przyjętym założeniem było, że podczas wdechu średnia częstotliwość dźwięku jest wyższa niż gdy osoba wydycha. Z pliku w formacie .wav dla bloków próbek pobieramy natężenie, które następnie odsumiamy za pomocą *noisereduce* (przy testach jakościowych stosujemy bramkę szumów dla 3 ostatnich sekund). Dzielimy próbki na bloki, dla których tworzymy spektrogramy i obliczamy średnią ważoną częstotliwość. Cechami, na podstawie których dokonywana jest predykcja są

$$\mathbf{x}(t_n) = [\bar{f}(t_0), \bar{f}(t_1), \dots, \bar{f}(t_n)]$$

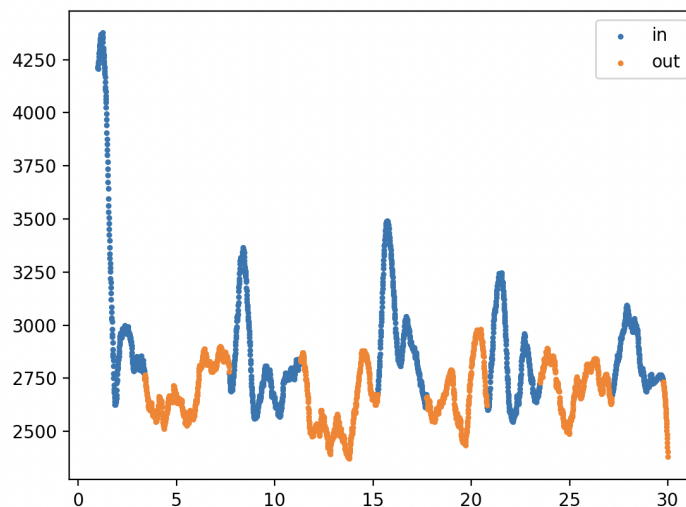
czyli średnie częstotliwości z kilku przeszłych chwil. Dla danych testowych spełniających powyższe założenie i modelu wytrenowanego SVM-em dawało nam to dokładność $\sim 60\%$.



Rysunek 3: Przykładowe działanie modelu dla danych spełniających założenie: wyższy wdech, niższy wydech. W pewnych momentach podobnie wyglądające fragmenty krzywej powinny być wdechem, w innych wydechem, mamy więc możliwy underfitting.

5.2 Problemy

Podejście to jednak mocno ogranicza nasze dalsze pole do rozwoju. Zmniejszenie całego spektrogramu do pojedynczej wartości częstotliwości daje nam mniej informacji, przez co być może ograniczylibyśmy się tylko do przyjętego przez nas modelu (tj. wdech - nos, wydech - usta) - w innych modelach różnica średnich częstotliwości między wdechem a wydechem może nie być taka wyraźna. Ponadto, dla niektórych osób zaobserwowaliśmy, że zależność między wdechem a wydechem jest niekoniecznie tak prosta jak założyliśmy. Obserwowana metoda była również bardzo niestabilna jeżeli chodzi o testy jakościowe (klasyfikację na żywo).



Rysunek 4: Nieoczywista zależność między wdechem a wydechem.

6 Dane wejściowe ze spektrogramu + SVM

6.1 Metoda

Kolejnym przyjętym przez nas podejściem było wzięcie całego spektrogramu (a przynajmniej jego części) jako dane wejściowe do SVM-a. Metoda pochodzi od przypuszczenia, że człowiek rozpoznaje i rozróżnia wdech/wydech na podstawie barwy dźwięku. Podobnie jak ostatnio pierw odsumiamy sygnał z pliku za pomocą *noisereduce*. **Dzielimy dane na bloki rozmiaru 1024 próbek, tworzymy dla każdego bloku spektrogram** ale tym razem nie liczymy średniej tylko zostawiamy całą taką klatkę. Wektor cech ma więc postać

$$\mathbf{x} = [I_{f_1}, I_{f_2}, \dots, I_{f_n}]$$

Nie są nam potrzebne wszystkie częstotliwości zwracane przez algorytm DFT, więc stosujemy ograniczenie podobne do tego stosowanego w telefonii komórkowej - **używamy 160 częstotliwości z przedziału do 6,9 kHz**. Stosując to podejście dla danych testowych otrzymaliśmy $\sim 70\%$ dokładności. W testach jakościowych jednak zaobserwować było można "przeskakiwanie" z jednej wartości na drugą i spowrotem np. w połowie brania wdechu, wydychania.

6.2 Poprzedni stan

W celu pozbycia się "przeskakiwania", dane z poprzedniego podpunktu rozszerzyliśmy o informację o **wcześniej przewidzianym stanie**, czyli

$$\mathbf{x} = [I_{f_1}, I_{f_2}, \dots, I_{f_{160}}, s]$$

gdzie $s = 1$ jeśli poprzednio stwierdzono wdech, $s = -1$ jeśli poprzednio stwierdzono wydech. Dodatkowo przeprowadziliśmy jeszcze **wygładzenie spektrogramu filtrem Savitzky-Golay**. Takim podejściem otrzymaliśmy dokładność $\sim 99\%$ dla danych testowych jednak wymagało to od nas modyfikacji naszych testów - jako wartość s stanu poprzedniego stosowaliśmy tą wziętą z danych testowych, czyli zakładaliśmy za każdym razem, że nasz model przewidzi dobrze poprzedni stan. Przypadek ten nie jest realny, ponieważ nie będzie znał poprawnej poprzedniej klasyfikacji tylko zakładał, że to co sam stwierdził jest poprawne. Biorąc poprzedni stan nie z danych testowych a z poprzedniej klasyfikacji modelu otrzymujemy mniejszą dokładność, bo $\sim 80\%$. Wniosek z tego jest taki, że model z wysokim prawdopodobieństwem przewidzi dobrze kolejny stan, jeżeli dobrze przewidział poprzedni. Warto dodać, że rzeczywiście metoda z poprzednim stanem pozwoliła nam ograniczyć "przeskakiwanie" w testach jakościowych.

6.3 Problemy

W poprzednim podpunkcie wyszedł główny mankament tej metody - konieczność założenia, że poprzednio przewidziany stan jest dobry co nie zawsze jest prawdą. W testach jakościowych zaobserwować można było "wariowanie" modelu, czyli sekwencję źle przewidzianych stanów.

- 7 Spektrogram i sieć neuronowa
- 8 Przejście na inny model oddychania
- 9 Douczenie
- 10 Wnioski
- 11 Implementacja modeli
 - 11.1 SVM ze spektrogramem