KONWERTER AUDIO NA MIDI

Arkadiusz Zawalniak

Dane:

Wejściem programu jest nagranie muzyczne w formacie WAV lub MP3. Takie nagrania mogą być pozyskiwane z różnych źródeł, zarówno z internetu, jak i poprzez indywidualne nagrywanie, na przykład na instrumencie. Jedynym kryterium dotyczącym nagrania jest jego charakter monofoniczny, co oznacza, że w jednym czasie występuje tylko jedna wysokość dźwięku. Do trenowania modelu nagrania dźwięków zostały wygenerowane w programie Cubase. Jest to nagranie zawierające wszystkie dźwięki z oktaw 0-7 (4 x 12 x 7 = 336 osobnych dźwięków).

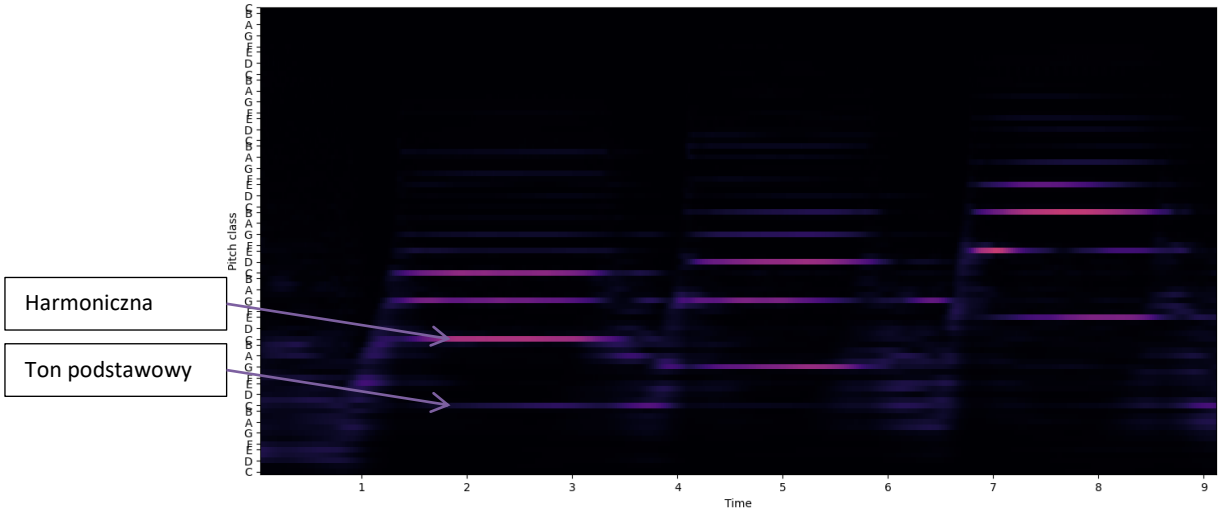
Pobrany plik audio jest przetwarzany poprzez konwersję za pomocą funkcji takiej jak np. `librosa.load()`. W wyniku tej operacji, wczytane nagranie przyjmuje postać wektora o długości, która wynosi iloczyn wartości sample_rate (reprezentującej ilość próbek na sekundę) i czasu trwania nagrania w sekundach.

Należy zaznaczyć, że algorytm zakłada jednoczesne występowanie tylko jednej wysokości dźwięku w czasie rzeczywistym, co jest istotnym założeniem dla poprawnego funkcjonowania metody.

Algorytm programu:

W pierwszym etapie nagranie w formacie WAV lub MP3 jest dzielone na fragmenty zawierające jedną nutę. Można tego dokonać używając fukncji onset.onset_detect z biblioteki Librosa. Librosa.onset.onset_detect lokalizuje zdarzenia rozpoczynające nutę, wybierając szczyty w obwiedni mocy sygnału. (1) Dalej odcinki utworu poddawane są transformacie CQT (wartości CQT są w przedziale <0,1>). Wartości CQT są akulumulowane, używając funkcji np.mean(). W ten sposób uzyskujemy cechę danego dźwięku w postaci wektora o długości 85 (C0-C7). Następnie dane zostały podzielone na dane treningowe, walidacyjne i dane testowe.

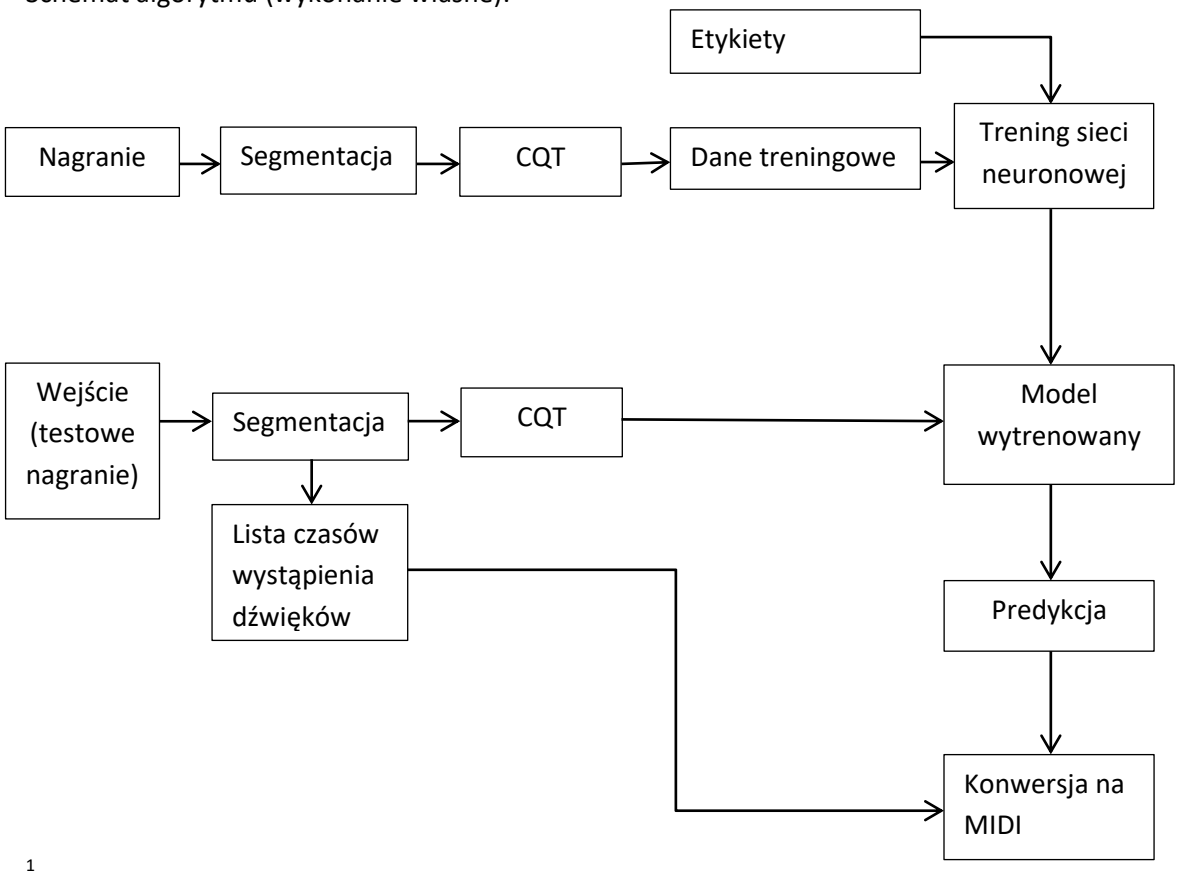
Tak przygotowane dane można wykorzystać do trenowania konwolucyjnych sieci neuronowych lub RNN. Do rozpoznawania wysokości dźwięku można wykorzystać cechę dźwięku jako chromagram w skali jednej oktawy. A do rozpoznania oktawy można wykorzystać całą skalę. W ten sposób program nie będzie musiał przetwarzać całej skali (88 dźwięków) by określić wysokość dźwięku. Wysokość dźwięku można określić jako numer_nuty_w_oktawie*numer_oktawy. Zapobiega to też problemowi złej oktawy podczas wykrywania dźwięku.



Rysunek 1 Spektogram dla dźwięków C1, G1 i E2 (wykonanie własne)

Jak widać harmoniczna ma większą wartość od tonu podstawowego i może być wzięta pod uwagę jako szukany dźwięk.

Schemat algorytmu (wykonanie własne):



[1]

[1] W opisywanej metodzie są używane dwa wytrenowane modele. Schemat dla drugiego modelu wygląda tak samo tylko rózni się wymiarem wektora cech CQT – 12 zamiast 85.

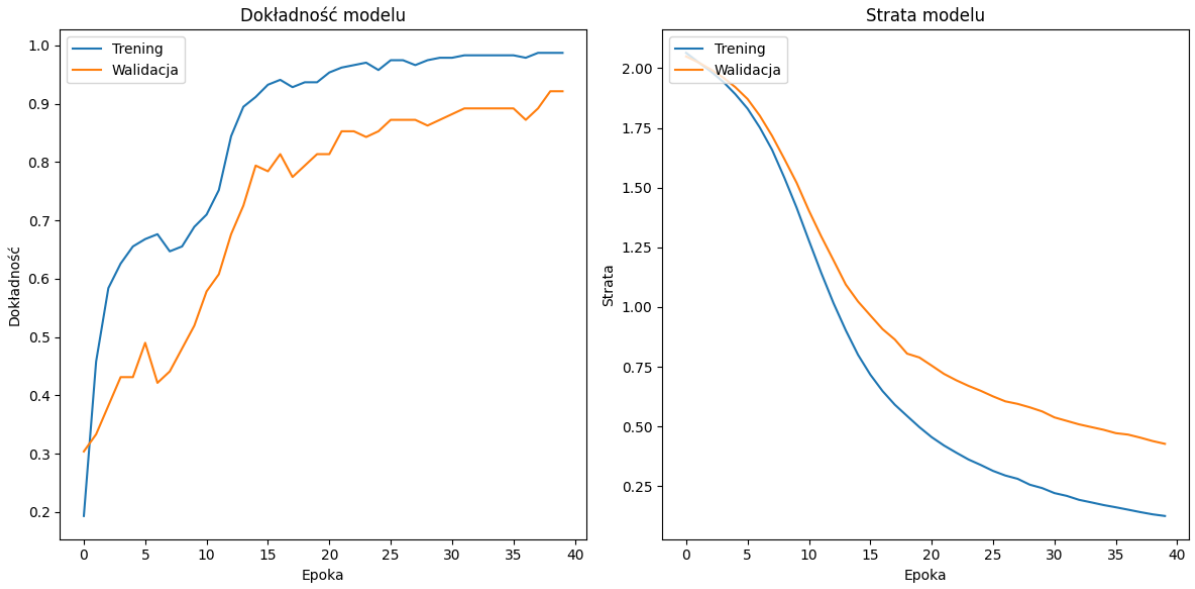
Budowa sieci:

Pierwszy model do wykrywania oktawy składa się z warstwy RNN o wymiarach wejścia 1x85, Dense z aktywacją relu oraz Dense jako wyjście o wymiarze 8 (liczba oktaw) z aktywacją softmax.

Drugi model do wykrywania dźwięku jest taki sam jak poprzedni model. Wymiary wejścia 1x12 oraz wyjścia to 12 (1x12 liczba dźwięków w oktawie).

Trening sieci:

W celu klasyfikacji oktawy została użyta funkcja straty kategoryzującej entropii krzyżowej. Learning rate został ustawiony na poziomie 0.001. Liczba epok wynosi 40.



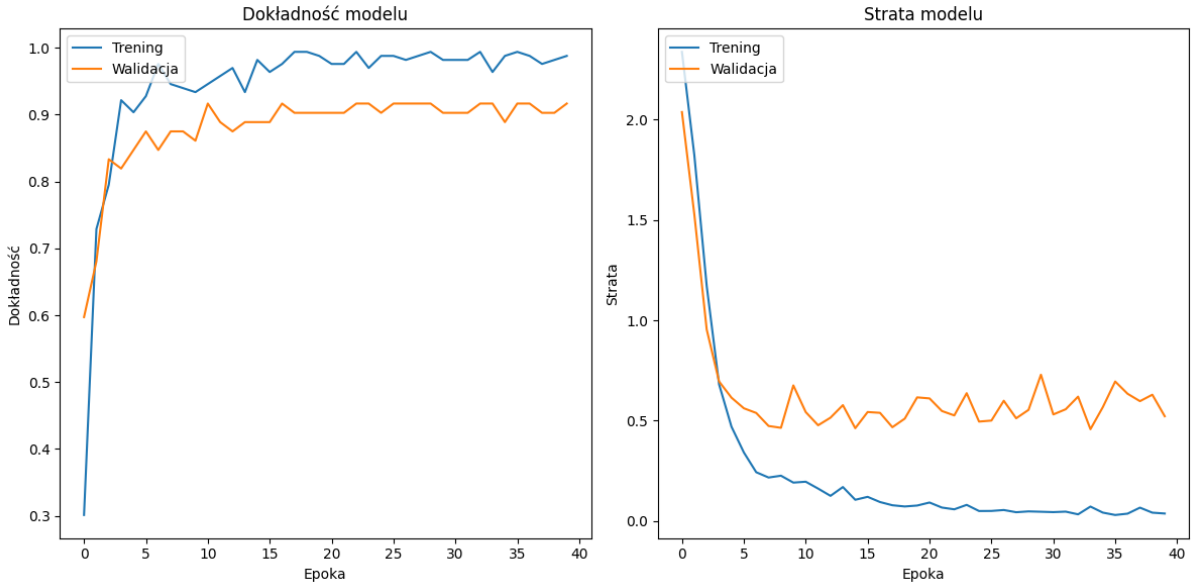
Rysunek 2 Przebieg treningu modelu nr 1

Dokładność: 0.9019607843137255

Precyzja: 0.9158565386939089

Czułość: 0.9019607843137255

W celu klasyfikacji wysokości dźwięku w oktawie została użyta funkcja straty kategoryzującej entropii krzyżowej. Learning rate został ustawiony na poziomie 0.01. Liczba epok wynosi 40.



Rysunek 3 Przebieg treningu modelu nr 2

Liczba epok	40	60	20
Dokładność	0.9705	0.9705	0.9513
Precyzja	0.9739	0.9741	0.9549
Czułość	0.9705	0.9705	0.9513

Dla 60 epok występuje overfitting.

Podobny algorytm został dokładnie opisany przez (2)

Film prezentujący działanie programu: https://youtu.be/2y9CeWTdmJc?si=qxInu8_6JWVc27Tn

Bibliografia

1. **librosa development team.** librosa.onset.onset_detect. [Online] https://librosa.org/doc/latest/generated/librosa.onset.onset_detect.html.

2. **Meng, Zhihang and Chen, Wencheng.** [Online] 2020. https://iopscience.iop.org/article/10.1088/1742-6596/1651/1/012192/pdf.