

Rozpoznawanie nut

Wojciech Piszczyk

145401

Wojciech Spychalski

145392

21 listopada 2021

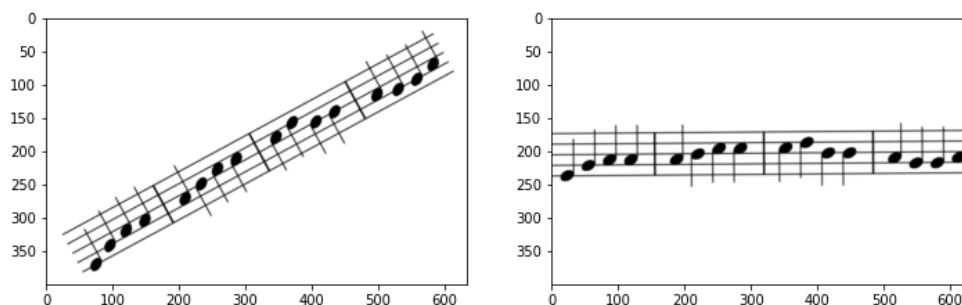
Wstęp

Zadaniem programu jest przetworzenie zdjęcia, na którym znajduje się pięciolinia i rozszyfrować nuty. Program wczytuje obrazek, znajduje na nim pięciolinie, a następnie na jej podstawie identyfikuje i odtwarza nuty.

Szczegółowy opis działania programu

Obracanie obrazu

Na początku program próbuje obrócić obraz. Najpierw wyznacza kąt obrotu obrazu za pomocą metody Hougha, a następnie obraca w przeciwnym kierunku, aby doprowadzić do wyprostowania zdjęcia. Do wyznaczenia kąta została użyta biblioteka `deskew`.

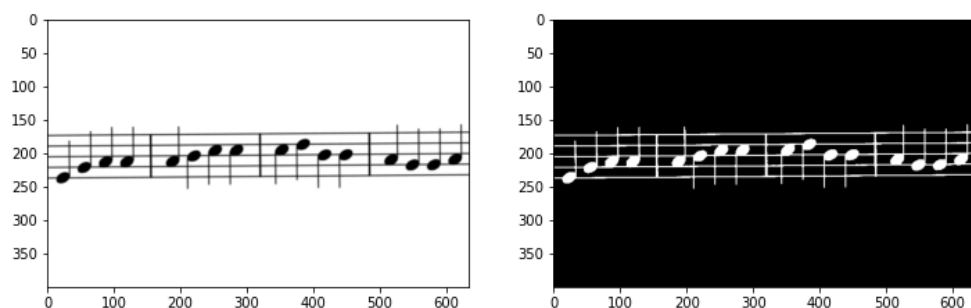


Rysunek 1: Obrócenie obrazu

Jeżeli obraz nie wymaga obrócenia, program po prostu przechodzi dalej.

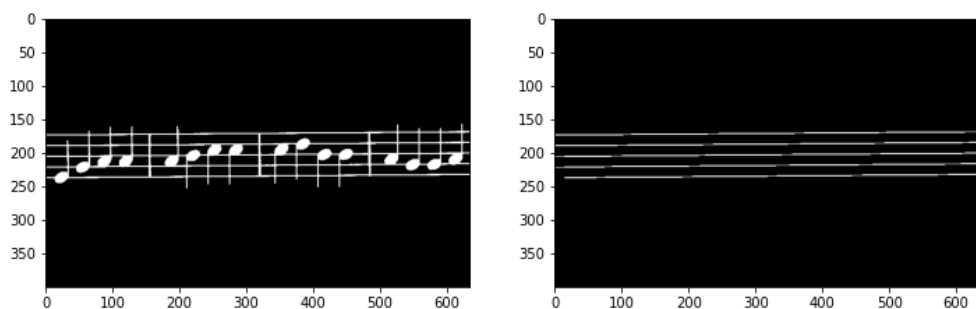
Odszukiwanie pięciolinii

Na samym początku odszukiwania pięciolinii odfiltrowujemy tło oraz odwracamy kolory obrazu przy pomocy metody Otsu.



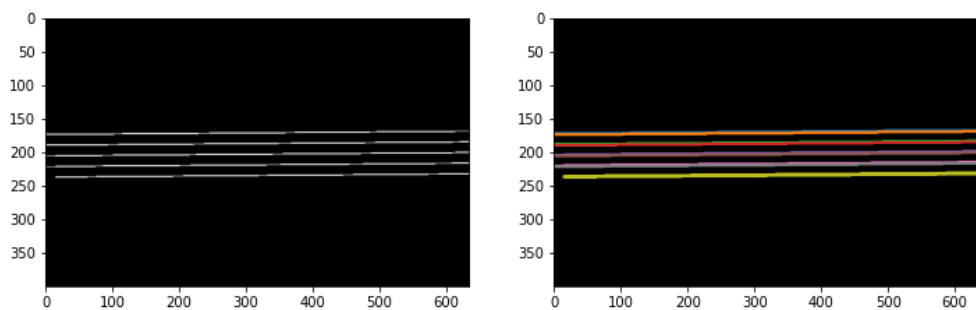
Rysunek 2: Odwrócenie kolorów obrazu

Następnie przy pomocy dość obszernej erozji poziomej usuwamy wszystkie linie pionowe oraz nuty, aby pozostały tylko linie pięciolinii.



Rysunek 3: Usunięcie nut i linii pionowych obrazu

Kolejno odszukujemy kontury każdej z nich, a następnie zapisujemy ich średnie wysokości.

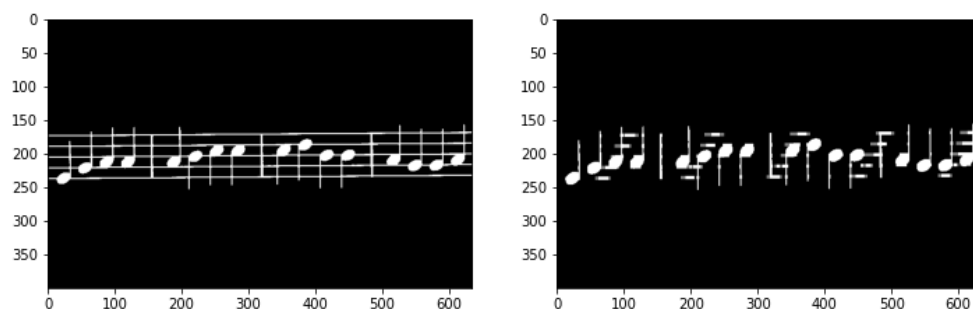


Rysunek 4: Kontury linii

Następnie obliczamy średnią odległość między liniami - przyda się ona na kolejnych etapach.

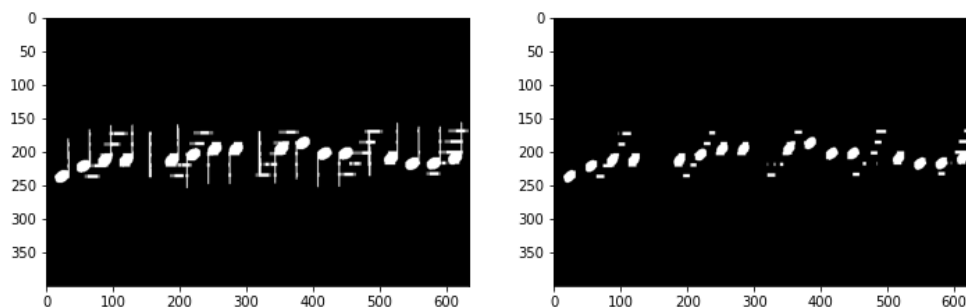
Odszukiwanie nut

Podobnie jak na etapie odszukiwania pięciolinii odfiltrowujemy tło oraz odwracamy kolory obrazu przy pomocy metody Otsu. Następnie korzystamy tym razem z erozji pionowej usuwając wszystkie linie poziome - czyli pięciolinie i dylatacji pionowej, która odbuduje kształt naszych nut.



Rysunek 5: Po erozji i dylatacji pionowej

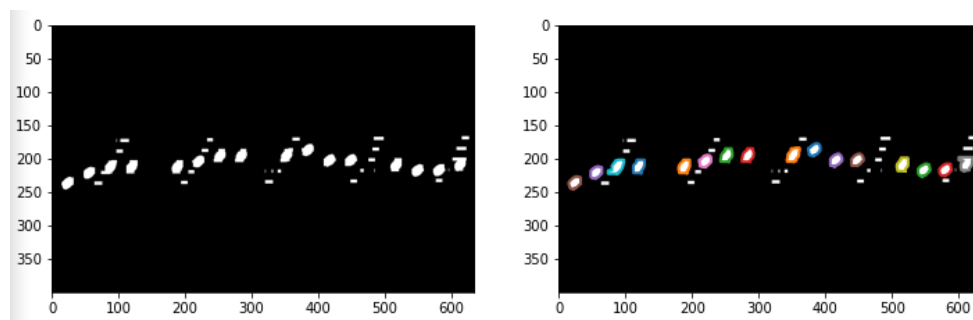
Ostatnią transformacją jest erozja pozioma, która usuwa “laski” nut zostawiając same “koło” reprezentujące konkretny dźwięk.



Rysunek 6: Po erozji poziomej

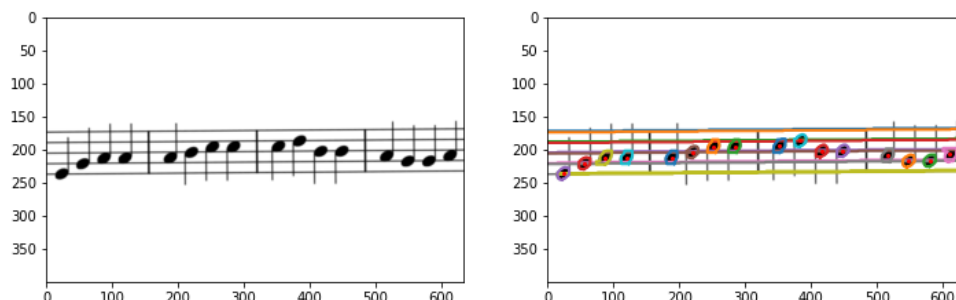
Na tym etapie przyda nam się średnia odległość między liniami, policzona podczas Odszukiwania pięciolinii. Odszukujemy kontury każdego pozostałego obiektu i liczymy jego pole powierzchni. Jeśli jest mniejsze niż

pole koła o promieniu podzielonej przez 2 średniej odległości między liniami to zaliczamy ten obiekt jako szum i pomijamy. W przeciwnym razie obiekt klasyfikujemy jako nutę, obliczamy jej środek i zapisujemy jego wysokość - przyda nam się na etapie identyfikacji dźwięku.



Rysunek 7: Kontury nut (po odfiltrowaniu)

Identyfikacja dźwięku



Rysunek 8: Kontury nut i pięciolinie oraz centroidy nut

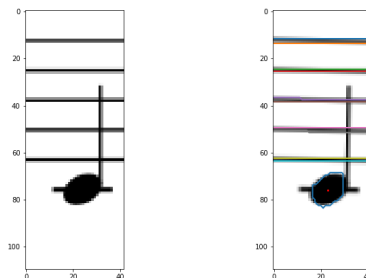
Identyfikacja dźwięku w naszym projekcie polega na klasyfikowaniu nut przy pomocy wysokości pięciolinii i wysokości wykrytych nut. Na samym początku kolekcjonujemy przydatne stałe do klasyfikacji metryki: średnią odległość między liniami - ją już uzyskaliśmy na etapie odszukiwania pięciolinii i maksymalne odchylenie od średniej wysokości dźwięku - obliczamy tą wartość dzieląc średnią odległość między liniami przez 4. Następnie kolejno dla każdej nuty klasyfikujemy ją sprawdzając czy jej wysokość mieści się w zakresie wysokości danego dźwięku.

Odtwarzanie dźwięku

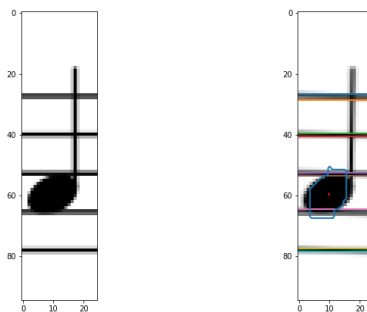
Odtwarzanie dźwięku odbywa się poprzez odtwarzanie konkretnych plików MP3 symbolizujących dane dźwięki. Rytmiczne odgrywanie melodii polega na tworzeniu wątków ze stałym odstępem czasowym uruchamiających dźwięk pojedynczej nuty.

Wyniki działania programu

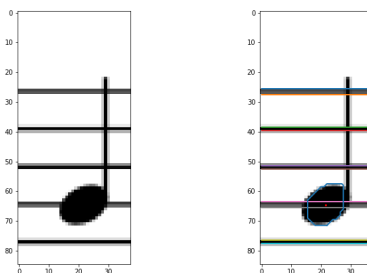
Zbiór łatwy



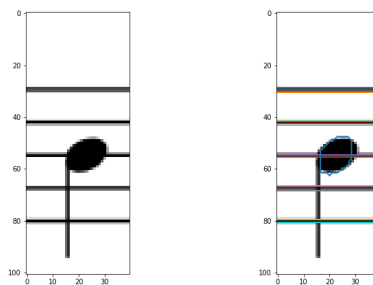
Oczekiwany output	c4
Output	c4



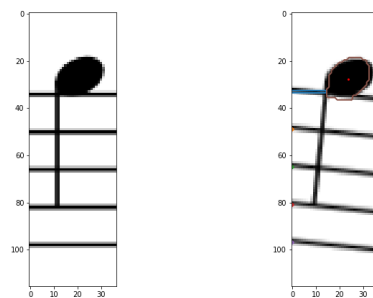
Oczekiwany output	a5
Output	a5



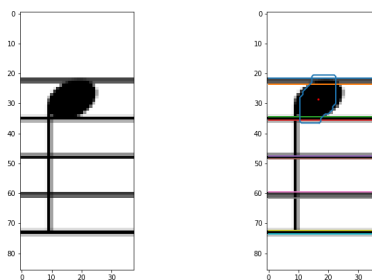
Oczekiwany output	g4
Output	g4



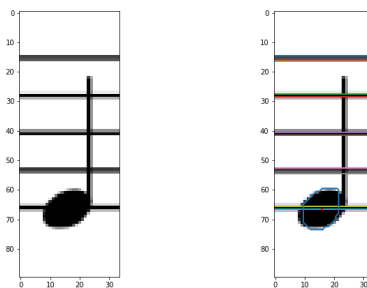
Oczekiwany output	b5
Output	b5



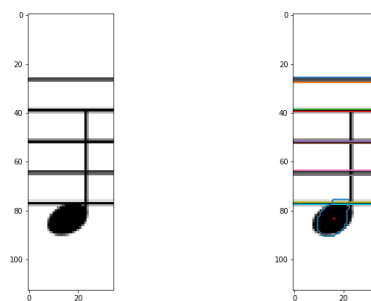
Oczekiwany output	g5
Output	g5



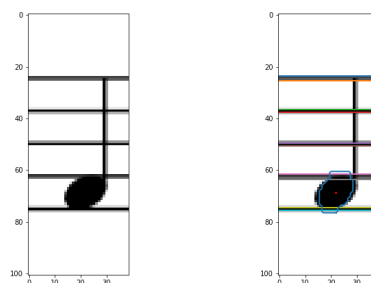
Oczekiwany output	e5
Output	e5



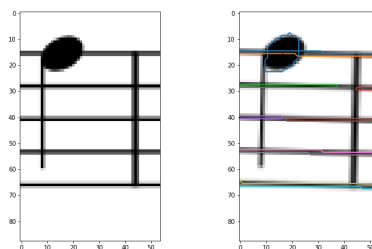
Oczekiwany output	e4
Output	e4



Oczekiwany output	d4
Output	d4



Oczekiwany output	f4
Output	f4

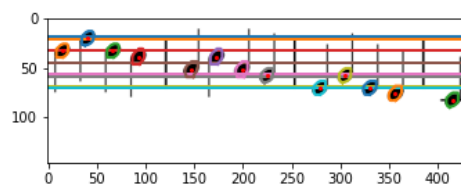
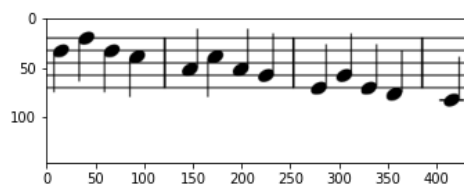


Oczekiwany output	f5
Output	f5

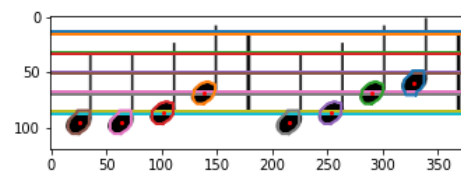
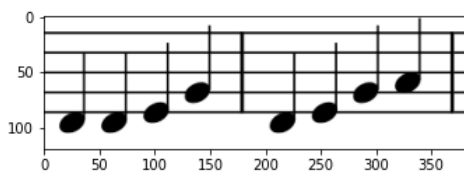
Statystyki

Poprawnie odczytane obrazy	10/10
Liczba poprawnie odczytanych nut	10/10

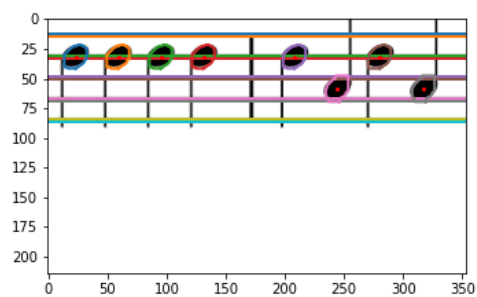
Zbiór średni



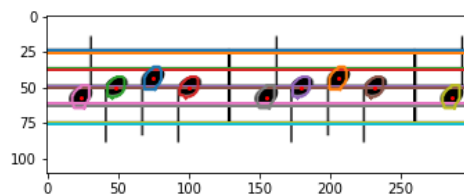
Oczekiwany output	d5 f5 d5 c5 a5 c5 a5 g4 e4 g4 e4 d4 c4
Output	d5 f5 d5 c5 a5 c5 a5 g4 e4 g4 e4 d4 c4



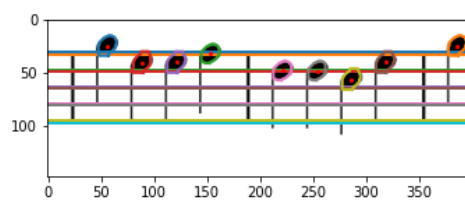
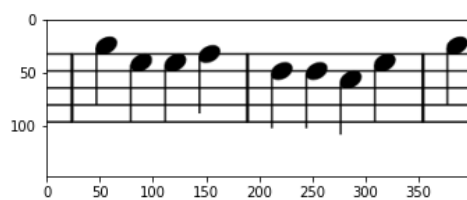
Oczekiwany output	d4 d4 e4 g4 d4 e4 g4 a5
Output	d4 d4 e4 g4 d4 e4 g4 a5



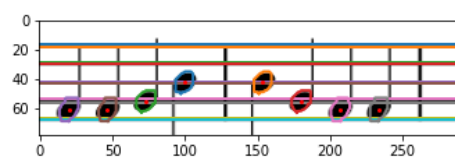
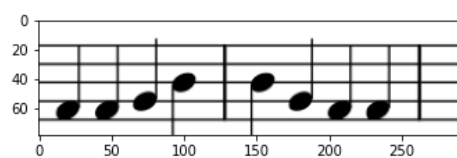
Oczekiwany output	d5 d5 d5 d5 d5 a5 d5 a5
Output	d5 d5 d5 d5 d5 a5 d5 a5



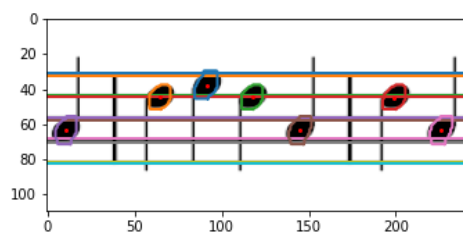
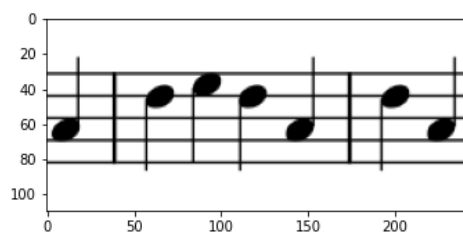
Oczekiwany output	a5 b5 c5 b5 a5 b5 c5 b5 a5
Output	a5 b5 c5 b5 a5 b5 c5 b5 a5



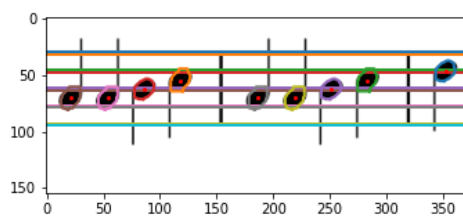
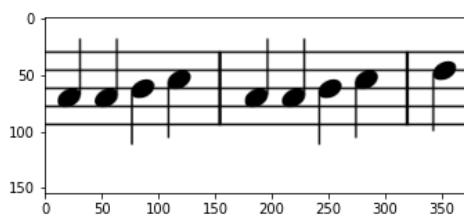
Oczekiwany output	g5 e5 e5 f5 d5 d5 c5 e5 g5
Output	g5 e5 e5 f5 d5 d5 c5 e5 g5



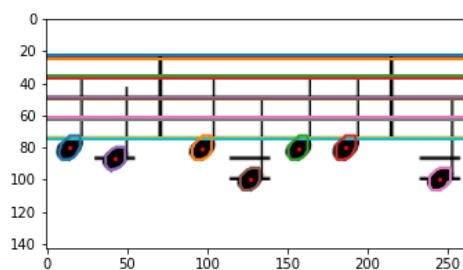
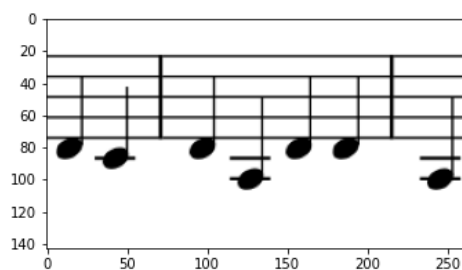
Oczekiwany output	f4 f4 g4 b5 b5 g4 f4 f4
Output	f4 f4 g4 b5 b5 g4 f4 f4



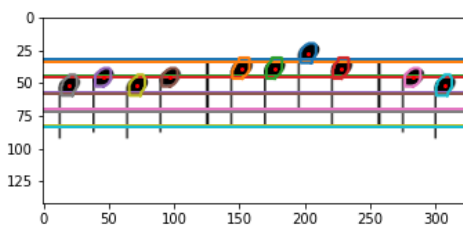
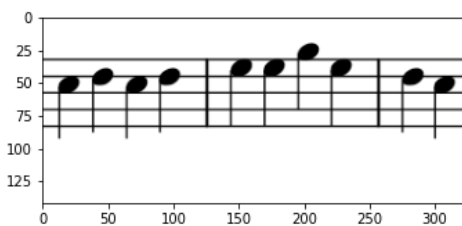
Oczekiwany output	a5 d5 e5 d5 a5 d5 a5
Output	a5 d5 e5 d5 a5 d5 a5



Oczekiwany output	a5 a5 b5 c5 a5 a5 b5 c5 d5
Output	a5 a5 b5 c5 a5 a5 b5 c5 d5



Oczekiwany output	d4 c4 d4 a4 d4 d4 a4
Output	d4 c4 d4 a4 d4 d4 a4

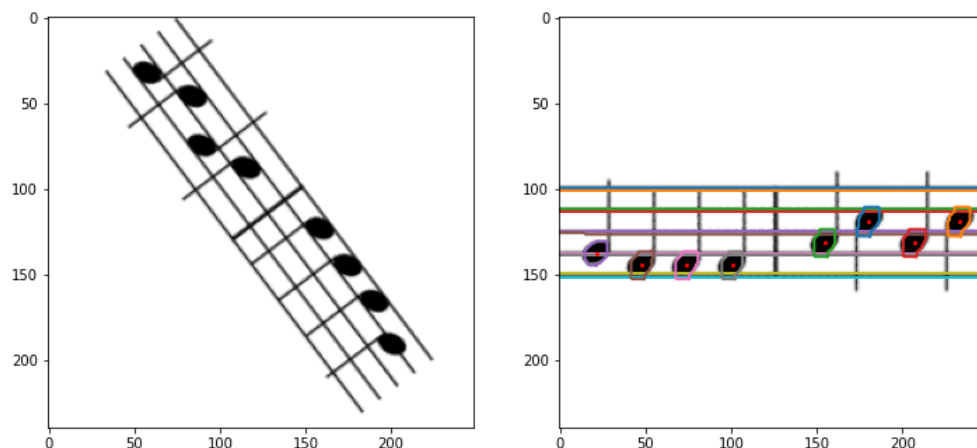


Oczekiwany output	c5 d5 c5 d5 e5 e5 g5 e5 d5 c5
Output	c5 d5 c5 d5 e5 e5 g5 e5 d5 c5

Statystyki

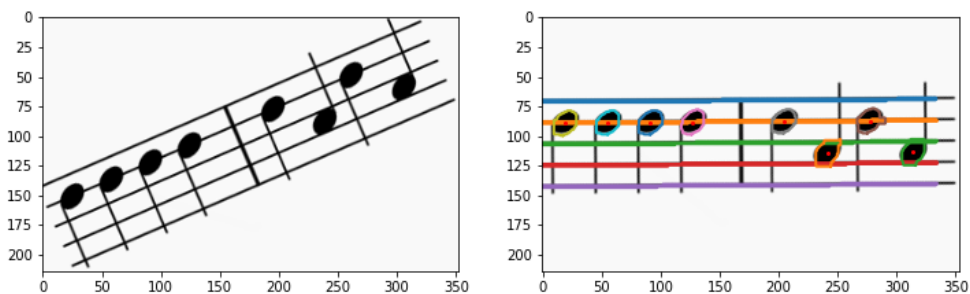
Poprawnie odczytane obrazy	10/10
Liczba poprawnie odczytanych nut	88/88

Zbiór trudny

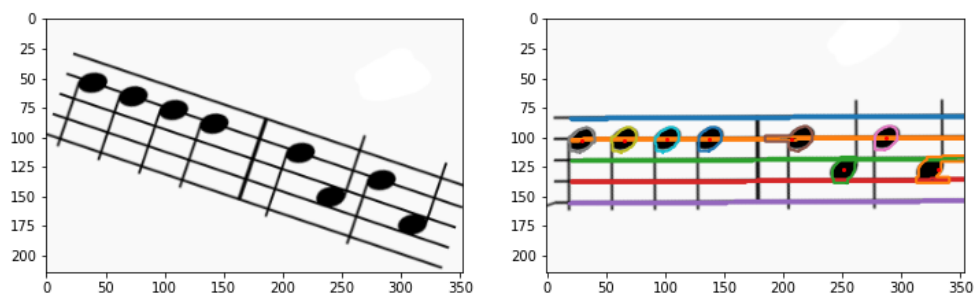


Oczekiwany output	a5 c5 a5 c5 e5 e5 e5 d5
Output	g4 f4 f4 f4 a5 c5 a5 c5

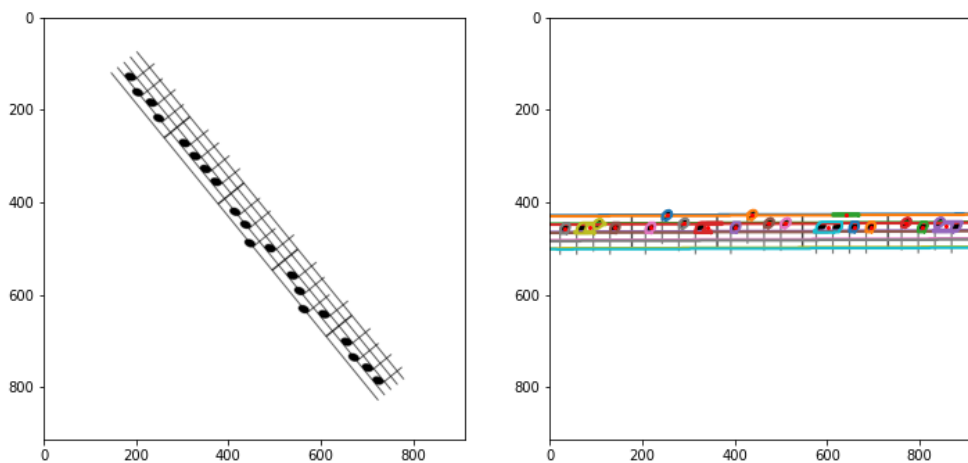
Algorytm wykrywania nut zadziałał poprawnie, ale obraz został obrócony w złą stronę.



Oczekiwany output	d5 d5 d5 d5 d5 a5 d5 a5
Output	d5 d5 d5 d5 d5 a5 d5 a5

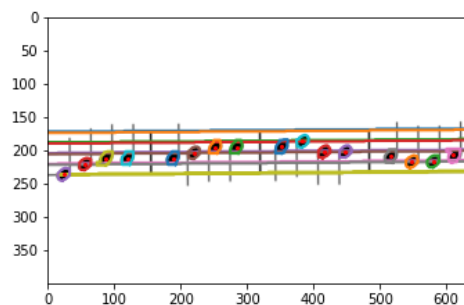
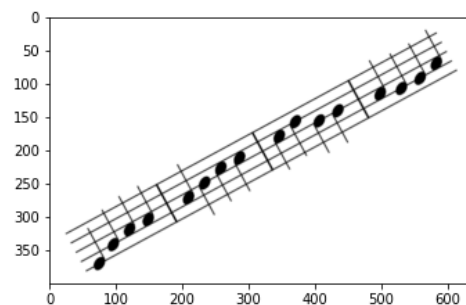


Oczekiwany output	d5 d5 d5 d5 d5 a5 d5 a5
Output	d5 d5 d5 d5 d5 a5 d5 a5

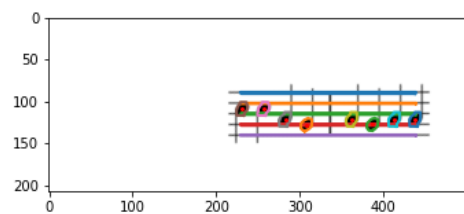
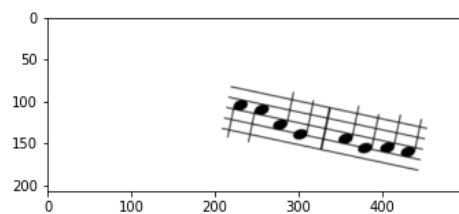


Oczekiwany output	a5 g4 a5 g4 a5 a5 a5 g4 g4 e4 a5 a5 g4 e4 a5 a5 g4 a5 a5
Output	c5 c5 c5 c5 f5 d5 c5 c5 f5 d5 d5 c5 f5 c5 c5 d5 c5 c5

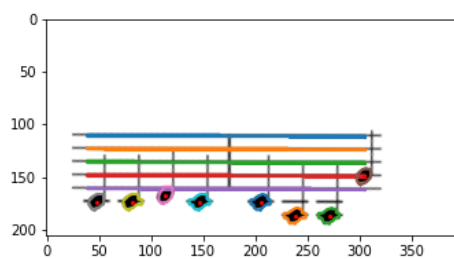
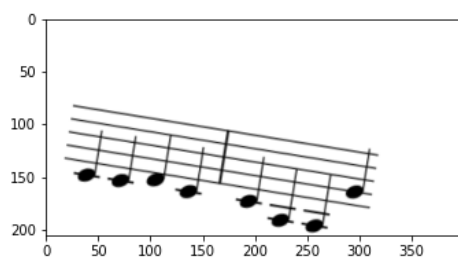
Obraz został obrócony w drugą stronę, ale algorytm i tak nie zadziałał - niektóre nuty nie zostały rozpoznane, czasem dwie nuty zostały rozpoznane jako jedna, raz też za nutę została uznana część pięciolinii. Prawdopodobnie zawiodło odfiltrowywanie zbędnych konturów.



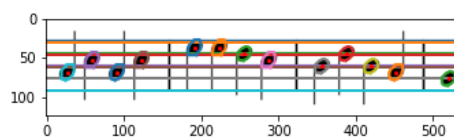
Oczekiwany output	e4 g4 a5 a5 a5 b5 c5 c5 c5 d5 b5 b5 a5 g4 g4 a5
Output	e4 g4 a5 a5 a5 b5 c5 c5 c5 d5 b5 b5 a5 g4 g4 a5



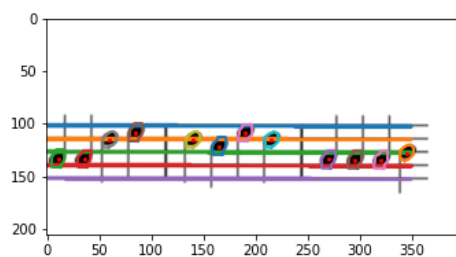
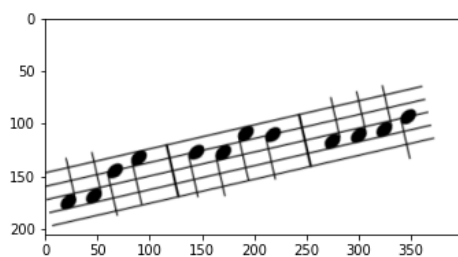
Oczekiwany output	c5 c5 a5 g4 a5 g4 a5 a5
Output	c5 c5 a5 g4 a5 g4 a5 a5



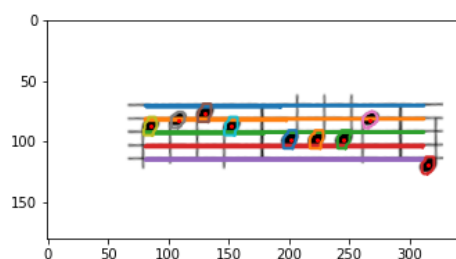
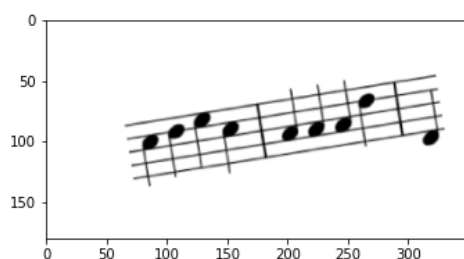
Oczekiwany output	c4 c4 d4 c4 c4 a4 a4 g4
Output	c4 c4 d4 c4 c4 a4 a4 g4



Oczekiwany output	a5 c5 a5 c5 e5 e5 d5 c5 b5 d5 b5 a5 g4
Output	a5 c5 a5 c5 e5 e5 d5 c5 b5 d5 b5 a5 g4



Oczekiwany output	a5 a5 d5 e5 d5 c5 e5 d5 a5 a5 a5 b5
Output	a5 a5 d5 e5 d5 c5 e5 d5 a5 a5 a5 b5



Oczekiwany output	c5 d5 e5 c5 a5 a5 a5 d5 d4
Output	c5 d5 e5 c5 a5 a5 a5 d5 d4

Statystyki

Poprawnie odczytane obrazy	8/10
Liczba poprawnie odczytanych nut	82/110

Wnioski

Generalnie cały algorytm wykrywania nut działa z zaskakująco wysoką skutecznością, jednak nie obyło się również bez błędów:

1. Obracanie pięciolinii w złą stronę - niestety na ten problem nie mamy żadnego wpływu.
2. Anomalie (pomijanie nut, klasyfikowanie pięciolini jako nuty, scalanie dwóch nut w jedną, itd.) - te błędy pojawiają się przy szczególnie zagęszczonych przez nuty obrazach, wynikają prawdopodobnie z nie-najlepiej dopasowanych wielkości macierzy przy transformacjach erozji i dylatacji. Przypuszczamy, iż dynamiczne dobieranie rozmiarów macierzy tych transformacji (rozmiar zależny od wielkości przetwarzanego obrazu) mogłoby zmniejszyć odsetek źle zakwalifikowanych nut.

Literatura

- [1] <https://github.com/sbrunner/deskew>
- [2] <https://pyshine.com/How-to-play-piano-using-Python/>
- [3] Sheet Music Reader - Sevy Harris, Prateek Verma