

# Komunikacja Człowiek-Komputer

## rozpoznawanie nut

Kuba Wąsik (132335), Dominik Szmyt (132326)

18.11.2018

## 1 Zastosowanie

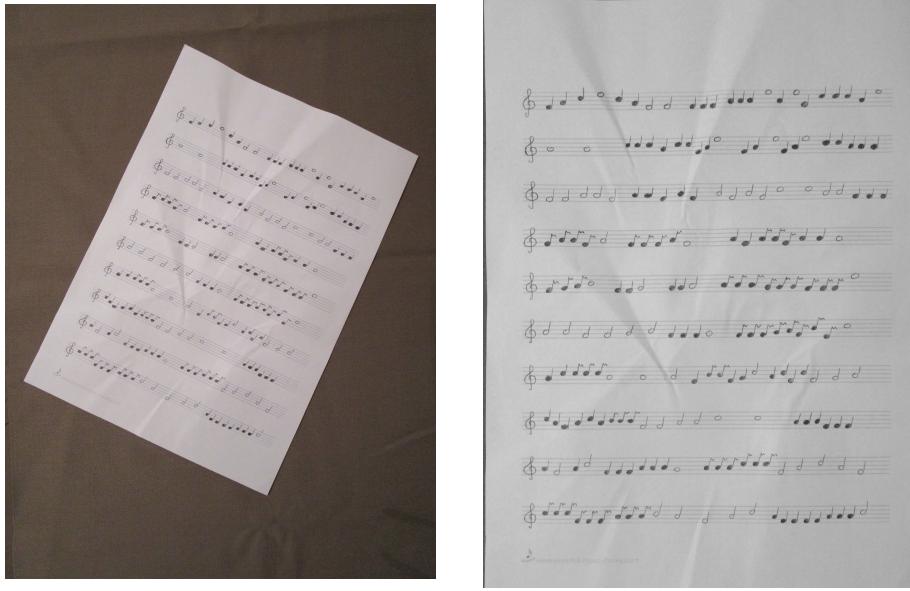
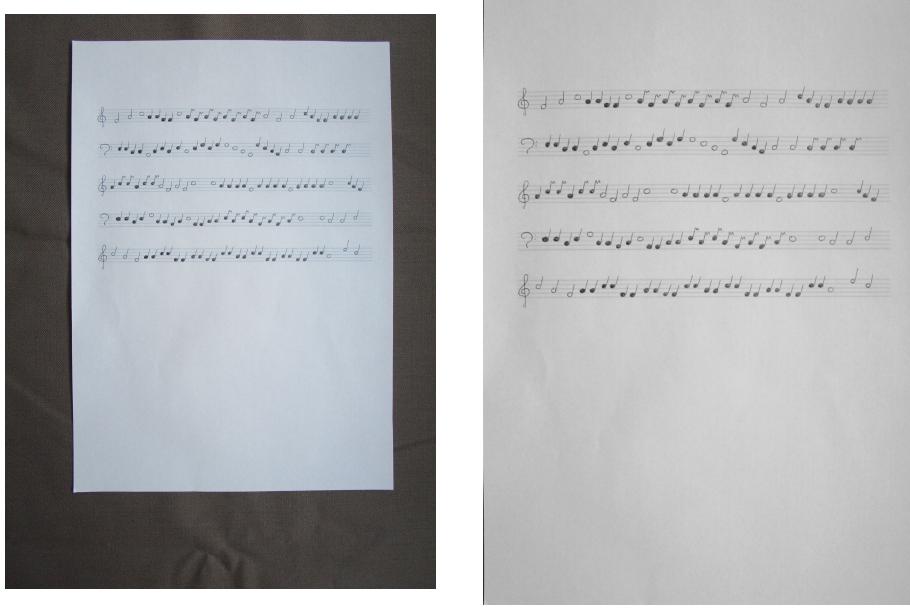
Program służy do wykrywania nut ze zdjęcia. Jest w stanie odnaleźć klucze (wiolinowy i basowy) oraz wypisać kolejność pojawienia się nuty, jej wysokość oraz numer pięciolini, w której się znajduje.

## 2 Przetwarzanie

Opiszemy najważniejsze kroki, jakie podjęliśmy, aby wykryć nuty.

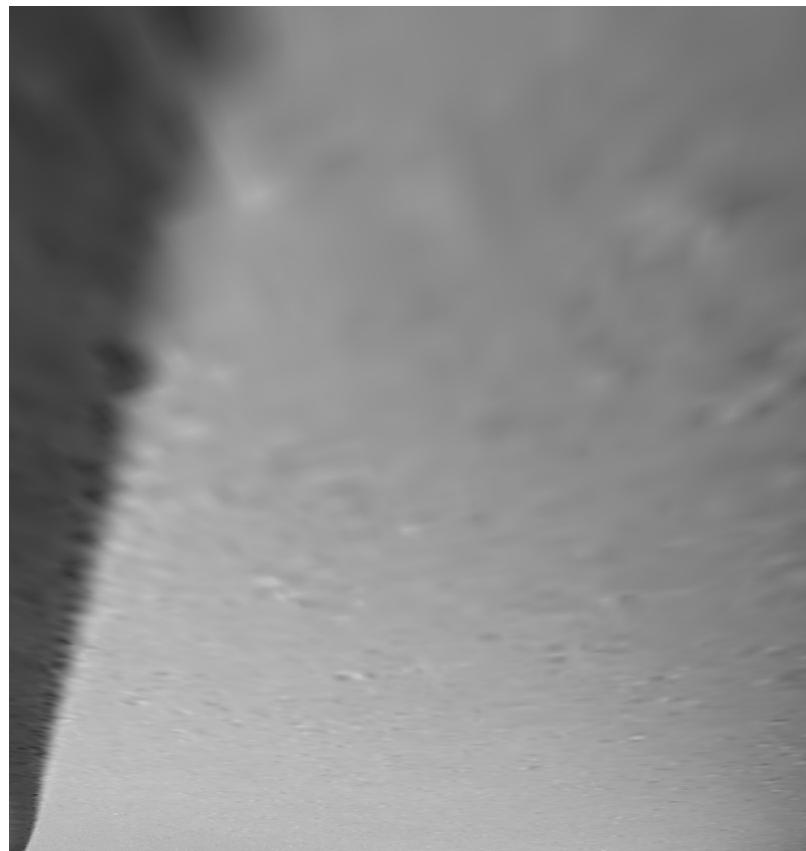
### 2.1 Wycinanie strony

Pierwszym krokiem, który należało wykonać było "wyłuskanie" ze zdjęcia samej kartki. W tym celu poszukiwaliśmy największego konturu na obrazie, który ma cztery narożniki, zakładając, że to nasza kartka. Aby było to możliwe, wcześniej nakładaliśmy filtr Canny. Kiedy już odnaleźliśmy nasze narożniki, po odpowiednim ich posortowaniu używaliśmy dla nich funkcji `cv2.getPerspectiveTransform()` oraz `cv2.warpPerspective()`, aby odpowiednio znaleźć macierz transformacji i zastosować ją na zdjęciu.



Rysunek 2: Obraz oryginalny i po wycięciu

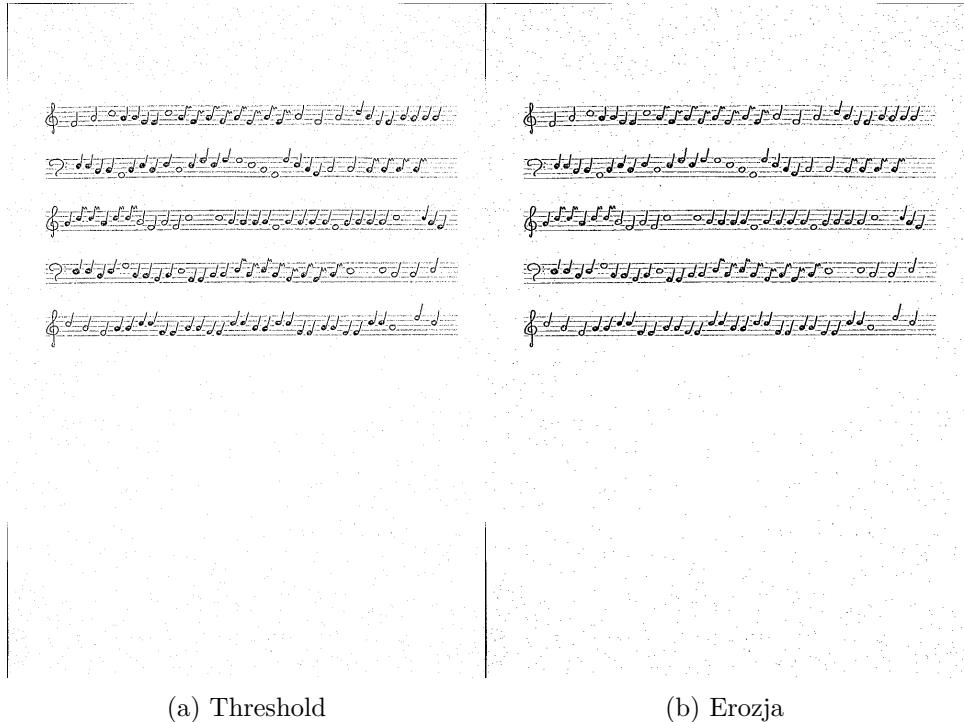
Nie dla wszystkich przypadków wykrywanie kartki zadziałało - dla nierównomiernego oświetlenia wykryte kontury kartki nie były zamknięte. Efekt takiego wycięcia wyglądał tak:

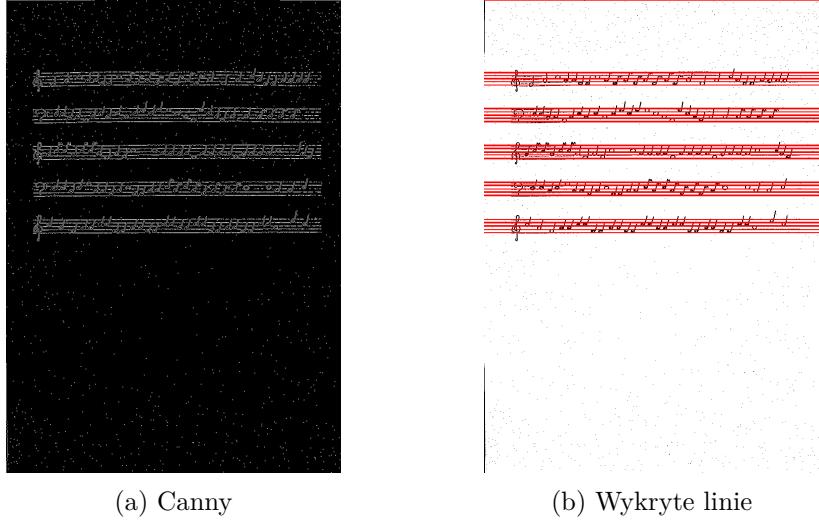


Rysunek 3: Nieudany przykład

## 2.2 Wykrywanie pięciolinii

Kolejnym zadaniem było wykrycie wysokości, na których znajdują się pięciolinie. Zastosowaliśmy do tego transformacje Hougha do wykrywania linii. Skorzystaliśmy z gotowej funkcji **cv2.HoughLines**. Jako jej parametry podaliśmy  $\rho = 1$  oraz  $\theta = \pi/100$  (ponieważ poszukujemy linii poziomych, kąt  $\theta$  będzie niewielki, tak samo jak długość  $\rho$ ). Wcześniej jednak trzeba było przetworzyć obraz za pomocą funkcji `threshold_local` oraz nakładając filtr Canny. Funkcja ta w zależności od jasności obrazu lepiej działała z parametrem 'method' ustawionym na 'median' albo 'mean'. Zbyt duży `threshold` usuwał nutki, zbyt mały pozostawiał szумy i utrudniał wykrywanie nut. Ostatecznie wybraliśmy nieco mniejszy i dodaliśmy erozję (`cv2.erode()`).

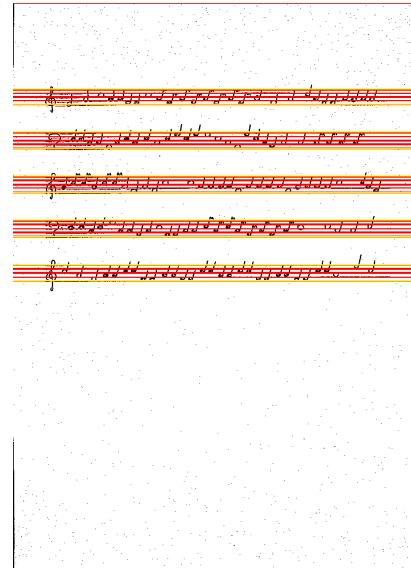




(a) Canny

(b) Wykryte linie

Po wykryciu linii za pomocą transformacji Hougha obliczamy współrzędne jej końców, a następnie centra. Sprawdzając odległość pomiędzy liniami, stwierdzamy, czy należą one do jednej pięciolinii, czy też do dwóch różnych. Jeśli znajdziemy pięć równoległych, leżących blisko siebie linii, zakładamy, że krańcowe z nich są granicami jednej z pięciolinii. Zwracamy obiekt Staff, który zawiera granice pięciolinii.

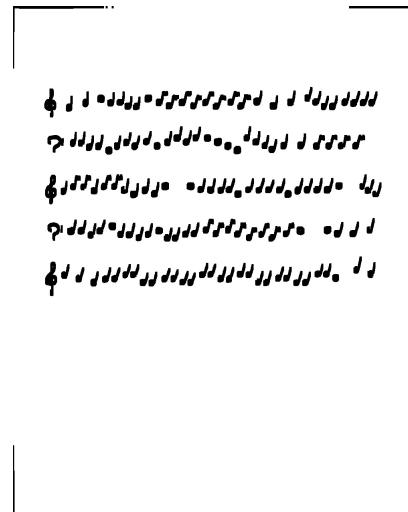


Rysunek 6: Wykryte pięciolinie

### 2.3 Usuwanie pięciolinii

Kolejnym krokiem jest usunięcie pięciolinii w celu łatwego wykrycia konturów nut. W tym celu stosujemy na obrazie threshold, a następnie obracamy go w negatyw. Do usunięcia pięciolinii używamy funkcji

**cv2.getStructuringElement** oraz **cv2.morphologyEx**, z których pierwsza tworzy element, który będzie wyszukiwany i usuwany (w naszym przypadku będzie to długi cieński prostokąt), a druga wykonuje wskazane przekształcenie morfologiczne (z parametrem **cv2.MORPH\_OPEN** wykonuje otwarcie - na zasadzie dylatacji) Kolejnym krokiem jest wykonanie dość silnej erozji, aby struktura nut była pełna. Następnie wyszukujemy kontury. W przypadku wyszukiwania nut okrajamy nieco kartkę (przypisuję barwę białą pewnym marginesom), aby uniknąć wyszukania tekstu na dole strony i kluczy wiolinowych/basowych. Następnie kontury sortujemy po obwodach. Wykrycie nut całych i ósemek jest trywialne (przedziały odpowiednio najmniejszych i największych obwodów). W przypadku ćwierćnuta i półnuta, które mają ten sam rozmiar, sprawdzam średnią wartość fragmentu obrazu wydzielonego przez kontur nuty (półnuta jest biała w środku - średnia wartość znacznie wyższa).



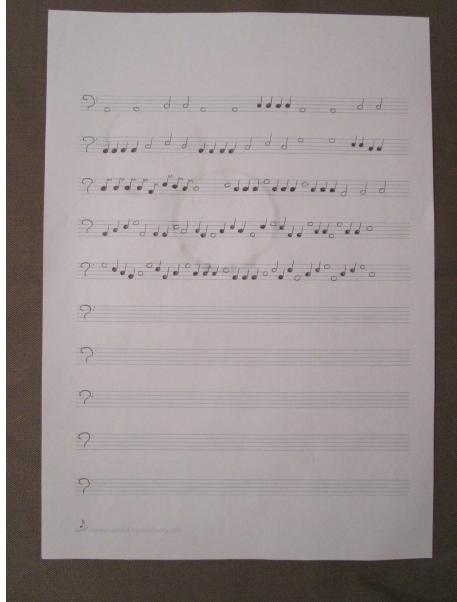
Rysunek 7: Usunięte pięciolinie + silna erozja

## 2.4 Wykrywanie kluczy

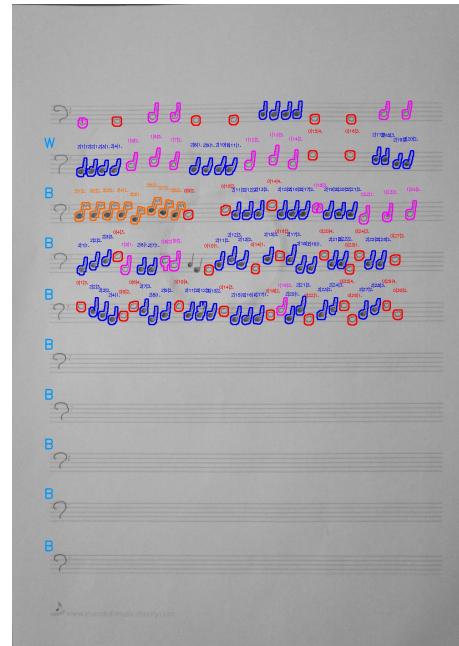
Po skorzystaniu z utworzonych już funkcji do przekształcenia obrazu, znajdowania położenia każdej pięciolinii oraz usuwania ich z obrazu, możemy rozpocząć samo znajdowanie kluczy. Przy szukaniu kluczy korzystamy z wiedzy, że muszą się one zawsze znajdować na początku pięciolinii. Wyznaczmy więc potencjalne miejsce klucza, proporcjonalnie do jego rozmiarów. Następnie, korzystając z różnic pomiędzy kluczem wiolinowym, a kluczem basowym, będziemy przeszukiwać obszar pod pięciolinią. Szukamy pierwszego wystąpienia czarnego piksela. Jeśli jakiś się pojawił to znaczy że znaleźliśmy klucz wiolinowy, w przeciwnym wypadku znaleźliśmy klucz basowy.



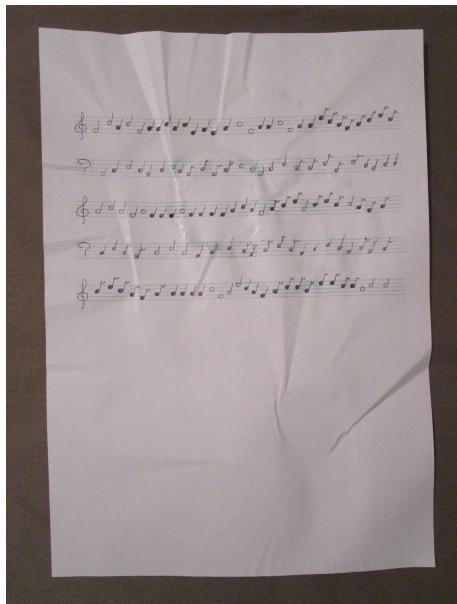
Rysunek 8: Obraz oryginalny nr 4 i i ze znalezionymi kluczami (łatwe)



(a) Oryginalny obraz nr 17



(b) i ze znalezionej kluczami (średnie)



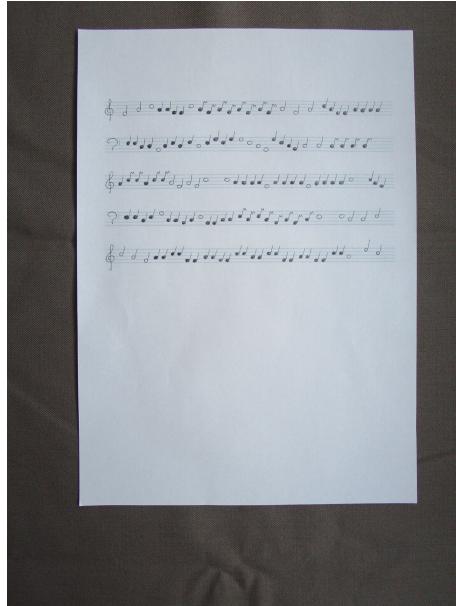
(c) Oryginalny obraz nr 30



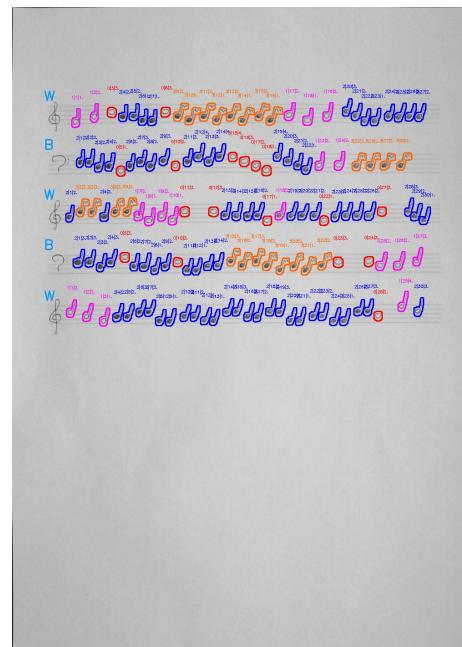
(d) i ze znalezionej kluczami (trudne)

### 3 Przykładowe wyniki

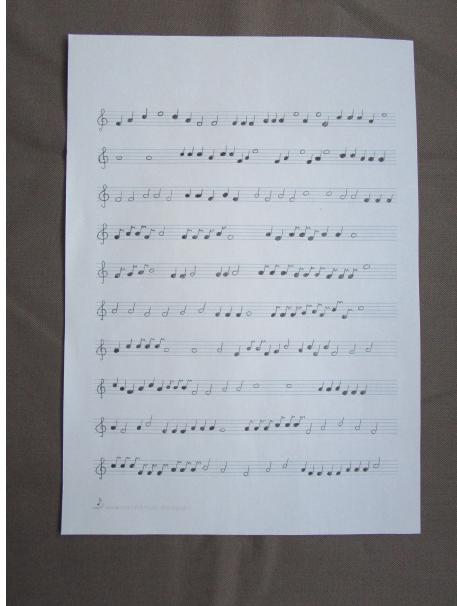
#### 3.1 Łatwe



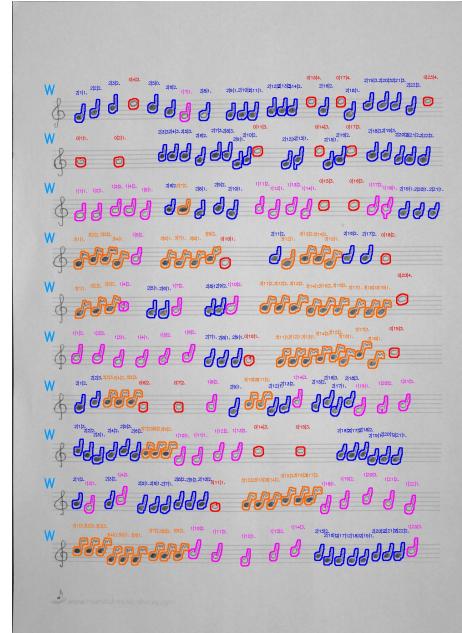
(a) próbka 4 przed procesem



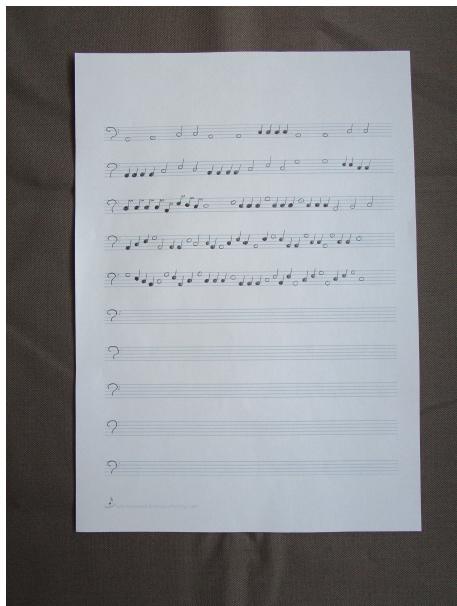
(b) próbka 4 wynik



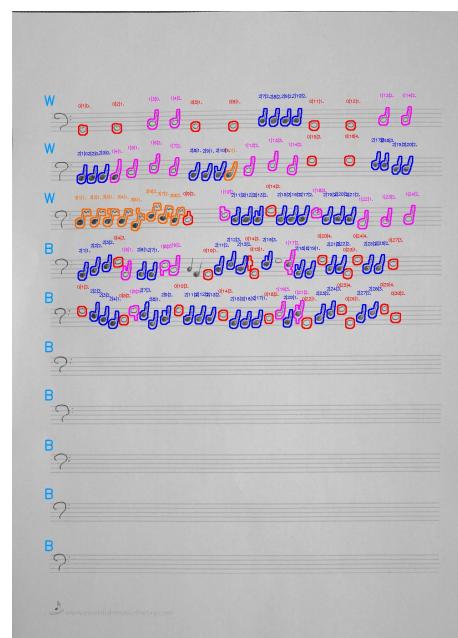
(a) próbka 5 przed procesem



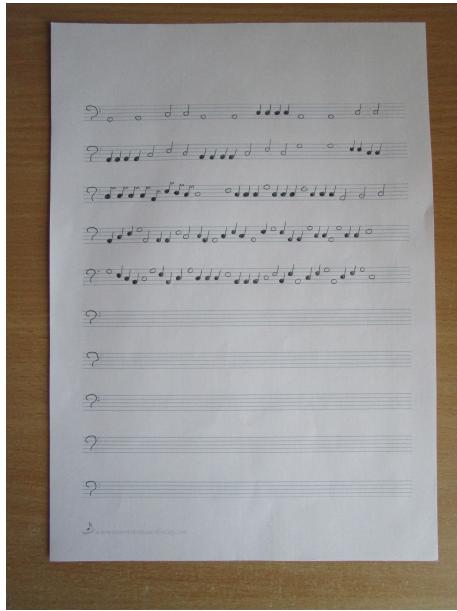
(b) próbka 5 wynik



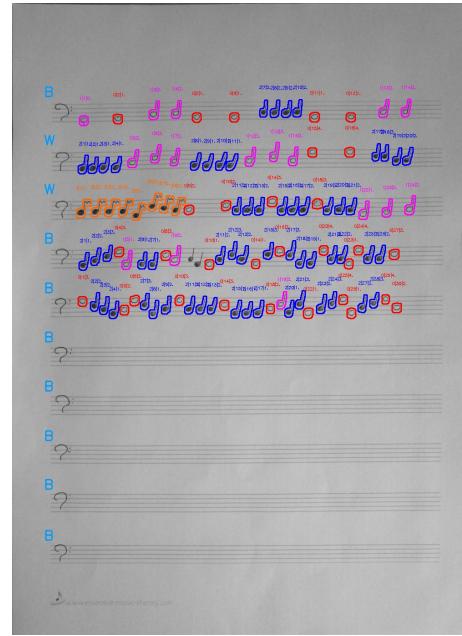
(c) próbka 6 przed procesem



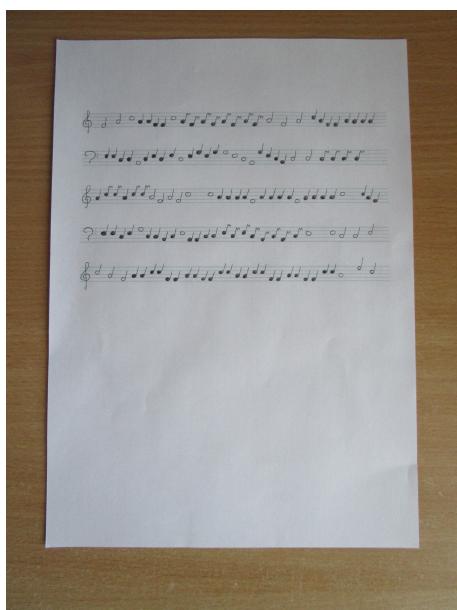
(d) próbka 6 wynik



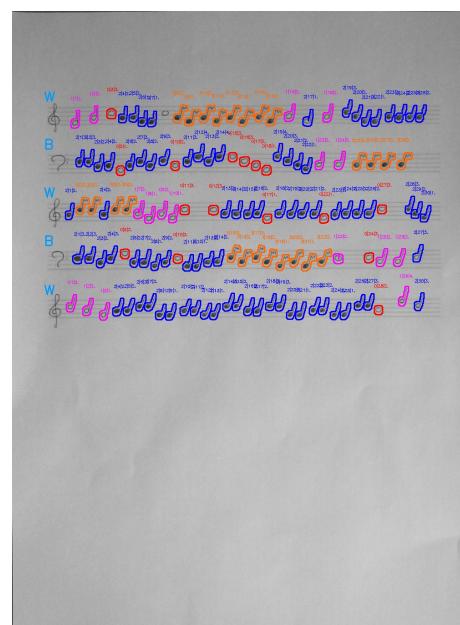
(a) próbka 8 przed procesem



(b) próbka 8 wynik



(c) próbka 13 przed procesem



(d) próbka 13 wynik

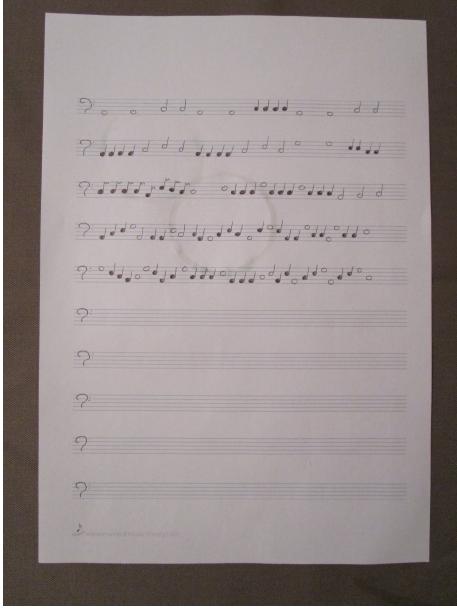
### 3.2 Średnie



(a) próbka 16 przed procesem



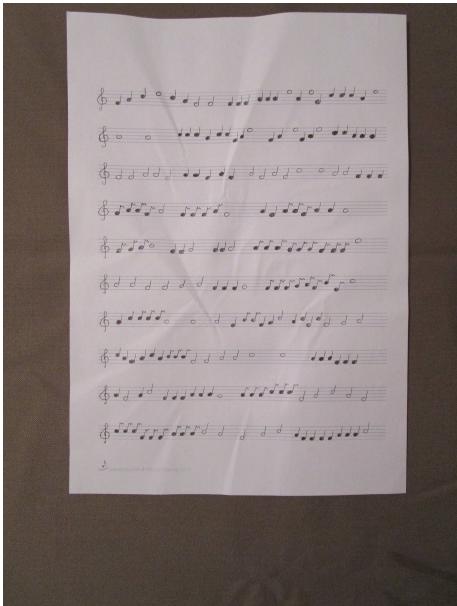
(b) próbka 16 wynik



(a) próbka 17 przed procesem

A musical score for sample 17 after processing. The score features two staves labeled 'W' (top) and 'B' (bottom). The notes are highlighted with various colors: blue, pink, and orange. Numerical values are placed above many of the notes, indicating specific processing parameters or detection scores. The notes are primarily eighth notes and sixteenth notes.

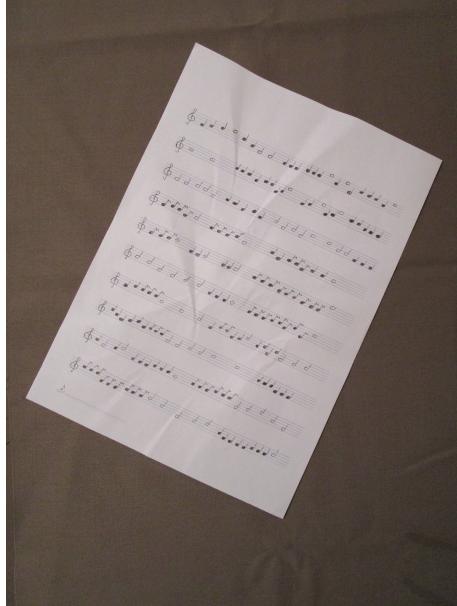
(b) próbka 17 wynik



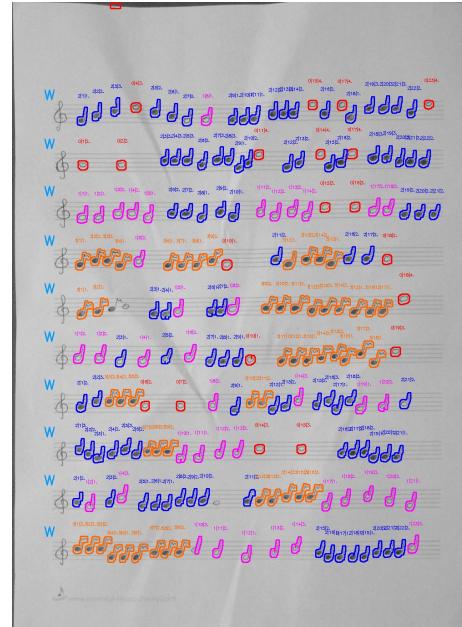
(c) próbka 20 przed procesem

A musical score for sample 20 after processing. The score features two staves labeled 'W' (top) and 'B' (bottom). The notes are highlighted with various colors: blue, pink, and orange. Numerical values are placed above many of the notes, indicating specific processing parameters or detection scores. The notes are primarily eighth notes and sixteenth notes.

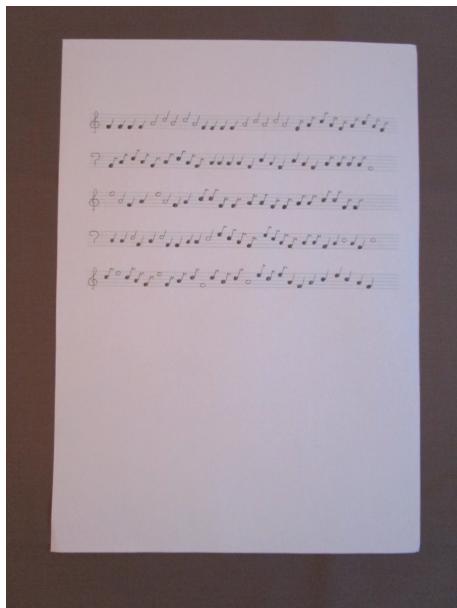
(d) próbka 20 wynik



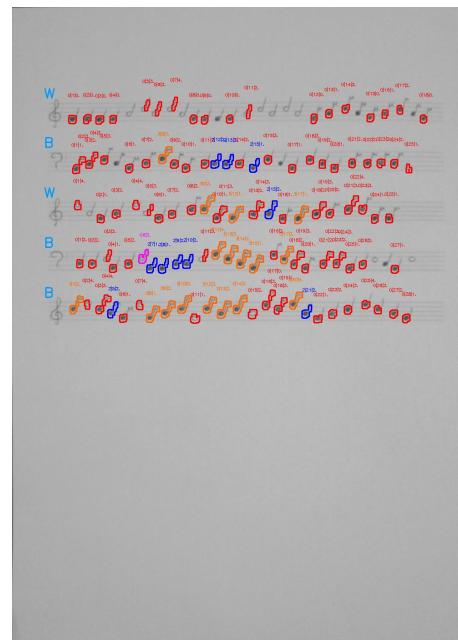
(a) próbka 21 przed procesem



(b) próbka 21 wynik

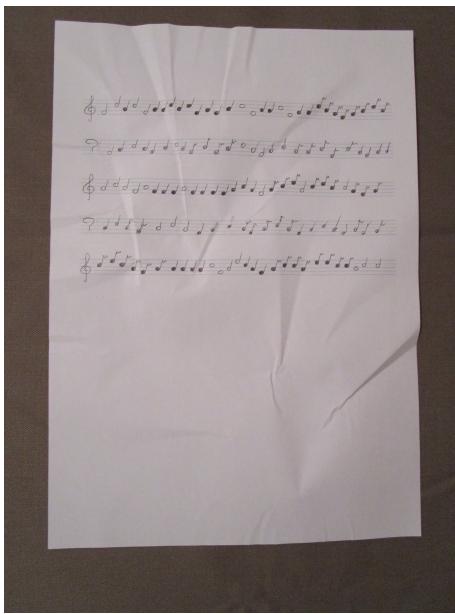


(c) próbka 41 przed procesem



(d) próbka 41 wynik

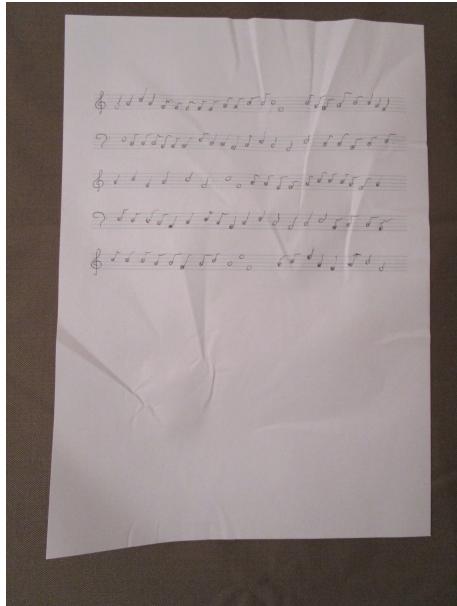
### 3.3 Trudne



(a) próbka 26 przed procesem



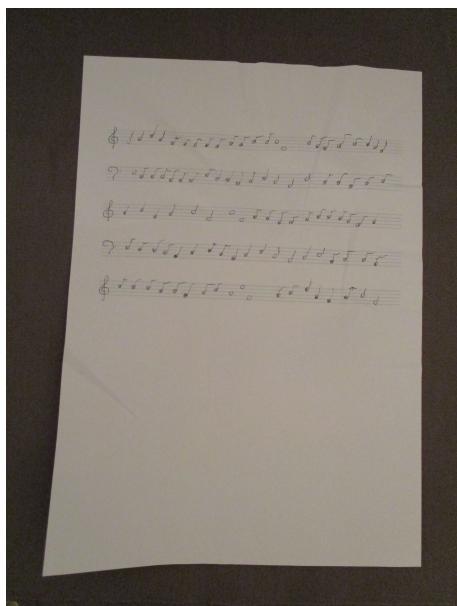
(b) próbka 26 wynik



(a) próbka 27 przed procesem



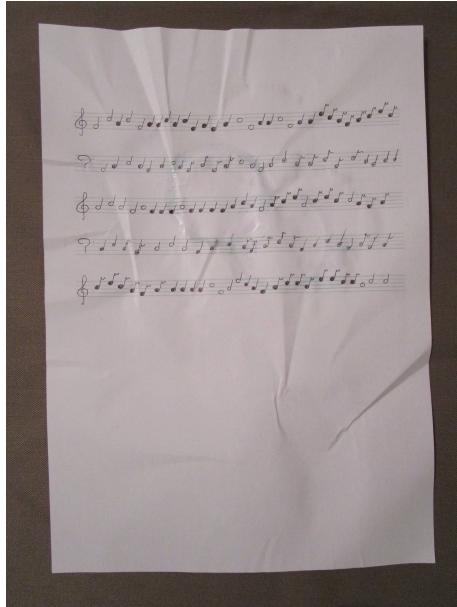
(b) próbka 27 wynik



(c) próbka 28 przed procesem



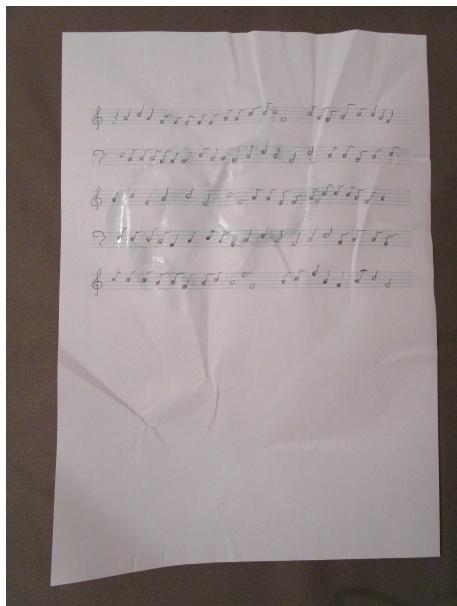
(d) próbka 28 wynik



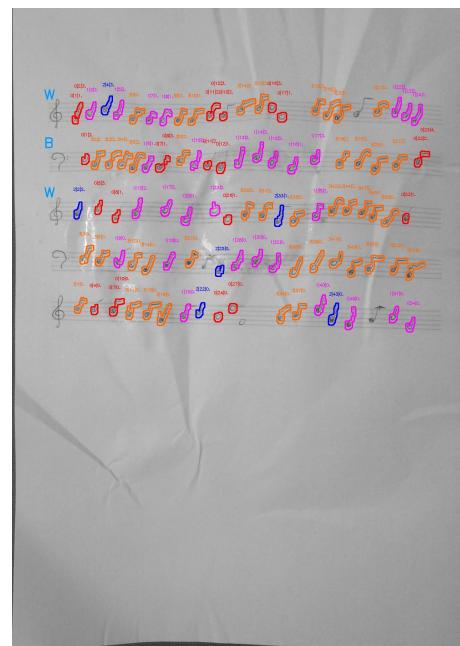
(a) próbka 30 przed procesem



(b) próbka 30 wynik



(c) próbka 31 przed procesem



(d) próbka 31 wynik