



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика, искусственный интеллект и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

ОТЧЕТ

по лабораторной работе № 6

на тему: «Компьютерное зрение. Поиск по шаблону. Четвертные ноты»

Студент ИУ7-13М
(Группа)

(Подпись, дата)

Шемякин А. А.
(И. О. Фамилия)

Преподаватель

(Подпись, дата)

Строганов Ю. В.
(И. О. Фамилия)

2022 г.

1 Теоретический раздел

1.1 Цель

Целью данной лабораторной работы является написание программы поиска четвертных нот на изображении по шаблону.

1.2 Поиск по шаблону

Поиск по шаблону - это метод поиска и нахождения местоположения изображения-шаблона в большем изображении. У OpenCV для такого случая имеется функция `cv.matchTemplate()`. Она буквально накладывает изображение шаблона на входное изображение (как при двумерной свертке) и сравнивает шаблон и участок входного изображения. В OpenCV реализовано несколько методов сравнения, был выбран `TM_CCOEFF_NORMED`. Формула

$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}} \quad (1.1)$$

Возвращается изображение в градации серого, где каждый пиксель обозначает, насколько окрестность этого пикселя совпадает с шаблоном.

Если входное изображение имеет размер (WxH), а изображение шаблона - размер (wxh), то выходное изображение будет иметь размер (W-w+1, H-h+1). Получив результат, можно использовать функцию `cv.minMaxLoc()`, чтобы найти, где находится максимальное/минимальное значение.

2 Практический раздел

2.1 Ход работы

На рисунке 2.1 показано исходное изображение.

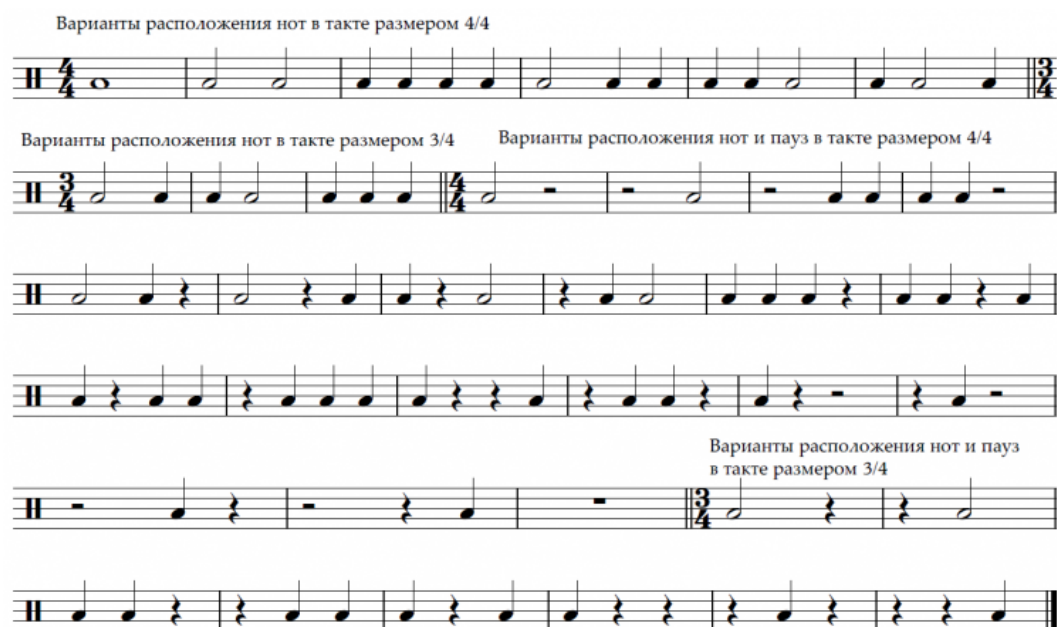


Рисунок 2.1 – Исходное изображение

Для шаблона я вырезал четвертную ноту с исходного изображения и убрал лишние детали. На рисунке 2.2 показан получившийся шаблон.



Рисунок 2.2 – Шаблон

Используем функцию `cv.matchTemplate()` и ее метод `TM_CCOEFF_NORMED`. Результат ее работы показан на рисунке 2.3.

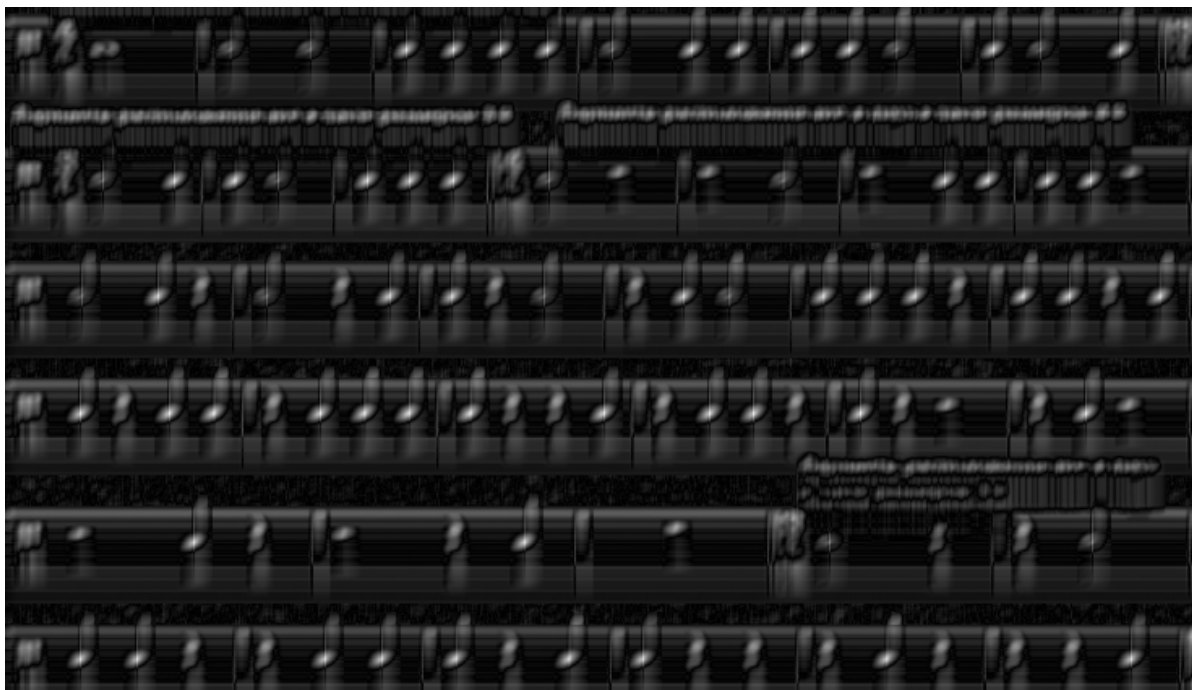
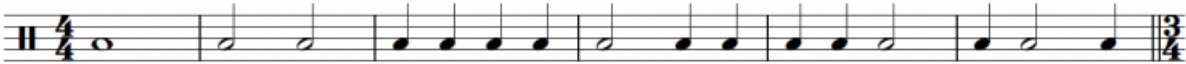


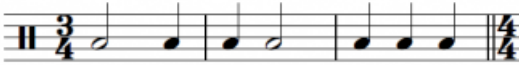

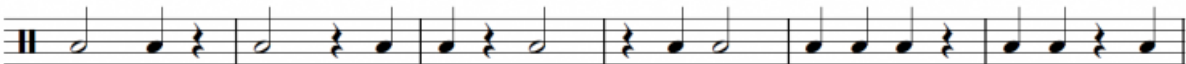

Рисунок 2.3 – Результат поиска

Найдем лучший результат совпадения. См. рисунок 2.4. Коэффициент совпадения 0.91. Интересный момент, что полученный лучший результат не совпадает с изначальным местом вырезки.

Варианты расположения нот в такте размером 4/4



Варианты расположения нот в такте размером 3/4 Варианты расположения нот и пауз в такте размером 4/4

Варианты расположения нот и пауз в такте размером 3/4

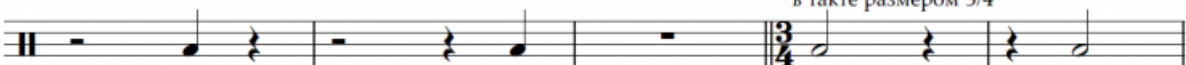




Рисунок 2.4 – Лучшее совпадение

Программа нашла 115 совпадений. А должна была найти всего 52. Чтобы убрать лишние дубликаты и сделать на каждый объект одно выделение - объединим близкие совпадения функцией `cv.groupRectangles()`. Полученный результат показан на рисунке 2.5

Варианты расположения нот в такте размером 4/4

Варианты расположения нот в такте размером 3/4

Варианты расположения нот и пауз в такте размером 4/4

Варианты расположения нот и пауз в такте размером 3/4

Рисунок 2.5 – Найденные четвертные ноты

3 Вывод

В результате проведенной лабораторной работы удалось написать программу поиска четвертных нот на изображении по шаблону. Следовательно, можно сделать вывод, что поставленная цель достигнута.