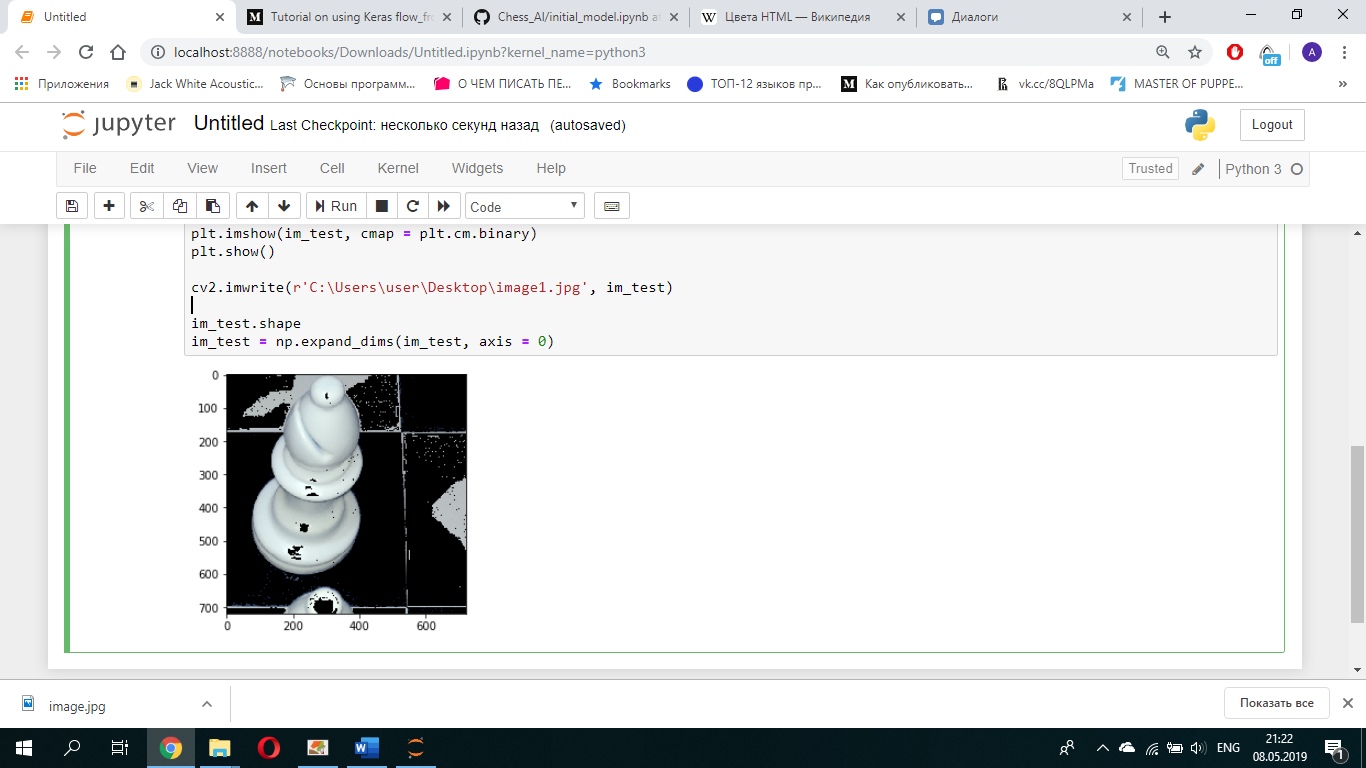
Точность распознавания можно значительно повысить (в данном случае от 67% до 92%) без увеличения количества данных с помощью преобразований над изображениями, упрощающими нейронной сети выделение ключевых особенностей. Выбор оптимального решения был произведен эмпирически, опробовав множество существующих способов.

Использованные преобразования:

**Маска**



(Для белых фигур)



(Для черных фигур)

Для использования маски подбирается цветовая модель, соответствующая цвету фигуры, которую необходимо преобразовать:

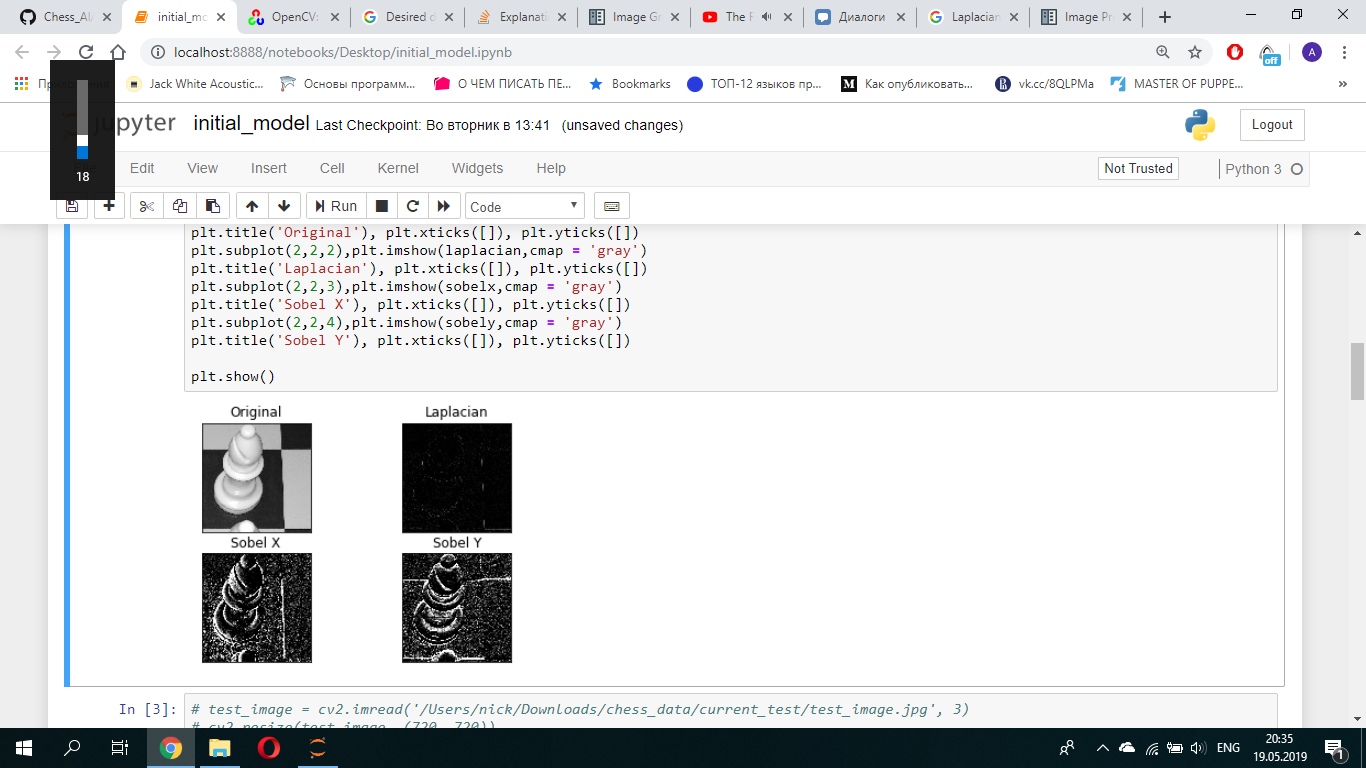
*lower\_black = np.array([16,10,10])* – цветовая модель для темного цвета

*upper\_black = np.array([255,255,255])* – цветовая модель для светлого цвета

Далее с помощью функций *cv2.inRange* и *cv2.bitwise\_and* остаются цвета, указанные в цветовой модели.

Данное преобразование не подходит для белых фигур, так как вследствие разной яркости изображений подбор модели для каждого изображения становится трудоемкой задачей.

**Операторы Лапласа, Зобеля**



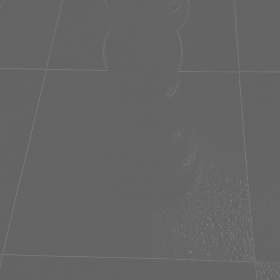
Данные преобразования используются для устранения шумов и подходит для изображений обоих цветов, что и послужило причиной выбора в их пользу для задачи обработки фотографий и черных, и белых фигур. Такой подход позволяет изменять изображения с фигурами обоих цветов одной и той же функцией. Это означает, что программа выигрывает в производительности, так как пропадает необходимость предопределять цвет фигуры и поля, на котором она стоит.

**Пороговая обработка или Threshold**



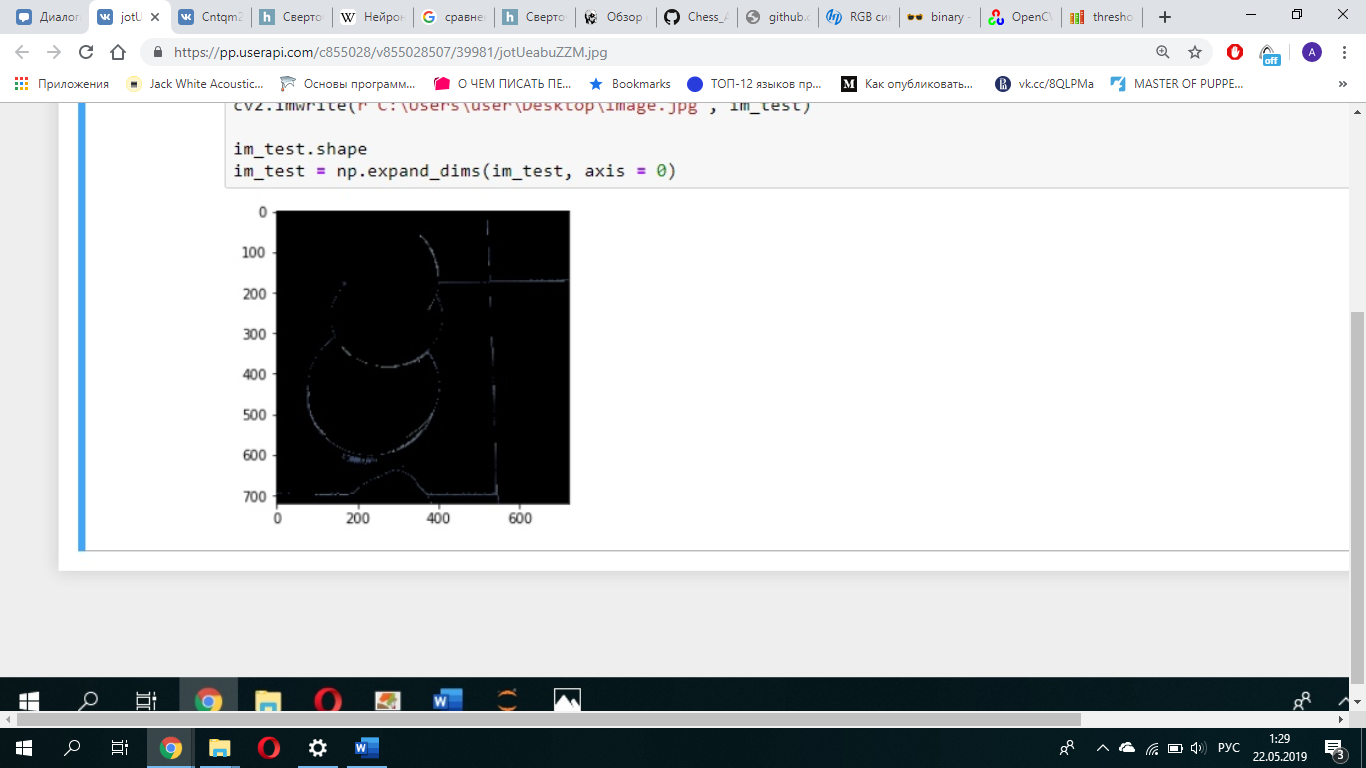
Это простые функции для избавления шумов на изображениях, но из-за особенностей набора данных фотографии становятся более зашумленными и качество классификации падает в сравнении с непреобразованными монохромными изображениями фигур.

**Гауссово размытие**



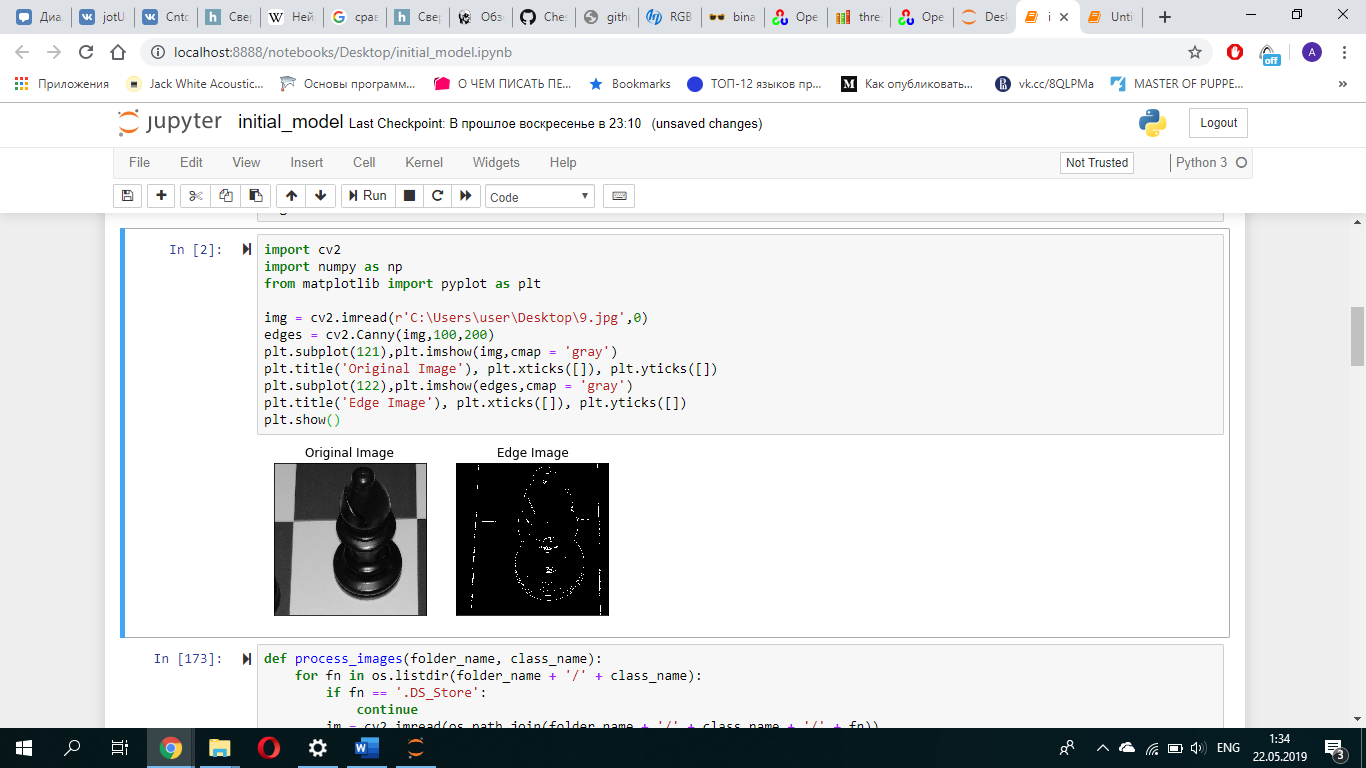
Также было использовано гауссово размытие, но желаемых результатов данный прием не принес.

**Градиент**



Градиент так же не был эффективным способом добиться увеличения качества распознавания фигур.

**Canny Edge**



Была опробована функция cv2.Canny, которая на практике не оправдала ожидания независимо от побора аргументов.

Вывод:

Эмперически было установлено, что самыми приемлемыми преобразованиями являются: маска -для черных фигур и оператор Sobel Y – для фигур любого цвета. Следовательно, финальный выбор пал на оператор Sobel Y ввиду его большей гибкости.