Федеральное государственное автономное образовательное учреждение высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"

Образовательная программа «Прикладная математика»
бакалавр

ОТЧЕТ
по проектной работе

Разработка системы распознавания шахматной позиции с помощью компьютерного зрения

Выполнили студенты

Пучко Николай Александрович

_____

Ахмадуллин Азат Робертович

_____

Вирютин Артём Валерьевич

_____

Руководитель проекта:

Доцент Иванов Фёдор Ильич

_____ _____
(оценка) *(подпись)*

_____
(дата)

24 мая 2019 г.

1 Введение

Современные методы трансляции шахматных партий являются потенциальным двигателем популяризации данного вида спорта. Крупные шахматные коммерческие компании (DGT, Fide, Square Off) и преуспевающие технические университеты (Stanford University, National Tsing Hua University) разрабатывают как инженерные решения, так и исключительно программные с применением компьютерного зрения. Данный проект нацелен на развитие второго способа как более перспективного и экономически обоснованного. Цель проекта – исследовать способы распознавания шахматной позиции и доработать существующие решения, повысив точность и стабильность алгоритмов компьютерного зрения.

Исследование было разбито на две задачи:

- Распознавание шахматной доски и построение координатной разметки.
- Определение класса шахматной фигуры по изображению.

Также по ходу разработки и выявления недостатков предыдущих моделей, были поставлены следующие дополнительные задачи:

- Собрать качественную тренировочную выборку данных.
- Эмпирически определить наиболее эффективные алгоритмы обработки изображений. Определить самую стабильную модель машинного обучения, где под стабильностью понимается наименьшее количество выбросов и артефактов в процессе обучения нейронной сети.

2 Основные функции

2.1 Разметка

Создание координатной разметки подразумевает разработку алгоритма, ставящего в однозначное соответствие каждому участку изображения, на котором расположена шахматная доска, верную шахматную нотацию, где нотация (название шахматной клетки) – двумерный вектор, по оси абсцисс имеющий буквы (a, b, c, d, e, f, g, h), по оси ординат числа $(1, 2, 3, 4, 5, 6, 7, 8)$.

Были использованы два подхода, оба из которых содержатся в специализированной на компьютерном зрении библиотеке OpenCV, реализованных ресурсами языка программирования Python.

findChessboardCorners() – функция OpenCV, более очевидное решение, но менее стабильное. Не смотря на название, данная функция была разработана для калибровки камер. Она возвращает массив координат углов шахматного поля.

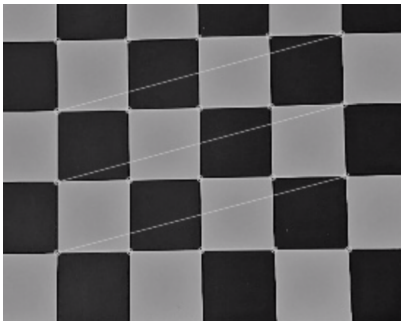

Рис. 1: Однако минимальные шумы на фотографии мешали работе функции, что недопустимо в реальном использовании.

goodFeaturesToTrack()– функция OpenCV, более общее решение, в чем и заключается его преимущество. Она обнаруживает замечательные особенности на изображении, что в случае с шахматной доской является пересечением черных полей по диагоналям. Таким образом удается получить массив координат 7×7, полностью характеризующим координатную сетку доски. У *goodFeaturesToTrack* были обнаружены два недостатка:

1. Цикл определения нотации заданной клетки ошибается при ракурсе изображения, отличного от перпендикулярного. Решение было найдено в изменении перспективы фотографии с помощью встроенных ресурсов OpenCV getPerspectiveTransform и WarpPerspective. Задав углы шахматной доски, можно получить преобразованное масштабированное изображение, исключающее артефакты при работе *goodfeaturesToTrack*.

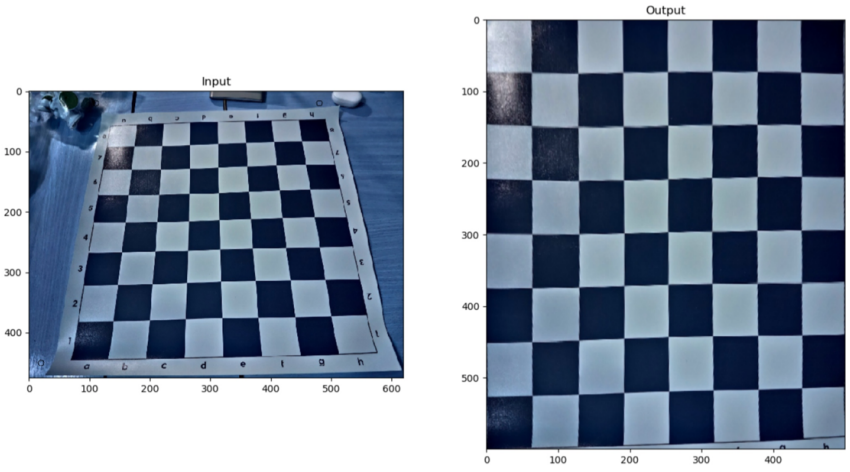

Рис. 2:

2. Функция не обнаруживает крайние углы шахматной доски ввиду отсутствия контакта с черными полями (за пределами доски нет объектов, позволяющих функции связать особенности). Проблема решается алгоритмически, добавляя крайние координаты на некоторое количество пикселей, равное средней разнице между ближайшими определяемыми углами. В общем случае этого метода достаточно, хотя не исключается возможность более адаптивного расширения.

По итогу разметка может быть использована для определения границ нарезки входного изображения и автоматического получения набора из 64-х изображений, соответствующих каждому из полей шахматной доски. Такой способ значительно упрощает алгоритм подачи данных нейронной сети, в перспективе позволяющий не нагружать ресурсы устройства сложными методами обхода вроде R-CNN (region convolutional neural network) или YOLO (you only look once). Тем не менее, вопрос поиска крайних углов шахматной доски при построении перспективы нестандартных изображений остается открытым.

2.2 Нейронные сети

Начиная с 2012 года лучшие результаты среди алгоритмов машинного обучения в области распознавания изображений показала сверточная нейронная сеть CNN (Convolutional Neural Network). CNN показывает такие результаты благодаря устойчивости к шумам, содержащимся в тренировочных данных: поворотам, изменениям ракурса, изменениям масштаба и т.д. Соответственно, был выбран именно этот метод для определения шахматных фигур на двумерных изображениях, так как фотография шахматной доски может быть получена под разнообразными ракурсами, не учтенными при подготовке обучающей выборки.

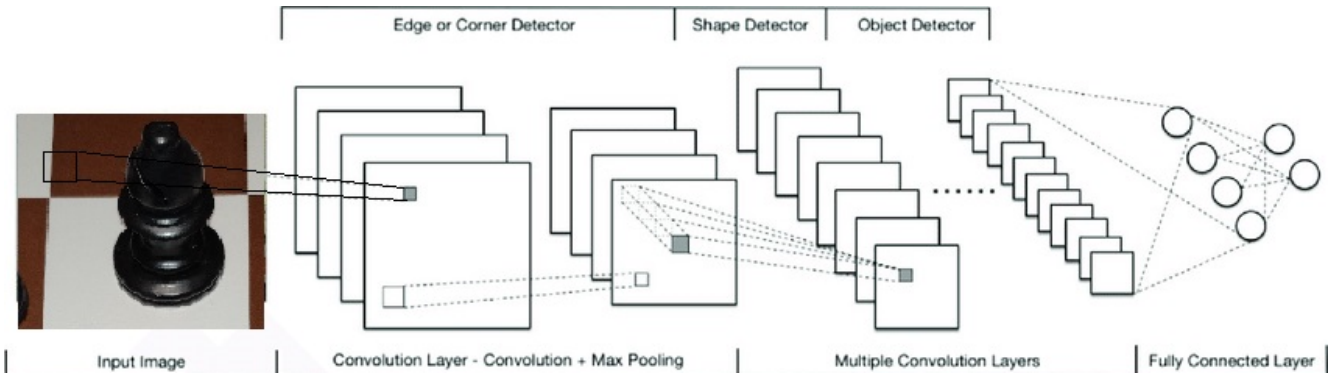

Рис. 3: Комплексная структура сверточной нейронной сети при обработке изображения

Примененная при решении задачи нейронная сеть имеет 3 сверточных слоя и 2 слоя нейронов, первый из которых – полносвязный слой, второй – выходной слой. Сверточные слои представляют собой условный набор карт признаков, где каждая карта имеет ядро, которое также называется фильтром и имеет размер 3×3. Этот фильтр реализован в виде матрицы, элементы которой – весовые коэффициенты, итерационно корректируемые в процессе обучения нейронной сети.

Такая структура нейронной сети позволяет находить особенные признаки на изображениях и затем отличать класс принадлежности фигуры на фотографии. В качестве функции активации была использована relu (rectified linear unit) - $f(x) = max(0, x)$.

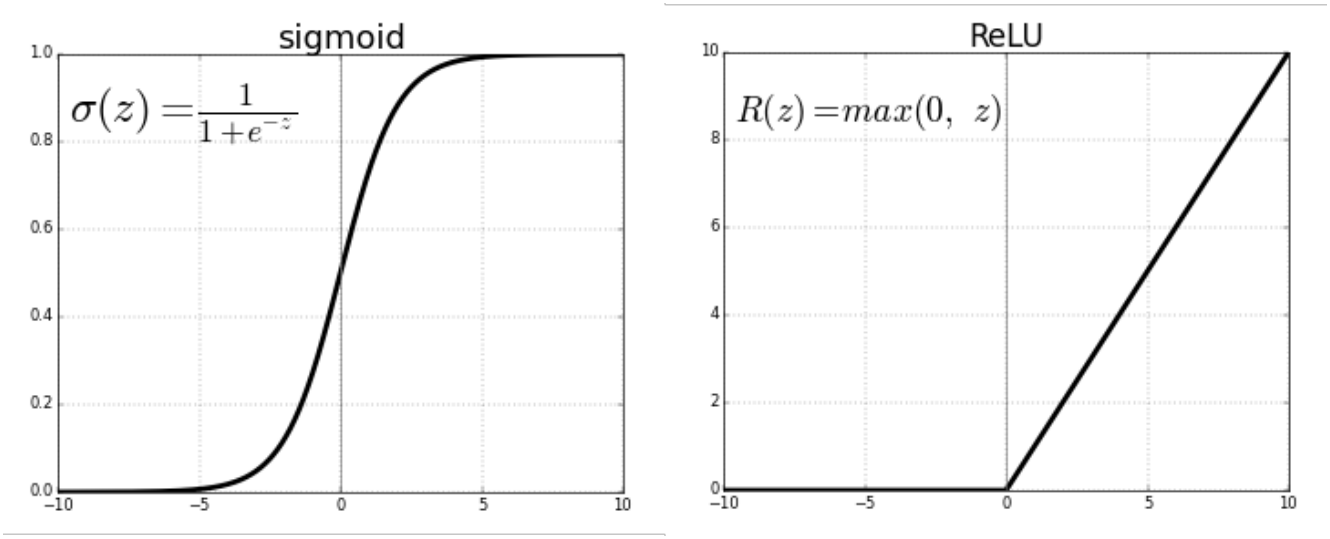

Рис. 4:

Достоинством такой функции является то, что в процессе обучения relu не использует ресурсоемкие операции, что сокращает время обучения и позволяет более эффективно обучать нейросеть на большем количестве устройств. Еще один плюс выбранной функции активации – она минимизирует влияние деталей, не играющих большой роли при классификации (в данном случае это преимущество перед аналогичной функцией – сигмоидой, которая является более толерантной к менее замечательным деталям). Поглощение шумов в виде контуров шахматных полей на фоне самих фигур увеличивает точность распознавания изображений в случае с не бинарным (categorical) классификатором.

3 Тренировочные данные

3.1 Набор данных

Хорошая нейронная сеть требует хороших данных. В нашем случае набор состоит из изображений шахматных полей, распределенных на 13 классов: черные и белые фигуры (пешки, кони, слоны, ладьи, ферзи, короли) и пустые клетки (черные и белые поля в одном классе ввиду особенностей обработки изображений). Ввиду того, что была выбрана библиотека TensorFlow, и использовались тензорные вычисления, каждое изображение проходило дополнительное преобразование в формат тензора.

Эмпирически была выбрана следующая пропорция данных между тренировочным, валидационным и тестовым наборами - 345 : 67 : 75. Распределение данных по классам неравномерно – более многочисленные классы, такие как пустые поля и пешки получили большие наборы данных. Также расширенный набор был собран для классов слонов ввиду ощутимых трудностей при обучении модели распознаванию именно этого класса (что примечательно, ни в одной ранее существовавшей статьей не упоминалась «проблема слона»).

```
Found 345 images belonging to 13 classes.
Found 67 images belonging to 13 classes.
Found 75 images belonging to 13 classes.
```

Рис. 5:

Для обучения различных моделей нейронных сетей в начале проекта были использованы тренировочные данные, свободно распространяемые в открытом доступе. Однако все эти наборы данных были собраны некачественно (недостаточно вариативные положения фигур и уровни освещения), что неминуемо вело к переобучению на любом из способов машинного обучения. Также коллеги ограничивали спектр возможных ракурсов, что не позволяло использовать в реальных условиях, когда фотография получается из произвольного положения. На эту тему была ранее опубликована статья, где теоретически было приведена аргументация в пользу того факта, что оптимальным углом по отношению к фигуре на плоскости для получения изображения является $\frac{\pi}{4}$ при расположении камеры вертикально над доской. Таким образом, стала задача создать полностью новый набор данных, лишенный приведенных выше недостатков.

Name	Train	Val	Test
	70%	15%	15%
BB	29	6	6
BN	26	6	6
BP	63	13	13
BR	14	3	3
BQ	14	3	3
BK	12	2	2
E	70	14	14
WB	29	6	6
WN	16	4	4
WR	13	3	3
WQ	14	3	3
WK	13	2	2

Рис. 6: Распределение тренироовчных данных по 13 классам

Специально для сбора данных был приобретен наиболее распространенный турнирный шахматный комплект – Staunton 6 с размерами клеток $5,7 \times 5,7$ см (самый крупный из существующих турнирных габаритов – крупные клетки лучше подходят для выявления замечательных особенностей нейронной сетью на изображении). 487 фотографий шахматных

полей были сделаны и обработаны вручную, что позволило сохранить качество тренировочной выборки на одинаково высоком уровне для всех классов и всех ракурсов. Был испробован более автоматизированный подход – функция *crop*, обрезающая изображения по заданными точкам, которые можно алгоритмически принять из функции *findChessboardCorners*. Однако этот способ корректно обрезает шахматные поля только в случае с перпендикулярным ракурсом, что противоречит основной идее нынешнего проекта. Все собранные данные будут оставлены в открытом доступе для дальнейших независимых разработок по данной теме.

3.2 Преобразования

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

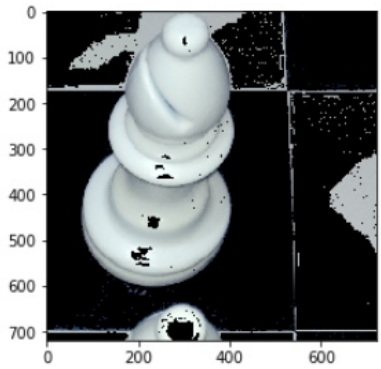

Рис. 7: Для белых фигур

Рис. 8: 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

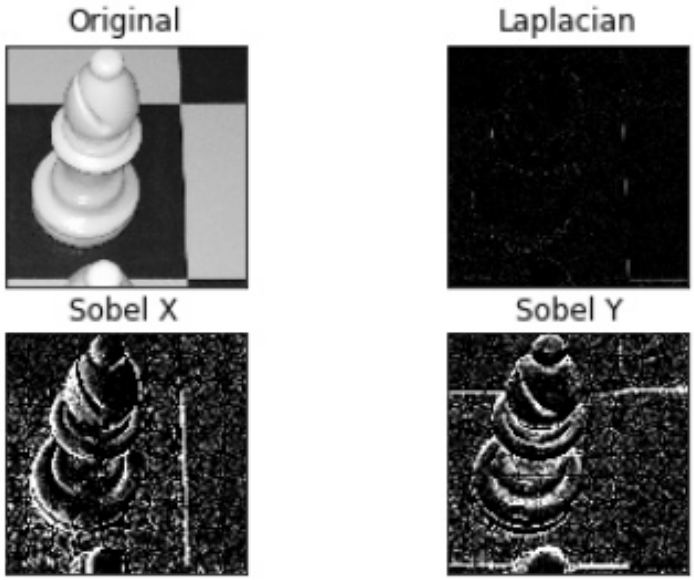

Рис. 9:

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

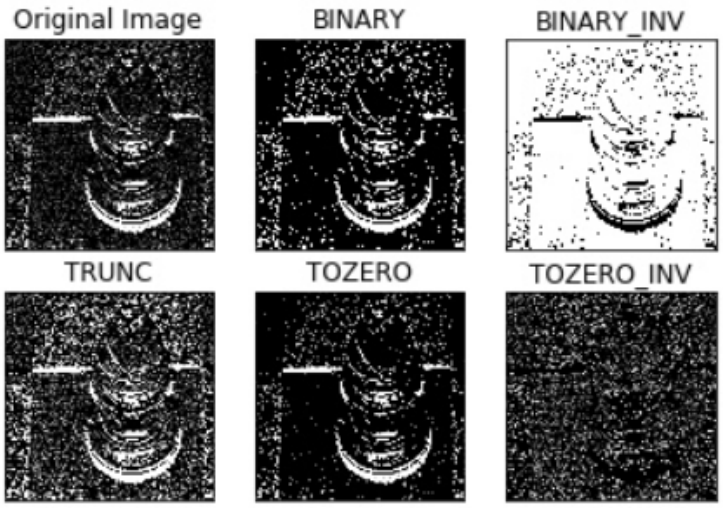

Рис. 10:

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

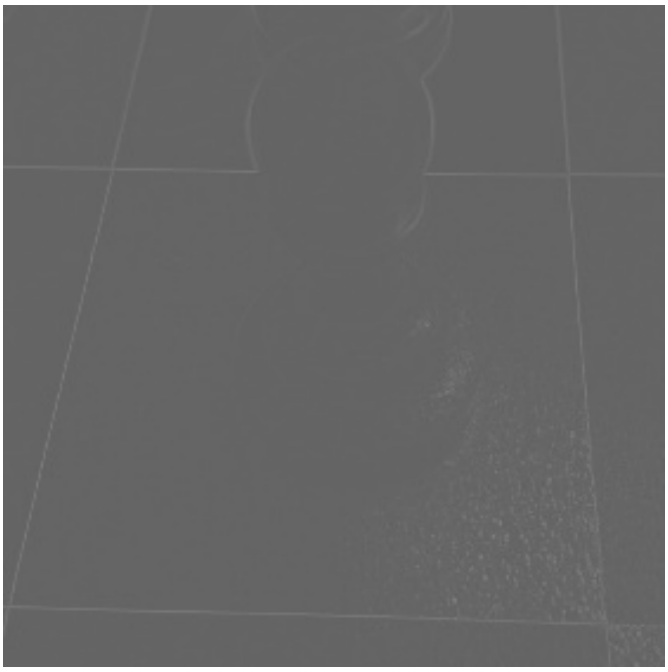

Рис. 11:

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

Градиент

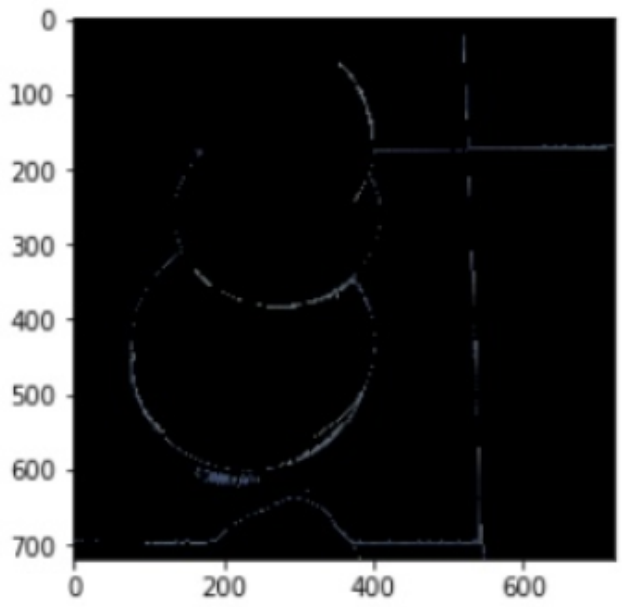

Рис. 12:

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

Canny Edge

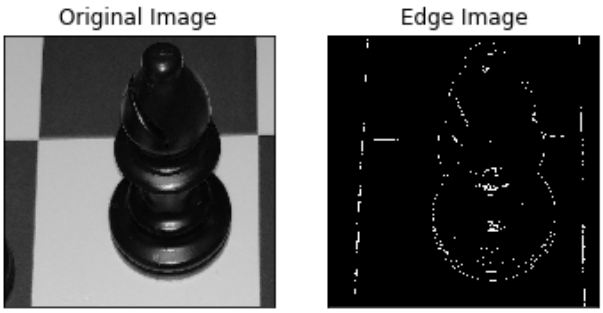

Рис. 13:

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

4 Результаты

Результат целевого показателя (вероятность распознавания верного класса из 13 возможных на произвольном тестовом изображении) превзошел начальные ожидания и приблизился к достижениям ведущих команд в этой области.

```
scores = model.evaluate_generator(test_generator)
print("Вероятность распознавания на тестовых данных: %.2f%%" % (scores[1]*100))
```

```
Вероятность распознавания на тестовых данных: 92.31%
```

Рис. 14:

Такая точность подает надежду на промышленное использование технологии в обозримом будущем, так как вероятность ошибочного определения в процессе трансляции шахматной партии стремительно снижается при дальнейшем уменьшении ошибки.

В процессе исследования был обнаружен важный фактор, который игнорировался во всех предшествующих разработках по данной теме – для распознавания шахматного хода (или всей позиции) необходимо обработать множество объектов (64), что принуждает к использованию таких моделей машинного обучения, как R-CNN или YOLO. Пусть данные алгоритмы и достигли значительно большей точности распознавания, чем на заре технологий динамической обработки классов, они все еще расходуют колоссальные ресурсы (даже относительно эффективный YOLO требует GPU для потоковой обработки в 25 fps) по сравнению с классическими классификаторами, что делает невозможным трансляции в динамичных шахматных режимах (блиц и рапид). Проблема была нивелирована применением библиотеки компьютерного зрения для предварительного деления изображения доски на сформированные участки, что представляется возможным в случае с шахматами, так как сравнительные габариты полей и самой доски определяются алгоритмически при заданных углах игрового поля.

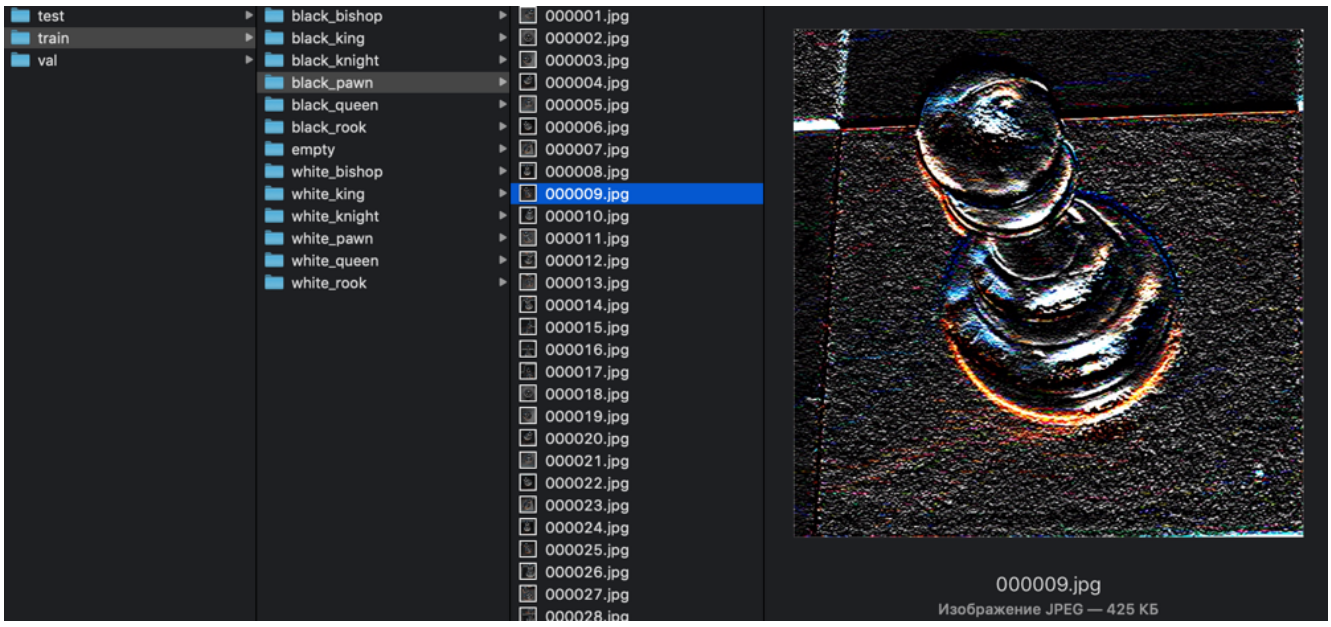

Рис. 15: файловая структура собранных данных

Важным достижением стало создание качественного набора тренировочных данных, содержащего все необходимое для обучения различных моделей распознавания. Он сократит порог входа для новых исследований по данной теме сторонним разработчикам.

Пусть средняя продолжительность классической турнирной шахматной партии составляет 40 ходов, что сводится к 80 итерациям распознавания, если использовать информацию о

предыдущем состоянии позиции, описывая ход по перемещению одной фигуры. Тогда в среднем ошибка будет происходить в трех ходах за партию $(1 - 0,9231) \times 80$. Хотя этого все еще недостаточно для турнирного использования (следуя опросу практикующих профессиональных шахматистов), но это результат, полученный в кустарных условиях на небольшой тренировочной выборке. Следуя гипотезе, при достижении $>97\%$ точности, технология может быть внедрена в рынок как альтернатива инженерным устаревающим решениям трансляции шахматных турниров голландской монополисткой компании DGT.

Список используемой литературы / References

[1] Jialin Ding - «ChessVision: Chess Board and Piece Recognition»

[2] Cheryl Danner, Mai Kafafy - «Visual Chess Recognition»

[3] Maciej A. Czyzewskia, Artur Laskowskia,b, Szymon Wasika - «Computer Vision and Image Understanding»

[4] Daylen Yang - «Building Chess ID»

Приложение

Ссылка на репозиторий GitHub. В нем содержится весь описанный в отчете код и ссылка на облачное хранилище с тренировочными данными.

https://github.com/NickPuchko/Chess_AI