«Национальный исследовательский университет
«Высшая школа экономики»

Групповой проект

«Классификация изображений, датасет на основе скриншотов/моделей видеоигры»

Выполнили:

Баранова Дарья

Павлов Сергей

Петренко Денис

Содержание

1. Введение

1.1 Цели и задачи работы

Целью работы ставится:

– Изучение подходов к обработке изображений и выделению из них ключевых признаков;
– Изучение способов классификации изображений с помощью классических подходов машинного обучения;
– Решение собственной задачи классификации изображений методами классического машинного обучения (без использования нейронных сетей).

В качестве собственной задачи была выбрана задача классификации изображений со скриншотами моделей оборудования из игры «Космические рейнджеры 2: Доминаторы».

Для достижения цели были выделены следующие задачи работы:

1) Собрать и обработать (при необходимости) датасет необходимого размера;
2) Выбрать метрики качества, с помощью которых будет оцениваться качество будущей модели;
3) Разработать baseline метод для классификации изображений, а также спроектировать и обучить модель;
4) Сформулировать и обосновать возможные способы улучшить полученное качество классификации и реализовать их. Оценить, насколько улучшилось качество классификации;
5) Подвести итоги и описать наилучшую полученную модель для решения поставленной задачи.

2. Сбор данных

В качестве изображений для классификации были выбраны статические рейнджеры оборудования в космическом корабле игрока из игры «Космические рейнджеры 2: Доминаторы».

Сбор данных был произведён вручную с помощью созданной поклонниками игры утилиты [1], предназначенной для извлечения графических элементов из дистрибутива игры. Такая возможность позволила не пользоваться внутриигровыми скриншотами, которые содержали бы в себе лишние элементы фона и, возможно, были бы хуже исходников по качеству.

В исходных файлах игры каждое изображение оборудования представляет собой циклический анимированный ряд из ~100 кадров. Поскольку для задачи необходимы статические изображения, для каждой модели оборудования был вручную выбран наиболее репрезентативный (на глаз) кадр.

В итоге каждое исходное изображение обладает следующими характеристиками:

- формат: PNG
- размер: 75 * 75 pixel
- фон: прозрачный

Исходная выборка содержит 102 изображения, которые относятся к различным 9 классам оборудования со следующим распределением:

Наименование класса	Количество уникальных изображений
Оружие (WEAPON)	15
Генератор защитного поля (DEFGENERATOR)	11
Двигатель (ENGINE)	11
Топливный бак (FUELTANKS)	11
Радар (RADAR)	11
Ремонтный дроид (REPAIRROBOT)	11
Сканер (SCANER)	11
Корпус (HULL)	11
Захват (CARGOHOOK)	10

Каждый класс представляет собой множество элементов, внешне сильно различающихся между собой. Например, класс «Оружие» состоит из 15 различных видов орудийный инструментов различной формы, размера и цвета. Однако все эти орудийные инструменты принадлежат категории «Оружие», а значит должны быть отнесены к одному классу несмотря на их внешнее сильное различие. Изображения элементов классов представлены в приложении.

Чтобы увеличить объем обучающей выборки исходные изображения были аугментированы разными искажениями до получения 3000 изображений со следующим распределением:

Наименование класса	Количество уникальных изображений
Оружие (WEAPON)	480
Генератор защитного поля (DEFGENERATOR)	325
Двигатель (ENGINE)	345
Топливный бак (FUELTANKS)	322
Радар (RADAR)	317
Ремонтный дроид (REPAIRROBOT)	312
Сканер (SCANER)	312
Корпус (HULL)	310
Захват (CARGOHOOK)	277

При обучении данные разбивались на train и test в отношении 3:1, после чего на каждой из подвыборок проводилась аугментация. Таким образом, до аугментации обучающая выборк а состояла из 76 изображений, а тестовая из 26.

Аугментация каждого изображения проводилась с помощью пайплайна изменений:

- Поворот (*rotate*) от -8 до +8 градусов с вероятностью 0.5;
- Изменение масштаба (*zoom*) с коэффициентом от 0.8 до 1.01 с вероятностью 0.5;
- Отражение по горизонтали (*flip_left_right*) с вероятностью 0.5;
- Случайное искажение (*random_distortion*) с вероятностью 0.7 (пример искажения представлен на рисунке ниже).

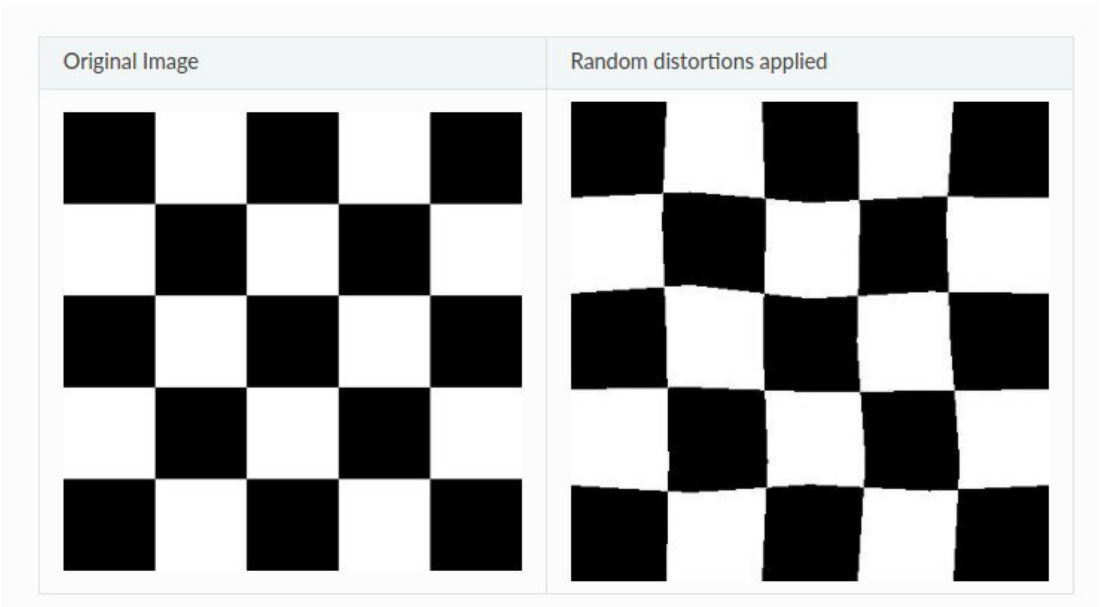

Рис. Пример случайного искажения (слева – оригинальное изображение, справа - искаженное)


Пример полученных в результате аугментации двух изображений одного и того же объекта

Количество сгенерированных с помощью аугментации изображений выбиралось исходя из двух предпосылок:

1) Чем больше обучающая выборка будет, тем лучшего качества можно будет достичь при обучении модели;
2) Слишком большое количество данных может повлечь слишком долгое обучение (подробнее это описано в разделе построения baseline модели).

3. Построение baseline модели

3.1 Предпосылки

Изображения разных классов не сильно различаются между собой. Можно сказать, что даже для человеческого глаза изображения разных классов трудно различимы, так как все модели игры имеют довольно разнообразный внешний вид.

Исходя из этого, было принято решение строить baseline на основе техники Bag-of-words для computer vision [2-4].

3.2 Описание метода

Для извлечения признаков из изображений использовалось масштабно-инвариантное преобразование функций SIFT (Scale Invariant Feature Transform) [5-6].

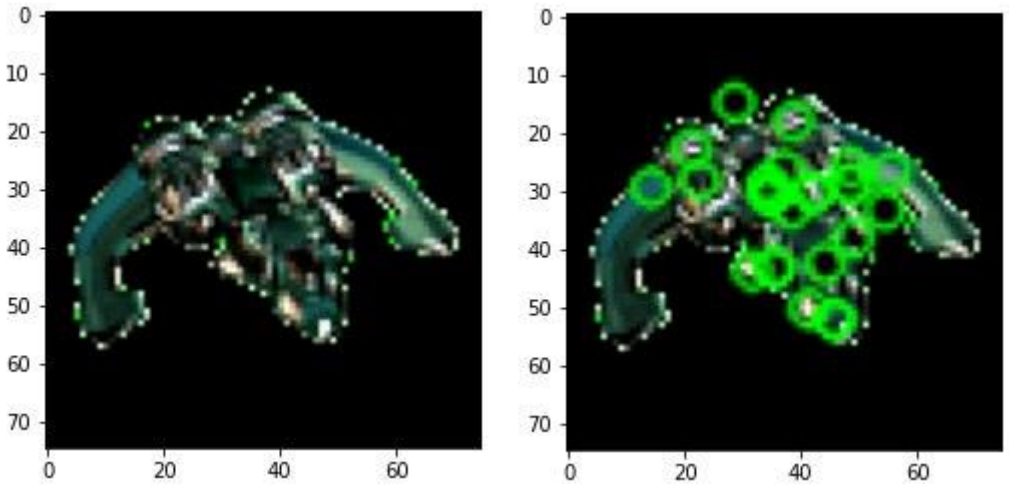

Рис. Определение ключевых точек методом SIFT (слева – исходное изображение, справа с отрисованными выделенныи точками)

После этого на основе выделенных дескрипторов каждого изображения строится соответствующее векторное пространство. Из множества дескрипторов для изображений получается набор векторов в этом пространстве с соответствующими координатами. Однако так как векторов много и нужно как-то выделить «ключевые точки», которые и будут представлять собой слова в словаре, нужно все это пространство кластеризовать. Проводится эта процедура с помощью K-Means алгоритма, который данное векторное пространство разбивает на K кластеров – опорных точек пространства. Эти опорные точки представляют собой «словарь» в представлении «Bag-of-words».

Следующим шагом исходные изображения описывались с помощью полученных опорных точек. Для каждого изображения определяется какие опорные точки лучше всего его описывают (какие кластеры находятся ближе всего к каждому из векторов его векторного представления). Таким образом, каждое изображение мы представляем как набор «слов из словаря».

После этого для каждого такого набора слов можно построить гистограмму. Таким образом, для каждого изображения мы получим по одной гистограмме. Именно их мы и будет классифицировать с помощью метода опорных векторов.

Описанный метод классификации накладывает значительное ограничение на объем обучающей выборки. K-Means алгоритм в процессе работы вычисляет матрицу расстояний, из-за этого время обучения на больших объемах данных может сильно возрастать.

3.3 Метрики и полученное качество

Поставленная задача представляет собой задачу многоклассовой классификации. Однако baseline не предполагает построения нескольких моделей для бинарной классификации. Вместо этого строится одна модель, которая на выходе может выдавать один из 9 классов.

В качестве метрики качества для такой модели будет использоваться Accuracy, задающаяся как:

$$Accuracy = \frac{\sum_{i=1}^{n}[y_i == c_i]}{n},$$

где:

- $y_i - $ ответ модели на $i -$ ом объекте;
- $c_i -$ истинный ответ для $i -$ ого объекта.

В собранных данных классы распределены почти что равномерно, поэтому адекватное применение Accuracy возможно.

Эмпирически были получены следующие метрики качества:

Количество итераций метода K-Means	Количество кластеров	Размер обучающей выборки	Размер тестовой выборки	Качество на train	Качество на test
1	200	900	763	1.0	0.208
		1500	763	1.0	0.218
		2238	763	0.999	0.207
	500	900	763	1.0	0.227
		1500	763	1.0	0.223
		2238	763	1.0	0.224
5	200	900	763	1.0	0.231
		1500	763	1.0	0.235
		2238	763	1.0	0.231
	500	900	763	1.0	0.203
		1500	763	1.0	0.239
		2238	763	1.0	0.187

Столбец «Количество кластеров» отражает величину K в методе K-Means.

Так как метод работает остаточно долго, K-Means в приведённой выше таблице проходил всего одну и пять итераций. При попытке выставить 5 итераций качество классификации выросло, но, к сожалению, не сильно. Увеличение количеств итераций K-Means улучшает качество, но очень сильно увеличивает и время обучения модели, поэтому дальнейшее увеличение количества итераций было признано нецелесообразным.

Также было замечено, что качество модели сильно зависит от разбиения выборки. Если не фиксировать random seed, то accuracy на тестовой выборке будет варьироваться от 0.22 до 0.29 при одних и тех же параметрах модели.

4. Улучшение модели

Следующие шаги могут привести к улучшению качества текущего решения:

1) Применить другую модель. Например, градиентный бустинг;
2) Провести подбор гиперпараметров модели;
3) Применить другой алгоритм для извлечения признаков из изображения. Это могут быть следующие алгоритмы:
   a. ORB (Oriented FAST and Rotated BRIEF) [7];
   b. SURF (Speeded Up Robust Features) [8].

4.1 Calibrated SVM

Возьмем построенные модели SVM и посмотрим на вероятности, с которыми модель определяет класс для каждого тестового изображения. После калибровки вероятностей и выбора наиболее вероятного класса увидим, что такое незатейливое улучшение сильно увеличивает качество классификации:

Количество итераций метода K-Means	Количество кластеров	Размер обучающей выборки	Размер тестовой выборки	Accuracy baseline	Улучшенное Accuracy
1	200	900	763	0.208	0.224
		1500	763	0.218	0.246
		2238	763	0.207	0.229
	500	900	763	0.227	0.219
		1500	763	0.223	0.227
		2238	763	0.224	0.245
5	200	900	763	0.231	0.254
		1500	763	0.235	0.229
		2238	763	0.231	0.249
	500	900	763	0.203	0.208
		1500	763	0.239	0.240
		2238	763	0.187	0.189

4.2 XGBoost

В качестве альтернативной модели также был опробован XGBClassifier. Результаты его работы отражены в таблице. Оптимальные параметры перебирались с помощью GridSearch, перебирались глубина и количество принимающих решения деревьев. Таблица для удобства отсортирована по «Accuracy test» столбцу.

Максимальная глубина дерева	Количество деревьев в модели	Качество на test	Качество на train
6	16	0.244114	1.000000
8	16	0.235857	1.000000
6	12	0.232251	0.998829
8	12	0.229060	0.999881
4	16	0.224459	0.995426
10	16	0.221202	1.000000
4	12	0.219464	0.988250

8	08	0.216995	0.998826
10	12	0.214813	1.000000
6	08	0.214510	0.994135
2	16	0.213475	0.910877
4	08	0.213155	0.967717
2	12	0.207840	0.874651
2	08	0.203418	0.818542
10	08	0.202061	1.000000
6	04	0.193195	0.967250
10	04	0.193014	0.995542
8	04	0.191901	0.986604
4	04	0.191824	0.907069
2	04	0.182342	0.716368

Видно 2 закономерности:

1. модель очень сильно переобучается;
2. увеличение количества деревьев дает прирост точности.

Таким образом, наилучшее качество показала модель с глубиной = 8 и количеством решающих деревьев = 16. Доля правильных ответов составила 0.244114.

Исходя из выводов по обучению модели можно сказать:

1. Нужно бороться с переобучением, которая, по нашему мнению, является основным препятствием при решении задачи;
2. Для улучшения качества нужно расширять сетку (добавлять деревья) - стоим на верхней границе.

4.3 Извлечение ORB признаков

Попробуем применить другой механизм извлечения признаков из изображения. Вместо SIFT признаков попробуем использовать ORB (Oriented FAST and Rotated BRIEF) дескрипторы особых точек изображения. После алгоритма будем получать векторы меньших размеров по сравнению с SIFT (32 и 128 соответственно), поэтому скорость обучения должна увеличиться.

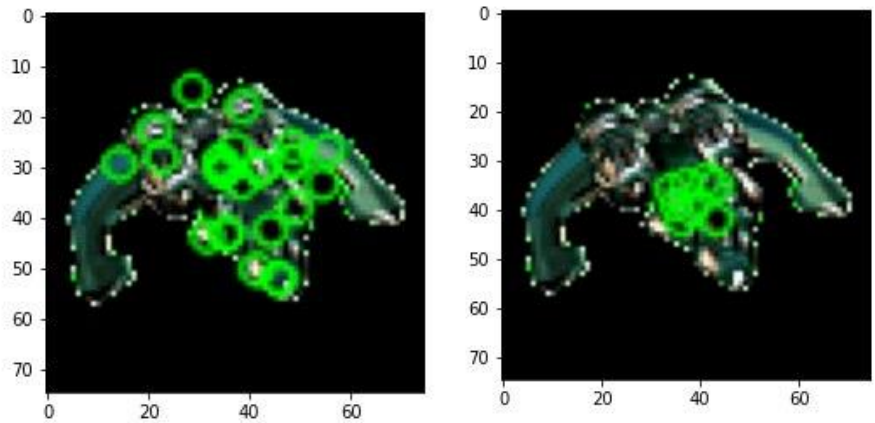

Рис. Визуализация выбранных особых точек алгоритмом SIFT (слева) и ORB (справа)

Исходя из изображения выше визуально кажется, что ORB алгоритм плохо выявляет особые признаки на изображении. Но попробуем обучить модель и посмотреть, правда ли качество упадет.

Количество кластеров	Размер обучающей выборки	Размер тестовой выборки	Accuracy на SIFT (train)	Accuracy на SIFT (test)	Accuracy на ORB (train)	Accuracy на ORB (test)
200	900	763	1.0	0.208	1.0	0.1
	1500	763	1.0	0.218	0.932	0.122
	2238	763	0.999	0.207	0.853	0.155
500	900	763	1.0	0.227	1.0	0.119
	1500	763	1.0	0.223	1.0	0.097
	2238	763	1.0	0.224	1.0	0.121

Действительно, качество модели очень сильно упало. Данный способ выделения особых точек на изображении не подходит для данной задачи.

4.4 Вывод

В результате работы была получена baseline модель с долей верных ответов около 0.239. Baseline модель работала на признаках, полученных с помощью SIFT алгоритма, что оказалось наилучшим выбором (работа на ORB признаках значительно ухудшила результат, что продемонстрировано во второй части проекта). В качестве улучшения был опробован XGBoost, который в свою очередь дал результат ~0.244, побив baseline.

При этом необходимо отметить, что качество работы моделей очень сильно зависит от случайного разбиения данных, и может приводить к разбросу качества, где амплитуда доли верных ответов на тесте может составлять до 10%.

Из этого хотелось бы сделать вывод, что основная трудность данной задачи состоит в очень маленьком количестве данных в исходном датасете и очень большой разнообразности объектов внутри каждого из классов. В качестве возможного дополнительного дальнейшего улучшения представляется сбор дополнительных исходных данных. Например, так как исходные изображения игры представляют собой анимацию, то покадровый сбор этой анимации в исходный датасет. Представляется, что такое улучшение даст значительный прирост в качестве классификации, намного больше чем при выборе другого метода или модели на текущем датасете.

Другой вывод касается самой постановки задачи. Сейчас стоит задача отделять классы WEAPON, SCANER, ENGINE и т.д. друг от друга. Эту задачу можно переформулировать и решать её сначала как классификацию каждой модели в каждом классе, а потом объединять классифицированные модели в соответствующий класс и выдавать на выходе его. В других задачах классификации такой подход обычно наоборот считается усложнением (Допустим, стоит задача отделять штаны от носков. Вместо этого расширим количество классов и будем сначала отделять все возможные цвета носков, а потом объединять их в один класс) и будет давать качество хуже. Однако в этой задаче такой подход поможет решить проблему большого разнообразия объектов внутри одного класса и облегчить классификацию.

5. Заключение

В рамках первого этапа проекта выполнены следующие шаги:

1) Сбор и подготовка данных;
2) Разработано baseline решение для задачи классификации изображений;
3) Разработан план по улучшению текущей модели.

В рамках второго этапа проекта было выполнено:

1) Сравнение разных подходов выделения особых точек на изображении и качества моделей на соответствующих полученных признаках;
2) Применена другая модель (XGBoost), побившая результат baseline;
3) Применен вероятностный подход к аналогичному baseline методу решения задачи, также немного улучшивший результат baseline;
4) Экспериментальным путем определены оптимальные параметры каждой из моделей;
5) Проанализированы дальнейшие возможные пути развития решения задачи.

В результате работы были изучены некоторые существующие способы обработки изображений и извлечения из них признаков для классификации. Был спроектирован и реализован baseline метод для решения поставленной задачи классификации моделей оборудования из видеоигры. Были реализованы и обучены дополнительные модели, улучшающие качество baseline модели.

6. Авторы работы

1. **Баранова Дарья**
   - Аугментация данных
   - Разработка основного baseline подхода
   - Обучение модели на ORB признаках
   - Реализация Calibrated SVM модели
   - Написание отчета

2. **Петренко Денис**
   - Реализация XGBoost

3. **Павлов Сергей**
   - Формулировка исходной задачи
   - Сбор исходных данных

Литература

1) Космические Рейнджеры Вики, ULR: https://rangers.fandom.com/ru/wiki/ResEditor

2) Википедия, Bag-of-words model in computer vision, 2021, URL:https://en.wikipedia.org/wiki/Bag-of-words_model_in_computer_vision

3) PyImageSearch, The bag of (visual) words model, URL: https://customers.pyimagesearch.com/the-bag-of-visual-words-model/

4) Sachin Mohan, Image Classification using Bag of Visual Words Model, 2020, URL:https://machinelearningknowledge.ai/image-classification-using-bag-of-visual-words-model/

5) Deepanshu Tyagi, Introduction To Feature Detection And Matching, 2019, URL:https://medium.com/data-breach/introduction-to-feature-detection-and-matching-65e27179885d

6) Deepanshu Tyagi, Introduction to SIFT (Scale Invariant Feature Transform), 2019, URL:https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40

7) Deepanshu Tyagi, Introduction to ORB (Oriented FAST and Rotated BRIEF), 2019, URL:https://medium.com/data-breach/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf

8) Herbert Bay, Tinne Tuytelaars, Luc Van Gool, SURF: Speeded Up Robust Features, 2006, URL:https://people.ee.ethz.ch/~surf/eccv06.pdf

Приложение

Изображения классифицируемых моделей класса «CARGOHOOK»

Изображения классифицируемых моделей класса «DEFGENERATOR»

Изображения классифицируемых моделей класса «ENGINE»

Изображения классифицируемых моделей класса «FUELTANKS»

Изображения классифицируемых моделей класса «HULL»

Изображения классифицируемых моделей класса «RADAR»

Изображения классифицируемых моделей класса «REPAIRROBOT»

Изображения классифицируемых моделей класса «SCANER»

Изображения классифицируемых моделей класса «WEAPON»