Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве**

**Российской Федерации»**

Департамент анализа данных и машинного обучения

Пояснительная записка к курсовой работе по дисциплине

«Технологии анализа данных и машинного обучения»

на тему:

**«Определение аномалий при анализе видеоинформации»**

Выполнила:

студентка ПИ20-3

Хертек Я.О.

Научный руководитель:

доцент, канд. физ.-мат. наук

Попов В.Г.

**Москва**

**2023**

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc135142804)

[Цель 3](#_Toc135142805)

[Задачи 3](#_Toc135142806)

[ОCНОВНАЯ ЧАСТЬ 5](#_Toc135142807)

[Обзор теоретического материала 5](#_Toc135142808)

[Понятие глубокого обучения и его роль в компьютерном зрении 5](#_Toc135142809)

[Особенности работы с видеоинформацией 7](#_Toc135142810)

[Методология исследования 9](#_Toc135142811)

[Сбор и описание используемых данных 9](#_Toc135142812)

[Настройка гиперпараметров 12](#_Toc135142813)

[Загрузка и анализ 13](#_Toc135142814)

[Препроцессинг 16](#_Toc135142815)

[Разработка моделей глубокого обучения для определения аномалий 20](#_Toc135142816)

[Сравнение. Выбор архитектуры нейронной сети 24](#_Toc135142817)

[Прогнозирование 28](#_Toc135142818)

[ЗАКЛЮЧЕНИЕ 30](#_Toc135142819)

[СПИСОК ЛИТЕРАТУРЫ 30](#_Toc135142820)

# ВВЕДЕНИЕ

Каждый день, каждую минуту в жизни современных людей происходит бесчисленное количество событий, и все они различаются по частоте повторения: какие-то могут возникать более сотен раз в сутки, а какие-то - всего 1 раз. Последние, зачастую, представляют особый интерес для исследования.

Любое событие, которое отличается от обычного или ожидаемого, а также редко наблюдается в привычном паттерне жизни, называют аномалией (Anomaly). Например, это может быть необычное поведение человека, необычный звук или запах, необычное движение транспорта или другие события, которые могут вызвать подозрения или опасения. Такие аномальные действия склонны к предпосылкам к непредсказуемым, а порой, и тяжким последствиям - преступлениям, которым уделяют отдельное внимание. Они являются одними из самых распространенных и опасных аномальных явлений в обществе. К ним относятся кражи, нападения, убийства и т.д. Все эти преступления противоречат установленным нормам поведения и могут причинить серьезный вред как конкретному человеку, так и всему обществу в целом. В связи с этим, возникает необходимость в их фиксации на камеру, чтобы иметь доказательства их происхождения и виновности.

Однако, стоит заметить, что человеческое отслеживание видеоинформации не всегда является эффективным, так как наблюдатель может упустить некоторые детали или не заметить преступление вовсе. В связи с этим, возникает необходимость использования инструментов цифровых технологий, таких как машинное обучение или искусственный интеллект (нейронные сети для распознавания преступлений на видео).

Сегодня методы машинного обучения применяются во всех сферах, где требуется классификация, предсказание или обобщение данных на основе определенных критериев. Это включает распознавание речи, обработку естественных языков, медицинскую диагностику, интеллектуальные игры и многое другое. Однако наиболее ресурсоемким направлением является компьютерное зрение (Computer Vision), которое включает в себя задачи многоклассового распознавания объектов на видеозаписях и фотографиях.

Выявление различных аномалий на видео является одной из актуальных задач в области компьютерного зрения. Она заключается в автоматическом обнаружении необычных событий или объектов на видео, которые могут указывать на наличие проблем или угроз для безопасности. Одним из примеров применения распознавания аномалий при анализе видеоинформации является система безопасности для общественных мест, таких как аэропорты, торговые центры и стадионы. Эти системы могут автоматически обнаруживать подозрительные объекты или поведение, такие как оставленные сумки или движение в запрещенных зонах. Другим примером является использование распознавания аномалий в производственных процессах, где системы могут обнаруживать необычные события, такие как прекращение работы оборудования или нарушение технологических параметров производства.

### Цель

Целью данной работы явилась разработка и реализация методов распознавания различных аномалий на фреймах видеоданных датасета Anomaly-Detection-Dataset-UCF при помощи алгоритмов глубокого машинного обучения (CNN, DenseNet121). В рамках работы было необходимо научиться автоматически определять аномалии (вид преступления) на каком-либо кадре на основе предоставленных данных.

### Задачи

Для достижения выше поставленной цели, были выполнены следующие задачи:

* выбор необходимых сущностей датасета и анализ полученного набора данных;
* выбор оптимальных методов обработки и классификации информации;
* оценка эффективности разработанных алгоритмов, сравнение и выбор лучшего метода.

В качестве командного интерпертатора был использован Python с интерактивным ноутбуком Jupyter Notebook. Для препроцессинга задействуется библиотека keras.preprocessing и инструменты аугментации ImageDataGenerator. При работе с глубокими нейронными сетями– Keras.

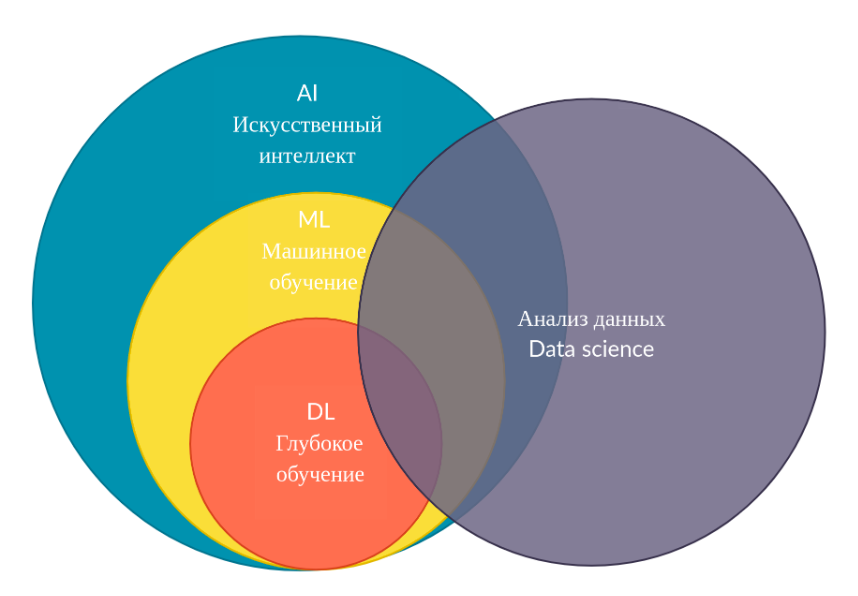
# ОCНОВНАЯ ЧАСТЬ

## Обзор теоретического материала

### Понятие глубокого обучения и его роль в компьютерном зрении

Машинное обучение – это «численная оптимизация параметрических моделей для описания определенного набора данных» [1]; подмножество искусственного интеллекта, которое связано с созданием алгоритмов, у которых есть возможность изменения своих свойств без ручного регулирования человеком для получения желаемого результата - путем подачи себя через структурированные данные.

Глубокое обучение – подмножество машинного обучения, где есть несколько уровней этих алгоритмов, каждый из которых обеспечивает различную интерпретацию данных, которые он передает. Такая модель алгоритмов называется искусственными нейронными сетями (ИНС).



Риc.1. Иерархия сущностей искусственного интеллекта [1]

Существует три различных подхода к обучению алгоритма обнаружения аномалий:

* 1. с учителем: этот подход требует вмешательства человека в том смысле, что нам нужно предоставить алгоритму метку для каждого набора данных до процесса обучения. Метка обычно состоит из двух возможных значений, например, нормальные точки данных будут помечены как 1, а аномальные точки данных будут помечены как 0 или 1.
  2. без учителя: этот подход не требует вмешательства человека, потому что нам не нужно предоставлять истинную метку каждой точки данных алгоритму до процесса обучения. Другими словами, алгоритм будет самостоятельно изучать паттерн точек данных во время процесса обучения. Затем, на основе плотности каждой точки данных, алгоритм решит, является ли точка данных выбросом или нет.
  3. смешанный (semi-supervized): этот подход является комбинацией двух предыдущих подходов. Это означает, что помечены только небольшие части данных. Эта техника полезна в реальной жизни, потому что почти невозможно предоставить метку каждой точке набора, учитывая огромный размер реальных данных. [2]

Глубокое обучение играет важную роль в компьютерном зрении, так как позволяет автоматически изучать признаки изображений и видео, используя множество слоев, которые могут обнаруживать различные уровни абстракции в изображении. Это позволяет создавать более точные и универсальные модели для решения задач компьютерного зрения, таких как распознавание объектов, сегментация изображений, детектирование лиц и распознавание жестов. Например, Deep Anomaly Detection (DAD, Глубокое обучение обнаружения аномалий) позволяет разрешить ряд ограничений, связанных с:

* неизвестностью новых данных (способность алгоритма генерализовывать информацию об объекте с уже имеющейся информацией из ранее полученных данных и выявлять различия, если имеются),
* гетерогенностью разных классов,
* редкостью появления аномалий (дисбаланс классов),
* различными типами аномалий. [3]

Однако, с точки зрения машинного обучения, задача обнаружения аномалий также имеет свои трудности. Например, низкие значения precision метрики в задаче классификации на аномальный / нормальный, проблема больших размерностей данных, отсутствие или недостаток размеченных данных, неустойчивость алгоритмов к зашумленным объектам, детекция аномалий целой группы объектов и низкая интерпретируемость результатов. [3]

Под алгоритмами глубокого обучения понимаются искусственные нейронные сети, которые входят только в глубокое обучение. Их создание будет происходить при помощи библиотеки Keras без использования ресурсов видеокарты GPU, так как на macOS была прекращена поддержка

### Особенности работы с видеоинформацией

Как правило, в общих случаях для обучения модели обнаружения аномалий на видео, видеопоток необходимо преобразовать в набор изображений, которые затем могут быть использованы для обучения модели. Это может быть выполнено путем разбиения видео на кадры (frame), каждый из которых будет представлен в виде отдельного изображения. Если же стоит определенная задача обработки видеоинформации, то она осуществляется на специализированных компьютерах или на других вычислительно мощных аппаратных средствах с оптимизацией процесса обработки. [4]

При работе с видеоинформацией в задачах компьютерного зрения необходимо учитывать ряд особенностей, связанных с предобработкой и загрузкой набора данных:

1. Большой объем данных: видеофайлы обычно имеют большой размер, что затрудняет их обработку и анализ.
2. Пространственно-временная структура: видео состоит из последовательности кадров, которые представляют пространственную и временную информацию о сцене.
3. Разнообразие объектов и сцен: видео может содержать множество различных объектов и сцен, что усложняет задачу анализа аномалий
4. Разрешение видео. Разрешение видео может существенно влиять на производительность алгоритмов машинного обучения. Например, при обработке видеопотока в режиме реального времени может потребоваться снижение разрешения для ускорения обработки. При этом необходимо учитывать, что слишком низкое разрешение может привести к потере важных деталей.
5. Наличие аудиодорожки. Видео может содержать аудиодорожку, которая может быть использована для улучшения точности алгоритмов машинного обучения. При работе с видеоинформацией необходимо учитывать наличие аудиодорожки и возможность ее использования. [5]

Часть этих характеристик также применима к кадрам. Отличительной чертой при обработке изображений является многофакторность – для изображения пикселей в RGB формате существует входных аргумента. Именно из-за этой особенности все операции для машинного обучения изображений обычно осуществляются не на CPU, где акцент смещен на оперативность обработки данных, а на GPU с массовым параллелизмом задач по обработке изображений ввиду архитектуры (рис. 2).

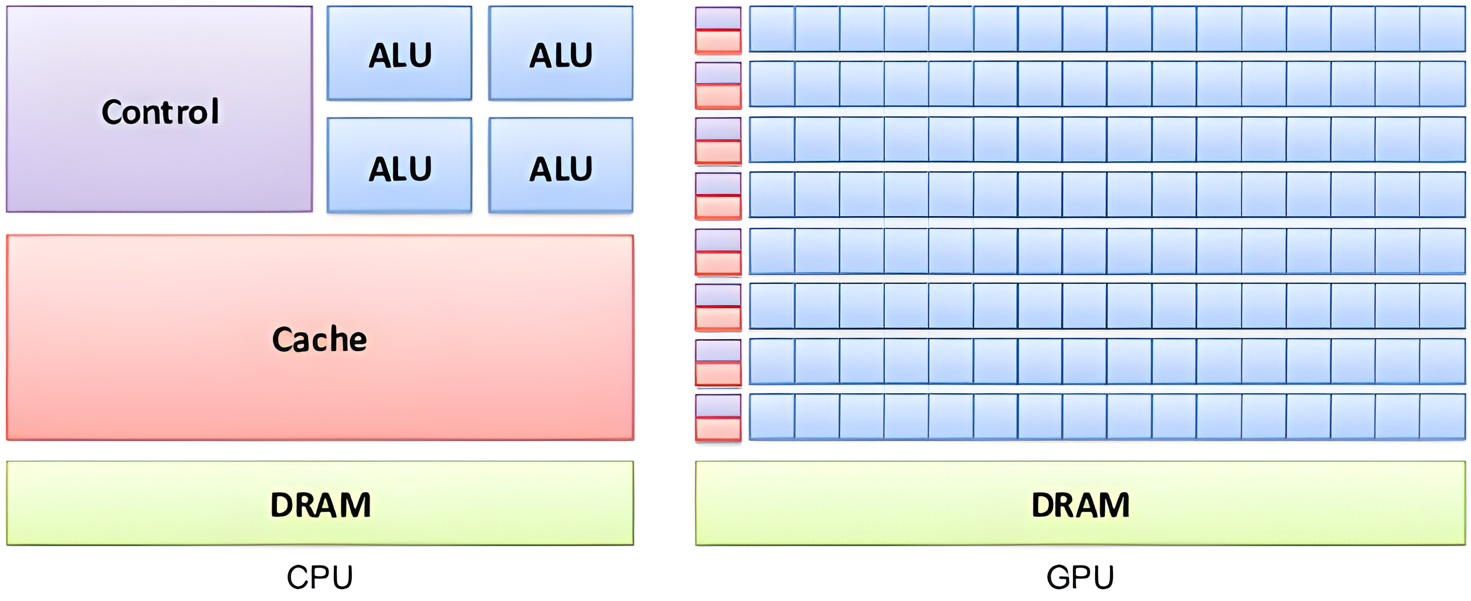


Рис. 2. Визуальное различие архитектур CPU и GPU.

Для работы с GPU разных производителей есть несколько наиболее популярных программно-аппаратных архитектур параллельных вычислений:

* CUDA – решение от NVIDIA, позволяющая производить вычисления только на видеокартах данного производителя. Проприетарная разработка, свой собственный компилятор, можно предкомпилировать под разные версии API. Существует несколько версий API для работы с разными поколениями видеокарт. На macOS поддержка прекращена.
* OpenCL – открытый фреймворк для параллельной обработки данных на видеоускорителях. Поддерживаются дискретные видеокарты от AMD, NVIDIA, а также встроенные в Intel, AMD и Apple. Возможности предкомпиляции нет

Данная работа выполнялась на компьютере MacBook Air без поддержки GPU, в связи с чем было предпринято несколько ограничительных мер в разработке алгоритма.

## Методология исследования

### Сбор и описание используемых данных

Вид и качество исходных наборов данных напрямую влияет на точность классификации моделей искусственного интеллекта, особенно для компьютерного зрения в задачах работы с видеоинформацией, которая чаще всего требует значительных ресурсов памяти и может хранить в себе несколько сотен тысяч разных по длине и качеству медиа. Согласно проведенному в 2014 году исследованию компании CISCO – «к 2019 году потребуется более 5 миллионов лет, чтобы просмотреть в режиме реального времени весь объем видеофайлов с IP-камер, отснятых лишь за месяц.» [6][7]

Данная работа основывается на датасете Anomaly-Detection-Dataset-UCF, размещенный на открытом интернет-ресурсе Kaggle. Стоить сразу отметить, что он не пригоден для полной загрузки (размер достигает 50 Гб), а для использования необходима лишь часть данных. Другими словами, он содержит в себе большое количество избыточных материалов: помимо разделенной на директории с названиями классов видеоинформации (колоссального размера), так же хранятся директории с текстовыми файлами. Они, очевидно, были сгенерированы авторами в процессе обучения собственной модели, которая так же присутствует среди файлов (рис. 3).

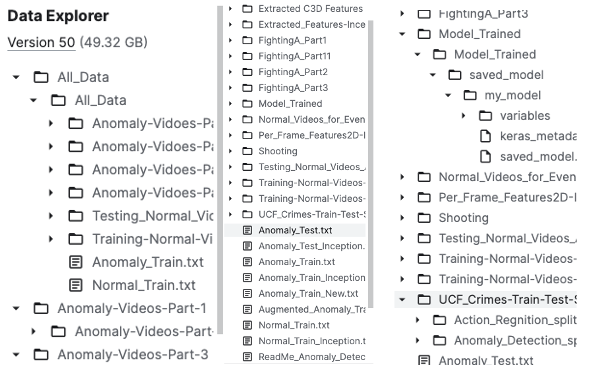


Рис. 3. Структура датасета Anomaly-Detection-Dataset-UCF.

В связи с более оптимальным использованием ресурсов было принято решение взять набор данных, на основе которого был сформирован Anomaly-Detection-Dataset-UCF - первоисходный UCF Crime, а точнее датасет с уже преобразованными в изображения (фреймы) видеофайлами.

UCF Crime – это крупномасштабный набор данных, разработанный для оценки алгоритмов, связанных с обнаружением аномалий. Датасет содержит 1900 длинных необрезанных видео наблюдения, охватывающие 13 реальных сценариев (аномальных – преступлений) и 1 класс с нормальными видеоматериалами, на которых не происходит противоправных явлений. Эти аномалии выбраны потому, что они оказывают значительное влияние на общественную безопасность. (рис. 4) [8][9]

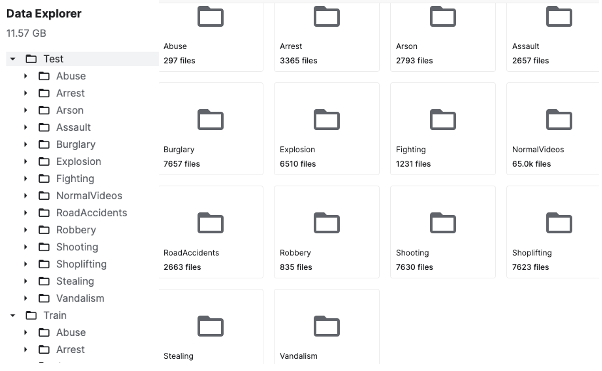


Рис. 4. Структура UCF Crime (с изображениями) на веб-сайте Kaggle.

Говоря о генерации такого типа набора данных, авторы - исследовательская группа UCF – в своем подходе рассматривают нормальные и аномальные видео с камер наблюдения как мешки (bags), а сегменты видео как экземпляры (instances). Используя как положительные (аномальные), так и отрицательные (нормальные) мешки, была обучена модель обнаружения аномалий с помощью предложенной глубокой функции потерь MIL ранжирования.

То есть для положительных (содержащих в каком-то моменте аномалию) и отрицательных (не содержащих аномалию) видео – каждое из них было поделено на несколько временных сегментов. Затем каждое видео представлено в виде мешка, а каждый временной сегмент – в виде экземпляра в мешке. После извлечения функций C3D для видео-сегментов, авторы обучили полностью связанную нейронную сеть, используя новую функцию потерь ранжирования, которая вычисляет потери между наивысшими оцененными экземплярами (показанными красным цветом) в положительном мешке и отрицательном (рис. 5). [8]

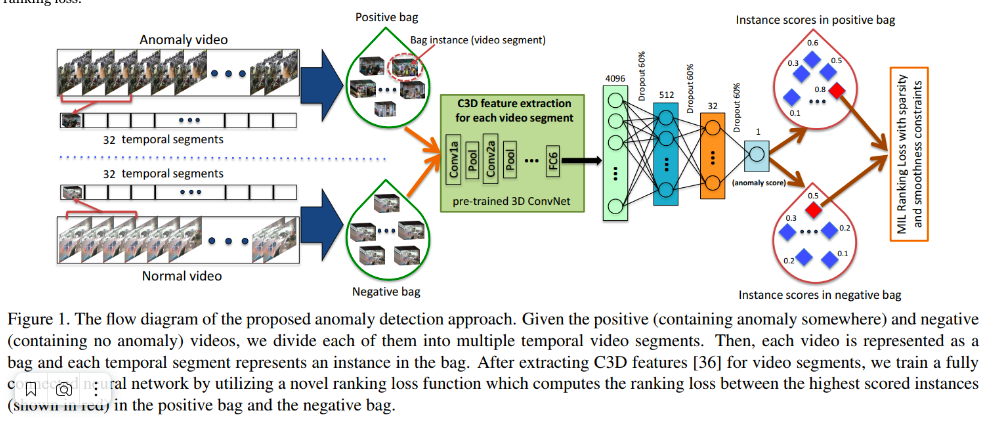


Рис. 5. Визуальное представление метода выявления аномалий из центра исследований UCF. [8]

UCF Crime/Anomaly Detection Dataset может быть использован для решения двух задач:

1. общее обнаружение аномалий с учетом всех аномалий в одной группе и всех нормальных действий в другой группе.
2. для распознавания каждой из 13 аномальных активностей.

Данная работа будет заниматься решением второго пункта – определением категории-аномалии каждой точки датасета.

Набор данных UCF Crime (кадры) с уже предобработанными в изображения видеофайлами (далее UCF Crime) разделен на тестовую и обучающую выборки, в каждой из которой находятся материалы по 14-ти категориям (13 аномальных и 1 нормальная) (рис. 6):

1. Жестокое обращение (Abuse)
2. Арест (Arrest)
3. Поджог (Arson)
4. Нападение (Assault)
5. Дорожно-транспортное происшествие (Road Accident)
6. Кража со взломом (Burglary)
7. Взрыв (Explosion)
8. Драка (Fighting)
9. Ограбление (Robbery)
10. Стрельба (Shooting)
11. Кража (Stealing)
12. Магазинная кража (Shoplifting)
13. Вандализм (Vandalism)
14. Нормальные видео (NormalVideos)

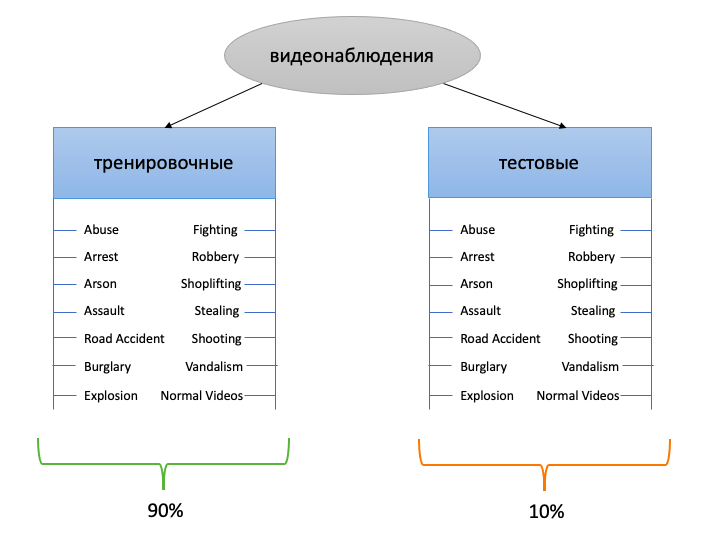


Рис. 6. Структура UCF Crime.

### Настройка гиперпараметров

Перед началом всей работы с данными и нейронными сетями следует определить неизменные значения. Они пригодятся в предобработке и разработке архитектуры моделей обучения (рис. 7).

1. Директории с изображениями: Train и Test – обе находятся в корневой папке data.
2. SEED – это число, которое используется для начальной настройки весов и смещений в нейронной сети. Если использовать одно и то же число при каждом запуске программы, то получится одинаковый результат, что может помочь улучшить качество обучения.
3. IMG\_HEIGHT = 224, IMG\_WIDTH = 224, IMG\_SHAPE, INPUT\_SHAPE=(IMG\_HEIGHT, IMG\_WIDTH,3) – параметры формирования вектора изображения (длина, ширина, размерность картинки и подаваемое на вход моделям INPUT\_SHAPE).
4. Batch\_size – это количество примеров из обучающей выборки, которые используются для обновления весов в нейронной сети за одну итерацию обучения. Чем больше batch size, тем быстрее происходит обучение, но при этом увеличивается потребление памяти.
5. Epochs – это количество раз, за которое обучающая выборка проходит через нейронную сеть в процессе обучения. Каждая эпоха состоит из нескольких итераций, где веса обновляются на основе batch size примеров. Увеличение числа эпох может улучшить качество обучения, но может также привести к переобучению модели.

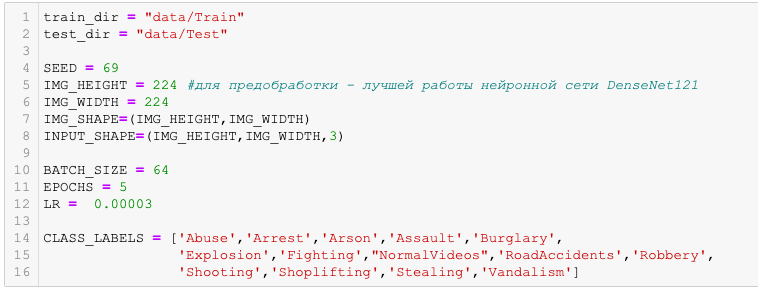


Рис. 7. Настройка гиперпараметров.

### Загрузка и анализ

Для начала работы с моделями и их обучением необходимо загрузить и проанализировать датасет. Так как данные предоставлены в виде файлов, распределенных по директориям, загрузить их и отобразить на экране не требует особо труда – достаточно воспользоваться библиотекой os и вызвать инструмент просмотра директорий listdir.

Напишем функцию для просмотра нескольких случайных картинок на экран и вызовем ее (рис. 8, 9).

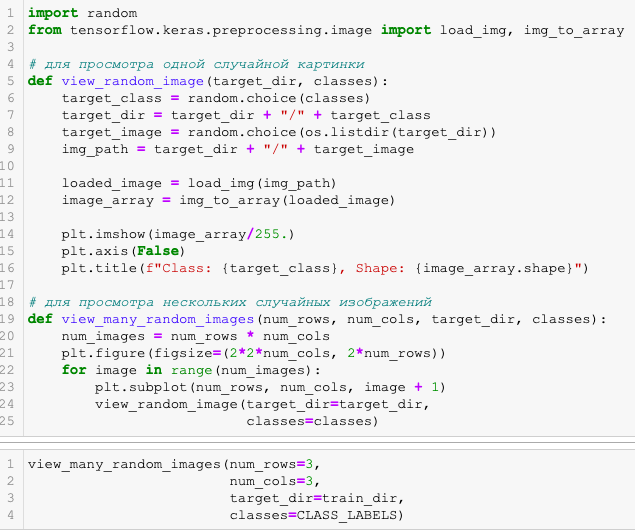


Рис. 8. Функция для вывода изображений.

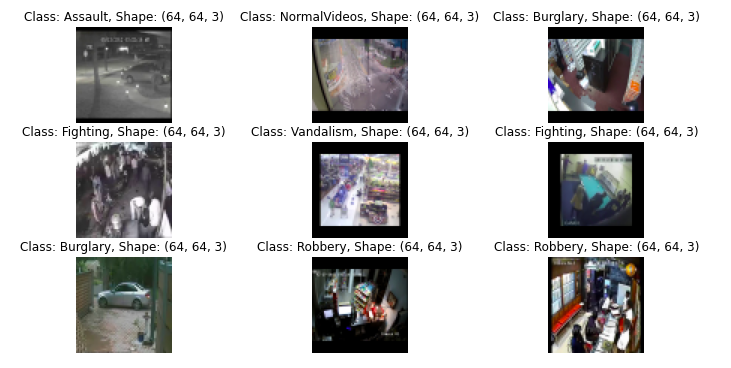


Рис. 9. Результат - 9 случайных изображений из набора данных.

Теперь проанализируем данные. Выясним, каково разделение данных во всем датасете, включая тестовую и обучающую выборки вместе (рис. 10).

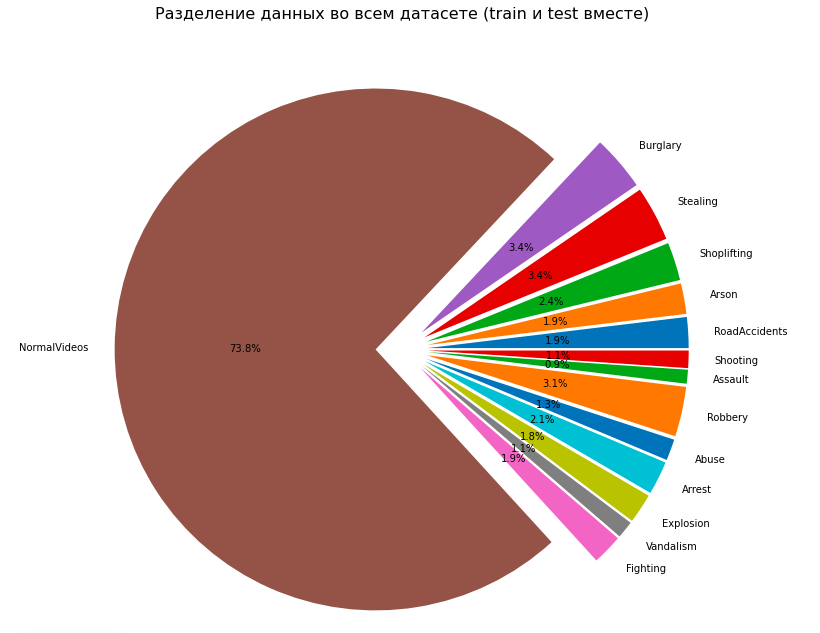


Рис. 10. Распределение данных во всем датасете.

Больше всего данных в категориях:

1. Normal / Нормальные - 73.8%
   1. Burglary / Кража со взломом - 3.4%
   2. Stealing / Кража - 3.4%
2. Robbery / Ограбление - 3.1%

Меньше всего данных в категориях:

1. Assault / Нападение - 0.9%
   1. Shooting / Стрельба - 1.1%
   2. Vandalism / Вандализм - 1.1%
2. Abuse / Насилие - 1.3%



Рис. 11. Соотношение обучающей и тестовой выборок.

Данные поделены на обучающую и тестовую выборки в соотношении ~90/10% соответственно (рис. 11).

### Препроцессинг

Во-первых, можно заметить, что работа проводится с колоссально огромным для процессора массивом данных, и, учитывая, что нейронные сети прогоняют их по несколько эпох, получаем неимоверно долгое время обработки и обучения. В таком случае было решено использовать лишь часть (0,2%) набора данных для демонстрации, но в дальнейшем код можно перезапустить на полном датасете.

Для этого мы берем случайные 0,002 доли изображений с каждой директории/категории каждой выборки (директории Train/Test) и получаем две новые папки TrainSample\_002 и TestSample002 с частью 0,2% оригинальных данных (рис. 12, 13, 14, 15).

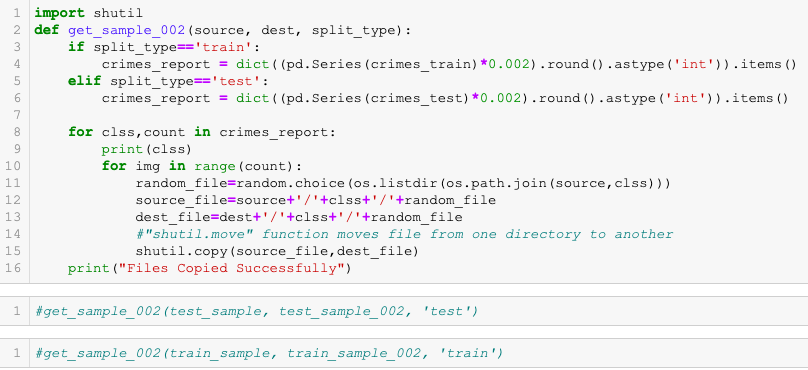


Рис. 12. Функция получения доли набора данных.

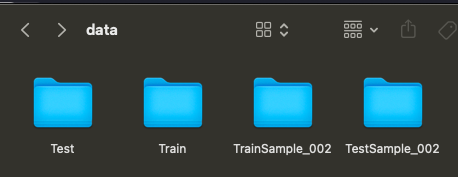


Рис. 13. Результат.

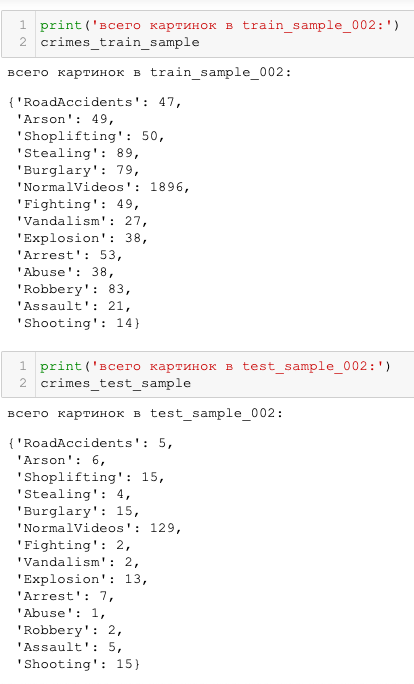
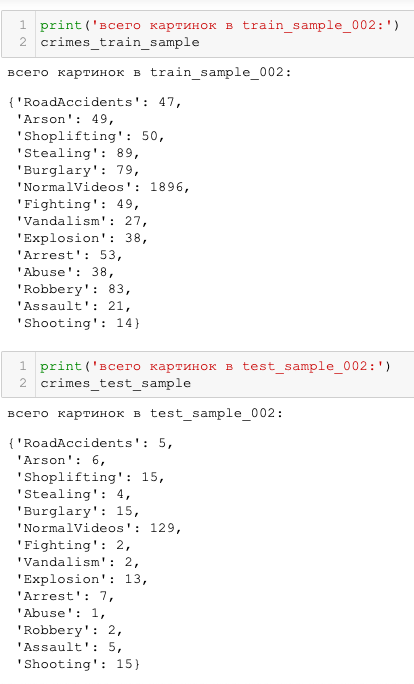


Рис. 14. Количество новых данных.



Рис. 15. Разделение точек.

Во-вторых, если мы работаем с достаточно объемным массивом изображений (здесь: ~15 Гб), загрузка всего набора данных в одну переменную Python не является вариантом, поэтому нам нужна функция-генератор.

Генераторная функция похожа на обычную функцию Python (def), но работает как итератор. Она имеет специальное ключевое слово yield, которое похоже на return, так как возвращает некоторое значение. Когда генератор вызывается, он возвращает некоторое значение и сохраняет состояние. В следующий раз, когда мы снова вызываем генератор, он продолжит работу с сохраненного состояния и вернет следующий набор значений, как итератор. [10]

Такая функция позволяет проходить по каждому (или по партиям) изображению в большом наборе данных и легко обучать нейронную сеть. Но здесь возникает проблема: размер набора данных большой, а вариации (точки набора) невелики. Например, в датасете есть изображения размером ~800 Кб, но количество их мало - 500). Поэтому нам нужно реализовать методы аугментации изображений, чтобы искусственно увеличить вариативность нашего набора данных.

Аугментация/Техники увеличения изображений (Image Augmentations) – это методы искусственного увеличения вариаций изображений в нашем наборе данных путем использования горизонтальных/вертикальных переворотов, поворотов, изменений яркости изображений, горизонтальных/вертикальных сдвигов и т. д. [10] В Python такой удобный функционал предоставляет библиотека keras – мы можем легко использовать инструмент ImageDataGenerator (класс) для увеличения нашего набора данных на порядок и не писать вручную все методы обработки изображений.

Преимущества использования класса ImageDataGenerator:

* Простота написания – необходимо просто вызвать keras.preprocessing.image.ImageDataGenerator() и установить значения для различных параметров, таких как horizontal\_flip, vertical\_flip, rescale, brightness\_range, zoom\_range, rotation\_range и т. д.
* Меньше нужно запоминать – не нужно вручную кодировать техники обработки изображений из различных библиотек для переворачивания, изменения яркости, масштабирования и т. д.

Проще говоря, для более точной работы с изображениями, нам важно учитывать каждое положение, размер и цвет пикселей для каждого кадра, поэтому, применив аугментацию, мы получаем по несколько различных предобработанных версий одной картинки в датасете. К тому же, генератор-функция ImageDataGenerator позволяет это сделать всего за пару строк кода.

Однако, наиболее полезным инструментом этого класса является метод flow\_from\_directory, который автоматически загружает изображения из указанного каталога. Он должен быть каталогом верхнего уровня, где все подкаталоги самостоятельных классов могут храниться отдельно друг от друга. Метод flow\_from\_directory автоматически обрабатывает все подкаталоги и получает изображения вместе с соответствующими метками (рис. 16).

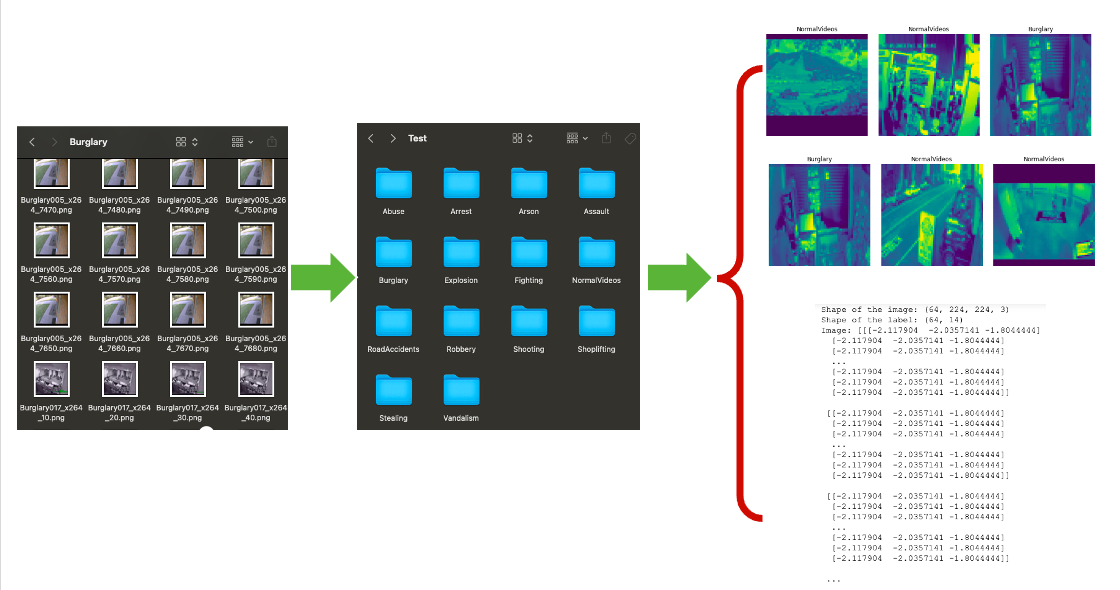


Рис. 16. Функция-генератор для изображений – концептуально.

Создание кода максимально простое: для начала объявляем экземпляр генератора ImageDataGenerator с необходимыми параметрами предобработки, в данном случае (рис. 17):

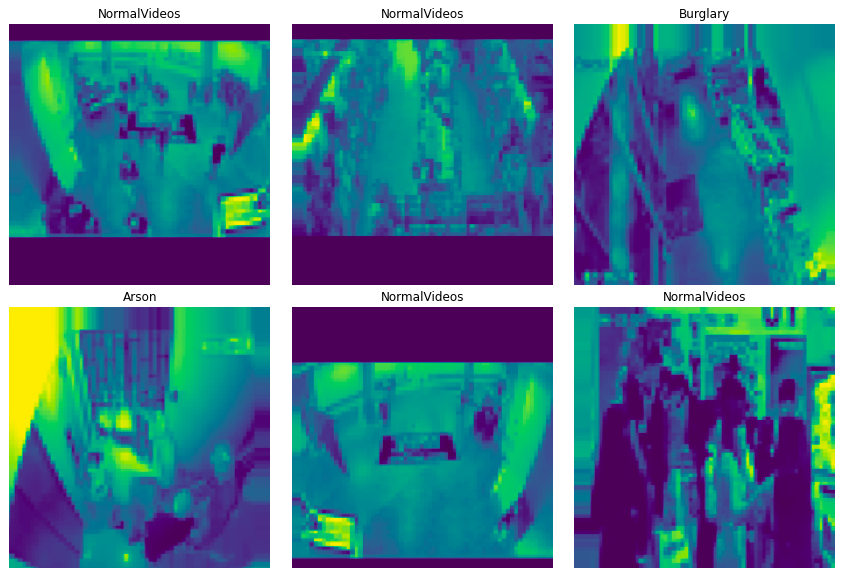
* featurewise\_center: Boolean. Если True, то выводит контур изображения усредняя данные по всему набору.
* featurewise\_std\_normalization: Boolean. Если True, то делит каждый элемент входного набора данных на сренеквадратическое отклонение по всему набору.
* horizontal\_flip: Произвольное переворачивание входных данных по горизонтали.
* width\_shift\_range: Float, 1D-массивоподобный объект или int. Задает случайный сдвиг по ширине.
* height\_shift\_range: Float, 1D-массивоподобный объект или int. Задает случайный сдвиг по высоте.
* preprocessing\_function: функция, которая будет применена к каждому изображению. Функция выполняется после всех преобразований. [11]

После получаем объект типа dataset при вызове метода .flow\_from\_directory(), которая берет изображения из directory и применяет заданные нами параметры.



Рис. 17. Создание датасетов.

Результат предобработки – изображения прошкалированы, переведены в необходимый цвет, нормализованы, изменено их положение:



### Разработка моделей глубокого обучения для определения аномалий

#### Сверточная нейронная сеть CNN

Сверточная нейронная сеть CNN – это один из видов искусственных нейронных сетей. Нацеленная на эффективное распознавание образов, входит в состав технологий глубокого обучения и наиболее предпочтительна на сегодняшний день.

CNN улавливает пространственные особенности изображения. Пространственные особенности относятся к расположению пикселей и взаимосвязи между ними на изображении. Они помогают точно идентифицировать объект, определить его местоположение, а также его связь с другими объектами на изображении. CNN также следует концепции разделения параметров. Один фильтр применяется к различным частям входного сигнала для создания карты характеристик. [15]

С помощью библиотеки Keras была спроектирована следующая сверточная нейронная сеть (рис. 18):

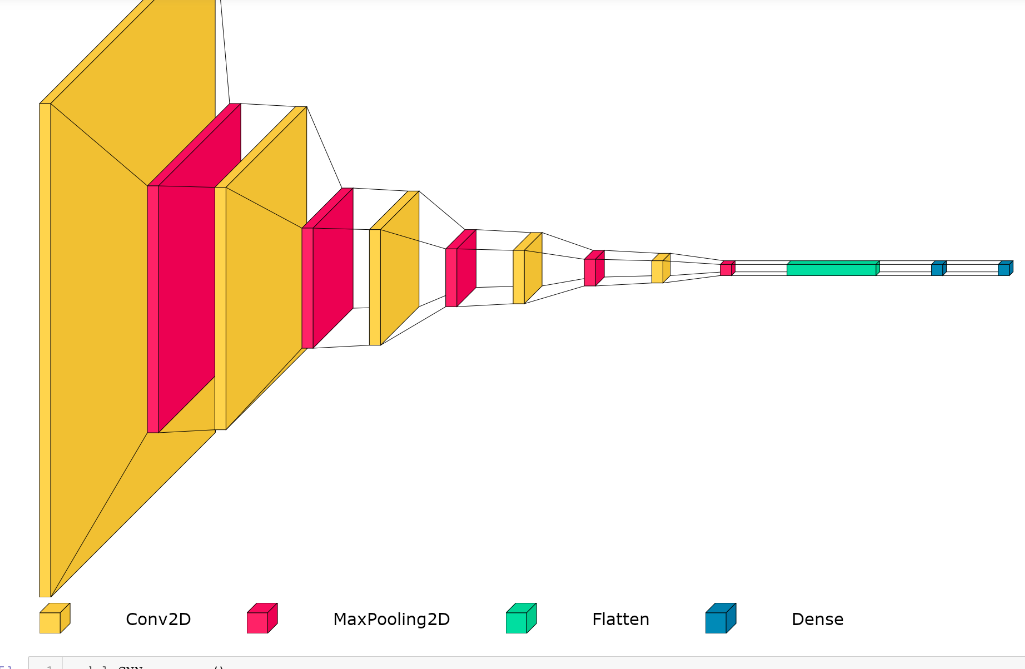


Рис. 18. Архитектура разработанной классической сверточной нейронной сети.

Данная сеть состоит из пяти блоков свертки и пулинга, а также двух полносвязных слоев:

1. Первый блок свертки состоит из 32 фильтров размера 3х3 и функции активации ReLU. Входным изображением должно быть изображение размером 224х224 пикселя и 3 канала (RGB). После свертки применяется слой пулинга размера 2х2.
2. Следующие четыре блока свертки и пулинга имеют аналогичную структуру, но количество фильтров увеличивается до 64 в последнем блоке.
3. После последнего пулинга результаты свертки преобразуются в одномерный массив с помощью слоя Flatten().
4. Затем следует полносвязный слой с 128 нейронами и функцией активации ReLU. В конце модели используется полносвязный слой с количеством нейронов, соответствующим количеству классов, и функцией активации softmax, которая преобразует выходы нейронов в вероятности принадлежности к каждому классу.

Эта модель является достаточно эффективной для обработки изображений, так как она использует сверточные слои для извлечения признаков из входных изображений и пулинг для уменьшения размерности. Также использование полносвязных слоев позволяет модели классифицировать изображения на различные классы. [15]

#### DenseNet121

DenseNet121 – это глубокая сверточная нейронная сеть, предназначенная для решения задач классификации изображений. Она состоит из 121 слоя и использует блоки DenseNet для связывания слоев. Блок DenseNet – это структура, в которой каждый слой получает на вход не только выход предыдущего слоя, но и выход всех предыдущих слоев. Это позволяет увеличить количество параметров и улучшить качество обучения. DenseNet121 имеет высокую точность классификации и хорошую устойчивость к изменениям входных данных. Она была обучена на большом наборе данных ImageNet и может быть использована для решения различных задач классификации изображений, таких как распознавание объектов на фотографиях или диагностика медицинских изображений. [16]

Реализуем нейронную сеть на основе предобученной модели DenseNet121 из Keras (рис. 19, 20):



Рис. 19. Реализация модели на основе предобученной DenseNet121.

Эта модель состоит из трех основных компонентов: feature\_extractor, classifier и final\_model.

Компонент feature\_extractor использует предобученную модель DenseNet121 из библиотеки Keras, которая содержит сверточные слои для извлечения признаков из входных изображений. Эти сверточные слои обучены на большом наборе изображений и могут эффективно извлекать признаки из различных классов изображений.

Компонент classifier использует полносвязные слои для классификации изображений на различные классы. Сначала используется слой GlobalAveragePooling2D для усреднения признаков, извлеченных из feature\_extractor. Затем следует несколько полносвязных слоев с функцией активации ReLU и слоями Dropout для предотвращения переобучения модели.

Компонент final\_model объединяет feature\_extractor и classifier в единую модель. Он принимает на вход изображения и выводит вероятности принадлежности к каждому классу.

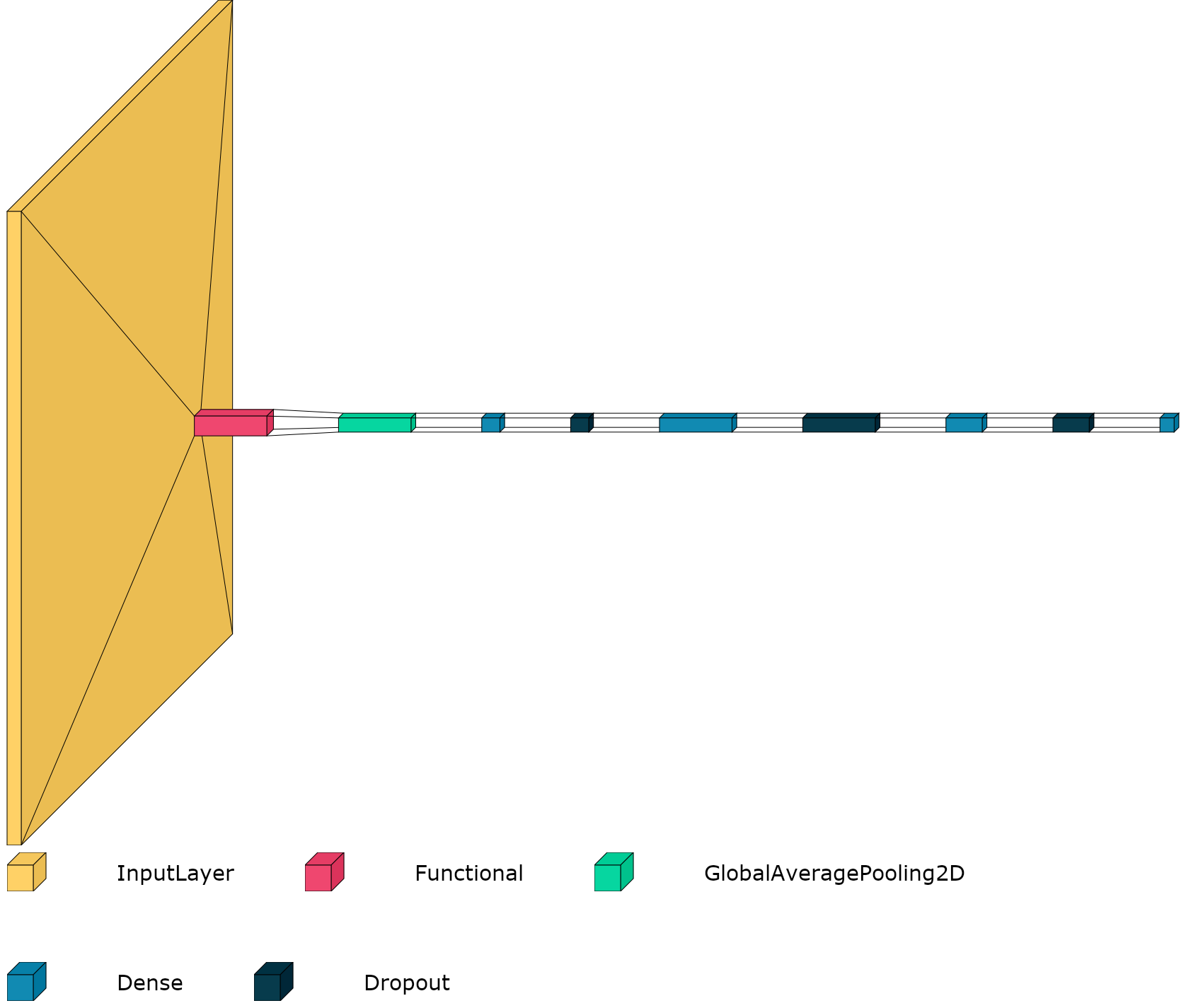


Рис. 20. Архитектура разработанной DenseNet121.

В целом, эта модель является эффективной для обработки изображений, так как она использует сверточные слои для извлечения признаков и полносвязные слои для классификации. Однако эффективность модели может зависеть сильно от качества и количества тренировочных данных: чем больше и лучше качество данных, тем лучше будет производительность модели.

Эта архитектура также была использована для обнаружения аномалий в видеонаблюдении, где вызов состоит в том, чтобы точно обнаруживать аномальные события, минимизируя ложные срабатывания [12]. DenseNet121 показала хорошую производительность в этой задаче, но одним из вызовов является правильная обработка изображений и видео данных и обеспечение высококачественной обработки и точных результатов с помощью соответствующих методов и технологий. [16]

### Сравнение. Выбор архитектуры нейронной сети

Определим показатели, по которым будем выявлять эффективность каждой модели: loss, accuracy и auc.

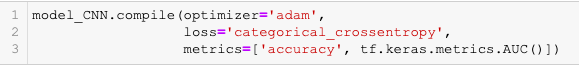


Рис. 21. Параметры модели CNN

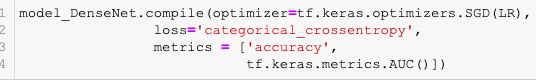


Рис. 22. Параметры модели DenseNet121

В нейронных сетях функция потерь (loss function) представляет собой численную меру расхождения между предсказанными значениями модели и фактическими значениями целевой переменной. Цель обучения нейронной сети - минимизировать значение функции потерь, чтобы сделать предсказания как можно более точными.

Функция потерь выбирается в зависимости от задачи, которую решает модель. Например, в задаче классификации обычно используют категориальную кросс-энтропию (categorical cross-entropy), а в задаче регрессии - среднеквадратичную ошибку (mean squared error).

Оптимизация функции потерь происходит с помощью градиентного спуска (gradient descent) или ее модификаций, таких как стохастический градиентный спуск (stochastic gradient descent).

Чем меньше значение loss, тем лучше производительность модели.

Accuracy (точность) – это метрика качества модели, которая измеряет долю правильных ответов, полученных моделью на тестовых данных. Это одна из наиболее распространенных метрик в машинном обучении и используется для оценки эффективности модели.

Формула точности выглядит следующим образом:

Accuracy = (количество правильных ответов) / (общее количество ответов),

где "количество правильных ответов" – это количество примеров, на которых модель дала правильный ответ (y\_pred=y\_true), а "общее количество ответов" – это общее количество примеров, на которых модель была протестирована.

Чем выше значение accuracy, тем лучше производительность модели. [13]

Метрика AUC (Area Under the Curve) является показателем качества классификации и часто используется для оценки эффективности нейронных сетей. AUC вычисляется путем построения кривой ROC (Receiver Operating Characteristic), которая показывает, как хорошо модель различает между собой классы. Затем AUC определяется как площадь под этой кривой. Чем выше значение AUC, тем лучше производительность модели.

AUC является особенно полезной метрикой при дисбалансе классов, что означает, что число объектов в одном классе значительно превышает число объектов в другом. В этом случае точность и полнота могут дать искаженную картину производительности модели, поэтому AUC может быть более информативной метрикой. [14]

AUC показатель будем использовать для конечной модели, построив ROC-кривую.

Итак, обучив обе модели на sample данных (0,02% от исходных данных), получаем следующие значения метрик:

1. CNN
   1. loss 0.993 (низкое значение – хороший результат) (рис. 23);

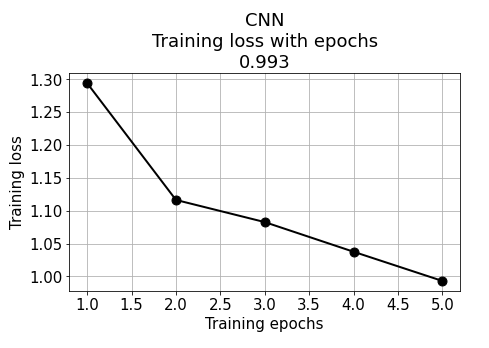


Рис. 23. CNN – динамика значения loss на протяжении эпох при обучении.

* 1. accuracy 0.756 (достаточно высокое значение – хороший результат (рис. 24);

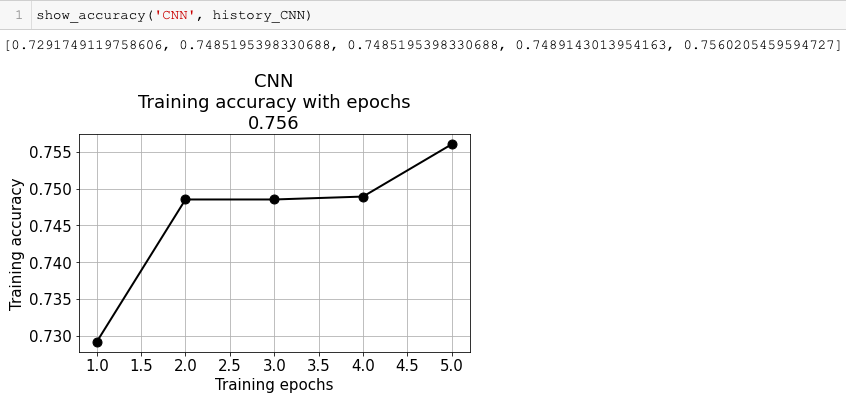


Рис. 24. CNN – динамика метрики accuracy на протяжении эпох при обучении.

1. DenseNet121
   1. loss 2.564 (значение выше, чем на CNN – неудовлетворительный результат) (рис. 25);

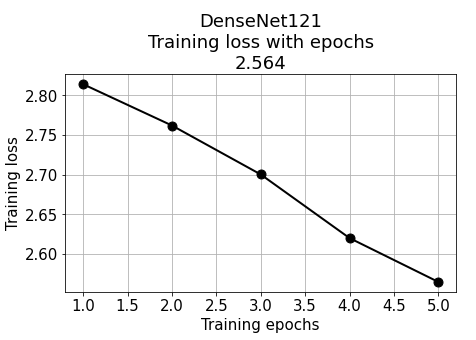


Рис. 25. DenseNet121 – динамика значения loss на протяжении эпох при обучении.

* 1. accuracy 0.13 (низкое значение – неудовлетворительный результат) (рис. 26);

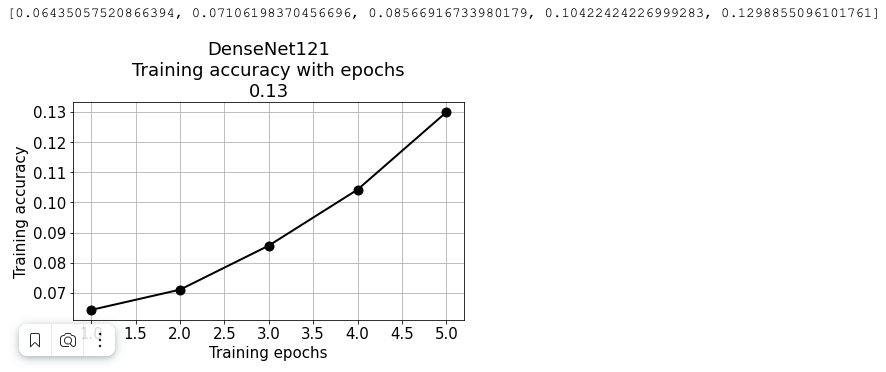


Рис. 26. DenseNet121 - динамика метрики accuracy на протяжении эпох при обучении.

После обучения и проверки работы моделей глубокого обучения следует выбрать подходящую, проанализировав ее выход на метриках.

Как видим, модель CNN показывает более высокую точность, чем DenseNet121. CNN использует сверточные слои для извлечения признаков, которые могут быть полезны для классификации изображений и видео. В данной работе была реализована собственная модель CNN (не предобученная), поэтому она имеет в своей памяти только изображения из поданных ей на вход тренировочных данных (sample), скорость обучения 762 сек – 12,7 мин.

DenseNet121, с другой стороны, является более общей моделью глубокого обучения и может использоваться для широкого спектра задач. Он имеет более высокую плотность связей между слоями, что может привести к более высокой точности в некоторых задачах, но не обязательно в задачах классификации изображений и видео. Здесь была использована уже предобученная модель DenseNet121 из библиотеки keras с добавлением нескольких слоев, в ее памяти множество различных значений, поэтому скорость обучения достаточно большая (13480 сек – 224,67 мин). К тому же, метрики могут показывать плохие значения из-за небольшого количества данных (0,02% от исходных). Без использования GPU, обучение на такой большой выборке занимает много ресурсов и времени

Если задача связана с обработкой изображений и видео, то CNN может быть более подходящей моделью с высокой точностью. Если же задача более общая, то DenseNet121 может быть более эффективным выбором.

После анализа выхода метрик и времени обучения было решено использовать самодельную модель сверточной нейронной сети CNN.

### Прогнозирование

Теперь можно приступить к предсказыванию на тестовой выборке (рис. 27).

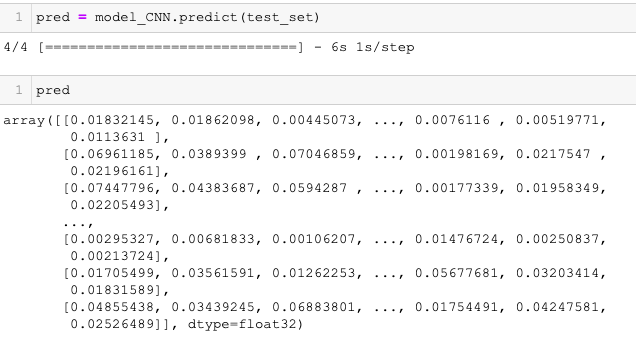


Рис. 27. Прогнозирование модели CNN.

Посмотрим метрику ROC-AUC и убедимся в эффективности (рис. 28).

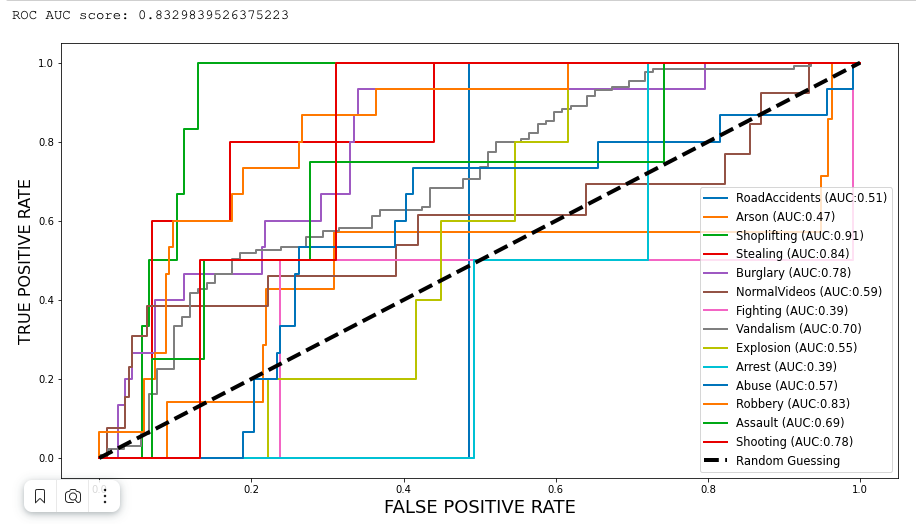


Рис. 28. Значение метрики ROC\_AUC.

Получили высокое значение ROC\_AUC – ~0.833. Следует заметить, что результат кривых немного несбалансирован. Причиной может являться небольшое количество данных в подаваемой на вход выборке. Однако, модель справляется со своей задачей на результат выше среднего.

# ЗАКЛЮЧЕНИЕ

Преступления являются одними из самых опасных и распространенных аномальных явлений в обществе. Их фиксация на камеру с использованием искусственного интеллекта может значительно помочь в раскрытии и расследовании преступлений, а также улучшить безопасность в обществе. В связи с этим, дальнейшее исследование в области распознавания преступлений на видео с использованием нейронных сетей имеет большую перспективу и может принести значительный вклад в развитие безопасности в обществе.

В результате анализа аномалий в видеоинформации с использованием набора данных Anomaly-Detection-Dataset-UCF / UCF Crime (раскадрованный) и нейронных сетей CNN и DenseNet121 была создана модель, способная обнаруживать аномалии в видео с высокой точностью. Результаты оценки производительности модели на тестовой выборке показали, что она может быть использована для обнаружения аномалий на кадрах реальных видеопотоков.

Однако стоит отметить, что модель может быть улучшена за счет более точной настройки параметров, использования дополнительных аугментаций данных и применения ансамблевых методов.

# СПИСОК ЛИТЕРАТУРЫ

1. Курс лекций Коротеева М.В., Определение машинного обучения – [Электронный ресурс]. – Режим доступа: <http://koroteev.site/pres/ml0/#6>
2. Machine Learning Algorithms Explained: Anomaly Detection – [Электронный ресурс]. – Режим доступа: <https://www.stratascratch.com/blog/machine-learning-algorithms-explained-anomaly-detection/>
3. Deep Anomaly Detection / Детекция аномалий с помощью методов глубокого обучения – [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/530574/>
4. Потапов А.С. Системы компьютерного зрения. Учебное пособие. – СПб: Университет ИТМО, 2016. – 161 с.
5. Лагунов Н. А., Мезенцева О. С. ВЛИЯНИЕ ПРЕДОБРАБОТКИ ИЗОБРАЖЕНИЙ НА КАЧЕСТВО ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ ДЛЯ ИХ РАСПОЗНАВАНИЯ // Вестник Северо-Кавказского федерального университета: научный журнал: Изд-во СКФУ, 2015. No 1(46)
6. А. Ю. Буйко, А. Н. Виноградов. «Выявление действий на видео с помощью рекуррентных нейронных сетей», Программные системы: теория и приложения, 2017, 8:4(35), с. 327–345.
7. VNI Global Fixed and Mobile Internet Traffic Forecasts, – [Электронный ресурс]. – Режим доступа: https: //www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html
8. Waqas Sultani, Chen Chen, Mubarak Shah, Real-world Anomaly Detection in Surveillance Videos, Cornell University Library, arXiv:1801.04264 [cs.CV], [v1] Fri, 12 Jan 2018. – [Электронный ресурс]. – Режим доступа: <https://www.crcv.ucf.edu/projects/real-world/>
9. Используемый датасет UCF Crime с кадрами на Kaggle – [Электронный ресурс]. – Режим доступа: <https://www.kaggle.com/datasets/odins0n/ucf-crime-dataset>
10. How to increase your small image dataset using Keras ImageDataGenerator, – [Электронный ресурс]. – Режим доступа: <https://medium.com/@arindambaidya168/https-medium-com-arindambaidya168-using-keras-imagedatagenerator-b94a87cdefad>
11. Генерация изображений с помощью ImageDataGenerator библиотеки Keras – [Электронный ресурс]. – Режим доступа: <http://www.100byte.ru/python/imgGenerator/imgGenerator.html>
12. Dalia Ezzat, Aboul Ella Hassanien, Hassan Aboul Ella. An optimized deep learning architecture for the diagnosis of COVID-19 disease based on gravitational search optimization // National Library of Medicine, 2020. – [Электронный ресурс]. – Режим доступа: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7505822/>
13. Koo Ping Shung Accuracy, Precision, Recall or F1? // Towards Data Science, 2018 . – [Электронный ресурс]. – Режим доступа: <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>
14. Vimash Karbhari What is AUC? // Towards Data Science, 2019. – [Электронный ресурс]. – Режим доступа: <https://medium.com/acing-ai/what-is-auc-446a71810df9>
15. Сикорский О.С. Обзор сверточных нейронных сетей для задачи классификации изображения, МГТУ им. Н.Э. Баумана – [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/obzor-svyortochnyh-neyronnyh-setey-dlya-zadachi-klassifikatsii-izobrazheniy/viewer>
16. Sik-Ho Tsang. Review: DenseNet — Dense Convolutional Network (Image Classification) // Towards Data Science, 2019. – [Электронный ресурс]. – Режим доступа: <https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803>
17. Репозитарий курсовой работы на GitHub – [Электронный ресурс]. – Режим доступа: <https://github.com/hyo0803/anomalyDetection_KhertekYO>