# Применение алгоритмов машинного обучения для обнаружения и классификации дефектных зон на видеоизображениях внутренней поверхности графитовых блоков РБМК.

Ключевые слова: Automated Visual Inspection, computer vision, defect detection, semantic segmentation, multi-class segmentation, графитовые блоки реактора большой мощности канального.

## Аннотация

На многих атомных станциях подходят к концу сроки эксплуатации РБМК, что делает проблему визуального контроля и устранения дефектов графитовых блоков актуальной, как никогда раньше. В проделанном исследовании предлагается алгоритм, использующий глубокое обучение, для детекции малозаметных дефектов разных классов на внутренней поверхности графитовых блоков. Приводится сравнение аналогов, подробно описывается предлагаемое решение.

## Введение

Распухание графитовых блоков - серьезная проблема современной атомной промышленности, игнорирование которой приводит к выводу энергоблока из эксплуатации и, как следствие, остановке выработки энергии. В настоящий момент проблема решена, но для устранения дефектов, необходимо сначала их обнаружить. В настоящий момент для детекции дефектов, вызывающих описанные проблемы, используется камера снимающая внутреннюю поверхность блоков и оператор, отмечающий маркером дефекты на сгенерированном панорамном изображении. Такой подход работает, но устранение человеческого фактора могло бы повысить точность выявления малозаметных дефектов, а их своевременное устранение позволило бы экономить немалые деньги на простое энергоблоков. Подобные задачи детекции объектов решают алгоритмы компьютерного зрения, которые заключаются в ручном кодировании правил, определяющих класс искомого объекта и его местоположение на изображении. Но на практике такой подход оказывается недостаточно точным из-за наличия шумов на изображении, таких как неравномерное освещение или расфокус камеры. В данной работе для решения поставленной проблемы предлагается использовать глубокое обучение. Этот подход сильно отличается от предыдущего, в данном случае на вход некоторому алгоритму оптимизации подаются исходные данные и ожидаемый результат, а на выходе получаются, описанные выше правила, способные многократно решать поставленную проблему. Целью данного исследования является разработка алгоритма детектирования и классификации дефектов внутренней поверхности графитовых блоков, с использованием глубокого обучения. Для достижения поставленной цели необходимо решить следующие задачи:

1. Разметка изображений для обучения
2. Предварительная обработка изображеий, на которых будет обучаться алгоритм
3. Подбор оптимальных параметров алгоритма, обеспечивающих высокое качество детектирования
4. Обучение и оценка алгоритма

Под предварительной обработкой понимается генерирование новых изображений для обучения, используя различные трансформации (отражение, перемещение, масштабирование). Такой метод очень эффективен для борьбы с переобучением модели при небольшом количестве исходных данных для обучения.

# <Обзор предметной области == Сравнение аналогов>

## Принцип отбора аналогов

Был реализован поиск статей, описывающих алгоритмы компьютерного зрения для детекции объектов, с использованием машинного обучения и без.

Использованные ключевые слова: Automated Visual Inspection, computer vision, defect detection, semantic segmentation, multi-class segmentation, графитовые блоки реактора большой мощности канального.

## Аналоги

### Контроль, с помощью оператора

Расположенный за машиной оператор отмечает маркерами дефекты на полученных изображениях. При таком подходе могут быть пропущены мелкие, малозаметные дефекты, также невозможно нанесение точного контура дефектов, так как это слишком долгая для человека процедура.

### Детектор границ канни

Алгоритм возвращает контуры объектов на изображении.[5] Алгоритм способен точно детектировать объекты, но в некоторых случаях ему свойственен большой шум и неспособность разделять 2 разных класса похожих друг на друга дефектов. Алгоритм лучше чувствителен к освещению, чем его аналоги, основанные на анализе цвета, что часто оказывается очень полезным.[7][8]

### Анализ цветовой гистограммы изображения

Строится гистограмма распределения значений пикселей, подбирается предел бинаризации, при котором будут выделяться нужные объекты. Предел бинаризации - это значение цвета, котороя используется для классификации. Если значение пикселя больше предела, то пикселю назначается один класс, если меньше, то оставшийся класс.[2] Такой метод очень чувствителен к освещению, которое не всегда одинаково, также в некоторых случаях фон не однотонный, и его цвет может пересекаться с цветом дефектов. Метод может быть модернизирован автоматической оптимизацией значения предела бинаризации.

### Использование алгоритмов кластеризации

Методы основанные на кластеризации разбивает изображение на объекты путем объединения значения цветов в заранее заданное количество кластеров.[1][4]

### Использование сверточных нейронных сетей

Применение моделей глубокого обучения, решающих задачу семантической сегментации, таких как Unet, DeepLab, позволяет извлекать из изображений маски объектов для каждого класса дефектов. Такой подход занимает больше времени на обработку изображений, но способность нейронных сетей к извлечению признаков объектов, позволяет получать

точность недостижимую для классических алгоритмов компьютерного зрения, рассмотренных ранее.[3][6]

## Критерии сравнения аналогов

### Скорость алгоритма

Скорость алгоритма характеризуется временем необходимым для обработки одного изображения. Для сравнения аналогов, значение скорости условно разделено на группы: быстро, средне, медленно.

### Точность алгоритма

Точность оценивается метрикой, характеризующей сходство результата алгоритма и ожидаемого результата, полученного ручной разметкой. Самые популярные метрики: mIoU, Dice.

### Зависимость от аппаратного обеспечения

Некоторое алгоритмы могут быстро работать на любом оборудовании, а некоторые нет, иногда это может быть критично.

### Способность распознавать несколько классов

Реализуемый алгоритм должен справляться не только с детекцией дефектов, но и их классификацией. В данной задаче дефекты из разных классов имеют очень схожие признаки, особенно цвет, поэтому подобрать несколько наборов параметров алгоритма для каждого класса, так чтобы классы не пересекались, может оказаться нетривиальной задачей.

## Таблица сравнения по критериям

| | **Точность** | **Скорость** | **Оборудование** | **Несколько классов** |
|---|---|---|---|---|
| Оператор (человек) | Низкая | Средняя | + | + |
| Канни | Средняя | Высокая | + | - |
| Гистограмма | Средняя | Высокая | + | - |
| Кластеризация | Средняя | Низкая | - | + |
| Глубокое обучение | Высокая | Средняя | - | + |

## Выводы по итогам сравнения

В результате обзора аналогов, можно сделать вывод, что решение с применением глубокого обучения является единственным, удовлетворяющим большинство выбранных критериев.

## Описание метода решения

Шаги, необходимые для выполнения поставленных задач, схематично описаны на рис. 1.

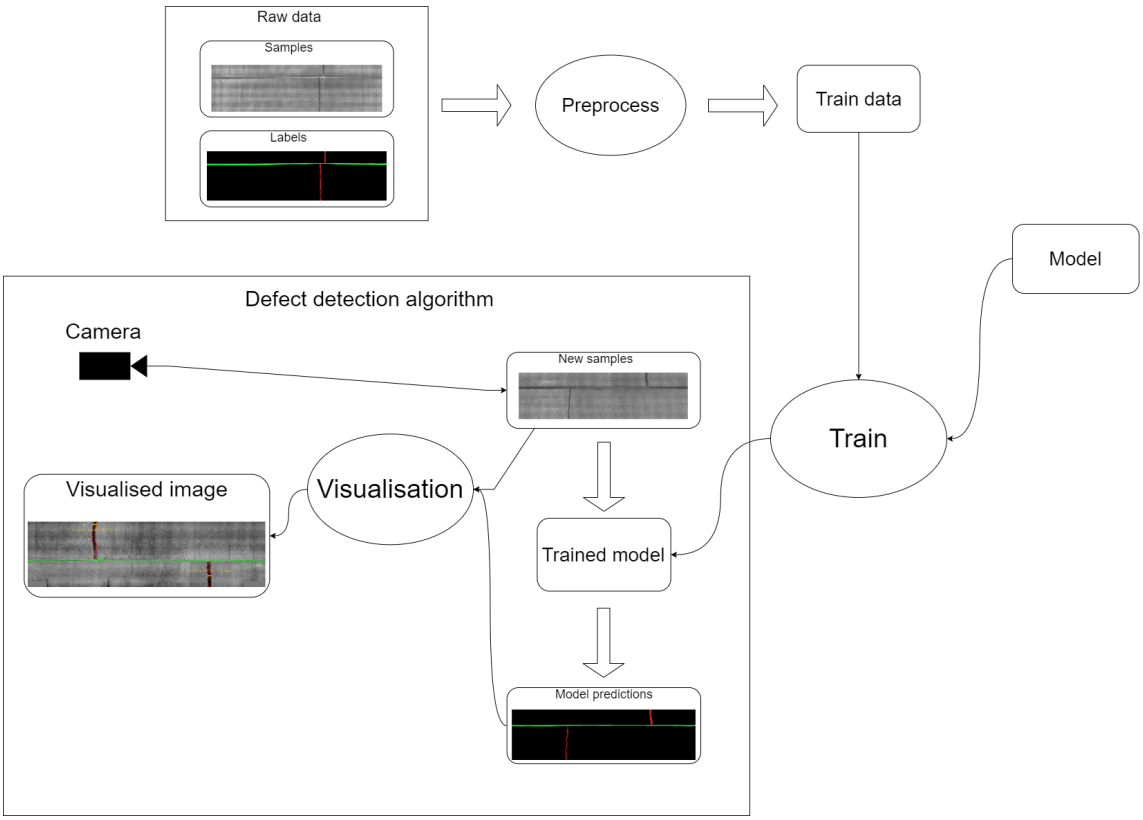


Рисунок 1 - Схематичное представление проделанной работы

В блоке Raw data находятся исходные данные, используемые для обучения модели. Samples - образцы, полученные в результаты визуального контроля внутренней поверхности графитовых блоков, Labels - разметки образцов, представляющие из себя трехцветные изображения, где каждому каждому цвету соотвествует отдельный класс объектов: красный - трещина, зеленый - зазор, черный - фон. В таком виде модель не способна принять данные, поэтому следующим идет процесс предобработки данных. Для начала тип каждого пикселя преобразуется из 8и битного целого к 32х битному числу с плавующей точкой, такой же тип традиционно имеют веса модели. Затем идёт нормализация, существует множество методов нормализации, здесь используется масштабирование данных, в результате которого пиксели принимают значения от 0 до 1, в данном случае для масштабирования достаточно поделить каждое значение на 255. Нормализация является не обязательной процедурой, но рекомендуемой, так как, она позволяет уменьшить значения данных, не теряя информации, а чем меньше значения данных, тем меньше значения весов модели, что препятствует переобучению. Следующим шагом, для каждой пары образец-разметка, применяются описанные ранее трансформации, позволяющие расширить имеющийся объем данных. Демонстрацию метода можно увидеть на рис.2.

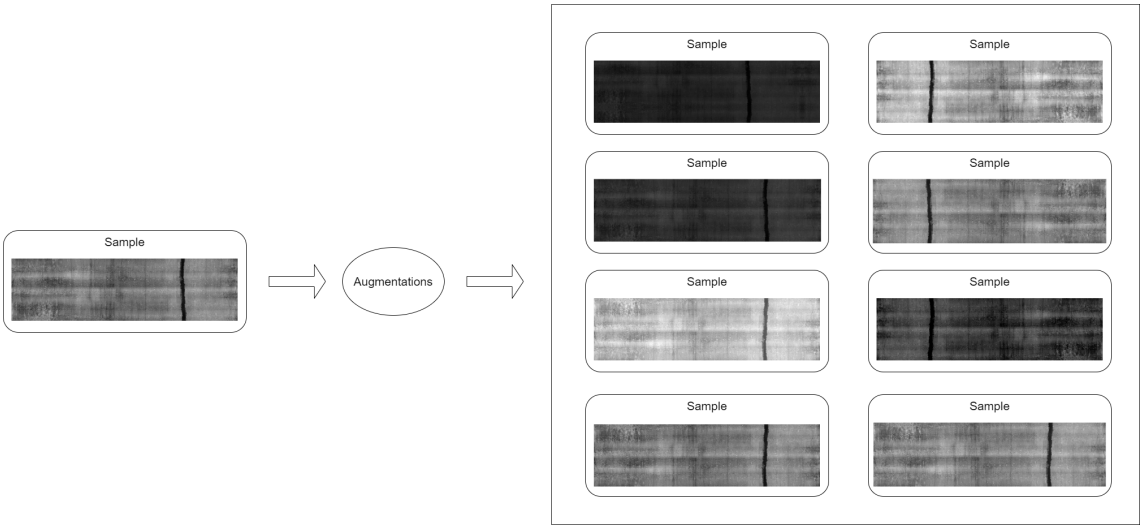


Рисунок 2 - Демонстрация метода аугментирования данных

Процедура Augmentations представляет из себя случайное, последовательное применение следующих операций: изменение яркости, изменение контраста, замыливание, вертикальный/горизонтальный повороты, масштабирование. Случайность в данном случае означает, что каждая операция может как выполниться, так и не выполниться, с некоторой вероятностью, значения изменяемых параметров также подбираются случайно из некоторого диапазона. Все это в сумме даёт огромное количество комбинаций, из которых генерируются непохожие на оригинал образцы. Как видно из рис.2, использование этого метода позволяет в разы увеличить количество кадров, участвующий в обучении, тем самым решив проблему переобучения модели и добившись большей точности.

Следующим шагом предобработки данных, будет приведение разметки к формату, необходимому моделе. Каждый пиксель разметки может принимать одно значение из трех категорий: зеленый, красный или черный. Для использования разметки в обучении, необходимо заменить сопоставить каждой категории число от 0 до число_категорий-1, затем заменить каждое категориальное значение на вектор, в котором 1 стоит на месте, соответствующем категории, к которой принадлежит пиксель, в остальных местах 0. Тогда может получится, что черному цвету соответсвует вектор (1, 0, 0), красному - (0, 1, 0) и зеленому - (0, 0, 1). Данная процедура называется one-hot encoding. Как и в случае с образцами, каждое число должно иметь тип с плавующей точкой.

Затем перед тем как начать обучение, необходимо определится с архетиктурой модели глубокого обучения. В данном исследовании была выбрана архитектура Unet, её схематичная реализация представлена на рис.3.

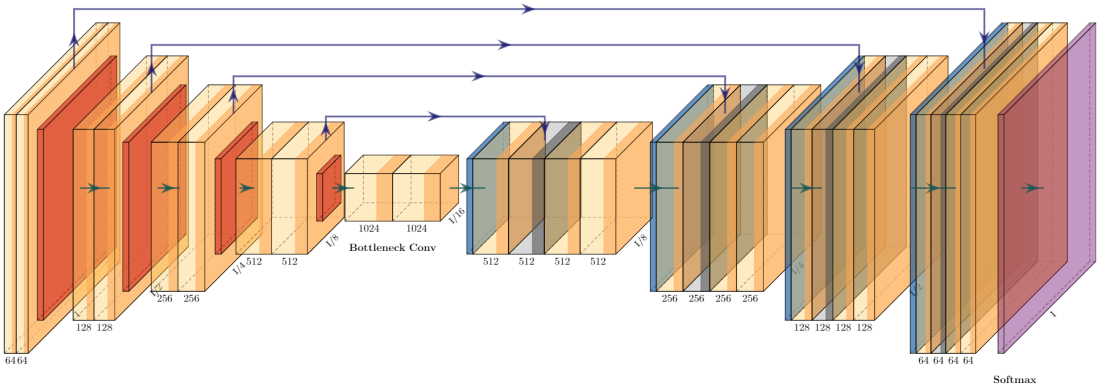


Рисунок 3 - Архитектура модели

Данная архитектура была выбрана по причине простоты реализации и высокой точности, что идеально для прототипизации.

Призмы на схеме представляют собой входные, выходные и промежуточные слои. Каждому цвету соотвествует своя операция, так оранжевому слою соответствует операция свертки, красному слою - операция maxpooling, синему - upsampling, последний слой - свертка ядром с размерностью 1x1 и количеством фильтров, равному количеству предсказываемых классов, следом функция активации softmax. Перечисленное - то, что Unet унаследовала от FCN архитектуры, а добавилась операция, которая на схеме отмечена синей стрелочкой. Она заключается в сохранении блока из которого выходит стрелочка и его конкатенации, в будущем, с блоком, в который входит стрелочка. Операция называется skip-connections, и, как выяснилось, она эффективно решает проблему затухания градиента, из-за которой обратное распространение ошибки не доходит до начальных слоев, в следствие чего веса не обновляются, и модель оказывается необучаема.

Входные данные модели: описанный выше формат предобработанных исходных образцов, выходные данные модели: тензор с формой, аналогичной форме, описанного ранее формата обработонных разметок - (ширина, высота, количество классов/категорий), также благодаря функции активации softmax, тензор обладает следующим свойством - сумма чисел каждого вектора последней оси тензора равна единице. Если принять, что принадлежность каждого пикселя к классу мы определяем по индексу с максимальным значением в векторе, который соотвествует пикселю, то свойство можно проинтерпретировать следующим образом: полученная модель ставит в соответсвие каждому пикселю входного изображения вероятностною оценку принадлежности к каждому из классов, или еще короче: модель классифицирует каждый пиксель.

Процесс обучения модели состоит в следующим: преобработанные образцы для обучения группируются в пакет фиксированной длины, пакет проходит через, описанные выше операции, формируя пакет выходных данных. Как уже отмечалось, форма выходных данных и результата обработки разметок одинакова, что позволяет оценить их сходство некоторой функцией ошибки и получить значение ошибки. Затем алгоритм обратного распространения ошибки обновляет веса таким образом, чтобы в следующую итерацию значение ошибки стало меньше, а сходство разметок и выходных данных модели - больше.

Обычно в задачах классификации в качестве функции ошибки выступает точность, расчитываемая отношением правильно угаданных образцов к количеству образцов, но при решении поставленной проблемы, данный подход неприменим по следующей причине: предположим на исходном изображении есть малозаметный дефект - маленькая трещина, но модель вернет результат, в котором каждый пиксель принадлежит фону, в таком случае при

расчете ошибки по формуле точности получим 99% правильно угаданных пикселей, которые принадлежат фону, поделим на 100% и получим значение точности, которое близко к максимальному, при этом модель не смогла выполнить ожидаемою от нее задачу - детектировать дефект. Описанная проблема называется - дисбалансный набор данных, и решается использованием специально смоделированных для таких случаев функций. В данном исследовании используется функция потерь Дайса, обобщенная до многомерного случая[7], которая высчитывается по следующей формуле:

$$Dice\_loss = 1 - 2\frac{\sum_{l=1}^{k} w_l \sum_n r_{ln} p_{ln}}{\sum_{l=1}^{k} w_l \sum_n r_{ln} + p_{ln}}$$

, где w - некоторый весовой коэффициент, соотвествующий классу, r и p - значения выхода модели и разметки на соотвествующей позиции, k - количество классов. Данная функция имеет область определения от 0 до 1, что позволяет удобно интерпретировать её значения от "очень похожи", до "сходства нет". Если рассмотреть, приведенный ранее пример, расчитывая потерю по формуле Дайса, то результат числителя будет равен нулю, а потери примут максимальные значения, что будет означать отсутствие сходства, как мы и ожидали.

Модель, полученная в результате обучения, далее может быть многократно использована для детекции дефектов на изображениях, приходящих с камеры, осуществляющей визуальный контроль внутренней поверхности графитовых блоков. Изображения, полученные с камеры, а также результат их обработки моделью, проходят через процесс визуализации, где на исходное изображение наносятся контуры дефектов с соответствующим цветом, характеризующим класс объекта, для каждой трещины считается ее ширина и также наносится на изображении.

Технологии, которые использовались в исследовании: Язык программирования - Python, так как он имеет большое количество библиотек и фреймворков, реализующих весь низкий уровень реализации машинного и глубокого обучения, обработки изображений и т.д. Фреймворк глубокого обучения - Keras, так как он наиболее высокоуровнев и прост в освоении, что хорошо для прототипизации. Библиотека для обработки изображений и компьютерного зрения - OpenCV, так как имеет очень большой набор функций для обработки изображений, также, как и Keras работает с numpy тензорами.

## Исследование метода решения (если есть)

0000000000

## Заключение

Данное исследование посвящено разработке алгоритма, с использованием методов компьютерного зрения, для детекции и классификации дефектов на внутренней поверхности графитовых блоков. Был проведен сравнительный анализ алгоритмов, способных решить поставленную задачу, в результате которого решено использовать глубокое обуение. К исходным изображениям и их разметкам были применены трансформации, позволяющие дополнить обучающую выборку новыми данными, решить проблему переобучения и увеличить точность. Также применение к исходным данным таких преобразований, как изменение яркости, контраста и замыливание, позволило полученной модели быть более устойчивой к таким шумам, как неравномерное освещение и расфокус. Получена модель, показывающая хорошую точность, позволяющую детектировать малозаметные дефекты. Написан программный модуль, визуализирующий выход модели в понятный человеку формат.

Несмотря на то, что поставленная цель достигнута, остаётся простор для развития. Исходные панормамные изображения имеют огромное разрешение, не позволяющее разместить в памяти все данные, необходимые для обучения и обработки, поэтому кадры просто сжимаются до маленького разрешения. При таком подходе может теряться информация, позволяющая обеспечить более высокую точность детектирования. Во избежание слепого сжатия изображений предлагается разбить исходные изображения на множество маленьких и работать с ними, при необходимости, восстанавливая полномасштабные изображения. Такой подход позволит повысить точность детектирования в ущерб скорости обработки каждого изображения. Подбор гиперпараметров, обеспечивающих более высокую скорость и точность, изучение различных архитектур моделей, применительно к данной проблеме, также являются направлениями для будущих исследований.

## Список литературы


1. Pham, V.H., Lee, B.R. An image segmentation approach for fruit defect detection using k-means clustering and graph-based algorithm. Vietnam J Comput Sci 2, 25–33 (2015). https://doi.org/10.1007/s40595-014-0028-3
2. Akarsu, B , Karaköse, M , Parlak, K , Akın, E , Sarımaden, A . (2016). A Fast and Adaptive Road Defect Detection Approach Using Computer Vision with Real Time Implementation . International Journal of Applied Mathematics Electronics and Computers , Special Issue (2016) , 290-295 . DOI: 10.18100/ijamec.270546
3. Zhenqing Liu, Yiwen Cao, Yize Wang, Wei Wang, Computer vision-based concrete crack detection using U-net fully convolutional networks https://doi.org/10.1016/j.autcon.2019.04.005. (http://www.sciencedirect.com/science/article/pii/S0926580519301244)
4. Hamdi A. A. et al. Unsupervised patterned fabric defect detection using texture filtering and K-means clustering //2018 International Conference on Innovative Trends in Computer Engineering (ITCE). – IEEE, 2018. – С. 130-144.
5. Peter W. T., Gaochao W. Sub-surface defects detection of by using active thermography and advanced image edge detection //J. Phys.: Conf. Ser. – 2017. – Т. 842. – С. 012029.
6. Dung C. V. et al. Autonomous concrete crack detection using deep fully convolutional neural network //Automation in Construction. – 2019. – Т. 99. – С. 52-58.
7. Sudre C. H. et al. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations //Deep learning in medical image analysis and multimodal learning for clinical decision support. – Springer, Cham, 2017. – С. 240-248.

**Reviews:**

8. Mohan A., Poobal S. Crack detection using image processing: A critical review and analysis //Alexandria Engineering Journal. – 2018. – Т. 57. – №. 2. – С. 787-798.
9. Rasheed A. et al. Fabric Defect Detection Using Computer Vision Techniques: A Comprehensive Review //Mathematical Problems in Engineering. – 2020. – Т. 2020.