

УДК 69.059.4

## **Автоматизация процесса определения трещин при обследовании кирпичных зданий**

**С.А. Крылов<sup>1</sup>, В.Ю. Платунов<sup>1</sup>, Г.Г. Кашеварова<sup>1</sup>**

<sup>1</sup>Пермский национальный исследовательский политехнический университет,  
г. Пермь, Российская Федерация

**Аннотация.** Постановка задачи. При эксплуатации зданий обслуживающие организации часто пренебрегают выполнением периодических осмотров и обследований для определения текущего технического состояния зданий, а также требуемых мероприятий для своевременного восстановления поврежденных конструкций. Это связано с низкой компетенции персонала обслуживающих организаций и высокой стоимости обследовательских работ, что ведет к снижению сроков нормальной эксплуатации зданий и снижению их безопасности.

Целью работы является разработка программы по распознаванию дефектов на поверхности фасадов зданий на основании существующих методов по определению объектов. Задачи работы заключаются в написании алгоритма, его кодировании, сборе обучающего набора данных, описании методов формирования обучающей выборки, определении параметров и выполнении обучения модели, а также тестировании программы.

Результаты. В работе описан алгоритм создания программы на основе сверточной нейронной сети с учетом негативного влияния внешних факторов. Определены оптимальное соотношение параметров модели обучения и значения показателей качества ее работы. Описан метод обучения нейронной сети.

Выводы. Значимость полученных результатов для строительной отрасли состоит в том, что применение созданной программы позволит увеличить точность и скорость определения дефектов и создания отчетов о техническом состоянии зданий, что в свою очередь повысит безопасность зданий.

**Ключевые слова:** обследование зданий, дефекты, техническое состояние, автоматизация, модель, обучение.

**Для цитирования:** Крылов С.А., Платунов В.Ю., Кашеварова Г.Г. Автоматизация процесса определения трещин при обследовании кирпичных зданий // Строительные конструкции, здания и сооружения, 2024, № 4(9), с. 22-31.

## **Automation of the crack detection process during the inspection of brick buildings**

**S. A. Krylov<sup>1</sup>, V. Yu. Platunov<sup>1</sup>, G. G. Kashevarova<sup>1</sup>**

<sup>1</sup> Perm National Research Polytechnic University,  
Perm, Russian Federation

**Abstract.** Problem statement. During the operation of buildings, maintenance organizations often neglect to perform periodic inspections and surveys to determine the current technical condition of buildings, as well as the required measures for the timely restoration of damaged structures. This is due to the low competence of the staff of service organizations and the high cost of survey work, which leads to a reduction in the normal operation of buildings and a decrease in their safety. The aim of the work is to develop a program for recognizing defects on the surface of building facades based on existing methods for determining objects. The tasks of the work are to write an algorithm, encode it, collect a training dataset, describe methods for



forming a training sample, determine parameters and perform model training, as well as test the program.

Results. The paper describes an algorithm for creating a program based on a convolutional neural network, taking into account the negative influence of external factors. The optimal ratio of the parameters of the learning model and the values of the quality indicators of its work are determined. The method of neural network training is described.

Conclusions. The significance of the results obtained for the construction industry lies in the fact that the application of the created program will increase the accuracy and speed of detecting defects and creating reports on the technical condition of buildings, which in turn will increase the safety of buildings.

**Keywords:** building inspection, defects, technical condition, automation, model, training.

**For citation:** Krylov S. A., Platunov V. Yu., Kashevarova G. G. Automation of the crack detection process during the inspection of brick buildings // Construction, buildings and structures, 2024, № 4(9), pp. 22-31.

### 1. Введение.

Повышение конструктивной безопасности – одно из приоритетных направлений комплексной безопасности зданий и сооружений РФ. Техническое обследование жилых зданий требуют обширных знаний в области строительства и существенно зависит от опыта эксперта.

На сегодняшний день комплекс мероприятий по определению и оценке фактических значений контролируемых параметров строительных конструкций, характеризующих работоспособность объекта обследования, а также выбор решений о возможности его дальнейшей эксплуатации об усилении или ремонте, как правило, выполняется без применения средств автоматизации [1]. Ручной способ выявления и классификации дефектов требует значительных затрат времени и усилий, что сказывается на стоимости работ по обследованию зданий. При этом высокая стоимость увеличивает сроки между плановыми осмотрами/обследованиями, так как имеется постоянный дефицит средств на обслуживание здания. Также количество организаций, предоставляющих данные услуги, недостаточно для полноценного контроля всего жилого фонда, поэтому большая часть осмотров/обследований выполняется на низком уровне и без учета технологии эксплуатации здания [2]. Это негативно влияет на сроки эксплуатации зданий, так как отсутствие регулярного мониторинга приводит к тому, что дефекты остаются незамеченными пока не достигнут критического уровня. Также это не позволяет эффективно планировать и экономично реализовывать мероприятия по предупреждению и не позволяет своевременно выполнять восстановительные работы для доведения здания до надлежащего технического состояния с минимальными затратами [3].

Статья посвящена рассмотрению процесса автоматизации обследования зданий, которое заключается в определении дефектов на фотографиях фасадов кирпичных зданий [4]. Для апробирования работоспособности создаваемого алгоритма вводятся ограничения по типам дефектов – рассматриваются только трещины на фасадах, по материалу стен – только кирпичные здания, по типу фасадов – без облицовки наружных стен. С этой целью сверточная нейронная сеть (Convolutional Neural Networks, CNN) с множеством скрытых слоев обучается на данных, размеченных вручную [5–7]. Сверточной называют вид нейронной сети, специализирующейся на обработке изображений и видео. Это особые типы нейронных сетей, которые помогают компьютерам видеть и понимать изображения и видео.

При обследовании зданий зачастую требуется привлечение дополнительных приспособлений, специалистов и спецтехники (лестницы, альпинисты, подъемники, автовышки и др.), поэтому актуальным является внедрение беспилотных летательных аппаратов (БПЛА) в обследования зданий, что позволит ускорить, удешевить и повысить качество обследований. Автоматизация определения дефектов на поверхностях фасадов зданий позволит упростить внедрение и обработку фото и видео файлов с БПЛА [8].



Анализ научных публикаций показывает, что за последние годы проведено большое количество исследований, в области определения объектов на фотографиях или видео [9–11]. Одна из основных задач анализа изображений в строительной сфере заключатся в проведении мониторинга существующей застройки, чтобы минимизировать или полностью исключить проживание людей в аварийных зданиях.

С этой целью внесены правки в Градостроительный и Жилищный кодексы Российской Федерации связанные с процедурой включения аварийных многоквартирных домов в границы комплексного развития территории (КРТ) для реновации жилья. При этом часто возникают ситуации, когда ни жильцы, ни обслуживающая организация, ни администрация не имеет информации о категории технического состояния здания, так как длительный срок не производились технические обследования. В связи с этим произошла актуализации нормативной базы и разработано СП 454.1325800.2019, проведение обследования по которому выполняется по упрощенной процедуре и по значительно меньшей стоимости, чем по ГОСТ 31937-2011. При определении технического состояния здания по СП 454.1325800.2019 в основном используется визуально-инструментальный метод обследования, который легче автоматизировать.

Задачи автоматизации определения технического состояния зданий заключаются в:

- снижении влияния фактора субъективности экспертов;
- увеличении точности и качества определения дефектов
- повышении производительности труда;
- повышении скорости создания отчетов о техническом состоянии зданий;
- подготовка к дальнейшему внедрению БПЛА в осмотры/обследования зданий.

Сложность распознавания трещин на фасадах кирпичных зданий с помощью нейронных сетей заключается в следующем:

- трещины с малым раскрытием. Они могут быть короткими и с малым раскрытием, в связи с чем разрешения изображений или видео может быть недостаточно чтобы их определить при съемке с уровня земли;
- наличие веток деревьев, кабелей и других предметов на фасадах зданий, которые негативно влияют на точность определения дефектов;
- зависимость от условий съемки таких как освещение, угол обзора. Они могут значительно влиять на качество изображений;
- неоднородность и разнообразность кладки кирпичных стен как по цвету кирпича, так и по виду кладки (ложковая, цепная и т.д.).

Все вышеперечисленное негативно влияют на точность определения дефектов и требуют большого объема данных для обучения в сравнении с дефектами на бетонных поверхностях, что в свою очередь требует значительных временных затрат. Поэтому для исследовательских целей и апробирования работоспособности разработанной программы авторы умышленно ограничили размер набора данных.

В связи с вышесказанным, цель исследования заключается в разработке программы по распознаванию трещин на фотографиях фасадов зданий.

Задачи исследования:

1. Разработка и кодирование алгоритма, сбор исходного набора данных для обучения модели;
2. Описание методов формирования обучающей выборки и процесса обучения нейронной сети;
3. Определение параметров модели, основных показателей качества работы и оптимального количества эпох обучения;

Проверка работоспособности/тестирование программы.

## **2. Материалы и методы.**

В рамках поставленной задачи собран набор данных состоящий из 780 различных обучающих изображений и 30 тестовых. Размер каждого изображения: 640×640 пикселей. Качество подготовленного набора данных и размер выборки оказывает значительное влияние на процесс обучения и качество работы сети, а именно нейронная сеть может выделять дефект, отсутствующий на истинном изображении, или наоборот [12]. Это снижает качества работы модели.



Обучение выполнено с на основе метода обратного распространения ошибки (метод вычисления градиента), который заключается в следующем: сначала модель (детектор объектов) делает прогноз расположения трещины, далее сравнивает его с правильным расположением и определяет, насколько она ошиблась. После чего выполняется обратное распространение (в направлении обратном прямому распространению в обычном режиме работы) и регулирует параметры модели (веса и смещения) пропорционально тому, насколько они способствуют общей ошибке. Корректировка позволяет модели с каждым разом быть более точной. Веса – это числовые значения, которые определяют силу связей между нейронами и фиксируют взаимосвязи между входными характеристиками и целевыми выходными данными. Они применяются для настройки модели, чтобы она могла правильно интерпретировать данные и производить более точные прогнозы. Веса в процессе обучения обновляются после каждой итерации как в пределах эпохи, так и в следующих эпохах. Это позволяет нейронным сетям адаптироваться, обучаться и входные обобщать данные. При обновлении весов происходит перерасчет градиентов, которые указывают насколько и в каком направлении нужно изменять веса, чтобы уменьшить ошибку прогноза и минимизировать функции потерь.

Написание кода программы выполнено на базе сервиса Google Colab с применением видеоадаптера<sup>1</sup>. Google Colab — это онлайн среда для разработки и выполнения программного кода в облаке, который позволяет писать и запускать код на языке Python в браузере, а также предоставляет доступ к мощным вычислительным ресурсам. Программная реализация сверточной нейронной сети (CNN) осуществлена на языке программирования Python 3.5 с использованием библиотеки TensorFlow, в которой применяется модель для обнаружения объектов You Only Look Once (YOLO) версии 7. TensorFlow - это открытая программная библиотека для машинного обучения, разработанная Google для решения задач построения и тренировки нейронной сети с целью автоматического распознавания объектов. Она содержит набор готовых функций, классов и модулей, которые можно использовать в различных проектах для машинного обучения.

YOLOv7 – это модель/алгоритм для обнаружения объектов на изображениях и в реальном времени, которая на данный момент превосходит все известные детекторы объектов по скорости и точности [13]. Она содержит встроенный массив данных для распознавания, сегментации и создания подписей к объектам.

Для работы в Google Colab первоначально необходимо загрузить исходные данные (обучающий набор данных и веса).

Далее требуется скорректировать количество и наименование классов в детекторе объектов. В данном случае применяется один класс с названием «Трещины» («Cracks»). Операция выполняется следующими командами:

```
with open('../yolov7.yaml', 'r') as f:
    yolo_text = f.read()
classes = ['Cracks']
new_yolo_text = yolo_text.replace('nc: **', f'nc: {len(classes)}')
with open('../yolov7.yaml', 'w') as f:
    f.write(new_yolo_text)
```

После чего запускается обучение детектора объектов, которое заключается в тренировке обнаруживать трещины на фотографиях фасадов зданий из набора исходных данных. Набор данных содержит изображения фасадов как с трещинами, так и без них, а также с различным расположением и раскрытием трещин. Для обучения в YOLOv7 загружается обучающий набор данных (data set) с размеченными (указанными) в специализированной программе трещинами на фотографиях фасадов, анализируя которые детектор объектов учится обнаруживать требуемые объекты на фотографиях. Под метками подразумеваются прямоугольные рамки с назначенными классами, которые наносятся вручную.

Значения параметров определялись путем подбора, находя оптимальный баланс

<sup>1</sup> Видеоадаптер технологии NVidia -SMI 535.104.05 CUDA Version: 12.2.



между точностью и скоростью. То есть изменяли параметры и производили повторное обучение. Путем анализа основных показателей определяли оптимальное сочетание параметров. В примере устанавливается количество обучающих фотографий за одну итерацию (batch) равное 16. Это означает, что за одну итерацию обрабатывается 16 изображений из обучающего набора данных. При этом необходимо понимать, что увеличение величины batch приводит к ускорению обучения, но в то же время к снижению точности.

Также принято количество потоков, равное 32, которое обозначает количество параллельных процессов, используемых для предварительной обработки и загрузки данных во время обучения модели. Не рекомендуется применять слишком большое количество потоков, так как это может привести к снижению производительности из-за дополнительных расходов на управление потоками.

В данном примере количество эпох переменное для определения наиболее оптимального значения. Под эпохой подразумевается один полный проход по всему обучающему набору данных во время тренировки модели машинного обучения. В течение эпохи модель обновляет свои веса, что позволяет ей улучшать свою точность.

```
!python train.py
--batch 16
--epochs **
--data ../data.yaml
--weights ../yolov7.pt
--workers 32
--name {RUN_NAME}
--project {PROJECT}
--device 0
```

Далее запускается процесс определения дефектов на тестовой выборке с использованием лучших из обученных весов для распознавания трещин на фотографиях фасадов зданий. При этом необходимо установить порог уверенности для предсказаний модели. В рассматриваемой модели данный порог равен 0,1, что означает вывод результатов с уверенностью предсказания не менее 10%.

```
!python ../detect.py
weights ../weights/best.pt
source ../images/
conf 0.1
```

Для того, чтобы увидеть на экране результаты обработки тестовых изображений необходимо добавить следующие команды:

```
import glob
from IPython.display import Image, display
i = 0
limit = 7 # max images to print
for imageName in glob.glob('../**/*.jpeg'): #assuming JPG
if i < limit:
display(Image(filename=imageName))
print("\n")
i = i + 1.
```

Для оценки качества работы алгоритмов сегментации в работе применяются следующие показатели (метрики) [14, 15]:

- Precision (P, точность) - показывает, сколько из объектов, которые алгоритм сегментации пометил как «верные», действительно являются «верными». Под понятием «верные» в данном примере подразумеваются трещины, которые авторы самостоятельно отметили в специализированной программе на изображениях фасадов зданий. Precision измеряет точность модели при определении заданного класса «Cracks».

- Recall (R, полнота) – это способность модели идентифицировать все экземпляры объектов на изображениях. Она определяется как отношение числа верных объектов, корректно классифицированных как «верные», к общему количеству «верных объектов».

Иллюстрация определения показателей точности и полноты представлена на рис. 1.



Intersection over Union (IoU) - это показатель, который измеряет степень перекрытия между предсказанными алгоритмом и истинными областями и определяется как разность пересечения истинной и предсказанной областей к площади объединения данных областей. Иллюстрация определения показателя IoU представлена на рис. 2.

- Mean Average Precision 0,5 (mAP50, средняя точность при IoU до 50% ) — это средняя точность, которая измеряется при пороге до 50% IoU в идентификации и классификации объектов на изображении. Это означает, что для того, чтобы прогноз считался верным, область предсказанного объекта должна перекрываться с истинным объектом как минимум на 50%.

- Mean Average Precision 0,5-0,95 (mAP50-95, средняя точность при IoU от 50% до 95%) - средняя точность при использовании различных порогов значений IoU от 50% до 95% с шагом 5%.

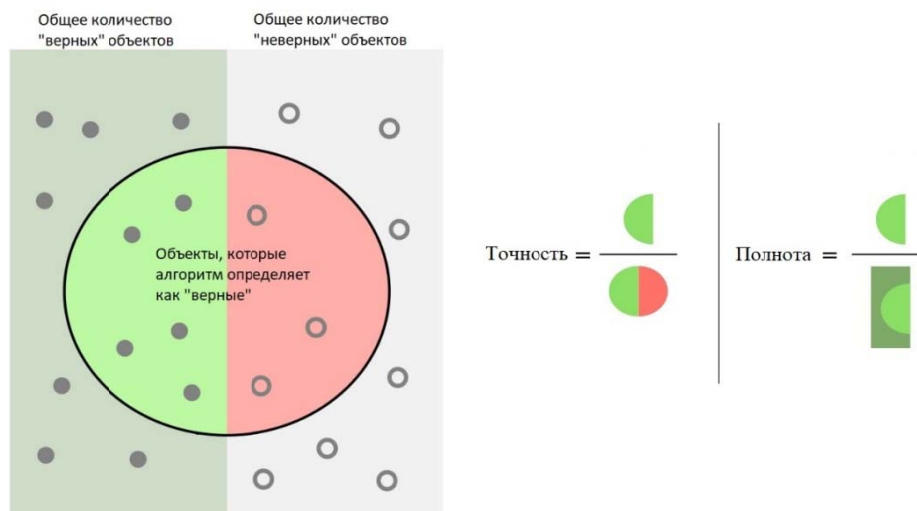


Рис. 1. Иллюстрация определения показателей точности и полноты (иллюстрация авторов)  
 Fig. 1. An illustration of the definition of accuracy and completeness indicators (illustration authors)

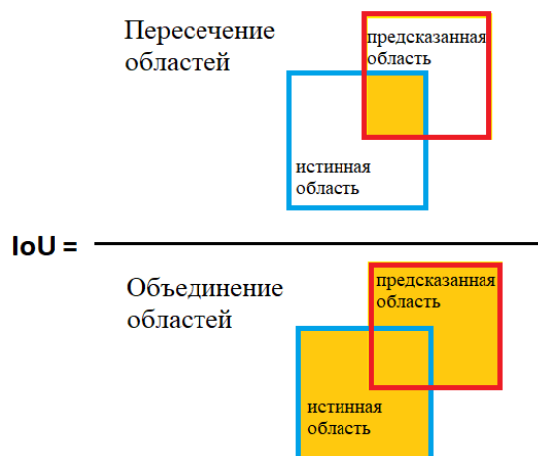


Рис. 2. Иллюстрация определения показателя IoU (иллюстрация авторов)  
 Fig. 2. Illustration of the definition of the IoU indicator (illustration authors)

Определение значений параметров выполнялось методом подбора. Значения метрик, которые получились в процессе подбора при обучении с различным количеством эпох приведены в табл. На рис. 3 представлены графики изменения основных показателей для варианта с 75 эпохами обучения. На горизонтальной оси отображены эпохи, на вертикальной оси значения метрик.



Таблица

Показатели метрик при различном количестве эпох обучения

№	Кол-во эпох	Время обучения, ч	Точность Precision	Полнота Recall	Средняя точность при IoU до 50%, mAP50	Средняя точность при IoU от 50% до 95%, mAP50-95
1	10	0,162	0,166	0,359	0,125	0,027
2	20	0,302	0,273	0,376	0,204	0,0574
3	35	0,515	0,327	0,415	0,234	0,0647
<b>4</b>	<b>50</b>	<b>0,731</b>	<b>0,328</b>	<b>0,421</b>	<b>0,263</b>	<b>0,0722</b>
5	65	1,011	0,293	0,338	0,205	0,0529
6	75	1,146	0,302	0,405	0,208	0,0498

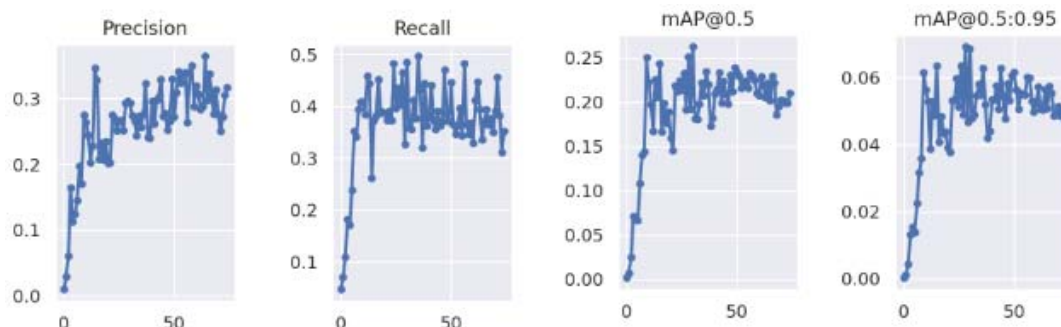


Рис. 3. Графики изменения основных показателей для варианта с 75 эпохами обучения (иллюстрация авторов)

Fig. 3. Graphs of changes in the main indicators for the variant with 75 epochs of learning (illustration authors)

### 3. Результаты и обсуждение.

Путем подбора количества эпох от 10 до 75 определили, что с собранными исходными данными целесообразно принять количества эпох равное 50, так как при меньшем значении происходит существенный разброс показателей, а при дальнейшем увеличении показатели точности и полноты значительно не изменяется, но время обучения заметно увеличивается, примерно в 1,57 раза для 75-ти эпох по отношению к 50-ти. При этом значения всех показателей после 50 эпохи снижаются. Это также указывает на оптимальное количество эпох около 50, а при большем количестве происходит переобучение модели. Для исключения переобучения после 50 эпох требуется: увеличить объем исходных данных, упростить модель (на обучающих фотографиях уменьшить количество объектов) и увеличить регуляризацию (добавить количество ограничений на параметры модели).

Отметим, что все количественные значения метрик достаточно небольшие. Это объясняется малым объемом данных для обучения, сложностью объекта распознавания и нерелевантности данных. Например, в работе [16] для обучения распознавания трещин на бетонных конструкциях использовалось 50 000 фотографий обучающего набора данных, при этом точность обучения рассчитывается на основе 40 000 изображений, а проверка — на основе 10 000 изображений, не включенных в обучающий набор. В дальнейшем применялись еще 40 изображений для тестирования результатов обучения. С учетом этого обученная модель показала среднюю точность равную 92,35%, что значительно превышает показатели точности, полученные в нашей статье. Большая разница показателей связана с тем, что в нашем случае при сборе в набор исходных данных вошли фасады с различными материалами и отделкой (из кирпича разных цветов, из различных блоков, с отделкой штукатурным слоем и без и т.д.), а также в связи с более высокой неоднородностью кирпичных стен в сравнении с железобетонными и бетонными конструкциями. Отметим, что сбор набора данных для бетонных конструкций происходит значительно быстрее, так как авторы не ограничивали data set только вертикальными конструкциями, а также использовали фотографии плит перекрытия, бетонных полов, стяжки полов. Количество данных фотографий в открытом доступе существенно больше,



чем фасадов кирпичных зданий. Исходный набор данных собран для апробирования работоспособности разработанной программы. Более точные результаты можно получить путем увеличения обучающего набора данных.

С учетом вышесказанного проведена проверка качества обучения на одинаковой тестовой фотографии, которая не входила в набор данных для обучения. Время на определение трещин на тестовой фотографии составило не более 20 мс, что позволяет распознавать не менее 3 кадров в секунду. Планируется улучшить данный показатель до 8 кадров в секунду, что позволит определять дефекты на видео в режиме реального времени и без снижения качества с БПЛА. Но даже со скоростью 3 кадра в секунду в работе [16] также удалось обнаружить трещины на видео, снятом БПЛА в режиме реального времени, с точностью 88%. Авторами представлены фрагменты изображений с распознанными дефектами с различным количеством эпох обучения (рис. 4). Над рамками указаны значения вероятности наличия дефекта.



Рис. 4. Примеры распознавания дефектов с использованием сверточной нейронной сети с различным количеством эпох обучения (иллюстрация авторов)

Fig. 4. Examples of defect recognition using a deep fully convolutional neural network with a different number of learning epochs (illustration authors)

Отметим, что распознавание трещин, на фотографиях фасадов кирпичных зданий на данной стадии является приемлемым. Это позволит в дальнейшем избавиться от ручного труда экспертов в оформлении фотографий для отчетов обследований. Увеличит точность и качество определения дефектов, а также снизит влияние субъективности эксперта, так как что не увидел или не заметил эксперт будет указывать разработанная программа. При этом с увеличением количества эпох обучения вероятность нахождения дефекта также увеличивается, а рамки, выделяющие обнаруженные дефекты, более правильно и точно описывают дефект.

#### 4. Заключение.

1. Разработана программа для автоматизации процесса обнаружения дефектов на изображениях фасадов зданий, которая в дальнейшем позволит избавиться от ручного труда экспертов в оформлении фотографий для отчетов обследований. Это позволит увеличить точность и скорость определения дефектов и создания отчетов, а также снизит влияние фактора субъективности экспертов.

2. Описаны методы формирования обучающей выборки и процесса обучения сети с учетом особенностей решаемой задачи. Результаты тестирования показали, что применение архитектуры CNN с использованием библиотеки машинного обучения TensorFlow и модели YOLOv7 для обнаружения объектов, успешно даже при небольшом объеме обучающих данных. В дальнейшем, возможно, расширить перечень и скорость





распознавания определяемых дефектов что позволит облегчить внедрение БПЛА в обследования зданий. Также описанная модель может быть использована в других задачах определения объектов.

3. Определены показатели основных метрик качества обучения (время обучения, точность, полнота) при различном количестве эпох на основе алгоритма сверточной нейронной сети (CNN) на языке программирования Python с использованием библиотеки TensorFlow.

### Список литературы /References.

1. Кашеварова Г.Г., Тонков Ю.Л., Фурсов М.Н. Нечеткая экспертная система диагностики повреждений строительных конструкций // Вестник волжского регионального отделения Российской академии архитектуры и строительных наук (ВРО РААСН). 2014. №17. С. 167-173. [Kashevarova G.G., Tonkov Y.L., Fursov M.N. Fuzzy expert system for diagnosing damage to building structures // Bulletin of the Volga Regional Branch of the Russian Academy of Architecture and Building Sciences (VRO RAASN). 2014. No. 17. P. 167-173].

2. Cardellicchio A., Ruggieri S., Nettis A., Renò V., Uva G. Physical interpretation of machine learning-based recognition of defects for the risk management of existing bridge heritage // Engineering Failure Analysis. 2023. Vol. 149 (5). P. 1-59. DOI: 10.1016/j.engfailanal.2023.107237.

3. Наумов А.Е., Юдин Д.А., Долженко А.В. Совершенствование технологии проведения строительно-технических экспертиз с использованием аппаратно-программного комплекса автоматизированной дефектоскопии // Вестник БГТУ им. В.Г. Шухова. 2019. № 4. С. 61–69. DOI:10.34031/article\_5cb824d26344e7.458 99508. [Naumov A.E., Yudin D.A., Dolzhenko A.V. Improving the technology of conducting construction and technical expertise using a hardware and software complex of automated inspection // Bulletin of BSTU named after V.G. Shukhov. 2019. No. 4. P. 61–69. DOI: 10.34031/article\_5cb824d26344e7.45899508].

4. Кашеварова Г.Г. «Искусственный интеллект», или «логические рассуждения и разумные решения» в технической диагностике объектов строительства // Academia. Архитектура и строительство. 2023. № 4. С. 166–180. – DOI: 10.22337/2077-9038-2023-4-166-180. [Kashevarova G.G. «Artificial Intelligence» or «Logical Discussion and Reasonable Solutions» in Technical Diagnostics of Construction Projects // Academia. Architecture and Construction. 2023. No. 4. P. 166–180. DOI: 10.22337/2077-9038-2023-4-166-180].

5. Ahmadi, A., Khalesi, S., Bagheri, M. Automatic road crack detection and classification using image processing techniques, machine learning and integrated models in urban areas: A novel image binarization technique // Journal of Industrial and Systems Engineering. 2018. Vol. 11. P. 85–97.

6. D. Xie, L. Zhang, L. Bai. Deep learning in visual computing and signal processing // Applied Computational Intelligence and Soft Computing. 2017. Vol. 2017 (10). P. 1–13. DOI: 10.1155/2017/1320780.

7. Y. Liu, J. Yao, X. Lu, R. Xie, L. Li. Deepcrack: A deep hierarchical feature learning architecture for crack segmentation // Neurocomputing. 2019. Vol. 338. P. 139–153. DOI: 10.1016/2019/01.036.

8. Бузало Н.А., Кундрюцков Д.Н., Пономарев Р.Р. Применение беспилотных летательных аппаратов при обследовании зданий и сооружений // Строительство и архитектура. 2022. №1 (34). С. 18-24. DOI 10.12737/5953. [Buzalo N.A., Kundrutskov D.N., Ponomarev R.R. The use of unmanned aerial vehicles in the inspection of buildings and structures // 2022. No.1 (34). P. 18-24. DOI 10.12737/5953].

9. Улыбин А.В., Федотов С.Д. Качество визуального обследования зданий и сооружений и методика его выполнения // Строительство уникальных зданий и сооружений. 2014. №10 (25). С.134-136. [Ulybin A.V., Vatin N.I. The quality of the visual inspection of buildings and structures, and the method of its execution // Construction of Unique Buildings and Structures. 2014. No.10 (25)]



10. Крылов, С.А., Кашеварова Г.Г. Автоматизация диагностирования причин возникновения дефектов на фасадах кирпичных зданий с применением технологий искусственного интеллекта // Вестник Пермского национального исследовательского политехнического университета. Прикладная экология. Урбанистика. 2023. № 4. С. 51–62. DOI: 10.15593/2409- 5125/2023.04.05. [Krylov S. A., Kashevarova G. G. Automation of diagnosing the causes of defects on the facades of brick buildings using artificial intelligence technologies // PNRPU Bulletin. Applied ecology. Urban development. 2023. No. 4. P. 51-62. DOI: 10.15593/2409-5125/2023.04.05]

11. Корнев В.В., Орлова Н.С., Улыбин А.В., Федотов С.Д. Строительный контроль зданий и сооружений с применением мультикоптеров и фотограмметрии // Строительство уникальных зданий и сооружений. 2018. №2(65). С. 40-58. [Korenev V.V., Orlova N.S., Ulybin A.V., Fedotov S.D. Building inspection of buildings and structures by means of multicopters and photogrammetry // Construction of unique buildings and structures. 2018. No. 2(65). P 40-58].

12. Соболев Б.В., Соловьев А. Н., Васильев П. В., Подколзина Л. А. Модель глубокой сверточной нейронной сети в задаче сегментации трещин на изображениях асфальта // Вестник Донского государственного технического университета. 2019. №1. С. 63-73. [Sobol B. V., Soloviev A. N., Vasiliev P. V., Podkolzina L. A. Deep convolution neural network model in problem of crack segmentation on asphalt images // Vestnik of Don State Technical University. 2019. Vol. 19. No. 1. P. 63–73].

13. Chien-Yao Wang<sup>1</sup>, Alexey Bochkovskiy, Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors // IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021. P. 1-15. DOI: 10.1109/CVPR46437.2021.01283.

14. Powers, D. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness and Correlation // Journal of Machine Learning Technologies. 2011. Vol. 2 (1). P. 37-63.

15. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019. P. 658-666. DOI:10.48550/arXiv.1902.09630.

Kim B., Cho S. Automated vision-based detection of cracks on concrete surfaces using a deep learning technique // Sensors. 2018. Vol. 18. No. 10. P. 1-18. DOI: 10.3390/s18103452.

