

Адаптивная сверточная нейронная сеть для выявления нарушений техники безопасности

П.С. Медведев, В.В. Мошкин, Р.С. Шамсутдинов

Казанский национальный исследовательский университет им. А. Н. Туполева-КАИ

Аннотация: В статье представлена адаптивная сверточная нейронная сеть для автоматизированного обнаружения нарушений техники безопасности в реальном времени. В отличие от существующих решений, использующих статические модели, предлагаемый подход включает два ключевых нововведения. Автоматическая адаптация весов модели с комбинированием методов стохастического и градиентного спуска. Алгоритм динамически регулирует скорость обучения и глубину модификации параметров, что позволяет сохранять ранее полученные знания при дообучении на новых данных без деградации точности. Оптимизированный механизм обработки контекста – модель анализирует не только объекты (например, отсутствие каски), но и их взаимное расположение (рабочий в опасной зоне без средств индивидуальной защиты), что снижает количество ложных срабатываний. Разработанная система интегрирует модули компьютерного зрения, генерации предупреждений и аналитики, обеспечивая не только мгновенное реагирование на нарушения, но и долгосрочный анализ рисков. Эксперименты подтвердили повышение точности на 15% при работе в изменяющихся условиях освещения и ракурсах съемки.

Ключевые слова: сверточная нейронная сеть, информационная система, несчастные случаи на производстве, техника безопасности, производство, обучение модели, нейронная сеть, адаптивный алгоритм.

Введение

Обеспечение безопасности на производственных объектах является критически важной задачей, требующей постоянного внимания. Нарушение правил техники безопасности может привести к тяжелым последствиям, включая производственный травматизм и человеческие жертвы. Современные технологии компьютерного зрения, в частности сверточные нейронные сети, открывают новые возможности для создания интеллектуальных систем мониторинга соблюдения норм безопасности в реальном времени.

Разрабатываемая система представляет собой комплексное решение, включающее не только модуль компьютерного зрения, но и подсистемы визуализации данных, генерации предупреждений и формирования отчетности. Такая архитектура позволяет не только оперативно реагировать

на нарушения, но и проводить аналитику производственной безопасности за произвольные периоды времени.

Основная проблема нарушений техники безопасности на производстве

Современные производственные предприятия сталкиваются с критически важной задачей - обеспечением безопасности персонала. Несмотря на комплексный подход, включающий обязательное использование средств индивидуальной защиты (СИЗ) и регулярное обучение технике безопасности, проблема нарушений остается актуальной. Анализ показывает, что работники часто пренебрегают элементарными правилами защиты, что создает потенциально опасные ситуации и увеличивает риск производственного травматизма.

Таким образом, внедрение одноканальных сверточных нейронных сетей (you only look once – YOLO) в систему производственного мониторинга представляет собой качественно новый этап в обеспечении безопасности труда. Это не просто технологическое решение, а стратегический инструмент, способный кардинально снизить уровень травматизма и создать более безопасную среду для работников.

Архитектура YOLO

Архитектура YOLOv8 представляет собой качественно новый этап в развитии алгоритмов компьютерного зрения, предлагая принципиально иной подход к обработке изображений. В основе системы лежит мощный механизм извлечения признаков, построенный на усовершенствованной сверточной нейронной сети.

В целом, архитектура YOLOv8 обеспечивает высокую скорость и точность обнаружения объектов на изображениях, что делает ее одной из

самых продвинутых моделей для решения задачи обнаружения объектов в реальном времени (рис. 1).

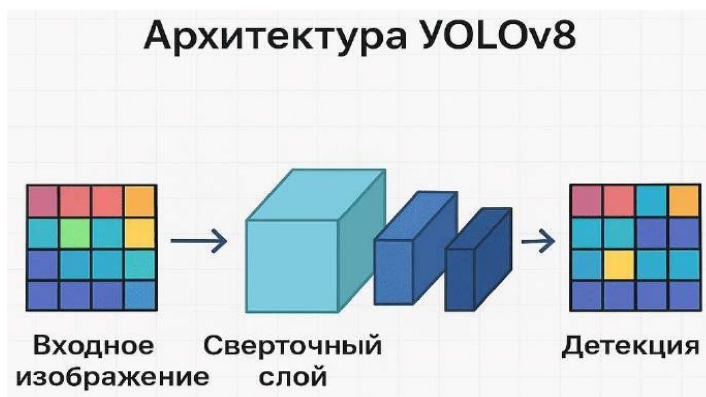


Рис. 1. – Архитектура нейронной сети, с применением модели YOLOv8

Набор данных и обучение. На производительность YOLOv8 также влияют набор данных, используемый для обучения, и сам процесс обучения. Хорошо оптимизированный процесс обучения с разнообразным и репрезентативным набором данных может привести к созданию модели, которая будет лучше работать при распознавании объектов [1].

Важно отметить, что YOLOv8 имеет достаточно хорошую производительность по сравнению с YOLOv5, был проведен анализ, в котором мы сравнили производительность результаты которого представлены в табл. 1.

Таблица №1

Сравнительные характеристики точности распознавания и сегментации объектов распознавания

Модель	Распознавание в %	Сегментация	Классификация
Nano	+33.21%	+32.97%	+3.10%
Small	+20.05%	+18.62%	+1.12%
Medium	+10.57%	+10.89%	+0.66%
Large	+7.96%	+6.73%	+0.00%
Xtra Large	+6.31%	+5.33%	-0.76%

Все представленные модели на рисунке 3 имеют один и тот же алгоритм, отличаются лишь некоторыми параметрами, которые приведены в сравнительной табл. 2.

Был проведен сравнительный анализ моделей YOLOv8 по результатам которого был выбран лучший вариант по параметрам. Исходя из таблицы видно, что по практически всем параметрам лучше модель Xtra Large благодаря емкости модели и ее предобученности, поэтому было принято использовать именно эту версию модели yolo [2].

Проведем сравнительный анализ, в котором сравним модель YOLOv5 с Nano версией модели YOLOv8 по времени обучения. Обучать модель будем до момента пока модель не научится определять определенный объект на видео с точностью 93%.

Таблица № 2

Сравнительные характеристики модели YOLOv8

Модель	Размерность	maP	Map(val) 0,5	Скорость b1 CPU (ms)	Параметры
Nano	820	46,0	47,3	49	2,1
Small	867	48,4	53,3	54	3,2
Medium	881	53,4	58,1	120	10,2
Large	980	82,1	83,4	670	50,8
Xtra Large	1113	102,8	108,9	920	97,8

В результате сравнительного анализа моделей выяснилось, что YOLOv8 обучается заметно быстрее. Результаты сравнительного анализа можно увидеть на рис. 2.



Рис. 2. – Сравнение моделей yolov8 и yolov5s при обучении

Из рис. 2 можно сделать вывод, что yolov8 имеет лучшее время вывода в сравнении с yolov5.

Описание обучения нейронной сети с помощью YOLOv8

Основной этап обучения строится на последовательной обработке обучающих изображений сложной архитектурой нейронной сети. Каждое изображение проходит через многослойную систему преобразований, включающую различные типы нейронных слоёв. Особенностью YOLOv8 является способность параллельно анализировать различные аспекты изображения, что позволяет достичь высокой скорости обработки без потери точности [3].

На рис. 3 представлены графики процесса обучения нейронной сети и ее ошибок.

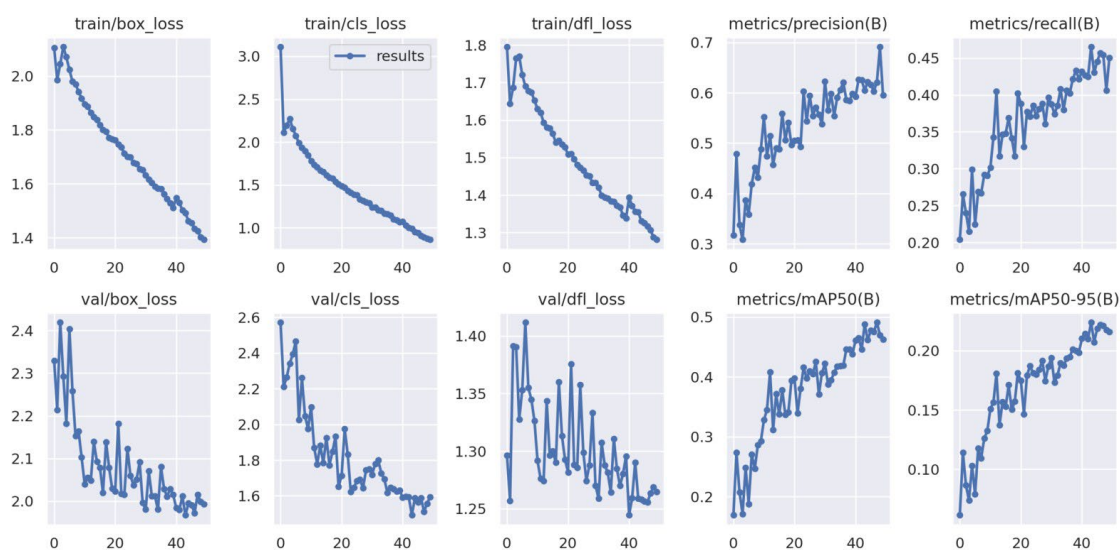


Рис. 3. – Графики процесса обучения нейронной сети и ее ошибок

Адаптивная настройка параметров модели YOLOv8 на этапе обучения

Веса модели, являющиеся параметрами, которые определяют, как модель машинного обучения преобразует входные данные в предсказания, играют ключевую роль в точности распознавания [4].

В процессе обучения модель корректирует веса, чтобы минимизировать ошибку предсказания. Чем лучше подобраны веса, тем точнее модель будет распознавать образцы.

На рис. 3 показаны веса в общей структуре YOLO, где представлены все слои и веса в модели YOLO. Веса сверточных слоев содержат набор фильтров, которые применяются к входному изображению или к выводу предыдущего слоя. Веса фильтров определяют, как каждый фильтр будет извлекать признаки.

Новизна обучения заключается в том, что в процессе обучения автоматически подбираются параметры весов (рис. 4).

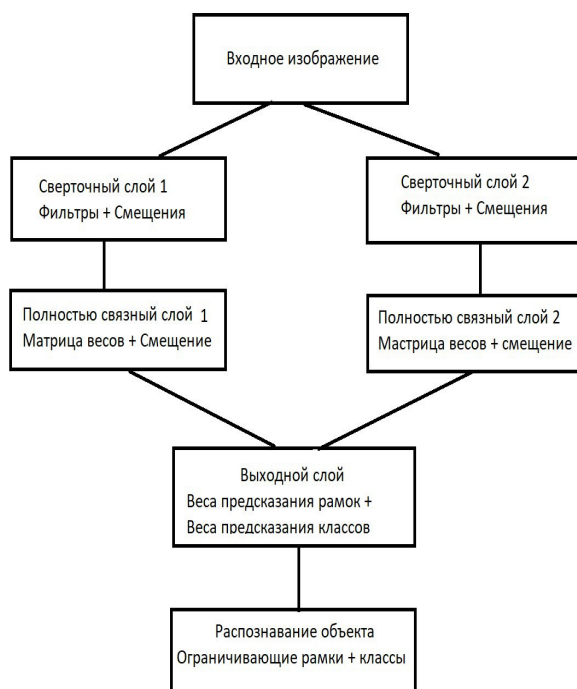


Рис. 4. – Веса в общей структуре YOLO

Если модель должна адаптироваться к новым данным, мы используем методы, такие как *fine-tuning*, где только некоторые веса корректируются, чтобы избежать потери уже полученных знаний. Чтобы обновление весов было более эффективным мы используем алгоритмы оптимизации SGD, которые помогают более оптимально настроить параметры весов. Алгоритм SGD – это один из алгоритмов оптимизации, которые используется в машинном обучении для настройки весов модели чтобы минимизировать функцию потерь. Работает алгоритм следующим образом: для каждой эпохи выбирается пакет, затем вычисляется градиент и обновляются веса [5].

Алгоритм автоматической настройки параметров показан на рис. 5, был реализован с помощью метода градиентного спуска для оптимизации обучения. В данном алгоритме мы определяем функцию потерь, которую хотим минимизировать в процессе обучения YOLOv8 [6].

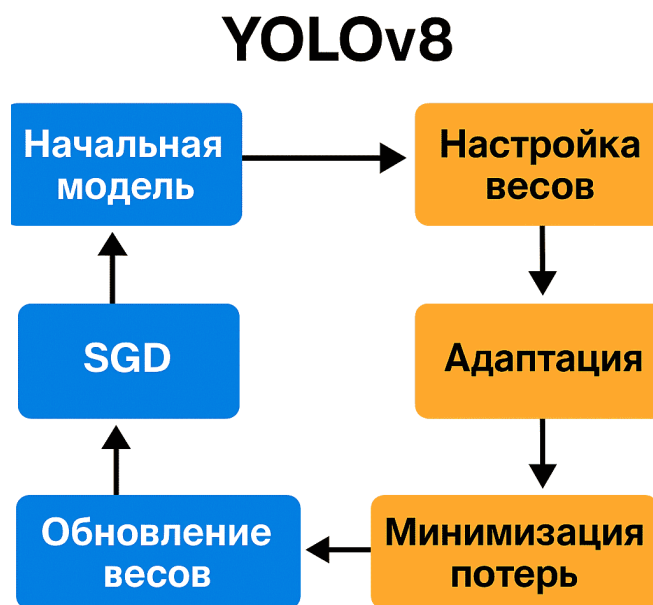


Рис. 5. – Обновленная схема настройки весов

У нас это будет функция которая учитывает ошибки обнаружения объектов на изображениях. Далее мы инициализируем параметры модели YOLOv8 случайными значениями. После этого запускается цикл обучения. На рис. 6 показан график оптимального подбора весов в нейронной сети. Ось X – это количество эпох, ось Y – это функция потерь.

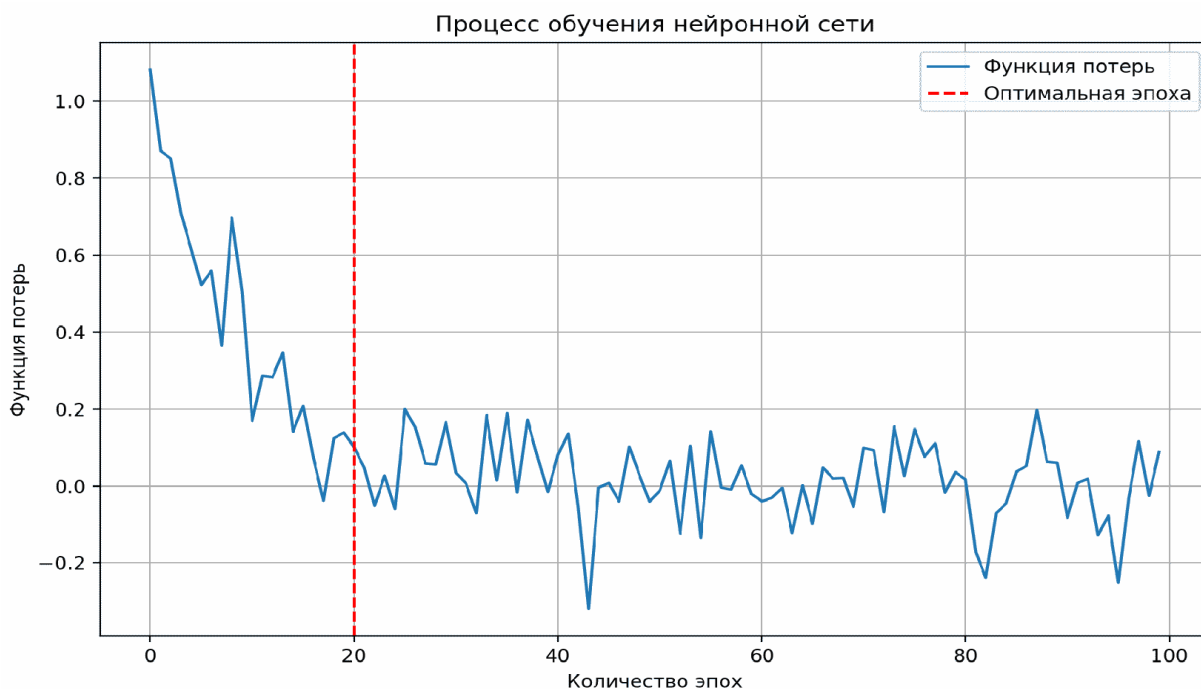


Рис. 6. – График оптимального подбора весов в нейронной сети

Красной пунктирной линией показана оптимальная эпоха, на которой обучение следует остановить [7]. Это может быть точка, где функция потерь достигает минимума или где дальнейшее обучение будет практически бесполезно.

$$\eta_t = \eta_f \cdot \left(\frac{t}{T} \right) \quad (1)$$

где η_t – текущая эпоха,

η_f – финальный learning rate,

t - текущая эпоха,

T - общее число эпох.

формула обучения Формула масштабирования весов:

$$w' = w \cdot \left(\frac{B}{N} \right) \quad (2)$$

где w' – масштабированный вес,

w - исходный вес,

B - размер батча,

N - число батчей для аккумуляции.

Формула аккумуляции градиентов:

$$\nabla' = \nabla \cdot \left(\frac{B_n}{B_a} \right) \quad (3)$$

где ∇' – скорректированный градиент,

∇ - исходный градиент,

B_n - номинальный размер батча,

B_a - фактический размер батча.

Цикл обучения: Вычисление градиента функции потерь по параметрам модели, обновление параметров модели в направлении, противоположном градиенту, с учетом индекса обучения, повторение шагов до достижения

сходимости или заданного числа итераций. Для анализа были взяты следующие значения весов для каждой из моделей (табл. 2).

Для более детального анализа мы сравнили разные параметры весов в различных показателях. На рис. 7 показан график, на котором видно, как меняется точность при конфигурациях весов. Чем выше точность, тем меньше ложных срабатываний у модели.

Таблица № 2

Сравнительные характеристики весов моделей

Модель	Глубина сети	Ширина сети	Размер выходного изображения	Выходные слои	
YOLOv8n	0.34	0.51	640x640	0.34	0.51
YOLOv8s	0.33	0.50	640x640	0.33	0.50
YOLOv8m	0.43	0.30	640x640	0.43	0.30
YOLOv8l	0.73	0.50	640x640	0.23	0.85
YOLOv8x	0.73	0.90	640x640	0.13	0.77

Полнота измеряет процент истинных положительных объектов модель смогла распознать, насколько хорошо и точно модель выявляет все объекты на видео или изображении. Полнота является наиболее важным параметром, когда нам важно обнаружить каждый объект и не допустить ложного распознавания объекта [8, 9]. Как видно из рис. 7 наиболее высокая полнота достигается при значении весов YOLOv8x.

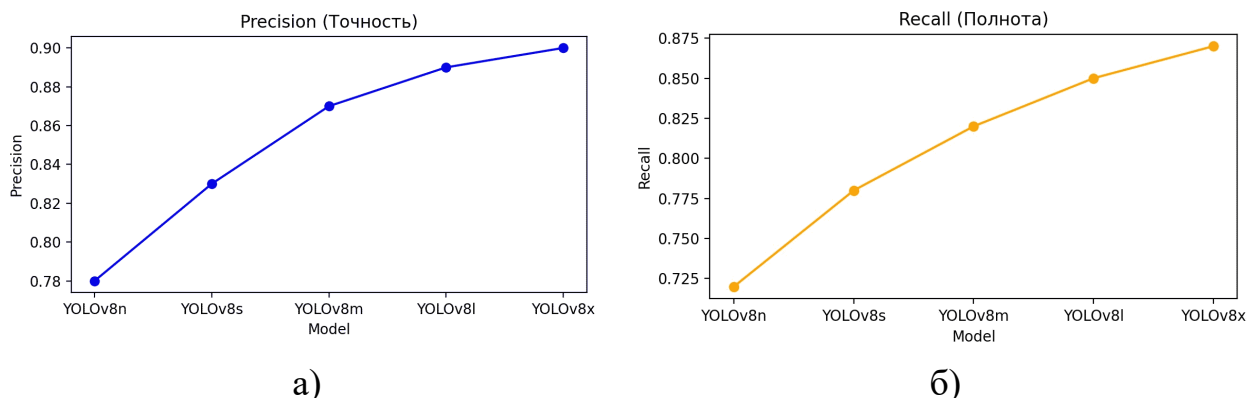


Рис. 7. – а) Точность предсказания при разных настройках весов, б) Полнота при разных показателях весов

Так же мы сравнили количество кадров в секунду (FPS) при разных конфигурациях весов в модели YOLOv8. На рис. 8 видно, что чем сложнее модель и выше параметры весов, тем меньше кадров в секунду, но выше точность распознавания [10]. На рис. 8 видно, что объем видеопамати (VRAM) используется больше при больших, сложных моделях в сравнении с менее сложными моделями YOLOv8.

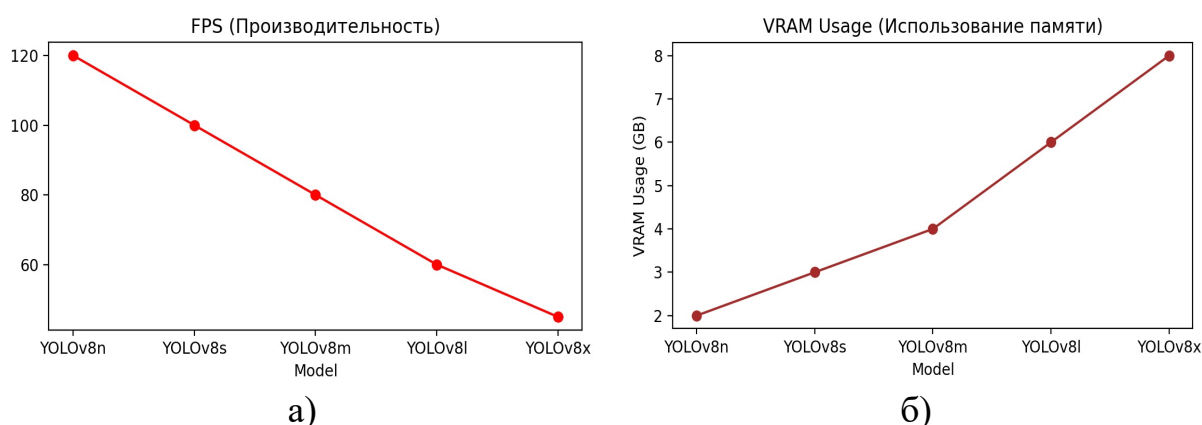


Рис. 8. – а) Количество кадров в секунду при разных параметрах весов, б) Использование видеопамати при разных моделях YOLOv8

Для вычисления координат (x, y) используется специальная параметрическая функция, график которой демонстрирует характерную S-образную форму, обеспечивающую плавную нормализацию выходных значений [11]. На рис. 9 показан график используемой функции активации.

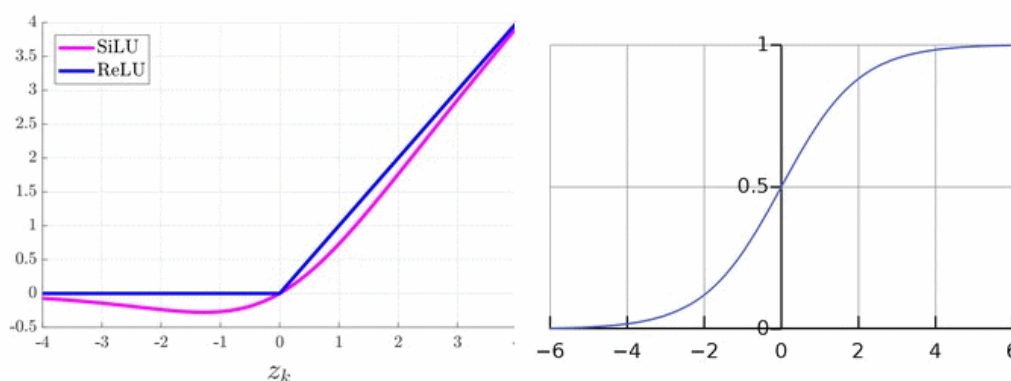


Рис. 9. – График используемой функции активации

SILU является функцией активации взмаха, данная функция используется в скрытых слоях [12-14]. Данная функция применена при

обучении модели [12-14]. Формула функции активации имеет следующий вид:

$$f(x) = \frac{e^{x_i}}{\sum_{k=1}^n e^{x_i}} \quad (4)$$

Модель yolo предсказывает 5 параметров для каждой клетки [12].

Расчет происходит по формулам [12]

$$b_x = q(t_x) + C_x \quad (5)$$

$$b_y = q(t_y) + C_y \quad (6)$$

$$b_w = b * w_{et} \quad (7)$$

$$b_h = b * \frac{e_h}{t_h} \quad (8)$$

t_x, t_y - это предсказания значений YOLO,

p_x, p_y - это ширина и высота anchor box,

b_x, b_y, b_h, b_w - это значения предсказанного bounding box.

Так же была при обучении была использована функция потерь. Модель нам возвращает классы объектов которые были обнаружены в ограничивающих рамках на фото или видео. Для этого используется формула для вычисления потери классов, потери объектности, а так же потери местоположения. Функция потерь имеет следующую формулу:

$$Loss = -\log(Y_{pred}) \quad (9)$$

где Y_{pred} -предсказанные значения

В качестве линейного фильтра был использован фильтр Калмана, называют линейным потому что отфильтрованное значение постоянно зависит от предыдущих состояний сенсора. Фильтр Калмана рассчитывается по следующим формуле:

$$KG = \frac{E_{est}}{E_{est} + E_{mea}} \quad (10)$$

где E_{est} – является погрешностью значения ,

E_{mea} – является ошибкой измерения,

KG – коэффициент Калмана.

Заключение

Была внедрена интеллектуальная система видеомониторинга, основанная на передовых технологиях компьютерного зрения. В основе решения лежит усовершенствованная модель YOLOv8x, демонстрирующая высокую эффективность в задачах промышленной безопасности (рис. 10) с показателем точности 0.87.



Рис. 10. – Работа информационной системы на предприятии

Архитектура нейронной сети была оптимизирована под специфические производственные задачи с параметрами глубины (0.73) и ширины (0.90) сети, что позволило достичь оптимального баланса между скоростью обработки и точностью детекции. Особое внимание было уделено калибровке выходных слоев, где значения глубины и ширины составили 0.13 и 0.77 соответственно, обеспечивая стабильную работу с изображениями разрешением 640×640 пикселей. На рис. 10 показана работа информационной системы на предприятии.

Литература

1. Zoph, B., Ghiasi, G., Lin, T.-Y., et al. Data Augmentation for Object Detection: A Comparative Study. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.
2. Гуриков С. Р. Основы алгоритмизации и программирования на Python. Москва: Форум, 2018. 991 с.
3. Аль-Хашеди А. А., Обади А. А., Нуриев Н. К. Разработка математического и программного обеспечения задач распознавания образов на основе персептрона // Вестник Казанского технологического университета, 2017. Т. 20. № 11. С. 85–88.
4. Андреас М. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными: моногр. Москва: Альфа-книга, 2017. 697 с.
5. Васильев А. Н. Python на примерах. Практический курс по программированию. Москва: Наука и техника, 2016. 432 с.
6. Галушкин А. И. Нейронные сети: основы теории. М.: РиС, 2015. 496 с.
7. Яхьяева Г. Э. Нечеткие множества и нейронные сети: Учебное пособие. М.: Бином. Лаборатория знаний, 2008. 316 с.
8. Барский А. Б. Логические нейронные сети: Учебное пособие. М.: Бином, 2013. 352 с.
9. Каллан Р. Нейронные сети: Краткий справочник. М.: Вильямс И. Д., 2017. 288 с.
10. Комашинский В. И., Смирнов Д. А. Нейронные сети и их применение в системах управления и связи. М.: ГЛТ, 2003. 94 с.
11. Medvedev P., Mokshin V. Computer Vision Method for Detecting Safety Violations Using the YOLOv8 Model // 2024 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). Sochi, Russian Federation, 2024. P. 1095–1100. – DOI: 10.1109/ICIEAM60818.2024.10554051.

12. V. Mokshin, A. Sultanova and L. Sharnin. Using Convolutional Neural Networks to Monitor Security at an Industrial Facility, 2022 VIII International Conference on Information Technology and Nanotechnology (ITNT), Samara, Russian Federation, 2022. pp. 1-6, doi: 10.1109/ITNT55410.2022.9848644.

13. Никитин Д.В., Тараненко И.С., Катаев А. В. Детектирование дорожных знаков на основе нейросетевой модели YOLO // Инженерный вестник Дона, 2023. № 7. URL: ivdon.ru/ru/magazine/archive/n7y2023/8531

14. Земцов А.Н., Кузнецов М.А., Садек Сажжад, Попов В.К., Ступницкий П.С. Автоматическое распознавание автомобильных номерных знаков в автомобильной самоорганизующейся сети // Инженерный вестник Дона, 2023. №12. URL: ivdon.ru/ru/magazine/archive/n12y2023/8880

References

1. Zoph, B., Ghiasi, G., Lin, T.-Y., et al. Data Augmentation for Object Detection: A Comparative Study. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

2. Gurikov S. R. Osnovy` algoritmizacii i programmirovaniya na Python [Basics of Algorithmization and Programming in Python]. Moskva: Forum, 2018. 991 с.

3. Al`-Xashedi A. A., Obadi A. A., Nuriev N. K. Vestnik Kazanskogo texnologicheskogo universiteta, 2017. Vol. 20 (11). P. 85–88.

4. Andreas M. Vvedenie v mashinnoe obuchenie s pomoshh`yu Python. Rukovodstvo dlya specialistov po rabote s danny`mi: monogr [Introduction to Machine Learning with Python: A Guide for Data Scientists: Monograph]. Moskva: Al`fa-kniga, 2017. 697 p.

5. Vasil`ev A. N. Python na primerax. Prakticheskij kurs po programmirovaniyu [Python by Examples. Practical Programming Course]. Moskva: Nauka i texnika, 2016. 432 p.

6. Galushkin A. I. Nejronny`e seti: osnovy` teorii [Neural Networks: Basic Theory]. M.: RiS, 2015. 496 p.
7. Yax`yaeva G. E`. Nechetkie mnozhestva i nejronny`e seti: Uchebnoe posobie [Fuzzy Sets and Neural Networks: A Tutorial]. M.: Binom. Laboratoriya znaniy, 2008. 316 p.
8. Barskij A. B. Logicheskie nejronny`e seti: Uchebnoe posobie [Logical neural networks: A tutorial.]. M.: Binom, 2013. 352 p.
9. Kallan R. Nejronny`e seti: Kratkij spravochnik [Neural Networks: A Quick Reference]. M.: Vil`yams I. D., 2017. 288 p.
10. Komashinskij V. I., Smirnov D. A. Nejronny`e seti i ix primeneniye v sistemax upravleniya i svyazi [Neural networks and their application in control and communication systems]. M.: GLT, 2003. 94 p.
11. Medvedev P., Mokshin V. 2024 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). Sochi, Russian Federation, 2024. P. 1095–1100. DOI: 10.1109/ICIEAM60818.2024.10554051.
12. V. Mokshin, A. Sultanova and L. Sharnin. 2022 VIII International Conference on Information Technology and Nanotechnology (ITNT), Samara, Russian Federation, 2022, pp. 1-6, doi: 10.1109/ITNT55410.2022.9848644.
13. Nikitin D.V., Taranenko I.S., Kataev A. V. Inzhenernyj vestnik Dona. 2023. № 7. URL: ivdon.ru/ru/magazine/archive/n7y2023/8531
14. Zemczov A.N., Kuznecov M.A., Sadek Sazhzhad, Popov V.K., Stupniczkij P.S. Inzhenernyj vestnik Dona, 2023. №12. URL: ivdon.ru/ru/magazine/archive/n12y2023/8880.

Дата поступления: 23.04.2025

Дата публикации: 29.06.2025