

УДК 004.932

DOI 10.52928/2070-1624-2024-43-2-24-33

АНАЛИЗ МЕТОДОВ ОПРЕДЕЛЕНИЯ АБСОЛЮТНОГО РАССТОЯНИЯ ДО ОБЪЕКТА ПО ИЗОБРАЖЕНИЮ С ОДНОЙ ВИДЕОКАМЕРЫ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

*Н. Л. ЛУПЕНКО, д-р техн. наук, доц. Р. П. БОГУШ
(Полоцкий государственный университет имени Евфросинии Полоцкой);*

*Ph. D. X. ЧЕН
(университет Чжэцзян Шурен, Китай)*

Рассмотрен ряд подходов для определения расстояния до объекта по изображению, сформированному монокулярной видеокамерой, которые применяют искусственные нейронные сети на различных этапах обработки. Проанализирован метод, основанный на нахождении карты глубины и обнаружении объекта с последующим проецированием его координат на карту глубины. Описан метод, использующий взаимосвязь реального размера определенного класса объекта и его размера на сформированном изображении. Рассмотрен подход, основанный на модификации нейронной сети YOLO, при котором в результирующей дескриптор включает дополнительный вектор, характеризующий расстояние до объекта, а также изменяется функция потерь. Описаны состав и особенности наборов данных, которые используются для обучения нейронных сетей, применяемых в алгоритмах вычисления абсолютного расстояния до объекта по изображению. Представлены результаты оценки эффективности различных подходов, описаны их преимущества и недостатки, а также перспективы применения при решении практических задач.

Ключевые слова: *определение расстояния, нейронная сеть, компьютерное зрение, определение объекта на изображении, карта глубины.*

Введение. С развитием аппаратных средств расширяется спектр практических задач, которые могут быть решены с использованием методов компьютерного зрения [1]. Определение расстояния до объекта необходимо для эффективной ориентации и навигации устройств в трехмерном пространстве во множестве различных областей, например, в сфере автономного вождения, в робототехнике и т. п. Для решения подобной задачи существует значительное количество подходов, которые можно разделить на две основные категории: с использованием активного и пассивного оборудования.

Активное оборудование включает в себя такие устройства, как лидар, сонар и т. п. Они точны и просты в использовании, поскольку представляют информацию о расстоянии как множество точек [2], но у них есть ряд недостатков, основными из которых являются высокая стоимость и небольшая дальность [3]. Пассивное оборудование, такое как RGB- или инфракрасные камеры, дешево и может работать для большего расстояния, чем лидар, но требует наличия сложного алгоритма компьютерного зрения для определения расстояния. Особое внимание стоит уделить подходу, основанному на использовании одной монокулярной видеокамеры, поскольку он является наиболее дешевым и доступным, что позволяет легко внедрять его практически в любые сферы. Однако этот метод также является самым сложным с точки зрения обеспечения достаточной точности.

Развитие нейронных сетей значительно расширяет и упрощает решение многих задач компьютерного зрения [4; 5]. В связи с этим данная статья посвящена анализу эффективности и возможности практического применения современных архитектур нейронных сетей и алгоритмов на их основе к задаче определения расстояния до объектов по изображениям с видеокамеры.

Определение расстояния до объекта на основе его координат и карты глубины. Принцип работы данного решения основан на параллельном использовании двух нейронных сетей: одной для обнаружения объектов и другой для оценки расстояния от видеокамеры до поверхности объекта. Карта глубины формируется в результате применения модели DepthNET [6].

Модель DepthNET основана на применении двух нейронных сетей. Одна из них служит для формирования карты глубины, а другая – для оценки позы изображения. Обе сети построены по принципу автокодировщика и состоят из последовательных кодера и декодера. Предварительно обученные веса модификаций моделей ResNet использовались для извлечения признаков [7]: ResNet-50 в качестве основной для прогнозирования глубины, а ResNet-18 – для оценки позы. В обеих сетях перед первым слоем данные проходят свертку (convolution), пакетную нормализацию (batch normalization) и операцию подвыборки (max-pooling). Декодер же состоит из обратной свертки (deconvolution) и операции увеличения разрешения (upsampling). Также для повышения точности определения глубины изображения в данной работе использовалась GCN (Graph Convolutional Network), которая обеспечивает правильное определение сходства пикселей и нахождение связи

между ними. В итоге на выходе мы получаем карту глубины изображения. Стоит отметить, что в данном решении значение глубины находится в интервале от 0 до 100 метров.

Обнаружение объектов выполняется с помощью быстрой и точной модели YOLOv5¹, которая формирует ограничивающий прямоугольник (bounding box) для объекта (значит, известны его координаты в пределах изображения) и определяет класс объектов. Далее границы объекта проецируются на карту глубины и находится относительное расстояние до него как медианное значение расстояния до каждого пикселя в границах объекта. На последнем шаге относительное расстояние преобразовывается в абсолютное.

В работе [8] показано (рисунок 1), что если наблюдаемое видеокамерой пространство пересекается с поверхностью, на которой располагается объект, то найти абсолютную дистанцию до объекта можно с помощью формулы

$$y = h * \tan\left(\beta + \frac{2\alpha(v)}{n-1}\right), \tag{1}$$

- где h – высота камеры;
 β и α – углы, показанные на рисунке 1;
 (u, v) – координаты точки в плоскости изображения;
 (m, n) – разрешение изображения;
 (x, y) – реальные координаты объекта.

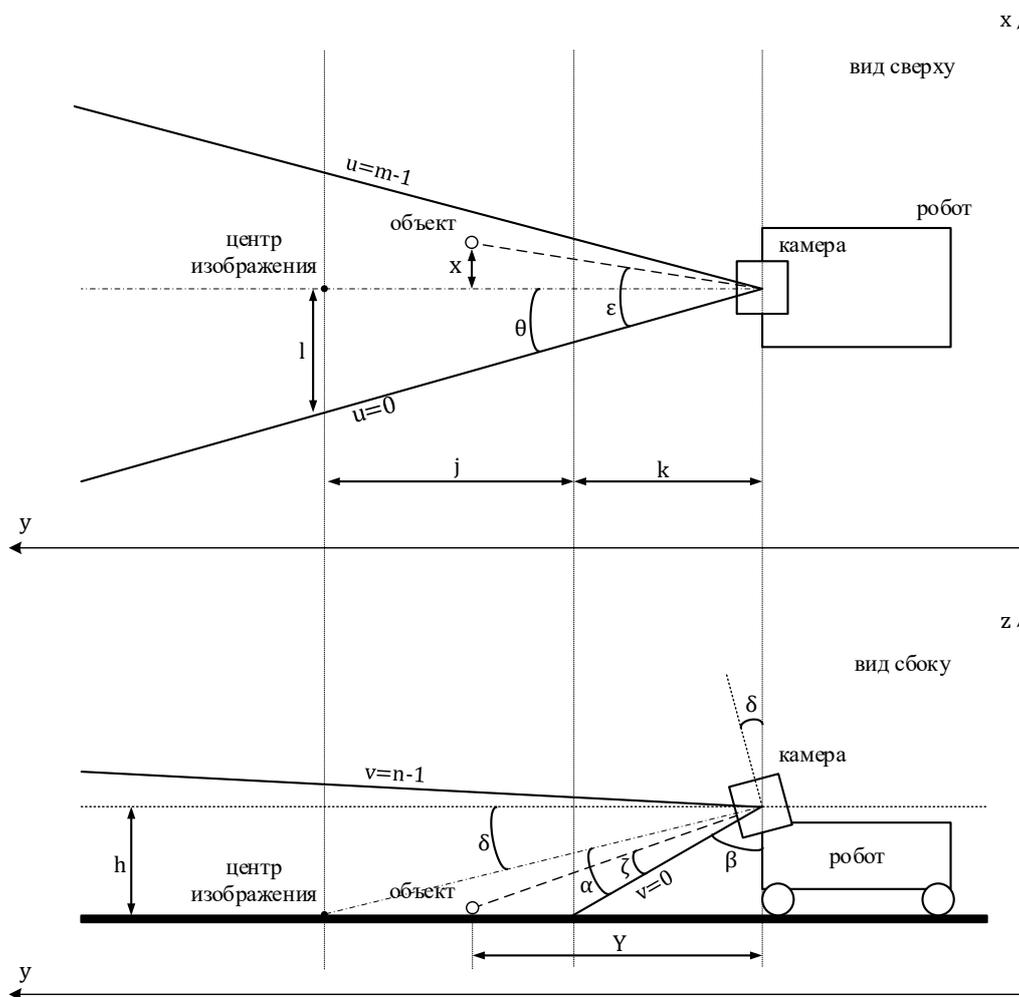


Рисунок 1. – Схема размещения видеокамеры для формирования изображения при определении расстояния

¹ Comprehensive Guide to Ultralytics YOLOv5 [Электронный ресурс]. – URL: <https://docs.ultralytics.com/yolov5> (дата обращения 30.08.2024).

В работе [6] выражение (1) преобразуется в квадратичную формулу для перевода относительного расстояния в абсолютное:

$$Y = (c_0 + c_1 X + c_2 X^2) * h, \quad (2)$$

где c_0, c_1, c_2 – коэффициенты, которые можно получить с помощью метода наименьших квадратов;

h – высота камеры;

X – относительное расстояние;

Y – абсолютное расстояние.

Для нахождения коэффициентов были выбраны 10 различных изображений, в которых расстояние до объектов известно. После определения относительного расстояния и сопоставления его с абсолютным был получен график зависимости данных величин (рисунок 2). Методом приближения с помощью кривых (curve fitting) и методом наименьших квадратов (least squares) были получены необходимые коэффициенты. Тогда

$$Y = 0,0036X^2 - 0,5373X + 21,714. \quad (3)$$

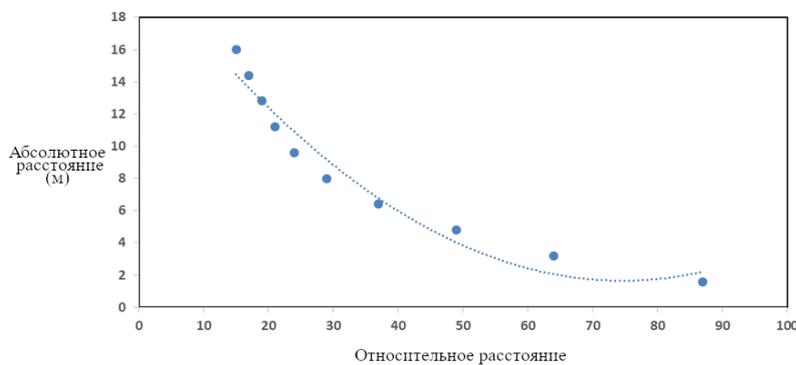


Рисунок 2. – График зависимости абсолютного и относительного расстояний

Известно, что эффективность нейронных сетей в значительной мере определяется характеристиками набора данных для ее обучения. При реализации метода определения расстояния до объекта на основе его координат и карты глубины было использовано два набора данных KITTY [9] и COCO [10]. Набор данных KITTY [9], который создавался для построения алгоритмов автономного вождения, состоит из изображений и лидарных данных, применяемых в исследованиях по компьютерному зрению в таких областях, как стереовидение, оптический поток, визуальная одометрия, обнаружение 3D-объектов и 3D-трекинг. Из данного набора были использовано 39 810 изображений для обучения, 4424 для валидации и 697 для тестирования. Все изображения были сняты с монокулярной камеры. Разрешение – 320×1024 пикселей.

Широко известным и применяемым для тренировки и тестирования нейронных сетей является набор изображений COCO [10], который используется при решении задач обнаружения объектов и сегментации. Он содержит 330 тыс. изображений (> 200 тыс. с маркерами), 1,5 млн экземпляров объектов, 80 категорий объектов. Из COCO-датасета было использовано 165 482 изображений для обучения, 81 208 для валидации и 81 434 для тестирования. Разрешение изображений – 640×480 пикселей.

Параметры обучения модели: тренировка – 20 эпох, размер пакета (batch size) – 10, скорость обучения (learning rate) – 0,001, оптимизатор – Adam. Обучение заняло 60 ч с использованием одной видеокарты GTX 1080 TI. В данной работе код для определения глубины был реализован с помощью Pytorch, в реализации обнаружения объектов на YOLOv5 также использовалась данная библиотека. Следует отметить, что для уменьшения вычислительной сложности была выбрана предварительно обученная модель YOLOv5s.

В ходе тестирования эффективность данного метода определения абсолютного расстояния до объекта была проверена с использованием 100 изображений с различными классами объектов, такими как человек, автомобиль, стул и т. д. Для оценки предлагаемой системы использовались две стандартные метрики: точность (accuracy) и среднеквадратичная ошибка (RMSE – Root Mean Square Deviation). Точность использовалась для оценки погрешности в соответствии с заданным пороговым значением, что служит показателем того, насколько часто предсказанное значение оказывается правильным. Пороговое значение, по сути, представляет собой ожидание того, что абсолютная погрешность определения расстояния до данного объекта в сцене ниже порогового значения T (в данной работе пороговое значение – 0,2 м). В ходе тестирования определено, что данная система обеспечивает точность и среднее значение RMSE, равные 96 % и 0,203 м соответственно. Для четырех изображений с известным расстоянием до объектов результаты представлены в таблице 1 [6].

Таблица 1. – Результаты эффективности метода определения расстояния с использованием нейронных сетей DepthNET и YOLO5

Изображение	Объект	Абсолютная дистанция, м	Предсказываемая дистанция, м	Погрешность, м
	Автомобиль	53,9	53,21	0,69
	Человек	21,5	21,35	0,15
	Автобус	48,7	48,13	0,57
	Стул	3,5	3,45	0,05
	Человек	8,0	8,09	0,09
	Автомобиль	10,1	9,83	0,27
	Человек 1	8,0	8,13	0,13
	Человек 2	12,0	11,69	0,31
	Человек 3	4,0	3,88	0,12

Определение расстояния до объекта на основе оценки его размеров на изображении. В основе данного метода лежит нейронная сеть DisNET [10], построенная на многослойной архитектуре (Multi Hidden-Layer Neural Network). Для определения положения объекта на изображении данный метод использует модель YOLOv3 [11]. Схема работы данного решения показана на рисунке 3.

На вход нейронная сеть DisNET получает информацию о фактических и средних размерах объекта, представленную в виде вектора v :

$$v = \left[\frac{1}{B_h} \frac{1}{B_w} \frac{1}{B_d} C_h C_w C_b \right], \tag{4}$$

где B_h, B_w, B_d – значения высоты, ширины и диагонали ограничивающего прямоугольника, измеряемые в пикселях;

C_h, C_w, C_b – средние значения высоты, ширины и глубины объекта определенного класса, измеряемые в сантиметрах. Средние значения задаются вручную и известны для каждого определяемого класса объекта, например, C_h, C_w и C_b для класса «человек» равны 175, 55 и 30 см соответственно, а для класса «автомобиль» – 160, 180 и 400 см.

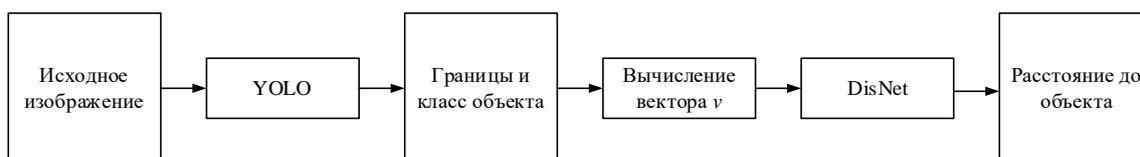


Рисунок 3. – Общая схема метода определения расстояния до объекта на основе оценки его размеров на изображении

На рисунке 4 представлены принятые отношения между значениями размеров ограничивающего прямоугольника и расстоянием до объекта. По мере удаления от камеры фактический размер объекта на изображении уменьшается. Следовательно, числа, обратные значениям B_h , B_w и B_d , будут увеличиваться. Из графиков видно, что расстояние до объекта и обратные размеры ограничивающего прямоугольника образуют практически линейную зависимость.

Структура самой модели DisNET (рисунок 5) состоит из 3 скрытых слоев (hidden layer), которые содержат по 100 нейронов каждый. Данные значения были получены экспериментальным путем. Входной слой (input layer) содержит 6 нейронов, которые соответствуют значениям из вектора v согласно выражению 4. Выходной слой (output layer) содержит один нейрон, который формирует предсказанное расстояние между камерой и объектом.

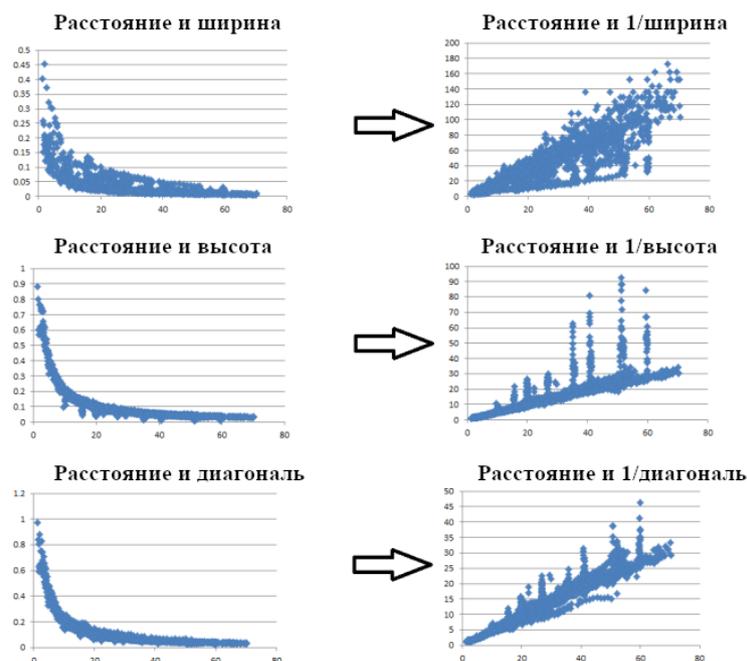


Рисунок 4. – Зависимости между расстоянием и размерами объекта на изображении

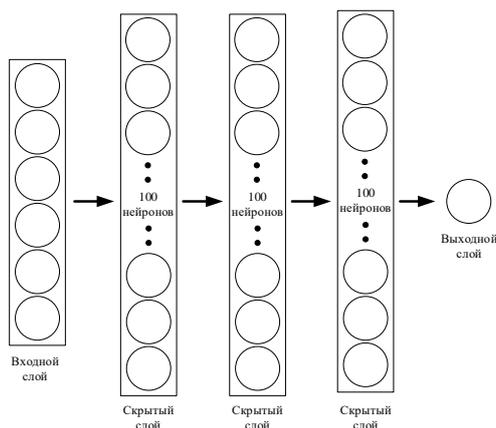


Рисунок 5. – Архитектура DisNET

Для обучения модели DisNET использовался датасет, состоящий из 2000 векторов, которые представляются согласно выражению 4. Он был разделен таким образом: 80 % – обучающие данные, 10 % – тестовые данные и 10 % – валидационные данные. Оценка эффективности работы данной модели для двух изображений (примеров) может быть проведена на основе анализа таблицы 2. Реализация данного решения доступна на странице GitHub².

² URL: <https://github.com/guanjianyu/DisNet> (дата обращения 03.09.2024).

Таблица 2. – Результаты работы модели DisNET

Номер изображения	Объект	Абсолютная дистанция, м	Предсказанная дистанция, м	Погрешность, м
1	Человек 1	100	101,89	1,89
	Человек 2		99,44	0,56
2	Человек 1	50	54,26	4,26
	Человек 2	150	167,59	17,59
	Человек 3	100	132,26	32,26
	Человек 4	300	338,51	38,51

На первом изображении было два человека, которые располагались на расстоянии 100 метров от видеокамеры, на втором – четыре человека, находившихся на различных дистанциях от видеокамеры. Изображение второго примера представлено на рисунке 6.



Рисунок 6. – Изображение примера 2

Определение расстояния до объекта на основе модификации архитектуры нейронной сети. Метод основан на изменении архитектуры YOLOv3 с целью включения в предсказываемые значения данной нейросетью вектора расстояния, а кроме того, предполагает модификацию функции потерь. Для обучения используется такой же датасет, как и для YOLOv3, но с добавлением абсолютной дистанции. Таким образом, можно сформулировать ключевые отличия:

1. Расширение вектора предсказываемых значений.
2. Модификация функции потерь, чтобы учесть дистанцию до объекта.
3. Дополнение обучающего датасета информацией о дистанции до объектов.

В YOLOv3 вектор предсказываемых значений p описывается как $p = (b, c, o)$, где $b = (x, y, w, h)$ – вектор координат ограничивающего прямоугольника объекта, $c = (c_1, c_2, \dots, c_n)$ – вектор значений, которые отражают степень уверенности в том, что объект принадлежит определенному классу от 1 до n , o отражает степень уверенности в том, что был обнаружен существующий объект.

Данный метод расширяет исходный вектор до $p = (b, c, o, d)$, где d – вектор, который содержит значения, описывающие расстояние до объекта. В этом решении рассматриваются два различных варианта вектора d . Первый из них не зависит от класса объекта $d = (d)$, а второй учитывает класс $d = (d_1, d_2, \dots, d_n)$. В результате получена архитектура Dist-YOLO [12], позволяющая находить абсолютное расстояние до объекта, используя изображения, снятые с помощью монокулярной камеры.

Dist-YOLO модифицирует функцию потерь l , используемую в YOLOv3, следующим образом [13]:

$$l = \sum_{i=0}^{G^w G^h} \sum_{j=0}^{n^a} q_{i,j} [l_1(i,j) + l_2(i,j) + l_3(i,j) + l_5(i,j)] + l_4(i,j), \quad (5)$$

где $l_1(i,j)$ – потери при определении центра ограничивающего прямоугольника (bounding box);
 $l_2(i,j)$ – потери при определении размеров ограничивающего прямоугольника объекта;
 $l_3(i,j)$ – потери при определении уверенности;
 $l_4(i,j)$ – потери при определении класса;
 $l_5(i,j)$ – потери при определении дистанции;
 $q_{i,j} \in \{0,1\}$ – константа, которая показывает, содержит ли i -я ячейка и j -я якорная рамка (anchor) объект.

Функция применяется к ячейкам сетки $G^w G^h$ и n^a якорным рамкам. Часть l_1, \dots, l_4 заимствована у YOLOv3, а l_5 является новой функцией для нахождения потерь при определении расстояния.

В последующих формулах будет использован знак \wedge , чтобы выделить значения, предсказанные нейросетью.

Потери при определении центра ограничивающего прямоугольника рассчитываются следующим образом:

$$l_1(i,j) = z_{i,j} [H(c_{i,j}^x, \hat{c}_{i,j}^x) + H(c_{i,j}^y, \hat{c}_{i,j}^y)], \quad (6)$$

где $c_{i,j}^x$ и $c_{i,j}^y$ – это координаты центра рамки;
 $H(.,.)$ – бинарная кросс-энтропия (binary cross-entropy);
 $z_{i,j} = 2 - w_{i,j} h_{i,j}$ служит для определения относительного веса (i,j) -го размера ограничивающего прямоугольника в соответствии с его шириной $w_{i,j}$ и высотой $h_{i,j}$.

Формула расчета потерь при определении размеров ограничивающего прямоугольника имеет вид:

$$l_2(i,j) = 0.5 z_{i,j} \left[\left(\log \left(\frac{w_{i,j}}{a_j^w} \right) - \hat{w}_{i,j} \right)^2 + \left(\log \left(\frac{h_{i,j}}{a_j^h} \right) - \hat{h}_{i,j} \right)^2 \right], \quad (7)$$

где a_j^w и a_j^h – ширина и высота j -й якорной рамки (anchor) соответственно.

Потери при определении уверенности рассчитываются по формуле

$$l_3(i,j) = q_{i,j} H(q_{i,j}, \hat{q}_{i,j}) + (1 - q_{i,j}) H(q_{i,j}, \hat{q}_{i,j}) I_{i,j}, \quad (8)$$

где $\hat{q}_{i,j}$ – предсказанная уверенность;

$I_{i,j}$ – маска, которая исключает часть потери для i -й ячейки, если $q_{i,j} = 0$, а метрика IoU (Intersection over Union) > 0.5 .

Расчет потерь при определении класса происходит по формуле

$$l_4(i,j) = \sum_{k=0}^c H(C_{i,j,k} - \hat{C}_{i,j,k}), \quad (9)$$

где $C_{i,j,k}$ – вероятность k -го класса в i -й ячейке.

Функция потерь дистанции без учета класса объекта имеет вид

$$l_5(i,j) = \omega (\hat{d}_{i,j} - d_{i,j})^2. \quad (10)$$

Функция потерь дистанции с учетом класса определена как

$$l_5(i, j) = \omega \sum_{k=0}^c C_{i,j,k} \left(\hat{d}_{i,j,k} - d_{i,j,k} \right)^2, \quad (11)$$

где ω – весовая константа.

Обучение выполнено на наборе данных KITTI 3D Object Detection Evaluation 2017³, который содержит 7481 тренировочных и 7518 тестовых изображений. Однако следует отметить, что для обучения и тестирования данной модели использовался только тренировочный датасет, который был разделен на тренировочную часть, состоящую из 5241 изображения, и тестовую, содержащую 2240 изображений. Разрешение изображений – 1216 × 366 пикселей. Классы, которые были выбраны, – пешеход, легковой автомобиль, фургон, грузовик, сидящий человек, велосипед и трамвай. Максимальное и минимальное расстояния ограничили значениями 150 и 0 м соответственно, но при обучении использовались нормализованные значения дистанции [0, 1].

Для тестирования производительности модели были выбраны следующие настройки: разрешение изображений было уменьшено до 608 × 192 пикселей, размер батча – 24, количество эпох обучения – 100, оптимизатор – Adam.

Для оценки полученных результатов были определены средняя абсолютная ε_A (в метрах) и средняя относительная ε_R (в процентах) погрешности:

$$\varepsilon_A = \frac{1}{n} \sum_{i=1}^n |d_i - \hat{d}_i|; \quad (12)$$

$$\varepsilon_R = \frac{1}{n} \sum_{i=1}^n \frac{|d_i - \hat{d}_i|}{\max(d_i, 1)}, \quad (13)$$

где n – количество найденных объектов;

d_i и \hat{d}_i – известная и предсказанная дистанции соответственно.

Оценка результатов представлена в таблице 3. Min, Mean и Max – минимальное, среднее и максимальное отклонения в метрах соответственно. При этом Dist-YOLOv3 G – результаты для метода определения расстояния вне зависимости от класса объекта, а Dist-YOLOv3 W – с учетом класса объекта.

Таблица 3. – Погрешности при определении расстояния до объекта на основе модификации архитектуры нейронной сети YOLOv3

Объект	Dist-YOLOv3 G						Dist-YOLOv3 W					
	Кол-во	Min	Mean	Max	ε_A	ε_R	Кол-во	Min	Mean	Max	ε_A	ε_R
Пешеход	1150	-16,81	-0,54	7,73	1,75	0,11	1117	-8,93	3,83	39,48	4,87	0,41
Автомобиль	7682	-24,87	-0,81	14,47	2,49	0,11	7798	-13,24	8,75	42,10	10,68	0,44
Фургон	749	-20,49	0,46	18,00	3,55	0,15	751	-26,88	13,94	42,25	14,17	0,64
Грузовик	318	-34,36	1,16	15,93	4,63	0,14	319	-27,33	4,51	26,30	6,49	0,21
Сидящий человек	57	-7,09	-0,59	4,74	1,84	0,22	56	-2,27	4,35	16,15	4,64	0,64
Велосипедист	406	-17,79	-1,25	14,66	2,47	0,11	409	-4,16	14,32	46,80	14,38	0,85
Трамвай	135	-26,66	1,85	19,57	4,72	0,19	132	-22,37	-2,65	11,36	4,41	0,17
Все	10 497	-34,36	-0,61	19,57	2,57	0,11	10 582	-27,33	8,52	46,80	10,22	0,46

Сравнив погрешности этих двух методов, можно сделать вывод, что точность вычисления расстояния с учетом класса объекта Dist-YOLOv3 W существенно уступает методу, который не учитывает класс объекта. Таким образом, тест производительности данной модели показал, что средняя погрешность в определении абсолютного расстояния составляет около 2,5 м, а средняя погрешность относительного расстояния – 11 %. Это показывает, что, интегрировав алгоритм определения расстояния в архитектуру YOLO, можно добиться вполне приемлемых результатов. Реализация данного решения доступна на странице GitLab⁴.

³ KITTI 3D Object Detection Benchmark [Электронный ресурс]. – URL: http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d (дата обращения 30.08.2024).

⁴ URL: <https://gitlab.com/EnginCZ/yolo-with-distance> (дата обращения 05.09.2024).

Заключение. Метод определения расстояния до объекта, основанный на его координатах и карте глубины, изначально разрабатывался для применения в области автономного вождения, однако его структура также позволяет использовать его в других сферах. Основным недостатком данного метода является то, что для корректного определения расстояния до объекта в угол обзора видеокамеры должна включаться поверхность, на которой располагается объект, а также необходимо знать высоту, на которой расположена видеокамера. В работе [6] эта проблема решается тем, что видеокамера размещена на штативе, высота которого фиксирована и поэтому может приниматься за константу, что дает возможность точно вычислить высоту расположения видеокамеры.

Метод определения расстояния до объекта на основе оценки его размеров на изображении также планировался для применения в сфере автономного вождения. Камера должна была монтироваться на поезд и определять объекты на больших расстояниях. Однако структура данного решения позволяет переобучить нейронную сеть для использования в других областях. Основным недостатком данного решения является то, что необходимо знать средние размеры для классов определяемых объектов. Также в статье [12] были проведены сторонние исследования, целью которых было найти зависимость между размерами объекта и расстоянием до него, а также определить, возможно ли предсказать дистанцию, используя эти данные. В итоге был сделан вывод, что определение точной дистанции до объекта только на основе его размеров практически невозможно.

Можно отметить, что метод, предполагающий модификацию архитектуры нейронной сети YOLO3, является самым универсальным из рассмотренных. Изначально он также был разработан для использования в сфере автономного вождения, однако его структура позволяет его переобучить для работы и в других областях при наличии требуемого обучающего набора данных. Недостаток же данного метода состоит в том, что при небольших расстояниях (примерно 1 м) относительная погрешность составляет около 45 %.

ЛИТЕРАТУРА

1. Мальцев С. В., Абламейко С. В., Богуш Р. П. Обработка сигналов и изображений средствами векторно-матричных вычислений. – Новополоцк: ПГУ, 2011. – 212 с.
2. Estimation of Absolute Scale in Monocular SLAM Using Synthetic Data / D. Rukhovich, D. Mouritzen, R. Kaestner et al. // IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). – 2019. – P. 803–812. – DOI: [10.1109/ICCVW.2019.00108](https://doi.org/10.1109/ICCVW.2019.00108).
3. DisNet: A novel method for distance estimation from monocular camera [Electronic resource] / M. A. Haseeb, J. Guan, D. Ristić-Durrant et al. // 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems / 10th Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV). – 2018. – URL: <https://project.inria.fr/ppniv18/files/2018/10/paper22.pdf> (дата обращения 02.09.2024).
4. Богуш Р. П., Захарова И. Ю. Алгоритм сопровождения людей на видеопоследовательностях с использованием сверточных нейронных сетей для видеонаблюдения внутри помещений // Компьютерная оптика. – 2020. – Т. 44, № 1 – С. 109–116. – DOI: [10.18287/2412-6179-CO-565](https://doi.org/10.18287/2412-6179-CO-565).
5. Повторная идентификация людей в системах видеонаблюдения с использованием глубокого обучения: анализ существующих методов / Х. Чен, С. А. Игнатьева, Р. П. Богуш и др. // Автоматика и телемеханика. – 2023. – № 5. – С. 61–112. – DOI: [10.31857/S0005231023050057](https://doi.org/10.31857/S0005231023050057).
6. Absolute distance prediction based on deep learning object detection and monocular depth estimation models / A. Masoumian, D. G. F. Marei, S. Abdulwahab et al. // Frontiers in Artificial Intelligence and Applications. – 2021. – Vol. 339. – P. 325–334. – DOI: [10.3233/FAIA210151](https://doi.org/10.3233/FAIA210151).
7. Deep Residual Learning for Image Recognition / K. He, X. Zhang, S. Ren et al. // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – P. 770–778. – DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
8. Taha Z., Jizat J. A. M. A comparison of two approaches for collision avoidance of an automated guided vehicle using monocular vision // Applied Mechanics and Materials. – 2012. – Vol. 145. – P. 547–551. – DOI: [10.4028/www.scientific.net/AMM.145.547](https://doi.org/10.4028/www.scientific.net/AMM.145.547).
9. Geiger A., Lenz P., Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite // IEEE Conference on Computer Vision and Pattern Recognition. – 2012. – P. 3354–3361. – DOI: [10.1109/CVPR.2012.6248074](https://doi.org/10.1109/CVPR.2012.6248074).
10. Microsoft COCO: Common objects in context / T. Y. Lin, M. Maire, S. Belongie et al. // Lecture Notes in Computer Science. 2014. – Vol. 8693. – P. 740–755. – DOI: [10.1007/978-3-319-10602-1_48](https://doi.org/10.1007/978-3-319-10602-1_48).
11. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement // ArXiv. – 2018. – DOI: [10.48550/arXiv.1804.02767](https://doi.org/10.48550/arXiv.1804.02767).
12. Vajgl M., Hurtik P., Nejezchleba T. Dist-YOLO: Fast Object Detection with Distance Estimation // Applied Sciences. – 2022. – Vol. 12, iss. 3. – P. 1354. – DOI: [10.3390/app12031354](https://doi.org/10.3390/app12031354).
13. Poly-YOLO: Higher speed, more precise detection and instance segmentation for YOLOv3 / P. Hurtik, V. Molek, J. Hula et al. // Neural Computing and Applications. – 2022. – Vol. 34. – P. 8275–8290. – DOI: [10.1007/s00521-021-05978-9](https://doi.org/10.1007/s00521-021-05978-9).

REFERENCES

1. Mal'cev, S. V., Ablamejko, S. V., & Bogush, R. P. (2011). *Obrabotka signalov i izobrazhenij sredstvami vektorno-matrichnyh vychislenij [Processing of signals and images by means of vector-matrix calculations]*. Novopolotsk: PSU. (In Russ.).

2. Rukhovich, D., Mouritzen, D., Kaestner, R., Rufli, M., & Velizhev A. (2019). Estimation of Absolute Scale in Monocular SLAM Using Synthetic Data. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (803–812). IEEE. DOI: [10.1109/ICCVW.2019.00108](https://doi.org/10.1109/ICCVW.2019.00108).
3. Haseeb, M. A., Guan, J., Ristic-Durrant, D., & Gräser, A. (2018). DisNet: A novel method for distance estimation from monocular camera. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems: 10th Workshop on Planning, Perception and Navigation for Intelligent Vehicles (PPNIV)*. URL: <https://project.inria.fr/ppniv18/files/2018/10/paper22.pdf>.
4. Bogush, R. P., & Zaharova, I. Ju. (2020). Algoritm soprovozhdenija ljudej na videoposledovatel'nostjakh s ispol'zovaniem svertochnyh nejronnyh setej dlja videonabljudenija vnutri pomeshhenij [Person tracking algorithm based on convolutional neural network for indoor video surveillance]. *Komp'yuternaja optika [Computer Optics]*, 40(1), 109–116. DOI: [10.18287/2412-6179-CO-565](https://doi.org/10.18287/2412-6179-CO-565). (In Russ., abstr. in Engl.).
5. Chen, H., Ichnatsyeva, S. A., Bohush, R. P., & Ablameyko, S. V. (2023). Person Re-identification in Video Surveillance Systems Using Deep Learning: Analysis of the Existing Methods. *Automation and Remote Control*, 84(5), 497–528. DOI: [10.1134/S0005117923050041](https://doi.org/10.1134/S0005117923050041).
6. Masoumian, A., Marei, D. G. F., Abdulwahab, S., Cristiano J., Puig D., & Rashwan H. A. (2021). Absolute distance prediction based on deep learning object detection and monocular depth estimation models. *Frontiers in Artificial Intelligence and Applications*, 339, 325–334. DOI: [10.3233/FAIA210151](https://doi.org/10.3233/FAIA210151).
7. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (770–778). IEEE. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
8. Taha, Z., & Jizat, J. A. M. (2012). A comparison of two approaches for collision avoidance of an automated guided vehicle using monocular vision. *Applied Mechanics and Materials*, 145, 547–551. DOI: [10.4028/www.scientific.net/AMM.145.547](https://doi.org/10.4028/www.scientific.net/AMM.145.547).
9. Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition* (3354–3361). DOI: [10.1109/CVPR.2012.6248074](https://doi.org/10.1109/CVPR.2012.6248074).
10. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.) *Computer Vision – ECCV 2014*. ECCV 2014. *Lecture Notes in Computer Science: Vol. 8693* (740–755). Springer, Cham. DOI: [10.1007/978-3-319-10602-1_48](https://doi.org/10.1007/978-3-319-10602-1_48).
11. Redmon, J., Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *ArXiv*. DOI: [10.48550/arXiv.1804.02767](https://doi.org/10.48550/arXiv.1804.02767).
12. Vajgl, M., Hurtik, P., Nejezchleba, T. (2022). Dist-YOLO: Fast Object Detection with Distance Estimation. *Applied Sciences*, 12(3), 1354. DOI: [10.3390/app12031354](https://doi.org/10.3390/app12031354).
13. Hurtik, P., Molek, V., Hula, J., Vajgl, M., Vlasanek, P., & Nejezchleba, T. (2022). Poly-YOLO: Higher speed, more precise detection and instance segmentation for YOLOv3. *Neural Computing and Applications*, 34, 8275–8290. DOI: [10.1007/s00521-021-05978-9](https://doi.org/10.1007/s00521-021-05978-9).

Поступила 23.09.2024

ANALYSIS OF METHODS FOR DISTANCE ESTIMATION TO AN OBJECT FROM A SINGLE VIDEO CAMERA IMAGE USING NEURAL NETWORKS

N. LUPENKO, R. BOHUSH

(Euphrosyne Polotskaya State University of Polotsk)

H. CHEN

(Zhejiang Shuren University, China)

This paper discusses about any approaches to determining the distance to an object based on an image generated by a monocular video camera, which use artificial neural networks at various stages of processing. Method based on finding a depth map, detecting an object, and then projecting its coordinates onto the depth map is analyzed. It describes a method that uses the relationship between the real size of an object and its size in the image. It considers a method based on a modification of the YOLO, which allows expanding the resulting descriptor with an additional vector characterizing the distance to the object. Data sets used to train neural networks used in algorithms for calculating the absolute distance to an object based on an image is analyzed. The paper discusses about the effectiveness of the methods considered, their advantages and disadvantages, as well as the prospects for using them for practical solutions.

Keywords: *distance estimation, neural network, computer vision, object detection, depth map.*