Д. А. Нестеров[1], *Б. М. Шурыгин*[2], *А. Е. Соловченко*[2], *А. С. Крылов*[1], *Д. В. Сорокин*[1]

НЕЙРОСЕТЕВОЙ МЕТОД ДЕТЕКЦИИ ПЛОДОВ НА ИЗОБРАЖЕНИЯХ ДЕРЕВЬЕВ ЯБЛОНИ*

Введение

Данная работа посвящена разработке метода детекции плодов в промышленных яблоневых садах и оценке его эффективности.

В настоящее время для осуществления большинства агротехнологических операций на фермах и в садах используется ручной труд, который отнимает много времени и увеличивает себестоимость продукции, а также не защищен от влияния человеческого фактора. Однако с развитием точного земледелия и информационных технологий появилась возможность автоматизировать многие сельскохозяйственные процессы. Так, получение и анализ изображений стали играть ключевую роль в создании автоматических систем для точного земледелия [1]. В частности, информацию, полученную из изображений, можно использовать для оценки размера урожая плодов в промышленных садах [2].

Детектирование, подсчет и локализация плодов в промышленных садах — одна из важнейших задач автоматизации плодоводства. Автоматизированные методы решения данных задач позволяют фермерам оценивать размер урожая, принимать бизнес-решения на основе этих оценок, а также экономить время и средства, исключая ручной учет числа плодов. Также детекция плодов является необходимым компонентом технологий автоматизированного сбора урожая, который является одним из наиболее трудоемких и затратных процессов [3].

[1]Лаборатория математических методов обработки изображений, факультет Вычислительной математики и кибернетики, МГУ имени М. В. Ломоносова, e-mail: dim.nesterov2015@gmail.com, dsorokin@cs.msu.ru kryl@cs.msu.ru.

[2]Кафедра биоинженерии, Биологический факультет, МГУ имени М. В. Ломоносова, e-mail: shu_b@mail.ru, solovchenkoae@my.msu.ru.

*Работа выполнена при поддержке проекта «Интеллектуальный анализ изображений для высокопроизводительного фенотипирования растений и точного земледелия» научно-образовательной школы МГУ «Мозг, когнитивные системы, искусственный интеллект».

В работе рассматривается задача *объектной сегментации*: по входному изображению предсказывается маска сегментации и ограничивающий прямоугольник для каждого яблока.

(a) Фрагмент изображения (b) Фрагмент с нанесенной разметкой

Рис. 1. Эталонная разметка для задачи объектной сегментации из набора данных Fuji-SfM [4]

В данной работе разработан и реализован метод детекции яблок с помощью двухэтапной нейросети Mask R-CNN [5]. На вход нейросеть получает изображение, на котором требуется осуществить детекцию объектов, после чего строится сжатое представление этого изображения, затем нейросеть обучаемым образом генерирует и анализирует гипотезы о местонахождении целевых объектов, используя это сжатое представление. Mask R-CNN, как двухэтапный детектор, отличается от одноэтапных (например, YOLO [6]), как правило, большей точностью предсказаний.

Предложенный алгоритм сравнивается с алгоритмом на основе Mask R-CNN из работы Fuji-SfM [7] и тестируется на одноименном наборе данных [4]. Авторы этой работы обучали нейросеть с помощью SGD с темпом обучения 0.001. В данной работе предлагается обучать нейросеть с помощью оптимизатора Adam [8] с меньшим темпом обучения, уменьшая его через некоторое количество эпох. В результате нам удалось увеличить метрики AP (average precision) и AR (average recall) на 4-12% по сравнению с алгоритмом авторов Fuji-SfM.

Нейросетевой метод детекции

В качестве основного подхода для решения задачи объектной сегментации использована архитектура нейронной сети Mask R-CNN [5].

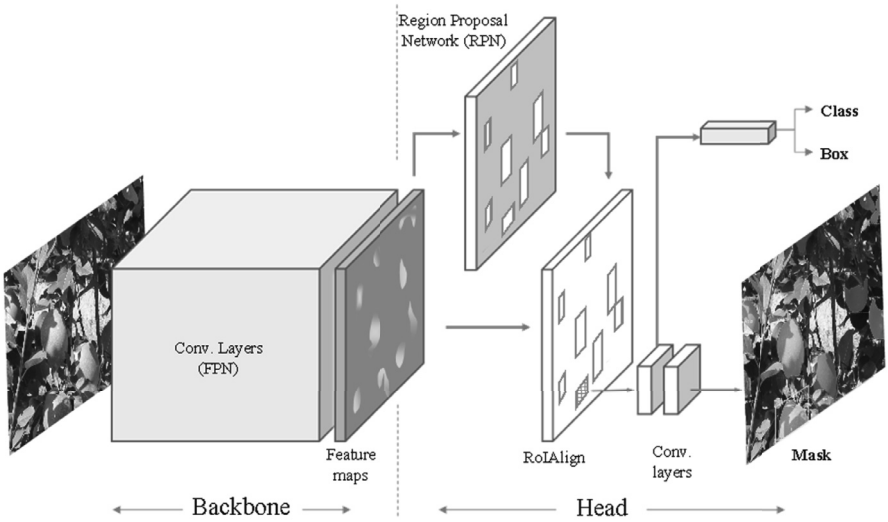

Рис. 2. Архитектура нейронной сети Mask R-CNN [5]

На вход нейросети подается изображение $I \in \mathbb{R}^{C \times H \times W}$, из которого извлекаются карты признаков $\mathscr{D}$ с помощью *кодировщика* (backbone):

$$f(I, P_{backbone}) = \{D_i^{C_i \times H_i \times W_i}\}_{i=1}^{k} \equiv \mathscr{D},$$

где f — отображение, задаваемое кодировщиком, $P_{backbone}$ — все обучаемые параметры кодировщика. Таким образом, $\mathscr{D}$ — множество, состоящее из k выходных тензоров, каждый из которых представляет собой сжатое или *закодированное* представление входного изображения I.

Полученные карты признаков используются нейросетью, генерирующей гипотезы $\mathscr{H}$ о местонахождении объектов на изображении (*Region Proposal Network*, RPN), которые называются *регионами интереса* (*Region of Interest*, RoI). Гипотезы генерируются в виде ограничивающих прямоугольников (bounding boxes), задаваемых с помощью координат левого верхнего угла и правого нижнего угла в системе координат, ассоциированной со входным изображением I:

$$g(\mathscr{D}, P_{rpn}) = \{(bbox_i, objectness_i)\}_{i=1}^{l} \equiv \mathscr{H},$$
$$bbox_i = [x_i^{min}, y_i^{min}, x_i^{max}, y_i^{max}],\ x_i \in [0, W),\ y_i \in [0, H)$$

где g — отображение, задаваемое RPN, P_{rpn} — все обучаемые параметры RPN, количество генерируемых гипотез l определяется гиперпараметрами. Также на основе карт признаков $\mathscr{D}$ нейросеть RPN для каждой гипотезы $bbox_i$ вычисляет $objectness_i \in [0, 1]$, показывающее

уверенность в нахождении объекта внутри ограничивающего прямоугольника, где $objectness = 0$ — в RoI точно нет объекта, $objectness = 1$ — в RoI точно есть объект.

Затем каждая гипотеза $bbox_i$, в зависимости от своего размера $h \times w$, проецируется на выбранную специальным образом карту признаков $\mathscr{D}_j \in \mathbb{R}^{C_j \times H_j \times W_j}$, $\mathscr{D}_j \in \mathscr{D}$ с применением билинейной интерполяции (RoIAlign):

$$r(bbox_i, \mathscr{D}_j) = proj_i \in \mathbb{R}^{C_j \times h' \times w'},$$

где r — отображение, задаваемое RoIAlign, $proj_i$ — результирующий тензор, содержащий в себе информацию, полученную из входного изображения I. При этом по пространственным размерам $proj_i$ соотносится с $\mathscr{D}_j$ примерно так же, как и гипотеза $bbox_i$ соотносится со входным изображением $I \in \mathbb{R}^{C \times H \times W}$, т.е.

$$\frac{h'}{H_j} \sim \frac{h}{H}, \frac{w'}{W_j} \sim \frac{w}{W}.$$

Таким образом, для каждого региона интереса получены нейросетевые признаки, по которым затем строятся предсказания (Рис. 3):

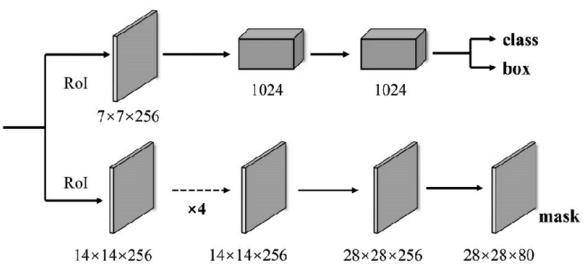

Рис. 3. Получение предсказаний из выделенных признаков

Тензор $proj_i$ проходит по нижней ветке через сверточные слои, тем самым создавая маску сегментации для данного региона интереса. Проходя по верхней ветке с полносвязными слоями, выполняется регрессия ограничивающего прямоугольника и классификация объекта.

После получения предсказаний для каждого тензора $proj_i$, маски сегментации расширяются нулями до размеров входного изображения I, после чего конкатенируются по каналам; предсказанные координаты ограничивающих прямоугольников переводятся в систему координат входного изображения.

Извлечение признаков

В качестве кодировщика использовалась нейросеть ResNet-50 [9] FPN (Feature Pyramid Network), предобученная на наборе данных для детекции объектов COCO [10], поскольку пирамиды признаков являются основным компонентом систем детектирования объектов на разных масштабах [11].

На Рис. 4 восходящий путь — это прямой проход входного изображения I по базовой для пирамиды сверточной нейросети, в данном случае это ResNet-50. На каждом уровне происходит понижение пространственного разрешения в 2 раза, при этом полученные с каждого уровня карты признаков будут использованы на нисходящем пути.

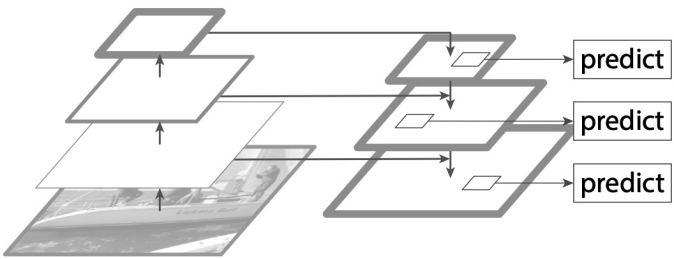

Рис. 4. Концепция Feature Pyramid Network

Рассмотрим тензор $B_k \in \mathbb{R}^{C_k \times H_k \times W_k}$, который является выходным на последнем верхнем k-ом уровне базовой для пирамиды сети. На нисходящем пути пространственное разрешение тензора B_k повышается в 2 раза путем дублирования соседнего пикселя:

$$B'_k = \text{Upsample}(B_k, 2), B'_k \in \mathbb{R}^{C_k \times H_{k-1} \times W_{k-1}},$$

после чего пространственное разрешение тензора B'_k соответствует пространственному разрешению тензора $B_{k-1} \in \mathbb{R}^{C_{k-1} \times H_{k-1} \times W_{k-1}}$. Затем к карте признаков B_{k-1} применяется операция свертки с размером ядра 1×1, впервые представленная в [12], позволяющая изменить число каналов у тензора (Рис. 5):

$$B'_{k-1} = \text{Conv1x1}(B_{k-1}),\ B'_{k-1} \in \mathbb{R}^{C_k \times H_{k-1} \times W_{k-1}}.$$

После всех перечисленных преобразований тензоры B'_{k-1} и B'_k поэлементно складываются, поскольку имеют одинаковые размерности:

$$D_{k-1} = B'_{k-1} + B'_k.$$

Полученный тензор $D_{k-1} \in \mathscr{D}$. Базовая нейросеть ResNet-50 имеет 5 уровней понижения пространственного разрешения, однако в данной работе используются карты признаков только с 4 верхних уровней, поскольку они обладают наибольшей семантической ценностью.

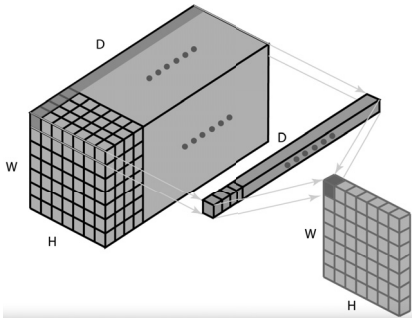

Рис. 5. Изменение количества каналов с помощью свертки 1x1

Генерация RoI-гипотез

Для генерации гипотез требуется задать *якоря* — эталонные ограничивающие прямоугольники, которые будут соотноситься с каждым пикселем на карте признаков. К спроецированному на карту признаков региону интереса применяется полносвязная нейронная сеть, выполняющая регрессию ограничивающих прямоугольников и их классификацию на предмет нахождения внутри них целевых объектов.

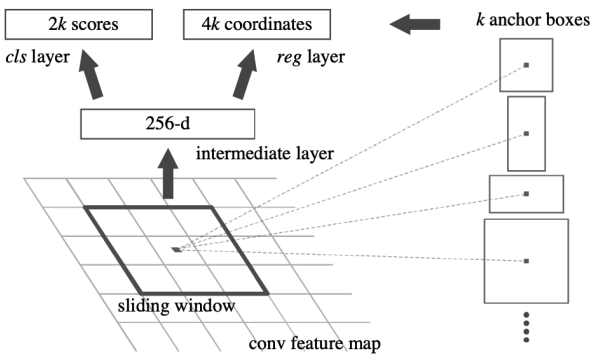

Рис. 6. Анализ предлагаемых с помощью якорей гипотез

Для задания якорей используются гиперпараметры для масштаба и соотношения сторон. В данной работе параметры масштаба (32, 64, 128, 256, 512), то есть генерируются якори с размерами $32 \times 32, 64 \times 64$ и т.д; параметры соотношения сторон (0.5, 1.0, 2.0), то есть генерируются якори с соотношением сторон 1:2, 1:1 и 2:1. В итоге на каждый пиксель карты

признаков приходится $k = 5 \times 3 = 15$ различных якорей, при этом со всей карты признаков $D_i \in \mathbb{R}^{C_i \times H_i \times W_i}$ RPN выдвигает $H_i \times W_i \times k$ гипотез. Гипотезы, выходящие за границы карты признаков, обрезаются, а гипотезы, не попадающие точно в пиксели карты признаков, интерполируются.

Выбор карты признаков в зависимости от размера гипотезы

Каждая гипотеза $bbox_i$ имеет свой пространственный размер $h_i \times w_i$ на исходном изображении $I \in \mathbb{R}^{C \times H \times W}$. Выбор выходного уровня кодировщика, на карту признаков которого будет проецироваться гипотеза $bbox_i$, осуществляется по следующей формуле:

$$k = \left\lfloor k_0 + \log_2 \left(\sqrt{\frac{HW}{h_i w_i}} \right) \right\rfloor,$$

где k_0 — уровень, принимаемый за начальный. Так, если размер гипотезы сопоставим с пространственным разрешением изображения, то гипотеза будет проецироваться на начальный уровень. С другой стороны, если размеры гипотезы в 16 раз меньше размеров изображения, то будет использоваться карта признаков с уровня $k_0 + 2$.

Используемые функции потерь

Зафиксируем RoI, по которому строятся предсказания, во время обучения. Пусть $p = (p_0, p_1, \ldots, p_K)$ — вектор дискретного распределения вероятностей, выдаваемый Softmax - преобразованием с ветки class на Рис. 3, а $t^k = (t_x^k, t_y^k, t_w^k, t_h^k)$ — предсказываемые с ветки box на Рис. 3 ограничивающие прямоугольники для каждого из K классов объектов, в данной работе $K = 1$. Пусть также $v = (v_x, v_y, v_w, v_h)$ — bounding box из эталонной разметки, соответствующий данному RoI (то есть $IoU(t, v) > t_0 = 0.5$), u — эталонный класс для ограничивающего прямоугольника v.

Во время обучения для каждого RoI из эталонной разметки, для которого существует соответствующий ему предсказываемый RoI, определяется многоцелевая функция потерь, совмещающая в себе функцию потерь для классификации, локализации и сегментации:

$$\mathscr{L}_{RoI} = \mathscr{L}_{cls} + \mathscr{L}_{loc} + \mathscr{L}_{mask},$$

где $\mathscr{L}_{cls}$ и $\mathscr{L}_{loc}$ определяются как в [13]:

$$\mathscr{L}_{cls}(p, u) = -\log(p_u),\ \mathscr{L}_{loc}(t^u, v) = [u \geq 1] \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_i^u - v_i),$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2, & |x| < 1, \\ |x| - 0.5, & |x| \geq 1. \end{cases}$$

Ветка mask на Рис. 3 генерирует маску размера $m \times m$ для каждого RoI и для каждого целевого класса. В качестве $\mathscr{L}_{mask}$ используется усредненная по всем пикселям масок бинарная кросс-энтропия, использующая маску только u-го эталонного класса:

$$\mathscr{L}_{mask}(Y, \hat{Y}^u) = -\frac{1}{m^2} \sum_{1 \le i,j \le m} [y_{ij} \log(\hat{y}^u_{ij}) + (1 - y_{ij}) \log(1 - \hat{y}^u_{ij})],$$

где $Y = (y_{ij})_{i,j=\overline{1,m}}, \hat{Y}^u = (\hat{y}^u_{ij})_{i,j=\overline{1,m}}$ — эталонная маска сегментации и предсказанная маска для u-го класса соответственно.

Набор данных Fuji-SfM

Для создания набора данных Fuji-SfM [4] использовалось 11 яблонь Фуджи, его часть для объектной сегментации (Mask-set) была сформирована из 12 снимков с разрешением 5184×3456 px. Поскольку нейронные сети плохо детектируют мелкие объекты [14], и обучение на изображениях такого большого разрешения является вычислительно трудновыполнимым для нейросетей, каждый снимок был поделен на 24 фрагмента размером 1024×1024 px, при этом по горизонтали размер перекрытия соседних фрагментов составляет 192 px, а по вертикали — 213 px.

a) Границы фрагментов. b) Примеры фрагментов. c) Эталонные маски.

Рис. 7. Пример изображения из Mask-set [4]

Полученные 288 фрагментов были вручную размечены — попиксельные маски сохранены в виде списков ограничивающих многоугольников (ОП), по 1 ОП на каждое яблоко.

Оценка качества работы метода

Ниже на Рис. 8 представлены численные результаты работы метода на валидационной части набора данных Fuji-SfM [4]. Для оценки результатов использовалась метрика average precision, AP:

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots, 1\}} precision(r),$$

где r — полнота, $precision(r)$ — точность при заданной полноте r на $precision - recall$ кривой. Полнота определяется как доля найденных детектором целевых объектов среди всех объектов в эталонной разметке изображения. Точность определяется как отношение количества верных срабатываний детектора к числу всех детекций на изображении. При этом «детектирование» происходит тогда, когда предлагаемый алгоритмом ОП или маска сегментации ($pred$) имеет совпадение с ОП или маской сегментации из эталонной разметки (gt) по метрике intersection over union (IoU) большее заданного порога:

$$IoU(pred, gt) = \frac{area(pred \cap gt)}{area(pred \cup gt)} > t.$$

При использовании порога t для метрики AP используется нотация AP@t, также рассматривается метрика AP@[0.5:0.95]:

$$AP@[0.5 : 0.95] = \frac{1}{10} \sum_{t \in \{0.5, 0.55, 0.6, \ldots, 0.95\}} AP@t,$$

то есть AP[0.5:0.95] — средняя точность по всем порогам от 0.5 до 0.95 с шагом 0.05. Также рассматривались значения метрик для объектов разных размеров: small, medium и large:

$$object\_size = \begin{cases} small, \text{если } area < 32^2, \\ medium, \text{если } 32^2 < area < 96^2, \\ large, \text{если } area > 96^2, \end{cases}$$

где area — площадь ограничивающего прямоугольника объекта в квадратных пикселях. Аналогично метрике AP рассматривалась метрика average recall (AR).

На всех графиках по горизонтали отложен номер эпохи обучения, по вертикали — значение метрики. Синим цветом отмечены графики воспроизведения обучения нейросети авторов статьи Fuji-SfM [7], голубым — графики метрик предлагаемого метода.Нейросеть обучалась с помощью Adam на протяжении 25 эпох: первые 15 эпох с темпом обучения $1e-4$, затем он уменьшался в 10 раз через каждые 5 эпох.

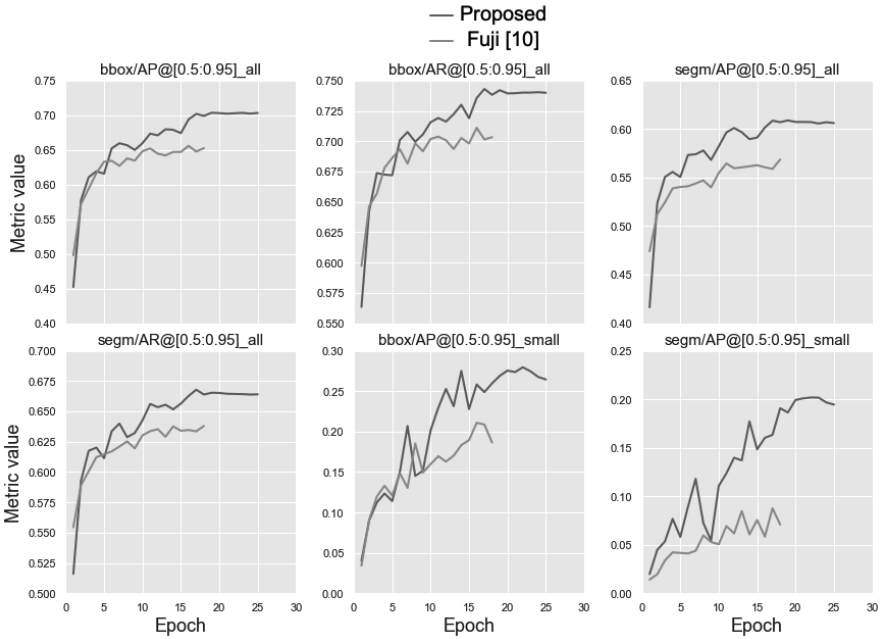

Рис. 8. Метрики на валидационной части набора данных Fuji-SFM

Удалось увеличить точность предсказания масок сегментации для маленьких яблок примерно на 12%, точность AP@[0.5:0.95] и полноту AR@[0.5:0.95] для ограничивающих прямоугольников на 5% и на 4% соответственно, остальные метрики на 4-10%.

Ниже на Рис. 9 приведено несколько примеров работы алгоритма на фрагментах из валидационной части набора данных Fuji-SfM [4], при этом визуализированы все предсказания, уверенность классификатора в которых больше 0.5. На примере а) видно, что яблоко в центре, присутствующее на эталонной разметке не было найдено алгоритмом из [7], в отличие от предложенного метода. На примерах б) и г) предложенный алгоритм смог лучше обработать множественные отклики (сверху и справа для примеров б) и г) соответственно). На примере в) предложенная модель объединила два яблока в одну детекцию, при этом выделив яблоко, не присутствовавшее в эталонной разметке.

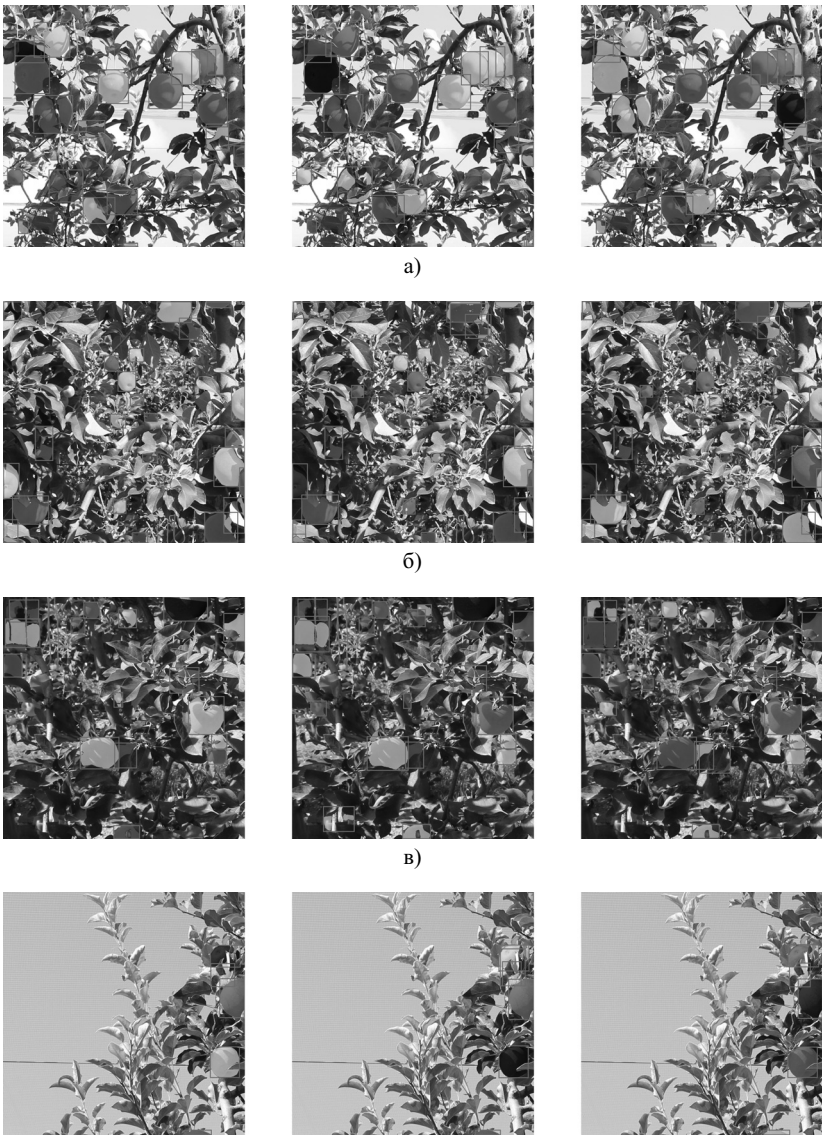

Рис. 9. Примеры работы метода на валидационной части набора данных Fuji-SfM из [7]. Слева — эталонная разметка, посередине — модель, обученная как у авторов Fuji-SfM [7], справа — предложенный метод

Ниже на Рис. 10 представлены численные результаты работы метода (без дообучения) для изображений плодоносящих яблонь сорта Гала, полученных в садах ООО «Сады Карачаево – Черкесии» (Карачаево – Черкесская республика). Для оценки результатов использовались метрики AP и AR. Удалось добиться метрик AP@[0.5:0.95]=58%, AR@[0.5:0.95]=63.5% для ограничивающих прямоугольников и AP@[0.5:0.95]=61%, AR@[0.5:0.95]=66% для масок сегментации.

Также на Рис. 11 изображено несколько примеров работы метода.

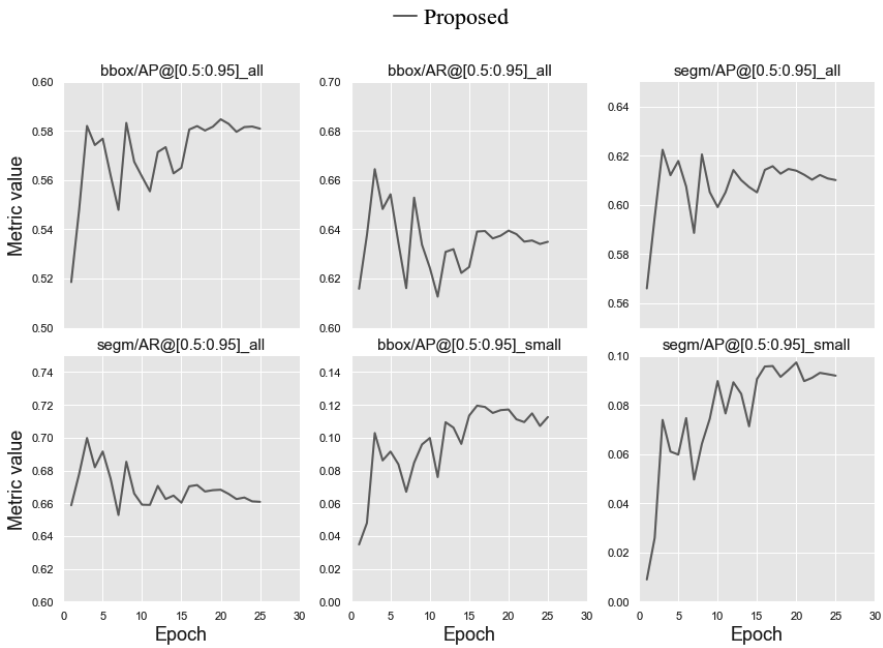

Рис. 10. Метрики, полученные на изображениях яблонь промышленных садов

Рис. 11. Примеры работы метода на изображениях яблонь промышленных садов. Слева — фрагмент, посередине — эталонная разметка, справа — предложенный метод

Литература

1. *Zhao, Yuanshen, Gong, Liang, Huang, Yixiang, Liu, Chengliang* // Computers and Electronics in Agriculture A review of key techniques of vision-based control for harvesting robot. – 2016. – P. 311–323.
2. *Wang, Qi, Nuske, Stephen, Bergerman, Marcel, Singh, Sanjiv* // Experimental robotics Automated crop yield estimation for apple orchards. – 2013. – P. 745–758.
3. *Calvin, Linda, Martin, Philip* // None The US produce industry and labor: Facing the future in a global economy. – 2010.
4. *Gené-Mola, Jordi, Sanz-Cortiella, Ricardo, Rosell-Polo, Joan R, Morros, Josep-Ramon, Ruiz-Hidalgo, Javier, Vilaplana, Verónica, Gregorio, Eduard* // Data in brief Fuji-SfM dataset: a collection of annotated images and point clouds for Fuji apple detection and location using structure-from-motion photogrammetry. – 2020. – P. 105591.
5. *He, Kaiming, Gkioxari, Georgia, Dollár, Piotr, Girshick, Ross* // Proceedings of the IEEE international conference on computer vision Mask R-CNN. – 2017. – P. 2961–2969.
6. *Redmon, Joseph, Farhadi, Ali* // arXiv preprint arXiv:1804.02767 Yolov3: An incremental improvement. – 2018.
7. *Gené-Mola, Jordi, Sanz-Cortiella, Ricardo, Rosell-Polo, Joan R, Morros, Josep-Ramon, Ruiz-Hidalgo, Javier, Vilaplana, Verónica, Gregorio, Eduard* // Computers and Electronics in Agriculture Fruit detection and 3D location using instance segmentation neural networks and structure-from-motion photogrammetry. – 2020. – P. 105165.
8. *Kingma, Diederik P, Ba, Jimmy* // arXiv preprint arXiv:1412.6980 Adam: A method for stochastic optimization. – 2014.
9. *He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian* // Proceedings of the IEEE conference on computer vision and pattern recognition Deep residual learning for image recognition. – 2016. – P. 770–778.
10. *Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, Zitnick, C Lawrence* // European conference on computer vision Microsoft COCO: Common objects in context. – 2014. – P. 740–755.
11. *Lin, Tsung-Yi, Dollár, Piotr, Girshick, Ross, He, Kaiming, Hariharan, Bharath, Belongie, Serge* // Proceedings of the IEEE conference on computer vision and pattern recognition Feature pyramid networks for object detection. – 2017. – P. 2117–2125.

12. *Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, Rabinovich, Andrew* // Proceedings of the IEEE conference on computer vision and pattern recognition Going deeper with convolutions. – 2015. – P. 1–9.

13. *Girshick, Ross* // Proceedings of the IEEE international conference on computer vision Fast R-CNN. – 2015. – P. 1440–1448.

14. *Gené-Mola, Jordi, Vilaplana, Verónica, Rosell-Polo, Joan R, Morros, Josep-Ramon, Ruiz-Hidalgo, Javier, Gregorio, Eduard* // Computers and Electronics in Agriculture Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities. – 2019. – P. 689–698.