

ВОПРОСЫ УПОРЯДОЧИВАНИЯ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ С ИСПОЛЬЗОВАНИЕМ НЕЙРОСЕТЕВЫХ И ЭВРИСТИЧЕСКИХ АЛГОРИТМОВ

Н. О. Бесшапошников¹, А. Г. Леонов^{1,2,3}, М. А. Матюшин^{1,3}

¹ *Федеральный научный центр Научно-исследовательский институт
системных исследований Российской академии наук*

² *Московский педагогический государственный университет*

³ *Московский государственный университет имени М. В. Ломоносова
nikita.beshaposhnikov@gmail.com, dr.l@math.msu.ru, itsaprank@yandex.ru*

В последнее время в космической отрасли и автомобилестроении широко применяются алгоритмы машинного зрения. В задачах компьютерного зрения может потребоваться определение порядка расположения предметов в пространстве. На первый взгляд, данная задача кажется тривиальной, но условия фотографирования сильно влияют на качество результата. Существует целое множество решений для нахождения порядка расположения объектов на изображениях. При этом большинство из них оказываются малоприменимыми для работы с рукотворными объектами. Один из частных случаев данной задачи – упорядочивание объектов по плоской таблице. Основными проблемами при ее решении являются фотографии с перспективными искажениями и небольшими смещениями объектов относительно друг друга. В данной статье изложены результаты проведенных исследований этого направления, рассмотрены эвристический и нейросетевой подходы к решению. Нейросетевой метод показал наиболее удовлетворительные результаты, а архитектура спроектированной нейронной сети, описанной в статье, названа авторами нейротабулятором.

Ключевые слова: нейросети, глубокое обучение, наборы данных, нейротабулятор, распознавание образов.

PROBLEMS OF ORDERING OBJECTS ON IMAGE BY USING NEURAL NETWORKS AND HEURISTIC ALGORITHMS

N. O. Beshaposhnikov¹, A. G. Leonov^{1,2,3}, M. A. Matyushin^{1,3}

¹ *System Research Institute, Russian Academy of Sciences*

² *Moscow Pedagogical State University*

³ *Lomonosov Moscow State University*

nikita.beshaposhnikov@gmail.com, dr.l@math.msu.ru, itsaprank@yandex.ru

Currently, in various branches, such as space and automotive industry, computer vision algorithms are widely applied. In different tasks of computer vision, it may be necessary to determine the order of the objects in space. At first glance, this task may seem insignificant, but the quality of the images strongly depends on shooting conditions. There is a set of solutions for finding the order of objects on images. At the same time, most of the solutions are not very useful for working with human-made objects. One of the special cases of this task is the ordering objects according to a flat table. The main problems in solving this task are photographs with perspective distortions and small displacements of objects relative to each other. The article presents the results of the research conducted in this area. Heuristic and neural network approaches to the solution are considered. The neural network method showed the most sufficient results. The authors named the design of described neural network as neurotabulator.

Keywords: neural networks, deep learning, data sets, neurotabulator, pattern recognition.

В настоящее время особый интерес представляют результаты исследования возможности распознавания рукотворных объектов с динамической интеграцией в виртуальном представлении в среду дополненной реальности, что позволит расширить применение дополненной реальности в промышленности, науке и образовании. При этом важным является не только вопрос успешного детектирования рукотворных объектов, но и их взаимное расположение в пространстве. Данная работа описывает результаты исследования для плоского случая, результаты которого будут обобщены в многомерный случай в дальнейших работах.

Представим, что на фотографии изображены объекты, которые выложены в некоторой упорядоченной двумерной структуре (таблице). Задача состоит в восстановлении табличной структуры расположения объектов по координатам их границ. Этот процесс назван здесь табулированием объектов.

Данная проблема возникает во многих областях компьютерного зрения и в зависимости от ограничений может решаться по-разному. Например, для детекции калибровочных сеток, в которых размеры всех значков и расстояния между ними одинаковы, а также заранее известны размеры искомой таблицы, может с успехом применяться стэнфордский алгоритм детекции [1]. Однако ограничения его применения заключаются в жестких требованиях к структуре детектируемой таблицы. Иными словами, такой подход может использоваться только для одной заранее заданной таблицы. Алгоритм будет требовать модификации для каждой новой таблицы.

Также данная проблема известна в области анализа документов с целью детектирования и построения таблиц в них [2]. В данной области успешно применяются как эвристические, так и нейросетевые подходы [3]. Однако применимость существующих в этой области решений ограничивается фронтальными кадрами (сканами) документов, тогда как для перспективно искаженных изображений данные подходы не применимы.

Эвристические подходы. В наиболее простом случае границы объектов на изображении очерчены прямоугольниками со сторонами, параллельными сторонам изображения, так называемыми bounding box'ами. Существуют различные эвристические способы восстановления табличной структуры по координатам границ.

Эвристический апостериорный подход. Один из простейших способов эвристического табулирования заключается в аппроксимации размеров ячеек таблицы по размерам bounding box'ов объектов. За ширину и высоту ячейки таблицы принимается, соответственно, средняя ширина и средняя высота bounding box'ов. Затем по минимальным и максимальным координатам объектов на изображении восстанавливается область, занимаемая таблицей на изображении, и все координаты объектов пересчитываются относительно верхнего левого угла области. Теперь табличные координаты объектов могут быть получены путем деления относительных координат нацело на ширину и высоту ячейки таблицы, соответственно.

К явным недостаткам такого подхода можно отнести очевидную неустойчивость к таблицам, в которых размер объектов значительно (на 20 % и более) меньше размера ячейки таблицы, т. е. качество работы данного алгоритма сильно падает с увеличением размера таблицы при неплотной укладке объектов.

Также данный алгоритм чувствителен к таблицам, составленным из объектов разных размеров, и качество работы сильно ухудшается выбросами. Для борьбы с выбросами можно заменить среднее арифметическое по ширине и высоте bounding box'ов на выборочную медиану этих же параметров.

При перспективных искажениях изображения корректность работы данного алгоритма резко падает.

Из плюсов стоит отметить простоту реализации и скорость работы. При идеальных условиях фотографирования и расположения объектов в таблице данный алгоритм ничуть не уступает рассматриваемым ниже вариациям.

Эвристический априорный подход. Если же к изображению прилагается некоторая априорная информация о таблице, например, о ее ширине и высоте, то можно воспользоваться априорным эвристическим подходом.

В данном алгоритме так же, как и в описанном выше случае, по минимальным и максимальным координатам выделяется область, занимаемая на изображении таблицей, затем она разрезается на нужное количество ячеек по высоте и по ширине в соответствии с априорным знанием о размерах таблицы. Далее для каждого объекта на изображении определяется его принадлежность ячейке. В самом простом случае можно считать, что объект занимает ячейку, если его геометрический центр в нее попадает. Есть и иные подходы к определению, в частности, можно судить о принадлежности объекта ячейке по показателю IoU [4] (площадь пересечения/суммарная площадь) или же по отношению площади пересечения с ячейкой к площади объекта. К усовершенствованиям данного подхода можно отнести возможность перемещения объектов в свободные ячейки при их попадании в одну ячейку, т. е. финальное исправление ошибок алгоритма определения инцидентности объекта ячейке.

Данный алгоритм, хотя и является устойчивым к неплотной укладке объектов, наследует проблемы апостериорного подхода в случае табулирования изображений с перспективными искажениями. Также очевидным минусом является неработоспособность алгоритма при отсутствии априорной информации о таблице.

Эвристические методы были реализованы с помощью библиотеки OpenCV [5], более подробно проведенное исследование было описано в статье [6].

Нейросетевые подходы. В связи с тем, что эвристические подходы обладают существенными недостатками в случае с перспективно искаженными изображениями и часто требуют некоторой априорной информации о структуре таблицы, были проведены исследования возможности реализовать нейросетевой подход к данной проблеме. При этом исходные данные, получаемые нейросетью, описываются как матрица ширины 4, высоты N , где N – количество объектов на изображении и 4 числа в диапазоне от -1 до 1 описывают относительные координаты верхнего левого и нижнего правого углов bounding box'ов каждого объекта. Относительные координаты вычисляются относительно области, занимаемой на изображении таблицей, область определяется так же, как и в эвристическом подходе.

В качестве выхода нейросети ожидается матрица ширины 2, высоты N , где первый столбец трактуется как относительные координаты объектов в таблице по горизонтали, второй, соответственно, по вертикали.

Относительные координаты считаются как $\frac{(x - x_0)}{w} - \frac{1}{2}$, где x – координата угла bounding box'a, x_0 – соответствующая координата верхнего левого угла области таблицы, w – ширина (высота в случае y координаты) области таблицы, в случае координат объектов на изображении.

Относительные координаты считаются как $\frac{i}{w} - \frac{1}{2}$, где i – координата ячейки в таблице, w – ширина (высота в случае j координаты) таблицы в случае координат ячеек в таблице.

Для реализации нейросетевого подхода к данной проблеме существуют несколько архитектур нейронных сетей.

Обучение сетей производилось на синтетически сгенерированной выборке, содержащей таблицы всех размеров вплоть до 8×8 , со случайным процентом заполнения, большим 0,8, случайными перспективными искажениями, каждый семпл перед обучением случайным образом перемешивался. Количество семплов в обучающей выборке составило 70000, в валидационной – 30 000. Конструирование сетей производилось во фреймворке pytorch, в качестве метрики точности на валидационной выборке было использовано $\exp(-\text{avg_mse})$, где avg_mse – средняя квадратическая ошибка по всем выходным значениям в сравнении с эталонными, функцией потерь была выбрана mse.

Полносвязная нейронная сеть. Данная архитектура является классической [7], состоит из одного или нескольких скрытых слоев нейронов, где каждый следующий слой нейронов реализован как умножение выхода предыдущего слоя на матрицу весов фиксированного размера и применение нелинейной функции активации.

Архитектура реализованной сети (рис. 1) представляет собой 4 скрытых линейных слоя nn.Linear, разделенных нелинейными слоями nn.PReLU. Конфигурации линейных слоев следующие:

- 1-й слой: 4×16 ;
- 2-й слой: 16×24 ;
- 3-й слой: 24×12 ;
- 4-й слой: 12×2 .

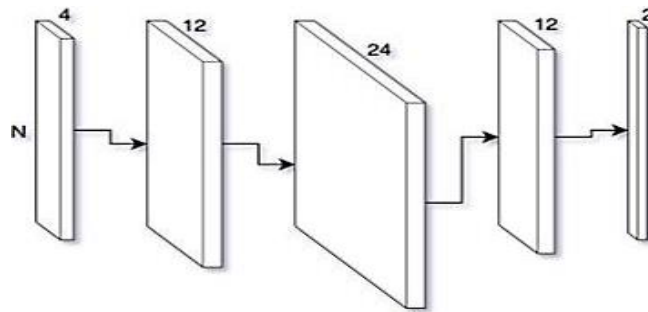


Рис. 1. Архитектура полносвязной нейронной сети

Данная архитектура зарекомендовала себя быстрым обучением, потери после 2 эпох составили 0,002, точность равна 0,996. На реальных данных точность составила 0,993, однако основные проблемы работы были связаны с перспективно искаженными изображениями, что заставило искать альтернативные архитектуры.

Полносверточная нейронная сеть [8] способна находить более глубокие закономерности между входными данными. Однако скорость ее обучения существенно ниже, чем у полносвязного аналога. В ходе данного эксперимента была построена сверточная нейронная сеть с двумя циклами эмбединга. Архитектура выглядит следующим образом (рис. 2):

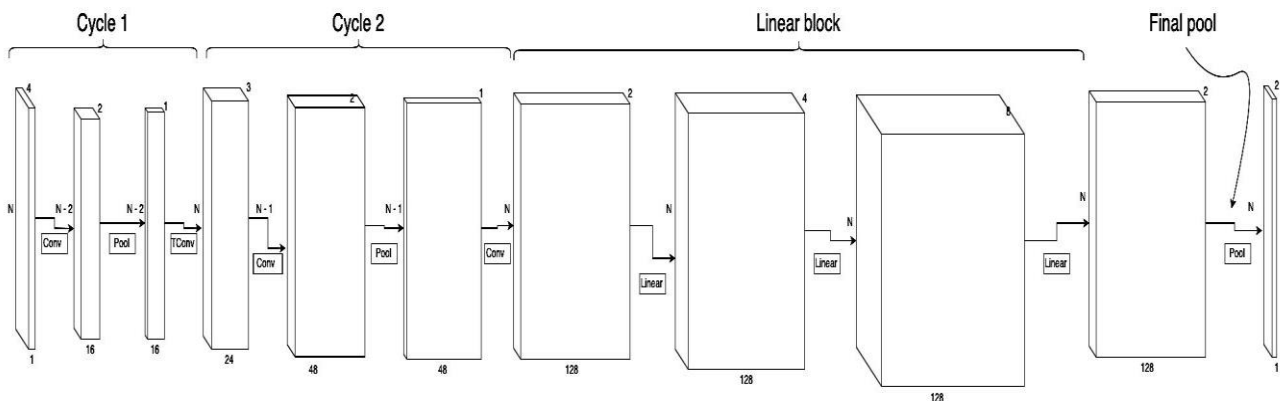


Рис. 2. Архитектура полносверточной нейронной сети: 1-й цикл эмбединга: свертка с ядром 3×3 , 16 фильтров; нелинейность nn.PReLU; пулинг с ядром 1×2 nn.MaxPool2d(1, (1, 2)); транспонированная свертка с ядром 3×3 , 24 фильтра; нелинейность nn.PReLU; 2-й цикл эмбединга: свертка с ядром 2×2 , 48 фильтров; нелинейность nn.PReLU; пулинг с ядром 1×2 nn.MaxPool2d(1, (1, 2)); транспонированная свертка с ядром 2×2 , 128 фильтров; нелинейность nn.PReLU; трехслойный перцептрон: линейный слой 2×4 ; нелинейность nn.PReLU; линейный слой 4×8 ; нелинейность nn.PReLU; линейный слой 8×2 ; нелинейность nn.PReLU; финальный пулинг с ядром $128 \times 1 \times 1$ nn.MaxPool3d(1, (128, 1, 1))

Данная архитектура обучалась значительно дольше полносвязной сети, результаты обучения после 14 эпох:

- потери составили 0,0002, точность равна 1,000, результаты проверки на реальных данных показали точность, равную 0,996, больше половины случаев некорректной работы

предыдущей архитектуры на перспективно искаженных изображениях не имели места в данном эксперименте. В целом полносверточная сеть показывает лучшие результаты в данной задаче в связи с тем, что последовательные свертки позволяют сети лучше выделять общие признаки среди соседствующих элементов входных данных, а в данной задаче входные признаки имели одинаковую природу, в связи с чем имело смысл обрабатывать их сверточно по несколько раз. Данный подход полностью себя оправдывает, показывая наилучшие результаты на данных, содержащих перспективные искажения.

Планируется добавить в синтетическую обучающую выборку случайные «бласти» – эффект, немного разбрасывающий объекты в стороны от эпицентра «бласти», усилить перспективные искажения в генерации синтетики, добавить изгибающиеся таблицы, увеличить количество тренируемых размеров до 12×12 .

Описание экспериментов. В ходе работы проводились эксперименты по обучению моделей вышеописанных архитектур на синтетических выборках. Параметры генерации обучающей выборки:

TRAIN_EVAL_COEF = 0.7 # Соотношение объемов тренировочной и валидационной выборок

FILL_COEF = 0.9 # Вероятность того, что ячейка таблицы останется заполненной

NUM_SAMPLES = 100000 # Полный объем выборки

PIC_SIZE = 2048 # Размер генерируемых изображений

MAX_TABLE_SIZE = 12 # Максимальный параметр (ширина и высота) таблицы в выборке

TABLES = ['%dx%d' % (((i % MAX_TABLE_SIZE) + 1), j + 1) for i, j in enumerate(reduce(lambda x, y: x + y), [range(MAX_TABLE_SIZE)[n:] + range(MAX_TABLE_SIZE)[n:] for n in range(MAX_TABLE_SIZE)])] # Список возможных шаблонов для таблиц ('ширина x высота')

CELL_SIZE_SCALLER = 0.7 # Размер генерируемого значка по отношению к размеру ячейки

CELL_JITTER_COEF = 0.3 # Коэффициент случайного дрожания значков

CELL_FADE_COEF = 0.2 # Коэффициент случайного уменьшения значков

TABLE_JITTER_COEF = 0.2 # Коэффициент случайного дрожания таблицы по отношению к генерируемому изображению

DRAW_COLOR = (62, 231, 99) # Цвет рисования отладочных линий

INITIAL_Z = 1. # Минимальная координата смещения изображения в глубину (перспективные искажения)

MAX_Z = 3. # Максимальная координата смещения изображения в глубину (перспективные искажения)

Перспективные искажения генерируемых изображений моделировались в проективных координатах [9]. После генерации плоских (x, y) координат объектов для всей плоскости изображения высчитывалось ее смещение вдоль оси z , затем для каждой точки определялась ее третья координата (x, y, z) , после чего новые координаты объекта определялись по формуле $(\frac{x}{z}, \frac{y}{z}, 1)$. Плоские координаты $(\frac{x}{z}, \frac{y}{z})$ затем наносились на генерируемое изображение.

После генерации синтетической обучающей выборки проводились эксперименты по обучению моделей. Длительность обучения составляла порядка 13, 14 часов реального времени, параметры ЭВМ, на которой проводилось обучение, следующие:

ОЗУ: 3,7 ГБ

ЦП: Intel Core i3-6006U CPU @ 2.00GHz x 4

ГУ: Intel HD Graphics 520 (Skylake GT2)

Тип ОС: 64-бит

За время обучения линейная модель показала лучшую точность на валидационной выборке, равную 0,996, полносверточная модель показала лучшую точность на валидационной выборке, равную 1,000 (рис. 3).

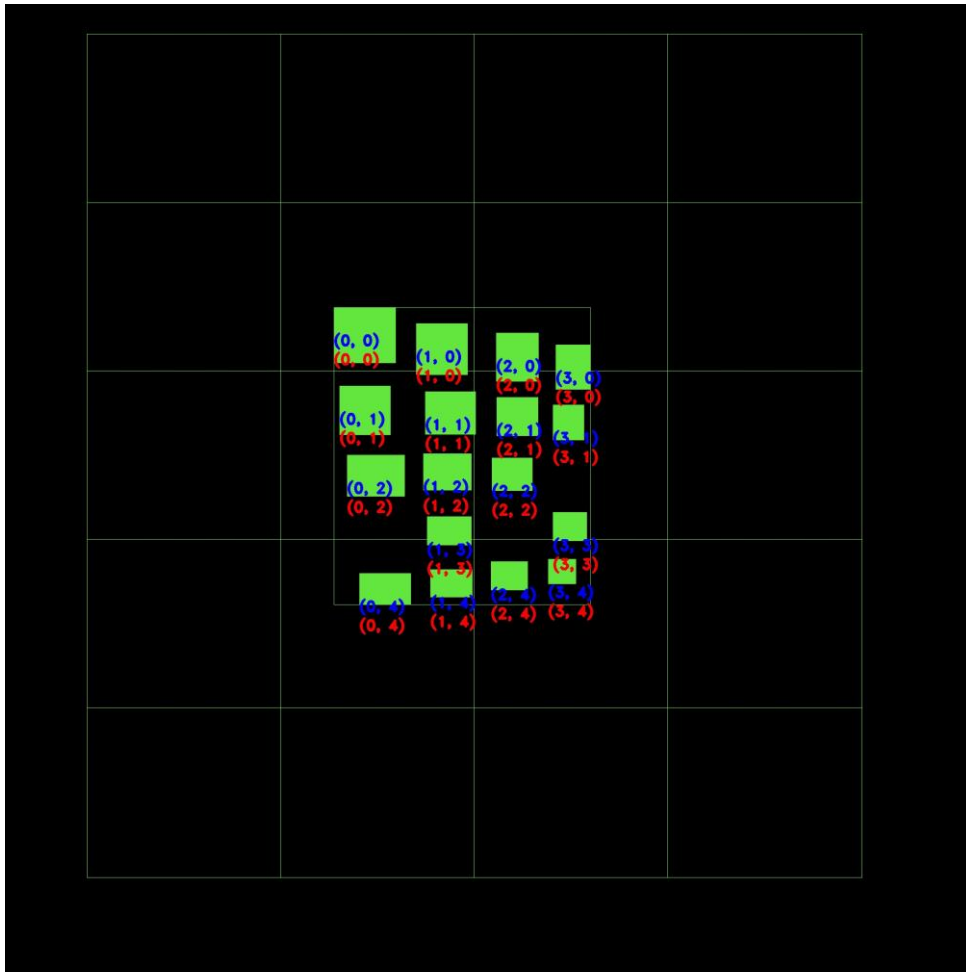


Рис. 3. Обучение моделей архитектур полносверточной и полносвязной нейронных сетей

После обучения проводилась проверка работы обученных моделей на валидационной выборке, сгенерированной из реальных изображений. Результаты точности на реальных данных следующие:

1. Линейная модель: 0,993.
2. Полносверточная модель: 0,996.

Авторы планируют продолжить работу по обобщению результатов исследования для пространственных объектов, а также разработку прототипов нейронных сетей распознавания взаимного расположения рукотворных объектов, включая движения человека.

Работа проводилась в рамках гранта РФФИ 18-07-00901 «Исследование и разработка системы распознавания элементов рукотворного интерьера на базе нейронных сетей для построения дополненной реальности и выработки алгоритмов взаимодействия управляемых объектов с реально-виртуальным окружением».

Литература

1. The Stanford Calibration Grid Detector. URL: <https://graphics.stanford.edu> (дата обращения: 20.11.2018).
2. Simple and effective table detection system from documents images. URL: <https://www.researchgate.net> (дата обращения: 20.11.2018).
3. Table Detection Using Deep Learning. URL: <https://www.researchgate.net> (дата обращения: 20.11.2018).

4. IOu. URL: <https://www.pyimagesearch.com> (дата обращения: 20.11.2018).
5. Minichino J., Howse J. Learning OpenCV 3 Computer Vision with Python. Second Edition. Birmingham : Packt Publishing, 2015. 266 p.
6. Бесшапошников Н. О., Кузьменко М. А., Леонов А. Г., Матюшин М. А. Некоторые вопросы эффективности детерминированных алгоритмов распознавания образов с помощью библиотеки OpenCV // Математическое и компьютерное моделирование сложных систем: теоретические и прикладные аспекты : тр. НИИСИ РАН. М. : ФГУ ФНЦ НИИСИ РАН, 2018. Т. 8, № 2. С. 65–68.
7. Перцептрон. URL: <http://www.machinelearning.ru> (дата обращения: 20.11.2018).
8. Сверточная нейронная сеть. URL: <https://habr.com> (дата обращения: 20.11.2018).
9. Проективные координаты и проективные преобразования. URL: http://old.pskgu.ru/ebooks/musch/musch_06_03.pdf (дата обращения: 20.11.2018).