МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

УДК 004.032.26; 004.93'1

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ В ЗАДАЧАХ ОБНАРУЖЕНИЯ И КЛАССИФИКАЦИИ ОБЪЕКТОВ

© **С. М. Борзов, Е. С. Нежевенко**

Институт автоматики и электрометрии СО РАН
630090, г. Новосибирск, просп. Академика Коптюга, 1
E-mail: borzov@iae.nsk.su, nejevenko@iae.nsk.su

Выполнен обзор основных идей, использованных при решении задач обнаружения и классификации объектов по их изображениям с применением нейросетевых технологий. Рассмотрены ключевые публикации, посвящённые наиболее популярным способам повышения точности классификации. Показано, что в последнее десятилетие нейросетевые методы обнаружения объектов достигли существенных успехов за счёт использования свёрточных технологий и практической реализации идеи глубокого обучения с применением объёмных баз данных. Проанализированы основные недостатки, ограничения и возможные направления развития существующих подходов.

Ключевые слова: нейросетевые технологии, обработка изображений, обнаружение и классификация объектов, свёрточные нейронные сети, глубокое обучение, комбинированные методы.

DOI: 10.15372/AUT20230307

Введение. Разработка и развитие современных методов обработки последовательностей регистрируемых изображений в оптико-электронных системах видеонаблюдения различного назначения с целью обнаружения объектов заданных классов является задачей, актуальность которой год от года только повышается [1, 2]. В их основе лежит комплекс алгоритмов компьютерного зрения, позволяющих проводить мониторинг и анализ информации без прямого участия человека. Указанные алгоритмы могут быть интегрированы в различные информационно-управляющие системы, использующие данные видеонаблюдения.

Компьютерное зрение [3, 4] является динамично развивающимся направлением современной науки, востребованным в различных областях, начиная с интеллектуальных человеко-машинных интерфейсов, принятия решений роботами и заканчивая системами автоматического контроля на производстве и системами поддержки принятия решений в специальных приложениях. Неотъемлемой его частью является распознавание образов, решающее задачу определения соответствия входного изображения к одному из хранимых эталонных изображений объектов.

Наибольшие успехи в рассматриваемой области, нашедшие эффективное применение в различных приложениях, связаны с программно-аппаратными средствами распознавания лиц и автомобильных номеров. Однако в настоящее время практическое применение методов обработки изображений может быть намного шире. В числе наиболее востребованных задач можно указать: детектирование и отслеживание траекторий движения, многокамерный трекинг, обнаружение, классификация и идентификация объектов, распознавание ситуаций, анализ поведения людей и т. д. Часть из них решается достаточно успешно, а другие требуют разработки новых информационных технологий.

Одним из ключевых моментов в современных системах видеонаблюдения является применение нейросетевых технологий обработки данных, позволяющих в реальном времени анализировать не только отдельные цифровые изображения, но и видеопотоки. Обнаружение объекта сводится к выделению прямоугольной области (региона) интереса на цифровом изображении в конкретный момент времени. Под областью интересов понимается множество пикселей цифрового изображения, очерчивающих искомый объект. Классификация подразумевает отнесение объекта, содержащегося в выделенной области интереса, к одному из предварительно заданных классов. Под обработкой в реальном времени понимается анализ видеопотока с частотой не менее 10 кадр/с. Все методы обнаружения и классификации объектов можно разделить на два основных класса: обучение с учителем и без учителя.

Обнаружение и классификация объекта в большинстве традиционных методов обучения с учителем осуществляется на основе анализа диаграмм рассеяния объектов обучающей выборки в многомерном пространстве признаков [5, 6]. На первом этапе в многомерном пространстве признаков на основе анализа изображений объектов, для которых известны классы, определяются области (кластеры), в которых наиболее часто встречаются экземпляры каждого из классов. На втором — рассчитывается близость классифицируемых объектов к каждому из образованных кластеров и определяется среди них ближайший. Различные методы обнаружения/классификации различаются между собой в первую очередь способом определения указанной близости. При реализации методов обнаружения/классификации с обучением без учителя также используется распределение данных в пространстве признаков, однако границы кластеров определяются не по обучающим, а непосредственно по классифицируемым данным.

Примерно каждые десять лет в области обнаружения объектов появляется новая эффективная идея, заставляющая пересмотреть все предыдущие достижения. Так, в 2001 г. был предложен эффективный алгоритм обнаружения объектов (метод Виола — Джонса) [7]. Его реализация осуществлялась в реальном времени на веб-камере и продемонстрировала большие потенциальные возможности компьютерного зрения. Этот алгоритм был реализован в OpenCV, и метод стал синонимом обнаружения лиц. В 2005 г. был представлен метод обнаружения на основе гистограммы направленных градиентов, который значительно превзошёл существующие на тот момент алгоритмы поиска пешеходов [8].

Прорыв в области обнаружения объектов с применением нейронных сетей (НС) произошёл благодаря появлению в 2010 г. набора данных ImageNet [9]. Он содержит более 14 млн изображений, каждому из которых вручную был присвоен класс объекта (более 22000 классов), который на нём изображён. Наличие такой базы данных, на которой можно обучать коэффициенты фильтров, дало старт новому этапу в развитии систем обнаружения и распознавания. Начиная с 2010 г., проводится ежегодный конкурс программного обеспечения ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [10], являющийся частью конкурса Pascal Visual Object Challenge (Pascal VoC) [11], в рамках которого программы соревнуются в правильной классификации и обнаружении объектов.

Новой идеей последнего десятилетия, хорошо показавшей себя в данном конкурсе, стало глубокое обучение. Алгоритмы глубокого обучения существовали и ранее, но они стали широко использоваться в компьютерном зрении именно благодаря их успеху на конкурсе ILSVRC в 2012 г. В этом конкурсе модель AlexNet, основанная на глубоком обучении, продемонстрировала поразительную точность 85 %, что на 11 % лучше, чем алгоритм, занявший второе место. Это была единственная работа, основанная на глубоком обучении. Но в 2013 г. уже все победившие работы были основаны на глубоком обучении, а в 2015 г. несколько алгоритмов под общим названием «Свёрточная нейронная сеть» превзошли уровень естественного распознавания человеком 95 %. Оценки точности получены при

обучении сетей на выборке из набора данных ImageNet, состоящей из ∼1000 изображений объектов для каждого из 1000 непересекающихся классов.

Ещё одной отличительной особенностью последних десятилетий стало создание большого количества свободно распространяемых библиотек, содержащих популярные модели нейронных сетей для классификации объектов, таких как Google Glass [12], Microsoft HoloLens [13], Kinect SDK [14], PyTorch [15], Vuforia SDK [16], Kudan SDK [17], OpenCV [18]. Создание системы распознавания с их применением возможно без глубоких технических знаний в области машинного обучения и не требует огромного количества ресурсов на оборудовании пользователя. Они позволяют легко реализовать алгоритмы обнаружения и классификации объектов, используя всего несколько строк кода. Это привело к тому, что в сети Интернет появилось большое количество публикаций с результатами применения многочисленных моделей для решения различных практических задач.

В настоящее время опубликованы объёмные и обстоятельные обзоры. Так, несомненный интерес представляет большой обзор статей по обнаружению объектов за 20 лет [19], в котором, в частности, приведена диаграмма увеличения числа публикаций с 1998 по 2018 гг. в этой области в Google scholar. Следует отметить тематический обзор [20]. В нём рассмотрены критерии качества алгоритмов обнаружения, классические статистические обнаружители, использующие математические модели случайных полей, а также современные решения на базе архитектур свёрточных нейронных сетей из популярной библиотеки для машинного обучения TensorFlow [21]. В этой работе приведена точка зрения на основные перспективы и тенденции в задаче обнаружения объектов на изображениях. Кроме того, имеется большое количество интернет-источников с описанием структур нейронных сетей. В частности, можно отметить ресурс [22], содержание которого постоянно дополняется.

Целью данной работы является анализ существующих и перспективных нейросетевых методов классификации объектов по изображениям и реализующих их нейронных сетей.

Классические нейронные сети. Основная идея, лежащая в основе НС — это последовательное преобразование сигнала параллельно работающими функциональными элементами, искусственными нейронами [23]. На вход такого нейрона поступает некоторое множество сигналов. Каждый вход умножается на соответствующий коэффициент, и все произведения суммируются. Полученный сигнал преобразуется активационной функцией нейрона. На вход нейронов первого слоя НС, предназначенной для классификации изображений объектов, поступают параметры изображения; выходной слой имеет размерность $1 \times 1 \times n$ (где n — количество определяемых классов) и содержит оценки принадлежности объектов к каждому из классов.

Активационная функция может быть обычной линейной функцией или пороговой (рис. 1). В последнем случае, если взвешенная сумма входных сигналов не достигает неко-

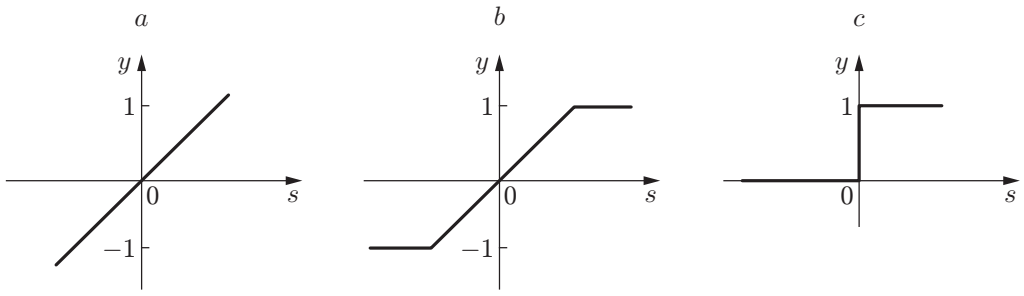

Рис. 1. Примеры простых функций активации: a — линейная; b — линейная с ограниченной областью изменения; c — пороговая (функция Хевисайда)

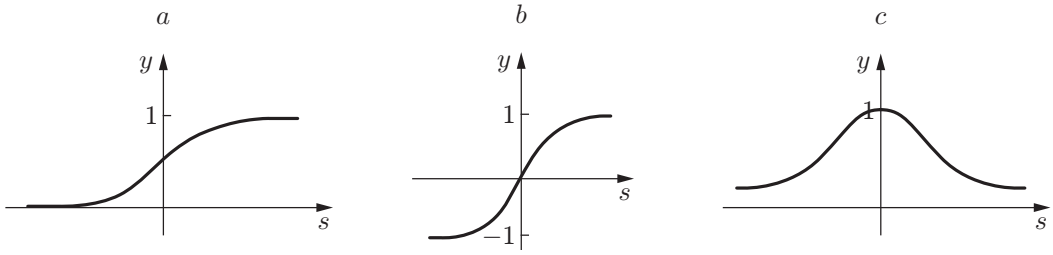

Рис. 2. Примеры нелинейных функций активации: a — сигмоидальная; b — гиперболического тангенса; c — радиально-базисная

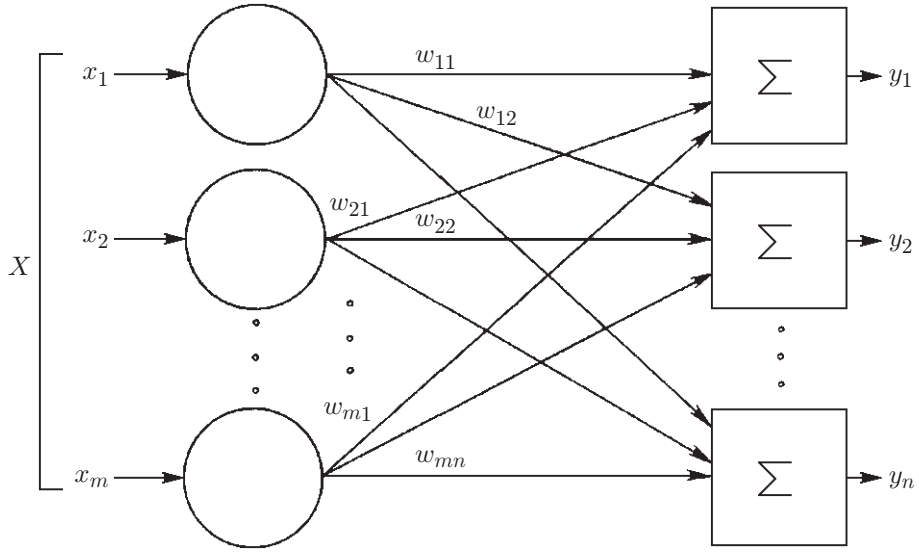

Рис. 3. Однослойная нейронная сеть [23]

торой пороговой величины, то нейрон не возбуждён и его выходной сигнал равен нулю, если достигает — то нейрон переходит в возбуждённое состояние и на его выходе образуется сигнал 1.

Для моделирования нелинейной характеристики нейрона в качестве активационной используются более сложные функции (рис. 2), которые расширяют возможности НС.

Простейшая сеть состоит из группы нейронов, образующих слой, как показано на рис. 3.

Каждый элемент из множества входов X со своим весовым коэффициентом соединён с каждым нейроном, на выход которого выдаётся взвешенная сумма входов (такая сеть называется полносвязной). На практике многие соединения могут отсутствовать, кроме того, могут также иметь место соединения между выходами и входами элементов в слое. Многослойные сети образуются каскадами слоёв, в которых выход одного слоя является входом для последующего. В более общих случаях сети имеют также соединения от выходов к входам нейронов предыдущих слоёв. Такие сети называют сетями с обратными связями.

Различные способы объединения нейронов между собой и организация их взаимодействия привели к созданию сетей разных типов [24]. На рис. 4 приведена схема, объединяющая основные типы искусственных НС.

Обучение НС, предназначенной для обнаружения и классификации объектов по их изображениям, осуществляется путём последовательного предъявления входных векторов

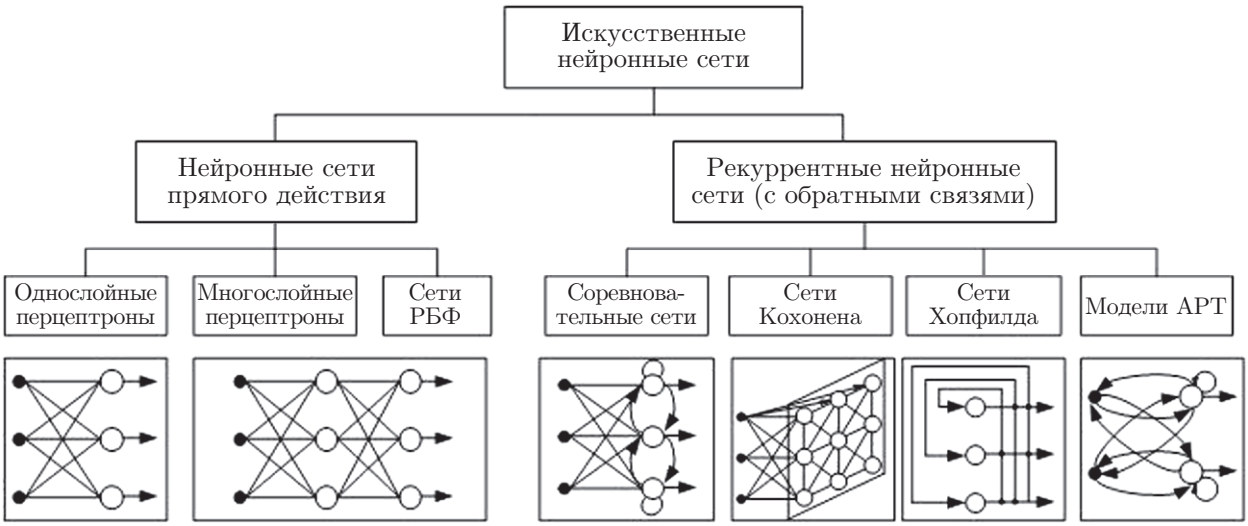

Рис. 4. Обобщённая классификация НС [24, 25]

(изображений известных классов) с одновременной подстройкой весов в соответствии с определённой процедурой. Цикл алгоритма, в течении которого предъявляются все обучающие примеры, называется эпохой. В процессе обучения веса сети постепенно становятся такими, что для каждого входного изображения вырабатывается выходной вектор, в котором компонента, соответствующая его классу, максимальна.

Алгоритмов обучения много, однако наиболее используемым является алгоритм обратного распространения ошибки (backpropagation), который применяется для обучения плоскослоистых НС прямого распространения (многослойных персептронов) [26–28]. Этот алгоритм относится к методам обучения с учителем, поэтому требует задания целевых значений в обучающих примерах. В основе идеи алгоритма лежит использование выходной ошибки НС для вычисления величин коррекции весов нейронов в её скрытых слоях:

$$E = \sum_{i=1}^{k} (y - y')^2,$$

где k — число выходных нейронов сети, y — целевое значение, y' — фактическое выходное значение.

Алгоритм является итеративным и применяет принцип обучения «по шагам», когда веса нейронов сети корректируются после подачи на её вход одного обучающего примера. Начальная конфигурация сети выбирается случайным образом. На каждой итерации происходит два прохода сети — прямой и обратный. На прямом входной вектор распространяется от входов сети к её выходам и формирует некоторый выходной вектор, соответствующий текущему состоянию весов. Затем вычисляется ошибка НС как разность между фактическим и целевым значениями. На обратном проходе эта ошибка распространяется от выхода сети к её входам и производится коррекция весов нейронов в соответствии с правилом

$$\Delta\omega_{j,i(n)} = -\eta \frac{\partial E}{\partial \omega_{ij}},$$

где ω_{ij} — вес i-й связи j-го нейрона; η — параметр скорости обучения, который позволяет дополнительно управлять величиной шага коррекции $\Delta\omega_{j,i(n)}$ с целью более точной

настройки на минимум ошибки и подбирается экспериментально в процессе обучения (изменяется в интервале от 0 до 1).

Причём корректируются сначала параметры выходного слоя, затем предыдущих. Процесс обучения останавливается тогда, когда пройдено определённое количество эпох, либо ошибка достигнет некоторого определённого уровня малости, либо когда ошибка перестанет уменьшаться.

При моделировании НС с линейными функциями активации нейронов можно построить алгоритм, гарантирующий достижение абсолютного минимума ошибки обучения. Для НС с нелинейными функциями активации в общем случае нельзя гарантировать достижение глобального минимума функции ошибки.

Главное преимущество НС — гибкость. Геометрически в многомерном пространстве признаков разделяющая классы поверхность в случае применения НС представляет собой множество гиперплоскостей. Каждая из областей, на которые гиперплоскости разбивают пространство признаков, относится к одному из классов. Важно понимать, что, поскольку найденный минимум среднеквадратичной ошибки сети будет локальным, найденная разделяющая поверхность не будет являться ни единственным, ни оптимальным решением.

При этом следует иметь в виду, что предложенный алгоритм склонен к переобучению — столь точному формированию разделяющей поверхности, что безошибочно классифицируются только обучающие данные. Цель же машинного обучения состоит в том, чтобы научить алгоритм обобщать полученную информацию и верно обрабатывать новые, ранее не встречавшиеся данные. Поэтому значительное внимание уделяется развитию методов, которые в частных случаях помогают решать эту проблему.

Обычные НС плохо подходят для обработки больших изображений. Так, в наборе данных CIFAR-10 [29] содержатся изображения размером $32 \times 32 \times 3$ (32 пикселя высота, 32 пикселя ширина, 3 цветовых канала), включающие объекты 10 классов. Обработка таких изображений требует, чтобы каждый полносвязный нейрон первого скрытого слоя имел $32 \times 32 \times 3 = 3\,072$ коэффициента, при этом с увеличением размера изображения количество коэффициентов увеличивается квадратично ($224 \times 224 \times 3$ пикселя — 150 528 коэффициентов на один нейрон). Учитывая, что для организации НС понадобится не один подобный нейрон, общее количество коэффициентов быстро начинает расти. Становится очевидным, что полносвязность чрезмерна и большое количество параметров быстро приведёт сеть к переобученности.

Вычислительная сложность нейросетевых алгоритмов классификации также квадратично зависит от числа нейронов в скрытом слое. Для задач обнаружения и классификации объектов на изображениях скорость обработки с применением классических НС, как правило, является недостаточной для решения задач в реальном времени (в темпе потока данных) [30].

Свёрточные нейронные сети. В последние годы большинство разработок по алгоритмам обнаружения и классификации объектов было сосредоточено на использовании многослойных свёрточных НС, модификации которых описаны в многочисленных публикациях, например [31–33].

Свёрточные НС очень похожи на обычные нейронные сети. Они состоят из нейронов, которые содержат изменяемые коэффициенты (ядро преобразования). В каждом нейроне вычисляется скалярное произведение связанного с ним фрагмента входного изображения с ядром, после чего применяется нелинейная функция активации. Вся сеть представляет собой единую функцию оценки от исходного набора пикселей изображения до распределения вероятностей принадлежности к определённому классу. У этих НС также на последнем полносвязном слое возникает ошибка, и все методы, относящиеся к обучению обычных НС для её минимизации, применимы для свёрточных НС.

Архитектура свёрточных НС явно предполагает получение изображений на входе, что позволяет учесть определённые свойства входных данных в самой архитектуре сети. Эти свойства позволяют реализовать функцию прямого распространения эффективнее и сильно уменьшают общее количество параметров в сети.

В свёрточных НС нейроны одного слоя связаны с небольшим количеством нейронов предыдущего слоя вместо того, чтобы быть связанными со всеми предыдущими нейронами слоя. Выходной же слой сети для классификации изображений объектов 10 классов будет иметь размер $1 \times 1 \times 10$.

В свёрточной НС используется три главных типа слоёв: свёрточный слой (convolution) с поэлементной функцией активации, слой подвыборки (pooling/subsampling) и полносвязный слой (full connection, как в обычной НС), которые осуществляют последовательное преобразование данных.

При классификации набора данных CIFAR-10 на вход сети поступает изображение размером 32×32 пикселя с тремя цветовыми каналами R, G, B ($32 \times 32 \times 3$).

Свёрточный слой представляет собой набор нейронов, которые связаны с локальной областью входного изображения, в каждом из которых вычисляется соответствующее скалярное произведение. Если использовать 12 ядер размером $3 \times 3 \times 3$, на выходе формируется массив размером $32 \times 32 \times 12$. К массиву поэлементно применяется функция активации. Это преобразование не изменяет размерности данных.

Слой подвыборки производит операцию сжатия изображения по двум измерениям — высоте и ширине, что в результате даст новое $3D$-представление размером $16 \times 16 \times 12$.

Полносвязный слой вычисляет оценки по классам, результирующий размер равен $1 \times 1 \times 10$, где каждое из 10 значений будет соответствовать оценке определённого класса среди 10 категорий изображений из CIFAR-10. Как и в обычных НС, каждый нейрон этого слоя связан со всеми нейронами предыдущего слоя.

Слои каждого типа могут повторяться по несколько раз. Именно таким образом свёрточная НС преобразует исходное изображение слой за слоем — от начального значения пикселя до итоговой оценки класса. При этом свёрточные и полносвязные слои осуществляют трансформацию, которая является не только функцией, зависящей от входных данных, но и от внутренних значений весов и смещений в самих нейронах. Параметры в этих слоях подбираются в процессе обучения таким образом, чтобы входные данные формировали соответствующие им выходные отклики.

В 1998 г. первая работающая свёрточная нейронная сеть LeNet была внедрена в США для распознавания индексов (рис. 5). Её архитектура [34] была стандартным шаблоном для построения свёрточных сетей: свёртка чередуется со слоем подвыборки несколько раз, затем следует несколько полносвязных слоёв. Такая НС содержит 60 тыс. параметров. Её основные операции: свёртка 5×5 (без учёта глубины) со сдвигом 1 и подвыборка 2×2 со сдвигом 2. Отметим, что в первом свёрточном слое, работающем с одноканальными

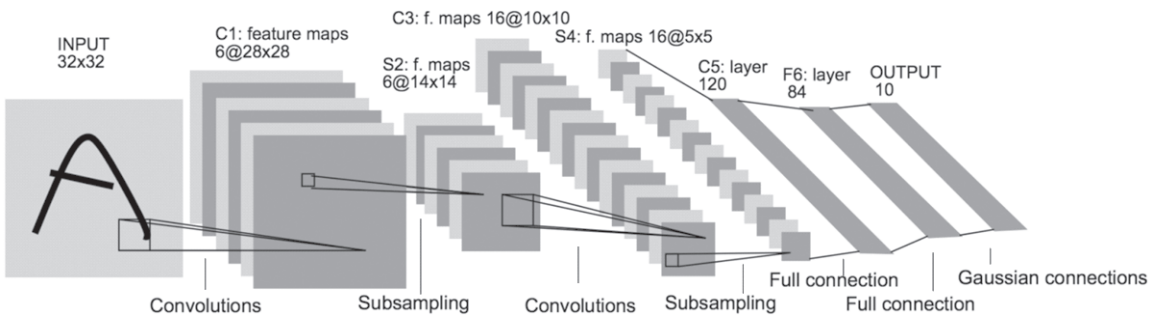

Рис. 5. Структура свёрточной нейронной сети LeNet для распознавания индексов [34]

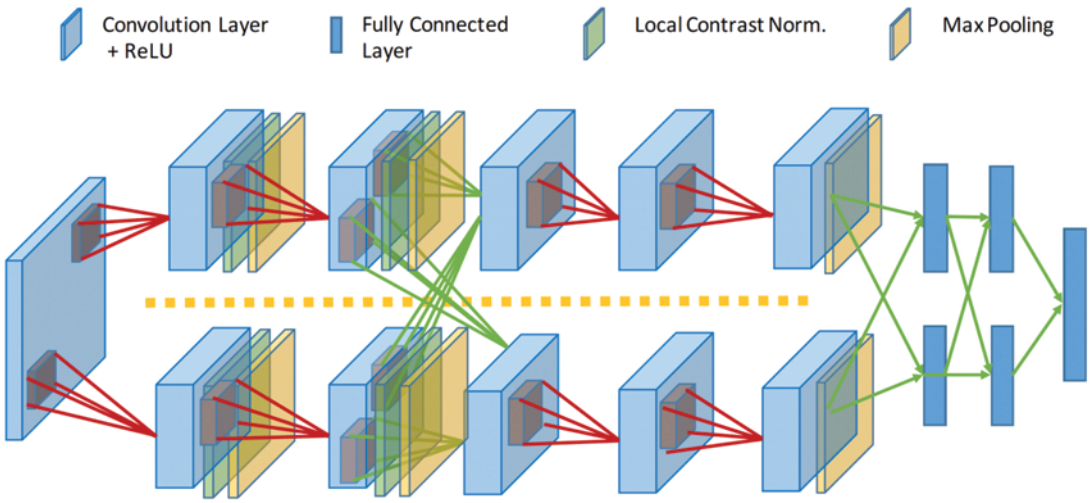

Рис. 6. Структура свёрточной нейронной сети AlexNet [35]

входными изображениями, размерность ядра составляет $5 \times 5 \times 1$. Поскольку таких ядер в слое 6, то в результате получается 6-канальное изображение, а во втором свёрточном слое применяются уже ядра размером $5 \times 5 \times 6$. И далее количество каналов выходного изображения после каждого свёрточного слоя равняется количеству ядер текущего слоя и глубине ядра следующего свёрточного слоя (третья компонента размерности).

Развитие аппаратных средств, в частности GPU, позволило значительно увеличить размер входного изображения и количество обучаемых параметров. Так, в 2012 г. была представлена модель AlexNet [35], содержащая уже 62,3 млн параметров и выполняющая 1,1 млрд операций при прямом проходе. Свёрточные слои, на которые приходится 6 % всех параметров, производят 95 % вычислений. Для обучения данной сети были использованы два графических ускорителя, поэтому она в своей исходной версии разделена на две части (рис. 6). В такой конфигурации 90 эпох обучения занимают 6 дней на двух GPU Nvidia Geforce GTX 580.

С точки зрения топологии AlexNet близка к LeNet, но по количеству параметров увеличена в 1000 раз. Добавилось ещё несколько свёрточных слоёв, а размер ядер свёртки уменьшается от слоя к слою. Это объясняется тем, что пиксели входных изображений сильно скоррелированы, и область свёртки может быть больше, чем в других слоях без потери полезной информации. В качестве функции активации используется функция $\max(0, x)$ (ReLU — rectified linear unit). Затем применяется подвыборка (в данном случае функция Max Pooling — выбор максимального значения по окрестности 3×3 с шагом сдвига 2×2), уменьшающая ширину и высоту изображения, и логично использовать меньшую область.

В итоге в [35] была предложена пирамида из свёрток $11 \times 11 \to 5 \times 5 \to 3 \times 3$. В частности, в модели сети AlexNet с сервиса PyTorch к исходному 3-канальному изображению в первом свёрточном слое применяются 64 фильтра размером $11 \times 11 \times 3$ с шагом 4×4, во втором — 192 фильтра размером $5 \times 5 \times 64$, в третьем — 384 фильтра размером $3 \times 3 \times 192$, в четвёртом — 256 фильтров размером $3 \times 3 \times 384$ и в последнем пятом свёрточном слое — 256 фильтров размером $3 \times 3 \times 256$. После первого, второго и пятого свёрточных слоёв применяется подвыборка в окне 3×3 с шагом 2×2. В результате исходный размер массива преобразуется: $224 \times 224 \times 3 \to 55 \times 55 \times 64 \to 27 \times 27 \times 64 \to 13 \times 13 \times 192 \to 13 \times 13 \times 384 \to 13 \times 13 \times 256 \to 6 \times 6 \times 256$ (рис. 7).

После последнего свёрточного слоя и подвыборки формируется 256-канальное изображение размером 6×6 пикселей. Далее все эти значения подаются на полносвязные слои

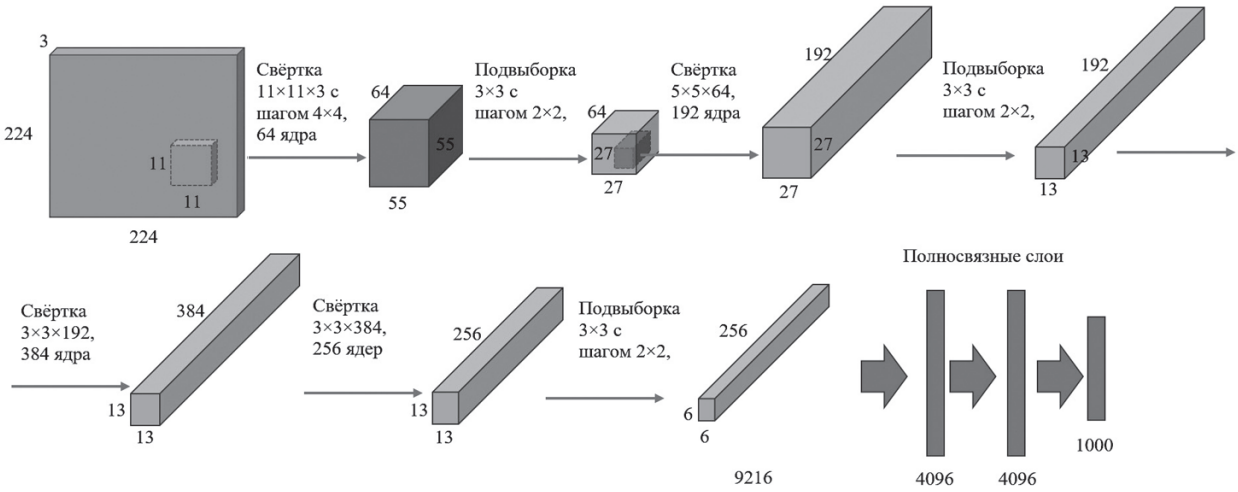

Рис. 7. Преобразование данных в свёрточной нейронной сети AlexNet

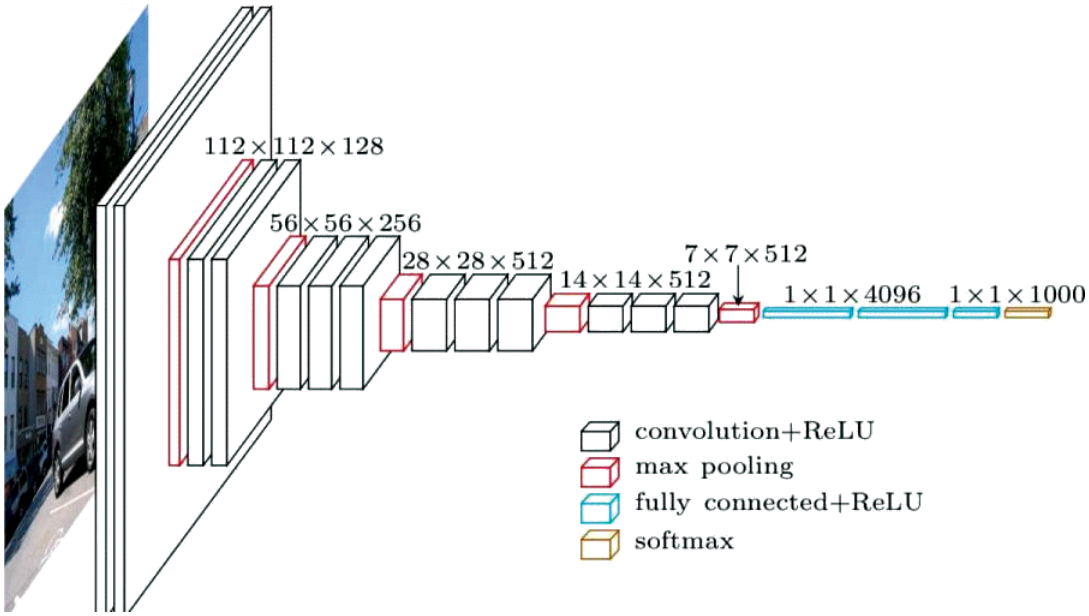

Рис. 8. Структура свёрточной нейронной сети VGG16 [40]

следующих размеров: $6 \times 6 \times 256$ (9216) $\rightarrow 1 \times 1 \times 4096 \rightarrow 1 \times 1 \times 4096 \rightarrow 1 \times 1 \times 1000$. Последнее число определяется количеством классов объектов для сети, обученной на наборе данных ImageNet. Именно в AlexNet, чтобы избежать переобучения на этапе полносвязных слоёв, впервые был применён инструмент DropOut (удаление заданной доли связей) [36], ставший стандартным для всех последующих свёрточных сетей. В оригинальной версии сети AlexNet доля удалённых связей составляет 0,5, т. е. удаляется каждая вторая связь.

Несмотря на то что сеть AlexNet была предложена одной из первых более 10 лет назад, она содержала многие ключевые моменты, обеспечившие успех нейронных сетей, и до сих пор используется для приложений [37, 38].

В свою очередь, улучшенной версией AlexNet явилась VGG16 [39]. В ней большие фильтры размером 11 и 5 в первом и втором свёрточных слоях заменены несколькими фильтрами размером 3×3, следующими один за другим (рис. 8) [40]. Используется тот факт, что свёртка с ядром 5×5 может быть представлена двумя свёртками размером 3×3.

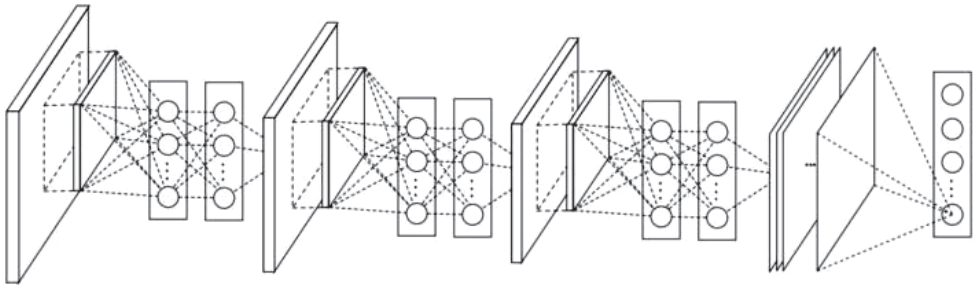

Рис. 9. Структура свёрточной НС с применением свёрток 1×1 [41]

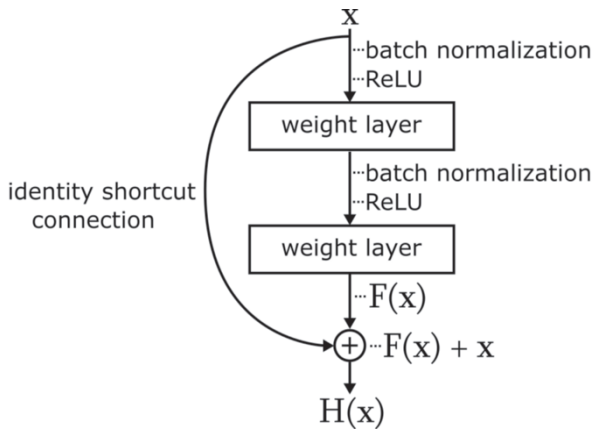

Рис. 10. Остаточный блок в том виде, в котором он использовался в ResNet [42]

При этом количество коэффициентов с 25 уменьшается до 18. Модель достигает точности 92,7 % при тестировании на ImageNet в задаче распознавания объектов на изображении. При этом сеть обучается на протяжении нескольких недель при применении видеокарт NVIDIA TITAN BLACK.

В основе дальнейшего развития подхода лежит простая идея: использование свёрток с ядром $1 \times 1 \times n$ для комбинирования полученных свёрточных признаков [41]. После каждого свёрточного слоя берутся дополнительные слои, чтобы скомбинировать полученные признаки перед подачей в следующий свёрточный слой (рис. 9). Глубина фильтра n (размерность по третьей координате) в дополнительном слое каждый раз выбирается равной количеству фильтров на данном свёрточном слое, а количество ядер $1 \times 1 \times n$ определяет глубину выходного представления. Ширину и высоту массива данных такое преобразование не изменяет. Этот подход отличается от непосредственного использования выходов нейронов в качестве входных данных для следующего слоя. Показано, что привлечение таких свёрток позволяет комбинировать признаки лучше, чем простое увеличение количества свёрточных слоёв.

Данный приём позволяет существенно повысить эффективность отдельных свёрточных слоёв посредством их комбинирования в более сложные группы. Эта идея позднее была применена в других архитектурах, таких как ResNet [42], Inception [43] и их вариантах.

Дальнейшие эксперименты с НС показали, что простое наращивание количества слоёв для повышения глубины сети рано или поздно вызывало ограничение или даже быстрое ухудшение её производительности. Это неожиданный результат, поскольку, если на предыдущих слоях был достигнут предел качества, сеть по логике далее должна просто использовать тождественное преобразование. Но этого не происходит, возможно, из-за сложности процесса обучения.

Важнейшим прорывом в архитектуре свёрточных НС стал механизм, предложенный в сетях класса ResNet [42] (Residual Network — «остаточная сеть»). Ключевая идея состоит во введении остаточных блоков (residual blocks), которые содержат «обходную связь идентичности» (identity shortcut connection) через один или большее количество слоёв (рис. 10). Таким образом разработчики архитектуры напрямую помогают сети при необходимости формировать тождественное преобразование. Такое решение позволило обучать сотни или даже тысячи слоёв с хорошей эффективностью. Сети ResNet достигли высокой общей производительности в задачах распознавания изображений и быстро стали одними из самых популярных в различных задачах компьютерного зрения.

Первой архитектурой ResNet была Resnet-34, которая включала в себя вставку коротких подключений для превращения обычной сети в её аналог — остаточную сеть (рис. 11).

В работе [44] представлена новая интерпретация остаточных сетей, показывающая, что они являются ансамблями экспоненциально большого числа мелких сетей. Это подтверждается крупномасштабным исследованием с удалением или изменением порядка остаточных блоков, которое демонстрирует, что они ведут себя во время тестирования точно так же, как ансамбли. Показано, что эти ансамбли в основном состоят из относительно неглубоких сетей ($\sim$20 слоёв). Это говорит о том, что в дополнение к описанию НС в терминах ширины и глубины существует третье измерение: множественность, размер неявного ансамбля. В конечном счёте остаточные сети не решают проблему ухудшения работы с увеличением количества слоёв, скорее, они избегают её, просто объединяя множество коротких сетей вместе.

Успех ResNet позволил предположить, что обходные соединения в CNN позволяют обучать более глубокие и точные модели. Развивая эту идею в [45], авторы представили сеть DenseNet (Densely Connected Convolutional Network), в которой введено соединение каждого слоя со всеми другими слоями (рис. 12). Важно отметить, что в отличие от ResNet признаки, прежде чем будут переданы в следующий слой, не суммируются, а объединяются (channel-wise concatenation). При этом количество параметров сети DenseNet намного меньше, чем у сетей с такой же точностью работы. Показано, что НС DenseNet работает особенно хорошо на малых наборах данных.

Автоматический поиск параметров и архитектуры сети. Дальнейшее развитие нейросетевых технологий связано с автоматическим поиском параметров и архитектуры сети.

В первую очередь внимание привлекает класс новых моделей EfficientNet, который получился из изучения способов масштабирования моделей разрешения в сети. В [46] предлагается новый метод составного масштабирования (compound scaling method), который равномерно масштабирует глубину/ширину/разрешение с фиксированными пропорциями между ними. Процедура получила название скейлинг и заключается в том, что фиксируются производимые внутри сети операции и меняются лишь глубина (количество повторений одних и тех же модулей), ширина (количество каналов в свёртках) и разрешение изображений. Скейлинг формулируется как проблема оптимизации точности классификации при существующих ограничениях по памяти и по частоте кадров.

Непосредственное продолжение данные исследования получили в работе [47]. Здесь предложена архитектура EfficientDet (рис. 13). В качестве основы она использует EfficientNet [48] с добавлением слоя по работе с двунаправленной пирамидой разномасштабных признаков под названием BiFPN, за которым идёт «стандартная» сеть вычисления класс/рамка объекта. Данная сеть, по утверждению авторов, достигает гораздо большей эффективности, чем предыдущие, при широком спектре ограничений ресурсов.

Следует также отметить архитектуру DetNASNet [49], использующую подход под названием Neural Architecture Search (NAS) для разработки архитектур, которые обнаруживают объекты. Утверждается, что здесь впервые был применён процесс NAS, т. е.

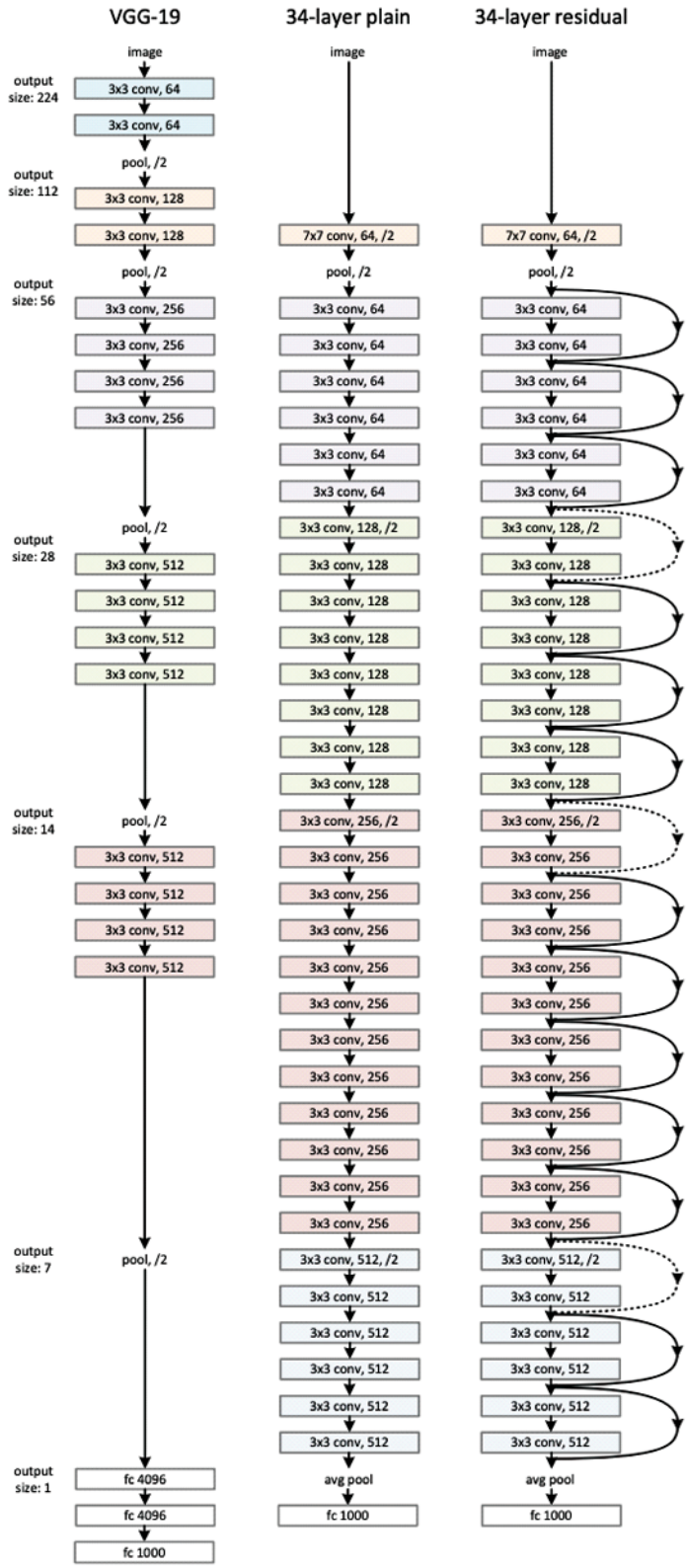

Рис. 11. Примеры сетевых архитектур: модель VGG-19 (слева), простая 34-слойная сеть (в центре), ResNet с 34 слоями [42] (справа)

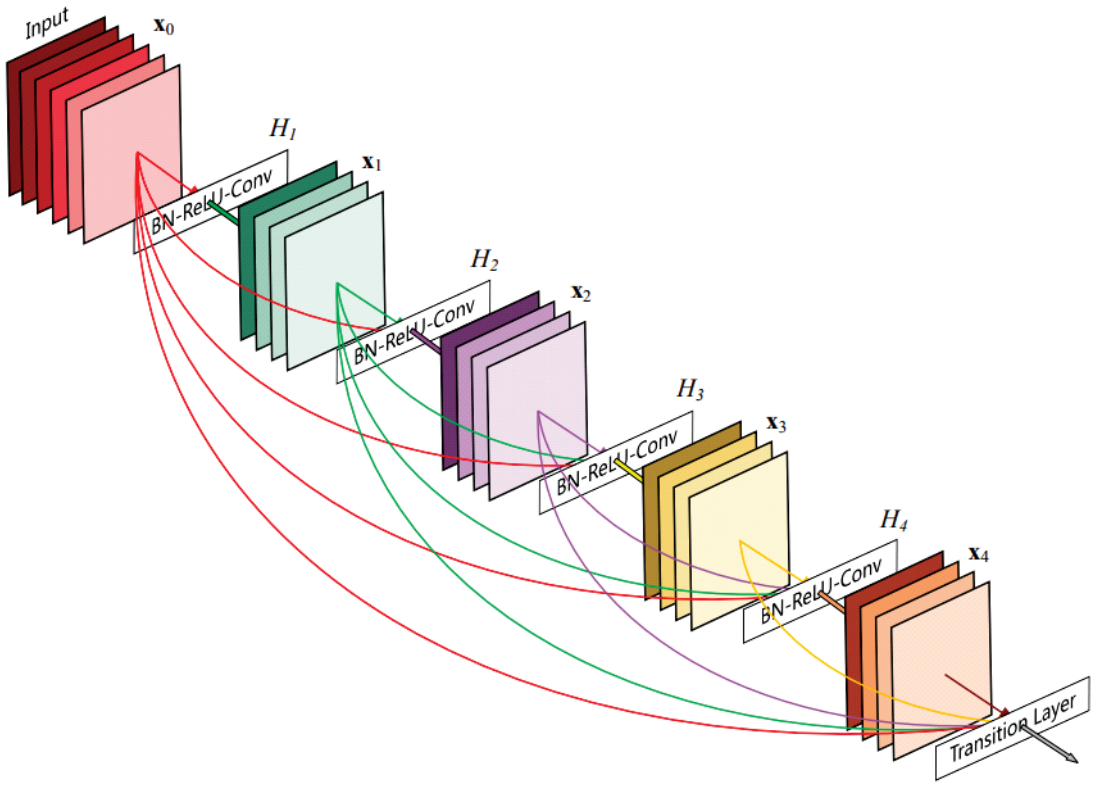

Рис. 12. Архитектура DenseNet [45]

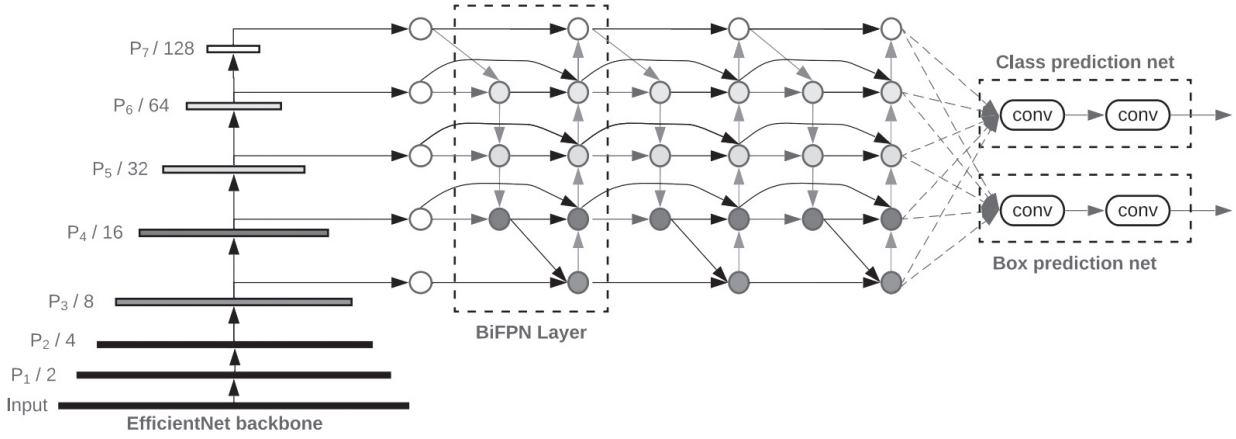

Рис. 13. Архитектура EfficientDet [47]

автоматический поиск оптимальных гиперпараметров НС для оптимизации задачи обнаружения объектов. Метод состоит из трёх этапов: предварительное обучение базовой сети на наборе данных Image NET, точная настройка параметров базовой сети на обучающей выборке из набора данных COCO [50], подбор архитектуры обученной сети с помощью эволюционного алгоритма на валидационной выборке набора COCO (рис. 14).

Длительность обучения средствами GPU на наборе данных COCO составляла 44 дня. Достигнутая точность оказалась лучше, чем у ResNet-101, при гораздо большем быстродействии.

В свою очередь, в [51] предлагается двухэтапная стратегия грубого поиска под названием Structural-to-Modular NAS (SM-NAS): первый этап на структурном уровне направлен

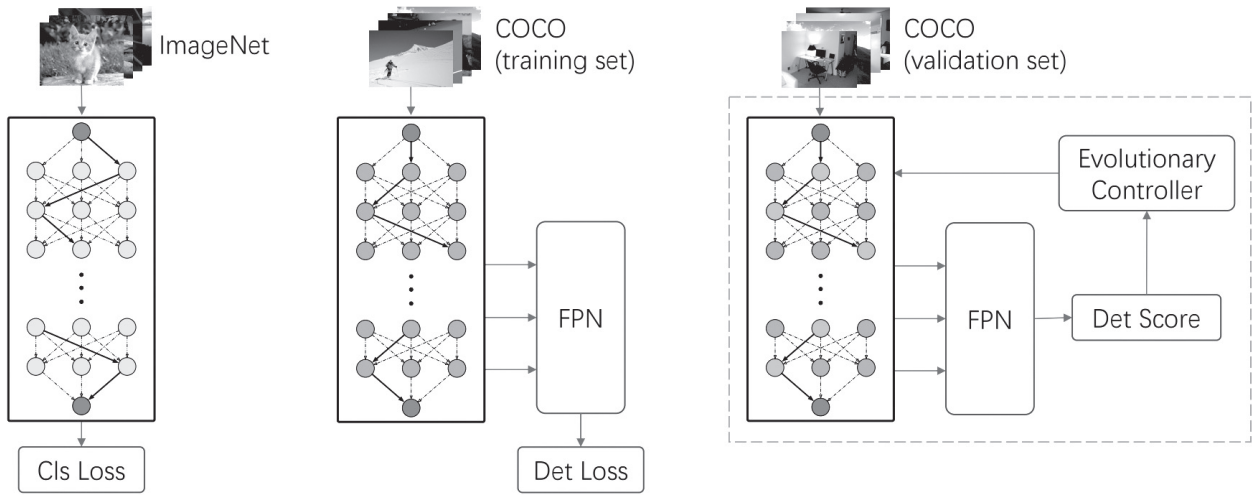

Рис. 14. Конвейер DetNAS для поиска архитектуры базовой сети детектора объектов [49]

на поиск эффективной комбинации различных модулей; второй этап поиска на модульном уровне развивает каждый конкретный модуль и продвигает фронт Парето [52] (состояние системы, при котором ни один показатель системы не может быть улучшен без ухудшения какого-либо другого показателя) вперёд к более быстрой сети для конкретных задач.

Комбинированные методы обнаружения и классификации объектов. В настоящее время понятие нейронных сетей сильно трансформировалось. Вначале они привлекали исследователей своей простотой и эффективностью. Действительно, относительно простые по своей структуре сети с простейшими однотипными нейронами оказались способными при соответствующем обучении успешно решать довольно сложные проблемы. В последнее десятилетие ситуация кардинально поменялась. Сети теперь — довольно сложные образования с осмысленной структурой. Они представляют собой ансамбли и каскады различных классификаторов. Основные усилия разработчиков направлены на правильное сочетание результатов работы отдельных классификаторов и на разработку наиболее эффективной тактики их обучения. Сами классификаторы могут быть реализованы как на основе НС, так и с применением классических информационных технологий. Важно, чтобы они были распараллеливаемыми и обеспечивали возможность проведения вычислений в реальном времени. По этой причине предпочтение отдаётся нейросетевой реализации (даже если реализуются классические методы классификации). К тому же для пользователя НС практически ничего не изменилось. Он получает некоторую уже предобученную сеть и пользуется ею как неким чёрным ящиком с инструкцией как его дообучать для решения конкретных прикладных задач. Бóльшую часть работы по созданию системы распознавания, заключающуюся в выборе архитектуры сети и тактики проведения обучения, выполнили её разработчики.

Комбинированные методы основаны на сочетании нескольких алгоритмов классификации для улучшения их эффективности. Такие методы более устойчивы к шуму и различным видам искажений объекта. Их можно разделить на две группы: применяющие каскадирование и ансамблевый подход. Каскадирование основано на объединении нескольких классификаторов с использованием всей информации, собранной с выхода данного классификатора в качестве дополнительной информации для следующего классификатора в каскаде. Ансамблевый подход предполагает параллельное применение ряда классификаторов к одним и тем же входным данным с последующим голосованием. Таким образом, ансам-

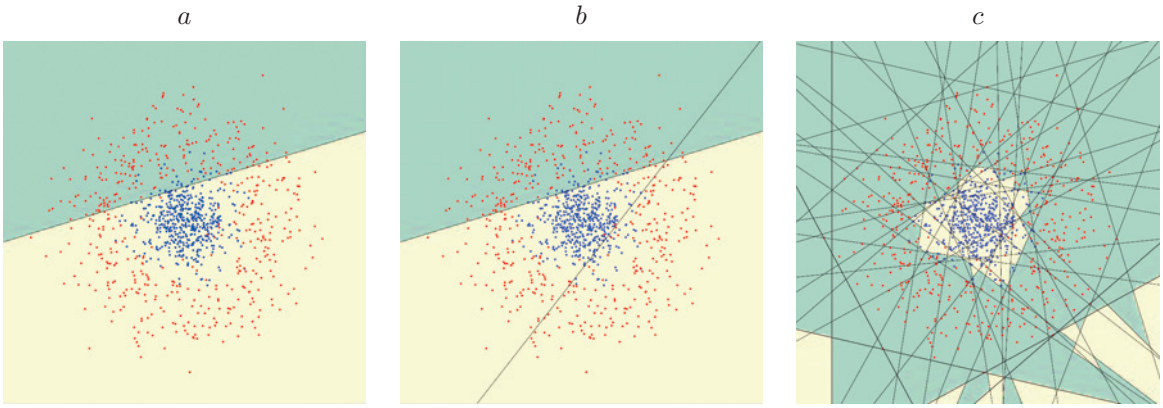

Рис. 15. Иллюстрация работы алгоритма бустинга: *a* — результат работы одного слабого (линейного) классификатора; *b* — применение второго линейного классификатора, обученного на неразделённых первым классификатором данных; *c* — результат работы комитета слабых классификаторов [53]

блевые классификаторы являются многоэкспертными системами, тогда как каскадные — многоступенчатыми. Методы совмещения результатов работы обычных классификаторов в составе комбинированного называются бустингом (boost — улучшение, усиление).

Идея бустинга была предложена в конце 90-х гг. [53], когда надо было найти решение вопроса о том, чтобы, имея множество плохих (незначительно отличающихся от случайных) алгоритмов классификации, получить один хороший. В основе такой идеи лежит построение «комитета» классификаторов [53, 54], каждый из которых (кроме первого) обучается на ошибках предыдущего с использованием ансамблевого или каскадного подхода. Объединяя классификаторы в «комитет», алгоритм усиливает их эффективность. Бустинг чувствителен к шуму в данных и выбросам. Однако он менее подвержен переобучению по сравнению с другими алгоритмами машинного обучения. Упрощённая иллюстрация работы алгоритма бустинга приведена на рис. 15.

Например, один из первых алгоритмов бустинга Boost1 использовал ансамбль из трёх моделей, первая из которых обучалась на всём наборе данных, вторая — на выборке примеров, в половине которых первая дала правильные ответы, а третья — на примерах, где ответы первых двух разошлись. Таким образом, имеет место последовательная обработка примеров ансамблем классификаторов, причём так, что задача для каждого последующего становится труднее. Результат определяется путём простого голосования: пример относится к тому классу, который выдан большинством моделей.

Каскадная модель сильных классификаторов — это, по сути, то же дерево принятия решений, где каждый узел дерева построен таким образом, чтобы обнаруживать почти все интересующие объекты и отклонять регионы, их не содержащие. Помимо этого, узлы дерева размещены следующим образом: чем ближе узел находится к корню дерева, тем более простые признаки он использует и тем самым требует меньшего времени на принятие решения. Данный вид каскадной модели хорошо подходит для обработки изображений, на которых общее количество обнаруживаемых образов мало.

Сначала данные анализируются первым «простым» классификатором. Его положительное решение запускает второй, несколько более приспособленный к решаемой задаче, и т. д. Отрицательное решение классификатора на любом этапе приводит к немедленному завершению процедуры классификации. Классификатор на каждом шаге становится более сложным, поэтому ошибок каскада становится меньше.

Для достижения баланса между излишней и недостаточной гибкостью в единой мо-

дели может быть собрано множество деревьев. Это будет классификатор на основе ансамбля каскадов классификаторов (или «комитета» деревьев принятия решений, «леса»). При этом каждое дерево как бы «голосует» за принадлежность объекта к определённому классу. Таким образом, на основе того, какая часть деревьев «проголосовала» за тот или иной класс, можно заключить, с какой вероятностью объект принадлежит к каждому из них [55].

Следует также отметить, что при обнаружении объектов в поле наблюдения необходимо учитывать, что они могут находиться на регистрируемых изображениях в разных местах. По этой причине для поиска объекта в кадре окно поиска (данные из которого подаются на вход классификатора) необходимо перемещать по всему изображению с некоторым перекрытием. При этом для достижения наилучшего результата необходимо осуществить полный перебор окон поиска не только по их положению, но и по размеру, поскольку объекты могут быть разных масштабов. Это делает подобный перебор неэффективным, занимающим очень большое количество времени.

Для решения данной проблемы параллельно развиваются две группы методов: двухэтапные и одноэтапные. Двухэтапные методы R-CNN [56], Fast R-CNN [57], Faster R-CNN [58] и т. п. (они же методы, основанные на регионах интереса) включают два этапа. На первом этапе селективным поиском или с помощью специальной нейронной сети выделяются области, с высокой вероятностью содержащие внутри себя объекты, которые на втором этапе рассматриваются классификатором для определения принадлежности к исходным классам, а также уточнения их местоположения и размеров. Одноэтапные методы YOLO [59–61], SSD [62] и т. п. не используют отдельный алгоритм для генерации регионов интереса. Вместо этого они одновременно за один проход формируют определённое количество ограничивающих рамок с различными параметрами и предсказывают класс объекта.

Заключение. Выполнен анализ и оценка возможности применения нейросетевых технологий для обнаружения объектов по последовательностям их изображений, регистрируемым оптико-электронными системами наблюдения различного назначения.

Нейросетевые методы, как и их развитие, комплексные методы, к которым приковано большое внимание исследований в последнее десятилетие, несомненно обладают большим потенциалом. Здесь значительный прогресс достигнут в первую очередь за счёт идеи глубокого обучения и создания огромных баз изображений, используемых для настройки параметров моделей. Это сделало возможным при обучении созданных моделей реализацию оценки функции распределения плотности вероятности для каждого класса объектов и формирование сколь угодно сложных разделяющих поверхностей в пространстве признаков. При этом алгоритмы, основанные на свёрточных нейросетевых технологиях, позволили полностью отказаться от наиболее плохо формализованного этапа классификации изображений — этапа эмпирического выбора системы признаков.

Значительные успехи в области анализа изображений продемонстрировали комбинированные методы, основанные на применении свёрточных многослойных нейронных сетей для извлечения признаков и ансамблевых классификаторов. Существенный прогресс обеспечила идея бустинга — сочетания ряда классификаторов для усиления их эффективности. При этом важную роль начинает играть тактика обучения последовательных и параллельных классификаторов. Отметим, что идея, связанная с автоматическим изменением структуры сети, несомненно перспективная, но недостаточно апробированная, чтобы делать окончательные выводы об её преимуществах и недостатках для практического применения.

Несмотря на бесспорные успехи в развитии нейронных сетей, достигнутые в последние десятилетия, они по-прежнему имеют ряд недостатков, главные из которых — длительное обучение и настройка, а также необходимость наличия большого объёма данных для обучения. Ещё один недостаток нейронных сетей — это то, что в случае их переобучения

новые виды объектов не могут быть классифицированы, если в обучающей выборке не было похожих размеченных объектов. Предложен ряд приёмов (довольно успешных) для борьбы с этим явлением, однако окончательно проблема не решена. Поэтому говорить об однозначном преимуществе нейронных сетей с глубоким обучением при решении практических задач обнаружения объектов пока преждевременно.

Финансирование. Работа выполнена при поддержке Министерства науки и высшего образования в рамках выполнения работ по Государственному заданию № 121022000116-0 в ИАиЭ СО РАН.

СПИСОК ЛИТЕРАТУРЫ

1. **Методы** компьютерной обработки изображений /Под ред. В. А. Сойфера. М.: Физматлит, 2003. 784 с.
2. **Лукьяница А. А., Шишкин А. Г.** Цифровая обработка видеоизображений. М.: «Ай-Эс-Эс Пресс», 2009. 518 с.
3. **Форсайт Д., Понс Ж.** Компьютерное зрение. Современный подход = Computer Vision: A Modern Approach. М.: «Вильямс», 2004. 928 с.
4. **Шапиро Л., Стокман Дж.** Компьютерное зрение = Computer Vision. М.: Бином. Лаборатория знаний, 2006. 752 с.
5. **Грузман И. С., Киричук В. С., Косых В. П. и др.** Цифровая обработка изображений в информационных системах. Новосибирск: НГТУ, 2002. 352 с.
6. **Журавлёв Ю. И., Рязанов В. В., Сенько О. В.** Распознавание. Математические методы. Программная система. Практические применения. М.: ФАЗИС, 2006. 147 с.
7. **Viola P.** Rapid object detection using a boosted cascade of simple features // Proc. of the Accepted Conf. on Computer Vision and Pattern Recognition (CVPR 2001). Kanai, Hawaii, Usa, 8–14 Dec., 2001. P. 511–518.
8. **Dalal N., Triggs B.** Histograms of oriented gradients for human detection // Proc. of the IEEE Comput. Soc. Conf.: Computer Vision and Pattern Recognition. San Diego, USA, 20–25 June, 2005. Vol. 1. P. 886–893.
9. **Deng J., Dong W., Socher R. et al.** ImageNet: A Large-Scale Hierarchical Image Database // Proc. of the Conf.: IEEE Computer Vision and Pattern Recognition (CVPR). Miami, USA, 20–25 June, 2009. P. 248–255.
10. **Russakovsky O., Deng J., Su H. et al.** (* = equal contribution) ImageNet large scale visual recognition challenge // Int. Journ. Computer Vision. 2015. **115**. P. 211–252.
11. **Everingham M., Eslami S. M. A., Van Gool L. et al.** The PASCAL visual object classes challenge: A retrospective // Int. Journ. Comput. Vis. 2015. **111**. P. 98–136. DOI: 10.1007/s11263-014-0733-5.
12. **Mann S.** «GlassEyes»: The Theory of EyeTap Digital Eye Glass // IEEE Technology and Society. 2012. **31**, N 3. P. 10–14. URL: http://wearcam.org/glass.pdf (дата обращения: 17.01.2022).
13. **Development** Edition. Официальный сайт Microsoft. 2016. URL: https://www.microsoft.com/microsoft-hololens/en-us (дата обращения: 19.11.2021).
14. **Meet** Kinect for Windows. Официальный сайт Microsoft. 2016. URL: https://dev.windows.com/en-us/kinect (дата обращения: 19.11.2021).
15. **PyTorch.** Официальный сайт PyTorch. URL: https://pytorch.org/ (дата обращения: 14.01.2022).
16. **Vuforia** 5.5 SDK. Vuforia Developer Portal. 2016. URL: https://developer.vuforia.com/downloads/sdk (дата обращения: 12.11.2021).

17. **Kudan** SDK 1.2.3 version. Kudan Augmented Reality. 2016. URL: https://www.kudan.eu/download/ (дата обращения: 17.11.2021).
18. **Kenneth D.-H.** A Practical Introduction to Computer Vision with OpenCV. Ireland: Trinity College Dublin, 2014. 234 p.
19. **Zou Z., Shi Z., Guo Y., Ye J.** Object Detection in 20 Years: A Survey. URL: arXiv preprint arXiv: 1905.05055м2. 2019.
20. **Андриянов Н. А., Дементьев В. Е., Ташлинский А. Г.** Обнаружение объектов на изображении: от критериев Байеса и Неймана–Пирсона к детекторам на базе нейронных сетей EfficientDet // Компьютерная оптика. 2022. **46**, № 1. С. 139–159.
21. **Tensor** Flow 2 detection zoo. URL: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md (дата обращения: 27.01.2023).
22. **The Neural** Network Zoo. URL: https://www.asimovinstitute.org/neural-network-zoo/ (дата обращения: 27.01.2023).
23. **Уоссермен Ф.** Нейрокомпьютерная техника: теория и практика. М.: Мир, 1992. 118 с.
24. **Области** применения нейронных сетей. Классификация нейронных сетей - Обзор и анализ нейросетей (studbooks.net). URL: https://studbooks.net/2030598/informatika/oblasti_primeneniya_neyronnyh_setey (дата обращения: 27.01.2023).
25. **Николаева С. Г.** Нейронные сети. Реализация в Matlab: Учеб. пособие. Казань: Казан. гос. энерг. ун-т, 2015. 92 с.
26. **Галушкин А. И.** Синтез многослойных систем распознавания образов. М.: «Энергия», 1974. 366 с.
27. **Werbos P. J.** Beyond regression: New tools for prediction and analysis in the behavioral sciences: Ph.D. thesis. Cambridge: Harvard University, 1974. 453 p.
28. **Rumelhart D. E., Hinton G. E., Williams R. J.** Learning Internal Representations by Error Propagation. In: Parallel Distributed Processing. Vol. 1. Cambridge: MA, MIT Press, 1986. P. 318–362.
29. **CIFAR-10** and CIFAR-100 datasets (toronto.edu). URL: https://www.cs.toronto.edu/k̃riz/cifar.html (дата обращения: 27.01.2023).
30. **Вежневец А. П.** Методы классификации с обучением по прецедентам в задаче распознавания объектов на изображениях // Лаборатория Компьютерной Графики и Мультимедиа факультета ВМиК, Московский государственный университет им. М. В. Ломоносова. М., 2006. URL: http://www.graphicon.ru/2006/fr10_34_VezhnevetsA.pdf (дата обращения: 27.01.2023).
31. **Badrinarayanan V., Kendall A., Cipolla R.** SegNet: A deep convolutional encoder-decoder architecture for image segmentation // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017. **39**, N 12. P. 2481–2495.
32. **Cicek O., Abdulkadir A., Lienkamp S. S. et al.** 3D U-Net: Learning dense volumetric segmentation from sparse annotation // Medical Image Computing and Computer-Assisted Intervention (MICCAI 2016). Lecture Notes in Computer Science. Springer, Cham, 2016. Vol. 9901. P. 424–432.
33. **Козик В. И., Нежевенко Е. С.** Классификация гиперспектральных изображений с помощью свёрточных нейронных сетей // Автометрия. 2021. **57**, № 2. С. 13–21. DOI: 10.15372/AUT20210202.
34. **Lecun Y., Bottou L., Bengio Y., Haffner P.** Gradient-based learning applied to document recognition // Proceedings of the IEEE. 1998. **86**, N 11. P. 2278–2324. DOI: 10.1109/5.726791.
35. **Krizhevsky A., Sutskever I., Hinton G. E.** ImageNet classification with deep convolutional neural networks // Advances in Neural Information Processing Systems (NIPS). 2012. P. 1097–1105.

36. **Hinton G. E., Srivastava N., Krizhevsky A. et al.** Improving neural networks by preventing co-adaptation of feature detectors, arXiv: 1207.0580 [cs.NE]. 2012.
37. **Борзов С. М., Карпов А. В., Потатуркин О. И., Хадзиев А. О.** Применение нейронных сетей для дифференциальной диагностики лёгочных патологий по рентгенологическим изображениям // Автометрия. 2022. **58**, № 3. С. 61–71. DOI: 10.15372/AUT20220307.
38. **Карпов А. В., Козик В. И., Нежевенко Е. С., Шварц Я. Ш.** О влиянии качества баз данных рентгеновских снимков туберкулёзных больных на диагностику болезни // Автометрия. 2022. **58**, № 5. С. 67–74. DOI: 10.15372/AUT20220508.
39. **Simonyan K., Zisserman A.** Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
40. **VGG16** — свёрточная сеть для выделения признаков изображений (neurohive.io). Sours. URL: https://neurohive.io/ru/vidy-nejrosetej/vgg16-model/? (дата обращения: 27.01.2023).
41. **Lin M., Chen Q., Yan S.** Network In Network. arXiv preprint arXiv:1312.4400, 2014.
42. **He K., Zhang X., Ren S., Sun J.** Deep Residual Learning for Image Recognition. arXiv:1512.03385v1 [cs.CV] 10 Dec 2015.
43. **Szegedy C., Liu W., Jia Y. et al.** Going deeper with convolutions // Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. Boston, USA, 7–12 June, 2015. P. 1–9.
44. **Veit A., Wilber M., Belongie S.** Residual Networks are Exponential Ensembles of Relatively Shallow Networks. In arXiv:1605.06431. 2016.
45. **Huang G., Liu Zh., van der Maaten L., Weinberger K. Q.** Densely Connected Convolutional Networks. arXiv:1608.06993v5. 2018.
46. **Tan M., Le Q. V.** EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Machine Learning (cs.LG); Computer Vision and Pattern Recognition (cs.CV); Machine Learning (stat.ML) arXiv:1905.11946. 2020.
47. **Tan M., Pang R., Le Q. V.** EfficientDet: Scalable and Efficient Object Detection Computer Vision and Pattern Recognition (cs.CV); Machine Learning (cs.LG); Image and Video Processing (eess.IV) arXiv:1911.09070v7. 2020.
48. **Ramachandran P., Zoph B., Le Q. V.** Searching for activation functions // Proc. of the Int. Conf. on Learning Representations (ICLR Workshop). Vancouver, Canada, 30 April – 3 May, 2018. URL: https://openreview.net/forum?id=SkBYYyZRZ (дата обращения: 27.01.2023).
49. **Chen Y., Yang T., Zhang X. et al.** DetNAS: Backbone Search for Object Detection. arXiv:1903.10979v4. 2019.
50. **Lin T. Y., Maire M., Belongie S. et al.** (2014). Microsoft coco: Common objects in context. arXiv preprint arXiv:1405.0312v3, 2014.
51. **Yao L., Xu H., Zhang W. et al.** SM-NAS: Structural-to-Modular Neural Architecture Search for Object Detection. arXiv:1911.09929v2. 2019.
52. **Ногин В. Д.** Множество и принцип Парето. СПб: Издательско-полиграфическая ассоциация высших учебных заведений, 2022. 111 с.
53. **Freund Y., Schapire R. E.** A Short Introduction to Boosting. Shannon Laboratory, 1999. P. 771–780
54. **Šochman J., Matas J.**, AdaBoost. Center for Machine Perception. Prague: Czech Technical University, 2010. URL: https://cmp.felk.cvut.cz/∼sochmj1/adaboost_talk.pdf (дата обращения: 27.01.2023).
55. **Utkin L. V., Ryabinin M. A.** A siamese deep forest // Knowledge-Based Systems. 2018. **139**. P. 13–22.

56. **Girshick R., Donahue J., Darrell T., Malik J.** Rich feature hierarchies for accurate object detection and semantic segmentation // Proc. of the IEEE Int. Conf. on Comp. Vis. and Pattern Recogn. Columbus, USA, 24-27 June, 2014. P. 580–587.
57. **Girshick R.** Fast RCNN // Proc. of the IEEE Int. Conf. on Computer Vision. Santiago, Chile, 11-18 Dec., 2015. P. 1440–1448.
58. **Ren S., He K., Girshick R., Sun J.** Faster RCNN towards real-time object detection with region proposal networks // Proc. of the IEEE Int. Conf. on Advances in Neural Information Processing Systems. Montreal, Canada, 7-12 Dec., 2015. P. 91–99.
59. **Redmon J., Divvala S., Girshick R., Farhadi A.** You only look once: Unified, real-time object detection // Proc. of the IEEE Confe. on Comp. Vis. and Pattern Recogn. Las Vegas, USA, 26 June – 1 July, 2016. P. 779–788.
60. **Redmon J., Farhadi A.** YOLOv3: An incremental improvement // Computer Vision and Pattern Recognition. Cornell Univers., 2018. P. 779–788. URL: arXiv preprint arXiv: 804.02767. 2018.
61. **Bochkovskiy A., Wang C.-Y., Liao H.-Y. M.** YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934v1, 2020.
62. **Liu W., Anguelov D., Erhan D. et al.** SSD: Single Shot MultiBox Detector // arXiv:1512.02325, 2016. DOI: 10.1007/978-3-319-46448-0_2.

Поступила в редакцию 27.01.2023
После доработки 02.02.2023
Принята к публикации 17.02.2023