УДК 608.2

ТЕХНОЛОГИИ ГЛУБОКОЙ НЕЙРОННОЙ СЕТИ МНОГОМАСШТАБНОГО ДЕТЕКТИРОВАНИЯ ЛИЦ

А.В. Суша
Аспирант кафедры ЭВС, БГУИР

М.И. Вашкевич
Доцент кафедры ЭВС БГУИР, к.т.н., доцент

Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь
E-mail: isushik94@bsuir.by, vashkevich@bsuir.by

А.В. Суша
Окончил Белорусский государственный университет информатики и радиоэлектроники. Аспирант кафедры ЭВС, БГУИР, магистр технических наук.

М.И. Вашкевич
Окончил Белорусский государственный университет информатики и радиоэлектроники (БГУИР) по специальности «Электронные вычислительные средства» в 2008 г. В 2013 г. защитил кандидатскую диссертацию по специальности 05.13.05 «Элементы и устройства вычислительной техники и систем управления». Работает доцентом кафедры ЭВС в БГУИР. Научные интересы: обработка сигналов, методы машинного обучения.

Аннотация. Целью настоящей работы являлось проектирование глубокой искусственной нейронной сети для детектирования лиц. Основное внимание при проектировании было уделено обеспечению высокой производительности и уменьшению требуемых вычислительных затрат за счет: 1) факторизации операции свертки; 2) применения точечных сверток; 3) комбинирования поканальных и точечных сверток. Разработанный детектор сравнивался со схожими детекторами лиц, полученными на основе широко распространенных архитектур нейронных сетей MobileNet и NasNet. Предложенная архитектура детектора лиц имеет вычислительную сложность 5.1 MFLOPs, что в два раза меньше, чем у MobileNet (11.7 MFLOPs) и в четыре раза меньше, чем у NasNet (22 MFLOPs). Соответственно время детектирования на изображении 416×416 составило 5.12 мс (или 195 FPS) с видеокарты GeForce 1080 Ti, а также 65.4 мс (или 15 FPS) на одном ядре процессора Intel Core i7-8700K. При этом точность нашей архитектуры равна 85% и уступает MobileNet лишь на 4%, а NasNet – на 9.5%.

Ключевые слова: детектирование лиц, глубокие нейронные сети, сверточные нейронные сети

Введение. В настоящее время разработано множество способов детектирования лиц, большинство из которых основано на *парадигме скользящего окна* и применении классических методов компьютерного зрения и машинного обучения (например, метод Виолы-Джонса на базе каскадов Хаара и HOG+SVM). Преимуществом этих подходов является высокая скорость детектирования, что позволяет их использовать в режиме реального времени. Существенным недостатком данных методов является большое число ложных срабатываний и слабая устойчивость к множеству факторов, затрудняющих детектирование: разный масштаб лиц, частичное перекрытие лиц другими объектами, ракурс, выражения лиц, засветка и прочее [1]. Применение глубоких искусственных нейронных сетей (далее – ИНС) для детектирования лиц на изображении позволяет повысить точность и достичь большую

инвариантность к указанным факторам. Однако, подход, использующий скользящее окно в связке с ИНС, имеет большую вычислительную сложность в сравнении с классическими методами. Для уменьшения сложности нейросетевых детекторов лиц и, в общем случае, объектов, были предложены новые архитектуры ИНС.

На смену парадигме скользящего окна пришла *парадигма двухэтапного детектирования*. Известным примером реализации этой парадигмы является детектор объектов R-CNN [2], который впоследствии был несколько раз модифицирован для повышения точности и скорости детектирования. Суть этого метода заключается в выделении на первом этапе т. н. *регионов интереса* (сегментация изображения на регионы объекта или фона), которые сокращают число кандидатов для следующего этапа классификации регионов. Такой подход позволил понизить вычислительную сложность детектора, так как сократилось число запусков «тяжелого» классификатора.

Следующей парадигмой является *парадигма одноэтапного детектирования*, которая возникла в стремлении ускорить детектирование объектов посредством ИНС. В некотором смысле она является переосмыслением парадигмы скользящего окна со стороны ИНС. Детектирование осуществляется за один вычислительный проход ИНС, в котором осуществляется одновременная классификация всех ячеек (окон) равного размера, на которые разбивается изображение. Реализацией этой парадигмы явился метод YOLO (You Only Look Once) [3, 4], преимущество которого заключается в том, что изображение анализируется сверточной ИНС только один раз. Однако у такого подхода имеется один существенный недостаток: он проводит детектирование объектов одного масштаба. Алгоритм многомасштабного детектирования предложен в работе SSD (Single Shot multibox Detector) [5]. Он был усовершенствован в методе FPN (Feature Pyramid Network) [6] путем передачи признаков одного масштаба на другой для осуществления классификации.

Целью настоящей работы является проектирование архитектуры ИНС для многомасштабного детектирования лиц методами, позволяющими снизить требования к памяти и вычислительным ресурсам. При этом уменьшение точности детектирования должно быть пропорционально меньшим, чем уменьшение параметров и операций в ИНС.

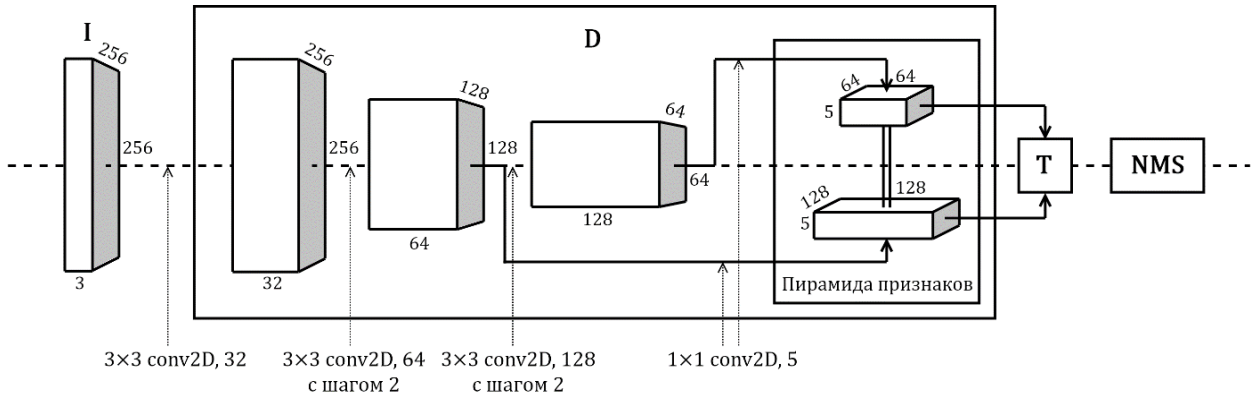

Рисунок *1*. – Структура детектора лиц

Описание архитектуры детектора лиц. На рисунке 1 показана обобщенная структура детектора лиц, реализующего парадигму одноэтапного детектирования. Изображение *I* подается на сверточную ИНС *D*, последний слой которой генерирует карту детектируемых лиц. Сама ИНС построена по принципу сети с пирамидой признаков [6]. Пирамида признаков позволяет проводить детектирование лиц разного размера за один проход ИНС путем генерирования карт разного размера. После вычисления всех карт детектируемых лиц осуществляется формирование списка рамок, найденных на изображении лиц (операция *T* на

рисунке 1). Затем для этого списка применяется алгоритм подавления немаксимумов (NMS – non maximum suppression) [7], который устраняет дубликаты и схожие рамки из списка.

В общем виде процесс детектирования можно записать следующими выражениями:

$$\left\{\left[B_{i_r,j_r}\right]_r\right\} = D(I), \quad i_r = 1,..,N_r, \quad j_r = 1,..M_r, \quad r = \{1,2,3\},$$
$$\left\{B_{r,v_r,q_r}\right\} = T\left(\left\{\left[B_{i_r,j_r}\right]_r\right\}\right), \quad v_r \in 1,..,N_r, \quad q_r \in 1,..M_r,$$
$$\{B_k\} = NMS\left(\left\{B_{r,v_r,q_r}\right\}\right), \quad k = 1,\dots,\left|\{B_{r,i_r,i_r}\}\right|,$$

где I – входное изображение; $D(\cdot)$– оператор, выполняющий детектирование лиц; r – номер карты детектируемых лиц; N_r и M_r – число строк и столбцов карты детектируемых лиц под номером r; B_{i_r,j_r} – значение карты детектируемых лиц под номером r в i_r-й строке и j_r-м столбце; $T(\cdot)$– операция преобразования карты детектируемых лиц в список рамок лиц, при этом осуществляется фильтрация рамок на основе вероятности обнаружения лица; B_{r,v_r,q_r} – элемент списка найденных лиц, соответствующий в v_r-й строке и q_r-м столбце карты под номером r; B_k – элемент списка оставшихся лиц после применения *NMS*, соответствующий в k-му элементу из списка найденных лиц.

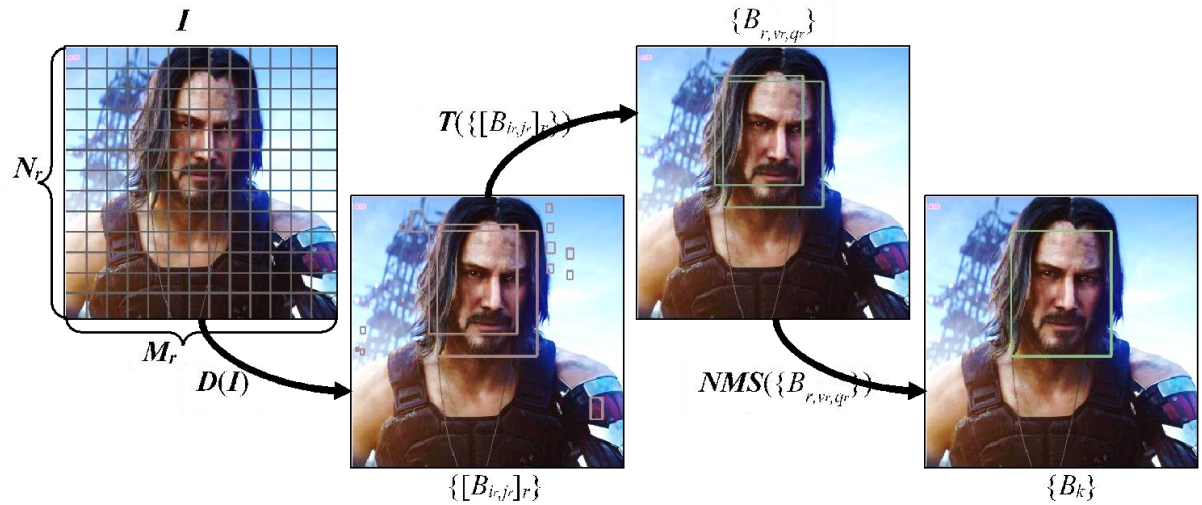

Рисунок *2*. – Пример процесса детектирования лица

Процесс детектирования лица на примере поэтапной обработки одного изображения представлен на рисунке 2. На первом этапе ИНС генерирует карту размером $N_r \times M_r$, где $r \in \{1,2,3\}$, элементы которой отвечают за некоторый небольшой участок на входном изображении. Если разбить входное изображение на ячейки, формирующие сетку $N_r \times M_r$, то каждая ячейка будет соответствовать одному элементу карты. Элементом B_{i_r,j_r} в i_r-й строке и j_r-м столбце карты под номером r является 5-мерный вектор, который имеет следующий формат:

$$B_{i_r,j_r} = [p,x,y,w,h]^T, \quad (1)$$

где p – вероятность обнаружения лица, $p = \sigma(l_p)$, где l_p – 1-й линейный выход сети в ячейке, $\sigma(\cdot)$– функция логистического сигмоида, x, y – относительное смещение центра рамки лица в границах ячейки, при этом: $x = \sigma(l_x)$ и $y = \sigma(l_y)$, где l_x и l_y – 2-й и 3-й линейные выходы сети в ячейке; w, h – нормированная ширина и высота рамки относительно размера

ячейки, при этом: $w = exp(l_w)$ и $h = exp(l_h)$, где l_w и l_h – 4-й и 5-й линейные выходы сети в ячейке.

Оптимизация архитектуры ИНС. Анализ научных статей позволил выделить следующие методы повышения производительности ИНС, суть которых заключается в уменьшении числа выполняемых машинных операций в различных типовых блоках:

–первую операцию свертки необходимо производить с некоторым шагом [8], например шаг окна равный 2, что в 4 раза уменьшает число операций, проводимых в этом слое;

–регулирование числа каналов признаков операцией точечной свертки [9];

–операции свертки можно факторизовать на несколько последовательных операций с меньшим числом параметров [10]. К примеру, свертка с окном 3×3 выполняется двумя свертками 3×1 и 1×3, что уменьшает число параметров сети, а также число операций умножений с накоплением;

–традиционная операция свертки может быть заменены на последовательные операции поканальной и точечной свертки [11], что уменьшает число параметров с $n = c \times h \times w \times k$ на $m = c \times h \times w + c \times k$, где c – число входных каналов, k – число выходных каналов, $h \times w$ – размер ядра свертки.

Пусть W – тензор параметров операции свертки; x – некоторый трехмерный тензор; i и j – индексы строки и столбца тензора соответственно; m и n – индексы канала входного тензора и канала выходного тензора соответственно; s_r и s_c – шаги выборки элемента тензора по строкам и по столбцам; K и L– высота и ширина окна свертки; M – число окон (каналов). Тогда математически операции свертки можно записать следующим образом:

$$conv2d(x, W, s_r, s_c) = [y_{i,j,n}] = \left[\sum_{k,l,m}^{K,L,M} x_{\left(s_r \cdot i + k - \frac{K}{2}, s_c \cdot j + l - \frac{L}{2}, m\right)} \cdot W_{k,l,m,n}\right],$$

$$dwConv2d(x, W, s_r, s_c) = [y_{i,j,m}] = \left[\sum_{k,l}^{K,L} x_{\left(s_r \cdot i + k - \frac{K}{2}, s_c \cdot j + l - \frac{L}{2}, m\right)} \cdot W_{k,l,m}\right],$$

$$pwConv2d(x, W, s_r, s_c) = [y_{i,j,n}] = \left[\sum_{m}^{M} x_{(s_r \cdot i, s_c \cdot j, m)} \cdot W_{m,n}\right],$$

где *conv2d* – операция свертки, *dwConv2d* – поканальная операция свертки и *pwConv2d* – точечная операция свертки.

Для симулирования поведения обычной свертки с помощью поканальных сверток применяется замыкающая точечная свертка, что математически записывается следующим образом:

$$sepConv2d(x, W_{dw}, W_{pw}, s_r, s_c) = pwConv2d(dwConv2d(x, W_{dw}, s_r, s_c), W_{pw}, 1, 1),$$

где W_{dw} – тензор параметров поканальной свертки и W_{pw} – тензор параметров точечной свертки.

В работе, описывающей VGG-16 [12] показано, что использование нескольких последовательных сверток 3×3 (например, две или три) имеют то же рецептивное поле как и одна свертка 5×5 или 7×7. При этом такая последовательность имеет меньшее число параметров, что уменьшает требования к памяти и число вычислительных операций. Более того, применение нелинейной функции активации после каждой свертки в последовательности позволяет улучшить аппроксимационные свойства ИНС, так как в работе

[13] было показано, что многослойная ИНС с нелинейными функциями активации является универсальным аппроксиматором.

При проектировании архитектуры ИНС для детектирования лиц было решено провести факторизацию свертки 3×3 на две свертки 3×1 и 1×3. Таким образом уменьшилось затраты памяти на для хранения параметров сети на треть. Дополнительно после каждой новой свертки применяется функция активации $ReLU(x) = max(x, 0)$ [14]. Выбор этой функции активации обусловлен тем, что она требует выполнения только одной операции: сравнения с нулем – которая сочетает в себе простоту линейной функции, а также необходимое свойство нелинейности. Также, помимо высокой скорости вычислений при прямом проходе, эта функция активации имеет простую функцию производной первого порядка, что позволяет обучать ИНС быстрее.

Описание архитектуры многомасштабного детектора лиц. Для простоты описания предлагаемой архитектуры введем понятие блока сверток. Блок сверток –последовательные операции свертки, которые являются факторизацией некоторой одной операции свертки с тем же рецептивным полем. На рисунке 3 показаны два варианта блока сверток (*block* и *sepBlock*), применяемые в предлагаемой архитектуре сверточной ИНС. Здесь и далее операции свертки записываются в следующем формате: размер окна и/или число выходных каналов, название операции и в скобках аргументы операции, входной тензор и тензор параметров опускаются.

Следующей структурной единицей является операция *bottleneck* (дословно – «бутылочное горлышко») [15]. Суть операции заключается в преднамеренном понижении или повышении числа каналов карт активаций внутри. Эта операция включает в себя блок сверток, определенный ранее. Соответственно применяется два варианта этой операции: *block bottleneck* и *sepBlock bottleneck*.

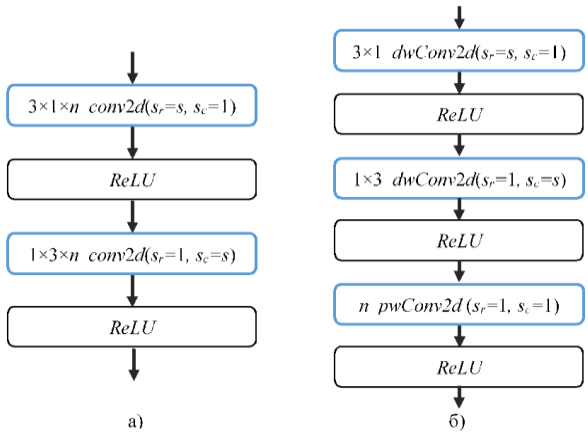

Рисунок *3*. – Схема применяемых блоков: а) – *block*, б) – *sepBlock*;
где *n* – число выходных каналов; *s* – шаг выборки окна свертки

Рисунок 4 содержит схему операции *bottleneck* в общем виде, где *batch normalization* – операция нормализации по обучающему мини-пакету [16]. Также применяется остаточная связь (англ. – residual connection) [15] для случая, когда ширина и высота тензора на входе блока и на выходе одинаковы. При разном числе каналов применяется операция точечной свертки для выравнивания числа каналов входного тензора.

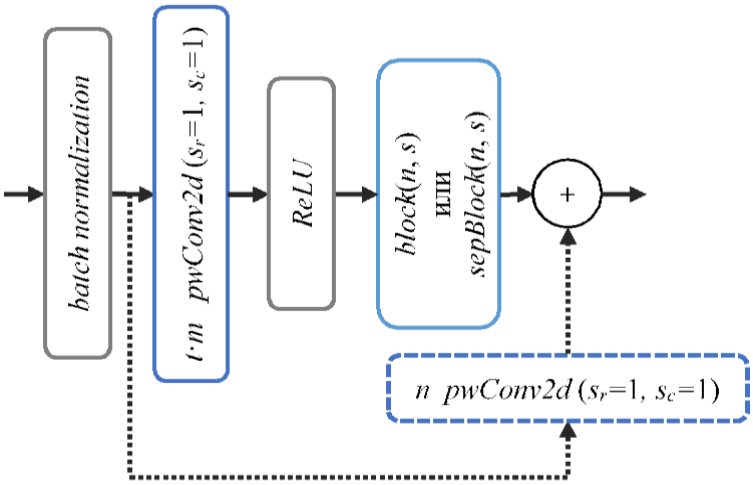

Рисунок 4. – Схема операции *bottleneck*; где m – число входных каналов; n – число выходных каналов; t – коэффициент расширения/сжатия числа каналов

На рисунке 5 приведена архитектура спроектированной сверточной ИНС. Выражением вида «×*N*» обозначается число повторений операции в блоке. Также на схеме имеется дополнительная операция *upsample* $m \times n$, которая выполняет пространственное увеличение размера карты признаков методом ближайшего соседа, где m – число повторов элемента карты по вертикали, а n – число повторов элемента карты по горизонтали.

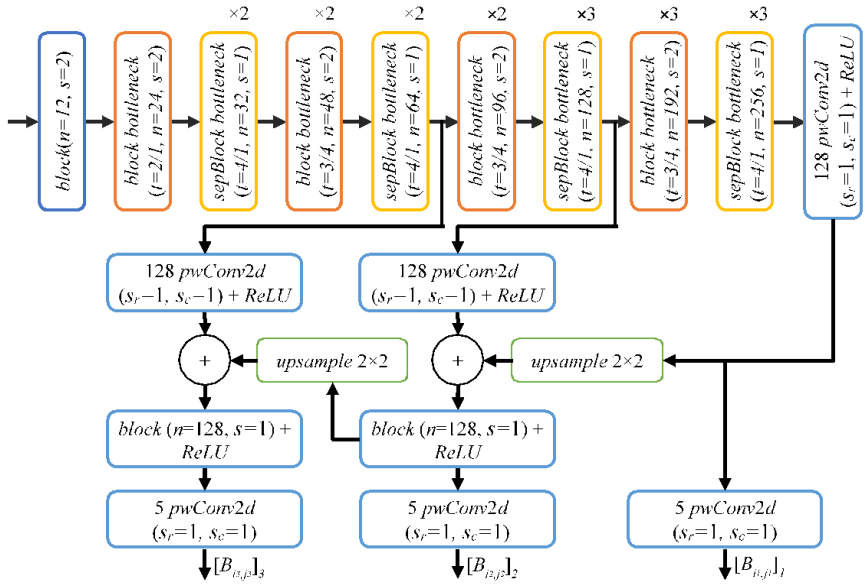

Рисунок 5. – Архитектура ИНС

Процесс обучения. Для обучения использовалось несколько функций ошибки для каждого из элементов вектора $B_{i_r j_r}$ в выражении (1). Для обучения выхода, предсказывающего вероятность обнаружения лица использовалась средняя взвешенная бинарная перекрестная энтропия:

$$L_p = -\frac{1}{n}\sum_{i=1}^{n} \quad k_p \cdot p_i \cdot log\ log\ (\hat{p}_i) + k_n \cdot (1 - p_i) \cdot log\ log\ (1 - \hat{p}_i),$$

где n – число обучающих примеров, p_i – целевое значение вероятности обнаружения объекта i-го примера, $\hat{p}_i$ – предсказанное значение вероятности обнаружения объекта i-го примера, k_p и k_n – весовые коэффициенты для положительных и отрицательных примеров наличия объекта, которые определяются по формулам: $k_p = \frac{n_{max}}{n_p}$ и $k_n = \frac{n_{max}}{n_n}$, где n_p – число положительных примеров, n_n – число отрицательных примеров и $n_{max} = max(n_p, n_n)$.

Функцией ошибки для выходов, предсказывающих параметры рамки лица, является сумма средних квадратических ошибок координат центра рамки и сумма средних квадратических ошибок логарифмов ширины и высоты рамки:

$$L_{xy} = \frac{1}{n}\sum_{i=1}^{n} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2,$$

$$L_{wh} = \frac{1}{n}\sum_{i=1}^{n} (log\,log\,(w_i) - log\,log\,(\hat{w}_i))^2 + (log\,log\,(h_i) - log\,log\,(\hat{h}_i))^2,$$

где x_i, y_i, w_i, h_i – целевые значения смещения центра и размера рамки объекта i-го примера; $\hat{x}_i$, $\hat{y}_i$, $\hat{w}_i$, $\hat{h}_i$ – соответствующие предсказанные параметры для объекта i-го примера.

Для вычисления функций ошибки параметров рамки лица используются только те примеры, для которых целевое значение вероятности обнаружения объекта равно 1. В обучении применяется линейная комбинация приведенных функций ошибок, которая записывается следующим образом:

$$L = a_p \cdot L_p + a_{xy} \cdot L_{xy} + a_{wh} \cdot L_{wh}, \quad (2)$$

где a_p, a_{xy}, a_{wh} – весовые коэффициенты.

Функция ошибки (2) минимизировалось методом стохастической оптимизации Adam [17]. При обучении применялись следующие значения коэффициентов: a_p = 1, a_{xy} = 10, a_{wh} = 5. Скорость обучения равнялась 0.001, и уменьшался линейно в 1,3 раз каждые 50 000 итераций. Обучение длилось 200 000 итераций.

Результаты экспериментов. Обучение детектора проводилось на базе данных WIDER [18]. Эта база содержит 393 703 размеченных лицам на 32 203 изображениях. Изображения в базе разбиты на 61 группу в зависимости от масштаба, позы, перекрытия и освещения и проч. На рисунке 6 показаны образцы изображений лиц из этой базы.

Рисунок 6. – Пример групп изображений из базы WIDER

Помимо ИНС изображенной на рисунке 5 были обучены широко известные архитектуры: NasNet [19] и MobileNet V2 [20]. Эти архитектуры на сегодняшний день

являются одними из самых экономных в вычислительном плане, что позволяет использовать их для сравнения.

Рисунок 7. – Пример изображений из базы FDDB с нанесенной разметкой

Тестирование детекторов лиц проводилось на базе FDDB [21], содержащей 2845 изображений с 5171 вручную размеченными лицами. Пример изображений из базы приведен на рисунке 7. Для проведения тестирования изображения из этой базы масштабировались таким же образом, как и для обучения. В качестве метрик использовались точность, полнота и F1-мера [22]. Пороговое значение вероятности детектирования – 0.5, пороговое значение *IoU* (англ. – intersection over union) для алгоритма подавления не максимумов – 0.4 и для фиксирования правильно сдетектированного лица – 0.4. Для сравнения вычислительной сложности оценивалось количество выполняемых операций с плавающей точкой каждой из тестируемых ИНС – FLOPs (float point operations). Результат тестирования приведен в таблице 1.

Таблица 1. – Результаты тестирования детекторов лиц на базе FDDB

Детектор	Размер изображения	FLOPs	Точность	Полнота	F1-мера	GPU, мс
NasNet	416×416	22 044 117	94.58	94.88	94.73	10.04±0.14
MobileNet V2	416×416	11 706 405	89.16	96.09	92.49	6.73±1.03
Предлагаемый детектор	416×416	5 155 823	84.93	96.70	90.44	5.12±1.02

Измерение времени прямого прохода (колонка GPU в таблице 1) проводилось путем сбора статистики о времени работы детектора на одиночных изображениях базы из FDDB. Изображения приводились к размеру 416×416. За один проход сети осуществлялось детектирование всех лиц на изображении. По собранным данным о времени работы детектора производился расчет математического ожидания и среднеквадратического отклонения, которые и представлены в таблице. Для замера времени использовалась видеокарта Nvidia GeForce 1080 Ti.

Заключение. Полученные результаты тестирования детектора лиц показывают, что при существенном уменьшении числа выполняемых операций сети, точность детектирования уменьшается в меньшей степени, в то время как полнота детектирования остается на том же уровне. Можно сделать вывод, что предложенная архитектура ИНС для детектирования лиц является оптимальным вариантом для приложений, работающих в условиях ограниченных ресурсов.

Список литературы

[1.] A survey on face detection in the wild: Past, present and future / S. Zafeiriou, C. Zhang, Z. Zhang // Computer Vision and Image Understanding, 2015. − Vol. 138. – PP. 1-24.

[2.] Rich feature hierarchies for accurate object detection and semantic segmentation / R. Girshick, J. Donahue, T. Darrell, J. Malik // Computer Vision and Pattern Recognition, 2014. – PP. 580-587.

[3.] You Only Look Once: Unified, Real-Time Object Detection / J. Redmon, S. Divvala, R. Girshick, A. Farhadi // Computer Vision and Pattern Recognition, 2016. – PP. 779-788.

[4.] YOLO9000: Better, Faster, Stronger / J. Redmon, A. Farhadi // Computer Vision and Pattern Recognition, 2017. – PP. 6517-6525.

[5.] SSD: Single Shot MultiBox Detector / W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A.C. Berg // European Conference on Computer Vision, 2016. – PP. 21-37.

[6.] Feature Pyramid Networks for Object Detection / Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie // Computer Vision and Pattern Recognition, 2017. – PP. 936-944.

[7.] Soft-NMS – Improving Object Detection with One Line of Code / N. Bodla, B. Singh, R. Chellappa, L. Davis // IEEE International Conference on Computer Vision, 2017. – PP. 5562-5570.

[8.] ImageNet Classification with Deep Convolutional Neural Networks / A. Krizhevsky, I. Sutskever, G.E. Hinton // Neural Information Processing Systems, 2012. – Vol. 25. – PP. 1-9.

[9.] Going Deeper with Convolutions / C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich // Computer Vision and Pattern Recognition, 2015. – PP. 1-9.

[10.] Rethinking the Inception Architecture for Computer Vision / C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna // Computer Vision and Pattern Recognition, 2016. – PP. 2818-2826.

[11.] Rigid-Motion Scattering for Texture Classification / L. Sifre, S. Mallat // arXiv.org. Preprint: 1403.1687.

[12.] Very Deep Convolutional Networks for Large-Scale Image Recognition / K. Simonyan, A. Zisserman // arXiv.org. Preprint: 1409.1556.

[13.] Approximation by Superpositions of a Sigmoidal function / G.V. Cybenko // Mathematics of Control, Signals and Systems, 1989. – Vol. 2. – PP. 303-314.

[14.] Rectified Linear Units Improve Restricted Boltzmann Machines / V. Nair, G. Hinton // International Conference on Machine Learning, 2010. – PP. 807-814.

[15.] Deep Residual Learning for Image Recognition / K. He, B. Singh, R. Chellappa, L. Davis // Computer Vision and Pattern Recognition, 2015. – PP. 770-778.

[16.] Batch normalization: accelerating deep network training by reducing internal covariate shift / S. Ioffe, C. Szegedy // International Conference on Machine Learning, 2015. – Vol. 37. – PP. 448-456.

[17.] Adam: A Method for Stochastic Optimization / D.P. Kingma, L.J. Ba // International Conference on Learning Representations, 2015. – PP. 1-13.

[18.] WIDER FACE: A Face Detection Benchmark / S. Yang, P. Luo, C. Loy, X. Tang // Computer Vision and Pattern Recognition, 2016. – PP. 5525-5533.

[19.] Learning Transferable Architectures for Scalable Image Recognition / B. Zoph, V. Vasudevan, J. Shlens, Q. Le // Computer Vision and Pattern Recognition, 2017. – PP. 8697-8710.

[20.] MobileNetV2: Inverted Residuals and Linear Bottlenecks / M. Sandler, A. Howard, V. Zhu, A. Zhmoginov, L. Chen // Computer Vision and Pattern Recognition, 2018. – PP. 4510-4520.

[21.] FDDB: A Benchmark for Face Detection in Unconstrained Settings // V. Jain, E. Learned-Miller // Technical Report UM-CS-2010-009, Dept. of CS, University of Massachusetts, 2010.

[22.] Precision and recall [Электронный ресурс] / Wikipedia – Режим доступа : https://en.wikipedia.org/wiki/Precision_and_recall – Дата доступа : 01.02.2020.

DEEP MULTI-SCALE FACE DETECTOR BASED ON DEEP NEURAL NETWORK

A.V. Susha
Postgraduate student of the BSUIR

M.I. Vashkevich,
Assistant Professor of the BSUIR

Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus
E-mail: isushik94@bsuir.by, vashkevich@bsuir.by

Abstract. The main objective of this work was a development of a deep artificial neural network for face detection purposes. The focus of its design was made on providing of the high performance of the detector and lowering of its computational power requirements by using: 1) factorization of convolution; 2) pointwise convolution; 3) combination of depthwise and pointwise convolution. The detector was compared with similar face detectors based on other well-known neural network architectures MobileNet and NasNet. The proposed face detector has a computational complexity equalling 5.1 MFLOPs, which is two times less than MobileNet's one (11,7 MFLOPs) and four times less than NasNet's one (22 MFLOPs). The detection time for 416 × 416 image was 5.12 ms (or 195 FPS) using GPU GeForce 1080 Ti, and 65.4 ms (or 15 FPS) using one processor core of Intel Core i7-8700K. The precision of our design is 85% and less on 4% than MobileNet has, and less on 9.5% than NasNet has.

Keywords: face detection, deep neural networks, convolutional neural networks.