

## ЧИСЛЕННЫЕ МЕТОДЫ И АНАЛИЗ ДАННЫХ

### Структурно-функциональный анализ и синтез глубоких конволюционных нейронных сетей

Ю.В. Визильтер<sup>1</sup>, В.С. Горбацевич<sup>1</sup>, С.Ю. Желтов<sup>1</sup>

<sup>1</sup> ФГУП «Государственный научно-исследовательский институт авиационных систем»  
(ФГУП «ГосНИИАС»), Москва, Россия

#### Аннотация

Предложен общий подход к структурно-функциональному анализу и синтезу глубоких конволюционных нейронных сетей, который позволяет регулярным образом определить: из каких типовых структурно-функциональных элементов могут строиться конволюционные нейронные сети; каковы необходимые математические свойства структурно-функциональных элементов; какие комбинации структурно-функциональных элементов являются допустимыми; каковы возможные пути построения и обучения глубоких сетей для анализа и распознавания нерегулярных, неоднородных или сложно структурированных данных (таких как нерегулярные массивы, данные различного размера и различной природы, деревья, скелеты, графовые структуры, 2D-, 3D- и ND-облака точек, триангулированные поверхности, аналитические описания данных и т.п.). Определён необходимый набор структурно-функциональных элементов. Предложены методики решения задач структурно-функционального анализа и синтеза конволюционных нейронных сетей на основе структурно-функциональных элементов и правил их комбинации.

**Ключевые слова:** глубокие нейронные сети, машинное обучение, структуры данных.

**Цитирование:** Визильтер, Ю.В. Структурно-функциональный анализ и синтез глубоких конволюционных нейронных сетей / Ю.В. Визильтер, В.С. Горбацевич, С.Ю. Желтов // Компьютерная оптика. – 2019. – Т. 43, № 5. – С. 886-900. – DOI: 10.18287/2412-6179-2019-43-5-886-900.

#### Введение

Функциональность перспективных систем технического зрения (СТЗ), входящих в состав систем управления сложными мобильными объектами, начиная с 2011–2012 гг. во многом определяется технологической революцией, происходящей в компьютерном зрении и машинном обучении. Эта революция прежде всего связана с появлением и широким распространением методов глубокого обучения и глубоких конволюционных нейронных сетей (ГКНС), обеспечивающих автоматическое решение задач анализа изображений и распознавания визуальных образов на уровне человека или даже выше. В силу этого имеющиеся до сих пор технические и технологические ограничения возможностей построения и обучения ГКНС напрямую влияют на достижимую функциональность и характеристики СТЗ и их носителей, в том числе пилотируемых и беспилотных летательных аппаратов различного назначения. В качестве таких наиболее важных ограничений возможностей ГКНС можно отметить следующие:

- 1) большинство существующих и наиболее успешных архитектур ГКНС предназначены для обработки регулярных массивов данных и не способны работать со сложно структурированными и неоднородными данными;
- 2) сам процесс конструирования и обучения ГКНС является сложным, длительным и нестабильным, т.е. не всегда завершается гарантированным успешным результатом.

Первая из указанных проблем является источником критического ограничения нынешней функциональности ГКНС, поскольку именно такие нерегулярные и сложно структурированные данные об окружающей среде необходимо анализировать в процессе решения задачи управления мобильными объектами в быстро меняющейся обстановке. При этом ГКНС, приобретающие знания путём обучения на примерах, должны плотно взаимодействовать с базами данных и знаний (в том числе геопространственных данных), где информация также представлена в виде графовых структур, вообще говоря, произвольного вида.

Вторая проблема связана с тем, что на сегодняшний день в области глубокого обучения сложилась классическая ситуация, когда практика намного опережает теорию. Множество научных групп и прикладных разработчиков используют глубокое обучение (*Deep Learning*) и глубокие конволюционные нейронные сети (CNN, ГКНС), получают при этом state-of-the-art результаты, но никакой теории построения и обучения CNN не существует. Есть лишь некоторые наборы стандартных топологий сетей (AlexNet [1], GoogleNet [2] и т.п.) и рецептов их обучения. Периодически появляются некоторые эвристические новации (например, MFM-пулинг [3]) без какого-либо теоретического обоснования, и они закрепляются в пуле ходячих рецептов, если дают какие-то полезные результаты. Такой способ развития аппарата машинного обучения на основе CNN без

теоретического обоснования сопряжён с необходимостью перебора конфигураций сети и стратегий обучения при решении каждой конкретной задачи, а для этого требуется многократно прогонять большие (или сверхбольшие) объёмы обучающих и тестовых данных через процедуру обучения с использованием мощных вычислительных кластеров.

Конечно, желательно иметь общую теорию построения и обучения CNN, которая была бы способна ответить на следующие основные вопросы:

- Какова должна быть сложность CNN для решения задач определённого типа на данных определённой сложности при обучении на выборках определённого размера?
- Как оптимально формировать структуру глубокой сети для конкретных типов задач и конкретных типов данных?
- Как оптимально формировать стратегию обучения глубокой сети для конкретных типов задач и конкретных типов данных?
- От чего зависит скорость обучения глубокой сети, и как на неё влиять в процессе обучения?
- Как заранее оценить достижимые результаты некоторой заданной глубокой сети при её обучении в некоторой конкретной задаче (тип данных, выборка и т.п.)?
- Как предсказать или хотя бы семантически интерпретировать структурные описания, которые порождает глубокая сеть на разных уровнях в результате обучения в некоторой конкретной задаче (тип и характеристики данных, тип и характеристики выборки и т.п.), т.е. как превратить CNN из «черного ящика» в «прозрачный»?

К сожалению, никаких предпосылок к созданию такой теории пока нет. Неясен даже математический аппарат, который позволил бы вести подобный теоретический анализ поведения таких сложных объектов, как CNN. Однако это не значит, что всякая систематическая работа по теоретическому анализу и развитию CNN сегодня невозможна. Представляется, что, напротив, она возможна и даже необходима, но с менее амбициозной целью – не моделировать и предсказывать характеристики и поведение CNN, а систематизировать и обобщать те удачные архитектуры и «рецепты», о которых шла речь выше. Такую теорию/методику предлагается назвать теорией структурно-функционального анализа и синтеза (СФАС) глубоких конволюционных нейронных сетей (ГКНС). СФАС должна ответить на следующие вопросы:

- Из каких типовых структурно-функциональных элементов (СФЭ) могут строиться глубокие сети?
- Каковы необходимые математические свойства этих элементов, и, соответственно, каковы могут быть их различные алгоритмические реализации?
- Какие комбинации СФЭ являются допустимыми и эффективными с точки зрения построения и обучения глубоких сетей?

- Каковы возможные пути построения и обучения глубоких сетей для анализа и распознавания нерегулярных, неоднородных или сложно структурированных данных (нерегулярные массивы, данные различного размера и различной природы, деревья, скелеты, графовые структуры, 2D-, 3D- и ND-облака точек, триангулированные поверхности, аналитические описания данных и т.п.)?

Последний вопрос, как было уже отмечено выше, является хотя и частным, но чрезвычайно важным для распространения методов глубокого обучения на всю совокупность возможных типов анализируемых данных в системах управления.

Результатом развития СФАС нам представляется некая многомерная систематическая таблица («таблица Менделеева»), каждое из «измерений» («строк» или «столбцов») которой соответствует набору возможных реализаций определённых структурно-функциональных элементов или способов их комбинирования и использования, а каждой ячейке таблицы (находящейся на пересечении этих «строк» и «столбцов») соответствует определённая структура (топология + способ функционирования/обучения) глубокой сети. Некоторые ячейки такой таблицы должны быть уже «заполнены», т.е. соответствовать каким-то известным структурам и конфигурациям CNN. Другие ячейки должны оказаться «пустыми», и это позволит на регулярной основе искать и предлагать варианты формирования глубоких сетей для решения тех или иных прикладных задач. Таким образом, СФАС должна указать на расширенный или обобщённый (по отношению к известному на сегодня) набор вариантов построения ГКНС. В данной работе мы не замахиваемся на вопрос «Как следует в тех или иных условиях наилучшим образом формировать и обучать ГКНС?». Рассматриваемый вопрос скромнее – «Какими способами в принципе возможно формировать и обучать ГКНС в этих условиях (в задачах такого-то типа применительно к таким-то данным)?». Ниже в данной работе мы постараемся установить, из каких типовых СФЭ строятся известные нам и зарекомендовавшие себя максимально эффективными ГКНС. Мы попробуем математически формализовать эти элементы и указать их характеристические свойства. Далее на основе этих свойств мы попробуем предложить несколько потенциально полезных вариантов альтернативных реализаций или комбинаций этих элементов.

Следует отметить, что в самой идее конструирования глубоких свёрточных сетей сложной структуры или сетей для анализа сложных нерегулярных данных нет ничего принципиально нового – такие идеи ранее неоднократно высказывались, обсуждались и реализовывались. Одна из первых попыток использовать структурно сложные ГКНС была предпринята в статье Network in network (NIN) [4]. В данной работе предлагалось использовать в конволюционных слоях вместо конволюционных фильтров простые классические нейронные сети, состоящие из нескольких

слоёв. Дальнейшим развитием такого подхода стало появление «модульных» сетей (GoogleNet [2], SqueezeNet [5], Inception v4 [6]). Например, в работе [2] был предложен т.н. Inception module, представляющий из себя набор параллельно выполняемых конволюционных слоёв, слоёв пулинга с различными параметрами и слоя конкатенации, объединяющего их результаты, а ГКНС состоит из последовательности таких «модулей». Другой подход к формированию сложной архитектуры сети был развит в работе [7], где ГКНС предлагается формировать как последовательность блоков, каждый из которых построен по принципу фрактальной структуры из классических конволюционных слоёв.

Во всех этих случаях сложная архитектура ГКНС была связана со сложной логикой обработки данных, а не с их собственной сложной пространственной структурой. Между тем графовое представление является естественным представлением первичных (анализируемых) данных для огромного количества задач, начиная от задач анализа химических элементов ([8, 9, 10]), заканчивая задачами понимания текста ([11, 12]) и описания сцены для решения задач обучения с подкреплением [12]. Рассмотрим проблему создания ГКНС для обработки таких сложных структурных данных на примере задачи обработки и анализа облаков трёхмерных точек. Такие облака точек обычно описываются графами очень большой размерности, имеющими тысячи и десятки тысяч вершин, и поэтому они требуют использования максимально быстрых алгоритмов обработки. Кроме того, в контексте создания систем технического зрения мобильных технических объектов наиболее важным приложением таких методов является работа с нерегулярными облаками точек от трёхмерных сканеров. Можно выделить несколько основных подходов к созданию таких ГКНС:

– *Многоракурсные сети (Multi-view CNN)* используют несколько проекций облака точек на плоскость либо несколько двумерных проекций трёхмерного объекта. Несмотря на примитивность данного подхода, ГКНС такого типа показывают достаточно высокие результаты при решении задач классификации и обнаружения.

– *Воксельные сети (Voxel networks)* используют воксельное представление данных. Например, в работе [13] предлагается представить облака точек в воксельном виде с последующей их обработкой средствами ГКНС с трёхмерными конволюционными фильтрами. Аналогичный подход используется и при обработке видеопоследовательностей в работе [14]. Основной проблемой такого подхода является очень большая размерность входных данных, что ведёт к малой скорости работы таких ГКНС и большим потребностям в памяти. Дальнейшим развитием такого подхода явилось использование иерархических решёток, например, в [15, 16, 17]. Это позволило существенно снизить необходимые вычисления и требуемую память. В [18] была предложена модификация

данного подхода, при которой отклики фильтров считались только по активным (непустым) вокселям, что в теории позволяет повысить скорость. Несмотря на существенные успехи этого направления, проблема большого объёма входных данных и низкой скорости трёхмерных конволюций является, по сути, главной проблемой воксельного подхода.

– В *облачных ГКНС (Point-cloud CNN)* входными данными является непосредственно облако точек как массив векторов  $(x, y, z)$ , т.е. вектор размерности  $3 \times N$  где  $N$  – число точек в облаке. Впервые ГКНС такого типа была описана в [18]. Т.к. входные данные представляют собой неупорядоченный набор точек, в данной работе предлагается использование специальных блоков, инвариантных к перестановке точек. Для сохранения информации о положении после каждого слоя к результирующему вектору конкатенируется вектор с координатами точек. Несмотря на простоту, данный подход позволяет с высоким качеством решать задачи классификации и сегментации. В работе [19] представлено дальнейшее развитие данного подхода, реализующее локальную агрегацию признаков с использованием пулинга.

– *Неевклидовы ГКНС* в качестве входных данных используют графовое представление. В ряде случаев графовое представление является исходным (например, 3D-модели, задаваемые триангуляцией поверхности). Прежде всего, к данной группе относятся методы, использующие спектральное представление графа [20, 21, 22]. Другой подход заключается в отображении исходного графа в двумерное пространство, в котором и происходит расчёт конволюций – глобально [23, 24, 25] либо локально [25, 26, 27].

Как видно из приведённого краткого обзора, в большинстве этих подходов при необходимости анализа нерегулярных структур входных данных (облаков точек) используются методы приведения таких данных к регулярной структуре с их дальнейшим анализом средствами классических регулярных ГКНС. Но даже там, где попытки построить структурные ГКНС предпринимаются, получающиеся архитектуры оказываются весьма экзотичными, поскольку на сегодняшний день не существует единых принципов, позволяющих применять ГКНС к нерегулярным данным напрямую.

В связи с этим ценность предлагаемого в данной статье подхода, как нам представляется, заключается в его системности и последовательности. Поскольку научное сообщество сейчас находится на этапе порождения эвристик, мы предлагаем искать и конструировать эти эвристики на регулярной основе и в рамках некоторой систематической терминологии, пригодной для описания широкого класса архитектур ГКНС. Потенциальным следствием разработки предлагаемой методики (систематических таблиц) СФАС для ГКНС может, как мы надеемся, явиться создание новой программной платформы для формирования и обучения ГКНС, обладающей существенно более широкими возможностями выбора структур глубоких сетей, чем ныне существующие среды типа Caffe и т.п.

### 1. Структурно-функциональный анализ ГКНС

Рассмотрим для примера структуру ГКНС, предложенную в [3], которую мы ранее использовали в своей работе по распознаванию лиц [28] (рис. 1). Это конволюционная нейронная сеть, решающая задачу построения описания лица (*face representation*) путём

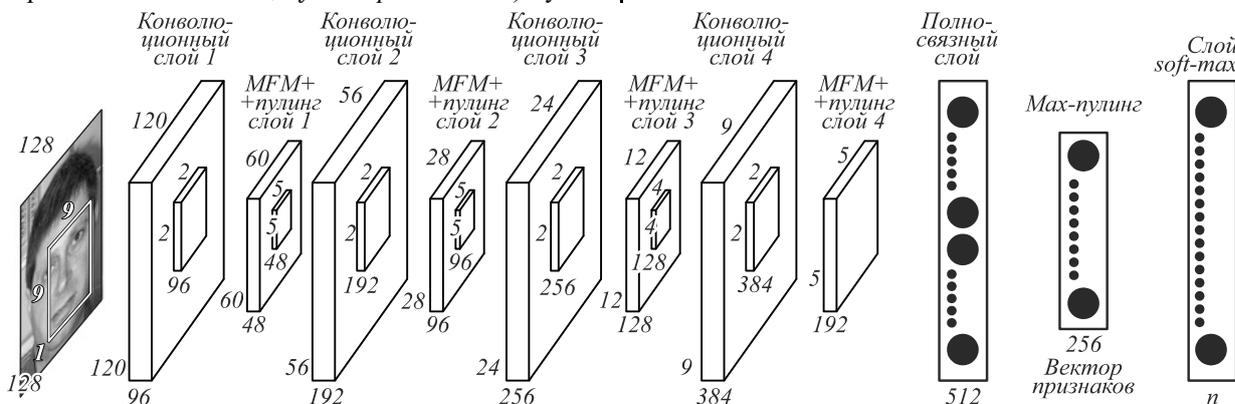


Рис. 1. Пример CNN (архитектура MFM deep net)

Снизу вверх (из «глубины» к «поверхности») последовательность изменения представления данных (*data representation*) от уровня к уровню может быть описана следующим образом:

2D-матрица (изображение) размера  $128 \times 128$  из элементов размерности 1 (скаляры)  $\rightarrow$

2D-матрица (изображение) размера  $60 \times 60$  из векторов размерности 48  $\rightarrow$

2D-матрица (изображение) размера  $28 \times 28$  из векторов размерности 96  $\rightarrow$

2D-матрица (изображение) размера  $12 \times 12$  из векторов размерности 128  $\rightarrow$

2D-матрица (изображение) размера  $5 \times 5$  из векторов размерности 192  $\rightarrow$

вектор размерности 512  $\rightarrow$

вектор размерности 256  $\rightarrow$

вектор размерности  $n = 10000$  (нужен только для обучения, впоследствии отбрасывается).

Заметим, что вместо «2D-матрица (изображение) размера  $60 \times 60$  из векторов размерности 48» можно писать «3D-матрица  $60 \times 60 \times 48$ », но такая запись нивелирует семантические и математические (геометрические, топологические) различия между измерениями. Разница примерно такая же, как между конструкциями «четырёхмерное пространство» и «пространство-время» = «трёхмерное пространство-время». В четырёхмерном пространстве  $R^4$  все измерения однородны. В пространстве-времени временное измерение кардинально отличается по своим свойствам: во времени точки полностью упорядочены, а в пространстве лишь частично; пространство изотропно по всем направлениям, а время имеет выделенное направление (стрела времени). Поэтому и геометрия пространства-времени – это не евклидова четырёхмерная геометрия, а геометрия пространства кватернионов (квадрат времени в метрике берётся с противоположным знаком относительно квадратов остальных коор-

динат). Аналогично *структурированные измерения* представлений данных в CNN имеют одну геометрическую и топологическую структуру (определяемую геометрией и топологией исходных данных, подаваемых на вход сети), а признаки измерения – другую (обычное евклидово пространство, причём все измерения являются независимыми).

Таким образом, описание данных на  $k$ -м уровне глубокой сети представляет собой объект из пространства

$$DR^{(k)} = X^{(k)} \times F^{(k)},$$

где  $X^{(k)}$  – некое конечное (поскольку состоит из конечного числа дискретных элементов) *структурированное многообразие* (традиционно для изображений  $X^{(k)} \subseteq R^2$ );  $F^{(k)} \subseteq R^{n(k)}$  –  $n(k)$ -мерное евклидово пространство признаков  $k$ -го уровня абстракции,  $n(k) = \dim(F^{(k)})$ ;  $\times$  – прямое декартово произведение, означающее, что к каждому элементу пространства  $X^{(k)}$  присоединён вектор из  $F^{(k)}$ .

В математических терминах  $DR^{(k)}$  называется *расслоением* многообразия  $X^{(k)}$ . В физике аналогами расслоений являются векторные поля. В области компьютерного зрения мы привыкли говорить о том, что точки изображения (или элементы его некоторого описания) характеризуются локальными векторами признаков (рис. 2).

Введём понятие *размера* (точнее *мощности* или *нормы*) многообразия  $\|X\|$ , равного количеству его элементов  $\|X\| = \#X$  (или монотонно зависящего от этого количества: в некоторых постановках могут учитываться не только элементы, но и связи между ними – см. ниже анализ в терминах гиперграфов). Легко заметить из приведённой выше схемы процесса преобразования представления данных в типовой CNN, что:

$$\|X^{(0)}\| \gg 1, \dim(F^{(0)}) = 1;$$

$$\exists k^*: \|X^{(k^*)}\| = 1, \dim(F^{(k^*)}) \gg 1;$$

$$\forall k < k^*: \|X^{(k)}\| > \|X^{(k+1)}\|, \dim(F^{(k)}) < \dim(F^{(k+1)}).$$

Иными словами, происходит процесс последовательной «перегонки» информации о геометрии и топологии объекта из структурированного многообразия  $X^{(0)}$  (или его первичного расслоения  $X^{(0)} \times F^{(0)}$ ) в глобальный вектор признаков  $F^{(k^*)}$ , который характеризует уже не отдельные элементы входного образа, а весь образ целиком. При этом на каждом шаге этого процесса размер (мощность) структурированного многообразия уменьшается, а размерность локальных векторов признаков растёт. В математических терминах здесь можно говорить о процессе *компактификации* многообразия  $X^{(k)}$  до тех пор, пока на шаге  $k^*$  оно не превратится в  $X^{(k^*)}$ , т.е. не окажется сжато в один элемент (точку), который уже не имеет структуры и, следовательно, уже ничем не напоминает исходное структурированное многообразие  $X^{(0)}$ . Такой результат достигается в первом полносвязном уровне ГКНС.

Дальнейшая обработка данных в ГКНС производится уже в неструктурированном пространстве глобальных признаков, не имеющих никакой локализации в исходной структуре многообразия. Этим занимаются остальные полносвязные уровни, и в результате размерность вектора признаков, как правило, уменьшается. Могут применяться и другие классификаторы (SVM, хеширование+метрики и т.п.). Но здесь уже нет специфики собственно конволюционных глубоких сетей (рис. 3). Таким образом, мы сосредоточимся в своём анализе на той части архитектуры

сети, которая итеративно работает с расслоениями структурированных многообразий и потому является собственно *глубокой*.

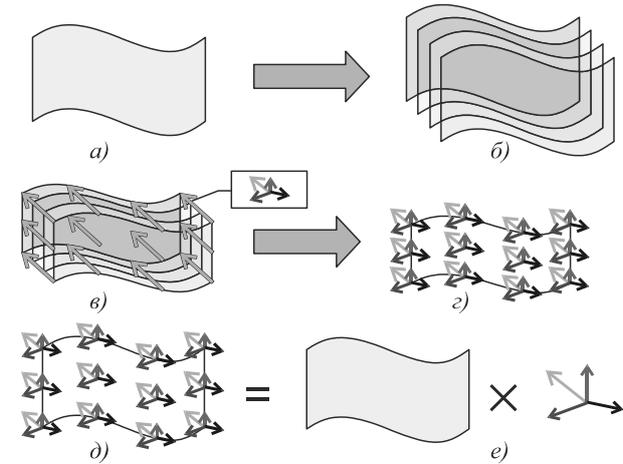


Рис. 2. Структура расслоения. Показаны: многообразие со своей структурой (внутренней геометрией и топологией) (а); расслоение многообразия как пачка одинаковых многообразий с заданными на них различными скалярными функциями (б); расслоение многообразия как многообразие с заданной на нём векторной функцией (аналог векторного поля в физике) (в); расслоение многообразия как многообразиие, в каждой точке которого привязано векторное пространство (г, д); расслоение многообразия как прямое произведение многообразия на векторное пространство (е)

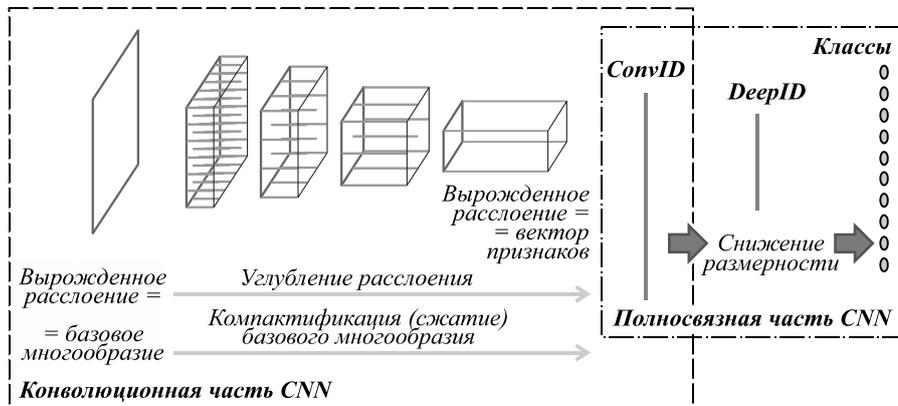


Рис. 3. Интерпретация CNN как процесса эволюции расслоения базового многообразия

Отображение  $t$ -расслоения некоторого многообразия  $X$  на вектор признаков большей размерности  $n$  будем далее называть *фичеризацией (featurization)* данных:

$$feat: X \times R^m \rightarrow R^n, n > t.$$

Как мы видели выше, *глубина* архитектуры ГКНС связана с последовательной компактификацией исходного многообразия с целью его финальной фичеризации. Однако мы пока никак не описали идею *конволюции* или *свёртки*. На самом деле она может быть описана в терминах *иерархической* (или *фрактальной*) *локальной фичеризации* с тем чрезвычайно важным (для задач обучения) условием, что локальная фичеризация во всех точках одного уровня представления данных (и одного типа – см. ниже) осу-

ществляется *одинаковым способом*. Возможно, вместо компактификации лучше говорить о *сжатии многообразия*. Опишем это формально.

Один шаг *сжатия (частичной компактификации)* с *частичной фичеризацией* будем обозначать

$$part\_shrink\_feat^{(k)}: X^{(k)} \times F^{(k)} \rightarrow X^{(k+1)} \times F^{(k+1)}, \|X^{(k)}\| > \|X^{(k+1)}\|, \dim(F^{(k)}) < \dim(F^{(k+1)}),$$

где  $k, k+1$  – *индексы последовательных слоёв CNN*.

Соответственно можно записать последовательность преобразований представления данных как:

$$global\_feat^{(k^*)}: X^{(0)} \times F^{(0)} \rightarrow F^{(k^*)}, gf^{(k^*)} = psf^{(0)} \circ psf^{(1)} \circ \dots \circ psf^{(k^*-1)}: X^{(0)} \times F^{(0)} \rightarrow X^{(1)} \times F^{(1)} \rightarrow \dots \rightarrow X^{(k^*-1)} \times F^{(k^*-1)} \rightarrow F^{(k^*)},$$

где  $psf$  – сокращенное обозначение  $part\_shrink\_feat$ .

Такой процесс итеративного сжатия с частичной фичеризацией представлен схематически на рис. 4.

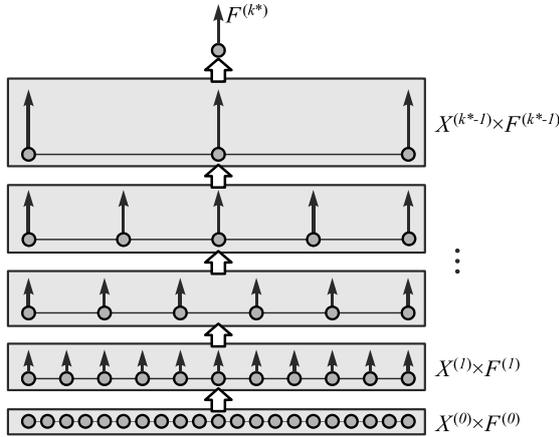


Рис. 4. Итеративное сжатие с частичной фичеризацией

В свою очередь, функцию  $part\_shrink\_feat^{(k)}$  можно представить как комбинацию трех функций:

- функция *сжатия (частичной компактификации) структурированного многообразия* (в стандартной реализации ГКНС это просто прореживание регулярного массива отсчетов изображения)

$$part\_shrink^{(k)}: X^{(k)} \rightarrow X^{(k+1)}, \|X^{(k)}\| > \|X^{(k+1)}\|;$$

- функция *выборки локального подмногообразия* для локальной фичеризации (в стандартной реализации ГКНС – это объединение носителей свёртки для точек нижнего уровня, участвующих в пулинге для точки следующего уровня)

$$get\_local\_submanifold^{(k)}: x \in X^{(k+1)} \rightarrow \mathbf{X}_x^{(k)} \subseteq X^{(k)}, \|\mathbf{X}_x^{(k)}\| = sub\_size(k) = const(x),$$

где  $x \in X^{(k+1)}$  – элемент выходного многообразия  $(k+1)$ -го уровня, которому ставится в соответствие подмножество (подмногообразие)  $\mathbf{X}_x^{(k)} \subseteq X^{(k)}$  входного многообразия  $k$ -го уровня, размер которого  $sub\_size(k)$  одинаков для всех  $x$ ;

- функция *фичеризации (описания вектором признаков) локального подмногообразия*

$$local\_feat^{(k)}: \mathbf{X}^{(k)} \times F^{(k)} \rightarrow F^{(k+1)}, \mathbf{X}^{(k)} \times F^{(k)} \subseteq X^{(k)} \times F^{(k)}, \dim(F^{(k)}) < \dim(F^{(k+1)}), \|\mathbf{X}_x^{(k)}\| = sub\_size(k).$$

Комбинация имеет вид:

$$part\_shrink\_feat^{(k)} = local\_feat^{(k)} \circ get\_local\_submanifold^{(k)} \circ part\_shrink^{(k)}: X^{(k)} \times F^{(k)} \rightarrow X^{(k+1)} \times (\mathbf{X}_x^{(k)} \times F^{(k)}) \rightarrow X^{(k+1)} \times F^{(k+1)} = X^{(k)} \times F^{(k)} \rightarrow X^{(k+1)} \times F^{(k+1)}.$$

Впрочем, можно описать функцию  $part\_shrink\_feat^{(k)}$  и в духе фильтров математической морфологии Серра [29] (рис. 5) – как комбинацию (объединение) операций фичеризации по всем подмногообразиям заданного размера  $\mathbf{X} \subseteq X^{(k)}, \|\mathbf{X}\| = sub\_size(k)$ :

$$part\_shrink\_feat^{(k)} = \cup \mathbf{X} \subseteq X^{(k)}, \|\mathbf{X}\| = sub\_size(k) \times local\_shrink\_feat^{(k)}(\mathbf{X} \times F^{(k)}),$$

$$local\_shrink\_feat^{(k)}: \mathbf{X}^{(k)} \times F^{(k)} \rightarrow x^{(k+1)} \times F^{(k+1)}, \mathbf{X}^{(k)} \times F^{(k)} \subseteq X^{(k)} \times F^{(k)}, x^{(k+1)} \in X^{(k+1)}, \dim(F^{(k)}) < \dim(F^{(k+1)}).$$

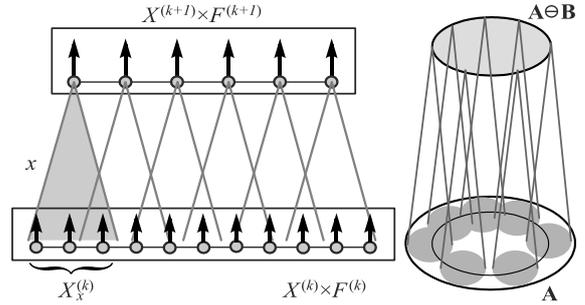


Рис. 5. Сжатие с фичеризацией как объединение локальных фичеризаций

Впрочем, наиболее традиционным способом декомпозиции функции  $part\_shrink\_feat^{(k)}$  является её представление в виде комбинации пары слоёв нейронной архитектуры – слоя конволюционных нейронов и слоя пулинга:

$$part\_shrink\_feat^{(k)} = convolution\_layer^{(k)} \circ pooling\_layer^{(k)}: convolution\_layer^{(k)}: X^{(k)} \times F^{(k)} \rightarrow X^{(k)} \times F^{(k+1)}, \dim(F^{(k)}) < \dim(F^{(k+1)}); pooling\_layer^{(k)}: X^{(k)} \times F^{(k+1)} \rightarrow X^{(k+1)} \times F^{(k+1)}, \|X^{(k)}\| > \|X^{(k+1)}\|.$$

Здесь также имеется полная семантическая аналогия с морфологией Серра:

- $part\_shrink\_feat^{(k)}$  – «морфологический» фильтр типа «закрытие», являющийся комбинацией операций «расширения» и «сжатия», при которых объём data representation сначала увеличивается, а затем уменьшается;

- $convolution\_layer^{(k)}$  – операция «расширения» расслоения за счёт увеличения размерности расслоения (глубины вектора признаков) в каждой точке многообразия;

- $pooling\_layer^{(k)}$  – операция «сжатия» расслоения за счёт уменьшения размера многообразия с сохранением достигнутой ранее глубины вектора признаков.

Данная аналогия схематично проиллюстрирована на рис. 6.

Рассмотрим теперь, как операция локальной фичеризации

$$local\_feat^{(k)}: \mathbf{X}^{(k)} \times F^{(k)} \rightarrow F^{(k+1)}, \mathbf{X}^{(k)} \times F^{(k)} \subseteq X^{(k)} \times F^{(k)}, \dim(F^{(k)}) < \dim(F^{(k+1)}), \|\mathbf{X}_x^{(k)}\| = sub\_size(k)$$

может быть реализована на основе линейной свёртки (конволюции). Для этого операцию  $local\_feat^{(k)}$  нужно окончательно представить в виде комбинации функций:

- функции *выборки локального подмногообразия* для локального признакового описания (в стандартной реализации ГКНС – это носитель ядра локальной свёртки)

$$get\_local\_conv\_submanifold^{(k)}: x \in X^{(k+1)} \rightarrow \rightarrow \mathbf{X}_x^{(k)} \subseteq X^{(k)}, \|\mathbf{X}_x^{(k)}\| = conv\_size(k) = const(x),$$

где  $conv\_size(k)$  – одинаковый для всех  $x$  размер локального подмногообразия, выбираемого для признакового описания, привязываемого затем к  $x$ ;

– функции *развёртывания расслоения* подмногобразия  $\mathbf{X}^{(k)} \times F^{(k)}$  в упорядоченный регулярный массив (двумерную матрицу) локальных признаков  $\mathbf{F}^{(k)}$  путём упорядочивания (последовательного сканирования) элементов подмногобразия

$$local\_scan^{(k)}: \mathbf{X}^{(k)} \times F^{(k)} \rightarrow \mathbf{F}^{(k)} = (F^{(k)}(x_1^{(k)}), \dots, F^{(k)}(x_{conv\_size(k)}^{(k)})), \mathbf{X}^{(k)} \subseteq X^{(k)}, \|\mathbf{X}^{(k)}\| = conv\_size(k);$$

– функции линейной свёртки (*конволюции*) двумерной матрицы признаков с обучаемым линейным ядром  $\mathbf{W}^{(k)}$  для проецирования на пространство признаков следующего уровня

$$conv\_proj^{(k)}: \mathbf{F}^{(k)} \times \mathbf{W}^{(k)} \rightarrow F^{(k+1)};$$

– функции *нелинейного преобразования признаков* (активации нейрона) с порогом  $t^{(k)}$

$$act\_func^{(k+1)}: F^{(k+1)} \times t^{(k+1)} \rightarrow F^{(k+1)}.$$

Таким образом,

$$local\_feat^{(k)}(\mathbf{W}^{(k)}, t^{(k+1)}) = get\_local\_conv\_subm^{(k)} \circ local\_scan^{(k)} \circ conv\_proj^{(k)} \circ act\_func^{(k+1)};$$

$$\mathbf{X}^{(k)} \times F^{(k)} \times \mathbf{W}^{(k)} \times t^{(k+1)} \rightarrow \mathbf{F}^{(k)} \times \mathbf{W}^{(k)} \times t^{(k+1)} \rightarrow F^{(k+1)} \times t^{(k+1)} \rightarrow F^{(k+1)}.$$

Данная модульная схема декомпозиции операций схематично представлена на рис. 7.

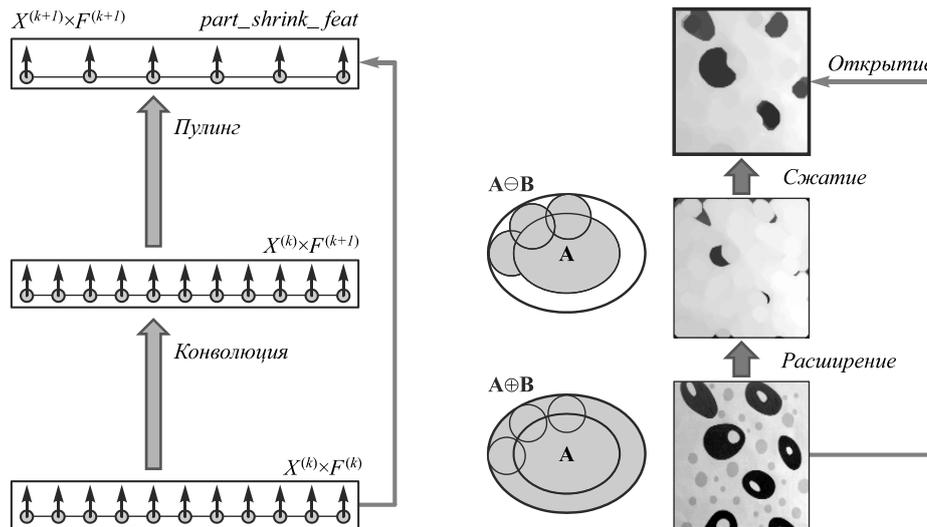


Рис. 6. Сжатие с фичеризацией как модульный оператор (свёртка + пулинг)

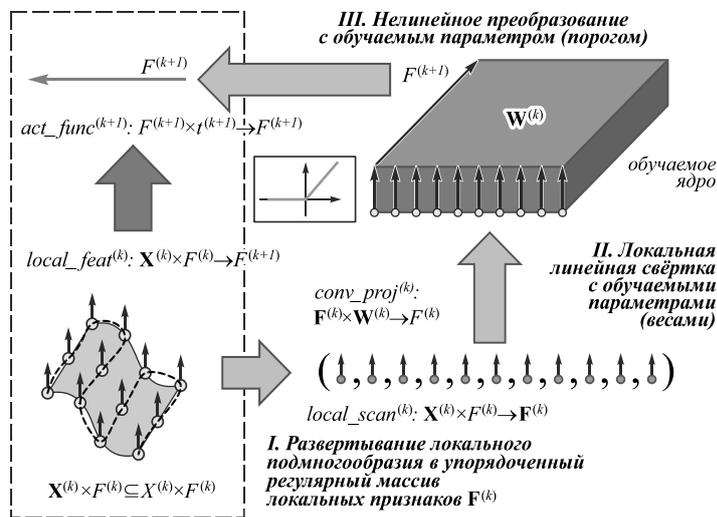


Рис. 7. Модульное представление локальной фичеризации

Заметим также, что, поскольку свёрточные признаки независимы, иногда семантически удобно разделить функцию  $conv\_proj^{(k)}$  на вектор (набор) отдельных скалярных свёрток  $scalar\_conv\_proj^{(k)}$  и объединяющую их функцию конкатенации  $concat\_features^{(k)}$ .

Наконец, операция пулинга  $pooling\_layer^{(k)}$  также может быть описана как комбинация одинаковых локальных операций локального пулинга  $local\_pool^{(k)}$ , состоящих из:

– функции *выборки локального подмногобразия* для локального пулинга (в стандартной реализации ГКНС – это множество точек нижнего уровня, участвующих в пулинге для точки следующего уровня)

$$get\_local\_pool\_submanifold^{(k)}:$$

$$x \in X^{(k+1)} \rightarrow \mathbf{P}_x^{(k)} \subseteq X^{(k)},$$

$$\|\mathbf{P}_x^{(k)}\| = pool\_size(k) = const(x),$$

где  $pool\_size(k)$  – одинаковый для всех  $x$  размер локального подмногобразия  $\mathbf{P}_x^{(k)}$ , по которому осуществляется пулинг с результатом в  $x$ ;

– функции *стирания структуры* расслоения подмногообразия  $\mathbf{X}^{(k)} \times F^{(k+1)}$  путём его преобразования в неупорядоченное множество локальных признаков  $\mathbf{P}^{(k+1)}$  путём последовательного присоединения к множеству всех элементов подмногообразия

$$local\_set^{(k)}: \mathbf{P}^{(k)} \times F^{(k+1)} \rightarrow \mathbf{P}^{(k+1)} = \{F^{(k+1)}(x_1^{(k)}), \dots, F^{(k+1)}(x_{pool\_size(k)}^{(k)})\}, \mathbf{P}^{(k)} \subseteq X^{(k)}, \|\mathbf{P}^{(k)}\| = pool\_size(k);$$

– функции *агрегирования признаков* входного неупорядоченного множества (в стандартной реали-

зации ГКНС – это операции типа mean, max, min и т.п.)

$$feat\_set^{(k+1)}: \mathbf{P}^{(k+1)} \rightarrow F^{(k+1)}.$$

Таким образом,

$$local\_pool^{(k)} = get\_local\_pool\_submanifold^{(k)} \circ$$

$$\circ local\_set^{(k)} \circ pool\_feat\_set^{(k+1)}:$$

$$\mathbf{P}^{(k)} \times \bar{F}^{(k+1)} \rightarrow \mathbf{P}^{(k+1)} \rightarrow \bar{F}^{(k+1)}.$$

Соответствующая модульная схема представлена на рис. 8.

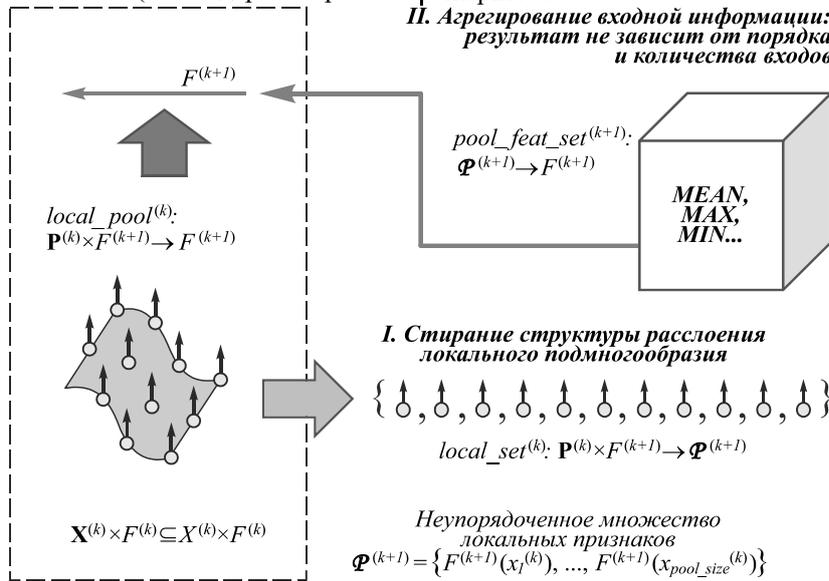


Рис. 8. Модульное представление локального пулинга

## 2. Свойства выделенных базовых СФЭ

Рассмотрим теперь ещё раз выделенные выше основные обобщённые структурно-функциональные элементы (СФЭ) ГКНС и опишем их свойства.

*Сжатие (уменьшение мощности) структурированного многообразия (part\_shrink)*. В типовой реализации ГКНС этой функции соответствует прореживание регулярного входного массива отсчётов. Математические свойства такой функции:

- *Неувеличение мощности*: результатом является многообразиие того же типа, мощность которого не больше, чем мощность входного многообразия.
- *Сохранение структуры*: желательное сохранение / близкое воспроизведение структурных характеристик многообразия (геометрии и топологии) с учётом масштаба преобразований (зависящего от мощности).

*Выборка локальной окрестности точки входного многообразия для локального признакового описания (get\_local\_conv\_submanifold)*. В типовой реализации ГКНС этому элементу соответствует прямоугольное окно-носитель ядра локальной свёртки заданного размера с центром в заданной точке. Математические свойства функции выборки окрестности:

- *Структурность*: результат операции при заданных центре и размере подмногообразия должен однозначно определяться локальной внутренней структурой (геометрией и топологией) входного многообразия.

- *Универсальность*: алгоритм формирования окрестности должен выдавать окрестность одинакового (заданного) размера (мощности = количества элементов) для всех точек многообразия.
- *Инвариантность*: алгоритм формирования окрестностей (подмногообразий) должен быть инвариантен к заданным группам преобразований, не меняющим внутреннюю структуру (мощность, состав, внутреннюю геометрию и топологию) входного многообразия.

*Развёртывание локальной окрестности в упорядоченный регулярный массив (двумерную матрицу) векторов локальных признаков путём некоторого стандартного упорядочивания (последовательного сканирования) элементов окрестности (local\_scan)*. В типовой реализации ГКНС этому элементу соответствует последовательное сканирование прямоугольного окна по строкам сверху вниз и в строке слева направо. Математические свойства данного СФЭ:

- *Структурность*: последовательность упорядочивания отсчётов должна однозначно определяться внутренней структурой (размером, геометрией и топологией) входного подмногообразия.
- *Стабильность*: алгоритм упорядочивания отсчётов должен выдавать одинаковую последовательность отсчётов для всех локальных подмногообразий, размер, внутренняя геометрия и топология которых совпадают.

- *Инвариантность*: алгоритм упорядочивания отсчётов должен быть инвариантен к заданной группе преобразований многообразия (это автоматически следует из стабильности, если глобальные преобразования многообразия сохраняют внутреннюю структуру локальных подмногообразий).

*Свёртка упорядоченного набора (матрицы) векторов признаков в единый вектор признаков большей размерности (глубины) с обучаемым набором (ядром) параметров свёртки (весов) (conv\_proj)*. В типовой реализации ГКНС этому элементу соответствуют:

- *Конволюция*: умножение матрицы на матрицу (линейная свёртка с ядром, комбинация входных признаков с обучаемыми весами).
- *Обучение*: веса свёрточных фильтров обучаются методом back propagation, основанным на дифференцировании сложных функций.

Математические свойства данного СФЭ:

- *Универсальность*: набор параметров (ядро) свёртки одинаков для всех наборов входных данных, получаемых в рамках одного входного многообразия (одного уровня обработки).
- *Неуменьшение глубины*: результатом свёртки является вектор, размерность которого не меньше, чем размерность векторов входного набора.
- *Упорядоченность входов*: результат свёртки зависит от количества и порядка записи/обработки входных отсчётов, поскольку каждому отсчёту (признаку) приписывается свой конкретный вес (значимость/корреляция с выходом).
- *Дифференцируемость*: операция свёртки должна позволять вычислять производные выходов как по параметрам, так и по входам (признакам) для обучения методом back propagation.
- *Линейность*: желательно, чтобы операция свёртки была линейной относительно как параметров, так и информационных входов (признаков), поскольку в таком случае производная сложной функции имеет простейший вид и обучение методом back propagation может быть эффективным.

*Нелинейное преобразование признаков (функция активации нейрона) с опциональными параметрами (act\_func)*. В типовых реализациях ГКНС этому элементу соответствуют:

- *Мягкая пороговая функция* (tanh, sof\_thr, сигмоида), ReLU, ... с параметром порога активации.
- *Обучение*: параметры (пороги) обучаются методом back propagation.
- *MfM-pooling*: групповое нелинейное преобразование (максимум) выходов нескольких нейронов (без явных параметров).

Основное математическое свойство данного СФЭ:

- *Дифференцируемость*: нелинейное преобразование признаков должно позволять вычислять производные выходов по входам и параметрам для обучения методом back propagation.

*Выборка локальной окрестности точки входного многообразия для локального агрегирования (пулинга) признаков (get\_local\_pool\_submanifold)*. В типовой реализации ГКНС этому элементу соответствует прямоугольное окно заданного размера с центром в заданной точке. Математические свойства данного СФЭ – структурность, универсальность, инвариантность (см. выше свойства *get\_local\_conv\_submanifold*).

*Стирание структуры локальной окрестности путём её преобразования в неупорядоченное множество отсчётов локальных признаков (local\_set)*. В типовых реализациях ГКНС в явном виде не производится – скрыто внутри функции локального пулинга. Математические свойства данного СФЭ:

- *Неупорядоченность входов*: результат операции не зависит от порядка записи и обработки элементов входного множества.
- *Сохранение значений*: значения признаков в ходе операции не изменяются.

*Агрегирование признаков входного неупорядоченного множества (pool\_feat\_set)*. В типовых реализациях ГКНС этому элементу соответствуют операции типа mean, max, min и т.п. Математические свойства данного СФЭ:

- *Универсальность*: ко всем наборам входных данных одного уровня применяются одинаковые операции агрегирования.
- *Неуменьшение глубины*: результатом операции является вектор, размерность которого не меньше, чем размерность векторов из входного множества.
- *Толерантность к составу входных данных*: результат операции не зависит от количества элементов входного множества (если оно не пусто).
- *Симметрия относительно порядка входных данных*: результат операции не зависит от порядка записи и обработки элементов входного множества.

*Операция сборки единого вектора признаков из упорядоченного набора входных векторов или скаляров (concat\_features)*. Формально можно считать, что в типовых реализациях ГКНС такая операция используется при формировании векторного выхода набора свёрточных фильтров. Математические свойства данного СФЭ:

- *Толерантность к типу входных данных*: результат операции не зависит от типа входных данных (можно объединять разнородные данные).
- *Упорядоченность входов*: результат операции зависит от порядка записи и обработки элементов входного набора данных.

Перечисленные свойства ограничивают множество возможных реализаций СФЭ множеством математических объектов, для которых эти свойства справедливы.

### 3. Методика анализа и синтеза CNN на основе СФЭ

Для решения задач синтеза CNN в приложениях, требующих работы со сложными нерегулярными

структурированными данными, предлагается использовать следующую последовательность шагов:

- 1) Выбрать базовую архитектуру CNN, которая может использоваться для решения рассматриваемой задачи в стандартном случае регулярных входных данных (массивов, тензоров).
- 2) Проанализировать (описать) структуру выбранной базовой CNN в терминах введённых выше СФЭ.
- 3) Синтезировать структуру новой CNN для работы с целевым типом входных данных путём замены исходных СФЭ на аналогичные по функциям и свойствам СФЭ для работы с целевым типом входных данных.
- 4) Осуществить программную реализацию и обучение новой CNN синтезированной структуры.

Обучение синтезированной по данной методике CNN может осуществляться стандартными методами, поскольку, как показано выше, структурные СФЭ конструируются таким образом, чтобы их математические свойства были полностью аналогичны свойствам исходных тензорных СФЭ. В силу этого, если базовая (тензорная) архитектура CNN успешно обучалась, то с высокой вероятностью будет успешно обучаться и преобразованная архитектура.

Для практической реализации такой методики структурно-функционального анализа и синтеза CNN на основе СФЭ и правил их комбинации предлагается построить и использовать т.н. *систематическую таблицу* СФАС. Столбцами систематической таблицы СФАС являются выделенные выше основные СФЭ. Строками систематической таблицы СФАС являются различные по семантике, геометрии и топологии типы входных данных, для которых необходимо реализовать каждый из СФЭ для того, чтобы построить ГКНС и осуществить её обучение. На пересечении строк и столбцов должны быть описаны способы (алгоритмы) реализации соответствующих СФЭ для соответствующих типов входных данных. Таким образом, задача определения общей структуры такой таблицы соответствует задаче *структурно-функционального анализа* глубоких сетей, а задача целенаправленного поиска нужных ячеек для решения задач распознавания различных типов сложных нерегулярных данных – *задаче структурно-функционального синтеза* CNN.

В качестве строк таблицы СФАС могут быть рассмотрены, например, следующие практически значимые типы входных данных:

- 1) 1D-, 2D- и 3D-регулярные массивы:
  - прямоугольные решётки (классический случай);
  - шестиугольные и др. регулярные решётки.
- 2) Скелеты и обогащённые скелеты (интересный частный случай с нетривиальной геометрией и топологией).
- 3) Деревья (иерархические деревья и просто графы без циклов).
- 4) Графы и гиперграфы:
  - сегментированные изображения;

- триангулированные 2D и 3D данные;
  - гиперграфы, описывающие марковские модели;
- 5) 2D-, 3D- и ND-облака точек.

При реализации процедур СФАС необходимо системно рассмотреть эти типы и структуры данных с точки зрения того, каковы их структурные (геометрические и топологические) особенности (внутренние метрики, симметрии, связность, фрактальность/маштабируемость и т.п.). После этого можно предложить различные способы заполнения соответствующих ячеек структурной таблицы.

В качестве примера различного заполнения ячеек таблицы СФАС можно рассмотреть различия в реализации функции выборки локального многообразия (окрестности) для регулярных и нерегулярных структур. На рис. 9 эти различия проиллюстрированы для случая скелетного описания формы фигуры.

Предложенная методика позволяет создавать, конфигурировать и обучать ГКНС, в том числе обладающие подобной нетривиальной и переменной (зависящей от самого анализируемого изображения или сигнала) структурой.

В заключение данного параграфа необходимо обсудить ограничения простейшего варианта методики СФАС, описанного в данной статье, и возможные направления будущего дополнения и расширения предложенного здесь общего подхода к построению структурных глубоких нейронных сетей.

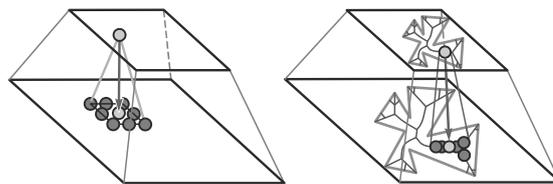


Рис. 9. Реализация функции выборки локального многообразия (окрестности) для регулярных массивов (слева) и нерегулярных структур (справа). В качестве другого примера нетривиальной структуры структурированных многообразий, на которые могут опираться глубокие сети, созданные в рамках подхода СФАС, можно привести дерево морфлетов [30] (результат морфологической древовидной сегментации изображения), к каждому узлу которого привязано соответствующее 2D-изображение (рис. 10)

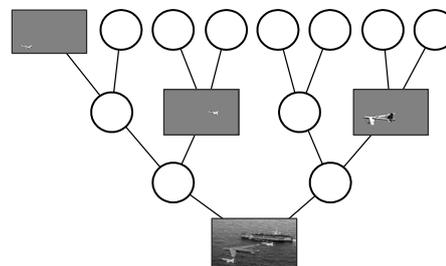


Рис. 10. Пример произведения многообразий различных типов: дерево морфлетов, к каждому узлу которого привязано 2D-изображение

Следует отметить, что в данной статье для простоты и единства изложения мы сосредоточились исключительно на тех функциях и структурах, которые имеют непосредственное отношение к математиче-

скому смыслу и реализуемости ключевых операций свёрточных сетей на нерегулярных типах данных. Между тем современные архитектуры CNN, конечно, включают ещё множество элементов и приёмов синтеза архитектур. В частности, такие архитектуры, как GoogLeNet и Inception-ResNet, включают Inception Module различных типов, которые осуществляют конкатенацию данных нескольких свёрток, не понижая мощности результирующего многообразия. Часто используется также подход с «переброской» данных через слои сети (Resnet Block), что также не укладывается явно в описанную выше простую схему функционирования CNN. Рекуррентные CNN (*Recurrent Convolutional Neural Network*) требуют дополнительных структурных модулей (LSTM и др). При реализации CNN часто используются такие служебные слои, как «dropout» (случайное отключение весов в процессе обучения) или «batch norm» (батч нормализация и другие типы нормализации). И этот список далеко не полон. Однако можно заметить, что программная и математическая реализация всех перечисленных структур и приёмов построения CNN для случая нерегулярных структур входных данных ничем не отличается от их реализации в случае регулярных данных (тензоров). Т.е. при реализации предложенной методики синтеза структурных CNN замене подлежат лишь те СФЭ, которые были нами рассмотрены выше (свёртки, пулинг и др.), а такие конструкции, как Inception Module, Residual Blocks или LSTM, просто автоматически переносятся в структуру новой сети без изменений, поскольку либо работают с векторами, а не с многообразиями (LSTM), либо сами являются комбинацией рассмотренных СФЭ, которые могут быть заменены с тензорных на структурные версии (Inception Module, Residual Blocks).

Отдельный случай, принципиально не сводимый к описанным инструментам СФАС, представляют собой архитектуры сетей типа автоэнкодер (Encoder-Decoder CNN), где после кодирования (снижения мощности многообразия с углублением признаков) на втором этапе происходит, наоборот, наращивание мощности многообразия со снижением глубины признаков. Для реализации такого типа сетей действительно требуются «обратные» СФЭ по отношению к описанным – элементы деконволюции, депулинга и т.п. Такие «обратные» СФЭ (которые используются и в некоторых других современных архитектурах, например, для выравнивания формата данных, приходящих с различных уровней), очевидно, можно также сконструировать для случая сложных структурированных данных по аналогии с тем, как они вводятся для регулярных (тензорных) данных. Однако в данной статье мы не рассматривали такую задачу, поскольку речь шла о конволюционных сетях в узком смысле, а автоэнкодеры являются «конволюционно-деконволюционными» сетями. Построение дополнительных инструментов СФАС для реализации всего богатого арсенала структур и подходов, реализуемых в современных CNN, должно, видимо, стать задачей дальнейшей регулярной работы по

уточнению и реализации предложенного общего подхода к построению структурных глубоких сетей. При этом суть подхода, заключающегося в разложении архитектур тензорных CNN на СФЭ, часть из которых затем при синтезе подлежит замене на аналоги для работы со структурированными данными, принципиальных изменений, вероятно, не потребует.

### Заключение

В работе предложен общий подход к структурно-функциональному анализу и синтезу глубоких конволюционных нейронных сетей. Формализм СФАС основан на рассмотрении процесса обработки информации в CNN как процесса фичеризации, т.е. последовательной (от уровня к уровню) «перегонки» информации о геометрии и топологии объекта из структурированного многообразия в глобальный вектор признаков, который характеризует уже не отдельные элементы входного образа, а весь образ целиком. Каждый шаг пространственного сжатия (частичной компактификации) с частичной фичеризацией представлен как комбинация трёх функций:

- функция сжатия (частичной компактификации) структурированного многообразия;
- функция выборки локального подмногообразия для локальной фичеризации;
- функция фичеризации (описания вектором признаков) локального подмногообразия.

В свою очередь, операция локальной фичеризации представлена в виде комбинации функций: функции выборки локального подмногообразия для локального признакового описания; функции развёртывания расслоения подмногообразия в упорядоченный регулярный массив локальных признаков; функции линейной свёртки (конволюции) массива признаков с обучаемым линейным ядром для проецирования на пространство признаков следующего уровня и функции нелинейного преобразования признаков (активации нейрона) с порогом. Операция пулинга также описана как комбинация одинаковых локальных операций локального пулинга, состоящих из: функции выборки локального подмногообразия для локального пулинга; функции стирания структуры расслоения подмногообразия путём его преобразования в неупорядоченное множество локальных признаков и функции агрегирования признаков входного неупорядоченного множества.

Перечисленный набор СФЭ является базовым для предлагаемой методики СФАС. Результатом развития СФАС должна явиться многомерная систематическая таблица, каждая из «строк» и «столбцов» которой соответствует набору возможных реализаций определённых структурно-функциональных элементов или способов их комбинирования и использования, а каждой ячейке таблицы соответствует определённая структура (топология + способ функционирования/обучения) глубокой сети. Задачу определения общей структуры такой таблицы предложено называть задачей структурно-функционального анализа глубоких сетей, а задачи целенаправленного поиска нужных ячеек для

решения задач распознавания различных типов сложных нерегулярных данных – задачами структурно-функционального синтеза CNN.

Для практической реализации предлагаемого подхода потребуется разработка специализированного программного обеспечения для формирования и обучения ГКНС на основе СФАС, которая составила бы конкуренцию ныне существующим библиотекам и средам разработки и обучения CNN (Caffe, Caffe2, TensorFlow, PyTorch, Teano, CNTK и др.). При этом потребуется разработать особые алгоритмы обучения и выполнения SCNN, которые позволят эффективно реализовывать быстрые массовые вычисления на основе удачной «трансляции» данных из произвольных структурных расстройств в регулярные массивы. Разработка таких алгоритмов и соответствующего программного обеспечения также должны явиться предметом дальнейшей работы в области реализации предложенного подхода.

### Благодарности

Исследование выполнено за счёт гранта Российского научного фонда (проект № 16-11-00082).

### Литература

- Krizhevsky, A.** ImageNet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G. Hinton // Proceedings of the 25th International Conference on Neural Information Processing Systems. – 2012. – Vol. 1. – P. 1106-1114.
- Szegedy, C.** Going deeper with convolutions / C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2015. – P. 1-9.
- Wu, X.** Learning robust deep face representation [Electronical Resource] / X. Wu. – 2015. – URL: <https://arxiv.org/pdf/1507.04844.pdf> (request date 02.12.2018).
- Lin, M.** Network in network [Electronical Resource] / M. Lin, Q. Chen, S. Yan. – 2013. – URL: <https://arxiv.org/pdf/1312.4400.pdf> (request date 17.04.2018).
- Forrest, N.** SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size [Electronical Resource] / N. Forrest, H. Song, W.M. Matthew, A. Khalid, J.D. William, K. Kurt. – 2016. – URL: <https://arxiv.org/abs/1602.07360> (request date 14.06.2018).
- Szegedy, C.** Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning [Electronical Resource] / C. Szegedy, S. Ioffe, V. Vanhoucke. – 2016. – URL: <https://arxiv.org/abs/1602.07261> (request date 01.11.2018).
- Larsson, G.** Fractalnet: ultra-deep neural networks without residuals [Electronical Resource] / G. Larsson, M. Maire, G. Shakhnarovich. – 2017. – URL: <https://arxiv.org/abs/1605.07648> (request date 02.11.2018).
- Duvenaud, D.** Convolutional networks on graphs for learning molecular fingerprints [Electronical Resource] / D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gomez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, R.P. Adams. – 2015. – URL: <https://arxiv.org/pdf/1509.09292.pdf> (request date 20.10.2018).
- De Cao, N.** MolGAN: An implicit generative model for small molecular graphs [Electronical Resource] / N. De Cao, T. Kipf. – 2018. – URL: <https://arxiv.org/pdf/1805.11973.pdf> (request date 10.10.2018).
- Gomes, J.** Atomic convolutional networks for predicting protein-ligand binding affinity [Electronical Resource] / J. Gomes, B. Ramsundar, E.N. Feinberg, V.S. Pande. – 2017. – URL: <https://arxiv.org/pdf/1703.10603.pdf> (request date 14.09.2018).
- Yao, L.** Graph convolutional networks for text classification [Electronical Resource] / L. Yao, C. Mao, Y. Luo. – 2018. – URL: <https://arxiv.org/pdf/1809.05679.pdf> (request date 02.12.2018).
- Xiong, W.** DeepPath: A reinforcement learning method for knowledge graph reasoning [Electronical Resource] / W. Xiong, T. Hoang, W.Y. Wang. – 2018. – URL: <https://arxiv.org/pdf/1707.06690.pdf> (request date 06.11.2018).
- Maturana, D.** VoxNet: A 3D convolutional neural network for real-time object recognition / D. Maturana, S. Scherer // IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – 2015. – P. 922-928. – DOI:10.1109/IROS.2015.7353481.
- Tran, D.** Learning spatiotemporal features with 3D convolutional networks / D. Tran, L. Bourdev, R. Fergus // IEEE International Conference on Computer Vision (ICCV). – 2015. – P. 4489-4497. – DOI: 10.1109/ICCV.2015.510.
- Riegler, G.** OctNet: Learning deep 3D representations at high resolutions / G. Riegler, A.O. Ulusoy, A. Geiger // CVPR. – 2017. – P. 6620-6629.
- Riegler, G.** OctNetFusion: Learning depth fusion from data / G. Riegler, A.O. Ulusoy, H. Bischof, A. Geiger // 2017 International Conference on 3D Vision (3DV). – 2017. – P. 57-66.
- Klokov, R.** Escape from cells: Deep KdNetworks for the recognition of 3D point cloud models / R. Klokov, V. Lempitsky // ICCV. – 2017 – P. 863-872.
- Qi, C.R.** PointNet: Deep learning on point sets for 3D classification and segmentation / C.R. Qi, H. Su, K. Mo, L.J. Guibas // CVPR. – 2017. – P. 652-660.
- Qi, C.R.** PointNet++: Deep hierarchical feature learning on point sets in a metric space / L. Yi, H. Su, L. Guibas // Proceedings of the 31st International Conference on Neural Information Processing Systems. – 2017. – P. 5105-5114.
- Bruna, J.** Spectral networks and locally connected networks on graphs / J. Bruna, W. Zaremba, A. Szlam, Y. LeCun // ICLR. – 2014.
- Henaff, M.** Deep convolutional networks on graph-structured data [Electronical Resource] / M. Henaff, J. Bruna, Y. LeCun. – 2015. – URL: <https://arxiv.org/pdf/1506.05163.pdf> (request date 15.12.2018).
- Defferrard, M.** Convolutional neural networks on graphs with fast localized spectral filtering [Electronical Resource]. M. Defferrard, X. Bresson, P. Vandergheynst. – 2016. – URL: <https://arxiv.org/pdf/1606.09375.pdf> (request date 21.09.2018).
- Sinha, A.** Deep learning 3D shape surfaces using geometry images / A. Sinha, J. Bai, K. Ramani // ECCV. – 2016. – P. 223-240.
- Maron, H.** Convolutional neural networks on surfaces via seamless toric covers / H. Maron, M. Galun, N. Aigerman, M. Trope, N. Dym, E. Yumer, V.G. Kim, Y. Lipman // ACM Transactions on Graphics. – 2017. – Vol. 36, Issue 4. – P. 71.
- Ezuz, D.** GWCNN: A metric alignment layer for deep shape analysis / D. Ezuz, J. Solomon, V.G. Kim, M. Ben-Chen // Computer Graphics Forum. – 2017. – Vol. 36, Issue 5. – P. 49-57.
- Masci, J.** Geodesic convolutional neural networks on Riemannian manifolds / J. Masci, D. Boscaini, M. Bronstein, P. Vandergheynst // ICCV. – 2015. – P. 832-840.

27. **Boscaini, D.** Learning shape correspondence with anisotropic convolutional neural networks / D. Boscaini, J. Masci, E. Rodoià, M.M. Bronstein // Proceedings of the 30th International Conference on Neural Information Processing Systems. – 2016. – P. 3197-3205.
28. **Vizilter, Yu.** Real-time face identification via CNN and boosted hashing forest / Yu. Vizilter, N. Kostromov, A. Vorotnikov, V. Gorbatsevich // The 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016). – 2016. – P. 78-86.
29. **Serra, J.** Image analysis and mathematical morphology. – London: Academic Press, Inc., 1982. – 610 p.
30. **Визильтер, Ю.В.** Морфлеты: новый класс древовидных морфологических описаний формы изображений / В.С. Горбацевич, С.Ю. Желтов, А.Ю. Рубис, А.В. Воротников // Компьютерная оптика. – 2015. – Т. 39, № 1. – С. 101-108. – DOI: 10.18287/0134-2452-2015-39-1-101-108.
31. **Su, H.** Multi-view convolutional neural networks for 3D shape recognition / H. Su, S. Maji, E. Kalogerakis, E.G. Learned-Miller // IEEE International Conference on Computer Vision (ICCV). – 2015. – P. 945-953.
32. **Huang, H.** Learning local shape descriptors with view-based convolutional neural networks / H. Huang, E. Kalogerakis, S. Chaudhuri, D. Ceylan, V. Kim, E. Yumer // ACM Transactions on Graphics. – 2018. – Vol. 37, Issue 1. – 6.
33. **Wu, Z.** 3D shapenets: A deep representation for volumetric shapes / S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2015. – P. 1912-1920.
34. **Maturana, D.** 3D convolutional neural networks for landing zone detection from LiDAR / D. Maturana, S. Scherer // ICRA. – 2015. – P. 3471-3478.
35. **Qi, C.R.** Volumetric and multi-view CNNs for object classification on 3D data / C.R. Qi, H. Su, M. Niener, A. Dai, M. Yan, L.J. Guibas // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – P. 5648-5656.
36. **Sedaghat, N.** Orientation-boosted voxel nets for 3D object recognition / N. Sedaghat, M. Zolfaghari, E. Amiri, T. Brox // Proceedings of the British Machine Vision Conference (BMVC). – 2017. – 97.
37. **Graham, B.** Submanifold sparse convolutional networks [Electronical Resource] / B. Graham, L. van der Maaten. – 2017. – URL: <https://arxiv.org/abs/1706.01307> (request date 10.12.2018).

#### Сведения об авторах

**Визильтер Юрий Валентинович**, 1970 года рождения, в 1992 году окончил ФГБОУ ВПО «Московский авиационный институт (национальный исследовательский университет)». В 1997 г. защитил кандидатскую, а в 2009 году – докторскую диссертацию. В настоящее время является начальником подразделения ФГУП «ГосНИИ Авиационных систем» (ГосНИИАС). Опубликовал более 200 научных работ (из них более 70 статей в рецензируемых журналах из списка ВАК). Область научных интересов: обработка и анализ изображений, цифровая фотограмметрия, машинное зрение, математическая морфология, распознавание образов, машинное обучение, биометрия. E-mail: [viz@gosniias.ru](mailto:viz@gosniias.ru).

**Горбацевич Владимир Сергеевич**, 1985 года рождения, в 2009 году окончил ФГБОУ ВПО «Московский авиационный институт (национальный исследовательский университет)». В настоящее время является начальником лаборатории ФГУП «ГосНИИ Авиационных систем» (ГосНИИАС). Опубликовал более 40 научных работ (из них 10 статей в рецензируемых журналах из списка ВАК). Область научных интересов: обработка и анализ изображений, цифровая фотограмметрия, машинное зрение, математическая морфология, распознавание образов, машинное обучение, биометрия. E-mail: [gvs@gosniias.ru](mailto:gvs@gosniias.ru).

**Желтов Сергей Юрьевич**, 1956 года рождения, известный учёный в области компьютерного зрения и обработки информации в системах управления, д-р ф.-м. наук, профессор, академик РАН, генеральный директор ФГУП «ГосНИИ Авиационных систем» (ГосНИИАС). Опубликовал более 200 научных работ, среди которых две книги. Область научных интересов: обработка и анализ изображений, цифровая фотограмметрия, машинное зрение, математическая морфология, распознавание образов, машинное обучение. E-mail: [zhel@gosniias.ru](mailto:zhel@gosniias.ru).

ГРНТИ: 28.23.37

Поступила в редакцию 7 марта 2019 г. Окончательный вариант – 27 июня 2019 г.

## Structure-functional analysis and synthesis of deep convolutional neural networks

*Yu.V. Vizilter<sup>1</sup>, V.S. Gorbatsevich<sup>1</sup>, S.Y. Zheltov<sup>1</sup>*

*<sup>1</sup> Federal State Unitary Enterprise "State Research Institute of Aviation Systems"  
(FGUP "GosNIIAS"), Moscow, Russia*

#### Abstract

A general approach to a structure-functional analysis and synthesis (SFAS) of deep neural networks (CNN). The new approach allows to define regularly: from which structure-functional elements (SFE) CNNs can be constructed; what are required mathematical properties of an SFE; which combinations of SFEs are valid; what are the possible ways of development and training of deep networks for analysis and recognition of an irregular, heterogeneous data or a data with a complex

structure (such as irregular arrays, data of various shapes of various origin, trees, skeletons, graph structures, 2D, 3D, and ND point clouds, triangulated surfaces, analytical data descriptions, etc.) The required set of SFE was defined. Techniques were proposed that solve the problem of structure-functional analysis and synthesis of a CNN using SFEs and rules for their combination.

**Keywords:** deep neural networks, machine learning, data structures.

**Citation:** Vizilter YuV, Gorbatshevich VS, Zheltov SY. Structure-functional analysis and synthesis of deep convolutional neural networks. *Computer Optics* 2019; 43(5): 886-900. DOI: 10.18287/2412-6179-2019-43-5-886-900.

**Acknowledgements:** The work was funded by Russian Science Foundation (RSF), grant No. 16-11-00082.

### References

- [1] Krizhevsky A, Sutskever I, Hinton G. ImageNet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems 2012*; 1: 1106-1114.
- [2] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015*: 1-9.
- [3] Wu X. Learning robust deep face representation. Source: <https://arxiv.org/pdf/1507.04844.pdf>.
- [4] Lin M, Chen Q, Yan S. Network in network. Source: <https://arxiv.org/pdf/1312.4400.pdf>.
- [5] Forrest N, Song H, Matthew VM, Khalid A, William JD, Kurt K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. Source: <https://arxiv.org/abs/1602.07360>.
- [6] Szegedy C, Ioffe S, Vanhoucke V. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Source: <https://arxiv.org/abs/1602.07261>.
- [7] Larsson G, Maire M, Shakhnarovich G. Fractalnet:ultra-deep neural networks without residuals. Source: <https://arxiv.org/abs/1605.07648>.
- [8] Duvenaud D, Maclaurin D, Aguilera-Iparraguirre J, Gomez-Bombarelli R, Hirzel T, Aspuru-Guzik A, Adams RP. Convolutional networks on graphs for learning molecular fingerprints. Source: <https://arxiv.org/pdf/1509.09292.pdf>.
- [9] De Cao N, Kipf T. MolGAN: An implicit generative model for small molecular graphs. Source: <https://arxiv.org/pdf/1509.09292.pdf>.
- [10] Gomes J, Ramsundar B, Feinberg EN, Pande VS. Atomic convolutional networks for predicting protein-ligand binding affinity. Source: <https://arxiv.org/pdf/1703.10603.pdf>.
- [11] Yao L, Mao C, Luo Y. Graph convolutional networks for text classification. Source: <https://arxiv.org/pdf/1809.05679.pdf>.
- [12] Xiong W, Hoang T, Wang WY. DeepPath: A reinforcement learning method for knowledge graph reasoning. Source: <https://arxiv.org/pdf/1707.06690.pdf>.
- [13] Maturana D, Scherer S. VoxNet: A 3D convolutional neural network for real-time object recognition. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2015*: 922-928. DOI: 10.1109/IROS.2015.7353481.
- [14] Tran D., Bourdev L, Fergus R. Learning spatiotemporal features with 3D convolutional networks. *IEEE International Conference on Computer Vision (ICCV) 2015*: 4489-4497. DOI: 10.1109/ICCV.2015.510.
- [15] Riegler G, Ulusoy AO, Geiger A. Octnet: Learning deep 3D representations at high resolutions. *CVPR 2017*: 6620-6629.
- [16] Riegler G, Ulusoy AO, Bischof H, Geiger A. OctNetFusion: Learning depth fusion from data. *3DV 2017*: 57-66.
- [17] Klovov R, Lempitsky V. Escape from cells: Deep KdNetworks for the recognition of 3D point cloud models. *ICCV 2017*: 863-872.
- [18] Qi CR, Su H, Mo K, Guibas LJ. PointNet: Deep learning on point sets for 3D classification and segmentation. *CVPR 2017*: 652-660.
- [19] Qi CR, Yi L, Su H, Guibas L. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *NIPS 2017*: 5105-5114.
- [20] Bruna J, Zaremba W, Szlam A, LeCun Y. Spectral networks and locally connected networks on graphs. *ICLR 2014*.
- [21] Henaff M, Bruna J, LeCun Y. Deep convolutional networks on graph-structured data. Source: <https://arxiv.org/pdf/1506.05163.pdf>.
- [22] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. Source: <https://arxiv.org/pdf/1606.09375.pdf>.
- [23] Sinha A, Bai J, Ramani K. Deep learning 3D shape surfaces using geometry images. *ECCV 2016*: 223-240.
- [24] Maron H, Galun M, Aigerman N, Trope M, Dym N, Yumer E, Kim VG, Lipman Y. Convolutional neural networks on surfaces via seamless toric covers. *ACM Trans Graph 2017*; 36(4): 71.
- [25] Ezuz D, Solomon J, Kim VG, Ben-Chen M. GWCNN: A metric alignment layer for deep shape analysis. *Computer Graphics Forum 2017*; 36(5): 49-57.
- [26] Masci J, Boscaini D, Bronstein M, Vandergheynst P. Geodesic convolutional neural networks on Riemannian manifolds. *ICCV 2015*: 832-840.
- [27] Boscaini D, Masci J, Rodoià E, Bronstein MM. Learning shape correspondence with anisotropic convolutional neural networks. *NIPS 2016*: 3197-3205.
- [28] Vizilter Yu, Kostromov N, Vorotnikov A, Gorbatshevich V. Real-time face identification via CNN and boosted hashing forest. *The 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016*; 78-86.
- [29] Serra J. *Image analysis and mathematical morphology*. London: Academic Press Inc; 1982.
- [30] Vizilter YuV, Gorbatshevich VS, Zheltov SY, Rubis AY, Vorotnikov AV. Morphlets: a new class of tree-structured morphological descriptors of image shape [In Russian]. *Computer Optics 2015*; 39(1): 101-108. DOI: 10.18287/0134-2452-2015-39-1-101-108.
- [31] Su H, Maji S, Kalogerakis E, Learned-Miller EG. Multi-view convolutional neural networks for 3D shape recognition. *ICCV 2015*: 945-953.
- [32] Huang H, Kalegorakis E, Chaudhuri S, Ceylan D, Kim V, Yumer E. Learning local shape descriptors with view-based convolutional neural networks. *ACM Trans Graph 2018*; 37(1): 6.
- [33] Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J. 3D shapenets: A deep representation for volumetric shapes. *CVPR 2015*; 1912-1920.
- [34] Maturana D, Scherer S. 3D convolutional neural networks for landing zone detection from LiDAR. *ICRA 2015*; 3471-3478.
- [35] Qi CR, Su H, Niener M, Dai A, Yan M, Guibas LJ. Volumetric and multi-view CNNs for object classification on 3D data. *CVPR 2016*; 5648-5656.

[36] Sedaghat N, Zolfaghari M, Amiri E, Brox T. Orientation-boosted voxel nets for 3D object recognition. BMVC 2017; 97.

[37] Graham B, van der Maaten L. Submanifold sparse convolutional networks. Source: <https://arxiv.org/pdf/1706.01307.pdf>.

---

#### *Author's information*

**Yuri Valentinovich Vizilter** (b.1970) graduated from Moscow Aviation Institute (National Research University) in 1992. Since 1997 is the Candidates of Technical Sciences, 2009 – the Doctor of Technical Sciences. Currently he works as the head of department at the FGUP “GosNIIAS”. Author of more than 200 scientific papers. Research interests are processing and image analysis, digital photogrammetry, computer vision, mathematical morphology, pattern recognition, machine learning, biometry. E-mail: [viz@gosniias.ru](mailto:viz@gosniias.ru).

**Vladimir Sergeevich Gorbatshevich** (b.1985) graduated from Moscow Aviation Institute (National Research University) in 2009. Currently he works as the head of laboratory at the FGUP “GosNIIAS”. Author of 40 scientific papers. Research interests are processing and image analysis, digital photogrammetry, computer vision, mathematical morphology, pattern recognition, machine learning, biometry. E-mail: [gvs@gosniias.ru](mailto:gvs@gosniias.ru).

**Sergey Yuryevich Zheltov** (b.1956) is a well - known scientist in the area of computer vision and data processing in control systems, Dr.Sci.Tech., professor, the full member of the Russian Academy of Sciences, General Director (CEO) of State Research Institute of Aviation Systems (FGUP “GosNIIAS”). He is an author and co-author of more than 200 scientific papers. Research interests are processing and image analysis, digital photogrammetry, computer vision, mathematical morphology, pattern recognition, machine learning. E-mail: [zhel@gosniias.ru](mailto:zhel@gosniias.ru).

---

*Received March 7, 2019. The final version – June 27, 2019.*

---