Featuring

1. Введение. Выделение признаков из изображения (*featuring*) является важной подзадачей во многих приложениях компьютерного зрения, таких как Structure from Motion (SfM), Visual Localization (VL), Object Detection, Image Matching и многих других. Правильное выделение признаков способно существенно улучшить качество решений в перечисленных задачах.

Признак (feature) – это часть информации, извлечённой из изображения, полезная для решения конкретной задачи. Признаками могут быть определённые структуры на изображении, такие как точки, рёбра или объекты. Они так же могут быть результатом применения алгоритма выделения признаков. Определять признаки для всего изображения слишком затратно, из-за чего признаки высчитываются лишь для особых точек.

Ключевая точка (key point) – это точка, которая на изображении является выразительной с точки зрения передаваемой информации. Например, точки, в которых направление границы резко меняется, или точка пересечения двух или более краевых рёберных сегментов. Обычно такие точки рассматриваются вместе со своими небольшими окрестностями (например, окно размера 5x5 вокруг ключевой точки).

Нахождение ключевых точек на изображении - отдельная, довольно сложная задача, решение которой осуществляется особым методом - *детектором* (feature detector). Качество детектора определяется обеспечением следующих свойств для ключевых точек:

- Отличимость (distinctness)
- Инвариантность (invariance)
- Стабильность (stability)
- Уникальность (uniqueness)
- Интерпретируемость (interpretability)
- Количество (quantity)
- Точность (accuracy)
- Эффективность (efficiency)

После нахождения детектором ключевых точек они передаются *дескриптору* (descriptor), который сопоставляет каждой точке вместе с её окружением некоторое информативное числовое представление для дальнейшего использования другими алгоритмами решения конкретных задач.

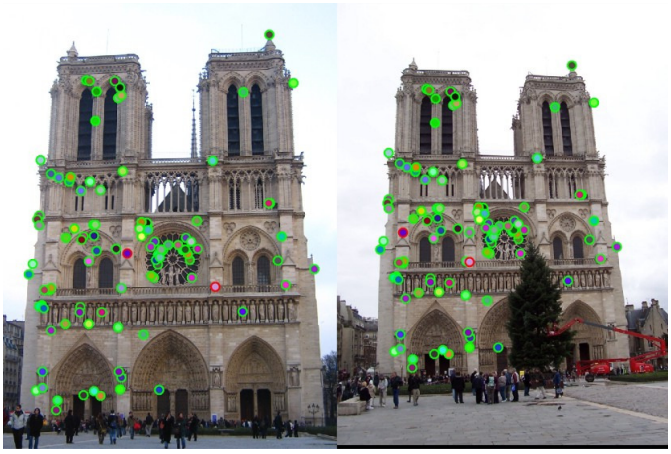

Рис. 1. Ключевые точки на изображениях.

Таким образом, алгоритм выделения признаков в общем случае состоит из трёх основных шагов:

1. Нахождение ключевых точек
2. Определение окрестностей для найденных ключевых точек
3. Вычисление вектора-признака

Полученные в результате векторы-признаки могут быть использованы, например, для задачи мэтчинга изображений.

2. Подходы. Для каждого из трёх основных шагов выделения признаков существуют свои подходы, которые могут быть разделены на созданные вручную (классические) и обучаемые (с применением глубокого обучения).

2.1. Классические подходы. До начала роста популярности свёрточных нейронных сетей в задачах компьютерного зрения стан-

дартом де-факто в таких задачах, как SoF, VL и Image Matching, были ручные методы детекторов и дескрипторов.

Методы детекторов были основаны на использовании интенсивности изображений или контуров. В качестве примеров классических методов можно привести Harris Corner или Moravec. Примерами же надёжных дескриптеров являются SIFT, SURF, BRIEF, ORB.

2.2. Подходы с обучением. С ростом популярности нейросетевых архитектур их применение в задаче выделения признаков не заставило долго себя ждать. Основная идея заключалась в делегировании сети задачи выделения необходимой информации из изображения.

3. Исследованные подходы. В процессе исследования научных статей по рассматриваемой задаче были отмечены довольно эффективные методы выделения признаков, обсуждаемые в данном разделе.

3.1. LF-Net. LF-Net: Learning Local Features from Images. Архитектура LF-Net имеет два основных компонента. Первая часть возвращает расположения ключевых точек, вторая - векторы признаков для них.

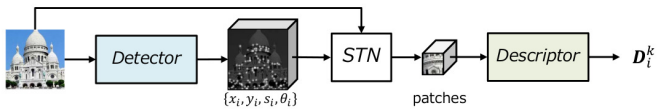

Рис. 2. Архитектура LF-Net.

Сначала к изображению применяется свёрточная нейронная сеть (ResNet с тремя блоками с 5x5 ядрами в каждом) для составления карты признаков, которая в дальнейшем может быть использована для поиска ключевых точек. Также для обеспечения невосприимчивого к масштабированию алгоритма поиска ключевых точек осуществляются изменения размеров полученных карт признаков и подача их объединения. Также для каждой точки вычисляется ориентация, путём накладывания на полученную ранее карту признаков ещё одной свёрточной нейронной сети с двумя выходами - синусом и косинусом.

Второй части сети подаются на вход некоторые окрестности ключевых точек, вычесленных ранее, на исхдном изображении, после че-

го они преобразуются к размеру 32x32 и подаются на вход свёрточным сетям, с наложенной на выход полносвязной нейронной сетью.

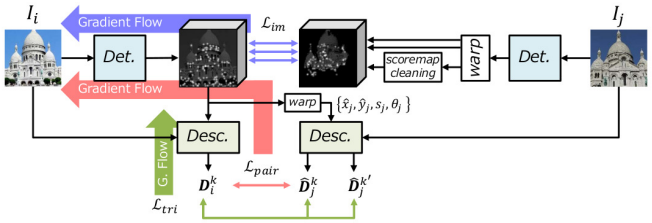

Рис. 3. Обучение LF-Net.

Однако наиболее удивительный подход применяется в этой статье к обучению. Рассматривается обычный датасет изображений, где в качестве целевых изображений берутся те же, но искажённые известными преобразованиями. Таким образом, после нахождения ключевых точек на исходном изображении, мы можем подать в качестве целевых ключевых точек - те же, но искажённые преобразованием.

Frame difference	SIFT	SURF	A-KAZE	ORB	LIFT	SuperPoint	LF-Net (w/rot-scl)	LF-Net (w/o rot-scl)
10	.320	.464	.465	.223	.389	**.688**	.607	**.688**
20	.264	.357	.337	.172	.283	**.599**	.497	.574
30	.226	.290	.260	.141	.247	**.525**	.419	.483
60	.152	.179	.145	.089	.147	**.358**	.276	.300
Average	.241	.323	.302	.156	.267	**.542**	.450	.511

Рис. 4. Качество в зависимости от изменения кадров.

Метод держится ну уровне state-of-the-art подходов в некоторых задачах, что удивительно, если учесть, что он совсем не требует размеченных данных. Также LF-Net хорошо справляется с очень различающимися изображениями, что говорит о приспособленности к серьёзным изменениям в угле изображения. Помимо этого метод является достаточно быстрым и выделяет 512 ключевых точек на видео разрешения QVGA (320x240) при 62 кадрах в секунду и VGA (640×480) при 25 кадрах на Titan X PASCAL. Ссылки на статью и код указаны в использованной литературе.

3.2. Key.Net. Keypoint Detection by Handcrafted and Learned CNN Filters. Авторы статьи использовали комбинацию ручного и

обучаемого подходов. Кроме того, извлечение признаков происходит по трём потокам, на два из которых подаются более сжатые версии исходного изображения, для выбора невосприимчивых к масштабированию ключевых точек.

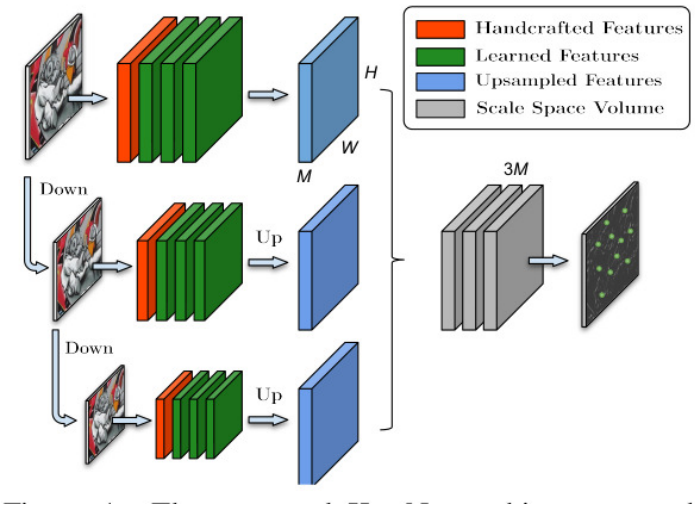

Рис. 5. Архитектура Key.Net.

Ручные фильтры были вдохновлены Harris и Hessian детекторами, которые используют первые и вторые производные (LocalJet) для определения характерных точек. Ручное представление подаётся на вход свёрточной нейронной сети с M фильтрами. После чего представления разных потоков конкатенируются и подаются на вход последней свёрточной сети для определения окончательной карты ключевых точек.

Было представлено две модели - Tiny-Key.Net (290 params) и Key.Net (5.9k params). Скорость архитектур для обработки изображений размером 600x600 составила 5.7 мс и 31 мс соответственно. Ссылки на статью и код указаны в использованной литературе.

3.3. D2-Net. A Trainable CNN for Joint Description and Detection of Local Features. Авторы статьи высказывают предположение о том, что шаг с нахождением ключевых точек изображения необходимо отложить и сначала определить векторы признаков для всех точек.

Помимо этого, предлагается использовать одну обучаемую сеть и для обнаружения ключевых точек, и для сопоставления векторов признаков.

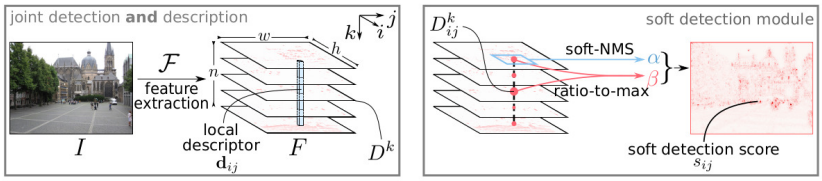

Рис. 6. Архитектура D2-Net.

Сначала к изображению применяется свёрточная сеть с n фильтрами. Таким образом, мы не только получаем n карт признаков, которые можно использовать для нахождения ключевых точек, но также становится известен и дескриптор размера n для точки с координатой (i, j), получаемый путём рассмотрения вектора точек из чисел, взятых из всех образованных карт признаков с координатами (i, j). В качестве свёрточной сети, используемой для извлечения карт признаков была использована предобученная VGG16. Авторы утверждают, что модель может плохо работать с большими изменениями углов ракурса, поскольку в обучающих данных около 90% данных имели отклонения угла меньше чем на 20 градусов. Тем не менее модель достигает лучших результатов на Aachen Day-Night localization dataset.

В статье архитектура применяется к задачам Image Matching, Visual Localization, 3D Reconstruction. Ссылки на статью и код указаны в использованной литературе.

3.4. DISK. (DIScrete Keypoints) Learning local features with policy gradient. Из-за того что выделение признаков и дальнейший мэтчинг производятся на разных этапаха, сложно обучить хорошие представления для ключевых точек. Авторы статьи решают эту проблему, используя принципы обучения с подкреплением. Вероятностная модель из статьи обучается с нуля и является довольно надёжной.

Авторы переходят к вероятностным терминам для определения функции математического ожидания выигрыша, зависящей от распределения дескрипторов на изображениях и параметров мэтчинга.

Распределения для дескрипторов задаются предобученной сетью U-Net. Далее в процессе обучения математическое ожидание увеличивается путём рассмотрения его градиента и обновления весов модели, которая включает в себя 1.1M обновляемых параметров.

Предложенная архитектура способна обрабатывать 7 кадров в секунду для входа 1024x1024. Ссылки на статью и код указаны в использованной литературе.

3.5. DELG (DEep Local and Global features). Unifying Deep Local and Global Features for Image Search. В данной статье авторы рассматривают два вида признаков - локальные и глобальные.

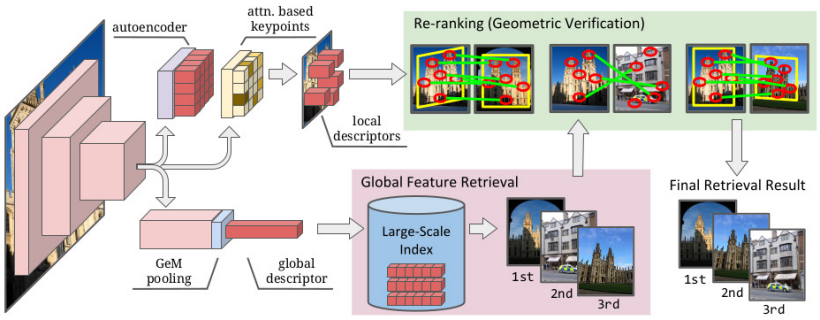

Рис. 7. Архитектура DELG.

Предлагаемая модель извлекает локальные и глобальные признаки. Глобальные признаки используются моделью для первичного выбора кандидатов для мэтчинга, локальные - для более точного установления соответствия. Первичная инициализация свёрточных нейронных сетей происходила моделями ResNet-50 и ResNet-101.

Для изображений размером 1024x1024 модель извлекает признаки в среднем за 198 мс NVIDIA Tesla P100 GPU.

Литература

1. Yuki Ono, Eduard Trulls, Pascal Fua, Kwang Moo Yi. LF-Net: Learning Local Features from Images // NeurIPS. 2018.

   Article: https://arxiv.org/abs/1805.09662

   GitHub: https://github.com/vcg-uvic/lf-net-release

2. Axel Barroso-Laguna, Edgar Riba, Daniel Ponsa, Krystian Mikolajczyk. Key.Net: Keypoint Detection by Handcrafted and Learned CNN Filters. // ICCV. 2019.

   Article: https://arxiv.org/abs/1904.00889

   GitHub: https://github.com/axelBarroso/Key.Net

3. Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, Torsten Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features // 2020.

   Article: https://arxiv.org/abs/1905.03561

   GitHub: https://github.com/mihaidusmanu/d2-net

4. Michał J. Tyszkiewicz, Pascal Fua, Eduard Trulls. DISK: Learning local features with policy gradient // NeurIPS. 2020.

   Article: https://arxiv.org/abs/2006.13566

   GitHub: https://github.com/cvlab-epfl/disk

5. Bingyi Cao, Andre Araujo, Jack Sim. Unifying Deep Local and Global Features for Image Search // ECCV. 2020.

   Article: https://arxiv.org/abs/2001.05027

   GitHub: https://github.com/tensorflow/models/tree/master/research/delf