

feature engineering в computer science

Отличная тема! **Feature Engineering** (Конструирование / Извлечение признаков) в компьютерном зрении — это фундаментальный этап, который предшествовал глубокому обучению и остается актуальным во многих задачах. Давайте разберем его детально.

Что это такое?

Feature engineering — это процесс создания информативных и дискриминативных **признаков** из необработанных данных (пикселей изображения), которые будут использоваться алгоритмами машинного обучения для решения задачи (классификация, детекция, сегментация).

В отличие от глубокого обучения, где нейросеть сама учится иерархии признаков из данных, здесь инженер или исследователь вручную проектирует эти признаки, опираясь на экспертные знания о задаче.

Ключевые методы и типы признаков

1. Признаки на основе цвета и интенсивности

Самые простые, но часто полезные.

- **Гистограмма цвета:** Распределение интенсивностей пикселей в цветовых каналах (RGB, HSV). Эффективно для поиска по цвету, классификации сцен.
- **Статистики:** Средняя яркость, контрастность, энтропия (мера беспорядка), стандартное отклонение.
- **Цветовые моменты:** Математические инварианты к масштабу и повороту.

2. Признаки на основе текстуры

Описывают пространственное распределение интенсивностей.

- **Матчики (фильтры) Габора:** Полосовые фильтры, выделяющие текстуры определенной частоты и ориентации. Похоже на работу ранних слоев сверточной нейросети (CNN).
- **LBP (Local Binary Patterns):** Кодирует локальные паттерны текстуры, сравнивая центральный пиксель с соседями. Очень популярен для классификации текстур и даже лиц.
- **Гистограмма ориентированных градиентов (HOG): Классический и мощный метод.** Описывает форму объекта, анализируя распределение направлений

градиентов яркости в локальных ячейках изображения. Широко использовался для детекции пешеходов и людей до эры CNN.

3. Признаки на основе формы и контуров

- **Края и контуры:** Выделяются с помощью фильтров (Собеля, Кэнни). На основе них можно строить дескрипторы.
- **SIFT (Scale-Invariant Feature Transform) и SURF (Speeded-Up Robust Features):** **Золотой стандарт до глубокого обучения.** Это **локальные дескрипторы**. Они находят ключевые точки (углы, участки с уникальной текстурой) и описывают их окружение таким образом, что дескриптор инвариантен к масштабу, повороту, частично — к освещению и небольшим деформациям. Основа для панорамирования (image stitching) и классического object recognition.
- **Дескрипторы контуров:** Например, дескриптор Фурье, дескриптор на основе расстояния до центра масс.

Пример пайплайна с фичами (до DL)

1. **Предобработка:** Приведение к серому, нормализация, подавление шума (Гауссовский фильтр).
2. **Обнаружение ключевых точек (для локальных признаков):** Используются детекторы (Harris corner, DoG для SIFT).
3. **Вычисление дескрипторов:** Для каждой ключевой точки считается ее "отпечаток" (SIFT-вектор размером 128, SURF — 64).
4. **Агрегация локальных признаков в глобальный вектор для изображения:**
 - **Bag of Visual Words (BoVW / Bag of Features):** Аналог Bag of Words из NLP.
 - Собираем все дескрипторы (например, SIFT) с обучающих изображений.
 - Кластеризуем их (K-means) — получаем **визуальный словарь** (Visual Vocabulary) из, например, 1000 "визуальных слов" (центроидов кластеров).
 - Каждое изображение представляется как гистограмма частот встречаемости этих "слов" (т.е. какой дескриптор к какому ближайшему центроиду принадлежит).
 - Эта гистограмма — и есть финальный вектор признаков для классификатора (SVM, Random Forest).
5. **Обучение классификатора:** На этих векторах обучается модель (например, SVM), которая отличает "кошку" от "автомобиля".

Сравнение: Feature Engineering vs. Глубокое обучение (CNN)

Аспект	Feature Engineering (Classic CV)	Глубокое обучение (CNN)
Подход	Инженерия "сверху вниз". Эксперт проектирует признаки.	Обучение "снизу вверх". Модель сама учит признаки из данных.
Требуемые данные	Может работать на небольших наборах данных.	Требует больших размеченных датасетов.
Вычислительная сложность	Часто меньше на обучении, но признаки могут быть "тяжелыми" для вычисления (SIFT).	Очень высокая на обучении, но инференс быстрый (особенно на GPU).
Интерпретируемость	Высокая. Признаки имеют понятный смысл (края, текстура).	Низкая. Признаки абстрактные, иерархические.
Усилия	Много времени уходит на подбор и комбинирование признаков.	Усилия смещаются на архитектуру сети, аугментацию, тонкую настройку.
Инвариантность	Нужно явно проектировать (SIFT инвариантен к масштабу).	Приблизительно учится инвариантности из данных (аугментации помогают).
Производительность	На многих задачах уступает современным CNN.	State-of-the-art на большинстве задач.

Когда Feature Engineering все еще актуален?

- Мало данных:** Нет огромного датасета для обучения глубокой сети с нуля.
- Области с устоявшейся практикой:** Например, в медицинской визуализации или дистанционном зондировании часто используют гибридные подходы.
- Высокие требования к интерпретируемости:** Нужно объяснить, почему модель приняла решение.
- Ограниченные вычислительные ресурсы:** На встраиваемых системах (IoT) предварительно вычисленные легкие признаки могут быть эффективнее запуска тяжелой нейросети.
- Специфические задачи:** Иногда простые признаки (например, соотношение сторон объекта) — это все, что нужно, и проще добавить их в модель, чем обучать CNN.
- Дополнение к DL:** Признаки, сконструированные вручную, могут конкатенироваться с выходом сверточных слоев, чтобы дать модели дополнительную экспертную информацию.

Итог

Feature engineering в компьютерном зрении — это богатая и важная дисциплина, которая заложила основу для современных методов. **Понимание этих методов критически важно**, даже если вы работаете с нейросетями, потому что:

- Оно дает интуицию о том, какие паттерны ищут алгоритмы.
- Помогает создавать эффективные аугментации данных.
- Позволяет решать задачи, где данных мало или нужна интерпретация.

Сегодня часто используется **гибридный подход**: CNN извлекает высокоуровневые признаки, а классические методы или domain knowledge добавляются на более поздних этапах или используются для предобработки.