

# Домашнее задание по лекции: Локальные дескрипторы и сверточные нейронные сети

В этом домашнем задании, основанном на материалах лекции, вы будете работать в две части. Первая часть посвящена локальным дескрипторам (алгоритм SIFT) и основам метода обратного распространения ошибки (*backpropagation*). Вторая часть посвящена сверточным нейронным сетям (CNN), архитектуре ResNet и проблеме затухающих градиентов. Для каждой части вам следует реализовать решения в Jupyter Notebook (Google Colab) и подготовить детальное пояснение выполненных шагов.

## 1 Часть 1. Локальные дескрипторы (SIFT) и *backpropagation*

В этой части задания вам предстоит работать с алгоритмом SIFT и проанализировать его основные этапы. Рекомендуется использовать библиотеку OpenCV или любую другую для вычисления SIFT-дескрипторов, а также понять, как в принципе работает метод обратного распространения ошибки при обучении нейронной сети.

- SIFT-дескрипторы и сопоставление.** Выберите два различных цветных или градиентных в сером оттенке изображения из набора ImageNet (размеры изображений можно уменьшить для ускорения обработки). Выполните следующие шаги:
  - Извлеките ключевые точки и вычислите SIFT-дескрипторы для каждого изображения.
  - Отобразите несколько ключевых точек и соответствующие им дескрипторы на самих изображениях (например, используя функции OpenCV или Matplotlib).
  - Выполните сопоставление (*matching*) дескрипторов между двумя изображениями (например, с помощью BFMatcher или FLANN из OpenCV) и визуализируйте найденные соответствия.
- Анализ шагов SIFT.** Опишите основные этапы алгоритма SIFT: масштабно-пространственную фильтрацию (Difference-of-Gaussians), выбор экстремумов (ключевых точек), назначение ориентаций ключевым точкам, построение гистограмм градиентов и нормализацию дескрипторов. Для каждого этапа укажите, какие операции выполняются над изображением (например, какие фильтры применяются, какие вычисляются градиенты). Затем прокомментируйте, как для каждого этапа можно было бы вычислить производные относительно входных пикселей (т.е. провести *backpropagation* через этот этап). Например, какие производные дают операции сглаживания (Gaussian blur), вычисления градиентов, заполнения гистограмм и нормализации дескриптора.
- Реализация SIFT (опционально).** Опишите идею реализации алгоритма SIFT без использования готовой функции, выделите ключевые шаги и математические формулы (например, фильтрация Difference-of-Gaussians, поиск максимума, построение гистограмм градиентов). Это задание необязательно выполнять полностью кодом, достаточно описать логику или псевдокод основных этапов.

В решении ожидается наличие графиков или изображений с отмеченными ключевыми точками и примерами сопоставления, а также текстовых пояснений, в которых вы объясняете каждый шаг алгоритма SIFT. Дополнительно приведите короткое рассуждение о принципах обратного распространения ошибки (*backpropagation*) в контексте рассматриваемых операций.

## 2 Часть 2. Сверточная нейронная сеть: ResNet и затухающие градиенты

В этой части задания вы сфокусируетесь на сверточных нейронных сетях, затухании градиентов и архитектуре ResNet. Основная цель — проиллюстрировать, как пропускные связи в ResNet помогают бороться с проблемой затухающих градиентов.

1. **Данные из ImageNet.** Используйте набор ImageNet (или его небольшую выборку) и выберите несколько классов (например, 2–5 классов) для эксперимента. Для ускорения вы можете работать с предварительно уменьшенными изображениями или их подмножеством.
2. **Сравнение CNN и ResNet.** Реализуйте (или загрузите из библиотеки, например, используя PyTorch или TensorFlow/Keras) две модели:
  - Простой сверточный классификатор (например, несколько сверточных слоев с последующим полносвязным слоем).
  - Сеть на основе ResNet (например, ResNet-18 или более простая версия ResNet).

Обучите обе модели на выбранных данных (можно использовать предобученную ResNet и обучать лишь последние слои (fine-tuning)).

Во время или после обучения оцените градиенты весов сети. Сравните, как изменяются градиенты в ранних и глубоких слоях для простой сети и для ResNet.

3. **Анализ затухающих градиентов.** Проанализируйте результаты обучения:
  - Постройте графики изменения функции потерь и точности (ассигасу) на обучающей и валидационной выборках для обеих моделей.
  - Вычислите норму градиентов (например, Frobenius-норму) для ранних и поздних слоев каждой сети.
  - Сравните, как ведут себя градиенты в простой сети и в ResNet. Проиллюстрируйте, уменьшаются ли градиенты сильнее в обычной сети по сравнению с ResNet.

Напишите выводы о том, как наличие *skip connections* влияет на обучение и проблему затухания градиента.

В решении второй части должны присутствовать обучающие кривые (потеря, точность), примеры работы моделей (например, изображения с предсказанными классами) и анализ полученных градиентов. Также поясните, как пропускные связи (skip connections) в ResNet помогают уменьшать влияние затухающих градиентов по сравнению с обычной сетью.

**Требования к оформлению.** Домашнее задание должно быть выполнено в виде Jupyter Notebook (Google Colab) с кодом, результатами выполнения (графики, таблицы, изображения) и подробными комментариями. В тексте решения приводите пояснения к каждому пункту, обосновывайте сделанные выводы и при необходимости используйте формулы. Код должен быть сопровожден поясняющими комментариями.