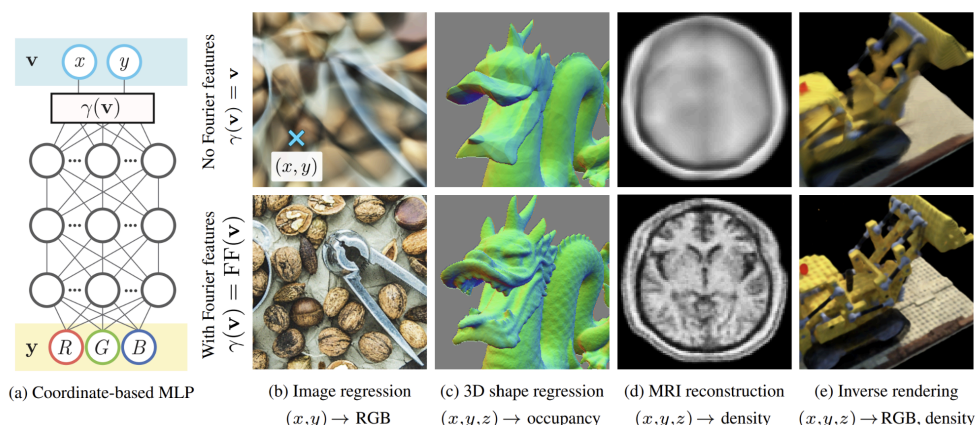


Задача 1. Восстановление изображения по томографическим проекциям с помощью координатной нейронной сети



В компьютерной томографии двумерное изображение объекта восстанавливается не по непосредственно наблюдаемым значениям его пикселей, а по набору **интегральных проекций**, полученных для различных направлений наблюдения. При небольшом количестве проекций задача восстановления становится недоопределённой, и качество результата существенно зависит от используемого представления изображения.

В работе Tancik et al., “**Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains**”, NeurIPS 2020 изображение представляется координатной нейронной сетью

$$f_{\theta}(x, y) \rightarrow I(x, y),$$

которая по пространственным координатам предсказывает интенсивность изображения. Авторы показывают, что обычные MLP имеют *spectral bias* и плохо восстанавливают высокочастотные детали. Для решения этой проблемы координаты предварительно преобразуются с помощью Gaussian Fourier Features:

$$\gamma(\bar{\mathbf{v}}) = [\cos(2\pi B\bar{\mathbf{v}}), \sin(2\pi B\bar{\mathbf{v}})]$$

где матрица B задаёт набор используемых пространственных частот.

Ссылка на статью:

https://proceedings.neurips.cc/paper_files/paper/2020/file/55053683268957697aa39fba6f231c68-Paper.pdf

Ссылка на код: <https://github.com/tancik/fourier-feature-networks>

Задание

Разобрать статью и воспроизвести в упрощённом виде эксперимент авторов для **2D computed tomography (CT)**.

В качестве объекта использовать синтетический фантом **Shepp–Logan** или другой синтетический фантом с известным ground truth.

Необходимо получить ограниченный набор его томографических проекций и по этим данным восстановить исходное двумерное изображение с помощью координатной MLP.

Сравнить два варианта:

- MLP без преобразования координат;
- MLP с Gaussian Fourier Features.

Исследовать влияние масштаба Fourier Features σ на результат. Сравнить методы количественно и визуально и сопоставить полученный эффект с результатами статьи.

Точное совпадение с числами авторов не требуется. Важно описать постановку эксперимента, объяснить сделанные упрощения и интерпретировать полученные результаты.

Вопросы

- Что такое spectral bias и как он проявляется в этой задаче?
- Почему признаки Фурье помогают MLP восстанавливать высокочастотные детали?
- Как параметр σ влияет на способность модели аппроксимировать изображение и обобщаться на неизвестные координаты?
- Какие ограничения есть у такого представления изображения и проведенного эксперимента?

Задача может быть выполнена на бесплатном **Google Colab** или **Kaggle Notebook (GPU)**. Использование больших датасетов, тяжёлых моделей и длительного обучения не требуется, но приветствуется при наличии ресурсов.

Контакт для связи: <https://t.me/DorogovaSo>

Задача 2. Сопоставление изображений разных доменов.



Huawei, 3072×4096



Sony, 6000×3376

В рамках международного соревнования по вычислительной фотографии (nightimaging.org) в качестве входных данных используются снимки с двух камер: DSLR камеры Sony и мобильного телефона Huawei. Для сравнения качества обработки изображений критически важно максимально качественно попиксельно сопоставить кадры с разных камер.

Задача:

Предлагается изучить способы сопоставлений изображений. В качестве метрики качества предлагается использовать корреляцию Пирсона между сопоставимыми изображениями. Для изображений с Huawei желательно ограничиться операциями изменения разрешения и кроем, чтобы минимизировать изменение сигнала.

1. Прочитать про методы template matching и RANSAC
2. Ознакомиться с статьей <https://habr.com/ru/companies/airi/articles/884176/>, чтобы понять специфику данных
3. Используя данные с `train_data` имплементировать сопоставление для .jpg данных. Хорошим показателем будет среднее качество выше 0.9. В качестве примера можно взять реализацию, описанную в <https://habr.com/ru/companies/airi/articles/884176/>
4. * До этого вы сопоставляли обработанные данные. Теперь нужно заменить данные с Huawei на сырые (.png) и сопоставить их. Сложностью будет наличие дисторсии, которую придется устранить. Для этого нужно с помощью `chess_huawei` научиться делать

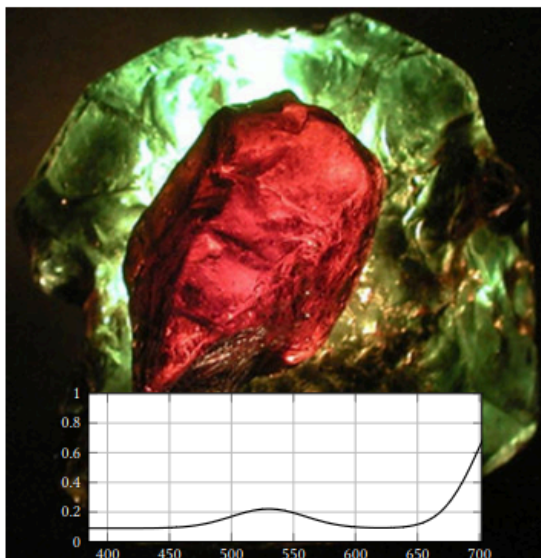
преобразование так, чтобы структура изображения совпадала. Посчитать метрики.

Возможные вопросы:

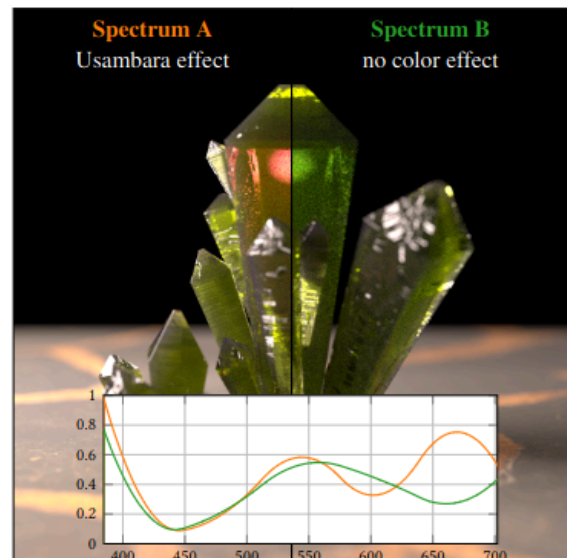
1. Какие метрики вообще можно использовать для сравнения изображений?
2. Что такое корреляция Пирсона? Почему она тут подходит или не подходит?
3. Какие способы выделения фичей для RANSAC вы знаете?
4. Что такое дисторсия?
5. Почему просто координатного сдвига в сопоставлении не хватит?

Контакт для связи: @neverlios

Задача 3. Генерация метамеров



a) Photograph of tourmaline



b) One-to-many spectral upsampling

Метамеризм — явление, при котором различные спектр-излучения вызывают идентичный цветовой отклик у наблюдателя. Исследование метамерных спектров имеет большое значение для решения прикладных задач колориметрии, таких как переход между цветовыми координатами и анализ границ человеческого цветового восприятия. Одним из способов генерации метамерных спектров является One-to-Many Spectral Upsampling of Reflectances and Transmittances (<https://arxiv.org/pdf/2306.11464>).

Задача

1. Разобрать работу и имплементировать предложенный метод.
2. Сравнить спектр отражения (reflectances) полученные этим методом и реальные. Для этого:
 - a. [Скачать](#) датасет спектр отражений и источников освещения
 - b. Получить цветовые отклики спектра под источником 1 и 2 с чувствительностями стандартного наблюдателя
xyz_matching_function.csv
 - c. Предложенным методом получить какой спектр бы давал соответствующие цветовых откликах под первым и вторым источником. (см. раздел 6.1 и 6.2 в работе)
 - d. Оценить точность восстановления между оригинальным и полученным спектром
3. * Сделать пример как в Figure 11/12

Контакт для связи: t.me/yasinowo

Задача 4. Оценка спектральной чувствительности камеры без доступа к камере.

Для задач точной цветопередачи, оценки освещения и преобразования цветовых координат между камерами нужно знать кривые спектральной чувствительности сенсора. Производители эти кривые не публикуют, а их прямое измерение требует монохроматора, спектрометра и физического доступа к камере. Поэтому открытые базы содержат измерения лишь для нескольких десятков моделей. В работе «Spectral Sensitivity Estimation Without a Camera» (<https://arxiv.org/abs/2304.11549>) предлагается обойтись без оборудования и без самой камеры. Кривые восстанавливаются по публичным цветовым матрицам из Adobe DNG Converter, а пространство допустимых решений ограничивается автоэнкодером, обученным на измеренных кривых.

Код и данные доступны в репозитории авторов:

<https://github.com/COLOR-Lab-Eilat/Spectral-sensitivity-estimation>

Задача

Разобрать статью и подробно описать предложенный метод. Воспроизвести* основной эксперимент: leave-one-out валидацию на базе измеренных кривых (Fig. 4–5 статьи). Сравнить полученные ошибки с заявленными авторами.

Вопросы

1. Почему система уравнений, полученная из цветовых матриц, недоопределена? Сколько в ней уравнений и сколько неизвестных?
2. Какую роль играет автоэнкодер? Чем он лучше классических ограничений вроде гладкости или неотрицательности?
3. Насколько надёжен ground truth, если для одной камеры кривые из разных источников расходятся до 17%? Как это влияет на интерпретацию метрики?
4. На каких прикладных задачах можно проверить полезность восстановленных кривых?

Бонусные задания

1. Сравнить автоэнкодер с регуляризацией на гладкость и с решением без регуляризации.
2. Провести leave-one-brand-out валидацию: исключать из обучения все камеры одного производителя.

Контакты куратора задачи: по любым возникающим вопросам обращаться к Владимиру Даниилу (тг: @AgentKolobok).

Задача 5. Оценка фенотипических признаков томата по RGB-D снимкам.



```
"D_fe87ef65": {  
  "height": 51.0,  
  "fw_plant": 212.34,  
  "leaf_area": 1147.05,  
  "number_of_red_fruits": 15.0
```

Для автоматического мониторинга теплиц важно уметь неразрушающим способом измерять параметры растений: высоту, массу, площадь листьев, число созревших плодов. Сейчас агрономы делают это вручную, а массу и площадь листьев измеряют деструктивно, срезая растение. В рамках международного соревнования Autonomous Greenhouse Challenge (Wageningen University & Research) опубликован открытый датасет: 479 растений карликового томата четырёх сортов (A, B, C, D) на разных стадиях роста. Каждое растение снято сверху камерой Oak-D S2 с высоты около 90 см: RGB-снимок, пара инфракрасных снимков и выровненная с RGB карта глубины. После съёмки у растения измерены высота, сырая масса, площадь листьев и число полностью красных плодов.

Задача:

Предлагается по RGB-D данным растения предсказать четыре признака. В качестве метрики качества используется официальная метрика соревнования RMSRE. Отложить 20% растений в тестовую выборку со стратификацией по сорту и использовать её один раз для финальной оценки; на остальных данных подбирать модель на кросс-валидации.

Прочитать про RGB-D камеры: как хранится карта глубины, что такое depth scale, что означают нулевые значения, как по карте глубины и внутренним параметрам камеры получить облако точек. Прочитать про вегетационные индексы для сегментации растений (например, ExG).

1. Ознакомиться с техническим описанием соревнования (приложено) и описанием датасета [4th Autonomous Greenhouse Challenge: Pre-trial Dwarf Tomato Measurements and Images](#), скачать архив Single_Plant_Camera.zip. Пример перевода карты глубины в облако точек: [WURgreenhouse/point-cloud-demo](#). Для понимания подходов прочитать статью про аналогичную задачу на салате: [Estimation of Greenhouse Lettuce Growth Indices Based on a Two-Stage CNN Using RGB-D Images](#).
2. Реализовать метрику RMSRE. Реализовать бейзлайн без нейросетей: выделить растение на снимке (порог по вегетационному индексу и/или по глубине), посчитать признаки (площадь маски, высота по карте глубины, доля красных пикселей и т.п.) и обучить линейную регрессию и/или градиентный бустинг. Посчитать метрику по каждому признаку и в среднем.
3. Дообучить предобученную CNN (например, ResNet18) на RGB, затем добавить канал глубины. Сравнить с бейзлайном из п. 3. Хорошим результатом будет заметное улучшение хотя бы по части признаков и понятное объяснение, откуда оно взялось.
4. Провести анализ ошибок: на каких растениях и сортах ошибка максимальна и почему, какой признак предсказывается хуже всего, какой вклад даёт глубина.
5. * Используя внутренние параметры камеры из конфигурационного файла, перевести карту глубины в облако точек и посчитать метрические признаки (реальная площадь проекции, высота над уровнем субстрата, объём). Проверить, улучшают ли они бейзлайн.
6. * Проверить обобщение на новый сорт: обучить модель на трёх сортах и протестировать на четвёртом. Насколько выросла ошибка и с чем это связано?

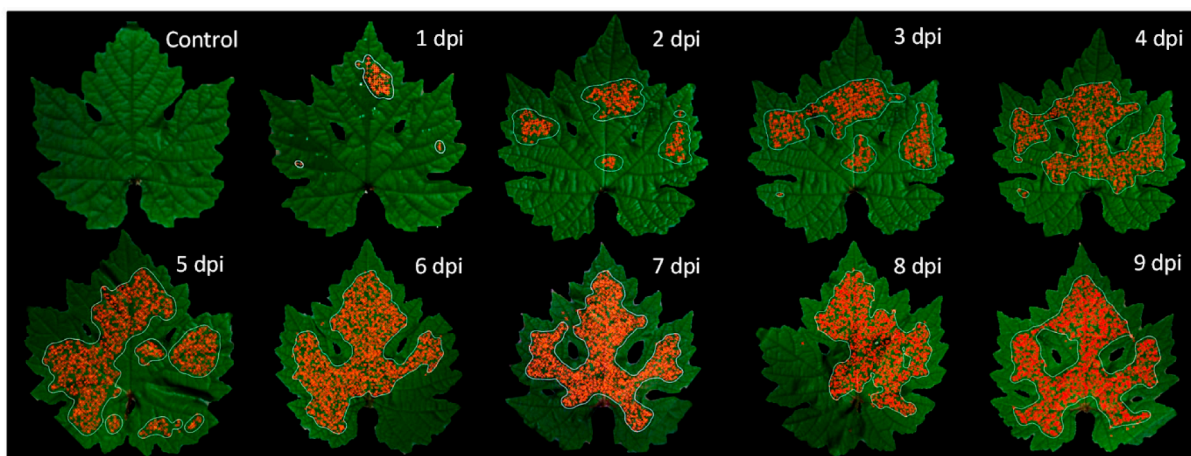
Возможные вопросы:

1. Как устроена карта глубины, что означают нулевые значения и как их обрабатывать?
2. Почему площадь растения в пикселях зависит от расстояния до камеры? Как это учесть?
3. Почему в метрике ошибка делится на истинное значение + 1? Что было бы без «+1» и как это влияет на признак «число плодов»?
4. Высота измеряется от уровня субстрата, а у сорта В он не такой, как у остальных. Как это учесть при оценке высоты по глубине?

5. Разметка ручная, высота и число красных плодов частично зависят от наблюдателя. Как это ограничивает достижимое качество?
6. Почему массу растения сложно оценить по снимку сверху?
7. Как подать карту глубины в сеть, предобученную на RGB-изображениях ImageNet?

Контакт для связи: @neverlios

Задача 6. Ранняя детекция болезни растения по гиперспектральным снимкам.



Обнаружить болезнь растения до появления видимых симптомов значит вовремя локализовать очаг и сократить применение фунгицидов.

Гиперспектральная камера измеряет спектр отражения в сотнях узких каналов, и изменения в листе, вызванные патогеном, проявляются в спектре раньше, чем становятся заметны глазу. В открытом датасете INRAE листья винограда сорта Каберне Совиньон заражены тремя штаммами ложной мучнистой росы (*Plasmopara viticola*) и ежедневно снимались камерой Specim FX10 (400-1000 нм, 224 канала) до 9 дней после инокуляции; для сравнения сняты здоровые контрольные листья. Для каждого снимка есть тёмный и белый референсы.

Задача:

Предлагается определить, начиная с какого дня после заражения болезнь надёжно обнаруживается по спектру. В качестве метрики качества предлагается использовать сбалансированную точность (balanced accuracy) классификации «здоровый / заражённый», посчитанную отдельно для каждого дня после инокуляции. Разбиение строго по листьям: один лист снимался в разные дни, и все его снимки должны попадать в одну часть выборки; из-за малого числа листьев удобно использовать leave-one-leaf-out.

Прочитать про гиперспектральную съёмку: как устроена push-broom камера, что такое гиперкуб, как по тёмному и белому референсам получить коэффициент отражения. Прочитать про вегетационные индексы (NDVI, PRI, индексы красного края).

1. Прочитать статью авторов датасета [Spatial–Spectral Analysis of Hyperspectral Images Reveals Early Detection of Downy Mildew on Grapevine Leaves](#), чтобы понять постановку эксперимента и их

результат: детекция на 2-й день после заражения при видимых симптомах на 4-й.

2. Скачать данные со [страницы датасета](#) (архивы feuille1-5.tar.gz по 1-1.6 ГБ, начать можно с одного-двух) и изучить [описание базы](#): какие листья, штаммы и дни есть, каких снимков не хватает. Прочитать гиперкубы, откалибровать в отражение, отделить лист от фона (например, порогом по NDVI или SAM3). Построить средние спектры контрольных и заражённых листьев по дням.
3. Реализовать бейзлайн: для каждого листа и дня извлечь спектральные признаки (средний спектр листа или спектры участков), обучить простой классификатор (PCA + логистическая регрессия или SVM). Посчитать метрику по дням и определить самый ранний день надёжной детекции. Сравнить с результатом статьи.
4. Сравнить три набора признаков: полный спектр, несколько вегетационных индексов и небольшое число отобранных каналов. Какие участки спектра важнее всего для ранней детекции?
5. * Построить попиксельную карту поражения: обучить классификатор на пикселях с явными симптомами на поздних днях и применить его к ранним снимкам тех же листьев. Показать, как растёт поражённая площадь во времени.
6. * Проверить, не выучил ли классификатор что-то кроме болезни: контрольные листья сняты только в часть дней, и модель может отличать дни съёмки, а не здоровье листа. Предложить способ это проверить и, если нужно, исправить.

Возможные вопросы:

1. Чем гиперспектральная съёмка отличается от RGB и мультиспектральной?
2. Зачем нужны тёмный и белый референсы? Что будет, если работать с сырыми значениями?
3. Почему разбиение нужно делать по листьям, а не по пикселям или снимкам?
4. Почему болезнь может быть видна в спектре раньше, чем глазом? В каких диапазонах вы ожидаете изменений?
5. Что такое проклятие размерности и как оно проявляется при 224 каналах и нескольких десятках листьев?
6. Почему для этой задачи ассурасу хуже, чем balanced accurasy?

7. Какие ошибки в постановке эксперимента могут дать завышенную точность?

Контакт для связи: по любым возникающим вопросам обращаться к Владимирову Даниилу (тг: @AgentKolobok).

Задача 7. Оценка цвета освещения сцены (цветовая константность).



Цвет объекта на снимке зависит не только от самого объекта, но и от освещения: один и тот же лист под солнцем, натриевой лампой или светодиодной досветкой камера запишет по-разному. Чтобы цвет можно было использовать как признак (например, для оценки спелости плодов или симптомов болезней), камера должна оценить цвет источника и скомпенсировать его, то есть сделать баланс белого. Для исследования этой задачи в ИППИ РАН совместно с Загребским университетом собран датасет Cube++: в каждой сцене стоит калибровочный объект SpyderCube, по которому измерен истинный цвет освещения. Его упрощённая версия SimpleCube++ содержит 2234 снимка с одним источником света в виде 16-битных PNG (линейные данные с сенсора, уменьшенные в 4 раза), разбитых на train и test, и занимает около 2 ГБ.

Задача:

Предлагается по линейному изображению оценить цвет источника освещения (вектор RGB). В качестве метрики качества используется угловая ошибка между оценкой и истинным цветом освещения из gt.csv. На тестовой выборке нужно привести среднее, медиану, trimean, среднее по лучшим 25% и худшим 25% снимков. Тестовую выборку использовать только для финальной оценки, все параметры подбирать на train.

Прочитать про цветовую константность и устройство конвейера обработки изображения в камере: линейные RAW-данные, уровень чёрного, насыщение, баланс белого, гамма-коррекция. Прочитать, как считается угловая ошибка.

1. Ознакомиться со статьёй [The Cube++ Illumination Estimation Dataset](#), чтобы понять, как получена разметка. Скачать SimpleCube++ (архив SimpleCube++.zip на [Zenodo](#)) и изучить описание в [репозитории](#)

[датасета](#): как читать PNG, какие уровни чёрного и насыщения, что лежит в gt.csv и в файлах свойств снимков.

2. Реализовать классические методы оценки освещения: Gray World, White Patch (max-RGB), Shades of Gray, Gray Edge. Корректно обработать уровень чёрного и насыщенные пиксели. Параметры методов (например, степень p в Shades of Gray) подбирать на train. Посчитать метрики на test.
3. Реализовать обучаемый метод: например, гистограмма хроматичности или набор статистик изображения на входе и небольшой регрессор (MLP, градиентный бустинг) или CNN на изображениях. Сравнить с классическими методами из п. 3.
4. Реализовать модель FC4: [ссылка](#).
5. Провести анализ ошибок: используя разметку свойств сцен (время суток, помещение или улица, тени), посчитать метрики по группам. Показать несколько снимков с наибольшей ошибкой до и после коррекции и объяснить, почему метод на них ошибается.
6. * Посчитать для лучших методов reproduction angular error и сравнить ранжирование методов с ранжированием по обычной угловой ошибке. Объяснить, откуда берутся различия.

Возможные вопросы:

1. Что такое цветовая константность и зачем она нужна в камере?
2. Почему освещение оценивают по линейным RAW-данным, а не по JPEG?
3. Зачем вычитать уровень чёрного и что делать с насыщенными пикселями?
4. Почему используется угловая ошибка, а не евклидово расстояние между векторами RGB?
5. Зачем, кроме среднего, приводить медиану и среднее по худшим 25% снимков?
6. На каких предположениях основаны Gray World и White Patch? Приведите сцены, где они ломаются.
7. Что произойдёт, если калибровочный объект останется в кадре при обучении и тестировании?

Контакт для связи: @neverlios

Задача 8. Какой вы наблюдатель?



Функции цветового соответствия (Color Matching Functions, CMF) — это математическая модель чувствительности трёх типов колбочек человеческого глаза (L, M, S) к свету в зависимости от его длины волны. На основе этих функций, стандартизированных CIE (Международной комиссией по освещению) в 1931 и 1964 годах, работают все современные цветовые модели, приборы и системы цветопередачи. Однако эти стандарты описывают усреднённое цветовосприятие, тогда как в реальности люди видят цвета по-разному из-за индивидуальных особенностей зрительной системы. Явление, при котором образцы или излучения с разными спектральными характеристиками кажутся одинаковыми по цвету одному человеку, но разными — другому, называется метамеризмом наблюдателя.

Вам предлагается разобрать **статью**, посвященную исследованию метамеризма наблюдателя, и воспроизвести описанные в ней эксперименты.

Задание:

1. Симулировать по Монте-Карло 10.000 виртуальных наблюдателей (функций CMF) с помощью **модели наблюдателя Асано**.

Python-реализацию модели можно найти в библиотеке **luxpy** (модуль **indvcmf**).

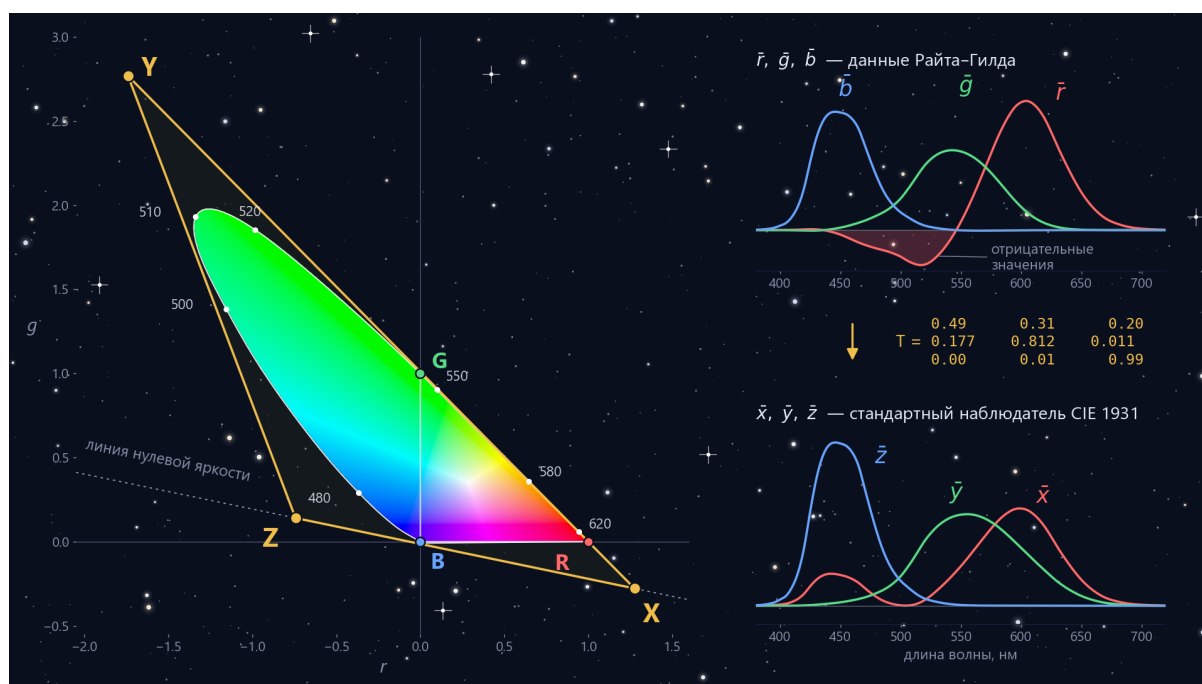
2. Выбрать из симулированных наблюдателей 10 категориальных наблюдателей по методу k-медоид, описанному в статье.
3. Визуализировать функции цветового соответствия (CMF) полученных категориальных наблюдателей.
4. * Проверить качество полученных наблюдателей для задачи наблюдательного метамеризма, воспроизведя (полностью или частично) эксперименты из глав 4-5 статьи.

Вопросы:

1. Какие параметры человеческого зрения учитывает модель Асано?
2. Для чего понадобились модели категориальных наблюдателей, если существуют модели стандартного наблюдателя (например, CIE 1931)?
3. Почему для выбора категориальных наблюдателей использовался метод k-медоид, а не стандартный k-средних?

Контакт для связи: @korch_sergey (Корчагин Сергей).

Задача 9. Одна ночь в Кембридже, изменившая мир колориметрии



Функции цветового соответствия (Color Matching Functions, CMF) $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$ стандартного наблюдателя CIE 1931 служат фундаментом почти всей колориметрии. На них опираются цветовые пространства XYZ, sRGB и CIELAB, калибровка мониторов и работа спектрофотометров. Между тем напрямую эти функции никто никогда не измерял. В конце 1920-х годов У. Д. Райт и Дж. Гилд провели серию визуальных экспериментов. Наблюдатель подбирал смесь трёх монохроматических излучений (700, 546,1 и 435,8 нм), неотличимую по цвету от тестового излучения. Так были получены функции $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, $\bar{b}(\lambda)$, которые на некоторых участках спектра принимают отрицательные значения.

Накануне заседания CIE 18 сентября 1931 года Смит, Гилд и Прист, по преданию, почти всю ночь пересчитывали эти данные в новую систему XYZ с мнимыми основными цветами. Новая система должна была удовлетворять ряду исходных принципов: подчиняться законам Грассмана, воспроизводить функцией $\bar{y}(\lambda)$ кривую относительной спектральной световой эффективности $V(\lambda)$ 1924 года, не содержать отрицательных значений, давать равноэнергетическому спектру равные координаты цветности и обращать $\bar{z}(\lambda)$ в ноль на возможно большем участке спектра. Числа, выбранные той ночью, остаются в употреблении уже почти столетие.

Вам предстоит разобрать статью «[How the CIE 1931 Color-Matching Functions Were Derived from Wright–Guild Data](#)» и, следуя ей, самостоятельно вывести функции CIE 1931 из исходных данных Райта–Гилда.

Задание:

1. Загрузить функции цветового соответствия Райта–Гилда $\bar{r}(\lambda)$, $\bar{g}(\lambda)$, $\bar{b}(\lambda)$ (CIE 1931 RGB) и функцию относительной спектральной световой эффективности $V(\lambda)$ CIE 1924. В библиотеке colour-science они доступны как colour.MSDS_CMFS['Wright & Guild 1931 2 Degree RGB CMFs'] и colour.colorimetry.SDS_LEFS_PHOTOPIC['CIE 1924 Photopic Standard Observer']. Построить графики функций и спектральный locus на диаграмме цветности rg (рис. 2 статьи).
2. Найти методом наименьших квадратов (формулы A13–A14) коэффициенты яркости основных цветов R, G, B и сопоставить их с принятыми в 1931 году (1 : 4,5907 : 0,0601). Вывести уравнение линии нулевой яркости (в статье она называется alychne, формула 13) и нанести её на диаграмму rg.
3. Руководствуясь исходными принципами, построить на диаграмме rg треугольник основных цветов X, Y, Z (рис. 3). Найти координаты (r, g) вершин и сравнить их со значениями из статьи (формулы 15–18).
4. Вычислить матрицу перехода от RGB к XYZ (формулы 19–20), нормировав её так, чтобы равноэнергетический стимул переходил в точку (1, 1, 1).

5. Получить функции $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, $\bar{z}(\lambda)$, выполнить их окончательную коррекцию по $V(\lambda)$ (формулы 21 или 25–27) и воспроизвести рис. 4–5 статьи. Сопоставить результат с официальными CMF CIE 1931 (colour.MSDS_CMFS['CIE 1931 2 Degree Standard Observer']) на графиках и по величине максимального отклонения.

* Выяснить, насколько итог зависит от произвольных решений, принятых в 1931 году:

1. Как меняются матрица и функции $\bar{x}$, $\bar{y}$, $\bar{z}$ при ином выборе стороны YZ или наклона стороны XY?
2. При каком наклоне функция $\bar{z}(\lambda)$ начинает принимать отрицательные значения?
3. Убедиться численно, что метамерная пара (два спектра разного состава с одинаковыми координатами RGB) остаётся метамерной в XYZ до коррекции по $V(\lambda)$ и может перестать быть таковой после неё.

Вопросы:

1. На каких исходных принципах строился переход от RGB к XYZ? Какие из них, по мнению авторов статьи, не были бы приняты, если бы систему создавали сегодня, и почему?
2. Почему у любой системы реальных, физически реализуемых основных цветов функции цветового соответствия неизбежно принимают отрицательные значения? Зачем ради неотрицательности пришлось вводить мнимые основные цвета X, Y, Z?
3. Зачем основные цвета X и Z помещены на линию нулевой яркости? Почему, как показывает эффект Гельмгольца–Кольрауша, яркость (luminance) плохо отражает субъективно воспринимаемую яркость (brightness)?
4. Масштабирование функций на $V(\lambda)/\bar{y}_0(\lambda)$ не меняет спектрального локуса на диаграмме цветности. Почему же при этом могут измениться предсказания метамерных равенств для немонахроматических стимулов?

Дополнительные материалы о том, как устроен эксперимент по измерению CMF:

1. [Цветовая вычислительная фотография. Часть 2: Стандарты CIE 1931.](#) Разделы «Эксперименты на людях», «CIE RGB» и «CIE XYZ» описывают эксперимент с разделённым полем зрения и переход от RGB к XYZ.

2. [История цветовой модели RGB](#). Статья прослеживает путь от опытов Гельмгольца до установки Гилда и объясняет, откуда в данных берутся отрицательные значения.
3. [Цвет: от шестнадцатеричных кодов до глаза](#). Раздел «Wright & Guild's color matching experiments», а также разделы о спектральном локусе и пространстве XYZ.

Контакт для связи: @korch_sergey (Корчагин Сергей).

Задача 10. Сбалансированная идентификация структурных математических моделей -

Представьте себя Иоганном Кеплером или Исааком Ньютоном, которые разглядывают астрономические схемы движения планет Солнечной системы и пытаются на основании этих (в тому же - неточных !!!) данных вывести закономерности орбитальной динамики. Вспомните насколько “просто” выглядит закон всемирного тяготения Ньютона, который вместе с (его же, Ньютона) 2-ым законом движения (в дифференциальной форме) позволяет неплохо моделировать движение небесных тел.

Эта “историческая аналогия” иллюстрирует типовую ситуацию: исследователь пытается вывести строгие математические закономерности поведения динамической системы на основании имеющихся у него неполных и неточных сведений о “положениях” этой системы в некоторые моменты времени. Если исследователь имеет основания полагать, что эти закономерности имеют вид дифференциального уравнения, то ему понадобится определить вид правой части такого уравнения. Именно эту проблему, иногда, называют обратной задачей, т.к. здесь не требуется рассчитать траекторию (решая “прямую” задачу Коши), а, наоборот, по набору точек на траектории определить сам диффуз.

Технология сбалансированной идентификации позволяет формализовать выбор подходящей математической модели, включающей различные уравнения, не только и не обязательно дифференциальные, но и произвольные соотношения с заранее неизвестными (!) функциями, подлежащими идентификации. Рабочее название технологии: SvF, Simplicity vs Fitting, поскольку применяемые алгоритмы используют идеи регуляризации (математический прием выражения интуитивной идеи “простоты”, “гладкости”, подбираемых функций), чтобы уравновесить тенденцию “подгонки” модели под имеющийся набор экспериментальных данных, содержащих ошибки измерений.

Основными публикациями по SvF-технологии являются

1. Sokolov A.V., Voloshinov V.V. Model Selection by Balanced Identification: the Interplay of Optimization and Distributed Computing // Open Computer Science, 2020, 10 — p. 283–295. <https://doi.org/10.1515/comp-2020-0116>

2. Соколов А.В., Волошинов, В.В. Выбор математической модели: баланс между сложностью и близостью к измерениям // International Journal of Open Information Technologies, 2018, 6(9) С. 33-41, <http://injoit.org/index.php/j1/article/view/612>

Технология активно развивается вместе со своей, свободно доступной, программной реализацией, <https://github.com/distcomp/SvF>, использующей программный инструментарий распределенных вычислений Everest, <http://everest.distcomp.org/>. Полный цикл расчетов можно запустить через Everest-приложение <https://optmod.distcomp.org/apps/vladimirv/svf-remote>.

По ссылке <https://github.com/distcomp/SvF/blob/main/svf-reference.md> приведен список публикаций по теме, включая результаты применения SvF-технологии в различных прикладных исследованиях (биологии, физике плазмы, метеорологии и т.д.).

Надо отметить, что SvF-технология, интересна еще как пример сочетания методов вычислительной математики, локальной и глобальной оптимизации, распределенных вычислений для организации трудоемких расчетов.

Задание. Попробуйте самостоятельно разобрать и рассказать пример идентификации моделей “маятник с трением”, представленный в двух вышеуказанных статьях.

Подумайте над тем, как можно практически решать возникающую там обратную задачу определения правой части диффура, где, фактически приходится иметь дело с композицией двух неизвестных функций: траектории маятника (как функции времени) и самой правой части (как функции траектории). Речь идет о замене исходной задаче с непрерывными функциями на их дискретизированные аппроксимации... Подсказка в статье

[Оптимизационные методы и кусочно линейные аппроксимации в обратных задачах с автономными дифференциальными уравнениями.](#)

Контакты для связи: <https://t.me/VladVol>, vladimir.voloshinov@gmail.com

Задача 11. QUBO-задачи дискретной оптимизации, некоторые свойства

Среди задач дискретной оптимизации выделяют задачи типа QUBO (Quadratic Unconstrained Binary Optimization), задачи квадратичной бинарной оптимизации без ограничений. В общем виде QUBO можно записать так

$$x^T M x = \sum_{i=1}^n \sum_{j=1}^n m_{ij} x_i x_j \rightarrow \min_{x \in B^n}, \text{ где} \\ B^n = \{0, 1\}^n = \{x \in \mathbb{R}^n : x_i \in \{0, 1\}, i=1:n\}. \quad (1)$$

Здесь M вещественная квадратная матрица с коэффициентами m_{ij} , x — n -вектор бинарных переменных x_i , принимающих только два значения 0, 1. Иными словами, минимум здесь ищется среди 2^n вершин единичного куба в n -мерном пространстве.

Задачи такого вида, в общем случае, относятся к классу NP-трудных задач, для которых не найдено эффективных алгоритмов нахождения точного оптимума. Многие другие известные задачи дискретной оптимизации, например, задачи о поиске в графе т. н. максимального разреза (MaxCut problem) можно записать в виде QUBO. Кстати верно и обратное, любую QUBO-задачу можно записать как MaxCut-задачу для некоторого графа, содержащего n вершин. Задачи QUBO имеют много особенностей, малую часть которых предлагается вам предлагается проверить.

Задание 1. Проверьте что целевую функцию в QUBO-задаче (1) можно немного упростить и заменить (1) на эквивалентную задачу (коэффициенты q_{ij} могут отличаться от m_{ij})

$$\sum_{i=1}^n q_{ii}x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^n q_{ij}x_i x_j = \sum_{i=1}^n q_{ii}x_i + \sum_{1 \leq i < j \leq n} q_{ij}x_i x_j \rightarrow \min_{x \in B^n}, \text{ где } B^n = \{0, 1\}^n = \{x \in \mathbb{R}^n : x_i \in \{0, 1\}, i=1:n\}. \quad (2)$$

Задание 2. Докажите, что если в задаче (2) убрать условие бинарности переменных и перейти к следующей задаче с непрерывными переменными, принимающими значения на отрезке $[0, 1]$:

$$\sum_{i=1}^n q_{ii}x_i + \sum_{1 \leq i < j \leq n} q_{ij}x_i x_j \rightarrow \min_{x \in C^n}, \text{ где } C^n = [0, 1]^n = \{x \in \mathbb{R}^n : x_i \in [0, 1], i=1:n\}. \quad (3)$$

то решения в задаче (3) совпадает с решением задачи (2) (в частности, достигается в какой-то вершине единичного n -мерного куба, возможно, что не в одной). Заметим, что это обстоятельство нисколько не «упрощает» QUBO-задачу, т. к. (3) является, в общем случае, примером «трудной» задачи *глобальной оптимизации*.

Контакты для связи: <https://t.me/VladVol>, vladimir.voloshinov@gmail.com

Задача 12. SDP-релаксация MaxCut-задачи (алгоритм Гоеманса-Уильямсона)

Задача о поиске, т.н. максимального разреза в графе, известная еще как MaxCut-задача, является одной из типовых NP-трудных задач дискретной оптимизации. Одна из формулировок выглядит следующим образом.

Рассмотрим неориентированный граф G , который характеризуется только набором длин ребер w_{ij} , соединяющих вершины с номерами i и j :

$$G = \{w_{ij} : 1 \leq i < j \leq N\}, w_{ij} \geq 0$$

Разрезом графа называется разбиение номеров его вершин на два множества, после чего вычисляется сумма длин ребер, соединяющих вершины из разных (!) множеств. Свяжем с каждой i -й вершиной графа переменную x_i , принимающую значение 1, если вершина входит в “первое” множество, и равную -1, если - во “второе”.

Задание 1. Проверьте, что решение следующей задачи и будет максимальным разрезом графа G .

$$\frac{1}{2} \sum_{1 \leq i < j \leq N} w_{ij} (1 - x_i x_j) \rightarrow \max_{x_i \in \{-1, 1\}} \text{ где } x_i \in \{-1, 1\} (i=1:N).$$

В общем случае (“распределения” значений длин ребер w_{ij}), для больших значений N , найти решение этой задачи очень непросто.

Задание 2. Проверьте, что MaxCut-задача, в некотором смысле эквивалентна некоторой QUBO-задаче (из Задачи 10 в этом списке). Это означает, что методы решения MaxCut задач можно переносить на QUBO и - наоборот.

В 1995 году был предложен (см. [1] в списке Ссылки) необычный подход к поиску приближенного решения MaxCut-задачи на основе ее замены на задачу оптимизации с непрерывными переменными и специальным типом ограничений в форме условия положительной полуопределенности квадратной матрицы, составленной из переменных задачи. Такие задачи принято называть SDP-задачами (Semidefinite Programming Problems).

Алгоритмическое значение замены MaxCut-задачи на SDP-задачу в том, что для SDP существуют и программно реализованы полиномиальные алгоритмы поиска точного решения. В работе [1] предлагается по найденному решению построенной SDP-задачи найти приближенное решение исходной MaxCut-задачи, причем, с гарантированной нижней оценкой неизвестного оптимального значения в дискретной задаче!

Если обозначить неизвестную величину максимального разреза через m , то алгоритм Гоеманса-Уильямсона (GW-алгоритм) дает длину разреза не меньше чем $\alpha \cdot m$, где $\alpha=0.87856$. При этом значение α не зависит от параметров графа G (числа вершин и длин ребер) !

ВАЖНО. Я не предлагаю студентам сейчас изучать алгоритмы решения SDP-задач. Для целей Задачи 12 достаточно трактовать SDP как некоторую эффективно разрешимую задачу оптимизации специального вида с непрерывными переменными.

Задание 3. Попробуйте разобраться с цепочкой замены переменных в GW-алгоритме. Вначале от исходных дискретных переменных к непрерывным переменным SDP-задачи:

1. дискретные переменные x_i со значениями $-1, +1 \Rightarrow$
2. непрерывные переменные $\mathbf{v}_i$ (N -мерные вектора единичной длины) $\Rightarrow$
3. непрерывные переменные y_{ij} (коэффициенты положительно полуопределенной матрицы)

Затем - в обратном “направлении” от SDP-переменных к MaxCut-переменным.

Задание 4. Разберитесь с тем, как авторы [1] получают гарантированную указанную выше нижнюю оценку оптимального значений в MaxCut-задаче. Обратите внимание на применяемую технику из теории вероятностей.

Задание 5*. Разберитесь, в чем заключается улучшение исходного варианта GW-алгоритма, предложенное в статье [2].

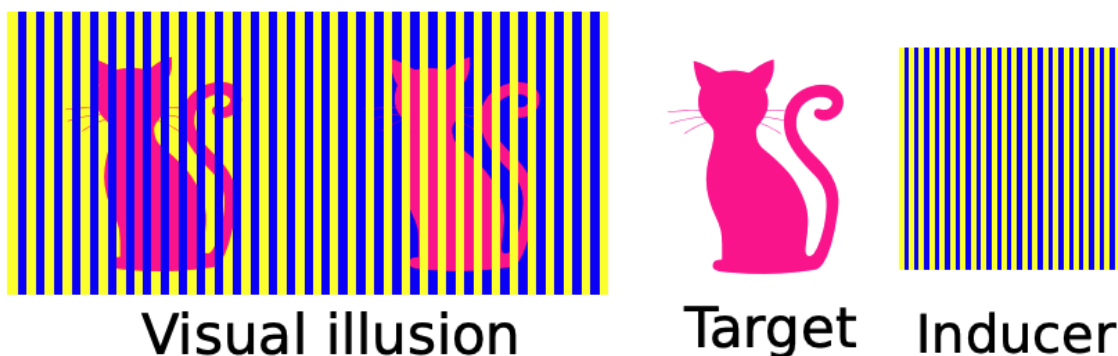
Ссылки

1. Michel X. Goemans and David P. Williamson. **Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming** // Journal of the ACM, 1995, 42(6), <https://dl.acm.org/doi/epdf/10.1145/227683.227684> (см. файл *goemans1995.pdf* в папке <https://disk.yandex.ru/d/izjIDJDQvnsGmq>)
2. Burer, S., Monteiro, R.D. and Zhang, Y. **Rank-Two Relaxation Heuristics for MAX-CUT and Other Binary Quadratic Programs** // SIAM Journal on Optimization, 2002, 12(2), 503-521. <https://doi.org/10.1137/S1052623400382467> (см. файл *RANK-TWO RELAXATION*.pdf* в папке <https://disk.yandex.ru/d/izjIDJDQvnsGmq>)

Контакты для связи: <https://t.me/VladVol>, vladimir.voloshinov@gmail.com

Задача 13. Обманывают ли цветовые иллюзии свёрточную сеть, обученную на естественных изображениях

Человек воспринимает цвет и яркость участка изображения в зависимости от окружения. Серый квадрат на тёмном фоне кажется светлее такого же квадрата на светлом фоне (одновременный контраст). На красном фоне серый кажется зеленоватым. Есть гипотеза, что такие иллюзии — побочный эффект адаптации зрительной системы к статистике естественных изображений. Тогда похожие эффекты должны возникать и у нейросетей, обученных на тех же изображениях.



В работе Gomez-Villa et al., “Convolutional Neural Networks Can Be Deceived by Visual Illusions”, CVPR 2019 несколько неглубоких CNN обучают на задачах низкоуровневого зрения (шумоподавление, удаление размытия) на естественных изображениях. Затем на вход подают стимулы классических иллюзий. Оказывается, выход сети смещает яркость целевой области в ту же сторону, что и восприятие человека.

Ссылка на статью.

https://openaccess.thecvf.com/content_CVPR_2019/papers/Gomez-Villa_Convolutional_Neural_Networks_Can_Be_Deceived_by_Visual_Illusions_CVPR_2019_paper.pdf

Продолжение про цветные иллюзии: Gomez-Villa et al., “Color illusions also deceive CNNs for low-level vision tasks: Analysis and implications”, Vision Research, 2020.

Задание

Разобрать статью и в упрощённом виде воспроизвести основной эксперимент.

- Обучить небольшую CNN (3–6 свёрточных слоёв) для шумоподавления на патчах естественных изображений: BSDS500, DIV2K, Kodak или любой другой небольшой набор фотографий.
- Написать генератор стимулов с известными пиксельными значениями: одновременный контраст (одинаковая серая цель на тёмном и светлом фоне) и иллюзия Уайта. Для каждого стимула сделать контрольную версию без иллюзии.
- Подать стимулы на вход обученной сети без шума и измерить смещение средней яркости цели на выходе относительно входа. Совпадает ли направление смещения с восприятием человека?
- Повторить для цветной версии: серая цель на красном, зелёном, синем и жёлтом фоне. Сдвигается ли цвет цели в сторону дополнительного к фону цвета? Сдвиг удобно измерять в CIELAB (библиотека [colour-science](https://github.com/bradlee/cvcolorscience)).
- Исследовать, как эффект зависит от размера цели и от рецептивного поля сети (число слоёв). Сравнить с необученной сетью той же архитектуры.

Точное совпадение с числами авторов не требуется. Важно описать постановку, объяснить упрощения и интерпретировать результат.

Вопросы

- Почему сеть, которую никто не учил видеть иллюзии, может их воспроизводить?
- Чем одновременный контраст отличается от ассимиляции (иллюзии Уайта)? Объясняет ли один и тот же механизм оба эффекта?
- Почему важно сравнение с необученной сетью и с контрольными стимулами?
- Что эксперимент говорит о зрении человека, а чего не говорит?

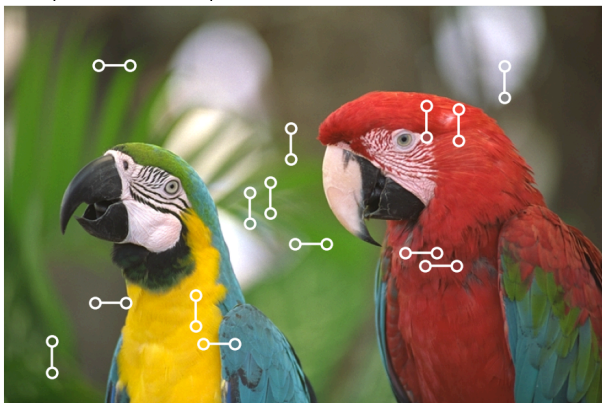
Задача выполняется на бесплатном Google Colab или Kaggle Notebook (GPU).

Контакт для связи: https://t.me/lobarev_e

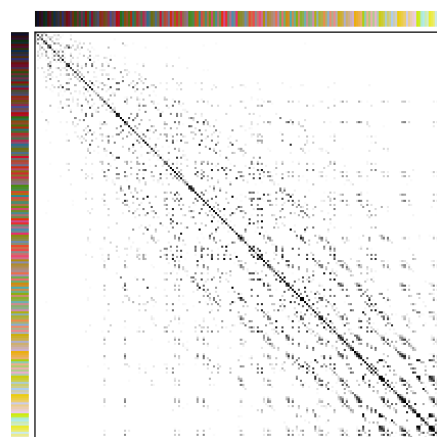
Задача 14. Метрика цветоразличения из статистики естественных изображений

Цветовые пространства вроде CIELAB и CAM16-UCS построены по психофизическим измерениям: расстояние в них приближает различимость цветов человеком. Есть гипотеза, что восприятие цвета адаптировано к статистике естественных сцен. Если так, то похожая геометрия должна получаться прямо из изображений, без участия человека.

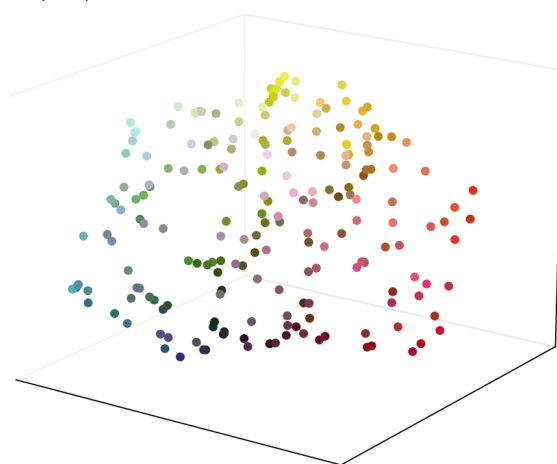
1. Пары пикселей на расстоянии ϵ



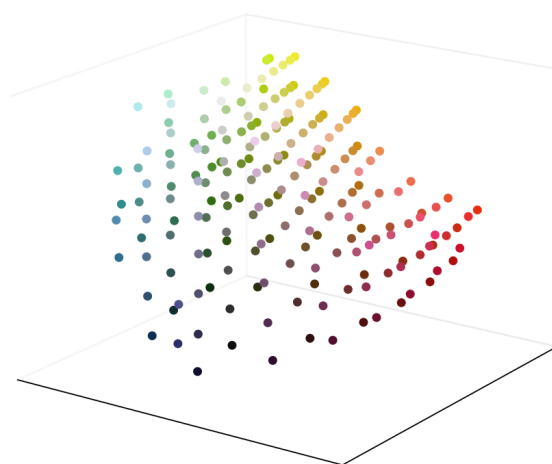
2. Матрица совместной встречаемости (PPMI)



3. Пространство из статистики



4. CAM16-UCS (по людям)



В обработке языка похожая идея лежит в основе word2vec: слова похожи, если встречаются в похожих контекстах. Levy & Goldberg (“Neural Word Embedding as Implicit Matrix Factorization”, NeurIPS 2014) показали, что word2vec эквивалентен факторизации матрицы точечной взаимной информации (PMI) слов и контекстов. В задаче та же идея применяется к цветам. Роль «слов» играют квантованные цвета, роль «контекста» — цвета пикселей на расстоянии ϵ .

Литература по статистике цвета в природных сценах (стоит прочитать хотя бы одну):

- Ruderman, Cronin, Chiao, “Statistics of cone responses to natural images”, JOSA A, 1998;
- Long, Yang, Purves, “Spectral statistics in natural scenes predict hue, saturation, and brightness”, PNAS, 2006;
- Yendrikhovskij, “Computing color categories from statistics of natural images”, J. Imaging Sci. Technol., 2001.

Задание

- Выбрать набор естественных изображений без сжатия с потерями: Kodak, DIV2K или аналог. Квантовать цвета на равномерной сетке в sRGB (например, 16^3 ячеек, далее оставить только встречающиеся).
- Для нескольких значений ϵ (например, 1, 2, 4, 8, 16, 32 пикселя) посчитать матрицу совместной встречаемости $C_\epsilon[i,j]$: сколько раз цвета i и j встречаются в парах пикселей на расстоянии ϵ . Для скорости пары можно сэмплировать.
- Превратить C_ϵ в PPMI-матрицу и построить две метрики:
 - (а) первого порядка: расстояние из PMI между цветами;
 - (б) второго порядка: косинусное расстояние между строками PPMI или между их SVD-эмбедингами (аналог word2vec).
- Так как сходство надёжно только для близких цветов, восстановить глобальные расстояния через kNN-граф и кратчайшие пути (Isomap).
- Сравнить полученные метрики с евклидовым расстоянием в sRGB и с расстоянием в CAM16-UCS ([colour-science](https://www.colour-science.org/)):
 - ранговая корреляция попарных расстояний отдельно для близких и для далёких пар;
 - визуализация трёхмерного вложения (MDS/Isomap): видны ли ось светлоты и оппонентные оси?
- Исследовать, как результат зависит от ϵ и шага квантования.

Вопросы

- Почему сырая матрица встречаемости плохо подходит как мера сходства и зачем нужна PMI?
- Чем отличаются метрики первого и второго порядка? Какая из них ближе к word2vec?
- Какие эффекты определяют метрику при малом и при большом ϵ : камера, освещение, объекты, композиция сцены?
- Как тени и неравномерное освещение влияют на то, какие изменения цвета метрика считает «близкими»? Что это значит для светлоты по сравнению с цветностью?
- К какой метрике ближе статистическая, к sRGB или к CAM16-UCS, и в каком диапазоне расстояний?

Задача выполняется на бесплатном Google Colab, GPU не требуется.

Контакт для связи: https://t.me/lobarev_e