

Использование среднего степенного для оценивания качества изображений

А.А. Нечаев

Аннотация — В статье приведены результаты исследования применения среднего степенного значения для оценки качества изображений. Объектом исследования в этой работе являлись коэффициенты линейной корреляции между значениями оценок качества изображений и экспертными оценками качества. Рассматриваются эталонные оценки качества. В качестве исследуемых оценок используются значения среднего степенного с различными параметрами. Принцип исследования заключается в использовании базы изображений TID2013 и экспертных оценок качества изображений для выявления наиболее коррелированных с экспертными оценками объективных оценок и использование множественной регрессии для повышения корреляции. Для проведения вычислительных экспериментов была использована среда разработки Embarcadero RAD Studio, для выполнения регрессии и оценки полученных результатов был использован язык программирования Python. Результаты исследования могут быть использованы для объективного оценивания качества искажённых изображений.

Ключевые слова — качество изображений, множественная регрессия, среднее степенное, эталонные оценки.

I. ВВЕДЕНИЕ

Измерение качества изображения является важным для большинства программ обработки изображений. В целом, метрика качества изображения имеет три вида применений.

Первое, она может быть использована для мониторинга качества изображений для систем контроля качества. Второе, она может быть применена для тестирования систем обработки изображений. Третье, она может быть включена в систему обработки изображений для оптимизации алгоритмов и настройки параметров [1].

Цифровые изображения подвержены большому множеству различных искажений во время обработки, сжатия, хранения, передачи и воспроизведения [2]. Например, потеря информации происходит при сжатии по алгоритмам JPEG и JPEG2000. Для программ, в которых в итоге изображения будут просмотрены людьми, единственным “правильным” методом оценивания визуального качества изображения является субъективное оценивание. Однако на практике субъективное оценивание является неудобным, дорогим и требующим много времени. Целью исследования

является нахождение такого объективного метода оценивания качества изображения, оценки которого будут максимально близки к субъективным.

Методы оценки качества изображений делятся на две группы: эталонные и неэталонные. Методы первой группы основаны на сравнении искажённого изображения с исходным. Такие методы применяются в тех случаях, когда доступно исходное изображение (не имеющее искажений). Вторая группа методов основана на вычислении таких характеристик цифрового изображения, как яркость, контрастность, резкость [3] с использованием исключительно искажённых изображений. Исследуемый в этой работе метод (основанный на вычислении средних степенных значений) относится к эталонным, так как использует как исходное, так и искажённое изображения.

II. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДОВ ОЦЕНКИ КАЧЕСТВА ИЗОБРАЖЕНИЙ

A. Mean Squared Error

Среднеквадратическая ошибка (Mean Squared Error, MSE) является основной количественной метрикой качества в области обработки сигналов более 50 лет, она остаётся стандартным критерием качества сигнала [4]. MSE вычисляется по формуле (1):

$$MSE(x, y) = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \quad (1)$$

где $x = \{x_i | i=1, 2, \dots, N\}$ и $y = \{y_i | i=1, 2, \dots, N\}$ – два дискретных сигнала конечной длины N . Применительно к изображениям x и y – это матрицы пикселей, а N равно произведению ширины на высоту.

К достоинствам метрики MSE относится простота формулы, отсутствие параметров, быстрота вычисления (на каждый отсчёт требуется одно умножение и два сложения). Для вычисления MSE не требуется хранить большое количество промежуточных результатов, что делает эту метрику нетребовательной к объёму оперативной памяти устройства. Недостатком MSE является то, что эта метрика не соответствует восприятию изображений человеком [5].

B. Signal-to-noise ratio

Signal-to-noise ratio (отношение сигнал-шум, ОСШ) – безразмерная величина, равная отношению мощности полезного сигнала к мощности шума [6]. ОСШ равно квадрату отношения амплитуды полезного сигнала к амплитуде шума. Обычно ОСШ измеряется в логарифмическом масштабе, тогда ОСШ в децибелах

равно двадцати логарифмам по основанию 10 от частного амплитуды полезного сигнала и амплитуды шума. Применительно к изображениям (например, в случае оценки качества цветопередачи камер) амплитудой полезного сигнала считается среднее арифметическое значений пикселей, а амплитудой шума – среднеквадратическое отклонение значений пикселей от амплитуды сигнала. ОСШ вычисляется по формулам (2.1-2.4):

$$A_c = \frac{1}{N} \sum_{i=1}^N L_i \quad (2.1)$$

$$A_{ш} = \sqrt{\frac{\sum_{i=1}^N (L_i - A_c)^2}{N}} \quad (2.2)$$

$$ОСШ = \left(\frac{A_c}{A_{ш}} \right)^2 \quad (2.3)$$

$$ОСШ_{дБ} = 20 \log \left(\frac{A_c}{A_{ш}} \right) \quad (2.4)$$

где L_i – яркость пикселя i , N – количество пикселей, A_c – амплитуда сигнала, $A_{ш}$ – амплитуда шума. Метрика ОСШ использует только один сигнал (а не исходный и искажённый, как это было с MSE), поэтому при наличии исходного сигнала можно вычислять значения ОСШ (обычные или логарифмические) для обоих изображений. Значения ОСШ слабо коррелируют с субъективными оценками качества, даваемыми человеком. К недостаткам ОСШ можно отнести и деление на ноль, возникающее при равенстве амплитуды шума нулю (то есть при равенстве всех пикселей изображения). Для предотвращения деления на ноль можно к амплитуде шума добавить небольшую положительную константу.

C. Peak signal-to-noise ratio

Peak signal-to-noise ratio (Пиковое отношение сигнал/шум, PSNR) используется для измерения различий между двумя изображениями. Более высокое значение PSNR соответствует лучшему качеству изображения [7]. В случае кодирования каждого пикселя восемью битами значение PSNR вычисляется по формуле (3):

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \quad (3)$$

где MSE – среднеквадратическая ошибка между пикселями исходного и искажённого изображений, вычисляемая в соответствии с (1). В более общем случае в (3) вместо 255 используется число, равное наибольшему беззнаковому целому числу, которое можно записать с использованием того количества бит, которое используется для кодирования каждого пикселя.

Чем меньше различия между исходным и искажённым изображениями, тем меньше значение MSE, тем больше значение PSNR. Технически значение PSNR для двух одинаковых изображений равно бесконечности. Прибавление небольшой положительной константы к квадратному корню из MSE предотвратит деление на ноль в случае вычисления PSNR для одинаковых изображений.

D. Universal Image Quality Index

Универсальный индекс качества изображения (Universal Image Quality Index, UIQI) моделирует любые искажения как комбинацию трёх разных факторов: потеря корреляции, искажение яркости и искажение контраста [8]. Динамическим диапазоном значений UIQI является диапазон [-1;1]. Лучшее значение (равное единице) достигается тогда и только тогда, когда $x_i=y_i$ для каждого значения i в диапазоне [1..N] (где x_i – пиксели исходного изображения, y_i – пиксели искажённого изображения, N – количество пикселей в каждом из этих изображений). Значение UIQI вычисляется по формулам (4.1-4.6):

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (4.1)$$

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (4.2)$$

$$\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (4.3)$$

$$\sigma_y^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})^2 \quad (4.4)$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}) \quad (4.5)$$

$$UIQI = \frac{\sigma_{xy}}{\sigma_x \sigma_y} * \frac{2 * \bar{x} * \bar{y}}{(\bar{x})^2 + (\bar{y})^2} * \frac{2 \sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \quad (4.6)$$

В (4.6) значение UIQI записано как произведение трёх компонентов. Первый компонент – это коэффициент корреляции между x и y , который измеряет степень линейной зависимости между исходным и искажённым изображениями. Значения первого компонента находятся в диапазоне [-1;1], значение 1 достигается, если $y_i=a*x_i+b$, где $1 \leq i \leq N$, $a=const$, $b=const$, $a>0$. Даже если между x и y линейная зависимость, искажения могут быть оценены вторым и третьим компонентами. Второй компонент имеет диапазон значений [0;1] и измеряет, насколько близка средняя яркость между x и y . Значение второго компонента равно единице тогда и только тогда, когда средние значения x и y , вычисляемые в соответствии с (4.1) и (4.2), равны между собой. σ_x и σ_y могут рассматриваться как ожидание контраста x и y . Третий компонент измеряет, насколько близки контрасты изображений. Диапазон третьего компонента: [0;1], значение третьего компонента равно 1 тогда и только тогда, когда $\sigma_x=\sigma_y$.

Помимо измерения значения UIQI для всего изображения целиком, можно вычислять значения UIQI для отдельных квадратных окон изображения (например, размером 8*8 пикселей), а потом вычислять итоговое значение как среднее арифметическое значений UIQI для каждого такого окна.

Достоинством UIQI (по сравнению с MSE) является значительно более высокая корреляция оценок этой метрики с субъективными оценками качества изображения. Недостатком UIQI является намного большая вычислительная сложность. При вычислении MSE для каждого отсчёта требовалось по два сложения и одному умножению. При вычислении UIQI в соответствии с (4.1-4.5) для каждого отсчёта требуется выполнить по девять сложений и три умножения. При вычислении UIQI для каждого окна по отдельности это окно каждый раз сдвигается на 1 пиксель по

горизонталы, а после окончания строки – на 1 пиксель по вертикали. В результате для изображения шириной w и высотой h с размером окна $m*m$ количество окон равно $(w-m+1)*(h-m+1)$. При значениях w и h порядка нескольких сотен или тысяч (разрешение изображений в большинстве случаев) и значениях m порядка единиц количество окон становится приблизительно равным количеству пикселей изображения. Например, если изображение имеет размер 1280*720 (HD), а размер окна равен 8*8 пикселей (такой размер используется в [8]), то количество пикселей равно 1280*720=921600, а количество окон размером 8*8 равно $(1280-8+1)*(720-8+1)=1273*713=907649$, что составляет примерно 0.9848 от количества пикселей изображения. При этом для каждого окна необходимо вычислить значения в соответствии с (4.1-4.5), в результате чего количество арифметических операций становится приблизительно прямопропорционально квадрату размера окна (то есть если для некоторого изображения при вычислении UIQI с использованием окна размером 8*8 требуется X операций, то при вычислении UIQI с использованием окна размером 16*16 потребуется $4X$ операций). Такая зависимость ограничивает возможность выбирать размер окна произвольным образом.

Другим недостатком UIQI является возможность возникновения деления на ноль. Деление на ноль при вычислении UIQI может возникнуть в двух случаях. Первый случай, если средние значения x и y , вычисляемые в соответствии с (4.1) и (4.2), одновременно равны нулю (в таком случае знаменатель второго компонента в (4.6) равен нулю). Для изображений в оттенках серого это означает два полностью чёрных изображения. Числитель второго компонента в таком случае тоже будет равен нулю. Деление нуля на ноль – это NaN (Not a Number, не число) в соответствии со стандартом IEEE Std 754-2019 [9]. Второй случай, если $\sigma_x^2=0$ и $\sigma_y^2=0$, тогда знаменатель третьего компонента равен нулю. Для изображений это означает, что каждое значение x_i равно среднему x , каждое значение y_i равно среднему y (то есть все пиксели исходного изображения равны между собой и все пиксели искажённого изображения равны между собой, эти изображения представляют одноцветные прямоугольники любого цвета). Таким образом, первый случай – это частный случай второго случая (оба изображения являются одноцветными прямоугольниками, но при этом ещё и y каждого чёрный цвет). В случае равенства нулю только одного из выражений $\{\sigma_x^2; \sigma_y^2\}$ нулю будет равен знаменатель первого компонента, но его можно сократить с числителем третьего компонента, поэтому деления на ноль не будет.

Недостаток метрики UIQI, связанный с возникновением ситуаций, приводящих к делению на ноль, для всего изображения проявляется редко (так как для этого требуется, чтобы оба изображения были абсолютно одноцветными прямоугольниками). Однако при вычислении UIQI по отдельности для каждого окна размером $m*m$ пикселей такие ситуации возникают со значительно большей частотой. В таком случае для

возникновения ошибки деления на ноль (при вычислении UIQI по отдельности для каждого окна) достаточно, чтобы во всём изображении был хотя бы один квадрат размером $m*m$, состоящий из пикселей одного цвета. Если изображение имеет несколько цветовых компонент (например, RGB), то для возникновения ошибки достаточно одного одноцветного квадрата размером $m*m$ в любом из цветовых компонент.

E. Structural Similarity Index

Индекс структурного сходства (Structural Similarity, SSIM) является объективной эталонной метрикой качества изображений. SSIM учитывает сходства яркости, контраста и структуры исходного и искажённого изображений. Индекс SSIM вычисляется по формулам (5.1-5.7):

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (5.1)$$

$$\mu_y = \frac{1}{N} \sum_{i=1}^N y_i \quad (5.2)$$

$$\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{1/2} \quad (5.3)$$

$$\sigma_y = \left(\frac{1}{N-1} \sum_{i=1}^N (y_i - \mu_y)^2 \right)^{1/2} \quad (5.4)$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (5.5)$$

$$C_1 = (K_1 L)^2 \quad (5.6)$$

$$C_2 = (K_2 L)^2 \quad (5.7)$$

$$C_3 = C_2/2 \quad (5.8)$$

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (5.9)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (5.10)$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (5.11)$$

$$SSIM(x, y) = [l(x, y)]^\alpha * [c(x, y)]^\beta * [s(x, y)]^\gamma \quad (5.12)$$

где x_i – пиксели исходного изображения, y_i – пиксели искажённого изображения, N – количество пикселей, L – динамический диапазон значений пикселей ($L=255$ для изображений в оттенках серого с глубиной цвета 8 бит на пиксель), $K_1=0.01$, $K_2=0.03$ (значения K_1 и K_2 могут быть и другими, но каждое из них должно быть намного меньше единицы), $l(x, y)$ – функция сравнения яркости (luminance comparison function), $c(x, y)$ – функция сравнения контраста (contrast comparison function), $s(x, y)$ – функция сравнения структуры (structure comparison function). $\alpha>0$, $\beta>0$, $\gamma>0$ – параметры, используемые для регулирования относительной важности трёх компонентов (то есть значений функций сравнения яркости, контраста и структуры соответственно). Значение SSIM всегда находится в диапазоне $[-1; 1]$, при этом $SSIM(x, y)=1$ тогда и только тогда, когда каждый пиксель изображения y равен соответствующему пикселю изображения x .

Достоинством метрики SSIM является высокая корреляция её результатов с субъективными оценками качества изображений. Ещё одним достоинством (по сравнению с UIQI) является невозможность возникновения деления на ноль в результате добавления констант C_1 , C_2 и C_3 к числителям и знаменателям функций сравнения яркости, контраста и структуры.

Сравнительный анализ существующих эталонных метрик оценки качества изображений показал достоинства и недостатки различных метрик. Выбор метрики для каждой конкретной задачи зависит от условий именно этой задачи.

III. ИССЛЕДОВАНИЕ ИСПОЛЬЗОВАНИЯ СРЕДНЕГО СТЕПЕННОГО

Исследуемыми в этой работе метриками эталонной оценки качества изображений являются средние степенные значения для пикселей исходного файла, пикселей искажённого файла и модулей различий между соответствующими пикселями исходного и искажённого файлов. Среднее степенное значение [10] вычисляется по формуле (6.1):

$$M_k = \sqrt[k]{\frac{1}{N} \sum_{i=1}^N x_i^k} \quad (6.1)$$

где N – количество чисел, x_i – неотрицательные числа (для которых вычисляется среднее степенное значение), k – порядок среднего степенного, M_k – значение среднего степенного. Значение M_k принадлежит диапазону $[\min; \max]$, где \min – минимальное число из всех x_i , \max – максимальное число из всех x_i . В данной работе предлагается в качестве x_i принимать числа из трёх наборов. *Первый* – значения пикселей исходного изображения. *Второй* – значения пикселей искажённого изображения. *Третий* – модули разности между значениями соответствующих пикселей исходного и искажённого изображений. При $k=2$ значение среднего степенного для модулей разности соответствующих пикселей исходного и искажённого изображений равно квадратному корню из MSE (1), поэтому средние степенные значения можно рассматривать как обобщение MSE на произвольную степень k .

Для исследования используется следующий метод. В качестве исходных и искажённых изображений используются изображения базы TID2013. Эта база содержит 25 исходных изображений и 3000 искажённых изображений [11]. Все изображения этой базы имеют формат BMP с глубиной цвета 24 бита на пиксель, шириной 512 пикселей и высотой 384 пикселя. В этой базе есть изображения с искажениями 24 типов (аддитивный Гауссов шум, аддитивный шум в цветовых компонентах, высокочастотный шум, размытие по Гауссу, сжатие JPEG, сжатие JPEG2000, изменение контраста, изменение насыщенности цвета, мультипликативный Гауссов шум и другие). Субъективной оценкой качества каждого искажённого изображения базы TID2013 является MOS. MOS (mean opinion score) – это средняя экспертная оценка, которая представляет собой просто среднее арифметическое значение оценок (голосов), полученных для соответствующего сценария использования. Задачей алгоритмов объективного измерения качества является прогнозирование значения MOS, которое будет получено при заданном входном сигнале в субъективном эксперименте [12]. Применительно к этой работе входными сигналами являются изображения.

Для проведения исследования все искажённые изображения базы TID2013 разделены на две части: обучающую выборку (training set) и тестовую выборку (testing set). Чаще всего для обучающей выборки используется от 60% до 80% данных [13]. В этом исследовании для обучающей выборки используется 80% (2400 изображений из 3000), соответственно для тестовой выборки остаётся 20% (600 изображений из 3000). Для каждого из 3000 изображений вычисляются значения среднего степенного для пикселей исходного изображения, пикселей искажённого изображения и модулей разности между соответствующими пикселями исходного и искажённого. Эти значения вычисляются для различных значений параметра k в (6.1). Так как изображения базы TID2013 имеют глубину цвета 24 бита на пиксель, цвет каждого пикселя кодируется тремя значениями (компонентами цветового пространства RGB): красного (Red), синего (Blue) и зелёного (Green). В этом исследовании значения среднего степенного вычисляются по отдельности для каждого из этих компонентов. Так как заранее неизвестно, в каком из цветовых пространств будут получены наилучшие результаты, помимо RGB исследование проводится и для некоторых других пространств (всего в 10 цветовых пространствах, в каждом из которых цвет состоит из трёх компонент). Для удобства каждому из используемых цветовых пространств присвоен идентификатор $cs_1, cs_2, \dots, cs_{10}$ (сокращение от “color space”). У некоторых цветовых пространств (например, XYZ) есть разные определения в разных источниках, поэтому при использовании нескольких из них будем называть их XYZ1, XYZ2 и т.д. Так как все используемые в этой работе цветовые пространства состоят из трёх компонентов, для удобства эти компоненты обозначены col_1, col_2, col_3 (сокращение от слова “color”). Разные компоненты цветовых пространств имеют разные диапазоны значений. Например, в RGB у каждого из компонентов диапазон значений $[0; 255]$ (результат квантования непрерывного значения оттенка по 256 уровням). В CIELAB компоненты a и b могут принимать отрицательные значения [14]. Для удобства последующей обработки результатов для значений всех цветовых компонент каждого из используемых цветовых пространств после преобразования выполняется нормализация к диапазону $[0; 1]$. Для этого выполнено преобразование всех квантованных по уровням 0-255 цветов пространства RGB в используемое цветовое пространство, для каждого из компонентов нового цветового пространства определены его минимальное и максимальное значения, из результатов вычтен минимум и итог поделён на разность максимума и минимума.

Первым используемым цветовым пространством (обозначенным как cs_1) является RGB. В этом цветовом пространстве хранятся цвета пикселей в файлах формата BMP с глубиной цвета 24 бита на пиксель. Для нормализации цветов этого пространства к диапазону $[0; 1]$ выполняется деление значения каждого компонента на 255. В RGB обозначим R как col_1 , G как col_2 , B как col_3 .

Для следующего цветового пространства (XYZ1) используется обозначение cs2. Преобразование в это цветовое пространство [15] выполняется по формулам (7.1-7.3):

$$X=0.49000*R+0.31000*G+0.20000*B \quad (7.1)$$

$$Y=0.17697*R+0.81240*G+0.01063*B \quad (7.2)$$

$$Z=0.00000*R+0.01000*G+0.99000*B \quad (7.3)$$

где R, G, B – значения цвета в координатах RGB, X, Y, Z – значения цвета в координатах XYZ1. В XYZ1 обозначим X как col1, Y как col2, Z как col3.

Следующее цветовое пространство (XYZ2) обозначим как cs3. Преобразование из XYZ2 в RGB выполняется [16] по формулам (8.1-8.3), из которых решением системы линейных алгебраических уравнений следуют формулы (8.4-8.6), позволяющие выполнить преобразование из RGB в XYZ2:

$$R=1.716651*X-0.355671*Y-0.253366*Z \quad (8.1)$$

$$G=-0.666684*X+1.616481*Y+0.015769*Z \quad (8.2)$$

$$B=0.017640*X-0.042771*Y+0.942103*Z \quad (8.3)$$

$$X=0.168881*B+0.144617*G+0.636958*R \quad (8.4)$$

$$Y=0.059302*B+0.677998*G+0.262700*R \quad (8.5)$$

$$Z=1.060985*B+0.028073*G+0.000000*R \quad (8.6)$$

где R, G, B – значения компонентов цвета в пространстве RGB, X, Y, Z – значения компонентов цвета в пространстве XYZ2. В XYZ2 обозначим X как col1, Y как col2, Z как col3.

Следующее цветовое пространство (XYZ3) обозначим как cs4. Преобразование из RGB в XYZ3 выполняется [17] по формулам (9.1-9.3):

$$X=0.675*R+0.220*G+0.130*B \quad (9.1)$$

$$Y=0.325*R+0.680*G+0.080*B \quad (9.2)$$

$$Z=0.000*R+0.100*G+0.790*B \quad (9.3)$$

где R, G, B – координаты цвета в пространстве RGB, X, Y, Z – координаты цвета в пространстве XYZ3. В XYZ3 обозначим X как col1, Y как col2, Z как col3.

Следующее цветовое пространство (XYZ4) обозначим как cs5. Преобразование из RGB в XYZ4 выполняется [18] по формулам (10.1-10.10):

$$R'_{sRGB} = R/255 \quad (10.1)$$

$$G'_{sRGB} = G/255 \quad (10.2)$$

$$B'_{sRGB} = B/255 \quad (10.3)$$

$$\text{Если } R'_{sRGB} \leq 0.04045, \text{ то } R_{sRGB} = R'_{sRGB}/12.92 \quad (10.4)$$

$$\text{Если } G'_{sRGB} \leq 0.04045, \text{ то } G_{sRGB} = G'_{sRGB}/12.92 \quad (10.5)$$

$$\text{Если } B'_{sRGB} \leq 0.04045, \text{ то } B_{sRGB} = B'_{sRGB}/12.92 \quad (10.6)$$

$$\text{Если } R'_{sRGB} > 0.04045, \text{ то } R_{sRGB} = \left[\frac{R'_{sRGB}+0.055}{1.055} \right]^{2.4} \quad (10.7)$$

$$\text{Если } G'_{sRGB} > 0.04045, \text{ то } G_{sRGB} = \left[\frac{G'_{sRGB}+0.055}{1.055} \right]^{2.4} \quad (10.8)$$

$$\text{Если } B'_{sRGB} > 0.04045, \text{ то } B_{sRGB} = \left[\frac{B'_{sRGB}+0.055}{1.055} \right]^{2.4} \quad (10.9)$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R_{sRGB} \\ G_{sRGB} \\ B_{sRGB} \end{bmatrix} \quad (10.10)$$

где R, G, B – значения в пространстве RGB в

диапазоне [0;255]; R'_{sRGB} , G'_{sRGB} , B'_{sRGB} – значения в пространстве RGB в диапазоне [0;1]; R_{sRGB} , G_{sRGB} , B_{sRGB} – значения после нелинейного преобразования; X, Y, Z – значения в пространстве XYZ4. В XYZ4 обозначим X как col1, Y как col2, Z как col3.

Следующее цветовое пространство (YCbCr) обозначим как cs6. Пространство YCbCr используется при сжатии изображений по алгоритму JPEG. Преобразование из RGB в YCbCr выполняется [19] по формулам (11.1-11.3):

$$Y=0.299*R+0.587*G+0.114*B \quad (11.1)$$

$$Cb=-0.1687*R-0.3313*G+0.5*B+128 \quad (11.2)$$

$$Cr=0.5*R-0.4187*G-0.0813*B+128 \quad (11.3)$$

где R, G, B – значения в пространстве RGB; Y, Cb, Cr – значения в пространстве YCbCr. Y – яркость, Cb и Cr – цветоразностные компоненты. В YCbCr обозначим Y как col1, Cb как col2, Cr как col3.

Следующее цветовое пространство – YCoCg (это цветовое пространство предложено как альтернатива YCbCr для сжатия изображений) обозначим как cs7. Преобразование из RGB в YCoCg выполняется [20] по формуле (12):

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 0 & -1/2 \\ -1/4 & 1/2 & -1/4 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (12)$$

где R, G, B – значения в пространстве RGB, Y, Co, Cg – значения в пространстве YCoCg. В YCoCg обозначим Y как col1, Co как col2, Cg как col3.

Следующее цветовое пространство – HSI, обозначим его как cs8. Преобразование из RGB в HSI выполняется [21] по формулам (13.1-13.3). Однако выполнение этого преобразования в соответствии с (13.1-13.3) может привести к возникновению деления на ноль. Для предотвращения этого в этом исследовании вместо (13.2-13.3) используются формулы (13.4-13.5):

$$int = \frac{(R+G+B)}{3} \quad (13.1)$$

$$sat = 1 - \frac{\min(R,G,B)}{int} \quad (13.2)$$

$$hue = \arctan \left[\frac{\sqrt{3}(G-B)}{(R-G)+(R-B)} \right] \quad (13.3)$$

$$sat = 1 - \frac{\min(R,G,B)+c}{int+c} \quad (13.4)$$

$$hue = \arctan \left[\frac{\sqrt{3}(G-B)+c}{(R-G)+(R-B)+c} \right] \quad (13.5)$$

где R, G, B – значения в пространстве RGB в диапазоне [0;255]; int – интенсивность цвета (Intensity), sat – насыщенность цвета (Saturation), hue – оттенок цвета в цветовом пространстве HSI; c – положительная константа, цель прибавления которой состоит в невозможности равенства нулю знаменателей в выражениях для sat и hue (в этой работе использовано значение c=0.0001). Константа c используется в (13.4-13.5) аналогично константам C₁, C₂, C₃ в (5.9-5.11). В HSI обозначим hue как col1, sat как col2, int как col3.

Следующее цветовое пространство – LMS, обозначим его как cs9. Преобразование из RGB в LMS выполняется

[16] по формулам (14.1-14.3):

$$L=(1688*R+2146*G+262*B)/4096 \quad (14.1)$$

$$M=(683*R+2951*G+462*B)/4096 \quad (14.2)$$

$$S=(99*R+309*G+3688*B)/4096 \quad (14.3)$$

где R, G, B – значения компонентов пространства RGB; L, M, S – значения компонентов пространства (линейного представления) LMS. В LMS обозначим L как col1, M как col2, S как col3.

Следующее цветовое пространство – CIELAB, обозначим его как cs10. Преобразование из RGB в CIELAB выполняется через промежуточное пространство XYZ (обозначенное в этой работе как XYZ4). Преобразование из RGB в XYZ4 выполняется в соответствии с (10.1-10.10). Преобразование из XYZ4 в CIELAB выполняется [18] по формулам (15.1-15.4):

$$f(t) = \begin{cases} t^{1/3}, & t > (6/29)^3 \\ \frac{1}{3} * \left(\frac{29}{6}\right)^2 * t + \frac{16}{116}, & t \leq (6/29)^3 \end{cases} \quad (15.1)$$

$$L = 116 * f(Y/Y_n) - 16 \quad (15.2)$$

$$a = 500 * [f(X/X_n) - f(Y/Y_n)] \quad (15.3)$$

$$b = 200 * [f(Y/Y_n) - f(Z/Z_n)] \quad (15.4)$$

где X, Y, Z – значения в пространстве XYZ4, полученные в соответствии с (10.1-10.10) без нормализации к диапазону [0;1]; L, a, b – значения компонент цвета в пространстве CIELAB; X_n, Y_n, Z_n – константы (в этой работе их значения взяты в соответствии со стандартом D65, X_n=0.95047, Y_n=1.0, Z_n=1.08883). В CIELAB обозначим L как col1, a как col2, b как col3.

В каждом из 10 рассмотренных цветовых пространств для каждого из 3 цветовых компонентов (нормализованных к диапазону [0;1]) вычислены значения среднего степенного для каждого из значений k в некотором диапазоне (например, от 1 до 100). Таким образом получен набор (множество) массивов, где каждый массив содержит 3000 чисел (по одному числу для каждого искажённого изображения базы TID2013). Для каждого из этих массивов определён коэффициент линейной корреляции между числами этого массива и значениями MOS для обучающей выборки (то есть для чисел этого массива с номерами от 1 до 2400). Коэффициент линейной корреляции показывает степень близости зависимости между этими числами и значениями MOS к линейной. Задача состоит в том, чтобы выбрать подмножество этих массивов из всего множества, выполнить множественную линейную регрессию для чисел 1-2400 всех массивов этого подмножества, получить набор коэффициентов (K0, K1, ..., KN) и вычислить коэффициент линейной корреляции между значениями MOS и вычисленными в результате применения найденных коэффициентов K значениями для тестовой выборки. Выполнение множественной линейной регрессии на обучающей выборке является обучением модели (то есть нахождением закономерностей и нахождением значений набора коэффициентов K), вычисление коэффициента линейной корреляции для результатов применения

модели к тестовой выборке является проверкой применимости этой модели за пределами обучающей выборки.

Изначально неизвестно, будут ли именно средние степенные значения (обозначим их как x) иметь самую сильную корреляцию со значениями MOS. Возможно наиболее сильную корреляцию будут иметь не значения x, а функции вида func1(x), func2(x), ..., funcm(x) – функции одной переменной, результаты которых тоже будут использоваться в качестве входных параметров множественной регрессии. Все значения x находятся в диапазоне [0;1] (так как они являются средними степенными наборов чисел, все из которых тоже входят в этот диапазон). Функции func1(x), ..., funcm(x) выбраны так, чтобы все их результаты тоже входили в диапазон [0;1] (при условии принадлежности их аргументов диапазону [0;1]). Рассмотрим используемые функции.

Func1(x)=x. При значениях аргумента в диапазоне [0;1] значения функции находятся в диапазоне [0;1]. На всём диапазоне [0;1] быстрота изменения значений функции одинаковая (модуль производной по x одинаковый для всех значений x диапазона [0;1]). Можно было бы и не выделять это в отдельную функцию, но выделено для единства обозначений.

Func2(x)=sqrt(x). Квадратный корень из x, при значениях аргумента в диапазоне [0;1] значения функции находятся в диапазоне [0;1]. Вблизи нуля значения функции меняются быстрее (модуль производной по x больше), ближе к единице значения функции меняются медленнее (модуль производной по x меньше).

Func3(x)=x*x. Возведение x во вторую степень. При значениях аргумента в диапазоне [0;1] значения функции находятся в диапазоне [0;1]. Вблизи нуля значения функции меняются медленно (модуль производной по x меньше), ближе к единице значения функции меняются быстрее (модуль производной по x больше).

Func4(x)=ln(a*x+1)/ln(a+1), где a – некоторая положительная константа. Вычисление логарифма приводит к быстрому изменению значений функции на одном промежутке значений аргумента и медленному изменению значений функции на другом промежутке значений аргумента. Функция y=ln(x) приводит к отрицательным значениям y для всех значений x в интервале (0;1) и к отрицательной бесконечности в пределе при x=0, поэтому была необходима модификация функции ln(x) в func4(x). Для всех значений x в диапазоне [0;1] значения func4(x) тоже находятся в диапазоне [0;1] (это выполняется для любого положительного значения a). Вблизи нуля значения func4(x) изменяются значительно быстрее, чем вблизи единицы. От значения параметра a зависит быстрота изменения значений функции вблизи нуля (то есть модуль производной func4(x) по x, большее значение a соответствует большему значению модуля производной, то есть более быстрому росту значений y при близких к нулю значениях x). Исследуем полученные результаты с разными значениями

параметра a (так как изначально неизвестно, при каком из этих значений будет получен лучший результат). Пусть $\text{func4}(x)=\ln(10*x+1)/\ln(10+1)$ ($a=10$), $\text{func5}(x)=\ln(100*x+1)/\ln(100+1)$ ($a=100$), $\text{func6}(x)=\ln(1000*x+1)/\ln(1000+1)$ ($a=1000$), $\text{func7}(x)=\ln(10000*x+1)/\ln(10000+1)$ ($a=10000$).

$\text{Func8}(x)=(\exp(x)-1)/(e-1)$, экспонента. Функция $f(x)=\exp(x)$ возвращает значения в диапазоне $[1;e]$ при значениях аргумента в диапазоне $[0;1]$. Вычитание единицы сдвигает диапазон возвращаемых значений на 1 вниз (то есть он становится $[0;e-1]$), деление на $(e-1)$ уменьшает верхнюю границу диапазона значений функции в $(e-1)$ раз, в результате чего при значениях x в диапазоне $[0;1]$ значения $\text{func8}(x)$ находятся в диапазоне $[0;1]$.

$\text{Func9}(x)=(\cos(\pi x)+1)/2$. У функции $\text{func9}(x)$ происходит медленное изменение значений y как при близких к нулю значениях x , так и при близких к единице значениях x . Наибольшая быстрота изменения значений y (максимум модуля производной $\text{func9}(x)$ по x) достигается при $x=0.5$ (то есть в середине рассматриваемого диапазона значений аргумента). Графики функций $\text{func1}(x)$, ..., $\text{func9}(x)$ приведены на рис. 1.

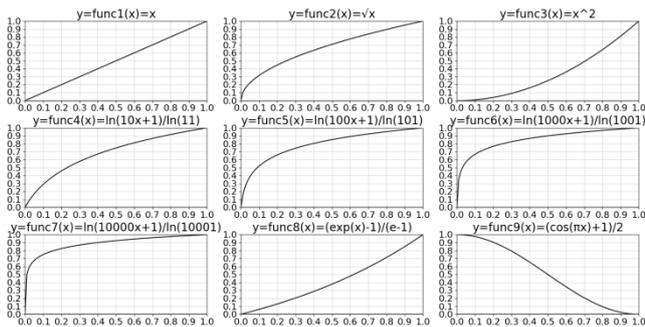


Рисунок 1 — Графики используемых функций в диапазоне значений аргумента $[0;1]$

На графиках видно, как меняются значения функций $\text{func1}(x)$, ..., $\text{func9}(x)$ и быстрота их изменения в зависимости от значений x для различных функций. Для $\text{func1}(x)$ быстрота изменения y одинакова для всех значений x рассматриваемого диапазона. Для $\text{func2}(x)$ вблизи нуля быстрота изменения y близка к бесконечности (производная функции $y=\sqrt{x}$ равна $1/(2*\sqrt{x})$, что в пределе справа при стремлении x к нулю даёт бесконечность). Функция $\text{func3}(x)$ возрастает наиболее медленно вблизи нуля и наиболее быстро (на рассматриваемом диапазоне) вблизи единицы. Графики функций $\text{func4}(x)$, ..., $\text{func7}(x)$ показывают наиболее быстрый рост вблизи нуля. Быстрота изменения значений функции $\text{func8}(x)$ увеличивается с увеличением x , но не от нуля (как в случае с $\text{func3}(x)$). График $y=\text{func9}(x)$ начинается и заканчивается (на рассматриваемом диапазоне) нулевой быстротой изменения значений функции. Цель использования различных функций от средних степенных значений в качестве входных данных для множественной линейной регрессии состоит в том, чтобы учесть возможную разную чувствительность человеческого зрения к

изменениям средних степенных значений (для исходных изображений, искажённых изображений и модулей разности) в разных частях диапазона этих значений (от минимального до максимального).

Вычисление средних степенных значений для трёх сигналов (исходное изображение, искажённое изображение и модули разности соответствующих пикселей) для каждого из трёх цветовых компонентов (col1 , col , col3) в десяти цветовых пространствах (cs1 , ..., cs10) для 100 значений параметра k (от 1 до 100) и использование девяти функций (func1 , ..., func9) для последующей обработки этих значений даёт $3*3*10*100*9=81000$ числовых параметров для каждого из 3000 изображений базы TID2013, каждое значение каждого из этих параметров принадлежит отрезку $[0;1]$. Так как каждый из 81000 полученных параметров в используемую модель множественной линейной регрессии либо входит, либо не входит (2 варианта), общее количество возможных моделей, которые можно составить из этих параметров (с одинаковым способом регрессии) равно $2^{81000} \approx 2.689*10^{24383}$. Выполнить полный перебор всех этих моделей и выбрать лучшую из них невозможно, поэтому задача сводится к следующему: на первом этапе по каким-либо признакам выбираются параметры (признаки), которые будут включены в модель (обозначим количество этих параметров как N). На втором этапе выполняется множественная линейная регрессия для обучающей выборки (параметров $X_1..X_N$ изображений 1-2400). На третьем этапе полученный набор значений (вещественных чисел с плавающей точкой) $K_0..K_N$ подставляется в уравнение $YK=K_0+K_1*X_1+...+K_N*X_N$, где X_1 , ..., X_N – наборы значений соответствующих числовых характеристик для тестового набора (то есть для изображений 2401-3000 базы TID2013). На четвёртом этапе вычисляется значение числового параметра, характеризующего близость значений YK для тестового набора (вычисленных по полученной в результате множественной линейной регрессии формуле, то есть предсказанных для тестовых данных на основе обучающих данных) к значениям Y для тестовых данных (то есть к значениям MOS для изображений 2401-3000). Этот параметр количественно оценивает качество модели. В качестве такого параметра могут использоваться различные значения, в данной работе будем использовать коэффициент линейной корреляции и среднеквадратическое отклонение. После выполнения всех четырёх этапов выполняется сравнение оценок качества этой модели с оценками качества лучшей из исследованных ранее моделей, чтобы таким образом выявить лучшую из моделей (наиболее подходящую для оценивания качества искажённых изображений). Так как выбраны два параметра оценки качества модели, возможно возникновение ситуации, когда из двух моделей каждая будет лучше любой другой из исследованных по одному из параметров.

Каждый из вычисленных 81000 параметров будем называть признаком (числовым признаком набора из одного исходного и одного искажённого изображений).

Для уменьшения количества рассматриваемых вариантов и исключения тех признаков, которые слабо коррелируют с оценками MOS, для каждого из 81000 признаков вычислен модуль коэффициента линейной корреляции между значениями этого признака и оценками MOS для изображений 1-2400 базы TID2013, затем выполнена сортировка всех признаков по убыванию модулей коэффициентов корреляции. Для дальнейшего исследования оставим 1000 наиболее коррелированных с оценками MOS признаков. Из этих 1000 признаков наиболее сильную линейную корреляцию показывает признак `diff_cs4_col3_k2_func6` (среднее степенное для модулей разности соответствующих пикселей в цветовом пространстве `cs4` для компонента цвета `col3` со значением $k=2$ при использовании функции `func6`), модуль коэффициента линейной корреляции для этого признака равен 0.6526. Наиболее слабую линейную корреляцию (из этих 1000 признаков) показывает признак `diff_cs5_col2_k=10_func2` (среднее степенное для модулей разности соответствующих пикселей в цветовом пространстве `cs5` для компонента цвета `col2` со значением $k=10$ при использовании функции `func2`), модуль коэффициента линейной корреляции для этого признака равен 0.5394.

Сначала рассмотрим модели множественной регрессии с использованием признаков только для одной из функций `func1`, ..., `func9`. Это связано с тем, что между значениями любого признака, использующего функцию `func1`, и значениями этого же признака, но использующего любую из функций `func2`, ..., `func9`, есть строгая функциональная зависимость, которая может привести к ухудшению результатов модели, если в этой модели используются одновременно оба этих признака.

Из 1000 наиболее сильно коррелированных признаков 117 получены с использованием функции `func1`. Общее количество всех возможных моделей с этими признаками достигает не менее $2^{117} \approx 1.662 \cdot 10^{35}$, следовательно перебрать все эти модели на практике невозможно. Поэтому полученные файлы с 117 признаками были переименованы в "X1.txt", ..., "X117.txt" (порядок, полученный при сортировке файлов с исходными названиями в алфавитном порядке). Далее были найдены коэффициенты и проверены результаты для моделей 01.1-01.117 (модель 01.1 использует только значения X1, модель 01.2 значения X1 и X2 и так далее). Каждая следующая модель использует на 1 признак больше предыдущей. Лучшие результаты этих моделей показаны в табл. 1.

Таблица 1. Результаты моделей, использующих признаки по возрастанию их количества

func	fttl	cor_max	cor_num	rmse_min	rmse_num
func1	117	0.6979	98	0.9097	98
func2	154	0.7298	131	0.8567	88
func3	13	0.4879	7	1.1115	7
func4	175	0.7400	102	0.8447	102
func5	142	0.7364	118	0.8564	118
func6	113	0.7274	92	0.8601	92

func7	109	0.7244	89	0.8635	89
func8	99	0.6847	73	0.9171	73
func9	78	0.5730	44	1.0260	52

В табл. 1 используются следующие обозначения столбцов: `func` - функция, результаты применения которой используются как признаки (`func1`, ..., `func9`); `fttl` - общее количество признаков из 1000 наиболее коррелированных с MOS, полученных с использованием этой функции (как сокращение от `features total`); `cor_max` - максимальное значение коэффициента линейной корреляции из всех моделей с признаками, использующими эту функцию; `cor_num` - используемое количество признаков, при котором достигнуто значение `cor_max`; `rmse_min` - минимальное значение среднеквадратического отклонения из всех моделей с признаками, использующими эту функцию; `rmse_num` - используемое количество признаков, при котором достигнуто значение `rmse_min`.

Подведём итог рассмотрения моделей, использующих наборы признаков, полученных применением одной из функций к средним степенным значениям, в каждый из которых входит на один признак больше предыдущего. Из 1000 рассмотренных моделей (01.1-09.78) наибольшее значение коэффициента линейной корреляции для тестовой выборки (0.7400) получено при использовании модели 04.102, наименьшее значение среднеквадратического отклонения для тестовой выборки (0.8447) тоже получено при использовании модели 04.102.

Не обязательно лучший результат (то есть с наибольшим коэффициентом линейной корреляции или наименьшим среднеквадратическим отклонением для тестовой выборки) покажет та модель, вхождение в которую признаков определяется их номерами, полученными при переименования после сортировки файлов по алфавиту. Возможна ситуация, когда, например, лучший результат будет показывать модель с признаками X1, X2, X5, X6 (без X3 и X4). Однако из-за слишком большого количества вариантов комбинаций признаков (и невозможности перебора всех комбинаций) в этом исследовании используется следующий подход. С использованием каждой из функций `func1`, ..., `func9` из всех признаков, вошедших в 1000 наиболее коррелированных с MOS, выбираются случайным образом несколько признаков. Для каждого из этих наборов выполняется множественная линейная регрессия, для полученной модели определяются коэффициент линейной корреляции и среднеквадратическое отклонение для тестовой выборки. Затем выбираются две лучшие модели (каждая по одному из этих двух критериев), которые могут совпасть между собой.

Для каждой из функций `func1`, ..., `func9` было исследовано по 10^5 моделей для каждого фиксированного количества используемых признаков от 5 до 50 (для `func3` максимальное количество признаков было 12, так как с этой функцией их всего 13 и модель со всеми 13 признаками уже проверена как 3.13). Из исследованных таким способом моделей наибольшее

достигнутое значение коэффициента линейной корреляции для тестовой выборки (назовём его cor_test) равно 0.7377, наименьшее достигнутое значение среднеквадратического отклонения для тестовой выборки (назовём его $rmse_test$) равно 0.8368.

Далее проведено исследование для моделей, в каждой из которых множество используемых признаков выбирается случайным образом (по отдельности для каждой из функций $func1, \dots, func9$). Из этих моделей лучшее значение cor_test равно 0.7526, лучшее значение $rmse_test$ равно 0.8231.

Далее исследуются модели, каждая из которых использует случайно выбранное число признаков из всех 1000 наиболее коррелированных с MOS (то есть сначала выбирается число n_{um} , а затем множество из n_{um} случайно выбранных признаков). До этого все рассматриваемые модели использовали признаки, полученные в результате применения только одной из функций $func1, \dots, func9$ к средним степенным значениям. Теперь рассмотрим модели, которые используют признаки вне зависимости от функции, с помощью которой они были получены. Из рассмотренных в этой части исследования моделей лучший результат по обоим критериям (cor_test и $rmse_test$) показала модель, использующая 113 признаков ($cor_test=0.8356$, $rmse_test=0.6864$).

Далее рассмотрим модели, использующие градиентный спуск (SGDRegressor в Python) для выполнения линейной регрессии. Идея градиентного спуска состоит в том, что на каждом шаге находится направление самого быстрого возрастания функции и происходит движение в противоположную сторону. Таким образом, на каждом шаге происходит перемещение в точку, в которой значение функции меньше предыдущего. Так как функция – это средняя ошибка обучения модели, то уменьшение значения этой функции означает улучшение результатов обучения модели. Помимо количества используемых признаков и самих признаков будем менять следующие два параметра модели SGDRegressor: max_iter (максимальное количество итераций) и tol (абсолютный допуск улучшения). Так как заранее неизвестно, изменение каких именно параметров и до каких конкретно значений приведёт к наилучшим моделям (по максимальному cor_test и по минимальному $rmse_test$), а полный перебор всех возможных комбинаций невозможен, для каждой модели будем выбирать эти значения (количество признаков, набор признаков, max_iter , tol) случайным образом. Наибольшее найденное значение cor_test равно 0.6841, наименьшее найденное значение $rmse_test$ равно 0.9031.

Теперь рассмотрим модели, использующие гребневую регрессию (ридж регрессия, Ridge). Во время выполнения Ridge регрессии к сумме квадратов остатков добавляется сумма квадратов коэффициентов k , в результате чего модель стремится не использовать слишком большие значения этих коэффициентов. Использование Ridge регрессии является способом борьбы с мультиколлинеарностью. Лучшие найденные в результате применения Ridge регрессии значения

cor_test и $rmse_test$ равны 0.7133 и 0.8655 соответственно.

Из всех исследованных в работе моделей по обоим критериям (максимальное значение cor_test и минимальное значение $rmse_test$) лучшей является модель, которая использует 113 признаков. Диаграммы рассеяния для этой модели приведены на рис. 2.

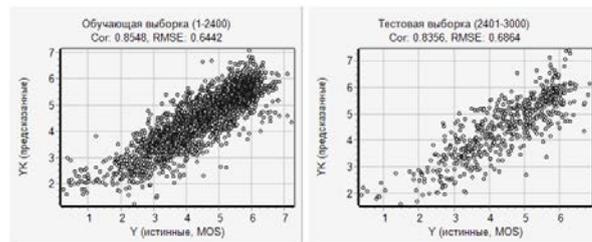


Рисунок 2 – Диаграммы рассеяния для обучающей и тестовой выборки лучшей найденной модели (использующей 113 признаков).

На диаграммах рассеяния видно, что данные обучающей выборки (результаты применения множественной линейной регрессии к средним степенным значениям для искажённых изображений 1-2400 базы TID2013) ближе к наклонной прямой $YK=Y$ по сравнению с данными тестовой выборки. Однако и для тестовой выборки полученные применением регрессии результаты близки к истинным значениям (экспертным оценкам качества, MOS). Видно, что коэффициент линейной корреляции между значениями Y и YK при переходе от обучающей выборки к тестовой уменьшился с 0.8548 до 0.8356, среднеквадратическое отклонение увеличилось с 0.6442 до 0.6864 (оба эти параметра ухудшились незначительно, что свидетельствует о хорошей обобщающей способности модели).

IV. ЗАКЛЮЧЕНИЕ

В результате выполнения работы было исследовано применение средних степенных значений для объективного эталонного оценивания качества искажённых изображений.

В ходе выполнения был проведён анализ предметной области, заключающийся в рассмотрении основных метрик качества изображений, их достоинств и недостатков. Также были рассмотрены различные цветовые пространства, для компонентов которых в дальнейшем были вычислены средние степенные значения для пикселей исходных изображений, пикселей искажённых изображений и модулей разности соответствующих пикселей.

Следующим этапом стало нахождение базы изображений TID2013, содержащей искажённые изображения и экспертные оценки качества изображений. База была разделена на две непересекающиеся части (обучающую и тестовую выборки) в соотношении 80% на 20%. Для каждого из 3000 искажённых изображений были вычислены значения 81000 числовых признаков. Для построения моделей были использованы 1000 наиболее

коррелированных с оценками MOS признаков (при этом учитывалась корреляция только для обучающей выборки, чтобы тестовая выборка не влияла на выбор признаков).

Заключительным этапом стало применение различных моделей линейной регрессии (LinearRegression, SGDRegressor, Ridge) с использованием библиотеки sklearn языка программирования Python с использованием некоторой части полученных признаков. Полный перебор всех комбинаций признаков был невозможен из-за слишком большого количества этих комбинаций. Каждая модель оценивалась по двум параметрам: коэффициенту линейной корреляции между Y (оценки MOS) и Y_K (оценки модели) и среднеквадратическому отклонению (RMSE, root-mean-square error) между Y и Y_K , оба этих параметра вычислялись для тестовой выборки. Таким образом, из всех рассмотренных моделей выбирались те, которые показывают лучшую производительность на тестовых данных (на которых они не обучались), что свидетельствует об обобщающих способностях этих моделей (то есть о том, что эти модели применимы не только на тех данных, на которых они обучались, но и на других данных). В итоге наибольшее значение коэффициента линейной корреляции (свидетельствующее о самой сильной линейной связи между истинными и предсказанными значениями) и наименьшее значение среднеквадратического отклонения (свидетельствующее о наименьшем отличии между истинными и предсказанными значениями) были получены на одной и той же модели, что делает эту модель лучшей по обоим критериям.

БИБЛИОГРАФИЯ

- [1] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult," in Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing, vol. 4, (Orlando), pp. 3313–3316, May 2002.
- [2] Z. Wang, A. Bovik, H. Sheikh and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity", IEEE Transactions on Image Processing, Vol. 13, Issue 4, pp. 600-612, 2004.
- [3] Лапшенков Е.М. Незатонная оценка уровня шума цифрового изображения на основе гармонического анализа // Компьютерная оптика. - 2012. - № 3 (36). - С. 439-447.
- [4] Z. Wang and A. C. Bovik, "Mean Squared Error: Love It or Leave It?" IEEE Signal Processing Magazine, vol. 9, pp. 98–117, Jan. 2009.
- [5] C. Ma, C. Yang, X. Yang and M. Yang, "Learning a no-reference quality metric for single-image super-resolution", Computer Vision and Image Understanding, Vol. 158, pp. 1-16, 2017.
- [6] Власюк И. В., Поташников А. М., Селиванов В. А. Исследование методик автоматизированного измерения основных параметров и характеристик телевизионных IP-камер : лаб. практикум. Москва, 2019. 18 с.
- [7] Будько А. А., Дворникова Т. Н., Мисулин Е. А., Снапко Р. Ю. Сжатие изображений с использованием функций Уолша. Доклады БГУИР. 2022. 20 (7). С. 88–94.
- [8] Z. Wang and A. C. Bovik, "A universal image quality index," IEEE Signal Processing Letters, vol. 9, pp. 81–84, Mar. 2002.
- [9] IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2019, 2019.
- [10] Мироновский Л.А., Слаев В.А. Оценивание результатов измерений по малым выборкам // Информационно-управляющие системы. — 2011. — № 1. — С. 69-78.
- [11] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola et al., "Image database TID2013: Peculiarities, results and perspectives", Signal Processing: Image Communication, Vol. 30, pp. 57-77, 2015.

- [12] ITU, Методы для объективной и субъективной оценки качества речи и видео, Рекомендация МСЭ-Т P.800.2 (07/2016).
- [13] А. В. Кугаевских, Д. И. Муромцев, О. В. Кирсанова. Классические методы машинного обучения. – СПб: Университет ИТМО, 2022. – 53 с.
- [14] G. Hoffmann. CIELab Color Space. – 63 p. Available: <http://www.docs-hoffmann.de/cielab03022003.pdf>
- [15] G. Hoffmann. CIE Color Space. – 37 p. Available: <http://www.docs-hoffmann.de/ciexyz29082000.pdf>
- [16] ITU, Объективная метрика для оценки потенциальной видимости цветовых различий в телевидении, Рекомендация МСЭ-Р BT.2124-0 (01/2019).
- [17] D. B. Judd, Color in Buisiness, Science, and Industry. New York: JOHN WILEY&SONS, INC, 1952, p. 148.
- [18] Penczek, J. (2014), "CIELAB Color Difference Analysis for Digital Color Photography", Proceedings of the ICC Medical Imaging Task Force Medical Photography Teleconference, Boulder, April, pp. 9-10.
- [19] E. Hamilton, JPEG File Interchange Format, Version 1.02, C-Cube Microsystems. [Online]. Available: <https://www.w3.org/Graphics/JPEG/jfif3.pdf>
- [20] H. S. Malvar, G. J. Sullivan and S. Srinivasan, "Lifting-based reversible color transformations for image compression", Proceedings of SPIE, Vol. 7073, Issue 07, 2008.
- [21] F. Perez and C. Koch, "Toward Color Image Segmentation in Analog VLSI: Algorithm and Hardware" International Journal of Computer Vision, vol. 12:1, pp. 17–42, Jan. 1994.

Using power mean for image quality assessment

A.A. Nechayev

Abstract — Results of research on using power mean values for image quality assessment are presented in this article. The objects of study in the work are linear correlation coefficients between image quality metrics values and mean opinion scores. Full Reference quality metrics are researched in this paper. Power mean values with different parameters are used as researched metrics. The guideline of the research is using TID2013 image database and mean opinion scores for detecting objective metrics that are the most correlated with mean opinion scores and using multiple regression for increasing the correlation. Embarcadero RAD Studio IDE was used for conducting computational experiments, Python programming language was used for performing regression and evaluating the results. Research results can be used for objective quality assessment of distorted images.

Keywords — image quality, multiple regression, power mean, full reference metrics.

REFERENCES

- [1] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult," in Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing, vol. 4, (Orlando), pp. 3313–3316, May 2002.
- [2] Z. Wang, A. Bovik, H. Sheikh and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity", IEEE Transactions on Image Processing, Vol. 13, Issue 4, pp. 600-612, 2004.
- [3] Lapshenkov E.M. No reference metric of digital image noise level based on harmonic analysis // Computer optics. - 2012. - № 3 (36). - P. 439-447.
- [4] Z. Wang and A. C. Bovik, "Mean Squared Error: Love It or Leave It?" IEEE Signal Processing Magazine, vol. 9, pp. 98–117, Jan. 2009.
- [5] C. Ma, C. Yang, X. Yang and M. Yang, "Learning a no-reference quality metric for single-image super-resolution", Computer Vision and Image Understanding, Vol. 158, pp. 1-16, 2017.
- [6] Vlasiuk I. V., Potashnikov A. M., Selivanov V. A. Research of methods for automated measurement of the main parameters and characteristics of IP television cameras : lab. practical work. Moscow, 2019. 18 p.
- [7] Budko A. A., Dvornikova T. N., Misulin E. A., Snapko R. U. Image compression using Walsh functions. Doklady BGUIR. 2022. 20 (7). P. 88–94.
- [8] Z. Wang and A. C. Bovik, "A universal image quality index," IEEE Signal Processing Letters, vol. 9, pp. 81–84, Mar. 2002.
- [9] IEEE Standard for Floating-Point Arithmetic, IEEE Std 754-2019, 2019.
- [10] Mironovskiy L.A., Slayev V.A. Evaluation of measurement results from small samples // Information and management systems. — 2011. — № 1. — P. 69-78.
- [11] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola et al., "Image database TID2013: Peculiarities, results and perspectives", Signal Processing: Image Communication, Vol. 30, pp. 57-77, 2015.
- [12] ITU, Methods for objective and subjective assessment of speech and video quality, ITU-T Recommendation P.800.2 (07/2016).
- [13] A. V. Kugaevskih, D. I. Muromtsev, O. V. Kirsanova. Classic machine learning methods. – S. Petersburg: ITMO University, 2022. – 53 p.
- [14] G. Hoffmann. CIELab Color Space. – 63 p. Available: <http://www.docs-hoffmann.de/cielab03022003.pdf>
- [15] G. Hoffmann. CIE Color Space. – 37 p. Available: <http://www.docs-hoffmann.de/ciexyz29082000.pdf>
- [16] ITU, An objective metric for assessing the potential visibility of color differences in television, ITU-R Recommendation BT.2124-0 (01/2019).
- [17] D. B. Judd, Color in Buisiness, Science, and Industry. New York: JOHN WILEY&SONS, INC, 1952, p. 148.
- [18] Penczek, J. (2014), "CIELAB Color Difference Analysis for Digital Color Photography", Proceedings of the ICC Medical Imaging Task Force Medical Photography Teleconference, Boulder, April, pp. 9-10.
- [19] E. Hamilton, JPEG File Interchange Format, Version 1.02, C-Cube Microsystems. [Online]. Available: <https://www.w3.org/Graphics/JPEG/jfif3.pdf>
- [20] H. S. Malvar, G. J. Sullivan and S. Srinivasan, "Lifting-based reversible color transformations for image compression", Proceedings of SPIE, Vol. 7073, Issue 07, 2008.
- [21] F. Perez and C. Koch, "Toward Color Image Segmentation in Analog VLSI: Algorithm and Hardware" International Journal of Computer Vision, vol. 12:1, pp. 17–42, Jan. 1994.