

Аскарян Каринэ Артуровна  
магистрант, МГТУ им. Н. Э. Баумана

Филиппов Михаил Владимирович  
к.т.н., доцент, МГТУ им. Н. Э. Баумана

## ОБЗОР МЕТОДОВ ВОССТАНОВЛЕНИЯ ЦВЕТОВОЙ ИНФОРМАЦИИ ИЗОБРАЖЕНИЙ

**Аннотация:** В статье представлен обзор методов восстановления цветовой информации изображений. Рассмотрены классические методы и методы машинного обучения, применяемые для восстановления цветовой информации изображений. Выделены особенности, преимущества и ограничения рассмотренных подходов.

**Ключевые слова:** восстановление цветовой информации; оптимизационные методы; перенос цвета; сверточные нейронные сети; генеративно-состязательные сети.

**Введение.** Колоризация изображений представляет собой процесс добавления цветовой информации к монохромному изображению. В общем случае такая задача является неоднозначной, поскольку одному и тому же значению яркости могут соответствовать различные цвета. Поэтому качество раскрашивания зависит не только от исходных значений интенсивности, но и от способа учета границ объектов, локального контекста и дополнительной информации о цвете [1].

Методы решения поставленной задачи можно глобально разделить на две группы: методы, с явно заданной моделью восстановления, и методы с использованием машинного обучения.

**Оптимизационное распространение цветовой информации.** В данном методе изображение рассматривается в цветовом пространстве  $YUV$ , где  $Y$  — монохромный канал яркости, а  $U$  и  $V$  — каналы, кодирующие цвет [1].

Основное предположение: два соседних пикселя  $r$  и  $s$  должны иметь похожие цвета, если их интенсивности похожи. В таком случае данная задача сводится к минимизации разницы между цветом  $U(r)$  в пикселе  $r$  и взвешенным средним цветом в соседних пикселях:

$$J(U) = \sum_r \left( U(r) - \sum_{s \in N(r)} w_{rs} U(s) \right)^2,$$

где  $U(r)$ ,  $U(s)$  — один из цветовых каналов пикселей  $r$ , соответственно, а  $w_{rs}$  — весовая функция, сумма значений которой равна единице, называемая функцией сходства. Такие функции принимают большие значения, когда  $Y(r)$  близко к  $Y(s)$ , и малые значения, когда интенсивности отличаются. Запись  $s \in N(r)$  означает, что пиксели  $r$  и  $s$  являются соседними [1].

В качестве весовой функции можно использовать функцию, представленную следующей формулой:

$$w_{rs} \propto e^{-\frac{(Y(r)-Y(s))^2}{2\sigma_r^2}}.$$

Имея набор точек  $r_i$ , в которых цвета заданы внешним пользователем:  $u(r_i) = u_i$ ,  $v(r_i) = v_i$ , необходимо минимизировать функции  $J(U)$ ,  $J(V)$ . Поскольку функции стоимости являются квадратичными, а ограничения — линейными, задача оптимизации приводит к большой разреженной системе линейных уравнений, которую можно решить несколькими стандартными методами [1].

Можно выделить следующие преимущества метода: высокая интерпретируемость; не требуется обучающая выборка; возможность контролировать распространение цвета с помощью подсказок пользователя. И следующие недостатки: требуется вмешательство внешнего пользователя; метод не учитывает семантику сцены; возможны распространения цвета через границы объектов [1, 2].



Метод можно дополнить адаптивным выделением границ или, к примеру, заменить весовую функцию. Такие модификации усложняют метод и увеличивают его вычислительную сложность [2].

**Перенос цветовой информации с эталонного изображения.** Основная идея этого метода состоит в переносе цветовой информации с эталонного цветного изображения на целевое изображение в оттенках серого.

Изображения рассматриваются в цветовом пространстве  $l\alpha\beta$ , которое разработано для минимизации корреляции между тремя координатными осями цветового пространства. Оно представляет три канала: ахроматический канал яркости  $l$  и два хроматических канала  $\alpha, \beta$ , которые соответствуют каналам «жёлтый — синий» и «красный — зелёный» [3].

Следующим шагом последовательно для каждого пикселя изображения в оттенках серого выбирается наиболее подходящий образец из цветного изображения. В результате значение цветности, а именно компоненты  $\alpha, \beta$ , переносятся на изображение в оттенках серого, формируя итоговую картинку [3].

В методе считается, что большая часть визуально значимых различий между значениями пикселей связана с различиями яркости. Это позволяет ограничить число образцов, используемых в качестве цветных пикселей-источников и уменьшить количество сравнений для каждого пикселя изображения [3].

На третьем этапе уже окрашенные области целевого изображения используются как локальные цветные примеры для окрашивания остальных пикселей [3].

Для поиска похожих участков внутри целевого изображения используются метрика  $L_2$  между окрестностью  $N_g$  в изображении в оттенках серого и окрестностью  $N_s$  в уже окрашенной области:

$$E(N_g, N_s) = \sum_{p \in N} (I(p) - S(p))^2,$$

где  $I$  — изображение в оттенках серого,  $S$  — канал яркости окрашенной области, а  $p$  — пиксели сравниваемых окрестностей. Чем меньше значение  $E$ , тем более похожими считаются две окрестности [3].

В сложных случаях используются области-соответствия, задаваемые пользователем на эталонном и целевом изображениях. Такие области позволяют указать, какие фрагменты двух изображений должны соответствовать друг другу, например область неба на эталонном изображении и область неба на целевом изображении [3].

Модификацией метода может стать поиск изображений в интернете. В таком случае пользователь задает семантическую метку объекта и выполняет сегментацию основных объектов переднего плана. Далее загружаются изображения из интернет-источников и выбираются наиболее подходящие эталонные области для переноса цвета [4].

Можно выделить следующие преимущества метода: не требуется обучение нейронной сети; используются реальные цветные примеры; возможность переноса общего цветового стиля. И недостатки: требуется подходящее эталонное изображение; качество результата зависит от сходства эталонного и целевого изображений; может зависеть от внешнего пользователя [3, 4].

**Сверточные нейронные сети.** Данный метод предлагает полностью автоматическое восстановление цветовой информации без задания пользовательских подсказок. Изображение рассматривается в цветовом пространстве CIE Lab. На вход подается канал яркости  $L$ , а сеть предсказывает два цветных канала  $a$  и  $b$ .

Пусть  $Y$  — истинные значения цветных компонент, а  $\hat{Y}$  — значения, предсказанные нейронной сетью. Так как расстояния в цветовом пространстве CIE Lab моделируют перцептивное расстояние, то естественной функцией потерь является евклидова ошибка [5]:

$$L_2(\hat{Y}, Y) = \frac{1}{2} \sum_{h,w} \|Y_{h,w} - \hat{Y}_{h,w}\|_2^2,$$



где  $Y_{h,w}$  — истинный вектор цветовых компонент пикселя с координатами  $(h, w)$ ,  $\hat{Y}_{h,w}$  — предсказанный сетью вектор цветовых компонент этого пикселя,  $h$  и  $w$  — координаты пикселя по высоте и ширине изображения, а  $\|\cdot\|_2$  — евклидова норма. Суммирование выполняется по всем пикселям изображения.

Однако такая функция потерь не учитывает внутреннюю неоднозначность задачи колоризации. Если объект имеет несколько разных допустимых значений  $ab$ , то оптимальное решение для евклидовой ошибки будет средним значением этих вариантов, что в итоге приводит к сероватым и малонасыщенным результатам [5].

В таком случае можно рассмотреть задачу как многоклассовую классификацию. Пространство цветовых компонент разбивается на конечное число допустимых цветовых значений, а нейронная сеть для каждого пикселя предсказывает распределение вероятностей по этим цветовым классам. Для сравнения предсказанного распределения с истинным цветом используется функция потерь многоклассовой кросс-энтропии:

$$L_{cl}(\hat{Z}, Z) = \sum_{h,w} v(Z_{h,w}) \sum_q Z_{h,w,q} \log(\hat{Z}_{h,w,q}),$$

где  $Z$  — истинное распределение по цветовым классам, полученное на основе реального цвета пикселя;  $\hat{Z}$  — распределение вероятностей, предсказанное нейронной сетью;  $h$  и  $w$  — координаты пикселя по высоте и ширине изображения;  $q$  — номер цветового класса;  $Z_{h,w,q}$  — принадлежность пикселя  $(h, w)$  к цветовому классу  $q$ ;  $\hat{Z}_{h,w,q}$  — предсказанная вероятность этого класса;  $v(\cdot)$  — весовой множитель, используемый для балансировки редких и часто встречающихся цветов [5].

Кроме многоклассовой классификации на локальных признаках можно использовать одновременно и локальную информацию, и глобальный контекст изображения. В таком случае локальные признаки будут описывать отдельные участки изображения, а глобальные дадут информацию о сцене в общем [6].

Тогда архитектура будет состоять из четырех частей: сети низкоуровневых признаков, сети среднеуровневых признаков, сети глобальных признаков и сети колоризации. Из входного изображения извлекаются общие низкоуровневые признаки, на их основе отдельно вычисляются среднеуровневые локальные признаки и глобальный вектор признаков изображения. После этого они объединяются и результат передается в сеть колоризации [6].

Можно выделить следующие преимущества метода: автоматический метод колоризации без участия человека; способность изучать сложные нелинейные отображения; адаптивность к сложным сценам выше, чем у рассматриваемых ранее методов. И следующие недостатки: зависимость от обучающей выборки; вычислительная сложность; возможны блеклые цвета или ошибки пространственного согласования [5, 6].

**Генеративно-сопоставительные сети (GANs).** GANs представляет собой архитектуру, состоящую из двух соревнующихся нейронных сетей — генератора и дискриминатора — играющих в игру с нулевой суммой [7, 8]. Обе нейронные сети являются многослойными перцептронами [9].

Сеть, называемая генератором ( $G$ ), отвечает за создание новых образцов данных, воспроизводящих, но не копирующих, исходное распределение данных, пока сеть, называемая дискриминатором ( $D$ ) пытается отличить сгенерированные данные от реальных. Процесс обучения GAN направлен на улучшение распознавания подделок дискриминатором и создание более реалистичных данных генератором [7, 8].

Генератор  $G$  представлен как  $G(z, \theta_G)$ , где  $z$  — распределенная переменная шума, которая используется в качестве входных данных для генератора, а  $\theta_G$  — набор обучаемых параметров генератора  $G$  [7, 9].

Аналогично Дискриминатор  $D$  представлен как  $D(x, \theta_D)$ , где  $x$  — цветное изображение,  $\theta_D$  — набор обучаемых параметров дискриминатора  $D$  [7, 9].



Выход дискриминатора интерпретируется как вероятность того, что входные данные принадлежат обучающему набору. Это позволяет определить задачу оптимизации:  $G$  обучается минимизировать вероятность того, что дискриминатор правильно классифицирует данные, а  $D$  — максимизировать вероятность правильной классификации [7].

Обучение GAN формулируется как минимаксная игра между генератором и дискриминатором. Такая постановка задаётся следующей функцией стоимости [7, 9]:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z)))]$$

где  $\mathbb{E}$  — математическое ожидание,  $p_{data}$  — распределение реальных данных, а  $p_z$  — распределение латентной переменной  $z$ , в качестве которого обычно используется равномерное или гауссово распределение.

Таким образом, генератор и дискриминатор обучаются совместно: первый стремится минимизировать функцию, приведённую ранее, а второй — максимизировать её [9].

Можно выделить следующие преимущества метода: повышенная точность, по сравнению с CNN; более эффективное обучение из-за двухэтапной стратегии; комплексная функция потерь, учёт правдоподобия изображения. И недостатки: вычислительная сложность; чувствительность к гиперпараметрам; интерпретируемая модель [7, 8, 9].

**Сравнение методов.** В качестве критериев сравнения методов выбраны: необходимость обучающей выборки; наличие вмешательства внешнего пользователя; возможность автоматического восстановления; учёт пространственного контекста изображения; учёт правдоподобия результата относительно реальных цветных изображений.

В таблице 1 представлено соответствие рассмотренных ранее методов описанным выше критериям. Приняты следующие обозначения: «ОРЦ» — оптимизационное распространения цветовой информации; «ПЦЭ» — перенос цветовой информации с эталонного изображения; «CNN» — сверточные нейронные сети; «GANs» — генеративно-сопоставительные сети.

Таблица 1

Соответствие рассмотренных методов выбранным критериям

Критерий	Методы			
	ОРЦ	ПЦЭ	CNN	GANs
Необходимость обучающей выборки	–	–	+	+
Требуется вмешательство внешнего пользователя	+	+	–	–
Автоматическое восстановление	–	–	+	+
Учёт пространственного контекста	–	–	+	+
Учёт правдоподобия	–	–	–	+

### Список литературы:

1. Levin A., Lischinski D., Weiss Y. Colorization using optimization //ACM SIGGRAPH 2004 Papers. – 2004. – С. 689-694.
2. Huang Y. C. et al. An adaptive edge detection based colorization algorithm and its applications //Proceedings of the 13th annual ACM international conference on Multimedia. – 2005. – С. 351-354.
3. Welsh T., Ashikhmin M., Mueller K. Transferring color to greyscale images //Proceedings of the 29th annual conference on Computer graphics and interactive techniques. – 2002. – С. 277-280.
4. Chia A. Y. S. et al. Semantic colorization with internet images //ACM Transactions on Graphics (ToG). – 2011. – Т. 30. – №. 6. – С. 1-8.
5. Zhang R., Isola P., Efros A. A. Colorful image colorization //European conference on computer vision. – Cham : Springer International Publishing, 2016. – С. 649-666.
6. Iizuka S., Simo-Serra E., Ishikawa H. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification //ACM Transactions on Graphics (ToG). – 2016. – Т. 35. – №. 4. – С. 1-11.



7. Фатхулин Т. Д., Боданюк А. П., Рахматова А. А. Сравнительный анализ ключевых особенностей генеративно-сопоставительных сетей //DSPA: Вопросы применения цифровой обработки сигналов. – 2025. – Т. 15. – №. 1. – С. 11-16.
8. Iglesias G., Talavera E., Díaz-Álvarez A. A survey on GANs for computer vision: Recent research, analysis and taxonomy //Computer Science Review. – 2023. – Т. 48. – С. 100553.
9. Goodfellow I. J. et al. Generative adversarial nets //Advances in neural information processing systems. – 2014. – Т. 27.

