

Оценка качества колоризации изображений природного ландшафта нейросетевого автокодировщика

А.А. Кудяшов, А.А. Новикова

Тихоокеанский государственный университет, Хабаровск

Аннотация: В статье рассматривается применение нейросетевого автокодировщика в задаче колоризации монохромных изображений. Приводится описание архитектуры сети, применяемый метод обучения и способ формирования обучающих и проверочных данных. При обучении использовался набор данных, состоящий из 540 изображений природного ландшафта с разрешением 256 на 256 пикселей. В результате была проведена оценка качества выводов полученной модели и были вычислены усредненные коэффициенты метрик, а также среднеквадратичная ошибка выводов модели VGG.

Ключевые слова: нейронные сети, машинное обучение, автокодировщик, анализ качества изображений, колоризация, CIELAB.

Введение

Колоризация – процесс преобразования монохромного изображения в цветное. Традиционная колоризация изображений часто требует больших человеческих усилий. В целях автоматизации данного процесса в работе применяется нейронная сеть с архитектурой автокодировщика [1].

Кодирующая часть (энкодер) формирует сжатое пространство наиболее важных признаков монохромного изображения. Декодированная часть (декодер) реконструирует цветное аппроксимированное изображение на основе данных этого пространства.

В данной работе в процессе обучения на вход сети подавались монохромные изображения и на выходе сравнивались с оригинальными цветными изображениями в цветовом пространстве CIELAB на основе метрики SSIM и среднеквадратической ошибки (MSE) выводов модели VGG [2]. При этом данные декодера и энкодера связаны слоями суммирования. Использование таких типов связей нашло применение в архитектурах моделей автокодировщика для удаления артефактов сжатия JPEG [3], обнаружении аномалий [4], а также при создании карт глубин сцен цветных изображений [5].

Цветовое пространство

Многие цветовые пространства используют три канала для формирования изображения. Данные, хранящиеся в этих каналах, объединяются для точного воспроизведения цвета [6]. Одним из наиболее часто используемых цветовых пространств является RGB. В данном цветовом пространстве три канала представляют «красный», «зеленый» и «синий» цвета, соответственно [7]. Поскольку для формирования чёрно-белого изображения в RGB необходимо задействовать все три канала, в данной работе используется цветовое пространство CIELAB. Изображение в CIELAB состоит из светлоты L и двух компонент a и b (рис.1), формирующих цветовую составляющую изображения [8]. В таком случае для обучения достаточно использовать два канала a и b , вместо трёх, что даёт возможность увеличить ёмкость сети, не меняя её общее количество параметров.

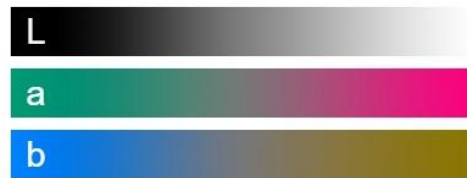


Рис. 1. – Цветовое пространство CIELAB

Перед началом обучения все изображения RGB были переведены в цветовое пространство CIEXYZ [9] и затем в CIELAB при помощи следующих преобразований:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.19334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix},$$

$$X = \frac{X}{0.950456},$$

$$Z = \frac{Z}{1.088754},$$

$$L = \begin{cases} 116 \cdot Y^{\frac{1}{3}} - 16, & \text{при } Y > 0.0008856, \\ 903.3 \cdot Y, & \text{при } Y \leq 0.0008856 \end{cases}$$

$$a = 500 \cdot (f(X) - f(Y)),$$

$$b = 200 \cdot (f(Y) - f(Z)),$$

$$f(t) = \begin{cases} t^{\frac{1}{3}}, & \text{при } t > 0.0008856 \\ 7.787 \cdot t + \frac{16}{116}, & \text{при } Y \leq 0.0008856 \end{cases}.$$

Значения каналов CIELAB затем были нормализованы в диапазоне [0,1]:

$$L = \frac{L}{100},$$

$$a = \frac{a + 127}{255},$$

$$b = \frac{b + 127}{255}.$$

Пример представления каналов цветового пространства RGB и CIELAB показан на рис.2.

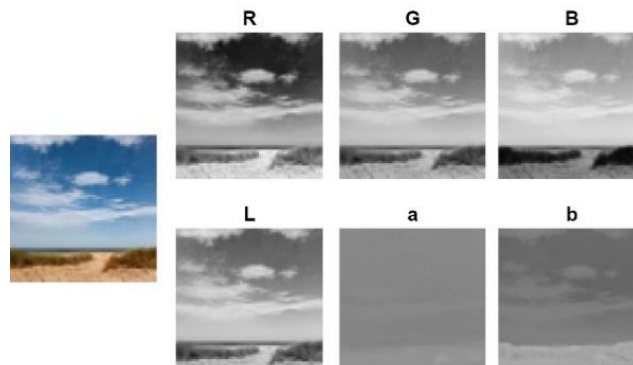


Рис. 2. – Каналы цветного изображения в представлении RGB и CIELAB

Архитектура модели

Модель автокодировщика состоит из слоёв энкодера (таблица № 1) и декодера (таблица № 2). На вход энкодера подавались значения компоненты светлоты L . В слоях свёртки (Conv2D) с размером ядра 3×3 использовалась функция активации ReLU. Для уменьшения размерности выходов некоторых слоёв энкодера использовался сдвиг ядра, равный 2. Для увеличения размерности выходов некоторых слоёв декодера использовался слой

UpSampling. На выходе сети значения компонент a и b приводились к диапазону $[0, 1]$ функцией сигмоиды.

Таблица № 1

Архитектура энкодера

№ сл.	Название слоя	Параметры
1	Input [256, 256, 1]	
2	RP2D и Conv2D [256, 256, 48] [Ядро: 3×3]	Активация: ReLU,
3	RP2D и Conv2D [128, 128, 48] [Ядро: 3×3]	Активация: ReLU, Сдвиг: 2
4	RP2D и Conv2D [128, 128, 96] [Ядро: 3×3]	Активация: ReLU,
5	RP2D и Conv2D [64, 64, 96] [Ядро: 3×3]	Активация: ReLU, Сдвиг: 2
6	RP2D и Conv2D [64, 64, 192] [Ядро: 3×3]	Активация: ReLU
7	RP2D и Conv2D [32, 32, 384] [Ядро: 3×3]	Активация: ReLU, Сдвиг: 2
8	BatchNormalization (Пакетная нормализация)	Момент: 0.95

Таблица № 2

Архитектура декодера

№ сл.	Название слоя	Параметры
9	UpSampling2D (Повышение размерности)	Размер: 2×2
10	RP2D и Conv2D [64, 64, 192] [Ядро: 3×3]	Активация: ReLU,
11	Add (№10 + №6)	
12	UpSampling2D (Повышение размерности)	Размер: 2×2
13	RP2D и Conv2D [128, 128, 96] [Ядро: 3×3]	Активация: ReLU,
14	Add (№13 + №4)	
15	RP2D и Conv2D [128, 128, 96] [Ядро: 3×3]	Активация: ReLU,
16	UpSampling2D (Повышение размерности)	Размер: 2×2
17	RP2D и Conv2D [256, 256, 48] [Ядро: 3×3]	Активация: ReLU,
18	RP2D и Conv2D [256, 256, 48] [Ядро: 3×3]	Активация: ReLU,
19	Add (№18 + №2)	
20	Output: Conv2D (Свёртка) [256, 256, 2] [Ядро: 3×3]	Активация: sigmoid,
Общее число параметров: 1 953 362		

Слои RP2D (ReflectionPad2D) с заданным числом отступов используются для дополнения выходов предыдущего слоя, повторяя элементы данных (рис.3).

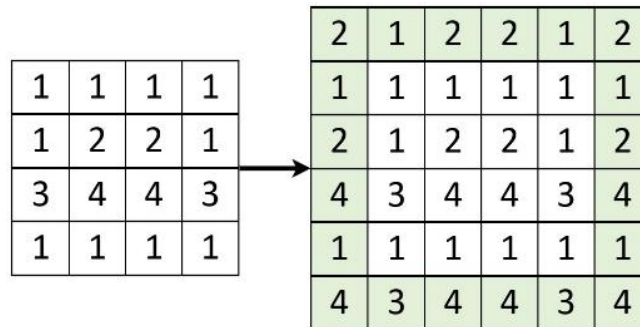


Рис. 3. – Пример выполнения ReflectionPad2D с числом отступов равным 1

Данный слой позволяет избежать возникновения артефактов на границах выводов операций свёртки (рис.4) [10].

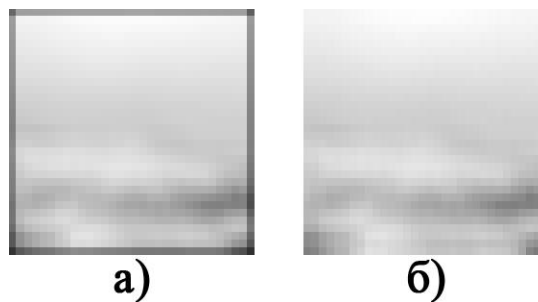


Рис. 4. – Пример применения операции свёртки с одинаковым ядром без ReflectionPad2D (а) и с ReflectionPad2D (б)

В качестве последнего слоя энкодера используется слой BatchNormalization (Пакетная нормализация), который применяется для регуляризации коэффициентов сети. Значения обрабатываются таким образом, чтобы иметь нулевое математическое ожидание и единичную дисперсию [11].

Слой Add (Сумма) используется для создания соединения между слоями энкодера и декодера. При работе с глубокими нейронными сетями градиенты, распространяющиеся по сети, становятся меньше, что приводит к незначительным изменениям весов. Основным преимуществом

использования данных слоёв заключается в том, что они позволяют снизить влияние проблемы затухающего градиента на сеть [12].

Наборы изображений и параметры сети

Перед началом обучения были сформированы два набора изображений: обучающий и проверочный. Все изображения были представлены в формате PNG и закодированы в соответствии с цветовой моделью CIELAB. Обучающий набор состоит из 540 изображений, проверочный из 10. В процессе обучения на выход сети подавались вырезанные по случайной координате компоненты цветных изображений a и b размером 256×256 пикселей. На вход сети подавались эти же данные, но с одной компонентой L , ответственной за светлоту изображения. С 50% вероятностью изображения подвергались операции зеркального отображения по горизонтали (рис.5).



Рис. 5. – Пример дополнения входных данных

Были использованы следующие основные параметры обучения: количество эпох $E = 4000$, размер выборки в итерации $B = 32$.

Значения функций потерь

Значения функции потерь A_{loss} вычислялись на основе суммы среднеквадратической ошибки фильтров 21 слоя модели VGG и значений функции метрики SSIM:

$$l_s = C(x, y); I_A = C(x, A(x)),$$
$$V = MSE(VGG(l_s), VGG(I_A)),$$
$$A_{loss} = V + (1 - SSIM(A(x), y)),$$

где l_s – объединенные каналы Lab оригинального изображения, I_A – объединенные каналы Lab вывода модели, C – функция объединения каналов, A – модель автокодировщика, x – цветовая компонента L , y – цветовые компоненты a и b , MSE – среднеквадратическая ошибка, VGG – модель VGG, $SSIM$ – функция метрики SSIM.

Были получены следующие значения потерь наилучшей контрольной точки по обучающему и проверочному набору: $\min(loss) = 0,116$; $\min(val_loss) = 0,266$ (рис.6.).

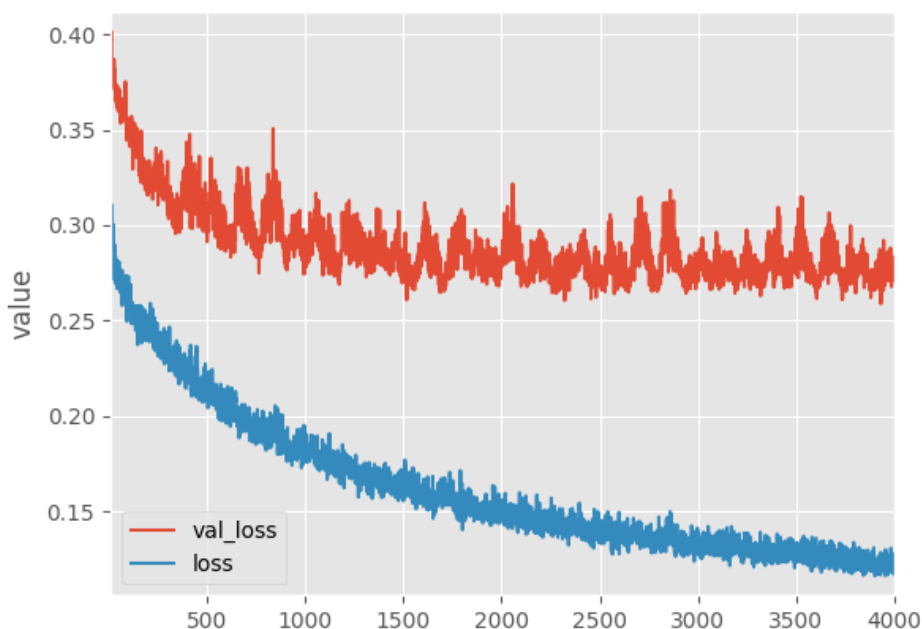


Рис. 6. – Графики значений функций потерь по обучающему ($loss$) и проверочному набору (val_loss)

Результаты анализа

Для оценки качества вывода модели использованы следующие метрики: PSNR, SSIM [13], средняя квадратическая ошибка выводов 21 слоя модели VGG. На рис.7 показаны примеры колоризации двух изображений.



Рис. 7. – Проверочные изображения (а) и вывод модели (б)

В таблице № 3 приведены результаты сравнения качества оригинальных цветных изображений с монохромными изображениями I_G и выводами модели I_A по каждому изображению из проверочного набора при помощи метрик PSNR, SSIM и VGG (MSE).

Таблица № 3

Результаты анализа

№	PSNR		SSIM		VGG (MSE)	
	I_G	I_A	I_G	I_A	I_G	I_A
1	16.686	24.140	0.793	0.876	0.0211	0.0059
2	22.667	25.057	0.898	0.930	0.0162	0.0051
3	17.523	23.763	0.819	0.888	0.0149	0.0047
4	15.830	20.099	0.774	0.846	0.0092	0.0026
5	16.648	22.590	0.793	0.855	0.0215	0.0119
6	18.535	24.295	0.855	0.904	0.0140	0.0063
7	21.929	22.935	0.844	0.890	0.0263	0.0070
8	20.581	21.591	0.832	0.876	0.0274	0.0104
9	20.702	21.896	0.878	0.885	0.0256	0.0149
10	18.921	20.988	0.845	0.867	0.0298	0.0121
Mean	19.002	22.735	0.833	0.882	0.0206	0.0081

Заключение

По результатам анализа коэффициентов SSIM и VGG (MSE), реализованная модель автокодировщика позволяет преобразовывать монохромные изображения в цветовое пространство CIELAB, при этом улучшая значения этих метрик. По визуальным оценкам выводы модели имеют менее насыщенные цвета в сравнении с оригинальными изображениями и способны придавать различные цветовые оттенки различным объектам.

Дальнейшим направлением исследования является создание улучшенной архитектуры сети и функции потерь для более точного цветового разделения различных объектов изображений и улучшения их насыщенности.

Литература

1. Bank D., Koenigstein N., Giryas R. Autoencoders // Machine learning for data science handbook: data mining and knowledge discovery handbook. 2023. pp. 353-374.
2. Pihlgren G.G., Nikolaidou K., Chhipa P.C., Abid N., Saini R., Sandin F., Liwicki M. A Systematic Performance Analysis of Deep Perceptual Loss Networks: Breaking Transfer Learning Conventions, 2023 URL: doi.org/10.48550/arXiv.2302.04032
3. Zini S., Bianco S., Schettini R. Deep Residual Autoencoder for Blind Universal JPEG Restoration // IEEE Access. 2020. V. 8. pp. 63283-63294.
4. Collin A.S., De Vleeschouwer C. Improved anomaly detection by training an autoencoder with skip connections on images corrupted with stain-shaped noise // 2020 25th International Conference on Pattern Recognition (ICPR). 2021. pp. 7915-7922.

5. Lai Z., Sun H., Tian R., Ding N., Wu Z., Wang Y. Rethinking skip connections in encoder-decoder networks for monocular depth estimation, 2022 URL: doi.org/10.48550/arXiv.2208.13441

6. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005. 1066 с.

7. Сифоров В.И., Ярославский Л.П. Адаптивные методы обработки изображений. М.: Наука, 1988. 244 с.

8. Акинин М.В., Никифоров М.Б., Таганов А.И. Нейросетевые системы искусственного интеллекта в задачах обработки изображений. М.: Горячая линия – Телеком, 2015. 154 с.

9. Smith T., Guild J. The C.I.E. colorimetric standards and their use. // Transactions of The Optical Society. 1931. V. 33. pp. 73-134.

10. Murase R., Suganuma M., Okatani T. How can CNNs use image position for segmentation? 2020 URL: doi.org/10.48550/arXiv.2005.03463

11. Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift // International conference on machine learning. 2015. pp. 448-456.

12. He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition // Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. pp. 770-778.

13. Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P. Image Quality Assessment: From Error Visibility to Structural Similarity // IEEE Transactions on Image Processing. 2004. V. 13. № 4. pp. 600-612.

References

1. Bank D., Koenigstein N., Giryes R. Machine learning for data science handbook: data mining and knowledge discovery handbook. 2023. pp. 353-374.

2. Pihlgren G.G., Nikolaidou K., Chhipa P.C., Abid N., Saini R., Sandin F., Liwicki M. A Systematic Performance Analysis of Deep Perceptual Loss

Networks: Breaking Transfer Learning Conventions, 2023. URL: doi.org/10.48550/arXiv.2302.04032

3. Zini S., Bianco S., Schettini R. IEEE Access. 2020. V. 8. pp. 63283-63294.

4. Collin A.S., De Vleeschouwer C. 2020 25th International Conference on Pattern Recognition (ICPR). 2021. pp. 7915-7922.

5. Lai Z., Sun H., Tian R., Ding N., Wu Z., Wang Y. Rethinking skip connections in encoder-decoder networks for monocular depth estimation, 2022 URL: doi.org/10.48550/arXiv.2208.13441.

6. Gonsales R., Vuds R. Tsifrovaya obrabotka izobrazheniy [Digital image processing]. M.: Tekhnosfera, 2005. 1066 p.

7. Siforov V.I., Yaroslavskij L.P. Adaptivny`e metody` obrabotki izobrazhenij [Adaptive methods of image processing]. M.: Nauka, 1988. 244 p.

8. Akinin M.V., Nikiforov M.B., Taganov A.I. Nejrosetevy`e sistemy` iskusstvennogo intellekta v zadachax obrabotki izobrazhenij [Artificial intelligence and neural networks in computer vision tasks]. M.: Goryachaya liniya – Telekom, 2015. 154 p.

9. Smith T., Guild J. Transactions of The Optical Society. 1931. V. 33. pp. 73-134.

10. Murase R., Suganuma M., Okatani T. How can CNNs use image position for segmentation? 2020. URL: doi.org/10.48550/arXiv.2005.03463

11. Ioffe S., Szegedy C. International conference on machine learning. 2015. pp. 448-456.

12. He K., Zhang X., Ren S., Sun J. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. pp. 770-778.

13. Wang Z., Bovik A.C., Sheikh H.R., Simoncelli E.P. IEEE Transactions on Image Processing. 2004. V. 13. № 4. pp. 600-612.

Дата поступления: 11.05.2024

Дата публикации: 22.06.2024