

УДК 681.518.2

DOI: 10.46960/1816-210X\_2022\_1\_43

## РЕСТАВРАЦИЯ И КОЛОРИЗАЦИЯ СТАРЫХ ФОТОГРАФИЙ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

**О.П. Тимофеева**ORCID: 0000-0002-1935-7697 e-mail: [optimofeeva@mail.ru](mailto:optimofeeva@mail.ru)Нижегородский государственный технический университет им. Р.Е. Алексеева  
*Нижний Новгород, Россия***А.Е. Савкин**ORCID: 0000-0003-4826-9087 e-mail: [sae.20@bk.ru](mailto:sae.20@bk.ru)Нижегородский государственный технический университет им. Р.Е. Алексеева  
*Нижний Новгород, Россия***Е.Р. Тимофеев**ORCID: 0000-0001-5703-3784 e-mail: [ertimofeev1@gmail.com](mailto:ertimofeev1@gmail.com)Нижегородский государственный технический университет им. Р.Е. Алексеева  
*Нижний Новгород, Россия***Т.И. Балашова**ORCID: 0000-0002-1328-3228 e-mail: [tibalashova@mail.ru](mailto:tibalashova@mail.ru)Нижегородский государственный технический университет им. Р.Е. Алексеева  
*Нижний Новгород, Россия***Е.В. Сидорова**ORCID: 0000-0002-6828-1625 e-mail: [sidorova-elena@yandex.ru](mailto:sidorova-elena@yandex.ru)Нижегородский государственный технический университет им. Р.Е. Алексеева  
*Нижний Новгород, Россия*

Рассматривается подход к решению задачи восстановления и раскрашивания старых фотографий на основе комбинации методов машинного обучения и традиционных подходов к построению алгоритмов работы с изображениями с целью сохранения исторического наследия. Рассмотрены методы выявления физических дефектов на фотографиях и способы их устранения, проведены сравнения методов колоризации изображений и увеличения их разрешения по скорости и качеству выполнения поставленной задачи. Разработан метод асинхронной обработки видеоряда для ускорения работы программы.

**Ключевые слова:** компьютерное зрение, нейронные сети, колоризация и реставрация фотографий, OpenVINO, OpenCV.

**ДЛЯ ЦИТИРОВАНИЯ:** Тимофеева, О.П. Реставрация и колоризация старых фотографий с использованием нейронных сетей / О.П. Тимофеева, А.Е. Савкин, Е.Р. Тимофеев, Т.И. Балашова, Е.В. Сидорова // Труды НГТУ им. Р.Е. Алексеева. 2022. № 1. С. 43-53. DOI: 10.46960/1816-210X\_2022\_1\_43

## RESTORATION AND COLORIZATION OF OLD PHOTOGRAPHS USING NEURAL NETWORKS

**O.P. Timofeeva**ORCID: 0000-0002-1935-7697 e-mail: [optimofeeva@mail.ru](mailto:optimofeeva@mail.ru)Nizhny Novgorod state technical university n.a. R.E. Alekseev  
*Nizhny Novgorod, Russia*

**A.E. Savkin**ORCID: **0000-0003-4826-9087** e-mail: **sae.20@bk.ru**Nizhny Novgorod state technical university n.a. R.E. Alekseev  
*Nizhny Novgorod, Russia***E.R. Timofeev**ORCID: **0000-0001-5703-3784** e-mail: **ertimofeev1@gmail.com**Nizhny Novgorod state technical university n.a. R.E. Alekseev  
*Nizhny Novgorod, Russia***T.I. Balashova**ORCID: **0000-0002-1328-3228** e-mail: **tibalashova@mail.ru**Nizhny Novgorod state technical university n.a. R.E. Alekseev  
*Nizhny Novgorod, Russia***E.V. Sidorova**ORCID: **0000-0002-6828-1625** e-mail: **sidorova-elena@yandex.ru**Nizhny Novgorod state technical university n.a. R.E. Alekseev  
*Nizhny Novgorod, Russia*

**Abstract.** Approach to solving a problem of restoring and coloring of old photographs based on a combination of computer-assisted instruction methods and traditional approaches to building algorithms for work with images in order to preserve historical heritage, is considered. Methods of detecting of physical defects in photographs and ways to eliminate them are considered, comparisons of images colorization methods, as well as magnification of resolution thereof in terms of speed and quality of the task, are carried out. Method of video sequence asynchronous processing is developed to speed up the program in operation.

**Key words:** computer vision, neural networks, colorization and restoration of photographs.

**FOR CITATION:** O.P. Timofeeva, A.E. Savkin, E.R. Timofeev, T.I. Balashova, E.V. Sidorova. Restoration and colorization of old photographs using neural networks. Transactions of NNSTU n.a. R.E. Alekseev. 2022. №1. Pp. 43-53. DOI: 10.46960/1816-210X\_2022\_1\_43

**Введение**

Создание машины, способной думать и принимать решения подобно человеку было мечтой всех ученых во все времена. Раньше за неимением технологий люди дрессировали животных для выполнения определенных задач или просто для возможности безопасного взаимодействия с ними. Однако с недавних пор у человека появилась возможность создать искусственный интеллект, способный выполнять вычисления событий, не поддающихся расчетам по заранее заготовленным формулам, на что раньше были способны лишь живые организмы. В настоящее время машинное обучение является одним из самых актуальных и быстро развивающихся направлений искусственного интеллекта в IT-сфере, а компьютерное зрение находит практическое применение во многих областях деятельности человека. Наиболее прогрессивным направлением в создании «компьютерного зрения» является применение нейронных сетей из-за их высокой точности и скорости работы.

В настоящей работе ставится задача исследования алгоритмов компьютерного зрения для создания автоматизированной системы «Time machine», способной восстанавливать, раскрашивать старые фотографии и видео, а также увеличивать их разрешение. Эта задача является одной из наиболее востребованных в компьютерном зрении в настоящее время, о чем свидетельствует большое количество публикаций на эту тему.

## Постановка задачи

В ходе работы требуется создать автоматизированную систему «Time machine» – программный продукт, способный восстанавливать старинные фотографии путем:

- нахождения и удаления с них различных дефектов;
- раскрашивания получившегося изображения;
- увеличения его разрешения, а также оптимизировать работу нейронных сетей для обработки видеоряда.

Для выполнения поставленной задачи требуется сравнить традиционные подходы к построению алгоритмов работы с изображениями и алгоритмы компьютерного зрения для выполнения тех же задач. Необходимо также исследовать различные фреймворки для создания, инфера (исполнения) и оптимизации нейронных сетей, сравнить популярные архитектуры по критериям качества и скорости работы.

## Стэк технологий

Для разработки системы «Time machine» был выбран язык программирования Python как один из наиболее популярных для использования при создании и сопровождении нейронных сетей. Также в качестве основного «плацдарма» для создания собственного приложения был применен инструментарий от компании Intel – OpenVINO Toolkit [1] – набор различных инструментов для сопровождения (оптимизации, валидации, оценки) и инфера (исполнения) нейронных сетей. Кроме того, использовалась библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом - OpenCV, входящая в пакет OpenVino. Для применения нейронных сетей за пределами «зоопарка» оптимизированных моделей OpenVINO были использованы фреймворки Caffe, ONNX, TensorFlow, PyTorch.

## Устранение дефектов на фотографии

В ходе решения задачи поиска и устранения дефектов было решено классифицировать их на две группы по размеру: мелкие трещины в краске и крупные дефекты. Разница между представителями этих групп заключается в происхождении. Мелкие трещины образовались под воздействием времени, которое привело к высыханию краски и, как следствие, эти дефекты трудно различимы для человеческого глаза, однако из-за плотности их расположения на изображении его качество заметно падает. Крупные же дефекты стали следствием физического воздействия на фотографию: перегибы – толстые и длинные линии, проходящие через всё изображение, царапины – одна или совокупность тонких линий, имеющих схожую направленность короткой длины, потертости – крупные эллипсовидные дефекты с рваными краями (рис. 1,а). Анализ существующих фотографий показал, что поврежденные места имеют белый цвет – цвет поверхности, на которую наносится краска. В связи с этим была предпринята попытка выявления дефектов путем нахождения на изображении пикселей со значениями близкими к белому цвету. Однако данное простое решение ожидаемо не дало удовлетворительного результата, так как не выделяло большинство мелких трещин, но выделяло все элементы схожего цвета на изображении (рис. 1,б).

В поиске решения была рассмотрена свертка изображения с небольшим ядром – оператор Собеля [2], который аппроксимирует градиент со значительной погрешностью. Другими словами, он находит границы объектов. Использование свертки выделяло на фотографии все мелкие трещины и границы крупных дефектов с высокой точностью (рис. 1,в). Однако, у данного метода был выявлен ряд недостатков, одним из которых явилось выделение границ всех объектов на фотографии. Другим недостатком стало определение областей, находящихся внутри границ крупных дефектов, так как при определении соседних

пикселей с цветами близкими к белому цвету снова выделялись, кроме самих дефектов, объекты схожих цветов, а при выявлении соседних схожих траекторий границ, выделялись области между совокупностью тонких царапин. Была предпринята попытка реставрации мелких трещин путем сравнения с соседними пикселями, но несмотря на удовлетворительное решение данной проблемы, было принято решение отказаться от данного подхода из-за размытия границ объектов на изображении при устранении дефектов.

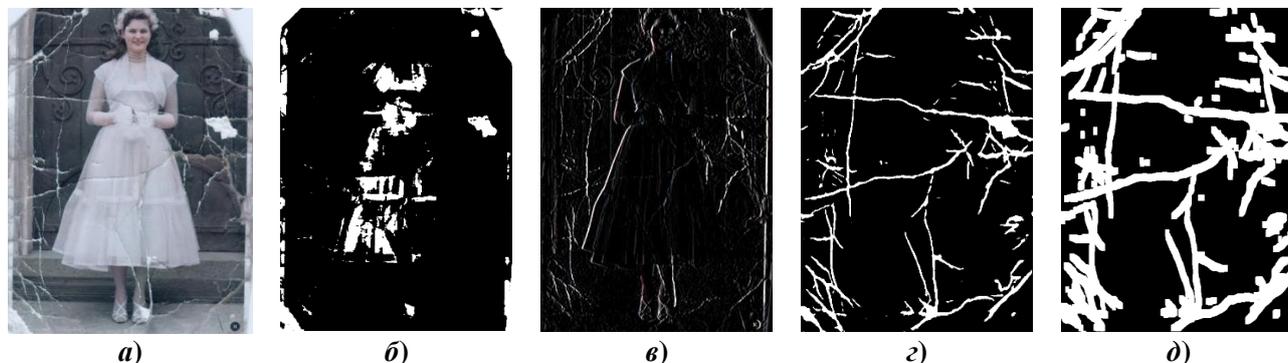


Рис. 1. Определение дефектов на фотографии

Fig. 1. Detection of defects on a photo

В связи с тем, что традиционные способы не дали желаемых результатов, решение задачи было реализовано с использованием нейронных сетей. Важным фактором в выборе сети послужило обязательное использование утилиты OpenVINO, что добавило параметр простоты конвертации в параметры выбора решения. В конечном итоге были выбраны сети *scratch\_detector* из проекта *Bring\_Old\_Photos\_Back\_to\_Life* [3] для выявления дефектов и *gmcnn-places2-tf* из *Open\_model\_zoo* [4] для их устранения. Сравнительные характеристики работы этих нейронных сетей представлены в табл. 1.

Таблица 1.

Модели поиска трещин и закрашивания дефектов

Table 1.

Models for crack detection and defect painting

Model	Work speed <sup>1</sup>
<i>scratch_detector</i> (на основе U-net)	Count: 36 iterations Duration: 68720.52 ms Latency: 7535.71 ms Throughput: 0.52 FPS <sup>2</sup> При разрешении 512x608
<i>gmcnn-places2-tf</i>	Count: 28 iterations Duration: 70695.08 ms Latency: 10157.57 ms Throughput: 0.40 FPS <sup>2</sup> При разрешении 512x680

Результатом работы нейронной сети *scratch\_detector* стало выявление крупных дефектов на фотографии с высокой точностью, однако мелкие дефекты в краске данная нейронная сеть была неспособна определить (рис. 1,з). После приведения выходных данных первой сети к формату входных данных сети *gmcnn-places2-tf* был получен результат, который не в полной мере удовлетворил ожидания, так как границы устраненных дефектов оставались различимы.

<sup>1</sup> Здесь и далее результаты производительности получены с помощью инструмента «Benchmark App», входящего в пакет OpenVINO и используемого для оценки скорости моделей нейронных сетей

<sup>2</sup> Здесь и далее FPS (Frames per second) указан с использованием процессора Intel core i5-8300h

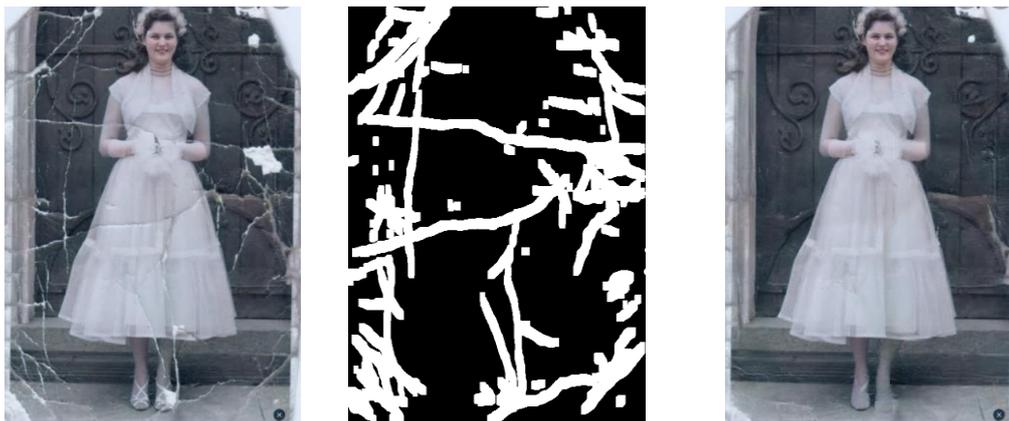


Рис. 2. Реставрация дефектов на фотографии

Fig. 2. Restoration of defects on a photo

Комбинация оператора Собеля и сети *scratch\_detector* не привела к значительному улучшению результата реставрации, – из-за проблемы оператора Собеля границы объектов снова оказались размытыми. Решением стало увеличение границ маски, полученной на выходе сети *scratch\_detector*, путем выбора соседних пикселей в область для реставрации (рис. 1,д). Несмотря на то, что это привело к предсказанию значений заранее известных нам, результат работы программы стал на порядок лучше по сравнению со всеми предыдущими попытками (рис. 2).

### Колоризация изображений

Второй этап работы – колоризация изображений (рис. 3). Эта задача является одной из сложнейших при работе с фотографиями. Люди, занимающиеся восстановлением фото, тратят огромное количество времени на то, чтобы раскрасить их. Задача усложняется ещё и тем, что достаточно трудно подобрать правильные цвета - для этого нужно хорошо знать историю и использовать дошедшие до нас артефакты того времени, когда была сделана фотография. До появления нейронных сетей автоматизировать эту задачу было практически невозможно из-за её огромной сложности. Однако современные алгоритмы компьютерного зрения позволяют сделать это. В работе были исследованы различные популярные нейросетевые модели, как из открытого зоопарка моделей OpenVINO, так и за его пределами (табл. 2).



Рис. 3. Демонстрация работы модели *Colorization-siggraph*

Рис. 3. Demonstration of the model *Colorization-siggraph*

Таблица 2

## Модели колоризации фотографий

Table 2.

## Models of colorization of photographs

Model	MOS	Work speed
<i>Deoldify ONNX</i>	+++	Count: 16 iterations Duration: 82994.15 ms Latency: 20610.55 ms Throughput: 0.19 FPS При разрешении 512x512
<i>Deoldify_stable</i>	++	Count: 24 iterations Duration: 88209.61 ms Latency: 14584.08 ms Throughput: 0.27 FPS При разрешении 512x512
<i>Deoldify_video</i>	+	Count: 20 iterations Duration: 75107.52 ms Latency: 15067.23 ms Throughput: 0.27 FPS При разрешении 512x512
<i>Deoldify_art</i>	++	Count: 16 iterations Duration: 86643.72 ms Latency: 21637.18 ms Throughput: 0.18 FPS При разрешении 512x512
<i>Colorization-v2</i> (для FP16 моделей)	+	Count: 268 iterations Duration: 61493.61 ms Latency: 908.73 ms Throughput: 4.36 FPS При разрешении 256x256
<i>Colorization-v2 Caffe<sup>3</sup></i>	+	Count: 340 iterations Duration: 61038.82 ms Latency: 712.75 ms Throughput: 5.57 FPS При разрешении 224x224
<i>Colorization-siggraph<sup>3</sup></i>	+++	Count: 156 iterations Duration: 62477.90 ms Latency: 1553.37 ms Throughput: 2.50 FPS При разрешении 256x256

Модели, представленные в табл. 2, были сконвертированы и оптимизированы для работы с OpenVINO с помощью встроенного инструмента «Model Optimizer»; для оценки качества работы сетей была использована метрика MOS – Mean opinion score (средняя оценка мнений), представленная для моделей в виде плюсов: чем их больше, тем модель лучше справляется с колоризацией изображений. Такая субъективная метрика была выбрана по той причине, что объективно оценить результаты колоризации довольно сложно. В целом все исследуемые модели показали хорошие или отличные результаты в скорости и качестве работы. Нейросетевые технологии работают существенно быстрее любого человека, однако и результаты работы профессионала и нейросетевой модели будут различаться. Большое преимущество искусственного интеллекта в том, что можно использовать различные модели, каждая из которых будет иметь свой «стиль» раскрашивания изображений, что позволит бесплатно и быстро раскрасить старинные фотографии. Профессионалам же останется взять лучшую (по их мнению) модель из полученных и скорректировать ее.

На рис. 4 представлены результаты колоризации, полученные с помощью нейронных сетей, исследованных в работе (табл. 2). Можно заметить, что наиболее близкие к реальности цвета показывают модели *Deoldify ONNX* и *Colorization-siggraph*. Модели семейства *Deoldify* [5] (за исключением *Deoldify ONNX*) довольно схожи и используют в основном холодные (синие цвета), особенно *Deoldify\_video*. Модели семейства *Colorization* (за исключением *Colorization-siggraph*) имеют тенденцию к более теплым (желтым) цветам. Экспериментально показано, что наибольшим быстродействием обладают модели семейства *Colorization* из Open\_model\_zoo OpenVINO [4], так как они оптимизированы самой компанией Intel под собственные процессоры. Скорость работы моделей семейства *Deoldify* оставляет желать лучшего, особенно учитывая выходное разрешение изображения, которое ниже, чем у моделей из зоопарка OpenVINO (табл. 2).

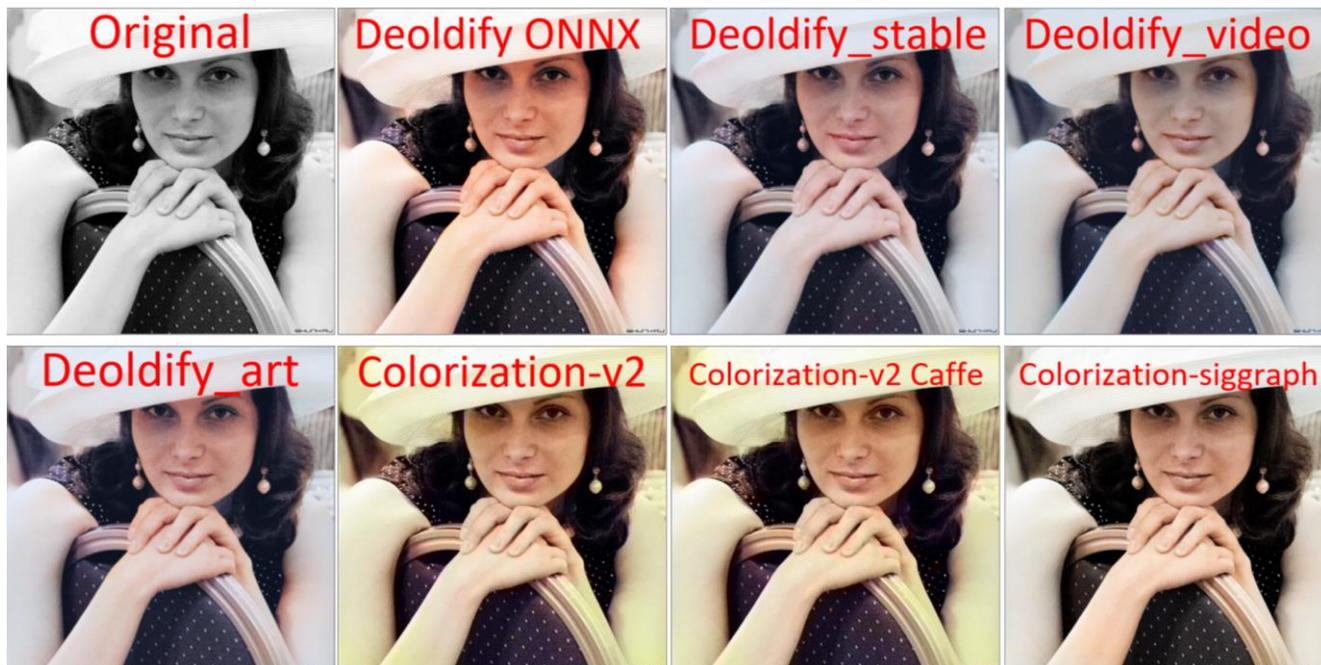


Рис. 4. Результат работы моделей колоризации фотографий

Fig. 4. Result of photo colorization models in operation

#### Увеличение разрешения

Поскольку старые фотографии обычно имеют малые размеры, в работе была поставлена задача увеличения масштаба этого изображения без потери качества (рис. 5). Классические алгоритмы интерполяции [6], такие как билинейная или бикубическая интерполяция, существовали задолго до появления современных нейронных сетей, однако данные алгоритмы не способны в полной мере увеличить разрешение без потери качества. В отличие от них, нейронные сети способны не просто использовать информацию с соседних пикселей, а буквально воссоздавать недостающие детали. Исследуемые в работе нейронные сети, направленные на увеличение разрешения, показывают лучший результат по сравнению с классическими видами интерполяции, даже если в связке с последними использовать алгоритмы или нейронные сети, направленные на уменьшение шума и размытия на изображении. В связи с этим было принято решение использовать именно нейросетевые модели для выполнения задачи увеличения разрешения.

Как и в случае с колоризацией, были рассмотрены как модели из зоопарка моделей OpenVINO, так и за его пределами (табл. 3). Для оценки результатов использовалась метрика PSNR – пиковое отношение сигнала к шуму (англ. peak signal-to-noise ratio). PSNR – инженерный термин, означающий соотношение между максимумом возможного значения сигнала и мощностью шума, искажающего значения сигнала [7]. Поскольку многие сигналы имеют широкий динамический диапазон, PSNR обычно измеряется в логарифмической шкале в децибелах.

Результаты работы нейронных сетей из табл. 3 представлены на рис. 6

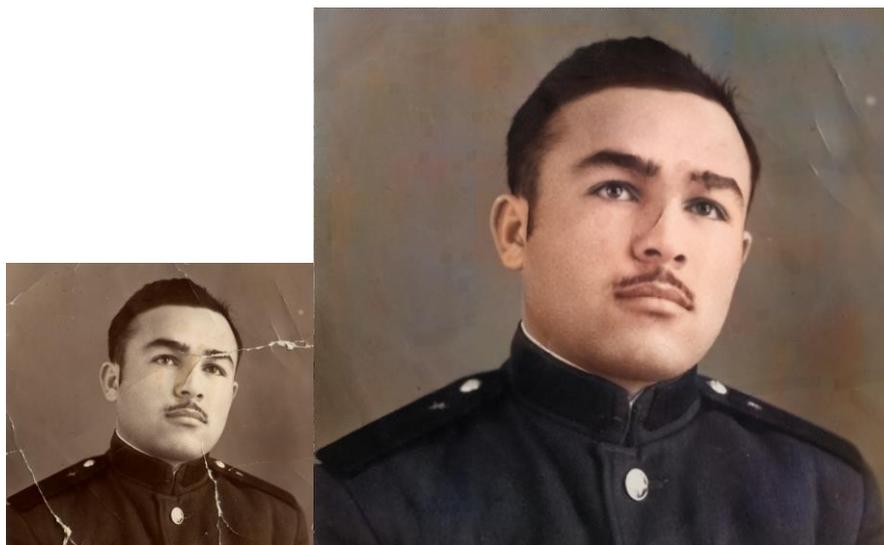


Рис. 5. Демонстрация работы программы «Time machine» от исходной фотографии до финального результата

Fig. 5. Demonstration of the «Time machine» program in operation from the original photo to the final result

Модели увеличения разрешения

Таблица 3

Models of magnification of resolution

Table 3

Model	Accuracy	Work speed
<i>single-image-super-resolution-1033</i> (для FP16 моделей)	PSNR: 40.39	Count: 364 iterations Duration: 60522.68 ms Latency: 659.88 ms Throughput: 6.01 FPS При разрешении 640x480
<i>single-image-super-resolution-1032</i> (для FP16 моделей)	PSNR: 49.62	Count: 308 iterations Duration: 61387.94 ms Latency: 788.56 ms Throughput: 5.02 FPS При разрешении 480x270
<i>EDSR</i>	PSNR: 68.18	Count: 92 iterations Duration: 65120.85 ms Latency: 2789.55 ms Throughput: 1.41 FPS При разрешении 256x256
<i>RCAN</i> (для FP16 моделей)	PSNR: 65.23	Count: 8 iterations Duration: 312597.10 ms Latency: 148520.46 ms Throughput: 0.03 FPS При разрешении 720x480
<i>SRGAN</i>	PSNR: 69.37	Count: 16 iterations Duration: 99383.44 ms Latency: 24717.16 ms Throughput: 0.16 FPS При разрешении 256x256

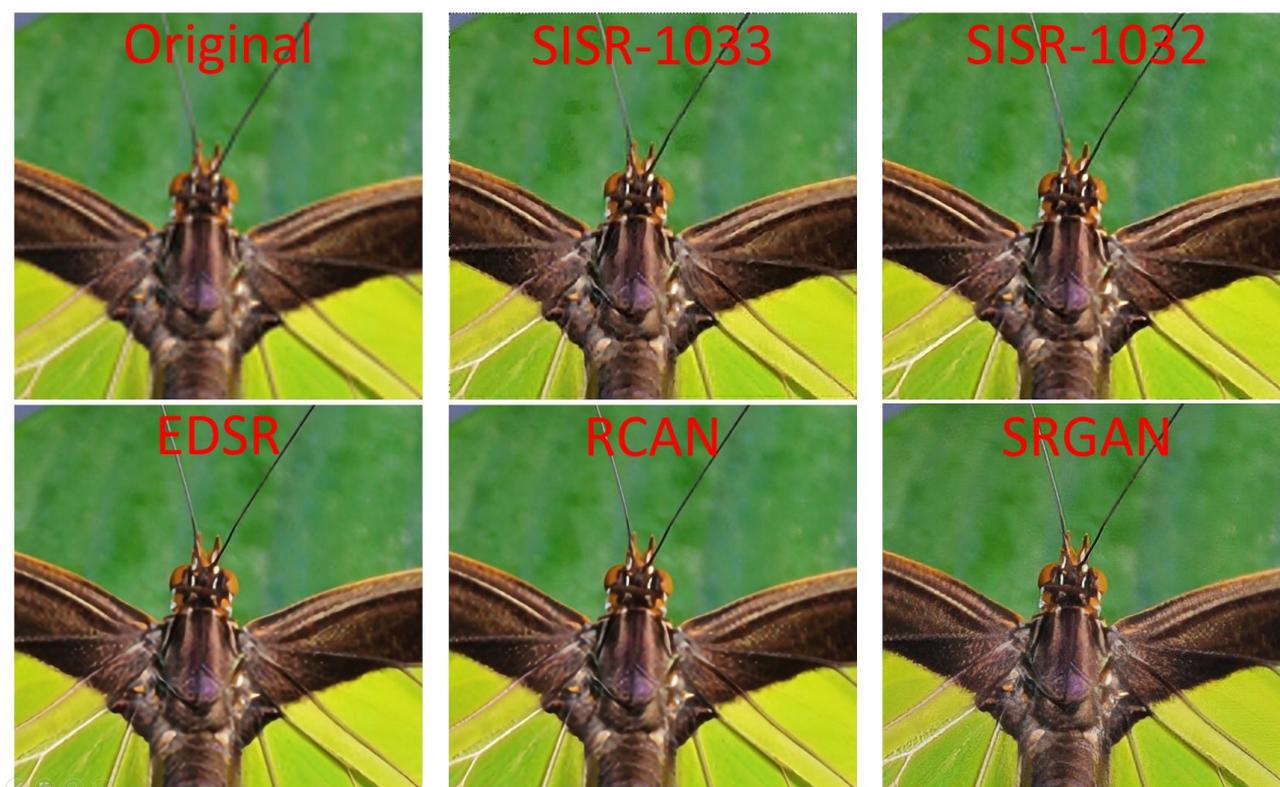


Рис. 6. Демонстрация работы моделей увеличения разрешения

Fig. 6. Demonstration of resolution magnification models in operation

Результат *single-image-super-resolution* (*SISR*) [4] виден сразу – изображение становится более четким. Если же мы сравним изображения, полученные *SISR* и *EDSR*, можем увидеть, что *EDSR* обработала усики насекомого лучше, и в целом прослеживается тенденция хорошей работы с линиями у *EDSR*. *RCAN* [8] обрабатывает линии еще лучше, они более «прямые», на них меньше искажений. Однако работа над одним изображением у этой сети занимает большое количество времени. Ещё одна нейронная сеть – генеративно-состязательная *SRGAN* [9]. Она обрабатывает линии с таким же качеством, что и *EDSR* [9]. Однако ее особенностью является то, что она сильно повышает резкость на изображении, из-за чего оно в каких-то моментах кажется более четким, а в каких-то добавляет излишние детали.

### Оптимизация работы сетей для обработки видеоряда

Все используемые в работе нейронные сети были оптимизированы или сконвертированы при помощи инструментов «Model optimizer» и «Converter», входящих в пакет инструментов OpenVINO; некоторые из сетей были дополнительно оптимизированы с помощью инструмента DL-Workbench в FP16 (для оптимизации работы веса были переведены в числа половинной точности, что практически не влияет на точность, однако существенно упрощает вычисления), что позволило улучшить показатели скорости их работы.

Исследуемые модели реставрации, колоризации и улучшения качества были также применены для обработки одного кадра видеоряда. Это привело к необходимости оптимизации не только скорости выполнения самих нейронных сетей, но и достижения оптимальности взаимодействия между ними. В разработанном приложении используется четыре нейронные сети (с возможностью выбора разных сетей для колоризации и увеличения разрешения). Были рассмотрены некоторые из способов организации взаимодействия между исследуемыми сетями [10].

Самый простой способ взаимодействия – синхронный. По очереди выполняется исполнение (инфер) каждой сети, передавая изображений от одной к другой. Такой способ имеет преимущество в плане написания кода и подходит для обработки одного изображения, но имеет очевидный минус в виде недостаточной производительности. Второй подход к организации взаимодействия сетей – использование готового класса AsyncPipeline из открытого зоопарка моделей OpenVINO. Благодаря ему удобно подавать изображения на вход сети и получать нужный результат. При этом предобработка и постобработка уже входит в класс модели, а также не нужно ждать, пока выполнится инфер. При использовании одной нейросети этот вариант выглядит замечательно. Однако в случае четырех нейросетей возникает проблема – как все это построить. Можно использовать класс AsyncPipeline несколько раз, но это приводит к сложности в восприятии кода: можно написать свой сложный класс модели для одного AsyncPipeline, но при этом необходимо продумать, как все эти модели связать внутри.

Было принято решение создать свой AsyncPipeline, суть которого заключается в создании четвертого экземпляра классов моделей. Каждый экземпляр на этапе своей инициализации запускает новый поток, который следит за входом сети и решает, когда вызывать инфер. Таким образом в коде четыре экземпляра классов, которые уже полностью готовы к работе. Затем запускается цикл, который будет проверять, можно ли подавать на вход первой сети изображение и в случае, если можно – делать это. Результат можно получить двумя способами - либо в цикле проверять выход последней модели, либо добавить callback, который вызывается всякий раз, когда изображение обработалось. Для устройств, имеющих графическое ядро от компании Intel, в работе предусмотрено распределение предобработки данных и вычисления нейронных сетей между процессором и видеокартой для увеличения скорости обработки данных.

### Выводы

Создан программный продукт «Time machine», позволяющий восстанавливать, раскрашивать и увеличивать разрешение старых фотографий и видео с применением компьютерного зрения. Исследованы различные архитектуры нейронных сетей, проведено их сравнение между собой и с традиционными алгоритмами. Разработанный программный продукт позволяет с довольно высокой скоростью и отличным качеством восстанавливать различные фотографии.

Репозиторий GitHub с исходниками программы – [https://github.com/eaidova/time\\_machine](https://github.com/eaidova/time_machine).

### Библиографический список

1. Intel Distribution of OpenVINO Toolkit [Электронный ресурс]. – Режим доступа: <https://software.intel.com/content/www/us/en/develop/tools/openvino-toolkit.html> (дата обращения 05.07.2021).
2. Алгоритмы выделения контуров изображений: своя игра [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/114452/> (дата обращения 07.07.2021).
3. Old Photo Restoration (Official PyTorch Implementation) [Электронный ресурс]. – Режим доступа: <https://github.com/microsoft/Bringing-Old-Photos-Back-to-Life> (дата обращения 08.07.2021).
4. OpenVINO Toolkit - Open Model Zoo repository [Электронный ресурс]. – Режим доступа: [https://github.com/openvinotoolkit/open\\_model\\_zoo](https://github.com/openvinotoolkit/open_model_zoo) (дата обращения 05.07.2021).
5. DeOldify [Электронный ресурс]. – Режим доступа: <https://github.com/jantic/DeOldify> (дата обращения 17.07.2021).
6. Методы интерполяции и аппроксимации [Электронный ресурс]. – Режим доступа: [https://portal.tpu.ru/SHARED/m/MBB/uchebnaya\\_rabota/Model/Tab/Interp\\_app.pdf](https://portal.tpu.ru/SHARED/m/MBB/uchebnaya_rabota/Model/Tab/Interp_app.pdf) (дата обращения 16.07.2021).

- 
7. PSNR и SSIM [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/126848/> (дата обращения 23.07.2021).
  8. Image Super-Resolution Using Very Deep Residual Channel Attention Networks [Электронный ресурс]. – Режим доступа: <https://github.com/yulunzhang/RCAN> (дата обращения 13.07.2021).
  9. Single Image Super-Resolution with EDSR, WDSR and SRGAN [Электронный ресурс]. – Режим доступа: <https://github.com/krasserm/super-resolution> (дата обращения 19.07.2021).
  10. Параллелизм против многопоточности против асинхронного программирования [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/337528/> (дата обращения 26.07.2021).

*Дата поступления  
в редакцию: 23.09.2021*