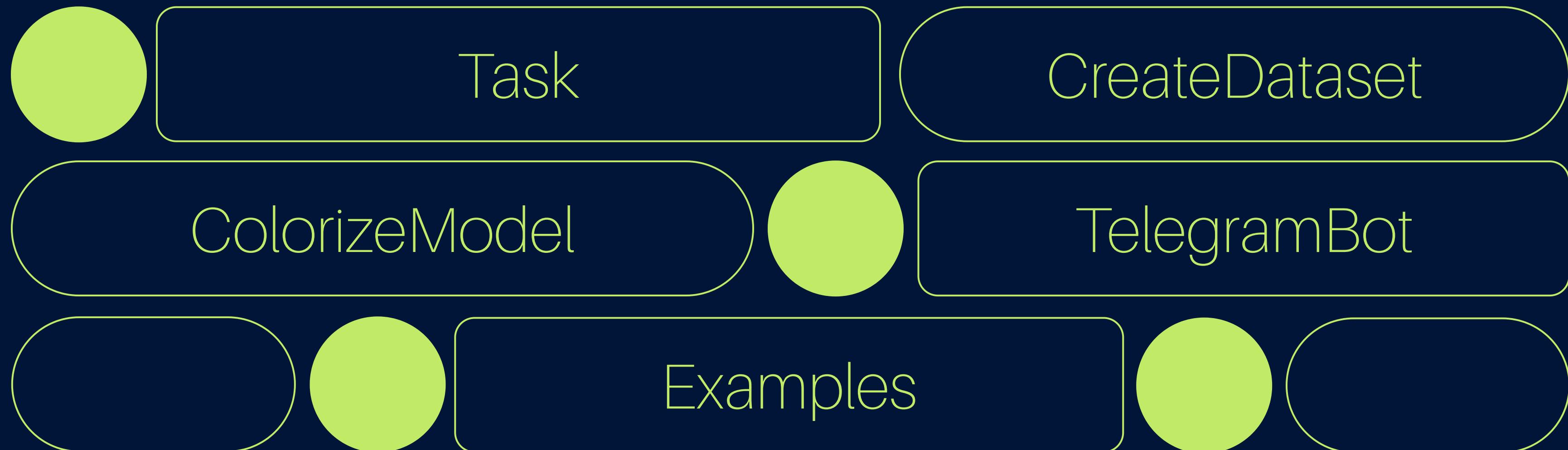


# ColorizeImage

Topper & Glazovskaya

Theoretical computer science 23

# Agenda



# Task

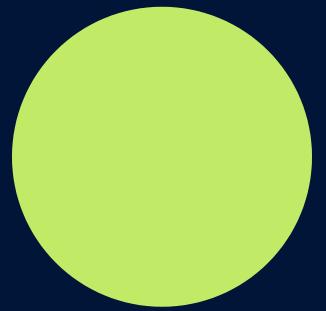
Проблема раскрашивания черно-белых изображений с помощью нейронных сетей заключается в том, что нейронная сеть должна научиться распознавать объекты на черно-белых изображениях и правильно раскрашивать их в соответствии с цветом, который был бы у объекта настоящего цветного изображения.

В ходе этого проекта были выполнены такие задачи как:

- создание датасета
- анализ многих моделей
- создание телеграм бота



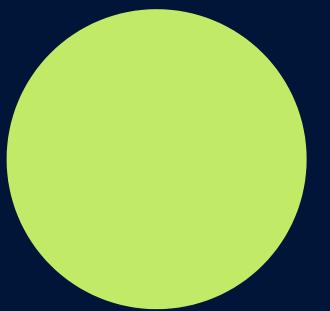
# CreateDataset



FindExistedDataset

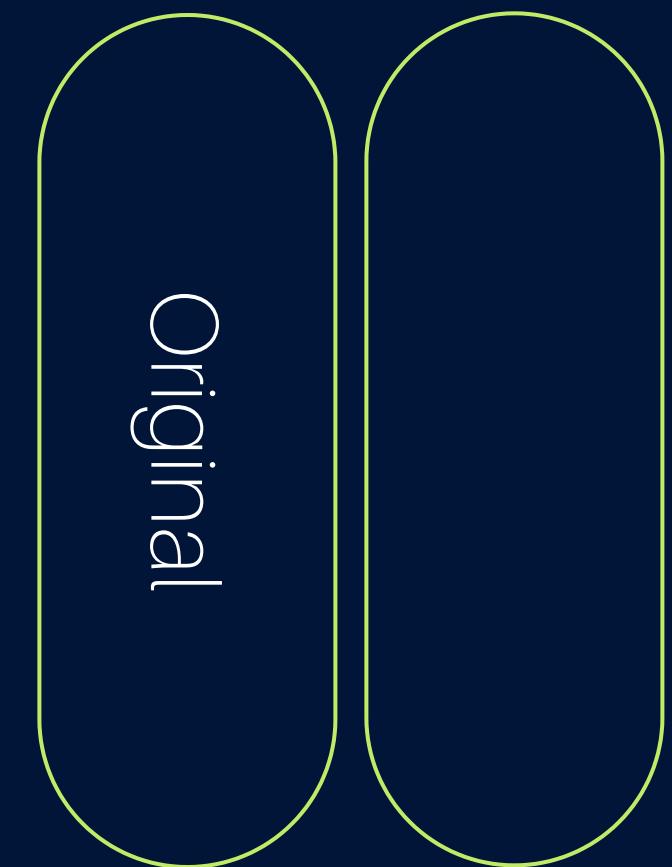
Создали программу, которая обрабатывает датасет из более чем 1700 картин и переводит их в черно-белый цвет. Так же разделили на тестовые и тренировочные наборы в парах оригинал-чб версия

После сравнения нескольких датасетов из картин был выбран - WikiArt. Основное его преимущество - разделения по жанрам. Вручную из классов "Реализм" и "Новый реализм" были отобраны пейзажи (также были исключены пейзажи-эскизы или в графике)

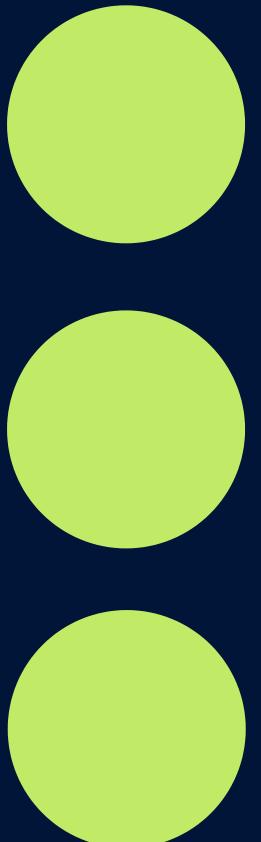


GrayscaleDataset

<https://colab.research.google.com/drive/1nr2aydWrLnROG8pOWiWQEJuKdm7AFUub?usp=sharing>



Grayscale



# ColorizeModel

# GAN



BW

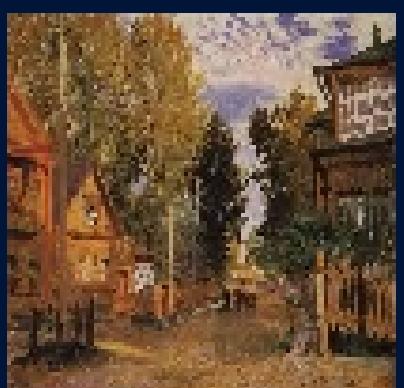
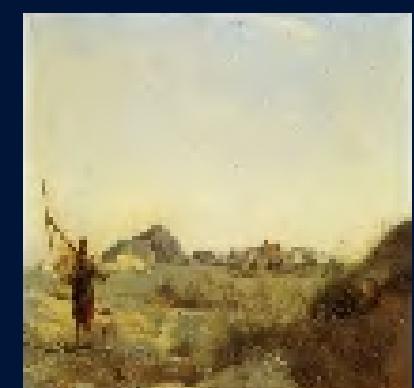
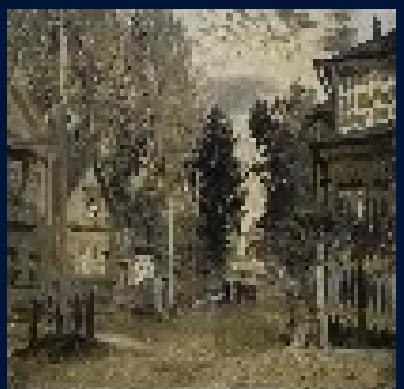
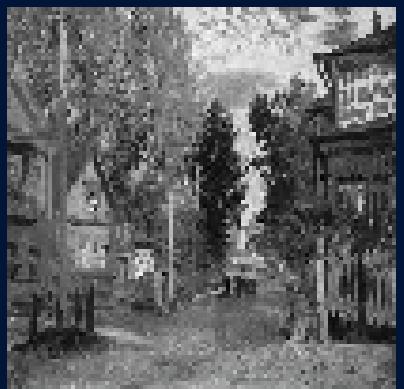
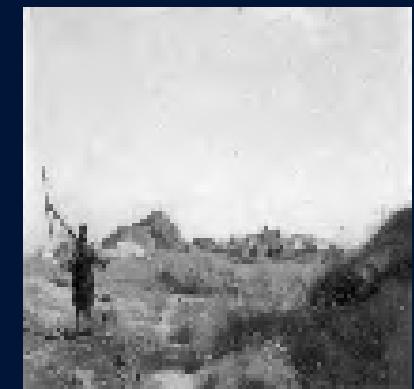


GENERATED



## ORIGINAL

## AUTOENCODER + RESNET

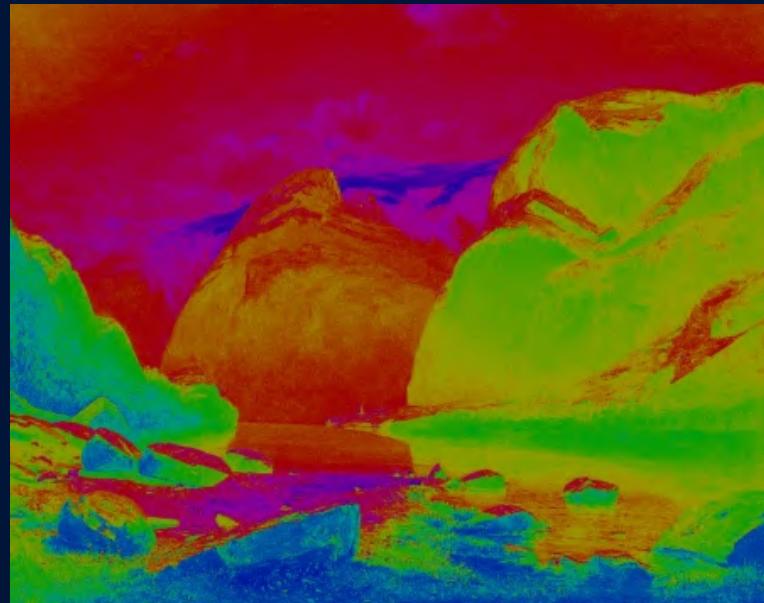


# OpenCV

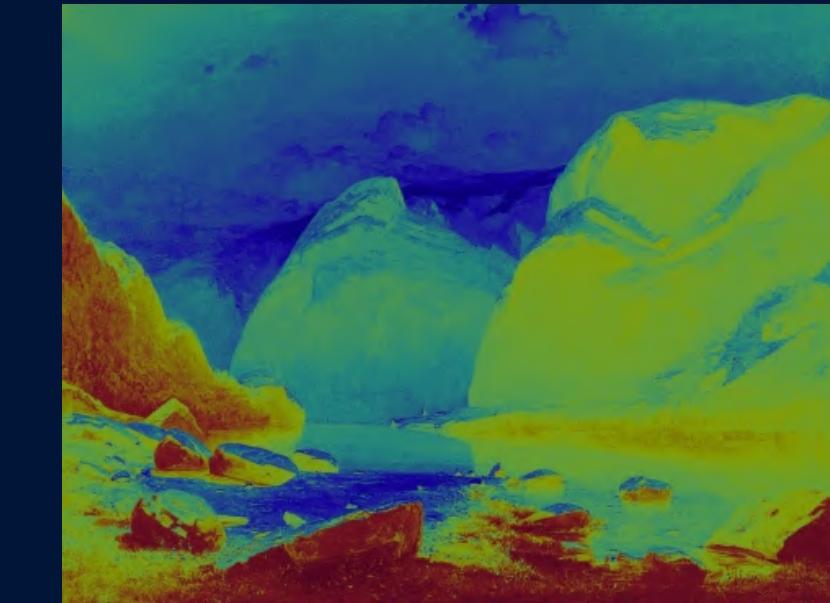


ORIGINAL

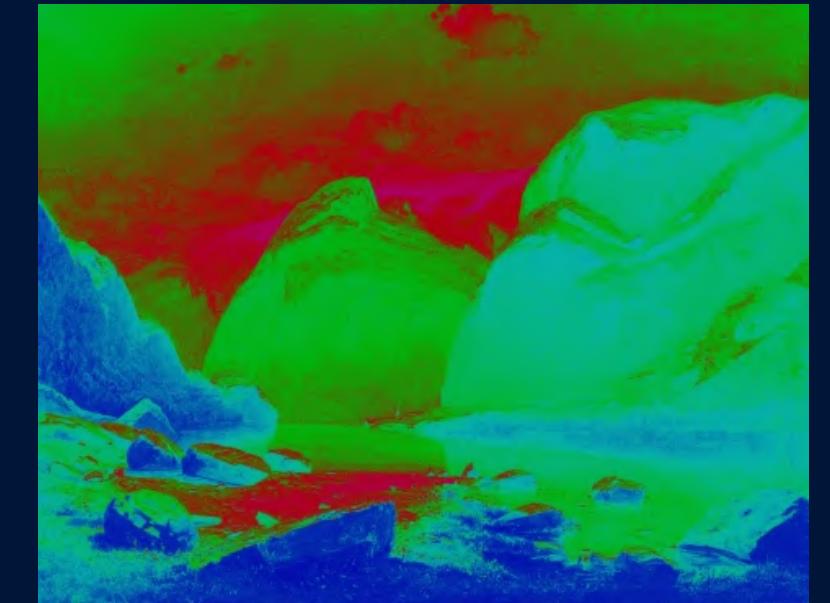
+



RAINBOW



PLASMA



TURBO

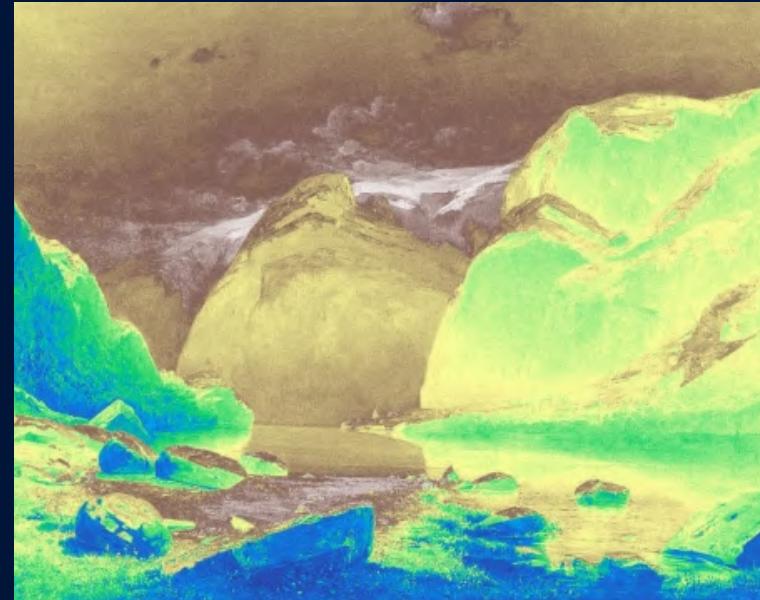
- Можно легко изменять параметры алгоритмов и экспериментировать с разными параметрами.
- Имеет высокую скорость работы и может обрабатывать большое количество данных в режиме реального времени.
- Удобный и простой в использовании интерфейс для обработки изображений.

- Может потребовать значительных затрат вычислительных ресурсов и времени.
- Сложно подобрать color\_map и соответствующие веса для параметров.
- Возможны ошибки и неточности в результатах работы алгоритмов, особенно если на изображении содержатся объекты с необычной формой или сложной текстурой.

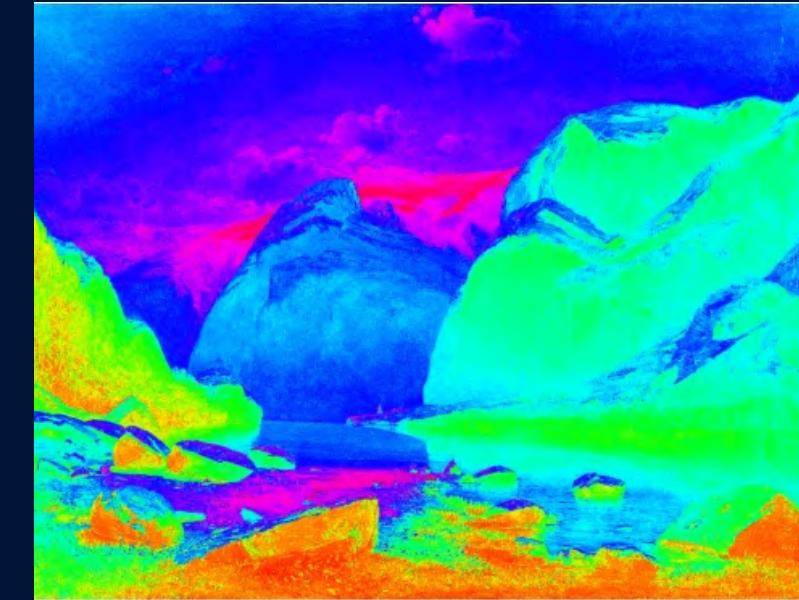
# МАТПЛОТЛИБ



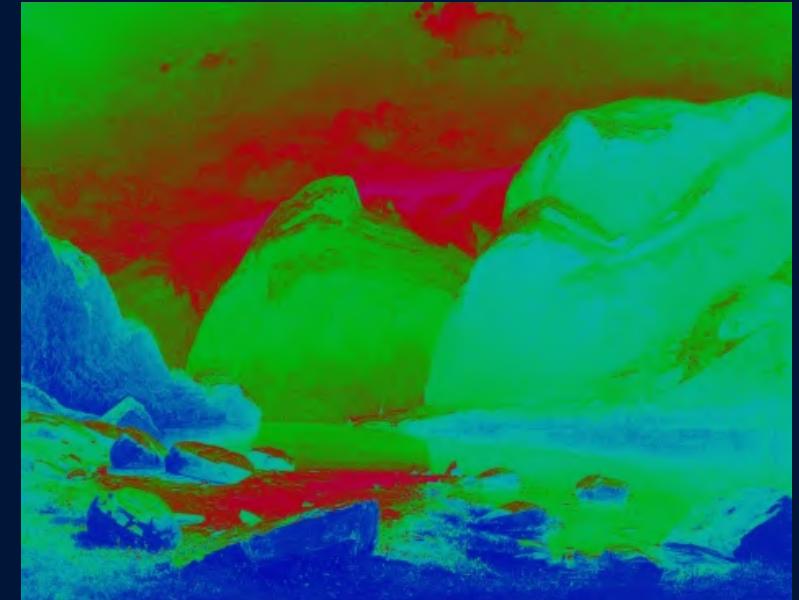
ORIGINAL



TERRAIN



HSV

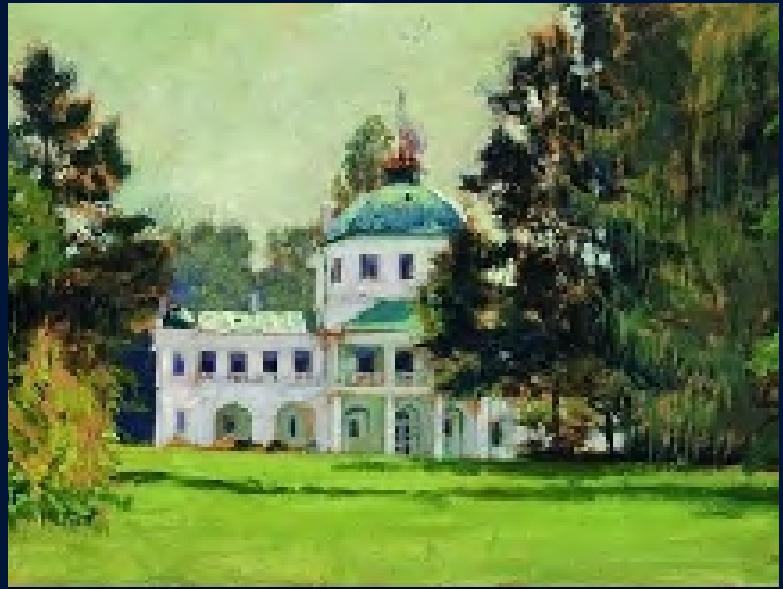


TURBO

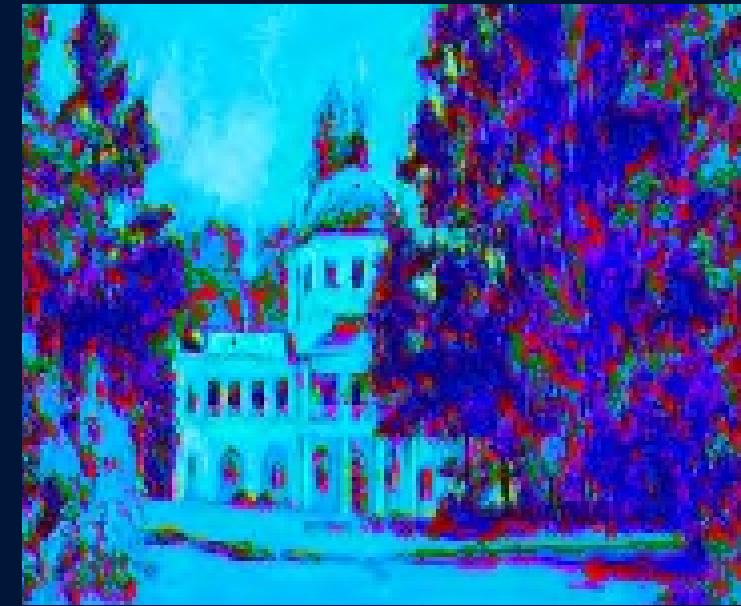
- Возможность быстро и просто изменять цветовую схему без необходимости более детальной работы с цветами.
- Удобство визуализации цветного изображения

- Однако, выбор цветовой схемы является важным и сложным аспектом в визуализации данных, и использование get\_stars может не всегда позволить получить оптимальный результат. В некоторых случаях может потребоваться большее управление цветами.

# CNN



ORIGINAL

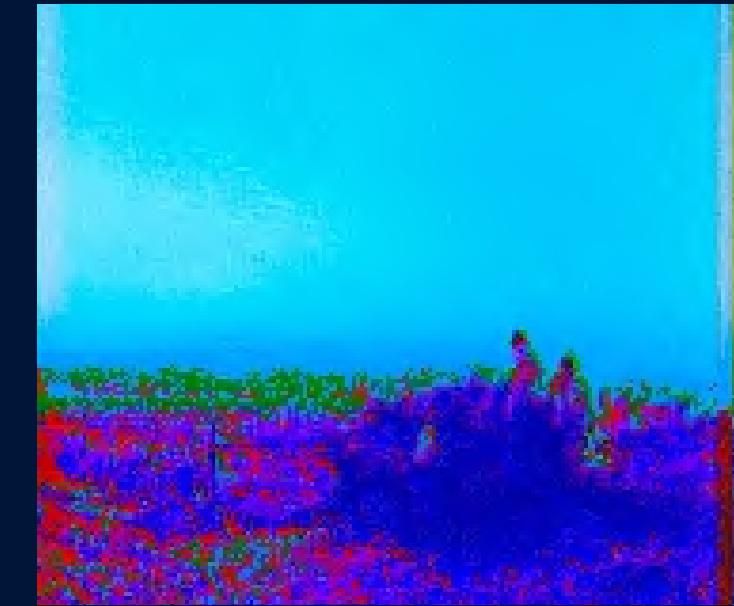


MODEL

+



ORIGINAL



MODEL

-

CNN может быть использован для задачи раскрашивания черно-белых изображений. В этом случае он может определить, какие цвета следует добавить в каждую часть изображения, чтобы получить цветную версию. Это происходит через обучение на большой выборке цветных и черно-белых изображений.

# AUTOENCODER + RESNET



12 epoch



8 epoch



# AUTOENCODER + UNET



BW



AUTOENCODER



BW



AUTOENCODER

- Позволяет автоматизировать процесс окрашивания и сохранить информацию о деталях и форме изображения.

- Для обучения и использования такой модели необходимо иметь достаточное количество данных и достаточную вычислительную мощность, так как эта модель достаточно сложна.

# GAN



BW

+



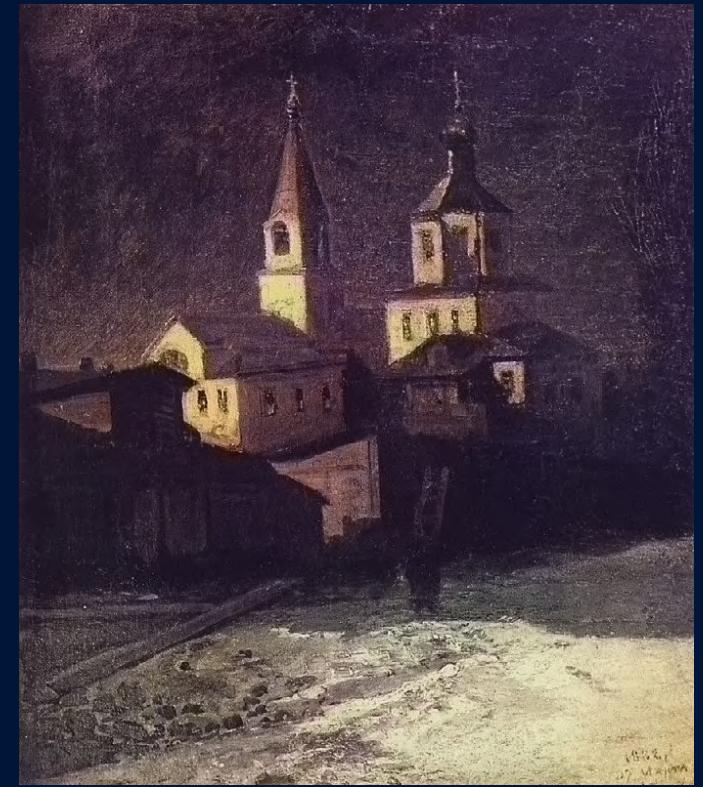
GENERATED



ORIGINAL

- Возможность создавать изображения высокого качества.
- Также GAN может учитывать контекст и структуру изображения, что позволяет генерировать цветные изображения, которые выглядят естественно.
- Способна использовать контекст и детализацию изображения, чтобы произвести наиболее правдоподобный результат.

- Проблема сходства между сгенерированными и оригиналами изображениями.
- При обучении сети, дискриминатор может ошибку высококачественное, но просто повторение входного изображения, в то время как генератор может научиться создавать соответствующее цветное изображение, не отражающее оригинальный контент изображения.



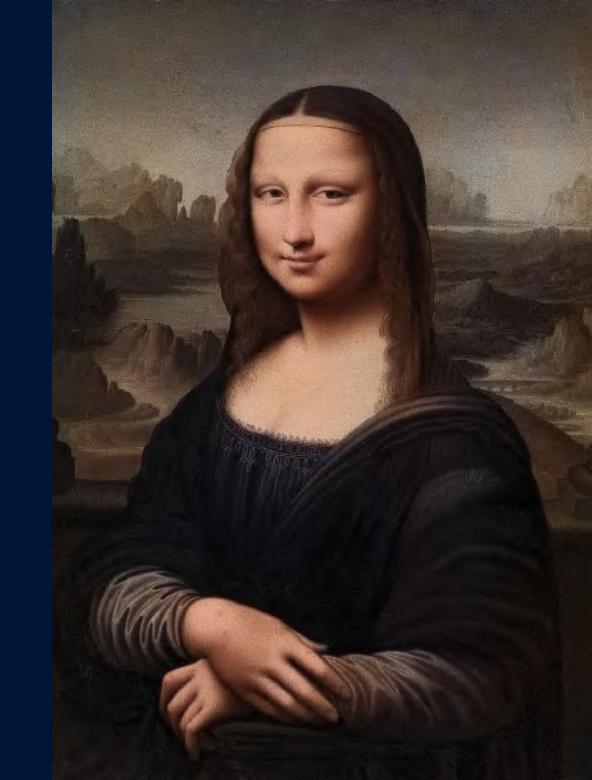
GENERATED 2016



ORIGINAL

# Richzhang/colorization

В 16 году в статье "Colorful Image Colorization" была представлена одноименная архитектура нейронной сети. Она состоит из трех основных компонентов: первый компонент - это сверточная нейронная сеть, которая извлекает признаки из входного изображения; второй компонент - это классификатор, который преобразует эти признаки в вероятности цветовых категорий; и третий компонент - это оптимизатор, который преобразует классификацию в цветизацию пикселей.



GENERATED 2017



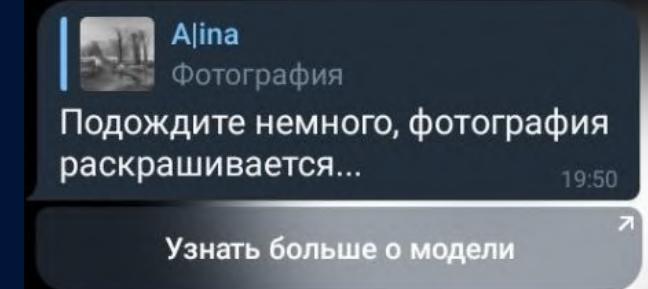
ORIGINAL

[HTTPS://GITHUB.COM/RICHZHANG/COLORIZATION](https://github.com/richzhang/colorization)

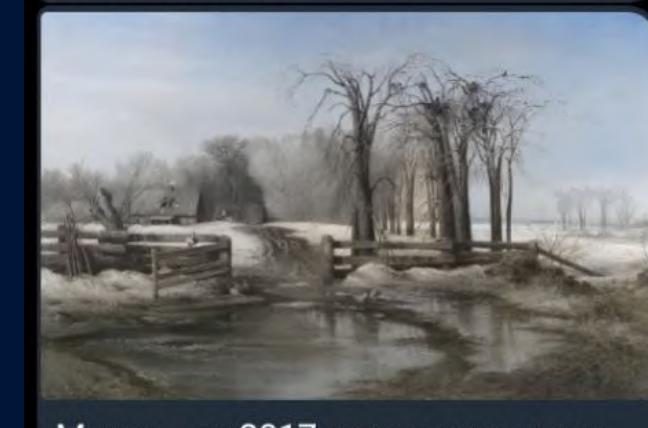
# TelegramBot

Телеграм-бот реализован с помощью библиотеки pyTelegramBotAPI. Бот принимает фотографии от пользователя, раскрашивает их с помощью моделей, автоматически отправляет обратно пользователю и удаляет загруженные файлы с сервера. Для работы с кодом требуется установка пакетов argparse и matplotlib, а также наличие файлов с весами моделей "eccv16" и "siggraph17", которые будут использоваться для раскраски фотографий.

[https://github.com/topperal/colorize  
\\_telegrambot.git](https://github.com/topperal/colorize_telegrambot.git)



Модель от 2016 года показывает  
такой результат  
19:50



Модель от 2017 года показывает  
такой результат  
19:50

# Examples

## Baroque



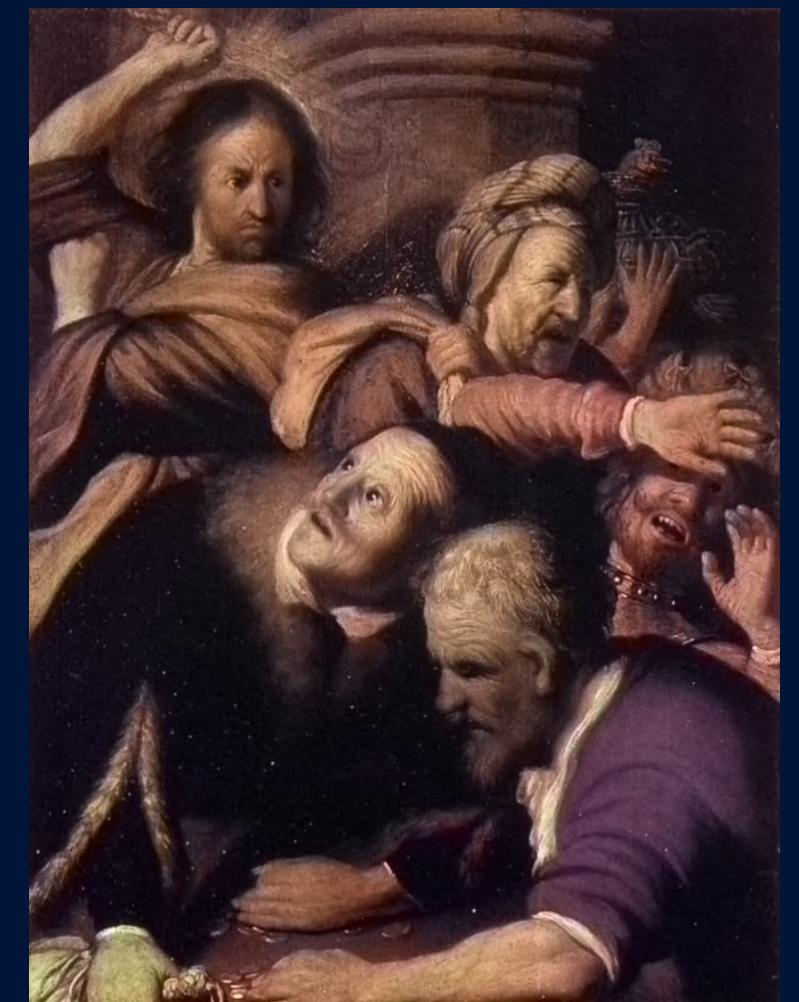
Original



Model



Original



Model

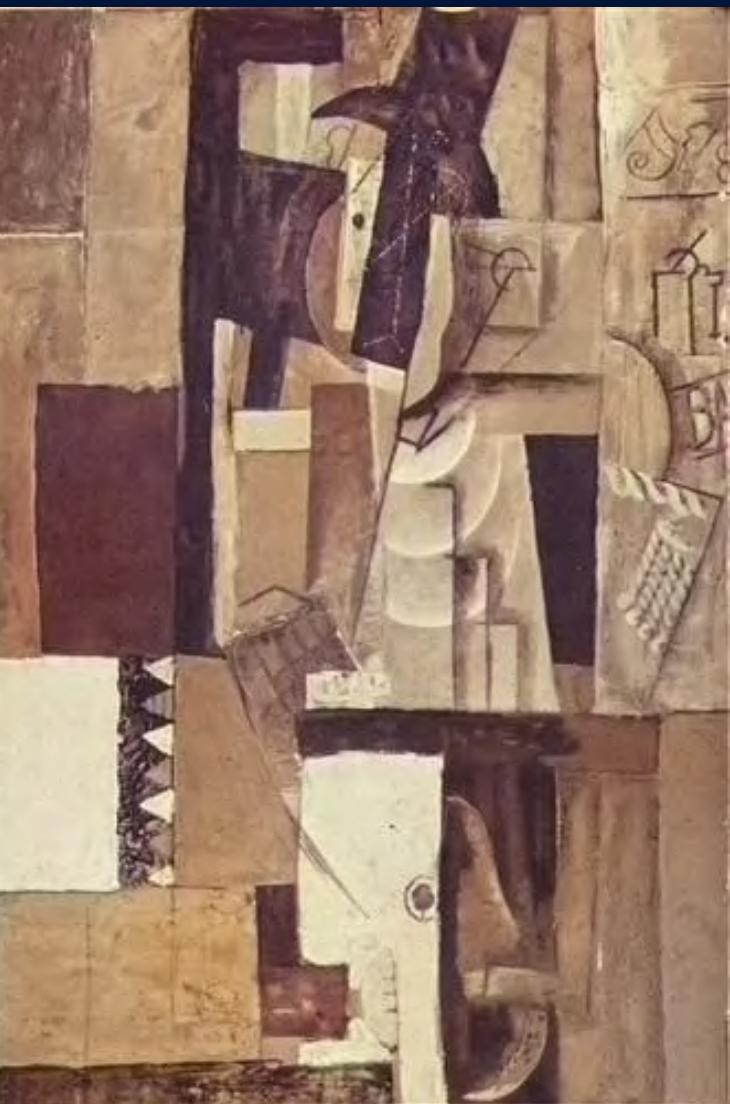


# Examples

## SyntheticCubism



Original



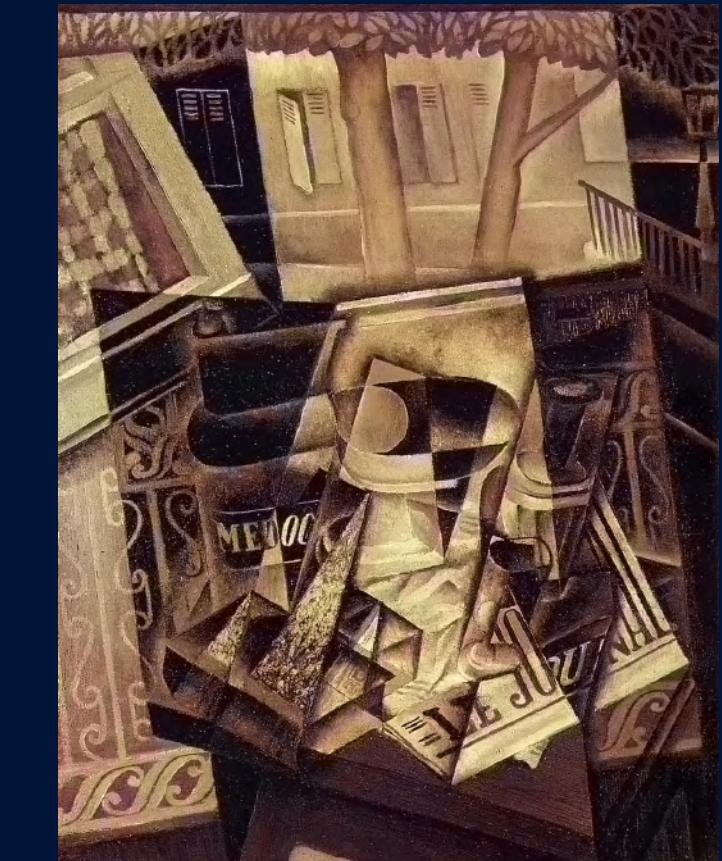
Model



Original



Model



# Examples

## Symbolism



Original



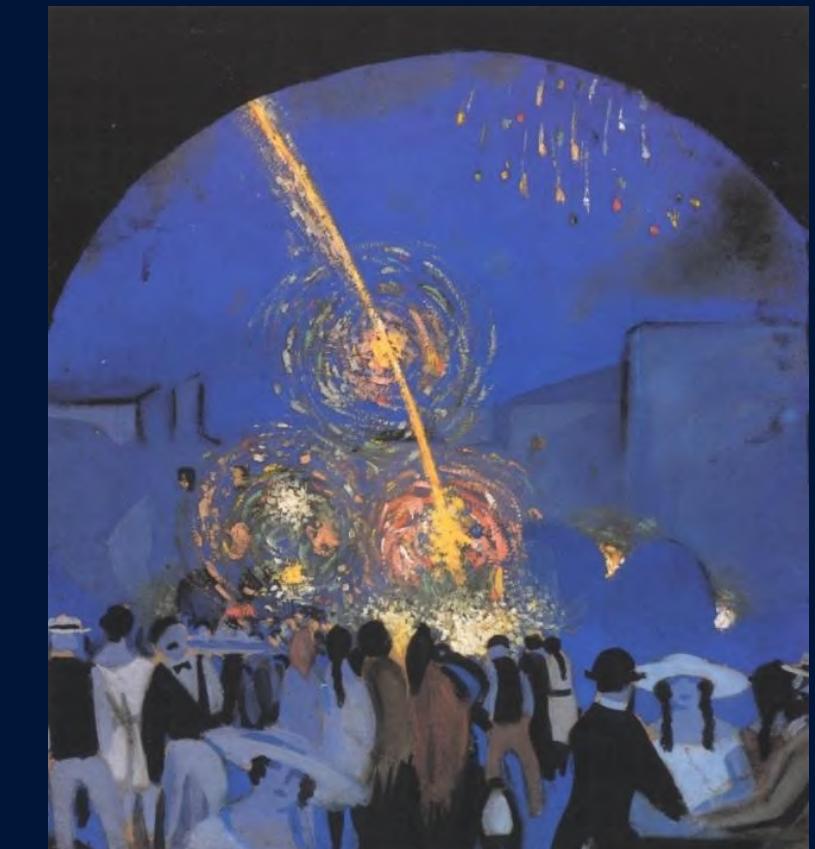
Model



Original

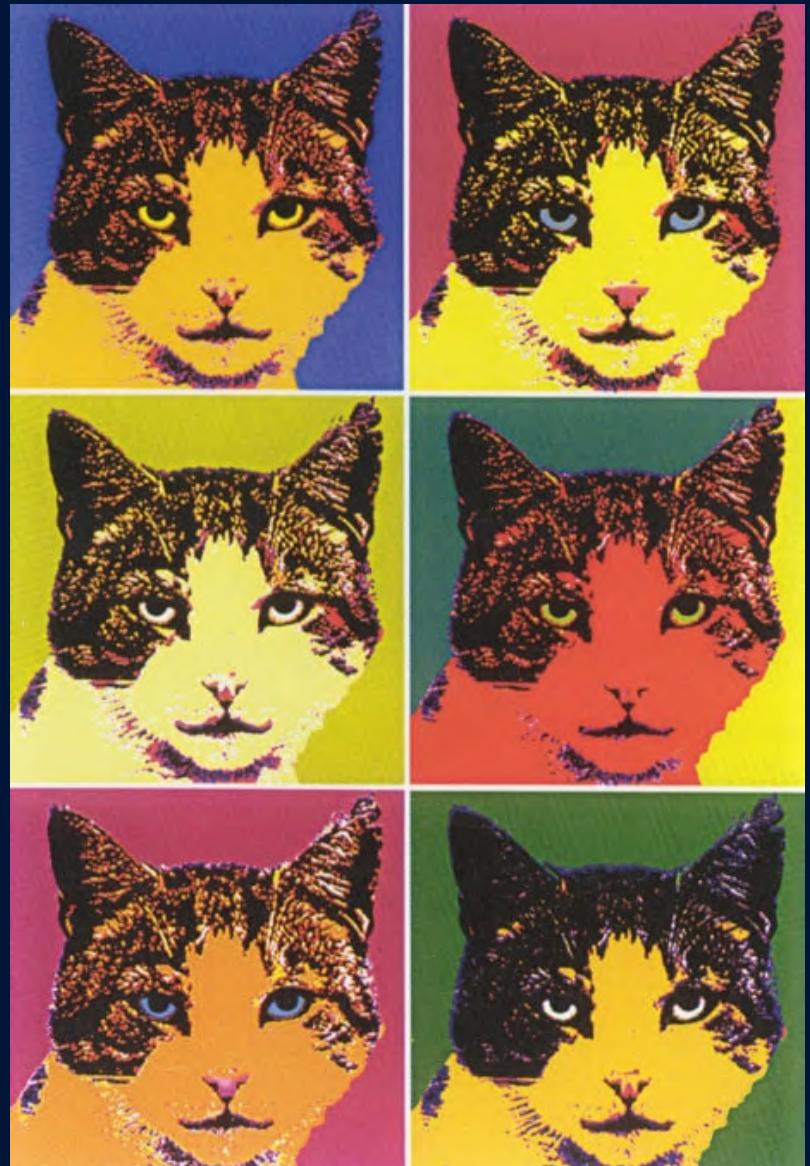


Model

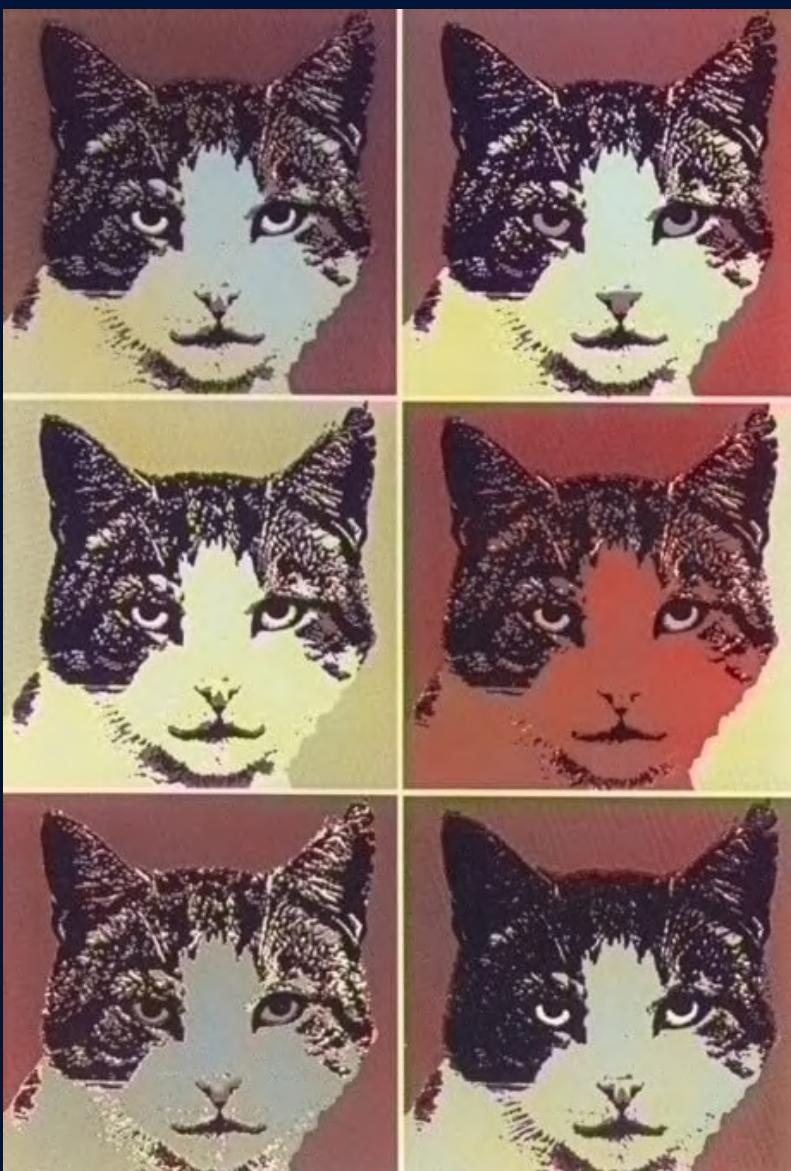


# Examples

PopArt



Original



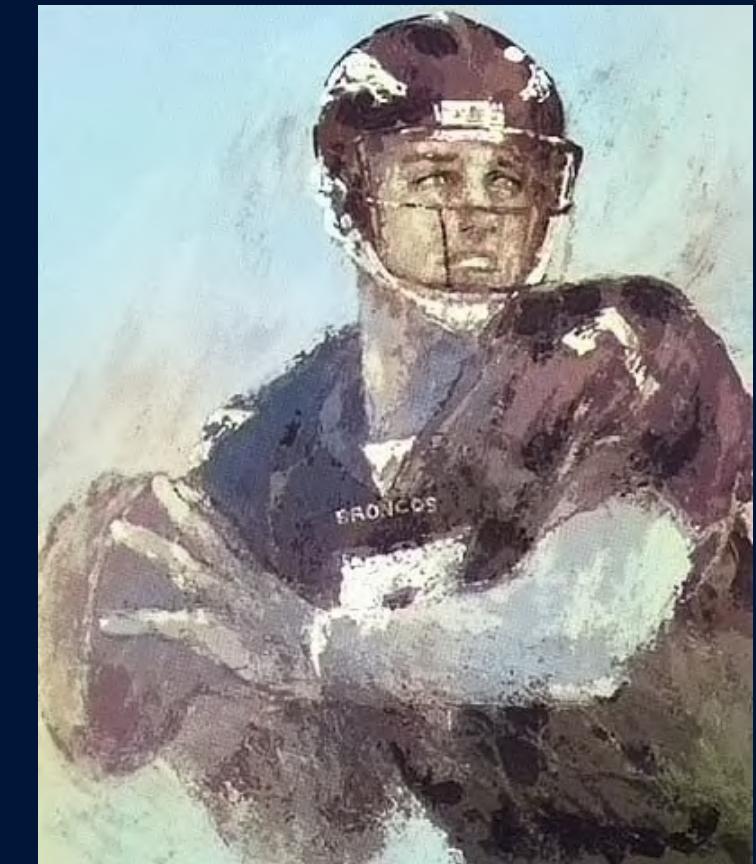
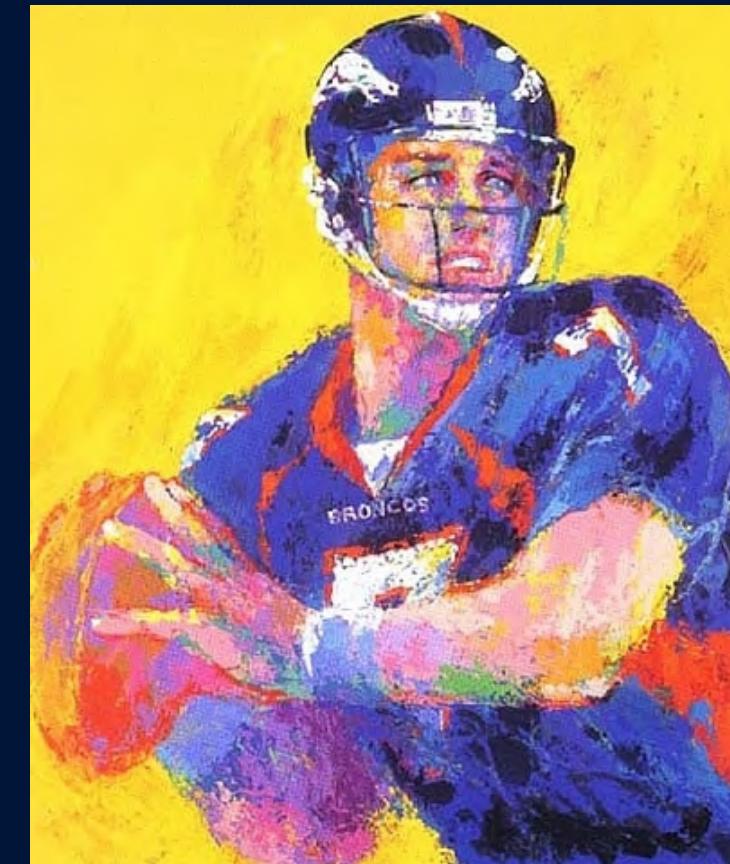
Model



Original



Model



# Examples

## Pointillism



Original



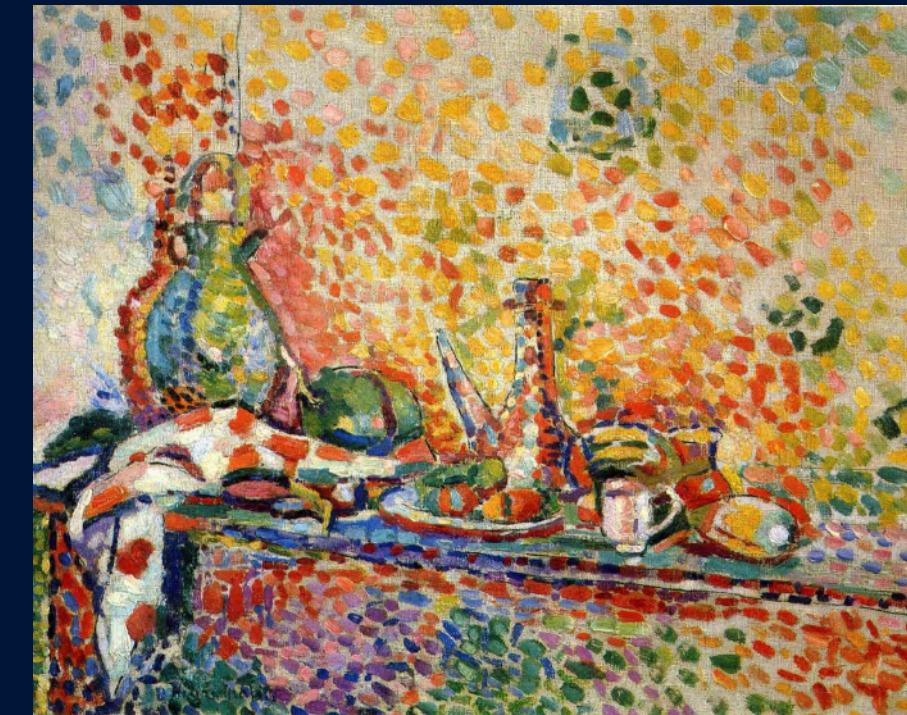
Model



Original



Model

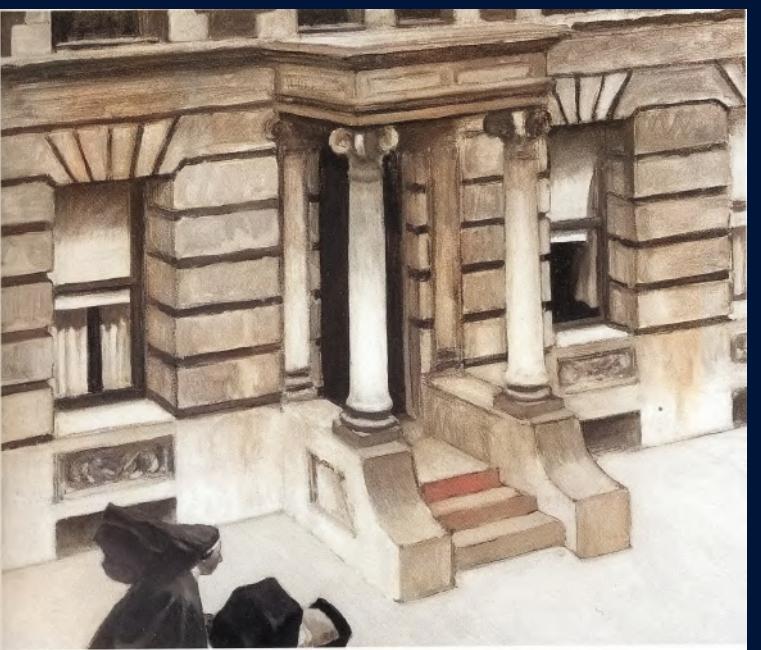


# Examples

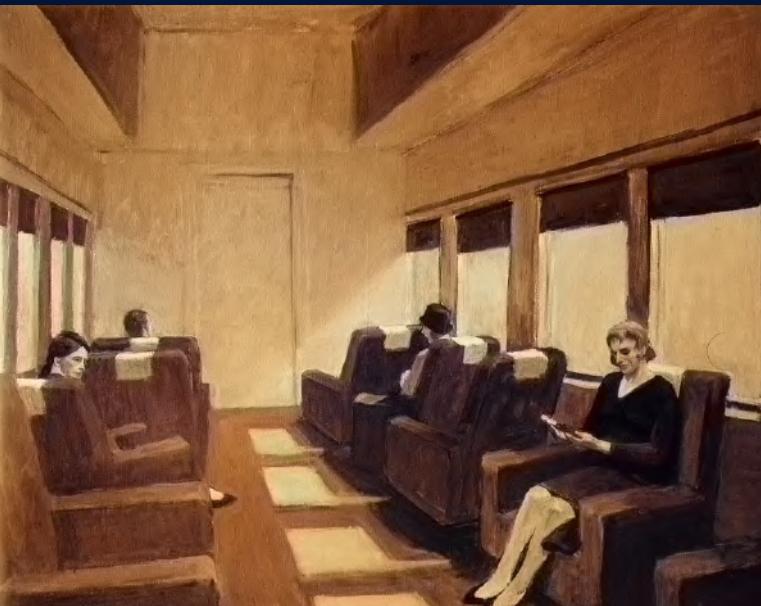
NewRealism



Original



Model



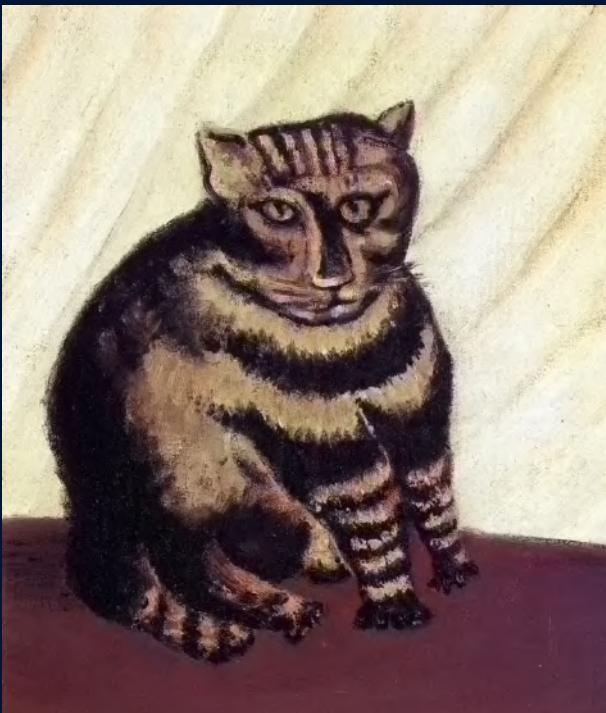
Original



Model

# Examples

## NaiveArtPrimitivism



Model



Original

Original



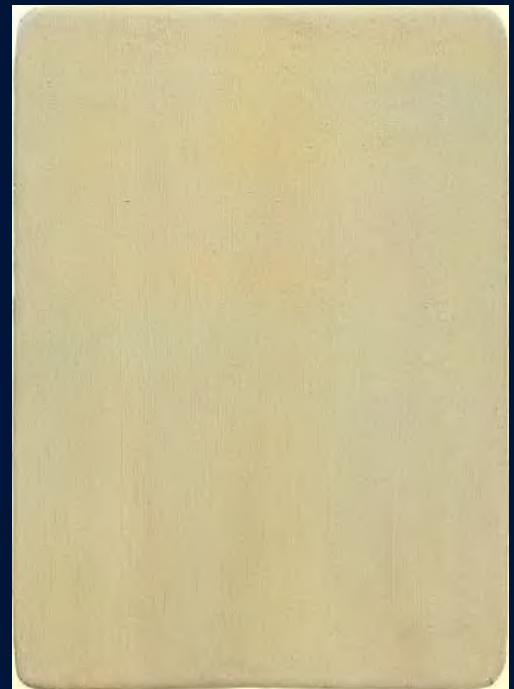
Model

# Examples

## Minimalism



Original



Model



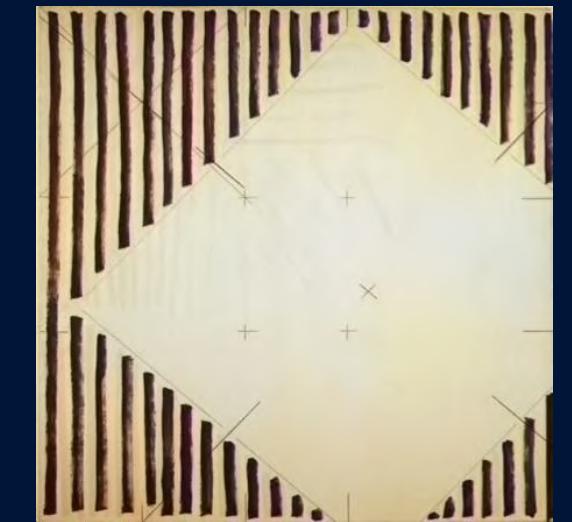
Original



Model



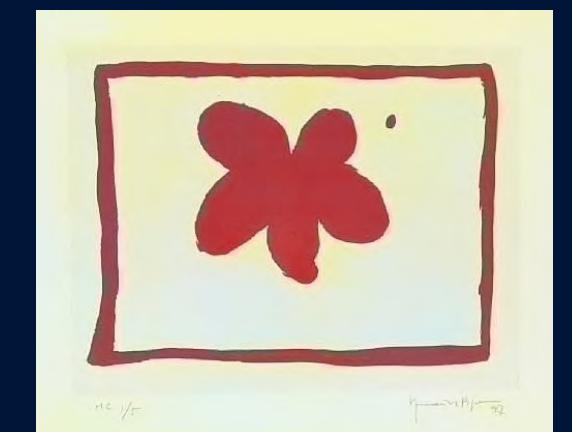
Original



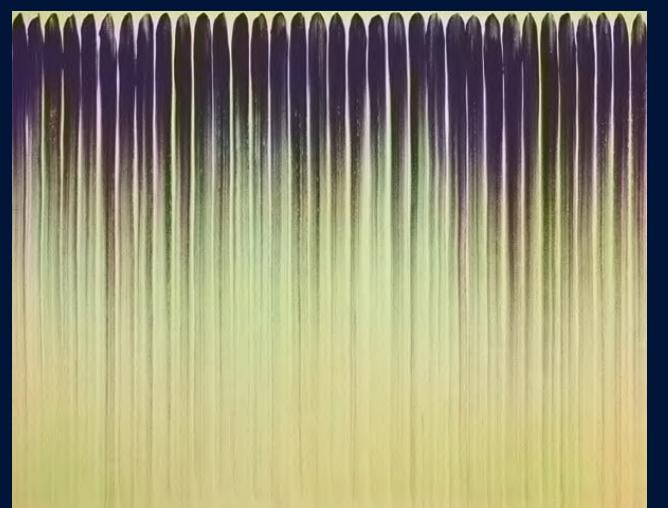
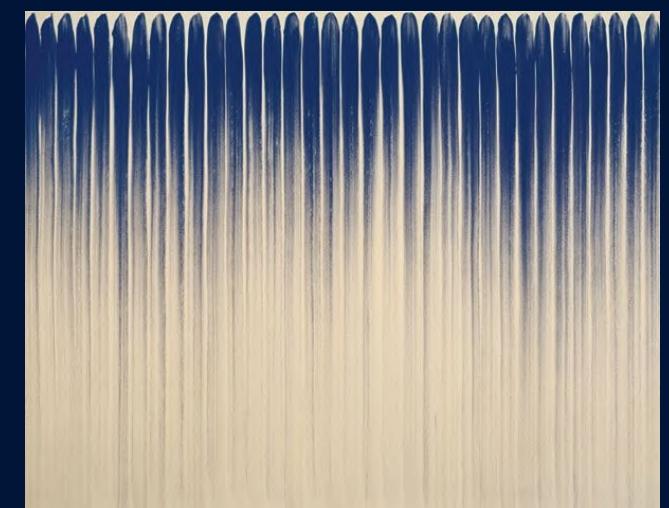
Model



mc /f

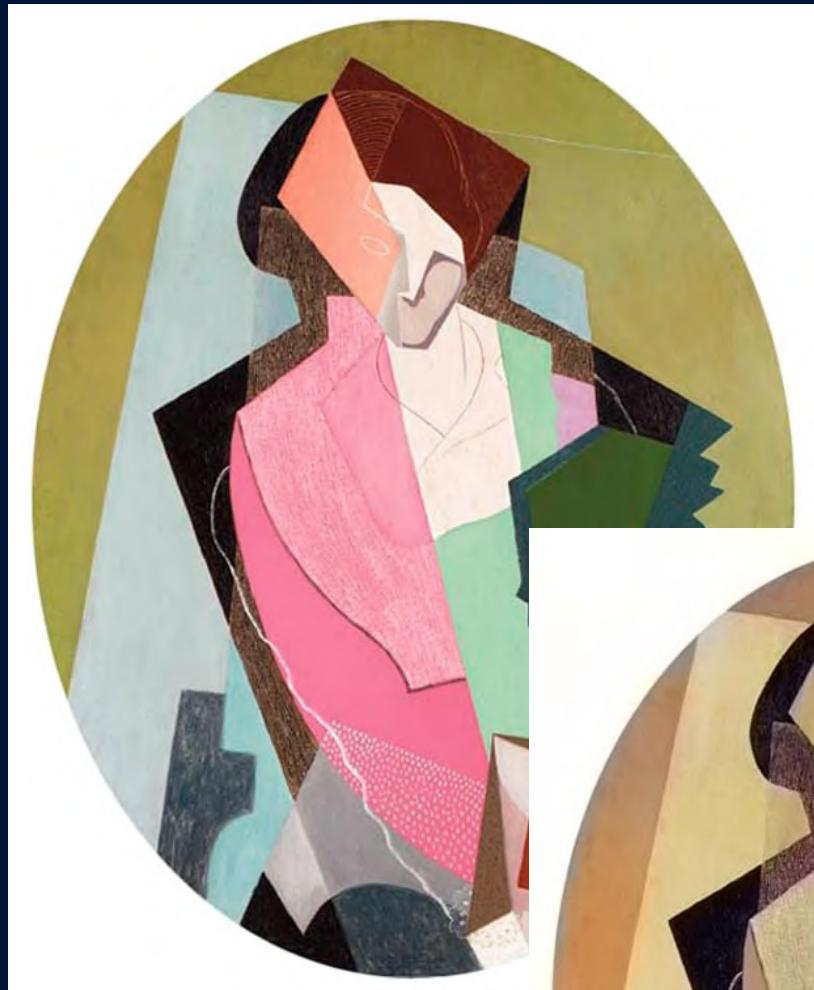


mc /f

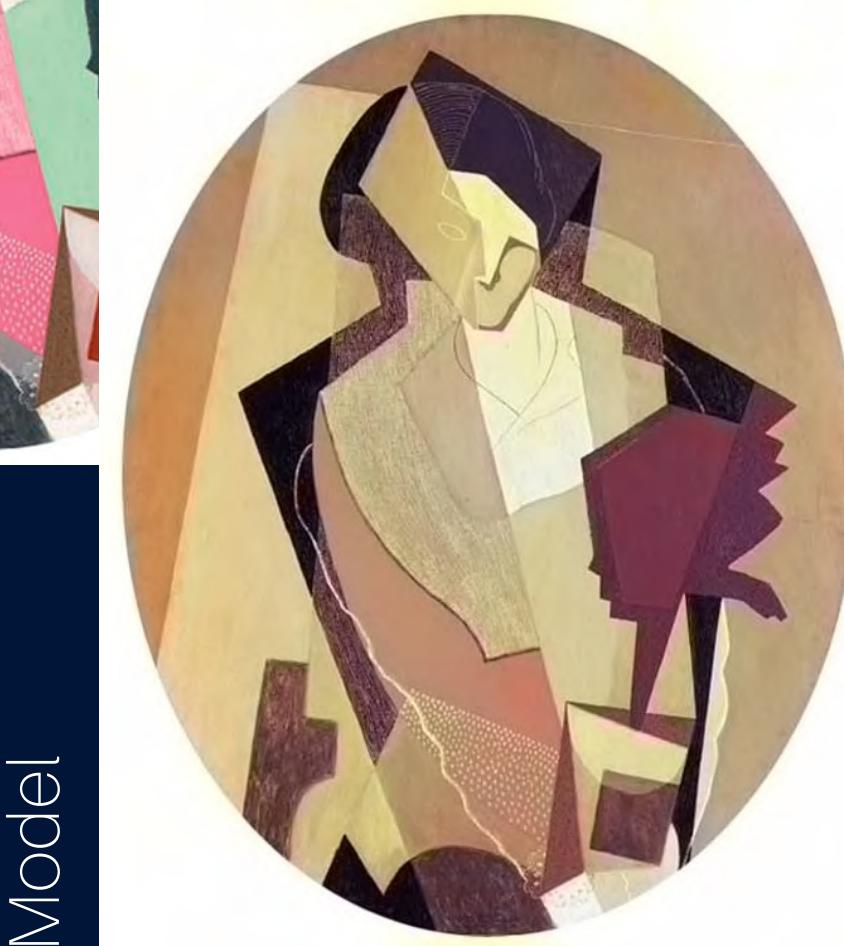


# Examples

## Cubism



Original



Model



Original



Model



Original



Model