Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

Prezentacja pracy dyplomowej:

Multimodal colorization of greyscale images using deep neural networks

na studiach: **Studium Podyplomowe**

Głębokie Sieci Neuronowe – Zastosowanie w Mediach Cyfrowych

mgr Tomasz Ostaszewicz

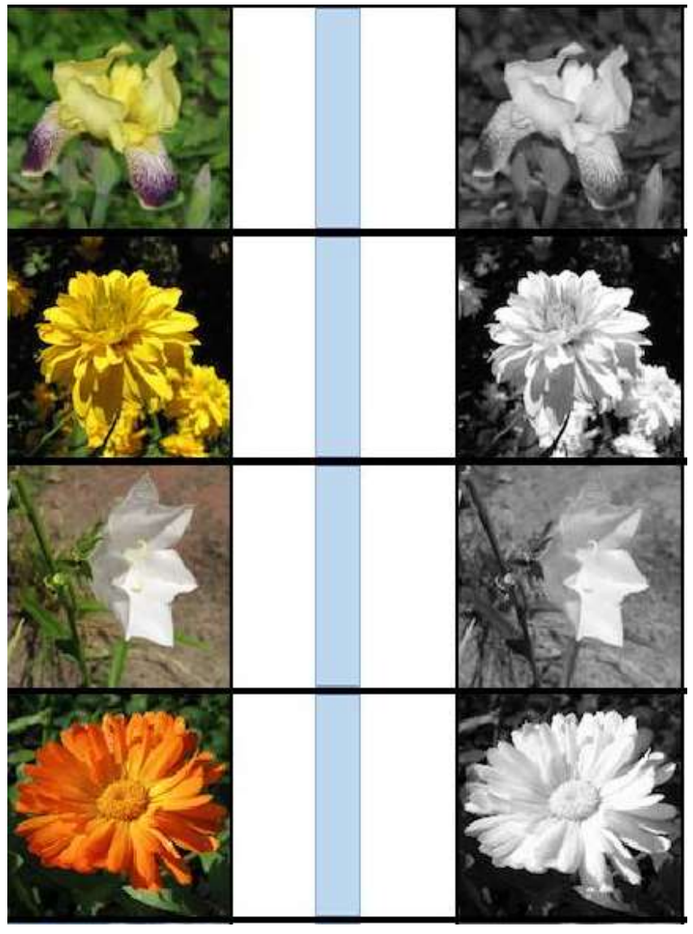
opiekun pracy dyplomowej

mgr inż. Rafał Protasiuk

Warszawa, 2020-09-30

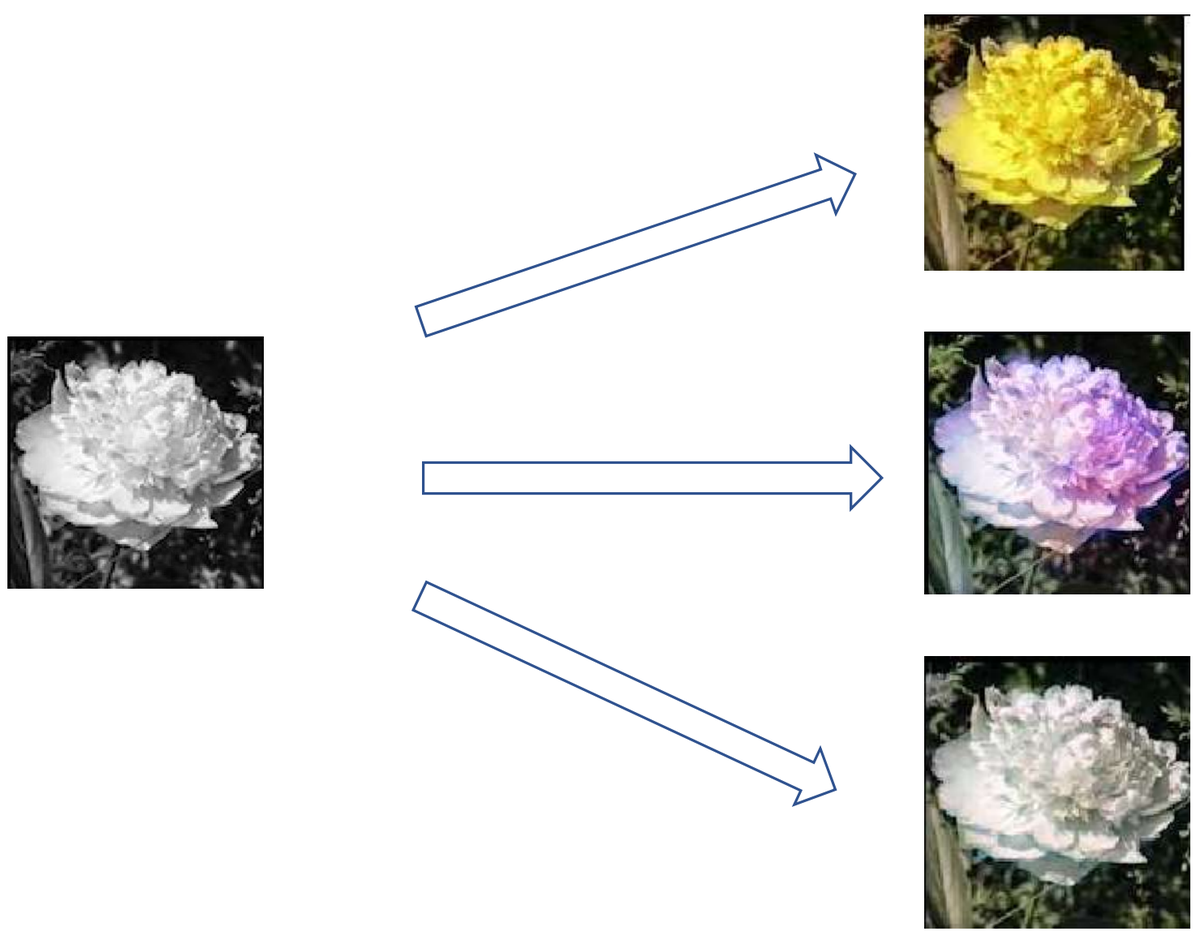
Traininig data set for colorization

Typical training data set for colorization has a form:
Input: grey scale image, Target: colorized image



Usually there are no multiple colorizations of the same grey scale image.

Multicolorization problem.

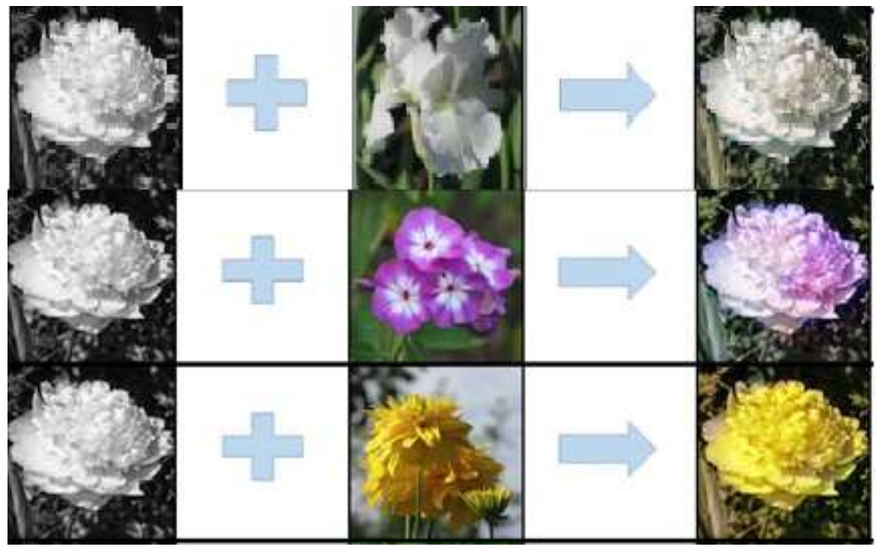


Colorize greyscale image in multiple ways so as to obtain naturally looking color images

How to approach problem using deep neural networks?

One of possible approches is to use reference image for colorization:

Grey scale image + reference color image = colorized image using colors from reference image



The training data

What kind of training data is needed for this task?

Training data requirement:

Input pair: (gray scale image + reference image) → Target: (colorized image)

Please note that this kind of training data containing conditinal colorization using reference images is not widely available for many tasks. Can we approch this problem in another way?

The training data

What is widely avilabe are just color images. For each color image we can easily obtain greyscale image. That is we can have only pairs:

(grey scale image, colorization of grey scale image)

So what if there is not data set available which uses reference images?

Is it possible to try to solve mutlicolorization problem having **only** one colorization of greyscale image in training data and no colorizations based on reference image?

The first thought is no.

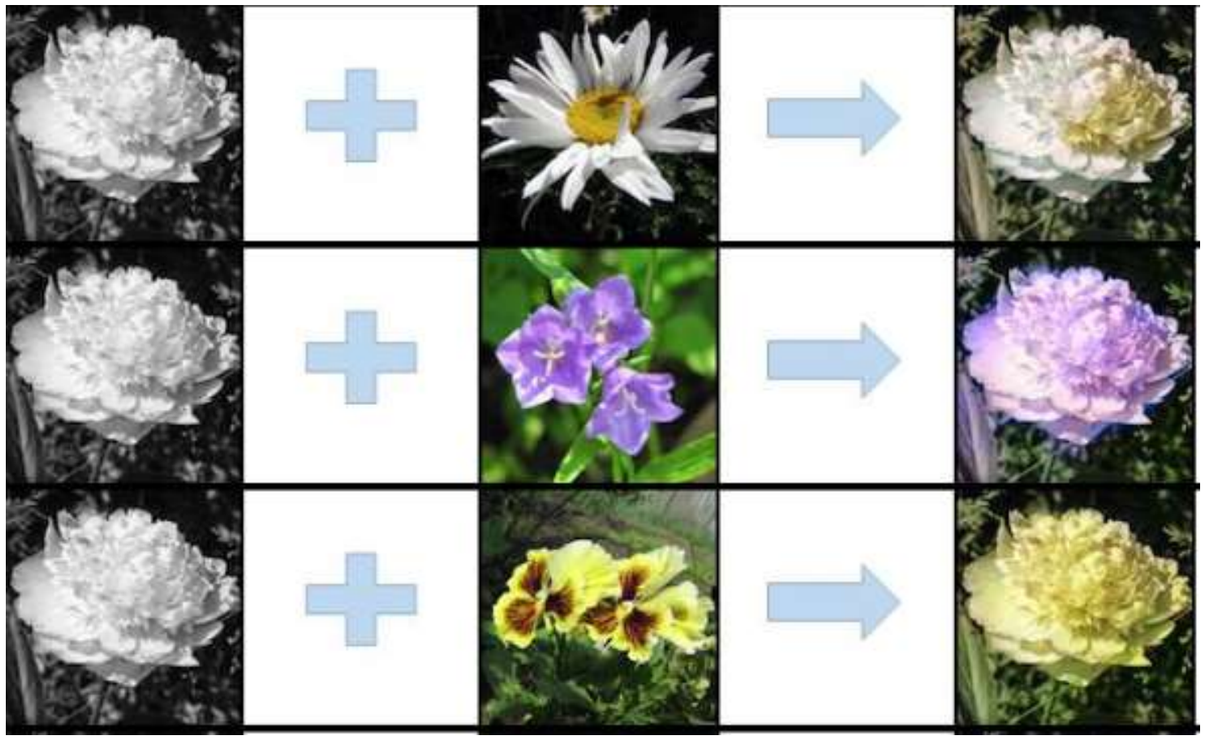
But it turns out that there is a clever way to approch this problem.

Proposed approch results

Training data

Colorization model

The end result is



Model architecture

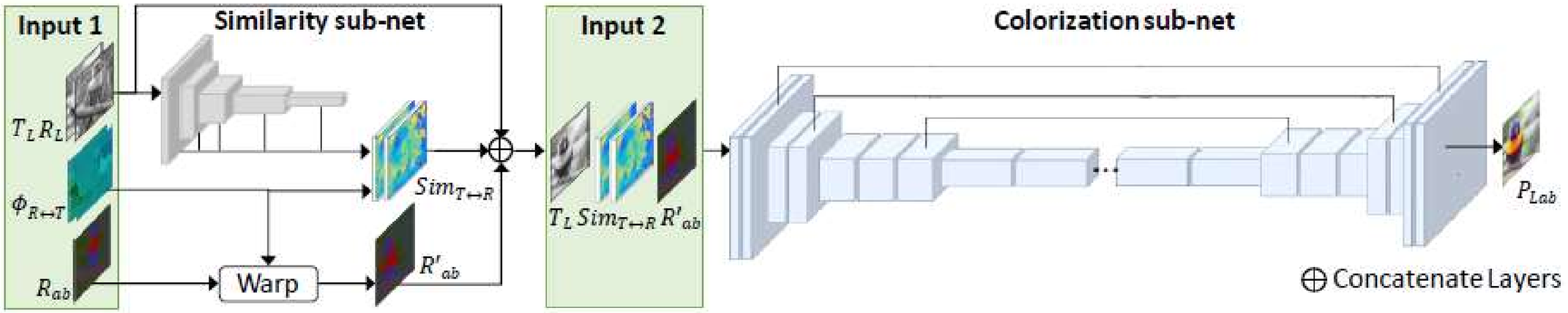


Image source: M. He, D. Chen, J. Liao, P. V. Sander and L. Yuan, "Deep exemplar-based colorization," ACM Transactions on Graphics, vol. 37, no. 4, p. 47, 2018.

Proposed model consists of three parts:

1. Similarity model which is based on Resnet34 architecture.
2. Patch Match algorithm
3. Colorization model which is based on U-net like architecture.

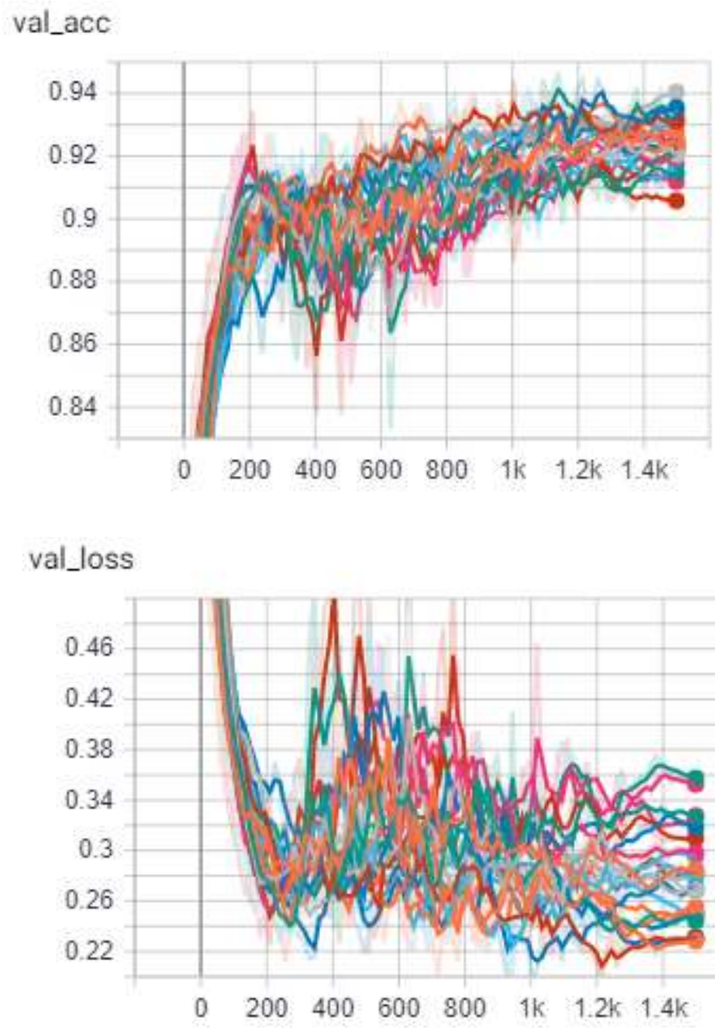
Similarity model pretraining

Different training runs have different hyper parameters: Similarly CIE L similarity model was trained.

The best obtained validation accuracy were:

95% for RGB similarity model,

89% for CIE L similarity model.

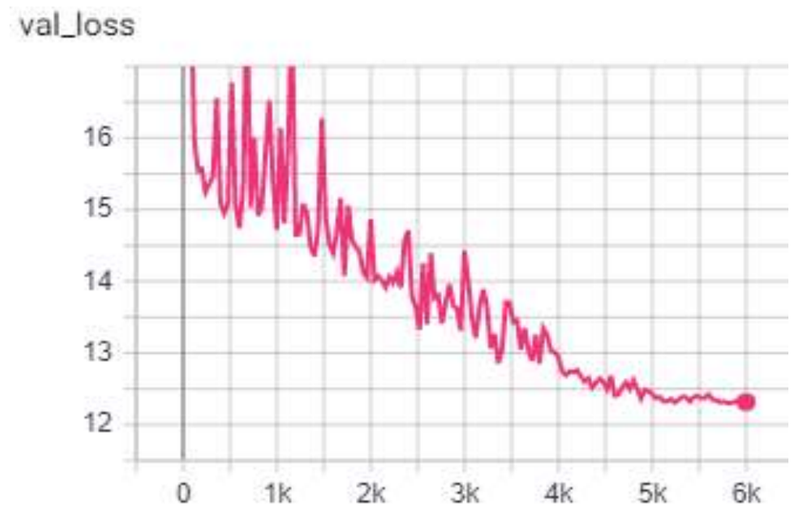


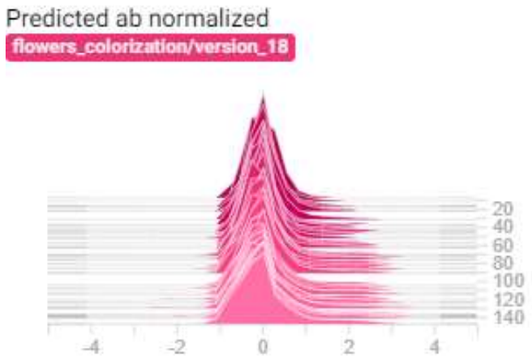
Patch Match algorithm implementation

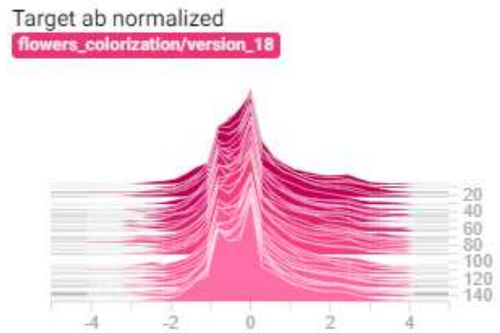
Implementation of Patch Match algorithm using basic tensor operations in Pytorch resulted in efficient code, which runs each distance calculation between patches in sequential order. To give more precise numbers, for example for target and reference tensors of shape [96, 64, 64, 64] Patch Match algorithm implemented in Pytorch runs 5 minutes and 12 seconds. This is a time needed to prepare input batch for Colorization network using “Pytorch” implementation. To correct this inefficiency Patch Match algorithm was directly implemented in CUDA C++ programming language. Patch Match algorithm CUDA implementation runs in 12 seconds, which means 26 times speed-up. Due to Patch Match algorithm implementation in CUDA C++ training of Colorization Network in 53 hours was possible.

Training Colorization model without Patch Match algorithm

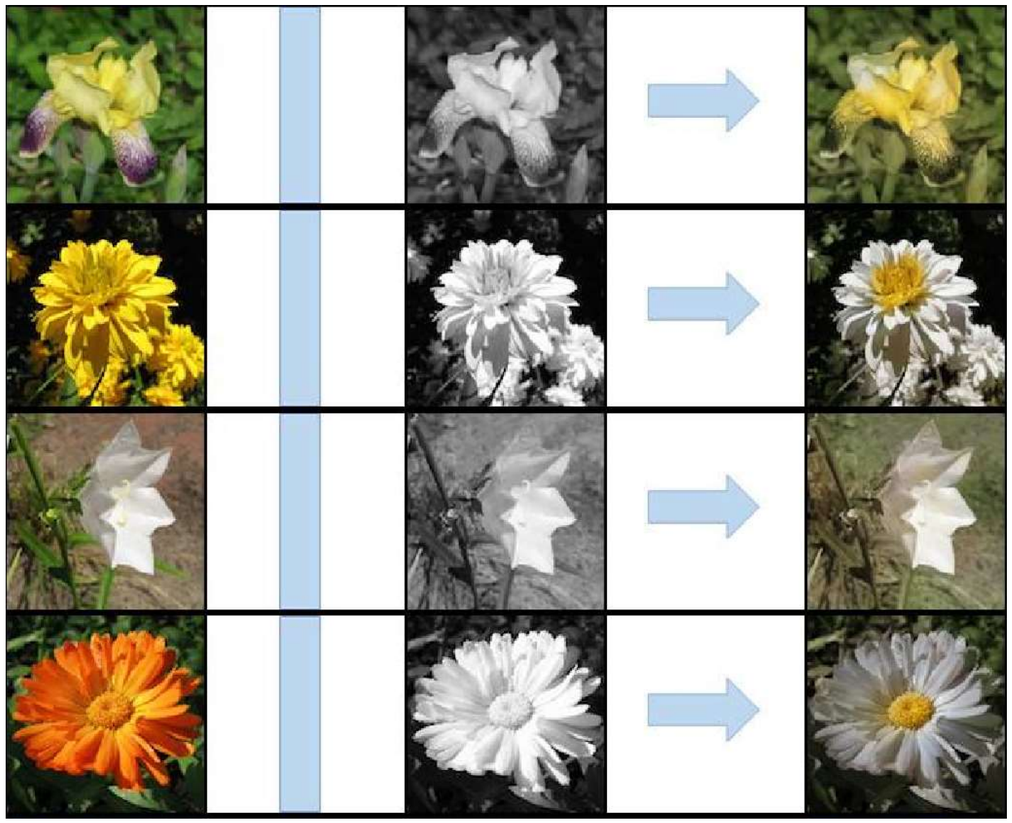
Colorization model without Patch Match algorithm did not use results of Patch Match algorithm. Patch Match results were replaced with zero tensor. This training was performed to check robustness of colorization model and training loop. Training was performed on 150 epoch using single V100 GPU on Google Colab. It took approximately 3 hours to train

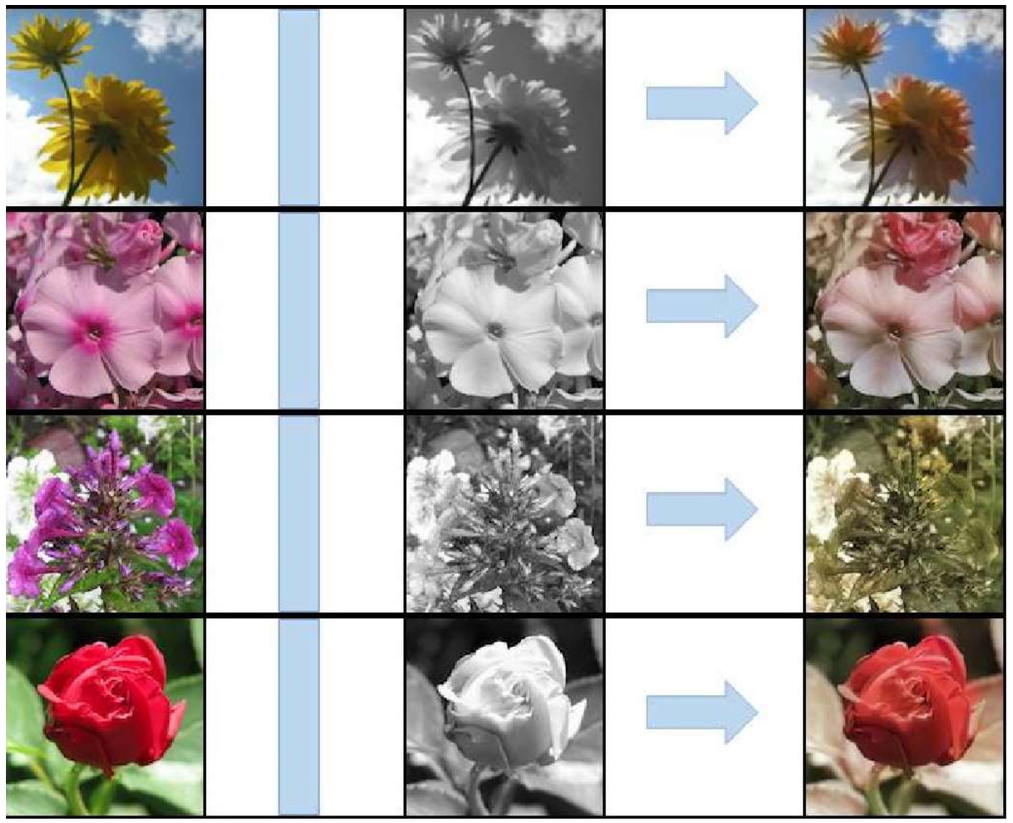




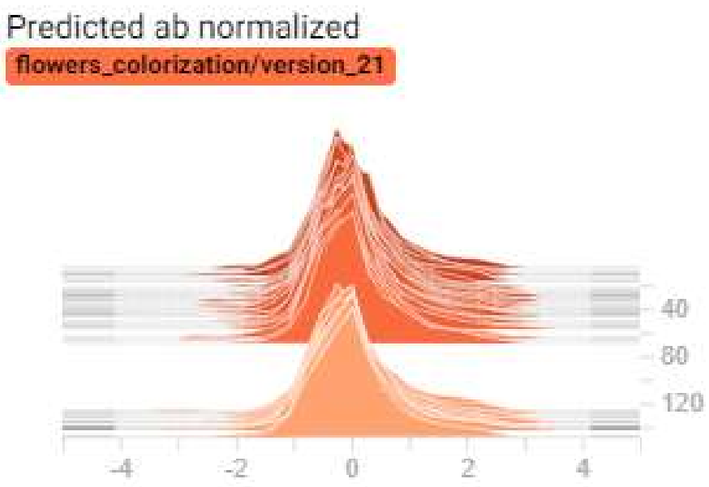


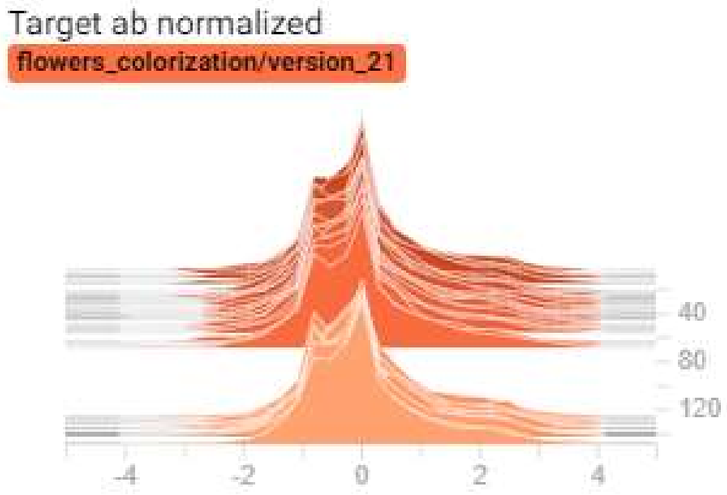
Results of colorization model without Patch Match algorithm

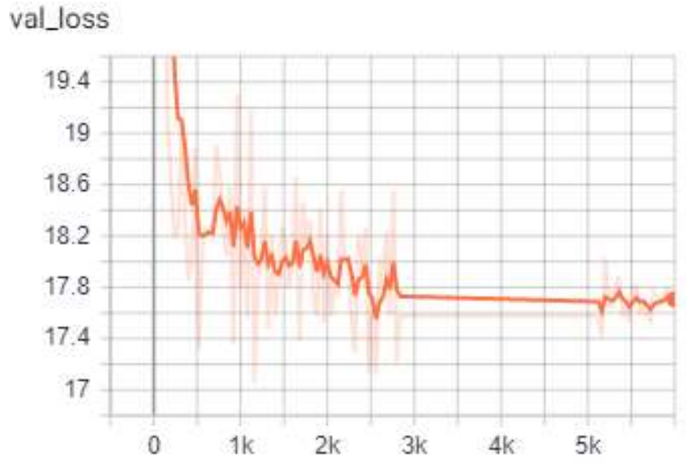




Training Colorization model with Patch Match algorithm

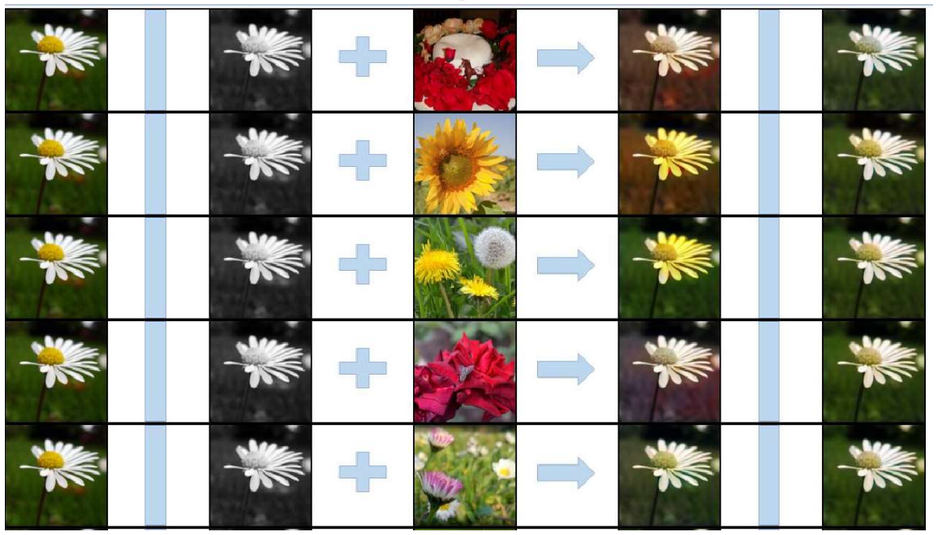


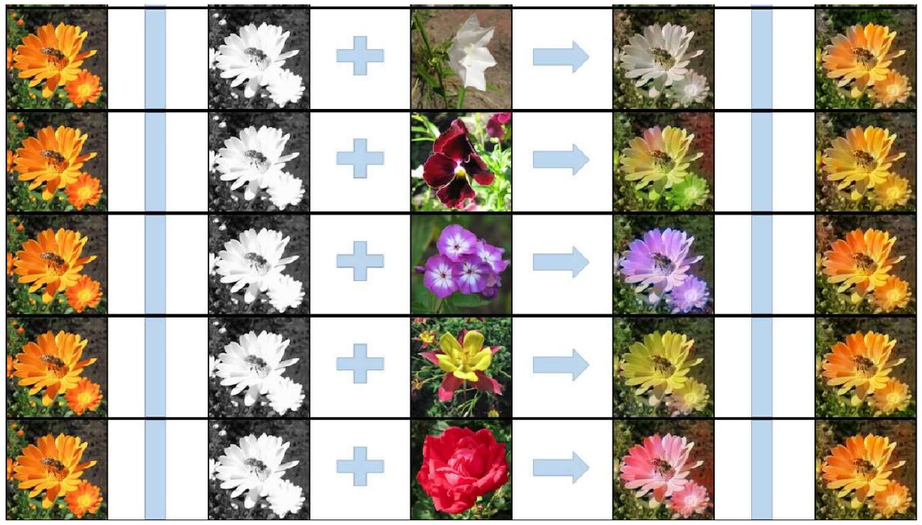


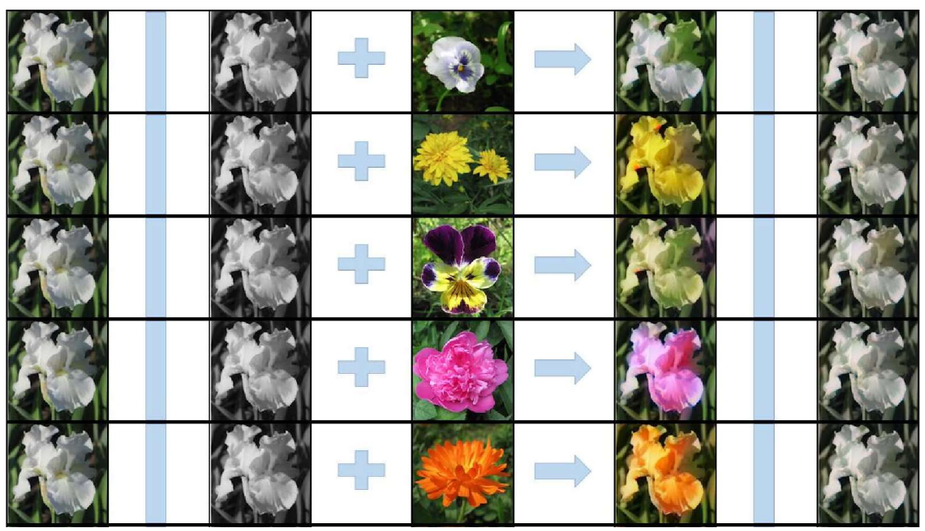


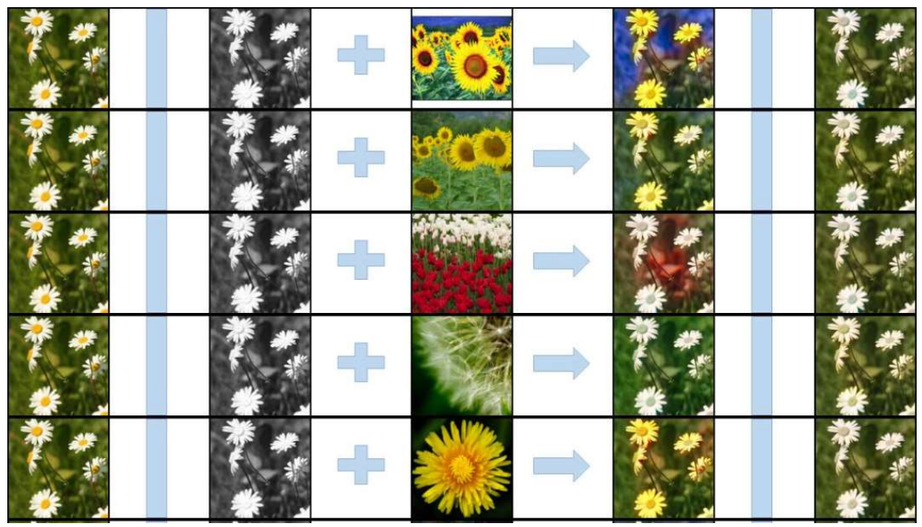
Training was performed on 150 epoch using single V100 GPU on Google Colab. It took approximately 53 hours to train.

Results of colorization model with Patch Match algorithm



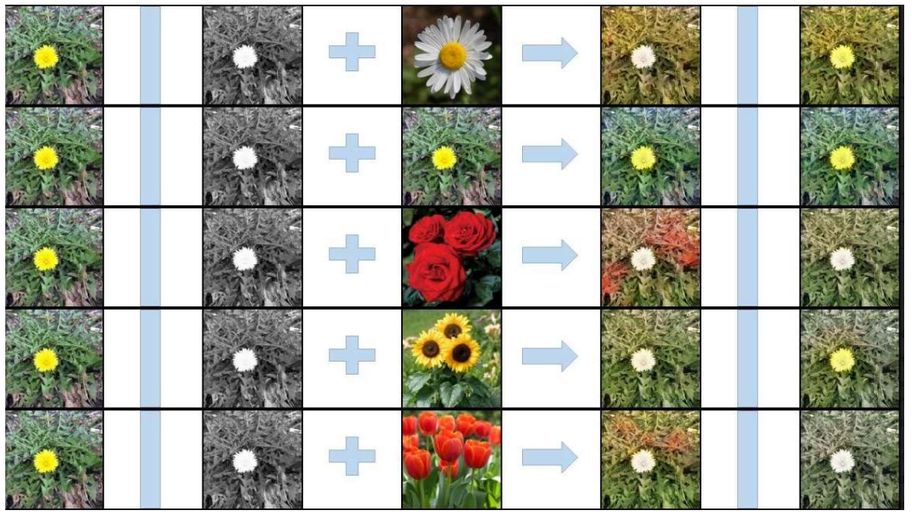




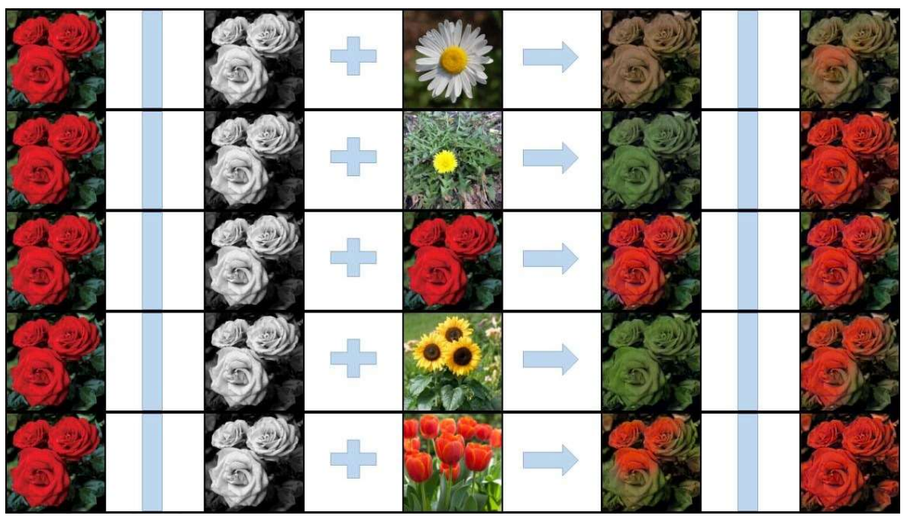


Results of colorization model with Patch Match algorithm – fail cases

Fail cases appear when image has many fine grain flowers or leaves. It seem that model cannot clearly decide how to approach colorization for that kind of images:



Collection of roses appears to be problematic. From semantic point of view it seems that model treats roses like leaves (thus tends to color roses using green color from background leaves).



Colorization comparison using different publicly available tools

Colorization by:
Deoldify &
deepai.org
Source: https://deepai.org/machine-learning-model/colorizer

Colorization by:
https://demos.algorithmia.com/colorize-photos

Colorization by:
https://pixbim.com/colorize-picture

Presented colorization model (without Patch Match)

Ground truth