IMAGE COLORIZATION

Final Project for Vision and Cognitive Systems

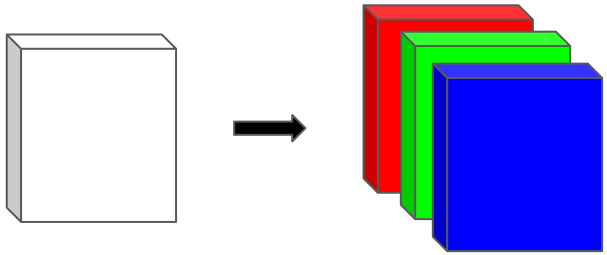
AY: 2022 - 2023

Alberto Sinigaglia ID: 2044712
Beatrice Sofia Bertipaglia ID: 2054766
Chiara Colato ID: 2062670

INTRODUCTION

The ***TASK*** performed is ***Colorization*** with which ***black and white images*** are turned into their colored version.

The ***main issue*** that makes this topic challenging is the ***multi-modal nature*** of the problem.

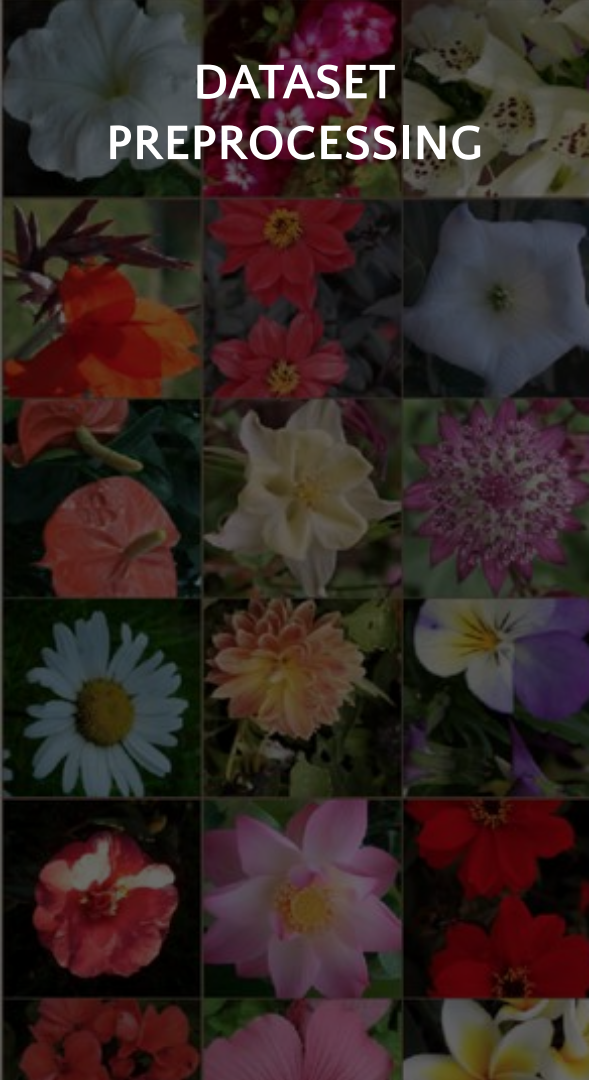


1 channel → 3 channels: RGB

DATASET

We consider the ***Flowers*** dataset, composed by 18540 colored images of flowers and plants with original dimensions 500×500 px.

The reason we chose it is that we want representations about a ***simple object*** that can be designed in slightly different ways and with ***different color*** possibilities.



DATASET PREPROCESSING

The ***original dimensions*** comport high computational costs.

So we proceed with a ***resizing of the images***, bringing them to size 128 × 128 px.

- We are able to maintain a ***good quality*** of the images, preserving visual features

- We can reach an ***accessible time of computation***.

APPROACHES

Maximum likelihood estimation

- MSE
- MAE

Implicit density estimation

- GAN
- cGAN

Hybrid:

MLE + GAN

MLE based

They are ***regression based***, they aim to reconstruct pixel wise the image giving as target the original image.

- Mean Squared Error (MSE) $J(\theta) = \mathbb{E}_{x,y \sim p_{data}}[(f_\theta(x) - y)^2]$
- Mean Absolute Error (MAE) $J(\theta) = \mathbb{E}_{x,y \sim p_{data}}[||f_\theta(x) - y||]$
- Huber loss

We will use ***MAE*** as seems to be able to better reconstruct visually the images.

Implicit Density Estimation

With implicit density function we refer to the family of methods under the name of Generative Adversarial Networks, that in the last years have gained a lot of attention thanks to their performance as generative methods.
In particular, they aim to solve a min-max problem of the following loss:

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

They are composed by ***generator G*** and ***discriminator D***, where the discriminator aims to classify if the given image comes from the true distribution or if it's been generated, and the generator tries to trick the discriminator mapping from a latent distribution to the target one, using the gradient coming from it to adjust the training.

Hybrid

Here they try to get the ***best of the two worlds together***.

The final loss results in the following expression:

$$J(D,G) = \left[\lambda\, \mathbb{E}_{x,y\sim p_{data}}[L(G(x),y)] + \mathbb{E}_{x,y\sim p_{data}}[\log D(x,y)] + \mathbb{E}_{x,y\sim p_{data}}[\log(1 - D(x,G(x))]\right]$$

This method is known in the literature as ***Pix2Pix***

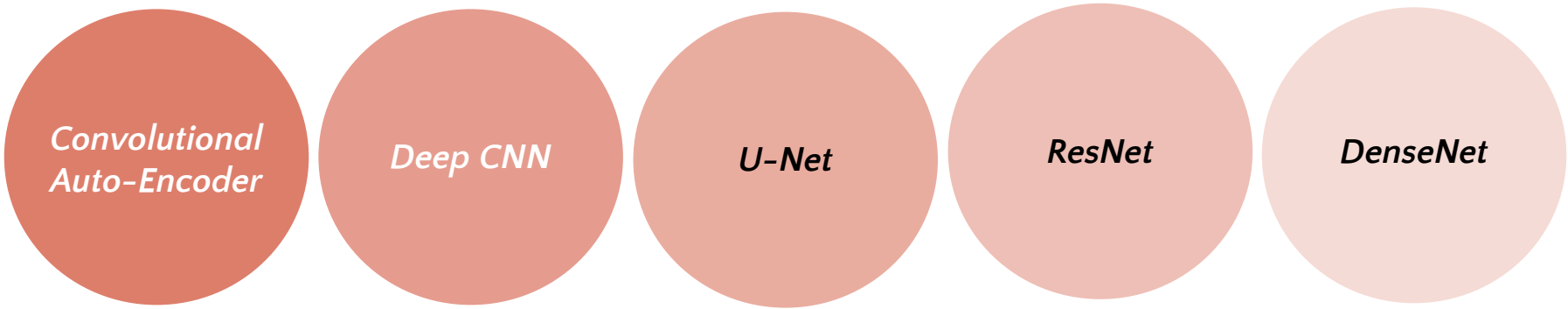
Even though GANs have initially been proposed using binary cross-entropy, it has been shown that is not very stable[i], and that practically they work better with regression losses[ii], as they provide a nicer gradient.

For this reason, we will use MSE also for the classification error made by the discriminator.

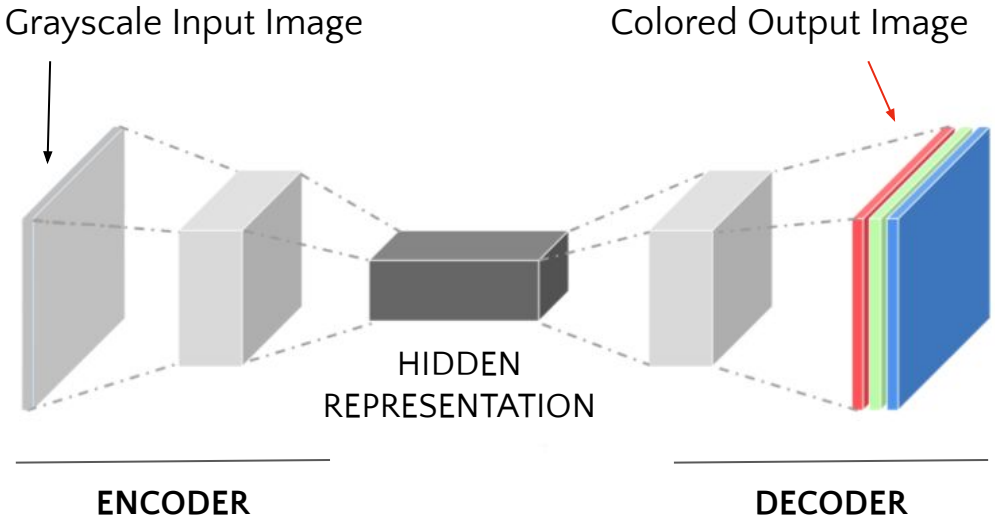
i: WGAN
ii: CycleGAN, LS GAN

MODEL STRUCTURE

In order to find the best suited model for the problem, we tried many different popular architectures, and optimized their hyperparameter with a full grid-search.



CONVOLUTIONAL AUTO-ENCODER



15 total blocks, each one **composed by:**

1. **Linear Convolutional Layer**
2. **Batch Normalization over the channel**
3. **Activation Layer element-wise.**

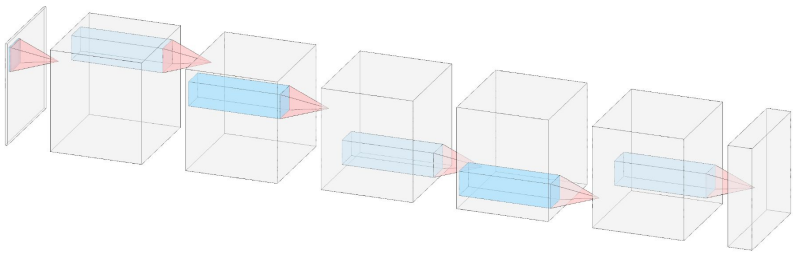
After **3** blocks we have a **Down/Up-Sampling layer.**

Batch Normalization for Covariate Shift Reduction

It's been shown[i] that internal covariate shift of deep networks is one of the main problems when optimizing deep networks, reason which batch norm is been introduced, and will be used across all the models that have been tested.

i: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift

DEEP CONVOLUTIONAL NEURAL NETWORK

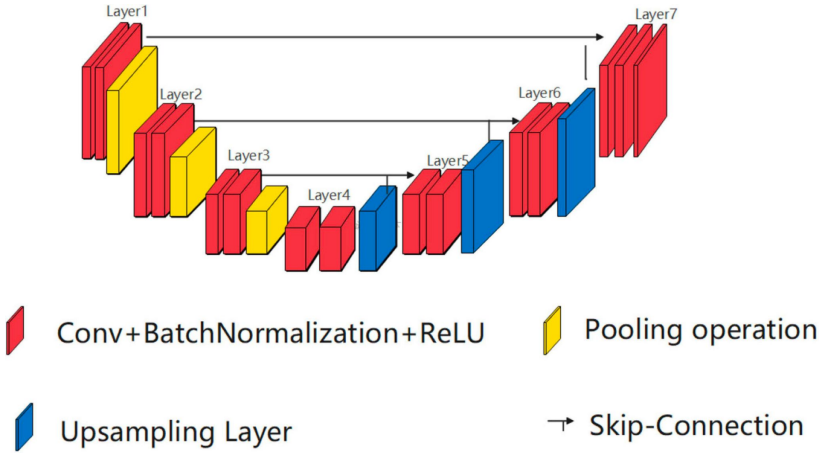


We try here a simpler model that we built simply by stacking a ***series of Convolutional Layers***, in order to gradually build up more complex models.

In this case, there is no bottleneck as for the CAE.

This network and the CAE will be used as baseline.

U-NET



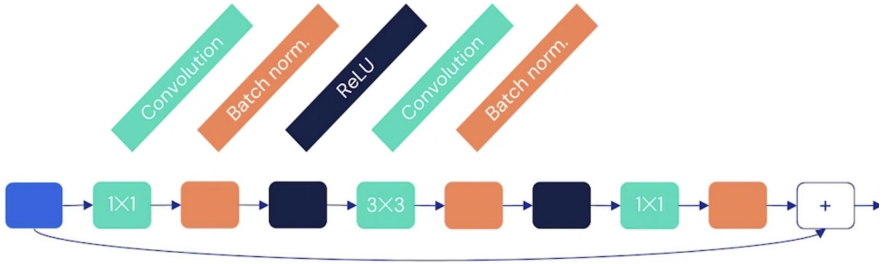
Famous model used for image-to-image translation in the literature, uses ***Residual Connections*** to reduce the distance between input and output, since already the input contains a lot of useful information even without any transformation.

Residual Connections

Residual Connections are a type of skip-connection that learn residual functions with reference to the layer inputs, instead of learning unreferenced functions: this helps both the network to learn the mapping, and the optimizer, as this helps keeping the layer norm constant across all layers[i].

i: Demystifying ResNet

RES-NET



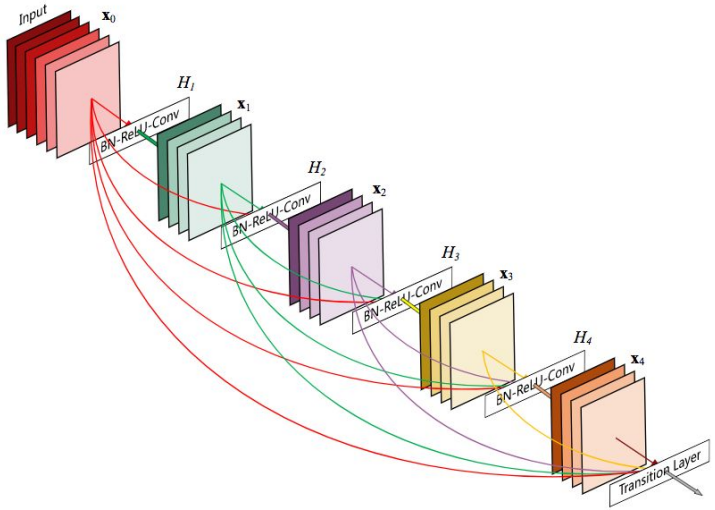
Stressing even mode the concept of ***residual connection***, we tested ResNet v2, which uses the ***bottleneck*** block in order to reduce the number of parameters.

This is achieved by using 1x1 convolution to downscale and upscale in the channel dimension, in order to reduce the number of parameters in in the 3x3 convolution.

This also matches what is been shown in the literature as best residual depth[i].

i: Demystifying ResNet

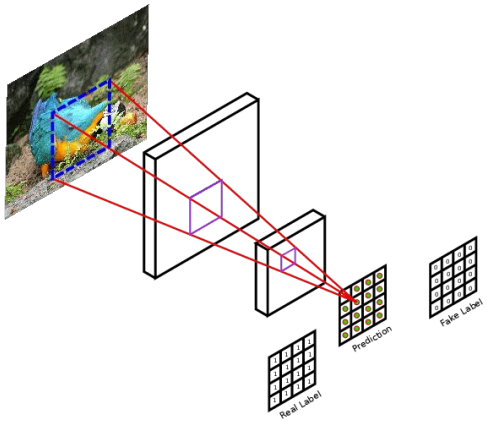
DENSE-NET



Taking the concept of residual connection to its limit, we tested dense-net, which also exploits the bottleneck block form ResNet v2 in order to reduce the input dimension, as the final layers will receive a tensor with many channels, as it comes from the concatenation of the output of all the previous layer.

Pix2Pix

U-Net +

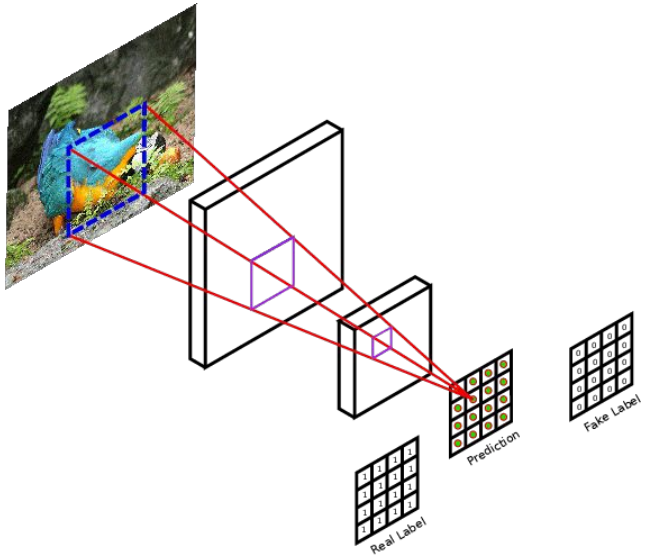


Pix2Pix uses a combination of ***generator + discriminator*** to improve colorfulness of the resulting images.

In this case, the generator is a U-Net, as it has the best compromise of complexity and output quality, and the discriminator is a fully convolutional neural networks, called PatchGAN.

$$J(D,G) = \left[\lambda\,\mathbb{E}_{x,y\sim p_{data}}[L(G(x),y)] + \mathbb{E}_{x,y\sim p_{data}}[(1-D(x,y))^2] + \mathbb{E}_{x,y\sim p_{data}}[D(x,G(x))^2]\right]$$

PatchGAN



The discriminator of PatchGAN uses a series of convolutional layers that, exploiting the fact that convolution has a **local receptive fields**, aims to classify only the underneath patch of the image.

Thanks to this, the image has to be credible only patchwise, as will be the reconstruction loss to drive the output structure, thus the PatchGAN should only drive to a more credible recolorization.

EFFICIENT TRAINING ON BIGGER IMAGES

We exploit the fact that ***batch normalization*** and ***convolutional operators/layers*** do not depend on the input size for their transformation, thus:

- We can train the models in bigger images without being forced to have bigger models.
- However, the convolution operator is not linear in the input size, thus bigger images means longer time for forward and backward pass.
- Resizing input images to 128x128 patches, will keep the ***patches distribution as in the original dataset***, thus a net trained on those patches should also be able to reconstruct full images

Training

- sample batch of 256x256 images
- pick a random crop of size 128x128 of each of them
- train the net on the patches

Evaluation

- build a net that accepts 256x256 with the same architecture and the same weights
- feed the full resolution image as input

METRICS FOR COLORIZATION EVALUATION

Regression Metrics

MAE
MSE
Huber
BCE

MSE =  2

PSNR

images are treated as signals, and the PSNR is then calculated

SSIM

similarity estimated using statistics of the pixel distributions

Density estimation

estimate density of dataset using VAE and evaluate with mean-KL divergence

Task specific

can be used mIoU for segmentation quality

INADEQUACY OF METRICS

As most of the “generative” tasks, is very complex to find adequate metrics, and this hold for the identified ones previously mentioned.

MAE/MSE assume a single possible correct output for each pixel, which by the nature of the problem, it’s not a correct assumption;

PSNR is based on the MSE, thus suffers of the same problems;

SSIM captures only statistics of the output, is not able to represent the credibility of the recolorization;

Density based tend to give more importance to the structure of the image instead of the colorization, as their grey scaled version is the expected recolorization.

Since those considerations, we relied on the ***“turing test”***,

evaluating results visually on a subset of the test-set.

RESULTS

Models are difficult to evaluate, however visually Pix2Pix outperforms visually all regression based methods, of which U-Net is been elected as best model.
U-Net, indeed, gives equal if not better results than more complex models, which training took much longer
(U-Net - 2h, ResNet - 8h)

FAILURES OF MODELS

- ***COMMON TO ALL MODELS***

Color bleeding
Inconsistent colorization

- ***COMMON TO ALL REGRESSION-BASED MODELS***

Greenish prior
Grayish colorization

U-Net vs Pix2Pix U-Net

U-Net

Pix2Pix

SUCCESS

FAILURE

Pix2Pix128 vs Patch-Pix2Pix 256

Patch-Pix2Pix 256

Pix2Pix 128





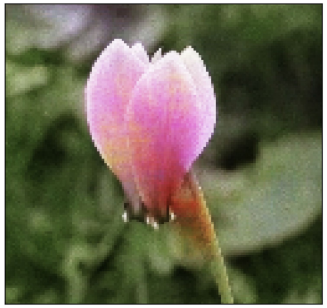

SUCCESS



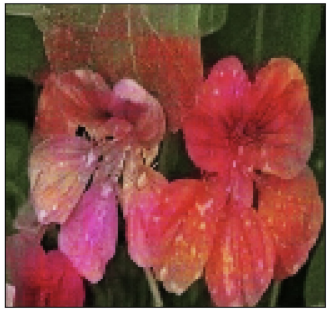

FAILURE

COLOR BLEEDING ERROR EXAMPLES

ResNet

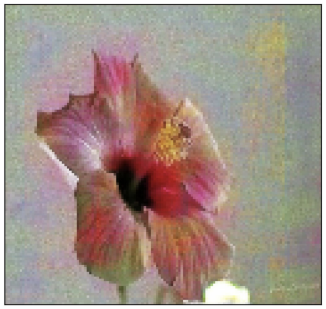


Deep CNN

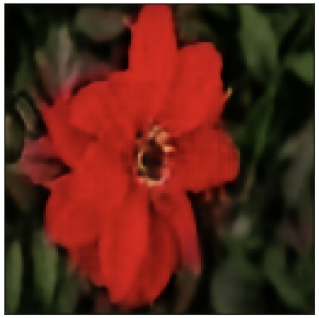


U-Net

Densenet



Auto-Encoder



Pix2Pix

GREENISH PRIOR ERROR EXAMPLES

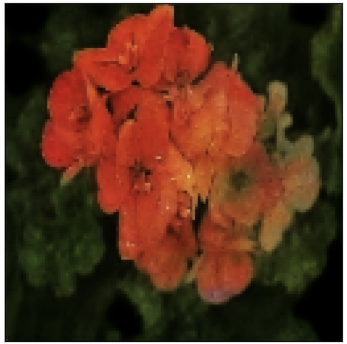
ResNet

Deep CNN

CAE

DenseNet

UNet



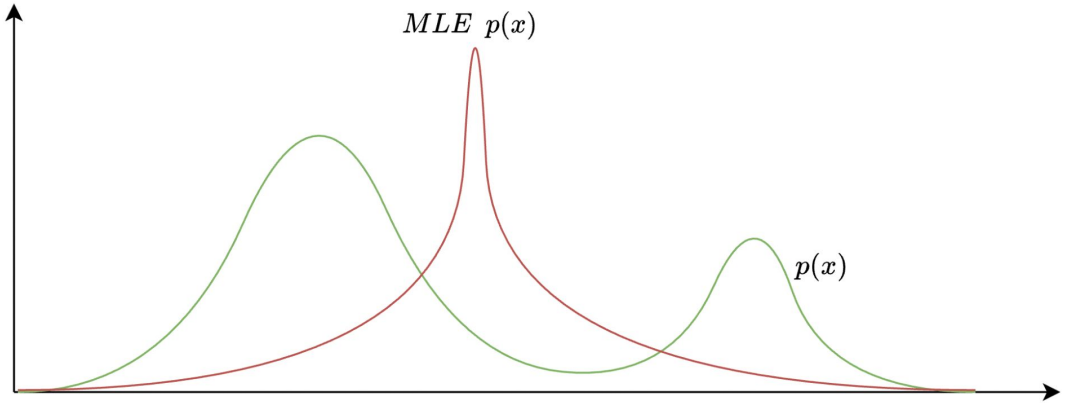
UNet

FOCUS: Why Adversarial Loss visually outperforms Regression

Regression based losses are founded on the idea of maximizing the likelihood, which means to find the best instance of a distribution family that explains the data, and then using MLE they try to optimize the corresponding log-likelihood.

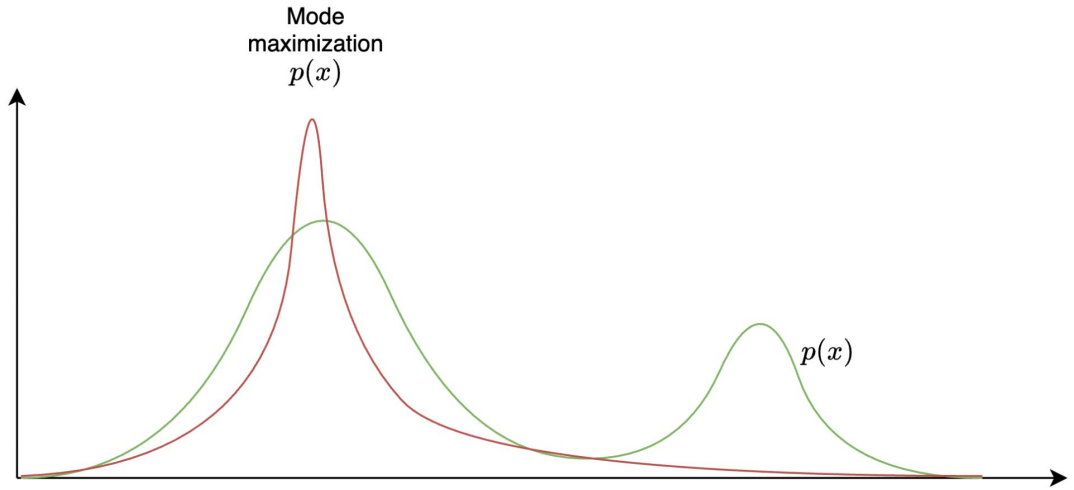
For example, MSE assumes that the target comes from a Gaussian distribution, where MAE from a Laplacian distribution. This is fine for regression problems, as most of the times this hypothesis is reasonable.

However, for multimodal problem such as colorization, the MLE might lead to a low-mass probability region.



FOCUS: Why Adversarial Loss visually outperforms Regression

Instead of using MLE, what should be actually done is aligning the target with a mode of the output distribution, in order to create samples as credible as possible.



FOCUS: Why Adversarial Loss visually outperforms Regression

Indeed, GAN's generators are notorious to be able to match ***mode distributions***, as that would make the ***generated data as close as possible to the original dataset***, and so most likely to trick the discriminator.

In our case, the additional GAN loss term of Pix2Pix allows the model to pick a mode instead of the MLE, if picking that mode would allow the final result to be very credible, even though far from the regression target.

This translates in ***more credible colorizations***, with much brighter/saturated/"riskier" colors, as they tend to better fool the discriminator, but they have the risk to be far from the target.

ADDITIONAL CONSIDERED VARIATIONS

In addition to what presented so far, the following is been considered and tested, but are not been considered for the final version as they brought little to no qualitative improvement, if not worse.

1. **different color spaces for output**: as already shown in the literature, the color space chosen slightly determines/changes the result, but there is no major difference between, if not a better color consistency observed when using RGB[i]

2. **different reconstruction loss**: as for the previous point, there is no go-to reconstruction loss for this task, as all of them have different fallacies, and might be more or less relevant depending on the problem that they are applied to[ii]

3. **different architectures with global context**: in the literature there are other possible architectures that takes advantage of popular architectures in computer vision to include global context in the latent dimension of an autoencoder[iii], however U-Net is built over the same idea and it's more effective and lighter to train.

i, ii: Influence of Color Spaces for Deep Learning Image Colorization
iii: Deep Koalarization: Image Colorization using CNNs and Inception-Resnet-v2, Let there be Color (SIGGRAPH)

ADDITIONAL CONSIDERED VARIATIONS

4

additional feature loss: adding as loss term a feature loss of a pre-trained classifier, however as already shown, classifiers tend to perform equally well in classification using colored images as well as their greyscale version, since they tend to use textures and shapes more than colors[i], thus it's more the additional computational burden than the improvement (helped only in the fully-GAN based version of Pix2Pix, stabilizing the gradient, since it acts almost as a reconstruction loss)

5

different discriminators for Pix2Pix models: as done in the original work [ii], we also tested different architectures of discriminator, and their input (cGAN/GAN), but as shown in the original paper, not much qualitative difference is been observed between the different settings.

i: ImageNet-trained CNNs are biased towards texture
ii: Image-to-Image Translation with Conditional Adversarial Nets

THANKS FOR THE ATTENTION.