Image colorization

5th March 2020

Problem Statement

Coloring gray-scale images can have a big impact in a wide variety of domains, for instance, re-master of historical images and improvement of surveillance feeds. The information content of a gray-scale image is rather limited, thus adding the color components can provide more insights about its semantics.

History of Image colorization

colorization methods broadly fall into three categories: scribble-based,transfer, and automatic direct prediction.

Scribble-based methods: require manually specifying desired colors of certain regions. These scribble colors are propagated under the assumption that adjacent pixels with similar luminance should have similar color, with the optimization relying on Normalized Cuts. Users can interactively refine results via additional scribbles. Further advances extend similarity to texture, and exploit edges to reduce color bleeding.

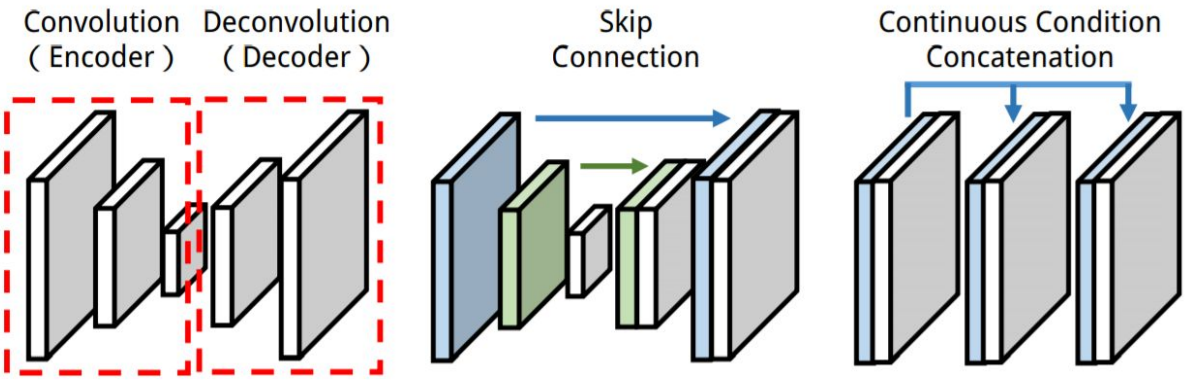
Transfer-based methods: rely on availability of related reference image(s), from which color is transferred to the target grayscale image. Mapping between source and target is established automatically, using correspondences between local descriptors,or in combination with manual intervention. reference image selection is at least partially manual.

Automatic direct prediction methods : colorize an entire image automatically without any hints.

Our project is based on the automatic direct prediction methods.

Available models

Summary of available models architectures



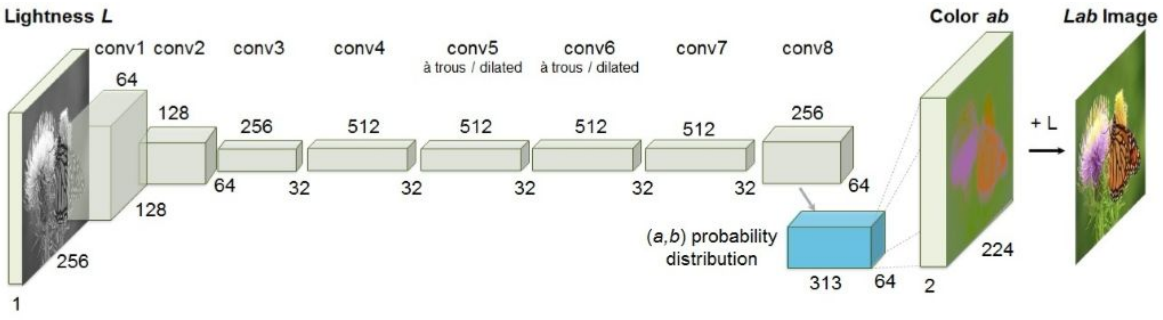
Our survey contains the following models

- Colorful Image Colorization (Encoder - Decoder)
- Deep Koalarization: Image Colorization using CNNs and Inception-Resnet-v2 (Encoder - Decoder)
- Follow up work for the Image Colorization using CNNs and Inception-Resnet-v2 (Encoder - Decoder)
- Learning Representations for Automatic Colorization (Unet with Skip Connections)
- Unsupervised Diverse Colorization via Generative Adversarial Networks (GANs) (Continuous Condition Concatenation)
- ChromaGAN: Adversarial Picture Colorization with Semantic Class Distribution (GANs) (Continuous Condition Concatenation)

Survey

Colorful Image Colorization (5 October 2016)

- Architecture



- Dataset

  ImageNet

- Code

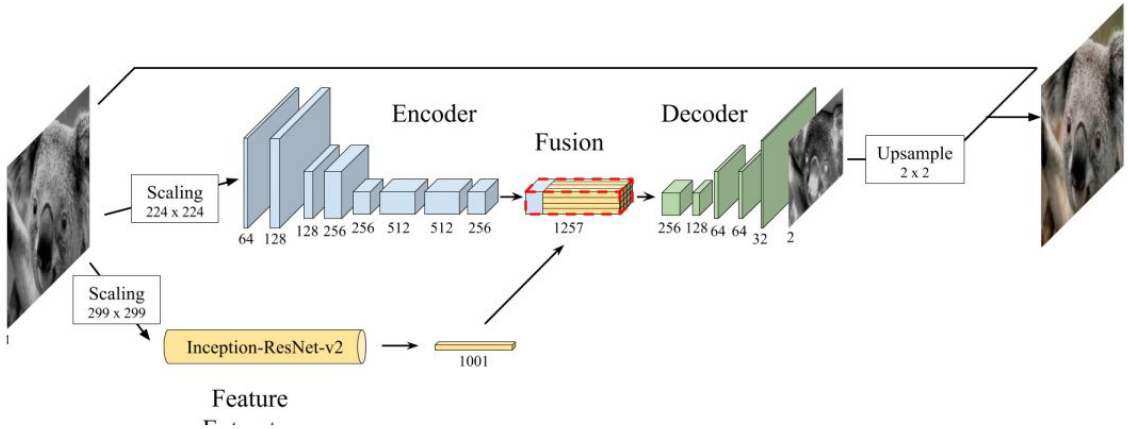
  Available

- Metrics

  Human Evaluation

Deep Koalarization: Image Colorization using CNNs and Inception-Resnet-v2 (2017)

- Architecture



- Dataset

  ImageNet

- Code

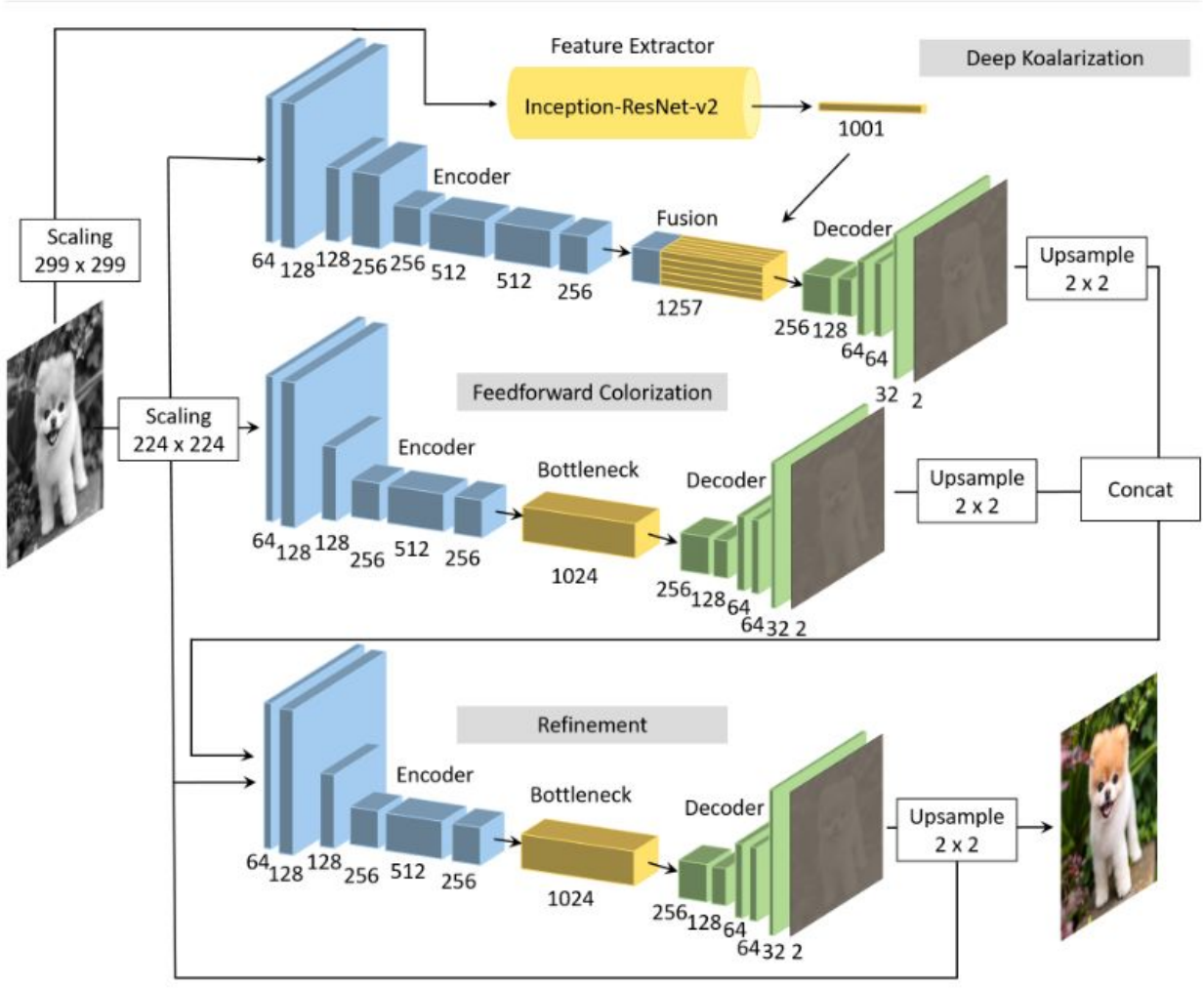
  Available

- Metrics

  Human evaluation

Follow up work for the Image Colorization using CNNs and Inception-Resnet-v2

- Architecture



- Dataset

  ImageNet

- Code

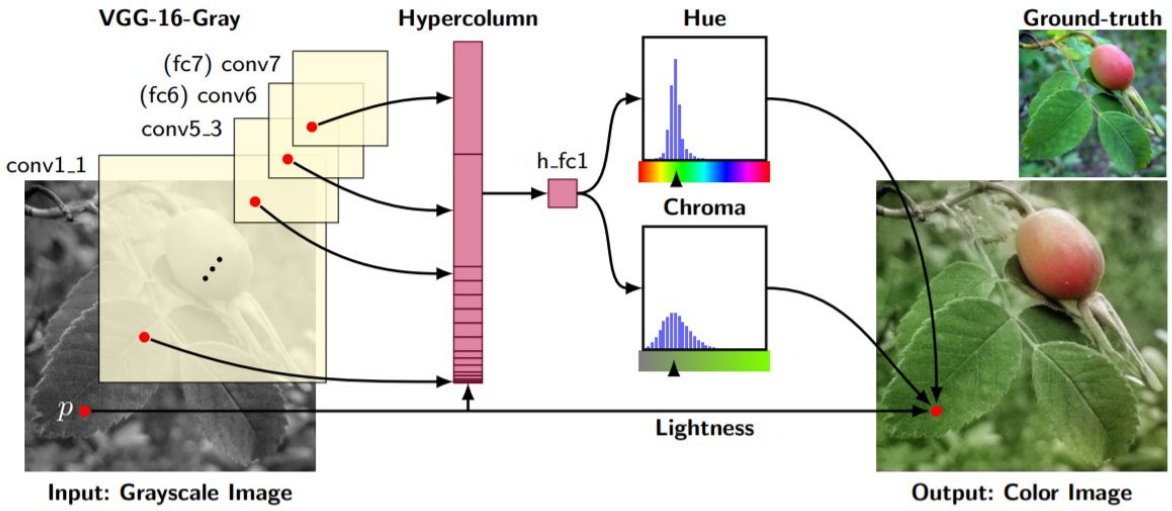
  Available

- Metrics

  Human evaluation

Learning Representations for Automatic Colorization (2017)

- Architecture



- Dataset

  ImageNet

  SUN

- Code

  Available

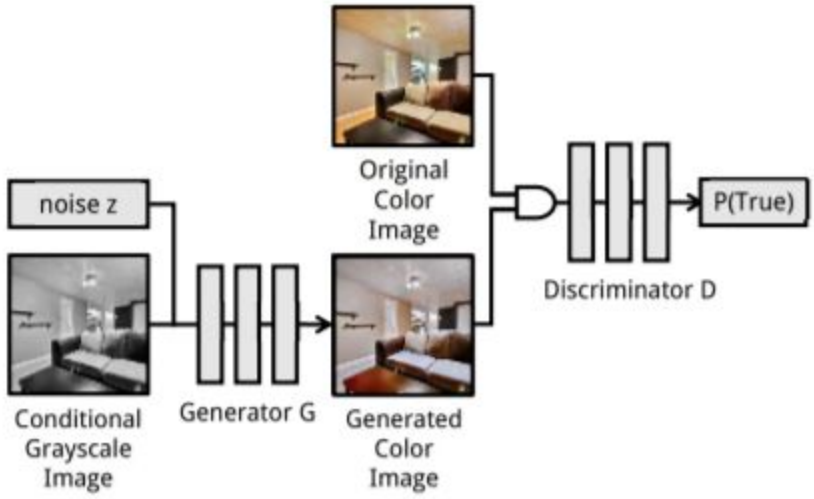
- Metrics

  Human evaluation

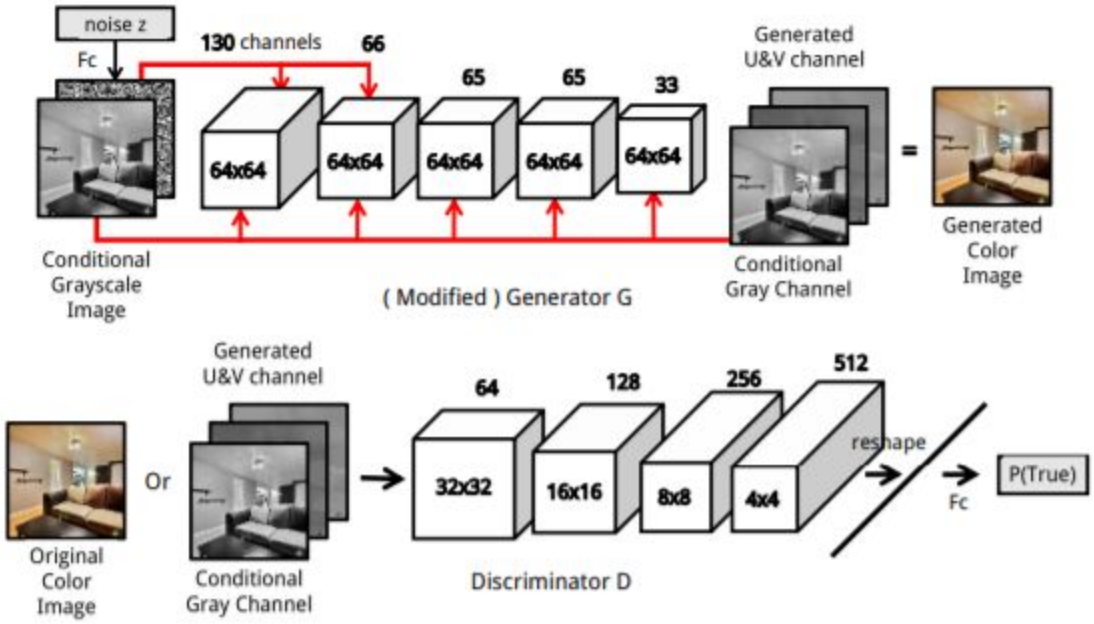
Unsupervised Diverse Colorization via Generative Adversarial Networks (ColorGAN)(2017)

- Architecture

  Simplified Architecture



More Detailed Architecture



- Dataset

  LSUN

- Code

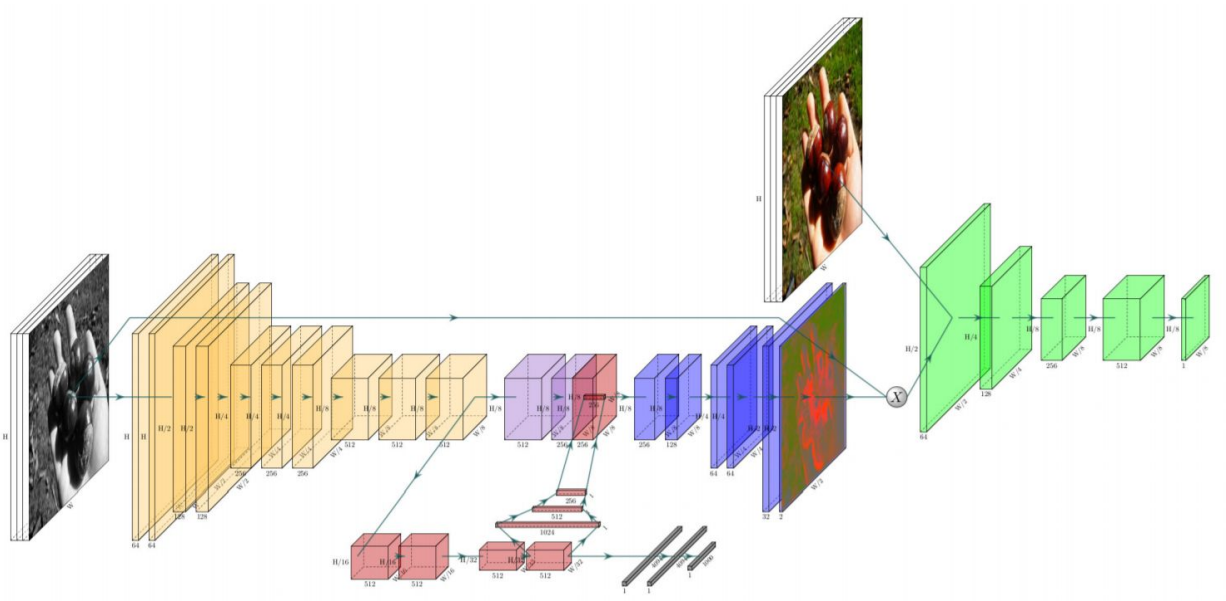
  Available

- Metrics

  Human evaluation

Adversarial Picture Colorization with Semantic Class Distribution (ChromaGAN)(2017)

- Architecture



- Dataset

  ImageNet

- Code

  Available

- Metrics

  Human evaluation

Model Description

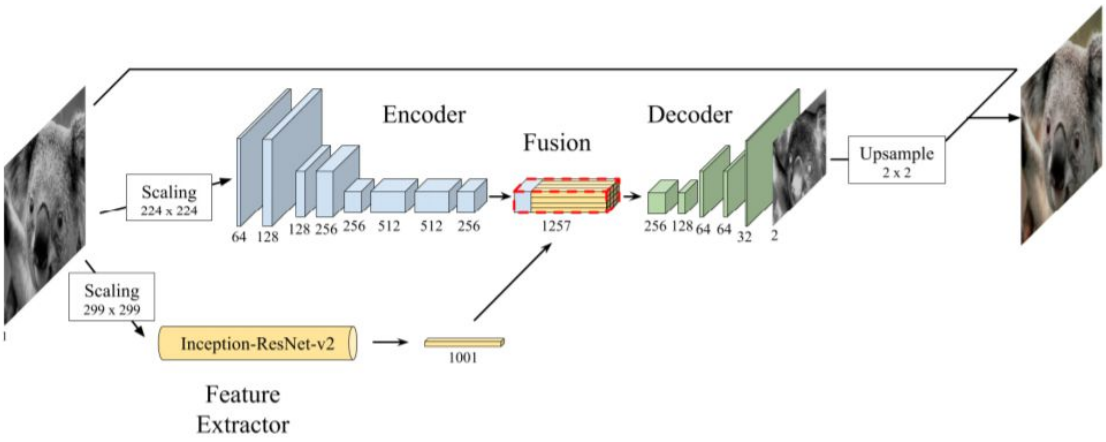
We chose to work on the Encoder-Decoder type of architectures because it has suitable training time compared to gans and we have different variances of the Encoder-Decoder architectures.

We chose our baseline [Deep Koalarization: Image Colorization using CNNs and Inception-Resnet-v2] because it has a better performance than [Colorful Image Colorization] and has an official published paper.

Approach

Given the luminance component of an image, the model estimates its a*b* components and combines them with the input to obtain the final estimate of the colored image.

Architecture details



The architecture mainly composed of four parts:

Encoder:

- Obtains the mid-level features
- processes H × W gray-scale images and outputs a H/8 × W/8 × 512 feature representation.

Feature Extractor:

- Obtains the high-level features.
- use a pre-trained Inception-ResNetv2 network.

Fusion:

- takes the feature vector from Inception and attaches it to the feature volume outputted by the encoder.

- Generates a feature volume of dimension H/8×W/8×256.

Decoder:

- takes the feature volume outputted of dimension H/8×W/8×256 from the fusion
- applies a series of convolutional and up-sampling layers in order to obtain a final image with dimension H × W × 2

Proposed Model Updates

- Hyper parameter tuning.
- Try different changes in the architecture components and study its effects on the results.

  (We may refer to the follow up work for this paper in the updates.)

Survey of available datasets

- ImageNet
- SUN
- LSUN

Our Dataset

> **ImageNet** is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently each node has an average of over five hundred images. Currently its size reaches 14 million images with more than 20,000 categories.
>
> We decided to train our proposed model on ImageNet as it commonly used with this problem statement and to compare our results with the baseline results which use ImageNet.

Evaluation Metrics

Most of the papers used human evaluation and turing tests to evaluate their models.

Some papers referenced another proposed method to evaluate the results which is **RMSE** root mean squared error which compares the original image to the colorization result but it is not a fair evaluation method because it limits the evaluation to the original image even if the generated image was realistic but different from the original one.

Our Evaluation Metrics

Turing test to measure the colorization results. It's about asking some participants (From our classmates) some questions. In each question, we display a number of color images chosen from the baseline model results and our updated model results, and ask them if any one of them is of poor reality. And it arranges all images randomly to avoid any position bias for participants.

Plots/Graphs

Pie chart for the result of the turing test.

Our Graduation project Ideas

- **Breast Mass Classification and Segmentation in Digital Mammogram**

  Breast mass is one of the most distinctive signs for diagnosis of breast cancer, and its marginal information reflects the growth pattern and biological characteristics. Generally speaking, benign masses are regular in shape, and masses with irregular margins are often malignant. So, our problem considers two tasks, predicting if breast mass is benign or malignant and segmenting it. We train two models separately for each task. Mainly, Our network is encoder-decoder architecture: the encoder is a densely-connected CNN and the decoder is a CNN integrated with AGs. Our dataset is CBIS-DDSM dataset, includes approximately 2,500 cases and every case contains two views of each breast, as well as some associated patient information (age, breast density rating, rating for abnormalities and keyword description of abnormalities) and image information (scanner, spatial resolution and so on). Images containing suspicious areas have associated pixel-level ‘‘ground truth’’ information about the locations and types of suspicious regions

- **Text Visualization**

  Text Visualization is a fundamental problem towards automatically generating images according to natural language descriptions. We use generative adversarial networks to generate images for text descriptions. Our baseline is Attngan and we study the effect of adding bert model in the preprocessing phase.

  Our dataset : CUB