Automatic Colorization

Minghui Zhou, Pengyang Zhou

1. Motivation

- Some gray photos are the memories of old times.
- People use different ways to add color to old gray photos. E.g. posting a request to Facebook or Reddit, using photoshop, etc.

1930's Chicago

Colorized Manually

1. Motivation

- Some gray photos are the memories of old times.
- People use different ways to add color to old gray photos. E.g. posting a request to Facebook or Reddit, using photoshop, etc.

Surf Girls, 1938–46

2. Project Overview

- Project
  - Investigated some possible solutions to the colorization problem.
  - Implemented convolutional-neural-network- based (CNN) and Deep Convolutional Generative Adversarial Networks-based (GAN) colorization and made some improvements on model performance.
- Input: grayscale images.
- Goal: automatically create plausible, natural-looking images.
- Challenge:
  - Colorful image have 3-Dimensions information, but grayscale images only have 1-dimension information.
  - Inherent Ambiguity: single grayscale image may have multiple plausible colorizations

3. Experiment Environment Setup

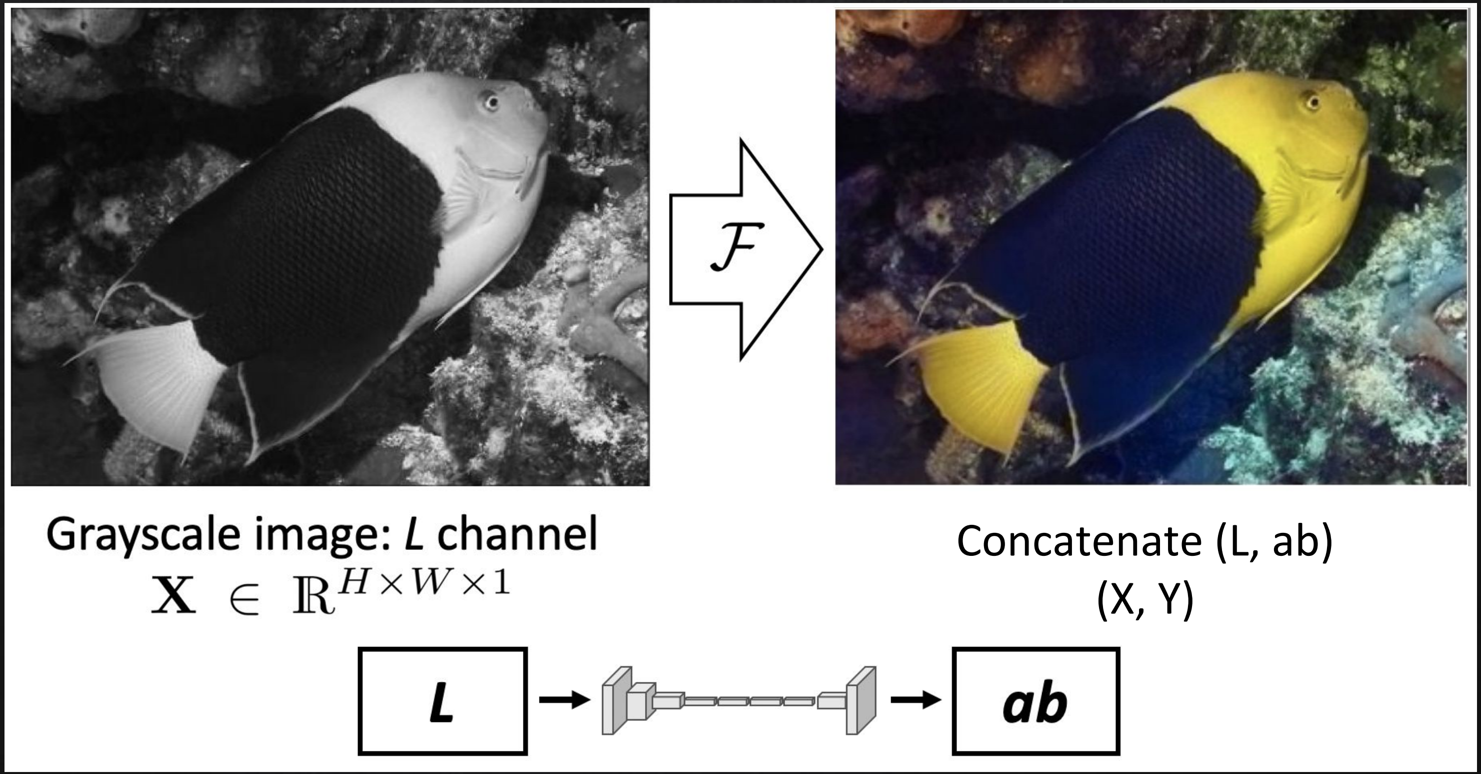
- 2 * Google Cloud Deep Learning VM Instances
  - NVIDIA Tesla P100 GPU, 13 GB memory
- Dataset: subset of Places365
  - Places365: 1.8 million train images
  - Image size: 256*256
  - Subset: forest, park, garden... (outdoor scenes)

4. CNN Based Colorization

- Iizuka et al. 2016 – Let there be Color! Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization
- Larsson et al. 2016 – Learning Representations for Automatic Colorization
- Zhang et al. 2016 – Colorful Image Colorization

	Iizuka et al.	**Larsson et al.**	**Zhang et al.**
Model Architecture	Two-stream convolutional network	Hypercolumn approach	Single-stream convolutional network
Training Dataset	Places365	ImageNet	ImageNet
Loss Function	Regression (mean squared error)	Classification (cross-entropy)	Classification (cross-entropy)

4. Zhang et al.



$\mathcal{F}$
Grayscale image: *L* channel
$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$
Concatenate (L, ab)
(X, Y)
L
ab

Zhang et al.

Zhang et al.

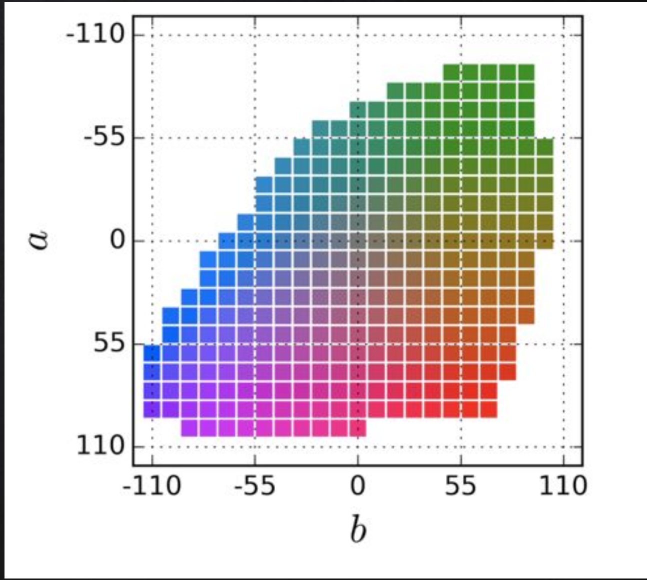
Zhang et al.: Loss Function

- Due to the multimodal nature of colorization:
  - Square Error Loss is not robust.

$$L_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

- Use multinomial classification
  - Divide the output ab space into discrete bins of size 10
  - Predict a distribution of possible colors for each pixel
  - Use multinomial cross entropy loss

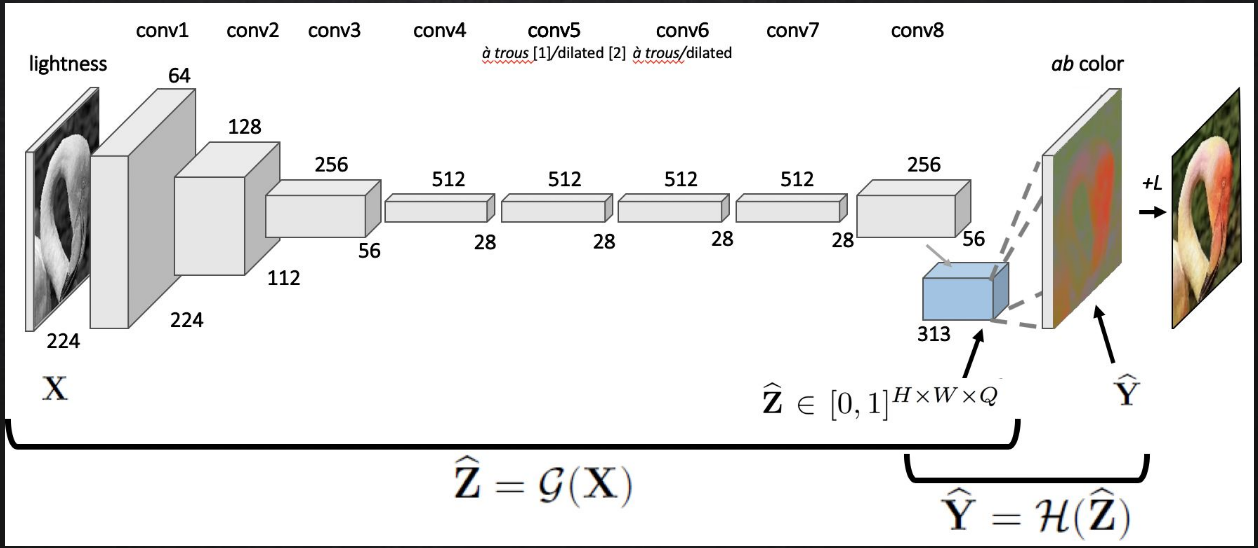
$$L(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\frac{1}{HW} \sum_{h,w} \sum_{q} \mathbf{Z}_{h,w,q} \log(\widehat{\mathbf{Z}}_{h,w,q})$$



Zhang et al.: Network architecture

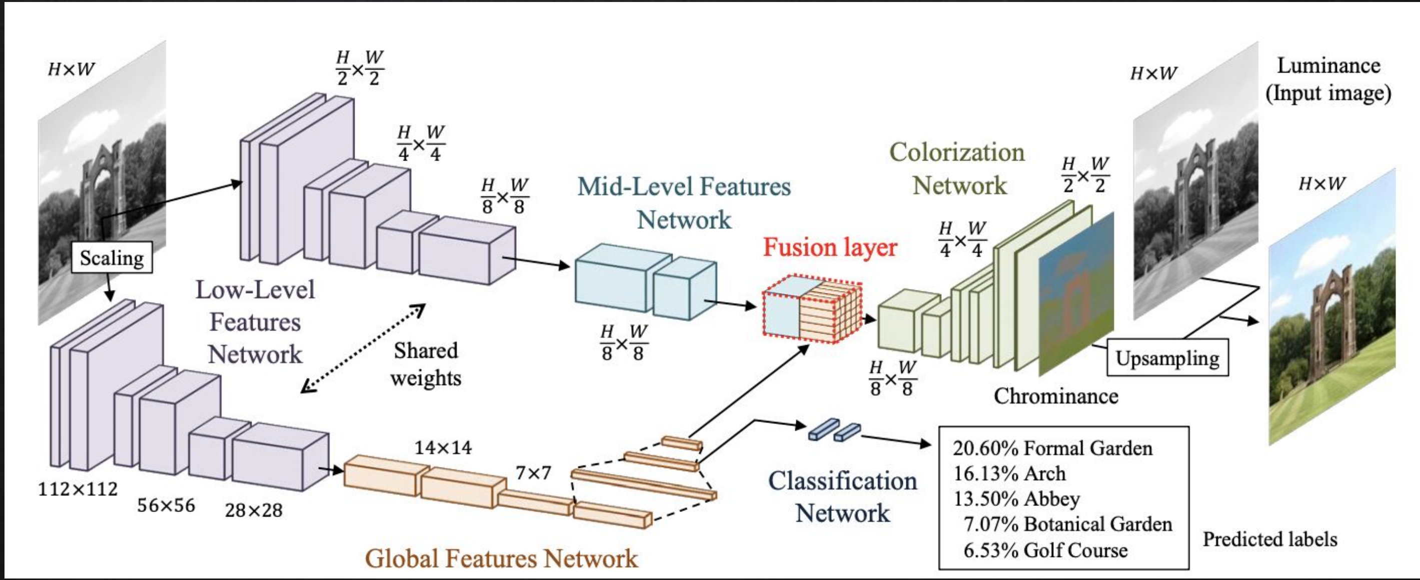
- Each conv layer refers to a block of 2 or 3 repeated conv and ReLU layers.
- No pooling layer

Image size is changed through spatial downsampling or upsampling



4. Lizuka et al.

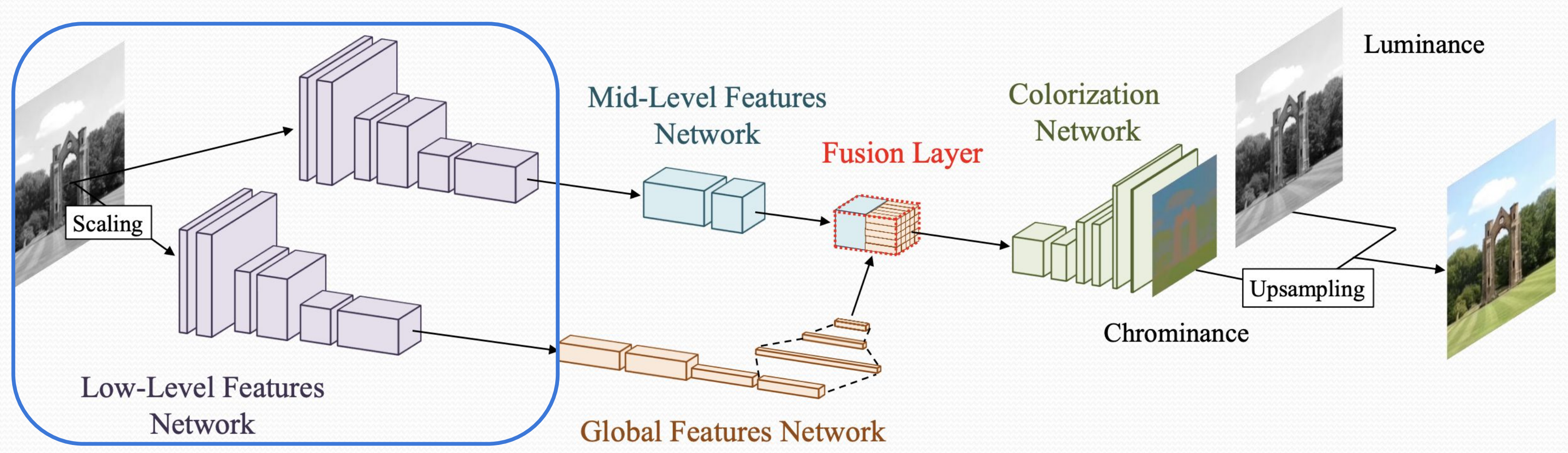
An end-to-end network that jointly learns global and local features for automatic image colorization.



- Two branches: local features and global features
- Composed of four networks

Lizuka et al.: Low-Level Features Network

An end-to-end network that jointly learns global and local features for automatic image colorization.

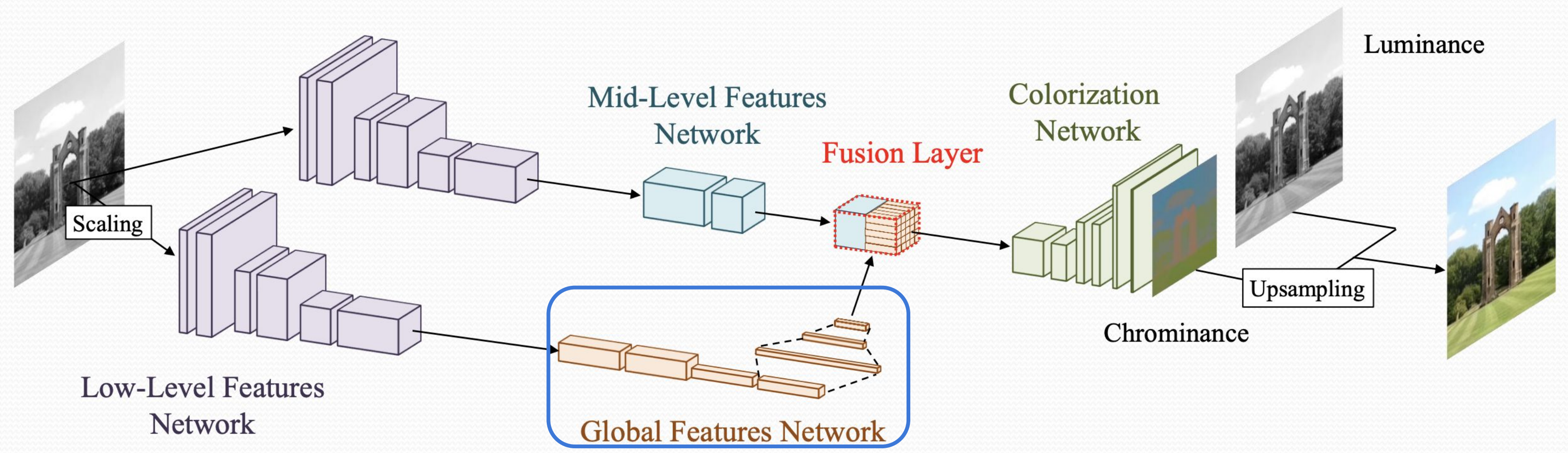


- Extract shared low-level features such as edges and corners
- Instead of using max-pooling layers, it uses convolution layers with increased strides to reduce size.

  (If padding added, the output is effectively half the size of the input layer)

Lizuka et al.: Global Features Network

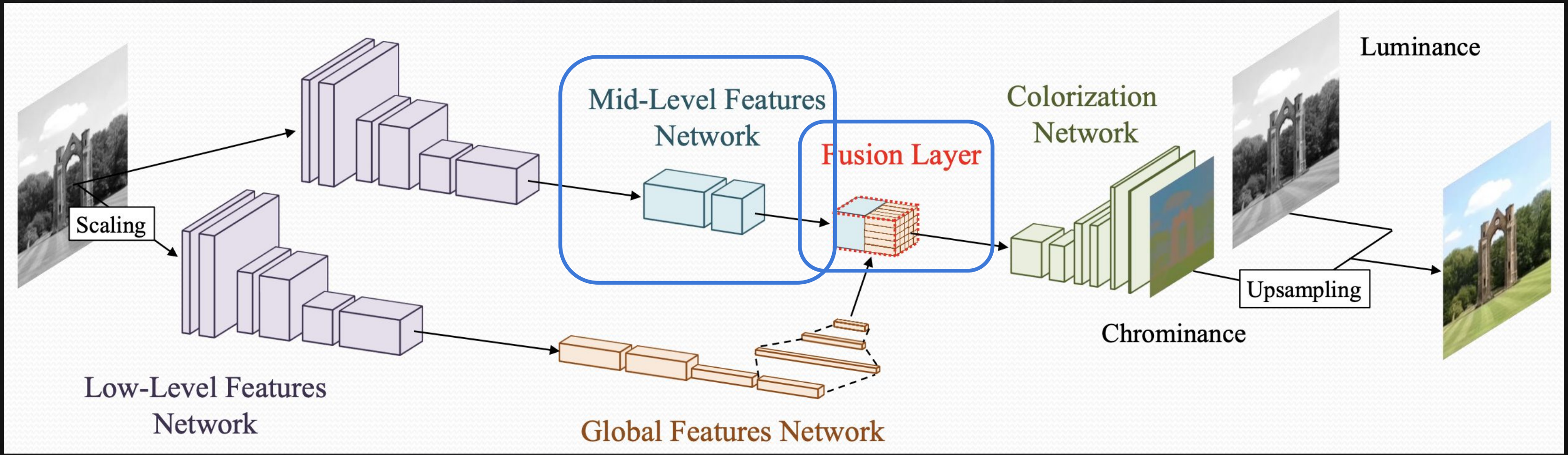
An end-to-end network that jointly learns global and local features for automatic image colorization.



- Further process low-level features with 4 convolutional layers followed by 3 fully-connected layers.
- Compute a **global** 256-dimensional vector representation of the image

Lizuka et al.: Mid-Level Features Network

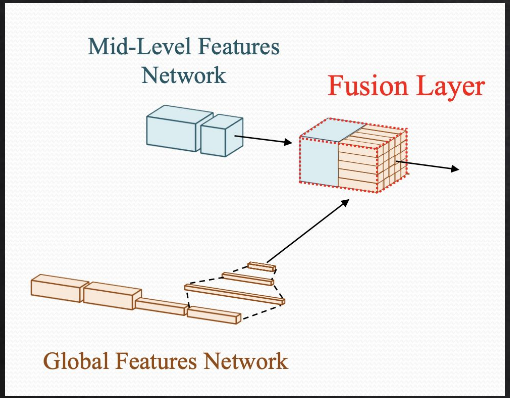
An end-to-end network that jointly learns global and local features for automatic image colorization.



- Further process low-level features with 2 convolutional layers.
- Low-Level and Mid-level features network are fully CNN.

Lizuka et al.: Fusion Layer

- Fusion layer incorporates the global features into local features.
- It effectively combines the two kinds of features and obtains a new feature map.

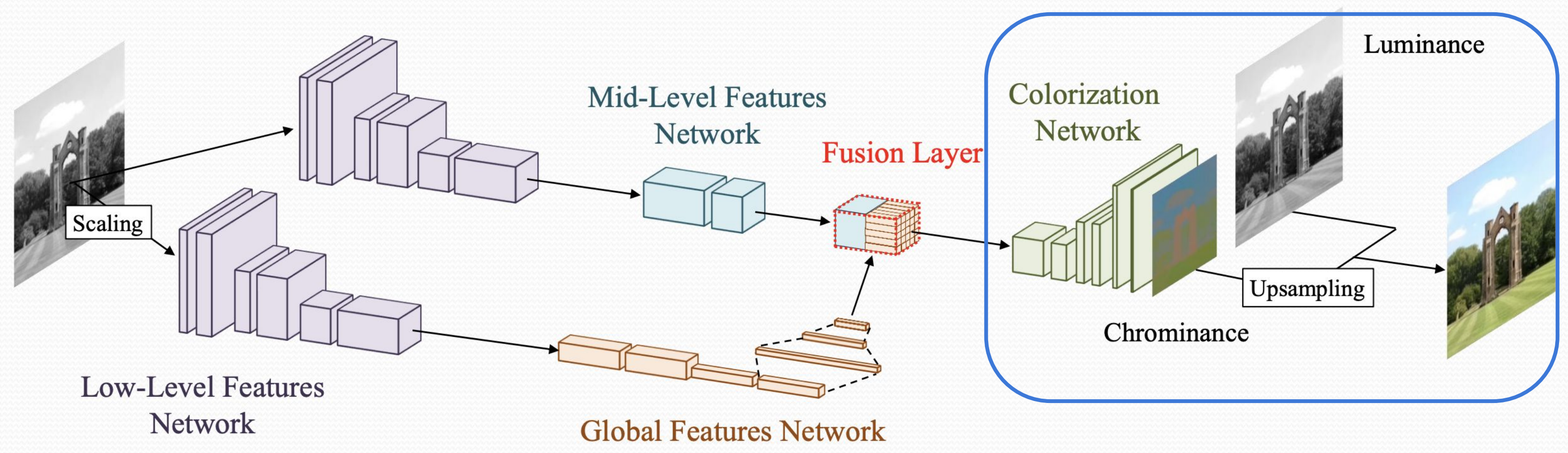


$$\mathbf{y}_{u,v}^{\text{fusion}} = \sigma\left(\mathbf{b} + W\begin{bmatrix}\mathbf{y}^{\text{global}} \\ \mathbf{y}_{u,v}^{\text{mid}}\end{bmatrix}\right)$$

- Global features: 256-dimensional vector
- Local (Mid-level) features: H/8*W/8*256.

Lizuka et al.: Colorization Network

An end-to-end network that jointly learns global and local features for automatic image colorization.



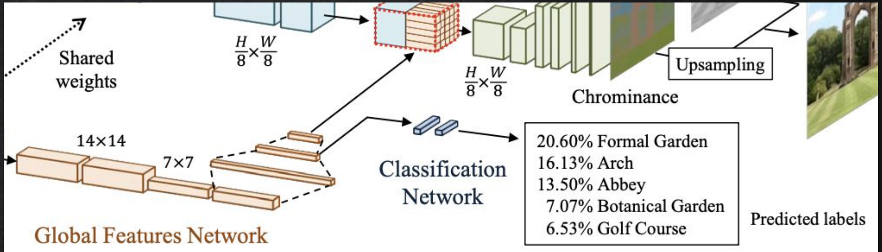
- Convolution layers + Upsampling layers.
- Upsampling layers: nearest neighbor technique.
- Output layers: convolutional layer with a sigmoid transfer function.
- Output: Chrominance (a, b channel) + Luminance (Lightness channer) = colorful image

Lizuka et al. – Loss function

Loss function:

- Mean Squared Error

  Couldn't learn global content properly

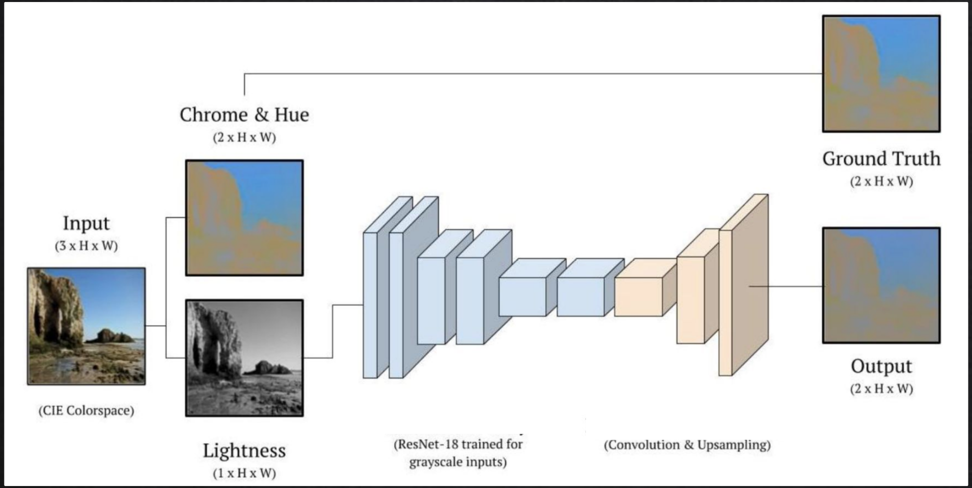


- Facilitate the optimization by also training for classification jointly with the colorization.
  - Add 2 FC layers to predict the class label of image.
  - MSE loss for colorization, cross-entropy loss for classification

$$
\begin{aligned}
L(y^{\text{color}}, y^{\text{class}}) = {} & \|y^{\text{color}} - y^{\text{color},*}\|^2_{\text{FRO}} \\
& - \alpha \left(y^{\text{class}}_{l^{\text{class}}} - \log \left(\sum_{i=0}^{N} \exp \left(y_i^{\text{class}} \right) \right) \right)
\end{aligned}
$$

4. Our CNN Model

- Construct a deep convolutional neural network

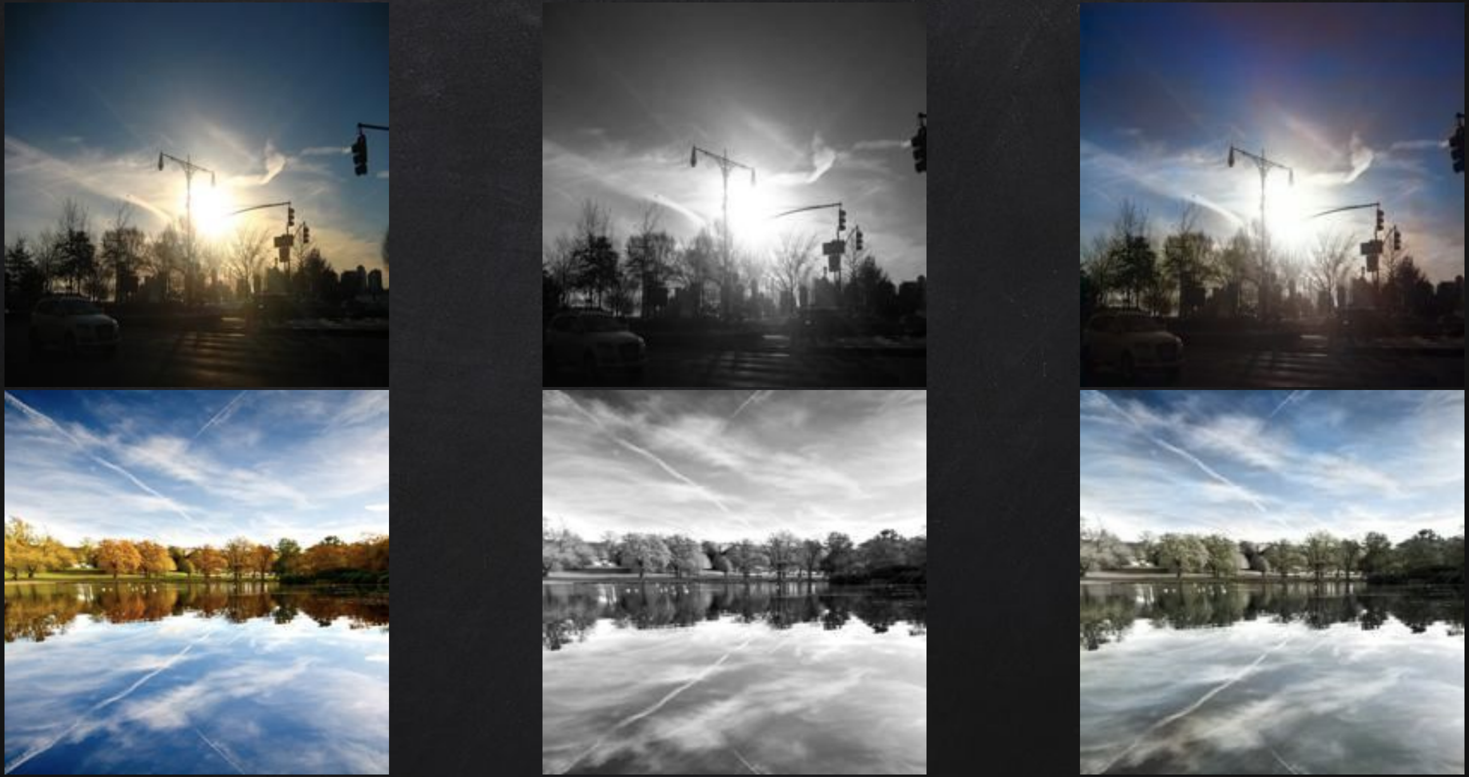


- Pre-training a ResNet-18 classifier
- Using pretrained ResNet model before
- ResNet classifier + 6 convolution layers + upsample layer
- Tried MSE and cross-entropy loss function

Type	Kernel	Stride	Output
conv.	4*4	2	128
conv.	3*3	1	128
conv.	3*3	1	64
conv.	3*3	1	64
conv.	3*3	1	32
conv.	3*3	1	2
upsample	-	-	2

CNN – Experiment Result

original | grayscale | colorization



CNN – Experiment Result

Colored by machine

CNN – Experiment Result






CNN - Experiment Result



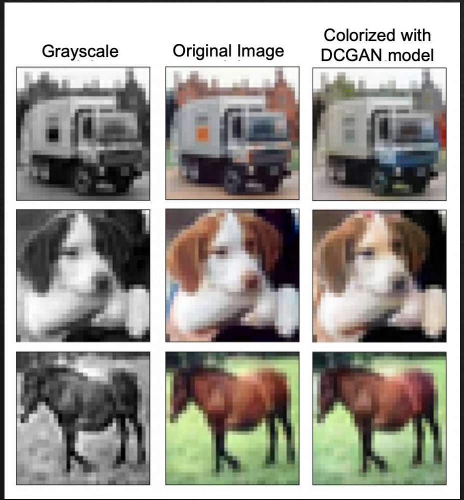


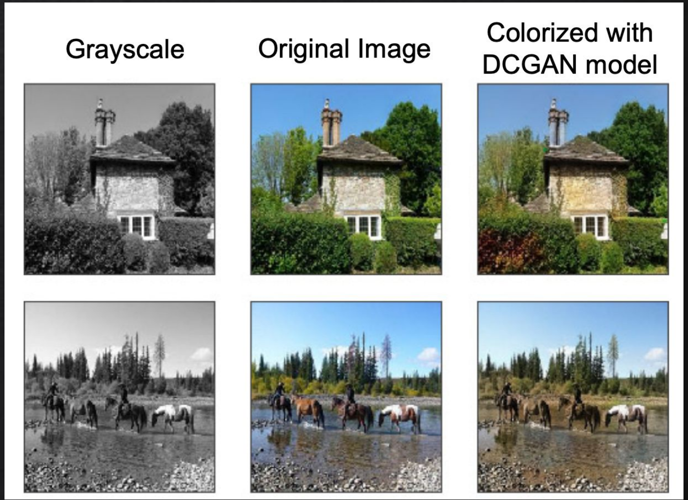

5. Image Colorization with DCGAN

- Nazeri, K., Ng, E., & Ebrahimi, M. (2018). Image Colorization with Generative Adversarial Networks.

This paper proposed a solution to image colorization problem by using a conditional Deep Convolutional GAN and trained it on CIFAR-10 and Places365 dataset.

32x32





256x256

5. Image Colorization with DCGAN

- GANs (Goodfellow et al., 2014):

  Generator G and Discriminator D. G tries to capture the real data distribution.

  D tries to determine whether a sample came from G or training data.

- DCGAN (Radford et al., 2015):

  Combine the ideas of CNNs with GANs.

  Replace all max polling with convolutional stride.

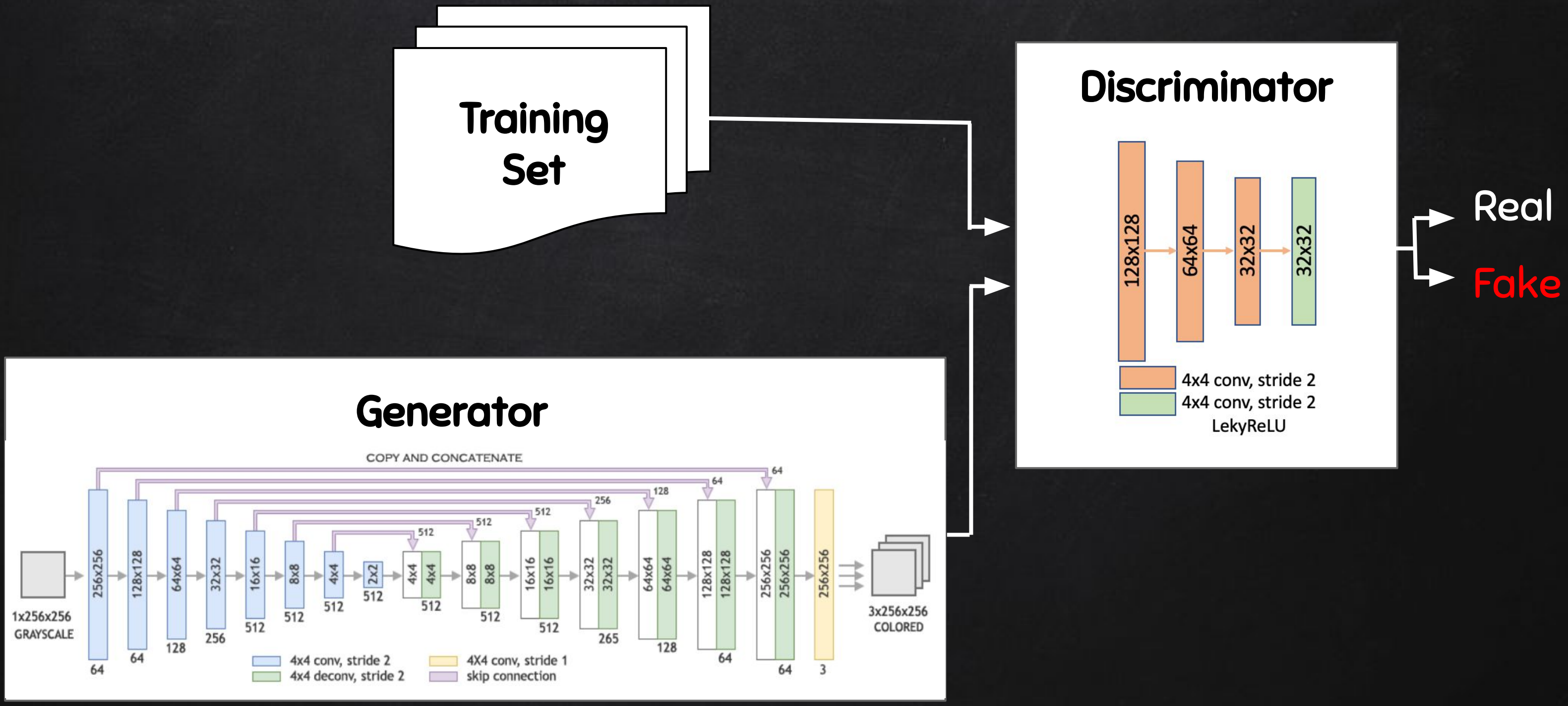
  Use transposed convolution for upsampling.

  ReLU in generator and LeakyReLU in discriminator, etc.

- Conditional GAN (Mirza and Osinero, 2014):

  Input data: grayscale images with zero noise.

DCGAN Model Architecture



Experiments with DCGAN

- Followed the conditional Deep Convolutional GAN architecture.
    - Dataset: subset of Places365
    - Trained on: Google Cloud Platform GPU
- Analyzed the model by training for different epochs.
- Improved the performance by changing the loss function, adjusting learning rate in different epochs, and one-sided label smoothing.
- Compared the testing results.

Observations in Experiments

- The number of epochs for training GAN matters.

  The more, the better? -> Generator – Discriminator Game Tricks

  After some epochs, the network collapse.

Experiment Results

- Original Model – changes for different epochs

original | epoch 10 | epoch 20 | epoch 30

Experiment Results

- Original Model – changes for different epochs

original | epoch 10 | epoch 20 | epoch 30

Observations in Experiments

- Alternative cost function

In the original paper, the loss function for generator is MAE (mean of the absolute error of generated and source images). -> MSE (mean of square error)

- Save several training checkpoints (Highly suggested)

Since the unstable training features of GAN, it is better to store some checkpoints for restoring and resuming training.

Change learning rate when loss function started to plateau.

Experiment Results

- Improved Loss Function – changes for different epochs

original | epoch 15 | epoch 18 | epoch 21 (lr adjusted)

Experiment Results

- Improved Loss Function – changes for different epochs

original | epoch 15 | epoch 18 | epoch 21 (lr adjusted)

Experiment Results

8. References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).
2. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
3. Nazeri, K., Ng, E., & Ebrahimi, M. (2018). Image Colorization with Generative Adversarial Networks. arXiv preprint arXiv:1803.05400.
4. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.
5. Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In European Conference on Computer Vision, pages 577-593. Springer, 2016.
6. Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM Transactions on Graphics (TOG), 35(4):110, 2016.
7. Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In European Conference on Computer Vision, pages 649-666. Springer, 2016.

THANKS!