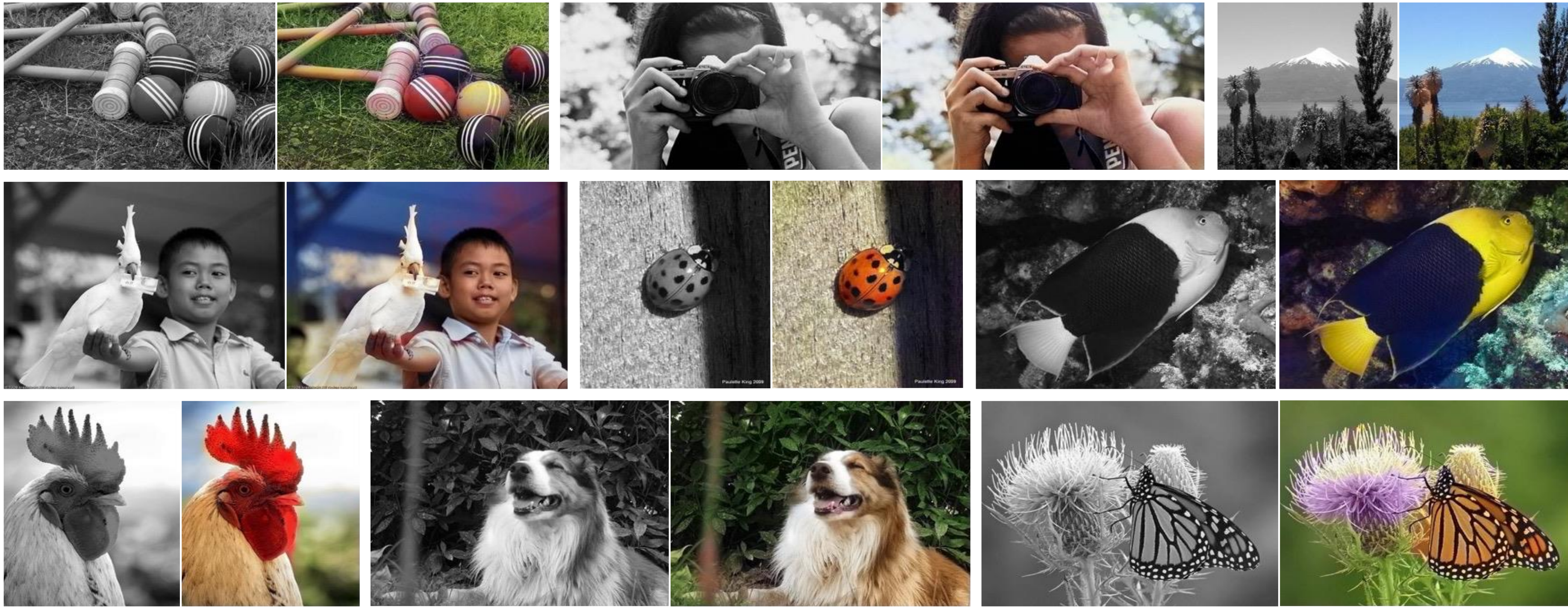


Colorful Image Colorization

황상원

2017.08.18

YONSEI UNIVERSITY
IMAGE AND VIDEO PATTERN RECOGNITION LAB

Contents

❖ **What is Colorization Problem?**

❖ **History of Colorization**
- Prior Work on Colorization
- Concurrent Work on colorization

❖ **Approach of Colorful Image Colorization**

❖ **Application**

What is Colorization Problem?

❖ Colorization 의 어려움

- <u>Two out of the three dimension</u> has been lost.

Gray Channel

What is Colorization Problem?

❖ Colorization 의 어려움

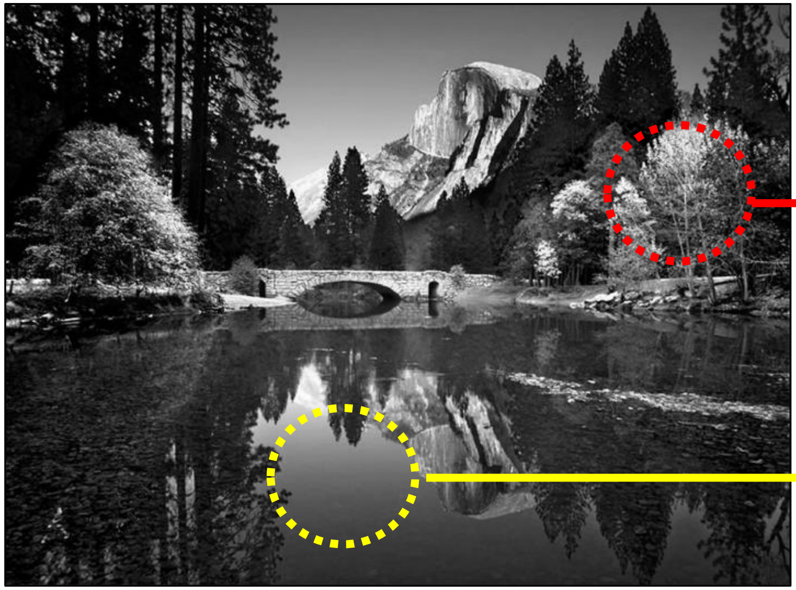
- Two out of the three dimension has been lost.

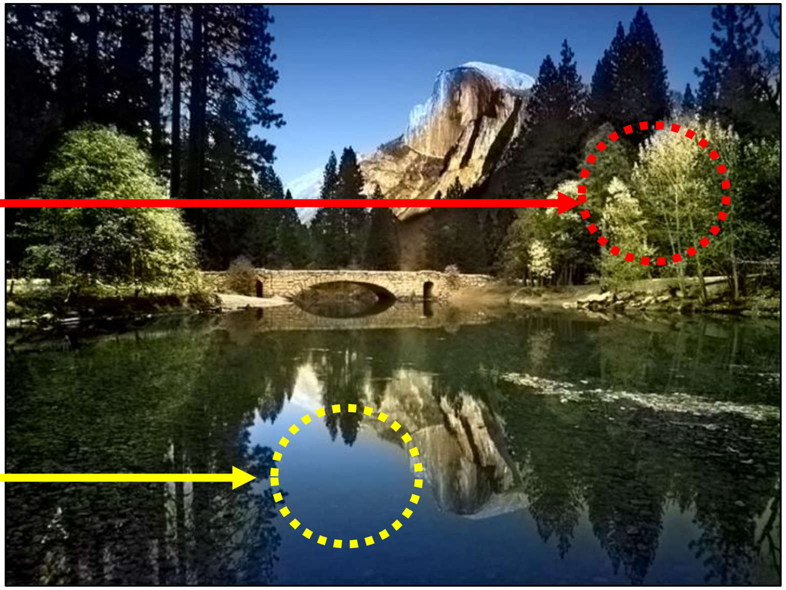
RGB Channel

What is Colorization Problem?

❖ Colorization 의 쉬움

- <u>Semantics</u> of the scene and its surface <u>texture</u> provide ample cue
  - ✓Ex. Grass → Green, Sky → blue..
  - ✓But Apple → Red? Green?





History of Colorization

❖ Prior Work on colorization

- Non-parametric Methods : Modeling correspondence between Grayscale and color. [17]

  Image Analogies

- Parametric method: learn prediction functions from large datasets of color images at training time, posing the problem as either regression onto continuous color space [1,2,22] or classification of quantized color values [3]

  Deep Colorization

❖ Concurrent Work on Colorization

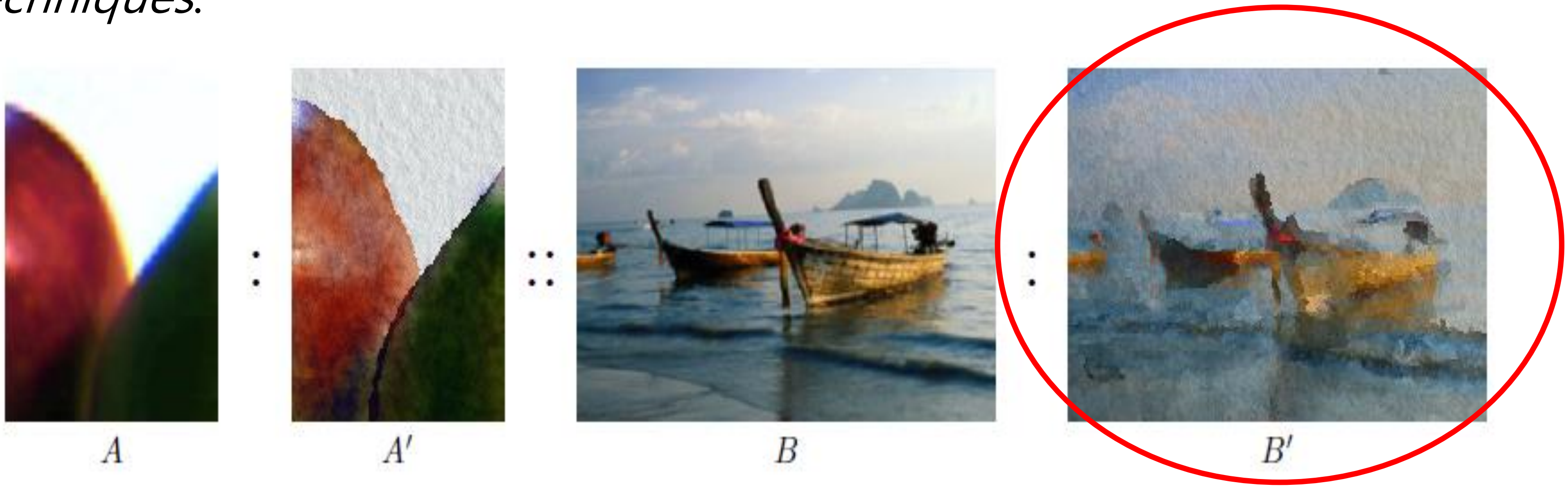
- Larsson[23] : **un-rebalanced classification loss**.

  ✓Hypercolumns [25]

Image Analogies Frame

History of Colorization

[ACM, 2001] Hertzmann, Aaron, et al. "**Image analogies**." *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*.



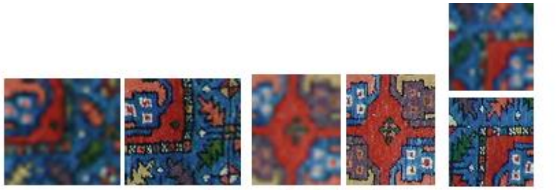
Training Image : A, A′

Input Image: B

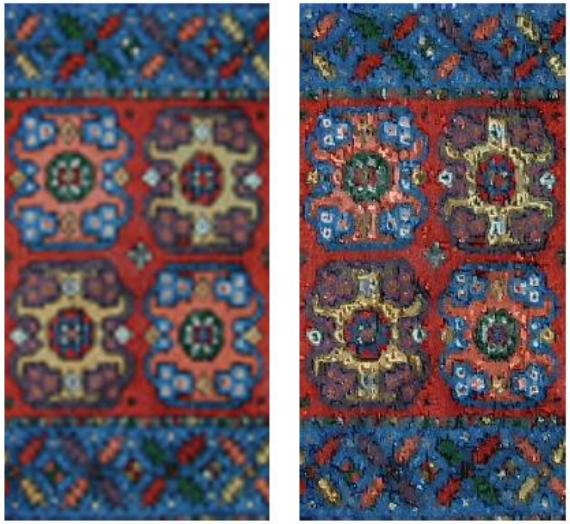
Output Image : B′

History of Colorization

Texture Transfer

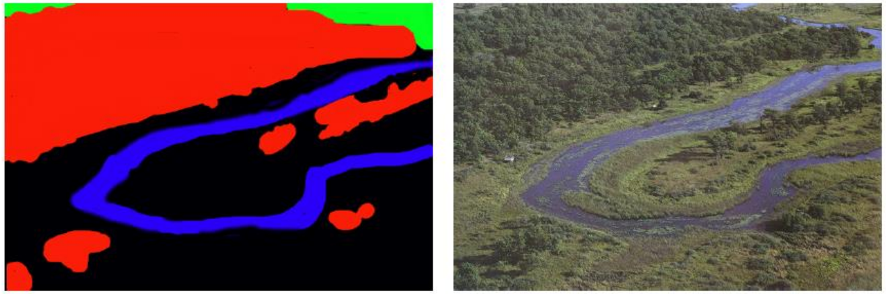


Training pairs (A, A')

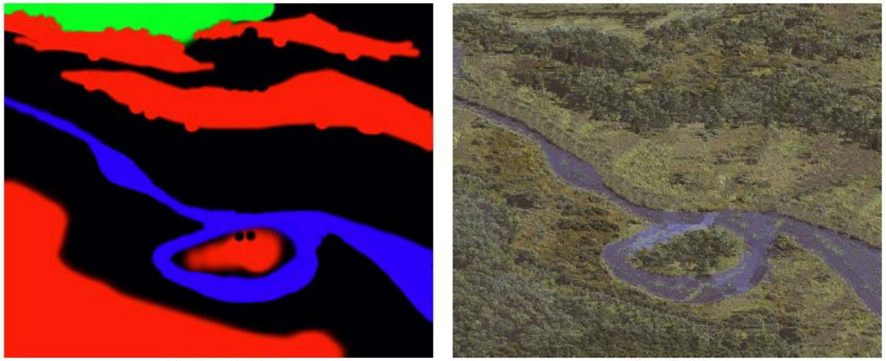


Unfiltered target (B) Filtered target (B')

Super-resolution



Unfiltered source (A) Filtered source (A')



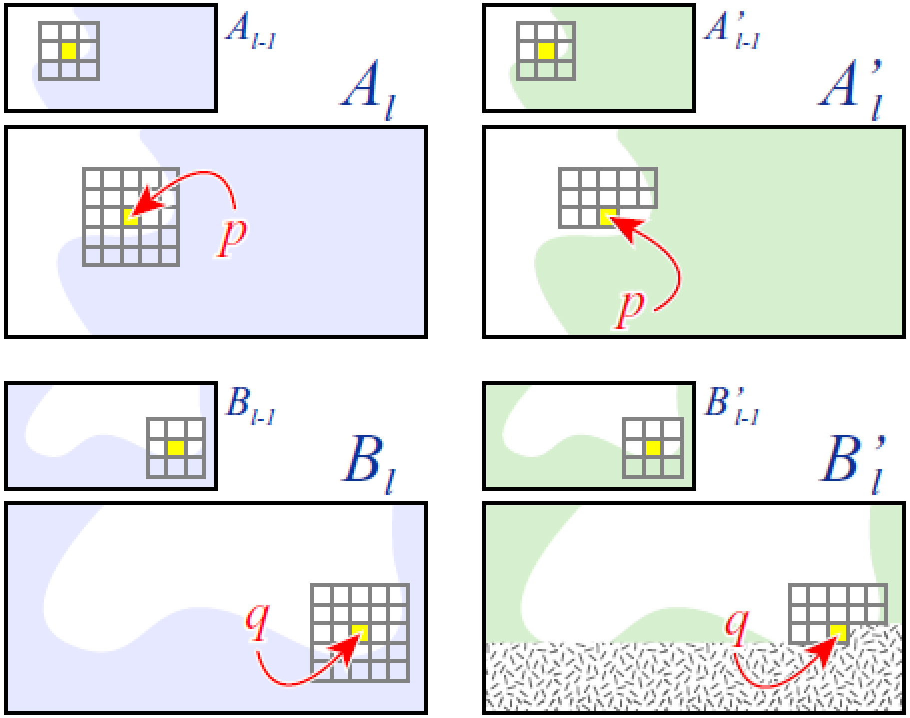
Unfiltered (B) Filtered (B')

Texture-by-numbers

History of Colorization

- Pseudocode

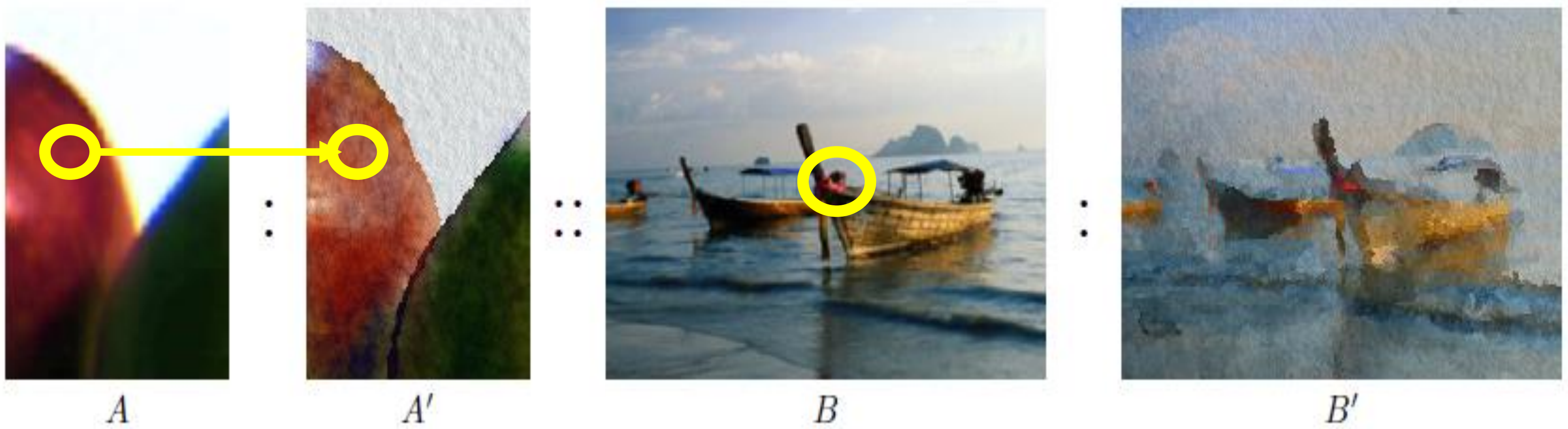
function CREATEIMAGEANALOGY(A, A', B):
  Compute Gaussian pyramids for A, A', and B
  Compute features for A, A', and B
  Initialize the search structures (e.g., for ANN)
  for each level ℓ, from coarsest to finest, **do**:
    for each pixel $q \in B'_\ell$, in scan-line order, **do**:
      $p \leftarrow$ BESTMATCH(A, A', B, B', s, ℓ, q)
      $B'_\ell(q) \leftarrow A'_\ell(p)$
      $s_\ell(q) \leftarrow p$
  return B'_L



History of Colorization

- Feature

RGB channel → **Curse of Dimensionality**



Ex) R: 200, G: 40, B: 0 → R: 170, G: 100, B: 0

R: 170, G: 40, B: 0 → ?

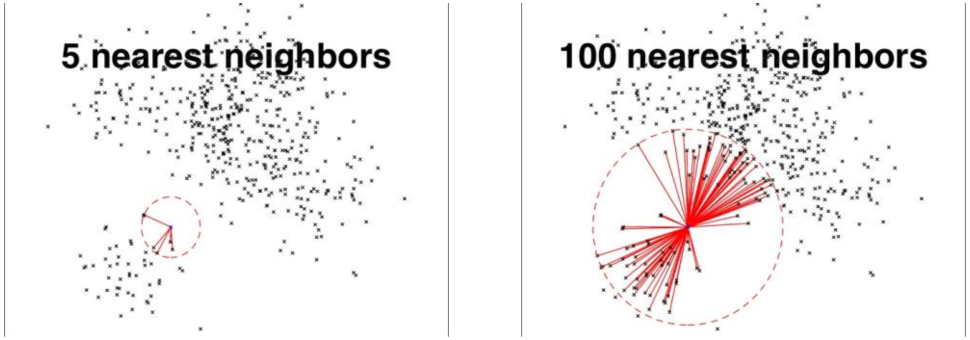
:RGB 는 3차원 (255 * 255 * 255) → Poorly Match (sparser sampling)

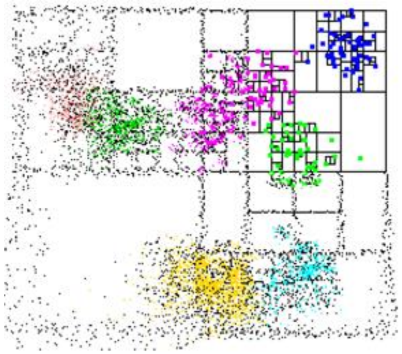
Grayscale 은 1차원 (255) .**But** 정보가 너무 없다.

History of Colorization

- Match

: ANN(Approximate Nearest Neighbor Searching)



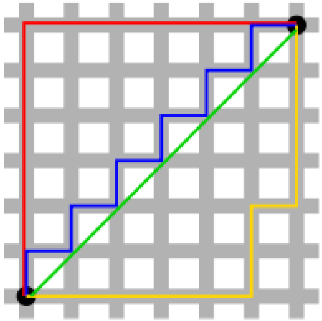


General NN

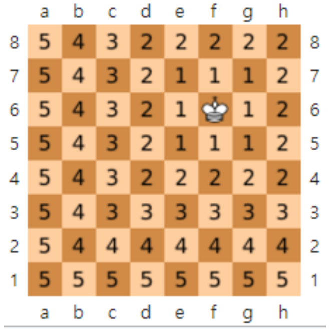
Approximate NN

Computing exact nearest neighbors in dimensions much higher than 8 seems to be a very difficult task.

: Distance Function



초록 : Euclidean distance
파랑 : Manhattan distance



Chebyshev Distance
(=Maximum metric)

Scribbling-based method

History of Colorization

- Scribbling-based method

Marked by user

Colorized images

Original Images

- Example-based colorization

Grayscale Image

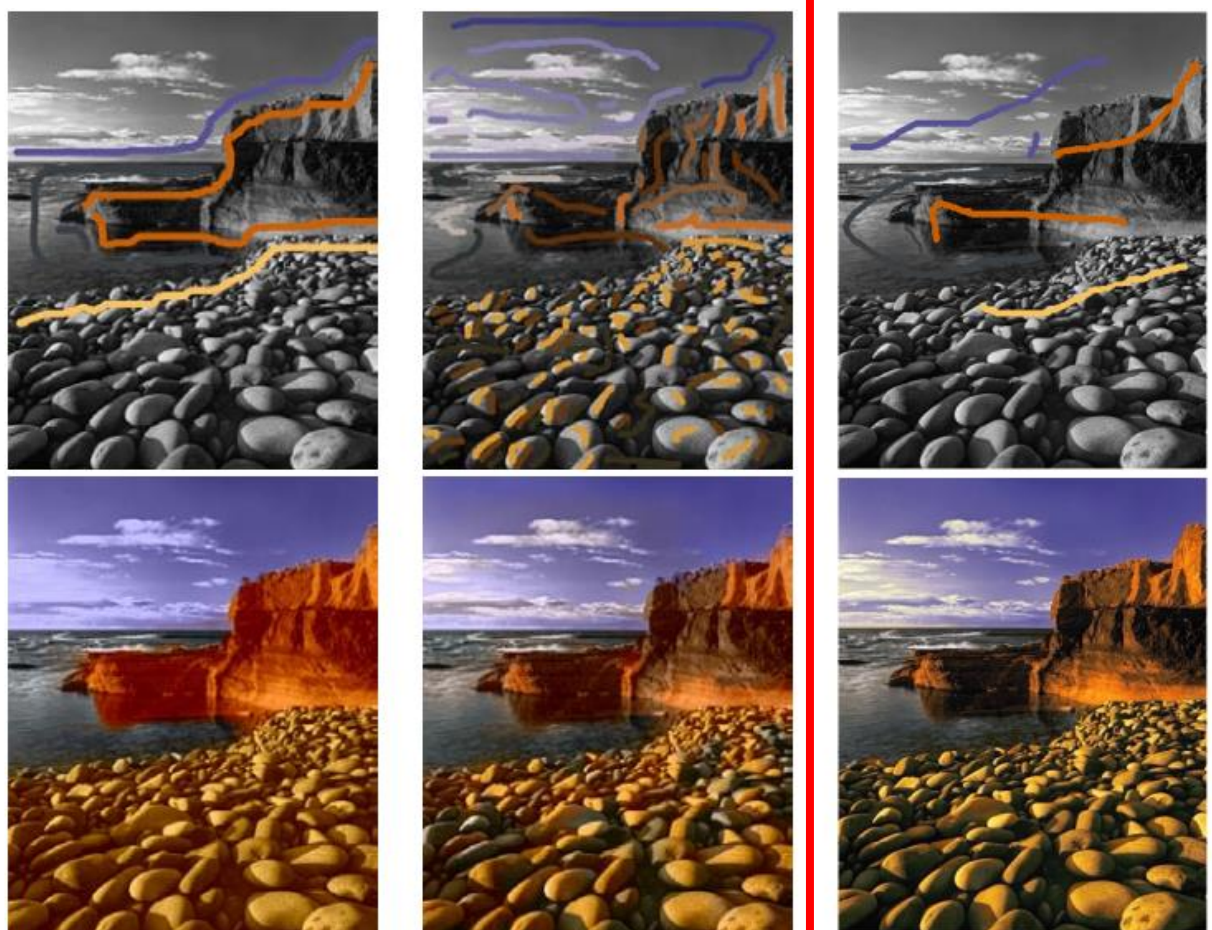
Colorized images

Original Images

❖ Scribble-based colorization

[2007. EGSR] Natural Image Colorization

1.Texture feature



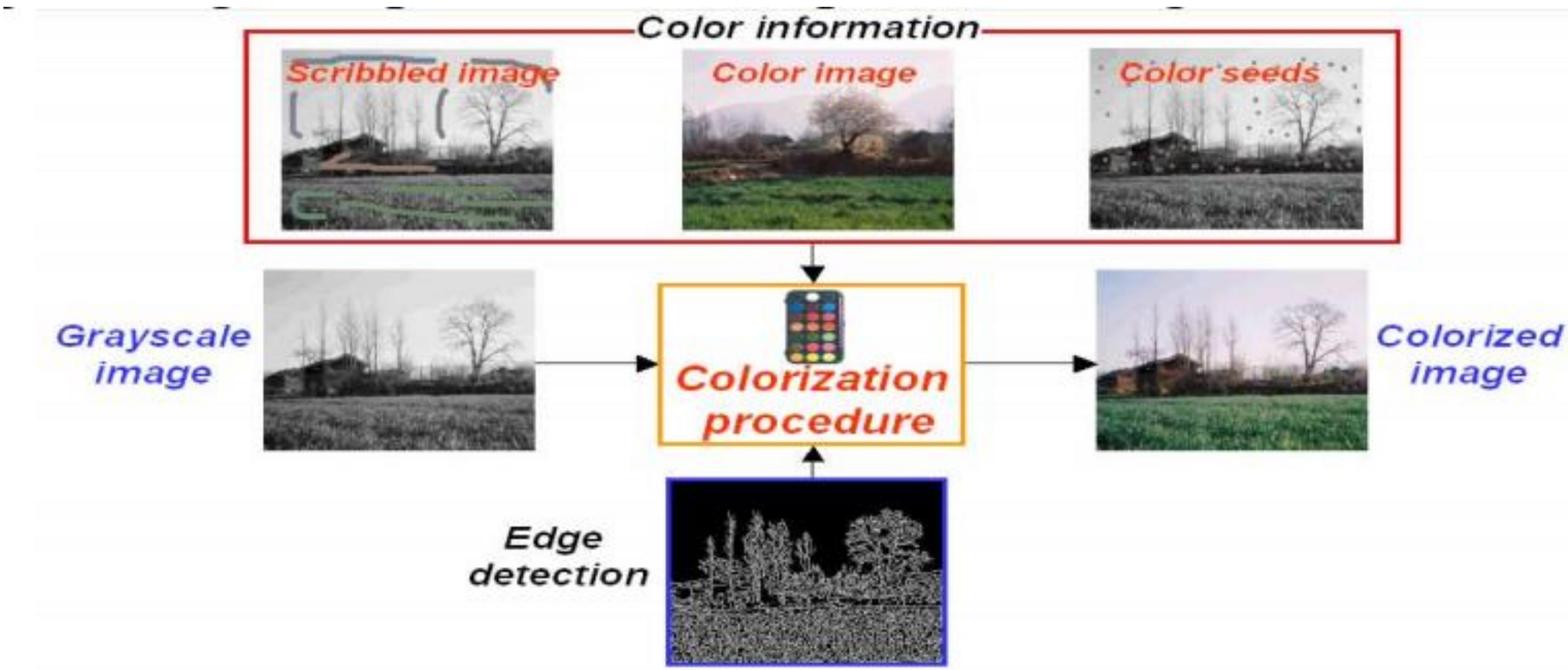
(a) Levin's result 1 (b) Levin's result 2 (c) Our result

YONSEI UNIVERSITY
IMAGE AND VIDEO PATTERN RECOGNITION LAB

❖ Scribble-based colorization

[2005. ACM] Natural Image Colorization

2. Edge feature



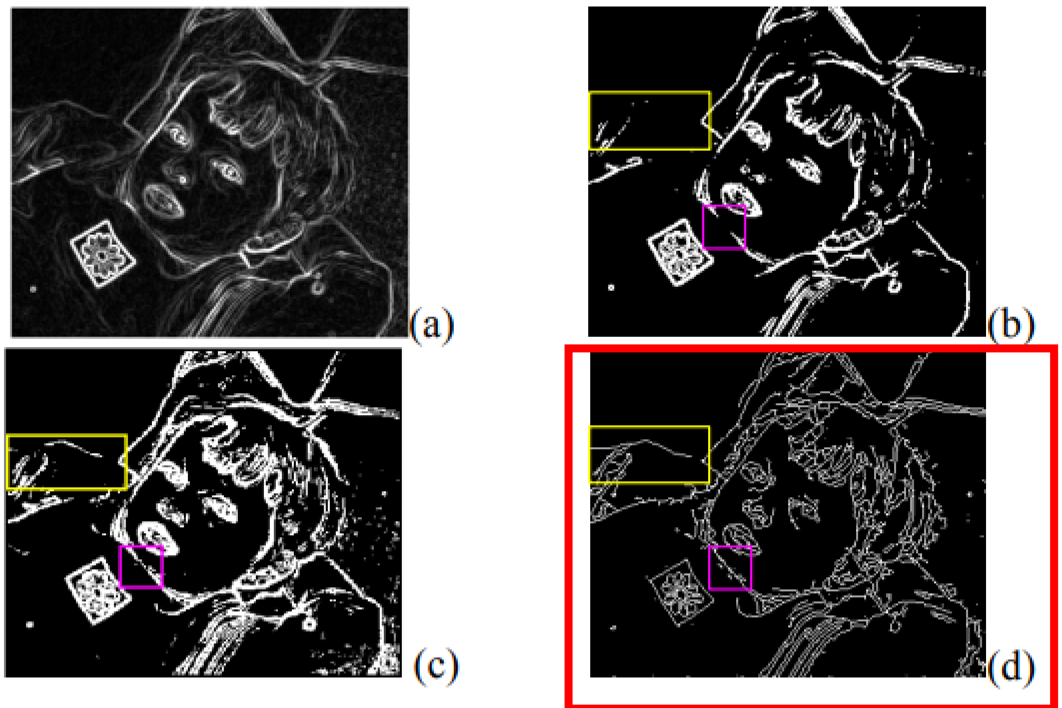
❖ Deep colorization

Scribble-based colorization

[2005. ACM] Natural Image Colorization

Edge feature

Bleeding 막을라고



(a) Edge energy of the Sobel filter output,
(b) Sobel edge detection with static threshold,
(c) edge detection by the popular image processing software, Photoshop 7.0
(d) the result of the proposed adaptive edge detection.

❖ Example-based colorization

(1) User provide a suitable reference image

→ suitable 이미지 찾는게 어려움

(2) Web-supplied examples

❖ Deep colorization

처음으로 Deep 으로 접근

Automatic by using large set of reference images

Deep Colorization

❖ Deep Colorization

- Large-scale reference dataset 사용(by using DL)
- For Artifact-free quality, a joint bilateral filtering based post-processing step is proposed

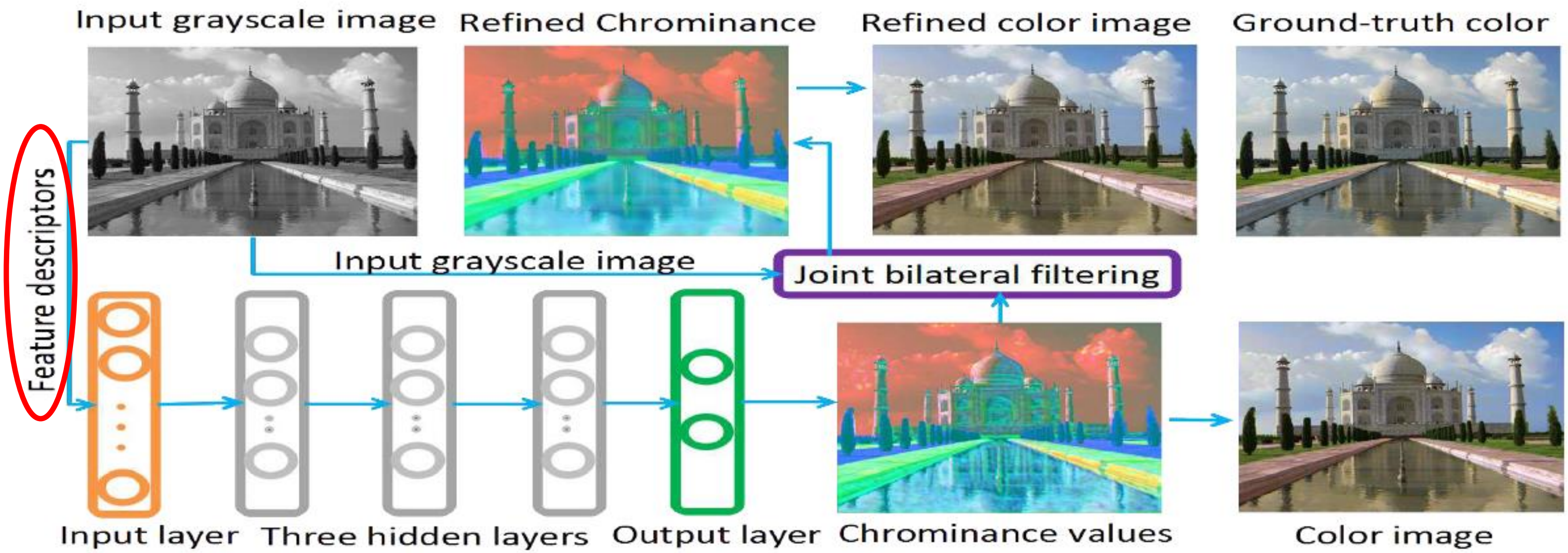
- Scribble-based method → burden on users.
- Example-based method → Highly **depends** to **a** **reference image.**

- Fully-automatic colorization → One reference image can't include all possible scenario. Matching noise is too high. **Images** must be needed

❖ Over View

Input image: **Feature descriptors** of Gray Scale image.

Output image: UV(color) channel.



Algorithm 1 Image Colorization – Training Step

Input: Pairs of reference images: $\Lambda = \{\vec{G}, \vec{C}\}$.
Output: A trained neural network.

1. Compute feature descriptors $\vec{x}$ at sampled pixels in $\vec{G}$ and the corresponding chrominance values $\vec{y}$ in $\vec{C}$;
2. Construct a deep neural network;
3. Train the deep neural network using the training set $\Psi = \{\vec{x}, \vec{y}\}$.

Algorithm 2 Image Colorization – Testing Step

Input: A target grayscale image I and the trained neural network.
Output: A corresponding color image: $\hat{I}$.

1. Extract a feature descriptor at each pixel location in I;
2. Send feature descriptors extracted from I to the trained neural network to obtain the corresponding chrominance values;
3. Refine the chrominance values to remove potential artifacts;
4. Combine the refined chrominance values and I to obtain the color image $\hat{I}$.

G: Grayscale image
C: corresponding image

❖ Idea

- There exists a complex gray-to-color mapping function **F**

(regression model → 얘네의 한계, 너무 DL을 믿음.)

- Deep Neural Network universal approximator that can represent arbitrarily complex continuous functions.

- For a pixel p in G, the output of F is simply the U and V channels of corresponding pixel in C

$$\boxed{c_p = \mathcal{F}(\Theta, x_p)}$$

$$\underset{\Theta \subseteq \Upsilon}{\operatorname{argmin}} \sum_{p=1}^{n} \|\mathcal{F}(\Theta, x_p) - c_p\|^2$$

x_p : **feature descriptor** extracted at pixel p

c_p : corresponding chrominance values

- YUV color space

❖ Feature Descriptor

Feature design is key to the success of the proposed colorization method

- Low-level feature
- Mid-level feature
- High level feature

$$\text{feature descriptor } x_p = \{x_p^L; x_p^M; x_p^H\}$$

❖ Low-level patch feature

- There exists too many pixels with **same luminance** but **different chrominance** in color images. → A pixel can't represent the luminance, but patch 7x7.
- Colorization에서 SIFT, DAISY 보다 성능 좋음

(a)Input (b)-patch feature (c)+patch feature

(b) 에서 모든 바다 색깔이 다 파란색인 문제점이 존재

❖ Mid-level DAISY feature

- More accurate discriminative description on **high-complex texture scenarios**.

(a)Target (b)-DAISY (c)+DAISY (d)Refined

→ Fully Automatic

[2010 TPAMI] DAISY : An Efficient Dense Descriptor Applied to Wide-Baseline Stereo

Dense keypoints, Wide-basedline stereo, Fast than SIFT and GLOH..

❖ High-level Semantic feature

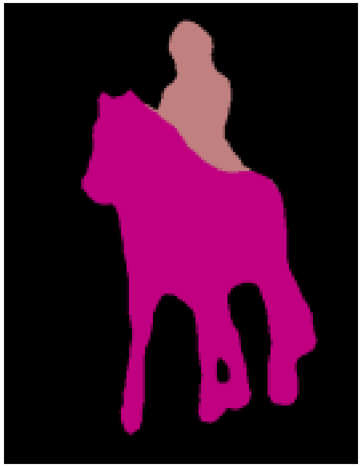
- Low & midle level indicate geometric structure of pixel.
- But Colorization is a **semantic-aware process**

→ Extract semantic feature at each pixel (e.g. Sky, sea, animal..)

- FCN 사용해서 각 픽셀 마다 category label 붙임.

(Regression model 의 한계 인 듯.. 사과는 무슨 색으로 칠함?)

FCN-8s



Image

❖ High-level Semantic feature

- N-dimension probability vector at each pixel.

- If a pixel detected a grass, only grass color values will be used.

→ Colorization Problem 문제를 단순화 했다고 주장!
(Regression의 한계…. ㅎ)

(a)Input (b)Patch+DAISY (c)+Sementic

❖ Global Features

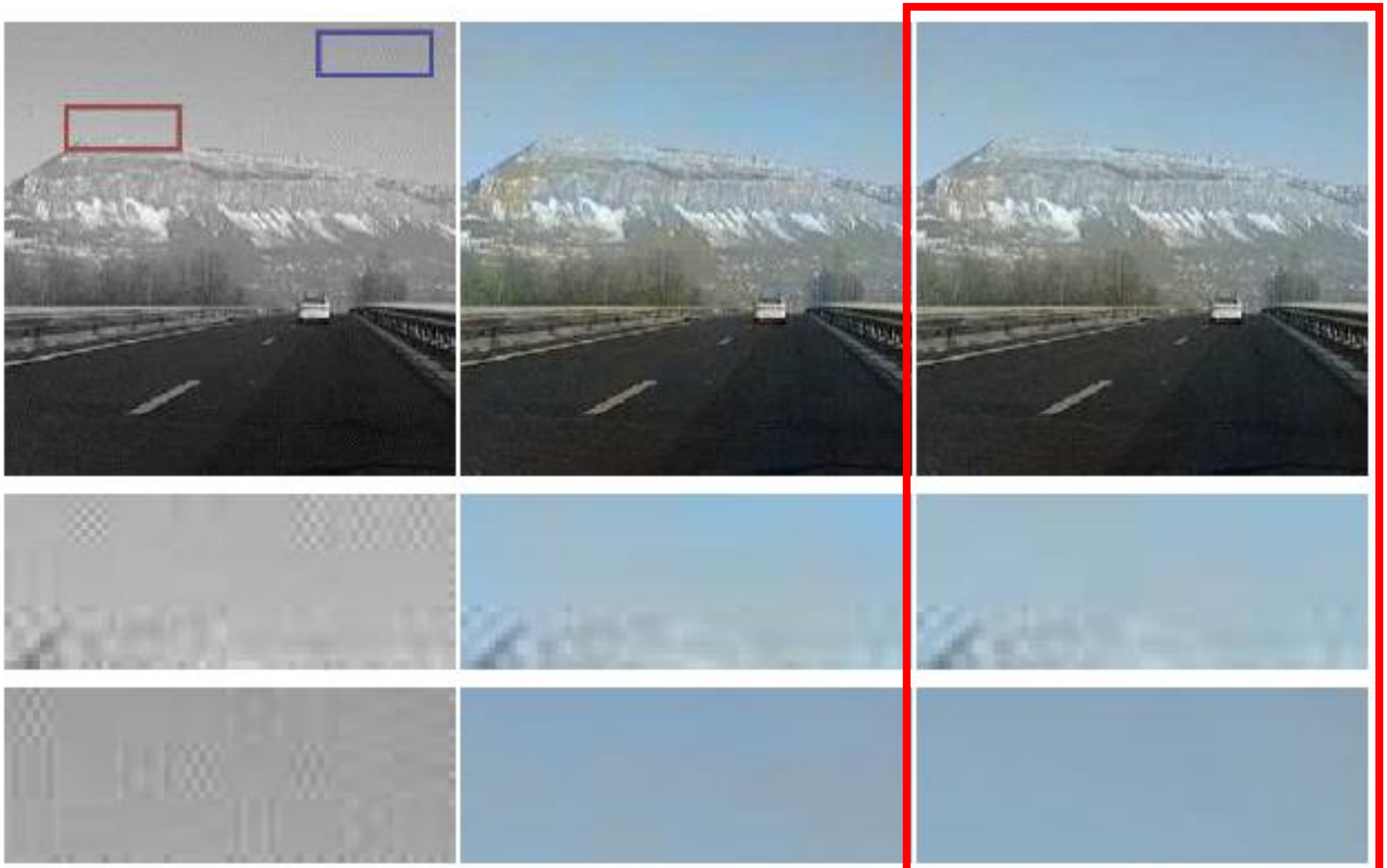
- Gist(gradient 를 나타내주는 Gabor filter)
- Histogram(color 의 분포도를 나타내주는 feature)

- Target image 를 globally similar but semantically different 하게 정하기 때문에 **안썼다.**

(a) Grayscale image (b) Global Feature X (c) Global Feature O

❖ Chrominance Refinement

- Artifact-free하게 하려고 bilateral filter 먹임



(a)Input (b)Before (c)After

❖ Experiment Result

33dB (b) our method | 22dB (c) Gupta et al.[6] | 21dB (d) Irony et al.[11] | 20dB (e) Welsh et al.[23] | 11dB (f) Charpiat et al.[1] | (a) input image | (g) reference image

Evaluation을 PSNR로 비교(얼마나 Ground Truth와 같은지) → Regression 한계..ㅎ

저자가 말하길, 초록색 박스는 reference 와 비슷하게 colorization 하는데

우리 꺼는 잘된다!!

What is Colorization Problem?

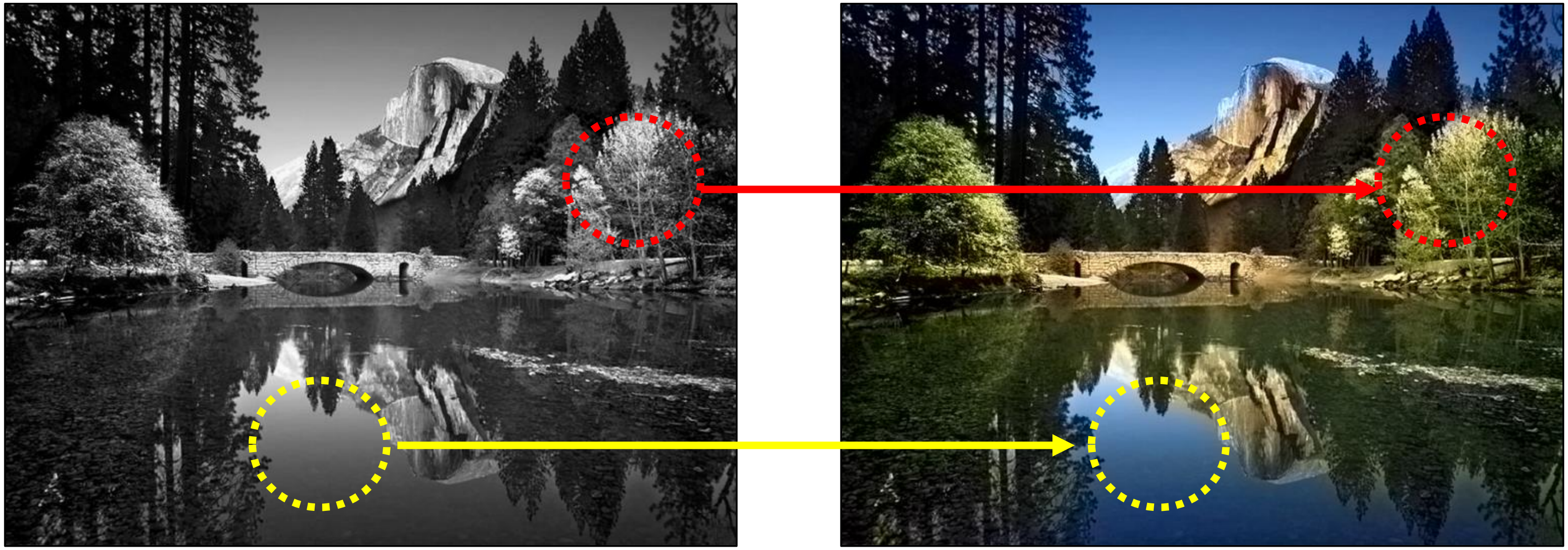
❖ PSNR 으로 성능평가를 해야되나?

- **꼭 Ground Truth 와 똑같이 색칠해야 하는 걸까?**

(나무를 초록색이 아닌 빨간색으로 칠해도 되는거 아님?)

- 그럴싸하게 Colorization 하는 것을 목표로 네트워크 설계!

(Not regression but Probabilistic!)



❖ Automatic Image Colorization via multimodal predictions [3] : param

[2008 ECCV] Automatic image colorization via multimodal predictions.

- color prediction is inherently multimodal – many objects can take on several plausible colorizations.
- For example, an apple is typically red, green, or yellow, but unlikely to be blue or orange. To appropriately model the multimodal nature of the problem, we predict a distribution of possible colors for each pixel

Colorful Image Colorization

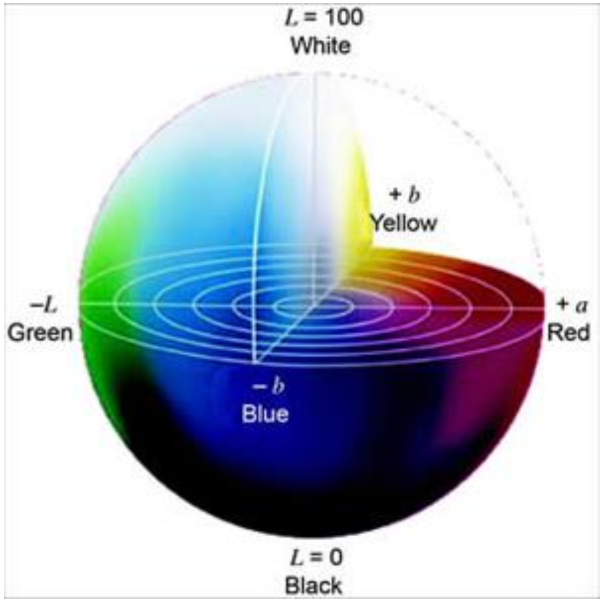
[ECCV 2016] Colorful Image colorization

❖ Purpose

- **Not** to recover the **actual** ground truth color, but rather than **Plausible** colorization that could potentially fool a human observer.
- Using **CIE Lab color space**
- Given the lightness Channel **L**, this network predicts the corresponding **a** and **b** channel.

◆ CIE Lab color space

- Since human **non-linearly** perceives the color, Lab color space has a **nonlinear** relationship between the wavelength of the actual light.
- The distances of two different colors in the Lab space are designed to be proportional to **the difference in color felt by humans**.



Introduction

❖ Contribution

- This paper make progress on the graphics problem of **automatic image colorization**
  - ✓ Appropriate Objective Function.
  - ✓ Novel Framework for testing colorization algorithms.

- **Self-supervised** representation learning

(Auto encoder 랑 느낌이 비슷)

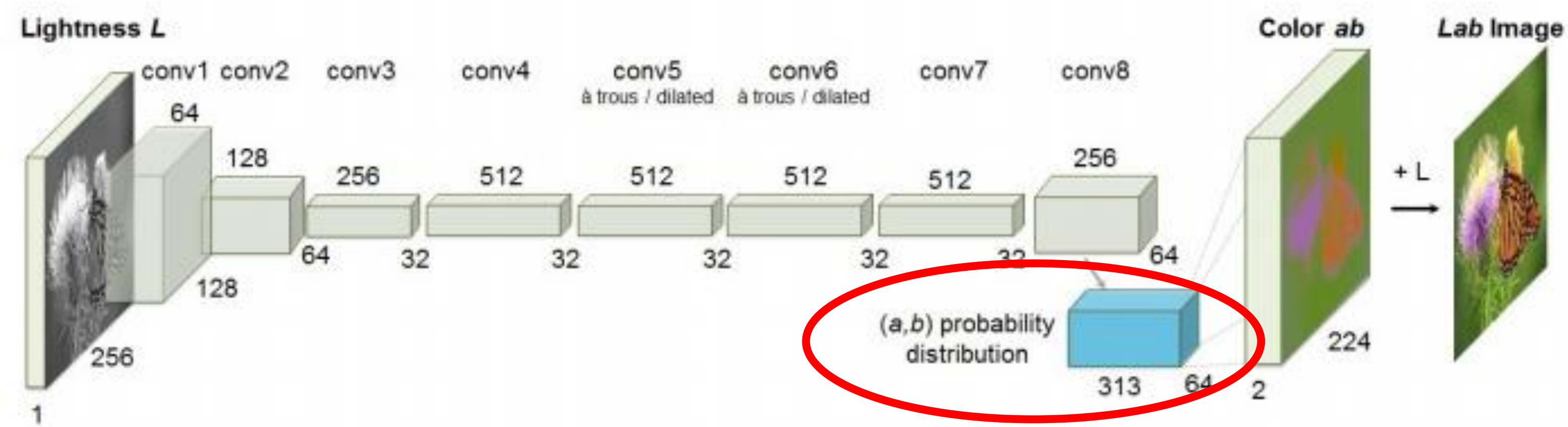
❖ Prior Work on colorization

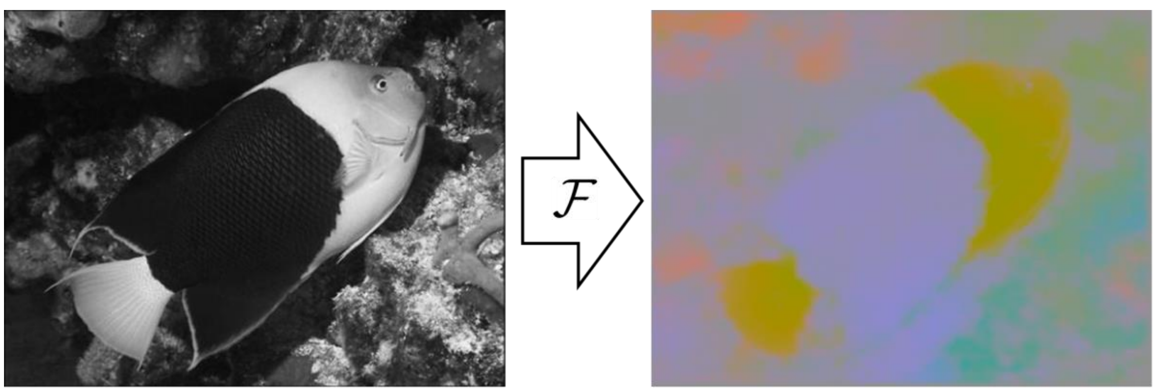
- Given Grayscale image, define one or more color reference images.

- Learn prediction as **regression** problem on continuous color space

Approach

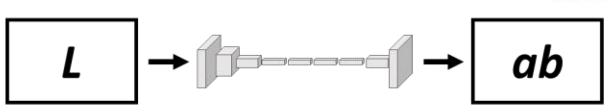
❖ OverView





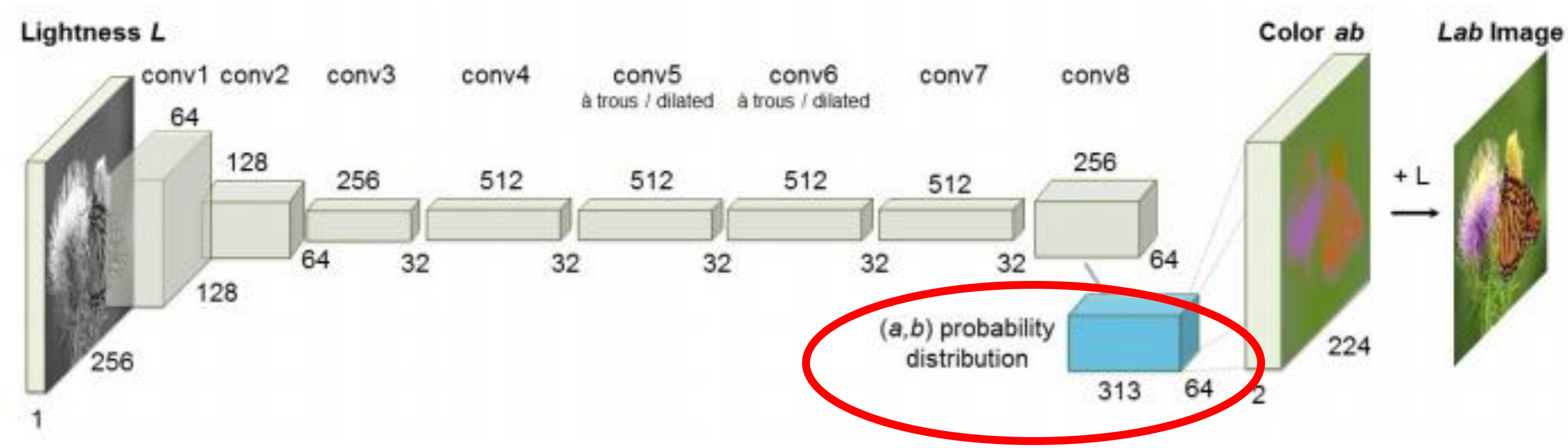
Grayscale image: *L* channel
$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$

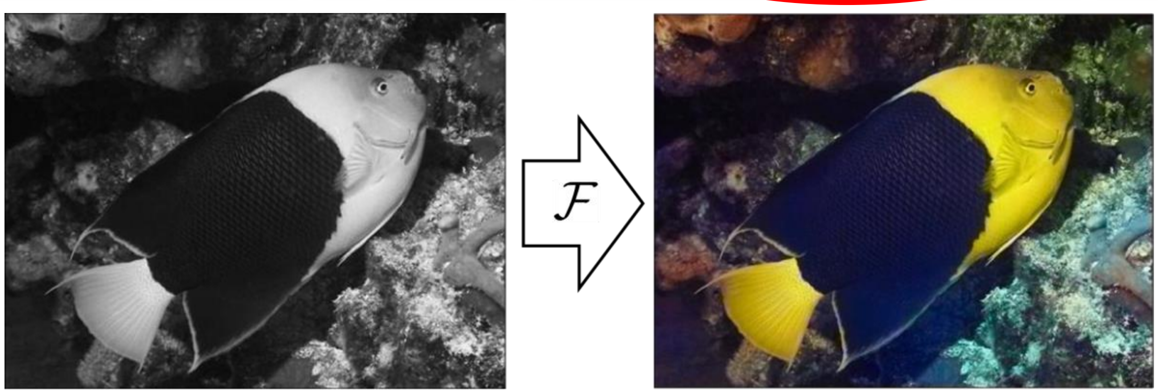
Color information: *ab* channels
$\widehat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$



Approach

❖ OverView



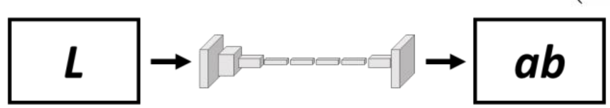


Grayscale image: *L* channel

$\mathbf{X} \in \mathbb{R}^{H\times W\times 1}$

Concatenate (L,ab)

$(\mathbf{X}, \widehat{\mathbf{Y}})$



❖ Objective Function

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

- Since objects can take on several plausible color, color prediction is inherently **multi-modal**. (ex. Apples may be red, green, or yellow, but orange can't be blue)
- To model this, the prediction is a **distribution** of possible color for each pixel.
- Not MSE(Mean Square Error)

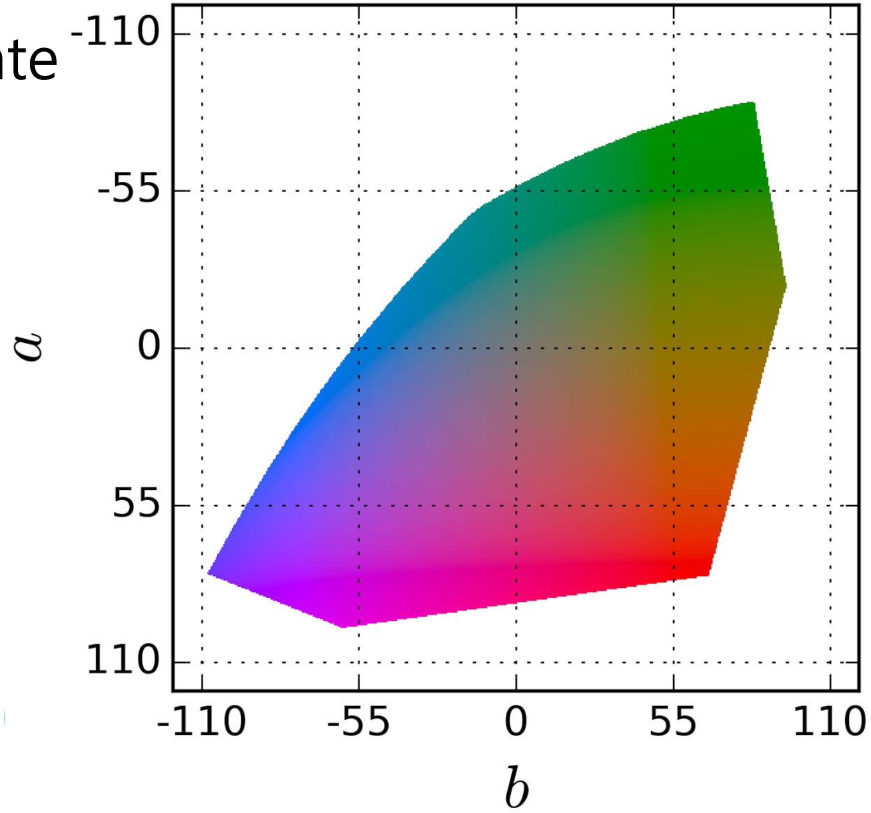
✓Averaging effect → grayish, de-saturated results.
✓One explanation is that [1,2] use loss functions that encourage conservative predictions.
✓These losses are inherited from standard regression problems, where the goal is to minimize Euclidean error between an estimate and the ground truth.

Better Loss Function

- Regression with L2 loss inadequate

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

Colors in *ab* space
(continuous)



Better Loss Function

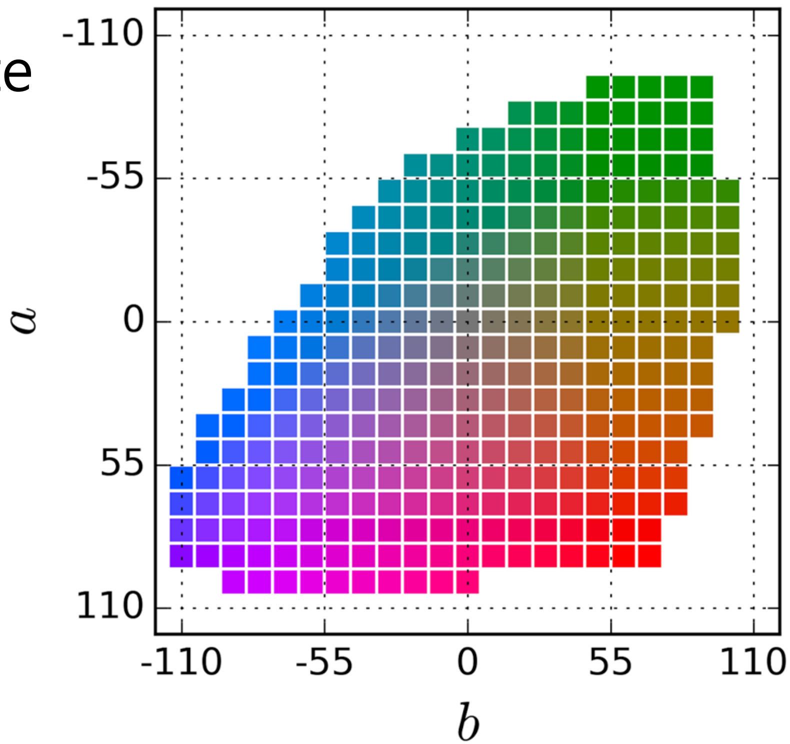
- Regression with L2 loss inadequate

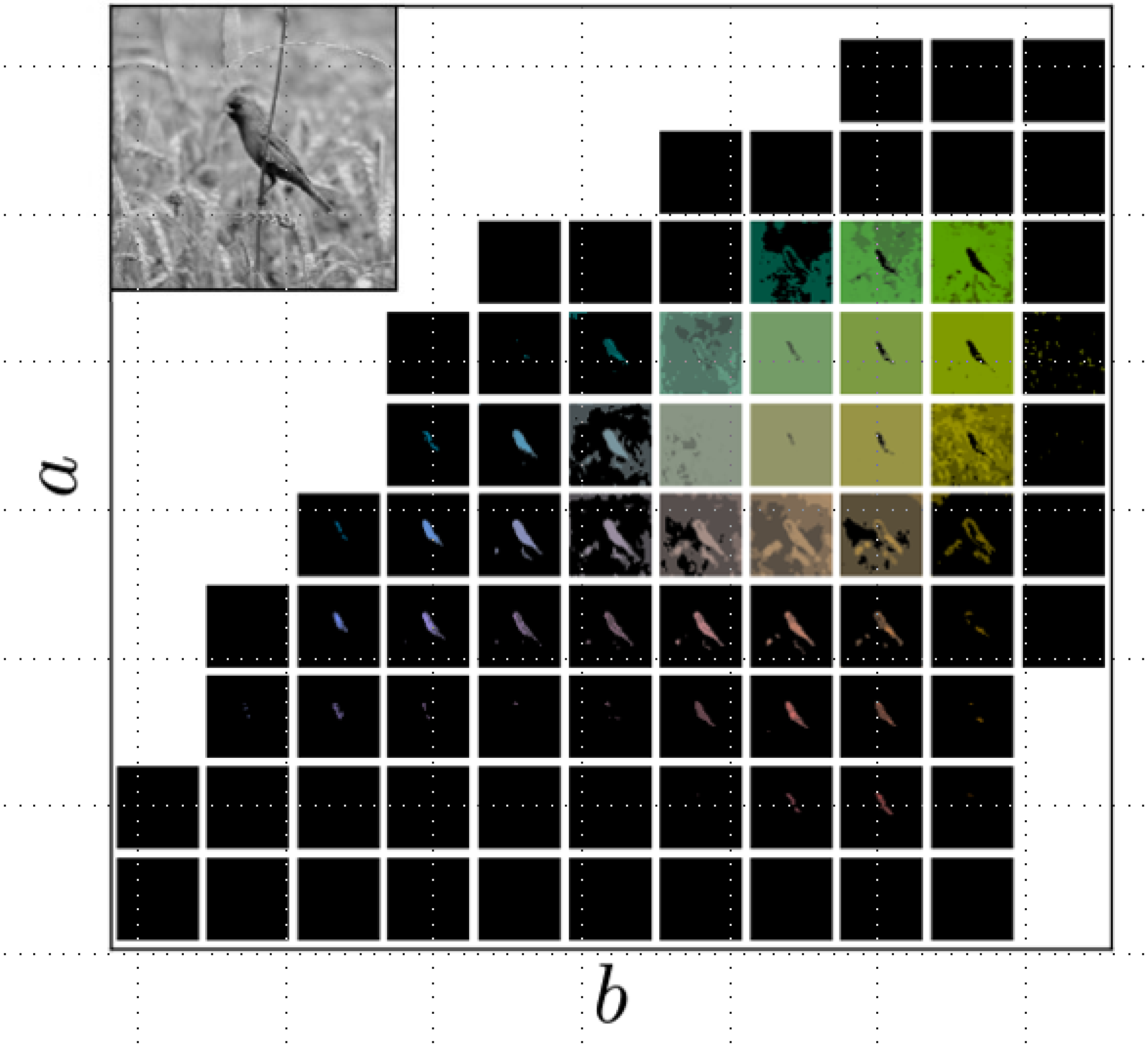
$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2} \sum_{h,w} \|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

- Use **multinomial classification**

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

Colors in *ab* space
(discrete)




a
b

Better Loss Function

- Regression with L2 loss inadequate

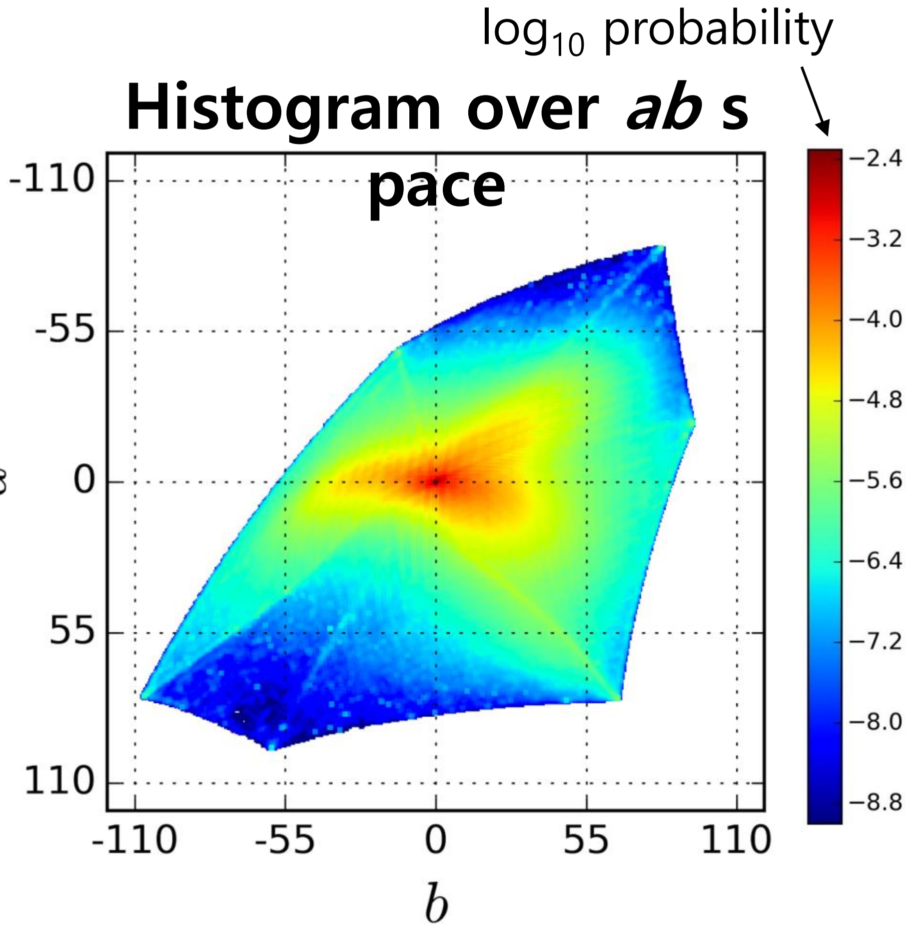
$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2}\sum_{h,w}\|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2$$

- Use **multinomial classification**

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

- **Class rebalancing** to encourage learning of *rare* colors

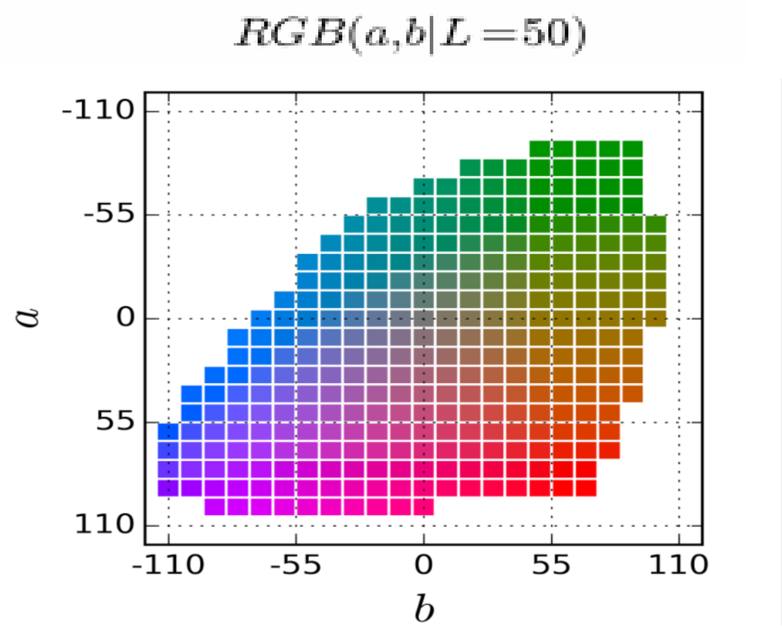
$$\mathrm{L}(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\frac{1}{HW}\sum_{h,w} v(\mathbf{Z}_{h,w})\sum_{q}\mathbf{Z}_{h,w,q}\log(\widehat{\mathbf{Z}}_{h,w,q})$$



❖ Objective Function

- ab output space is divided with grid(Q=313)
- The network **learns** a mapping(g) to a **probability distribution** over these Q color.

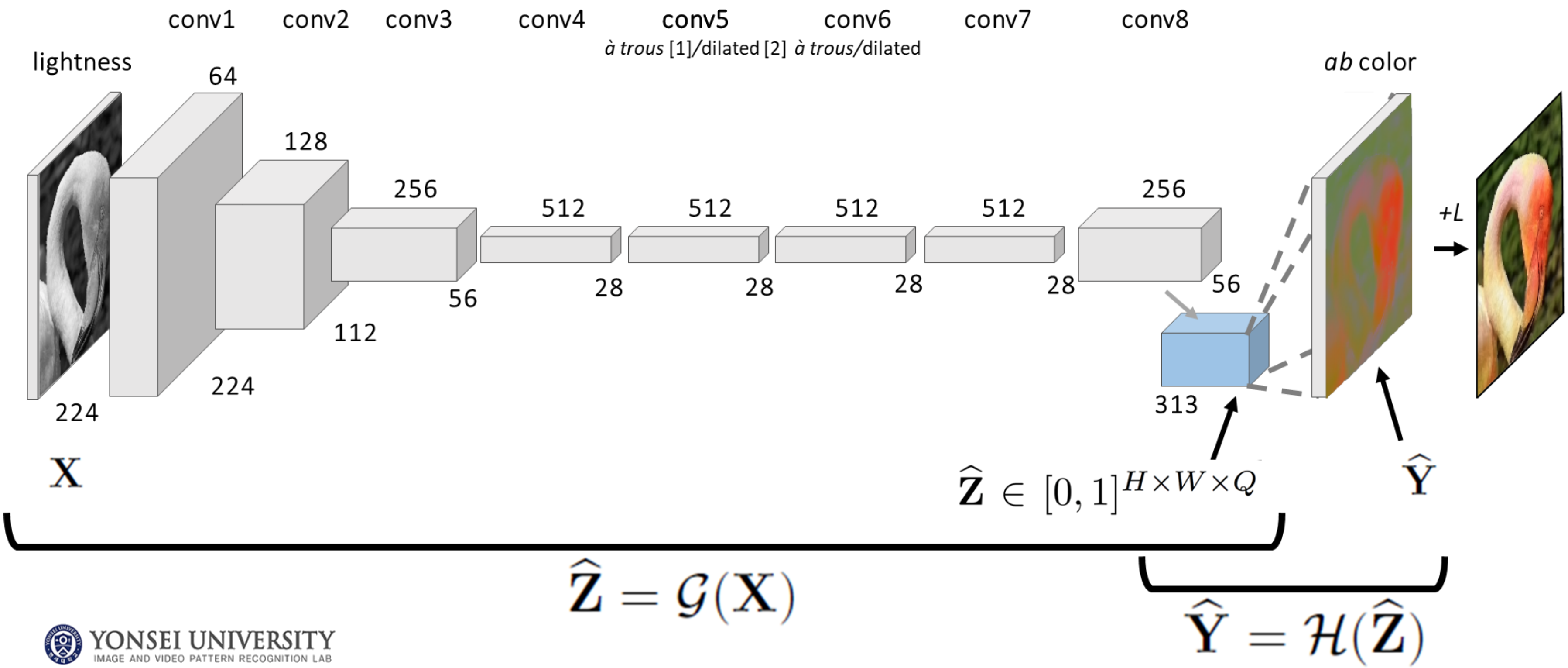
$$\widehat{\mathbf{Z}} = \mathcal{G}(\mathbf{X}) \qquad \widehat{\mathbf{Z}} \in [0,1]^{H \times W \times Q} \qquad \widehat{\mathbf{Y}} = \mathcal{H}(\widehat{\mathbf{Z}})$$



❖ Objective Function

- ab output space is divided with grid(Q=313)
- The network **learns** a mapping(g) to a **probability distribution** over these Q color.

$$\widehat{\mathbf{Z}} = \mathcal{G}(\mathbf{X}) \qquad \widehat{\mathbf{Z}} \in [0,1]^{H\times W\times Q} \qquad \widehat{\mathbf{Y}} = \mathcal{H}(\widehat{\mathbf{Z}})$$



Approach

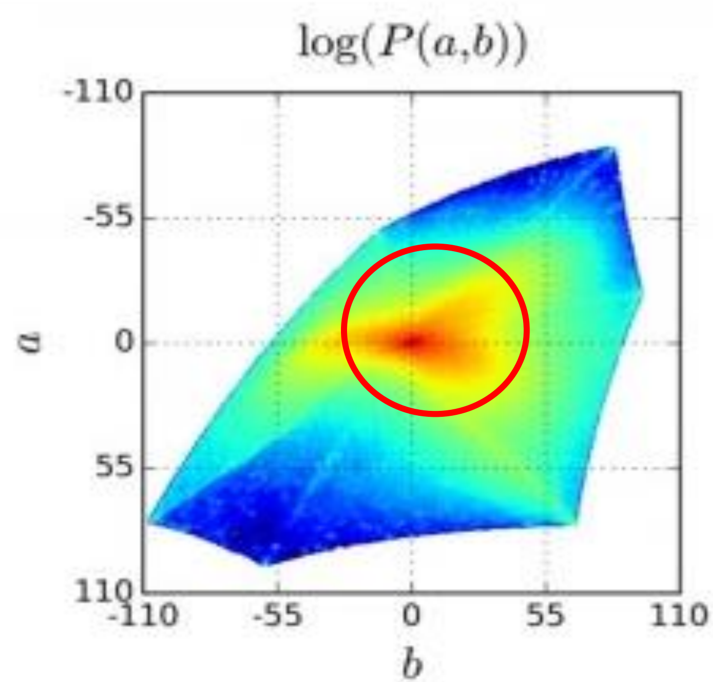
❖ Class rebalancing

- This term leads to vibrant distribution result.
- Comparing multinomial cross entropy loss by including weighting term

$$v(\mathbf{Z}_{h,w}) = \mathbf{w}_{q^*}, \quad \text{where } q^* = \arg\max_q \mathbf{Z}_{h,w,q}$$

$$\mathbf{w} \propto \left((1-\lambda)\widetilde{\mathbf{p}} + \frac{\lambda}{Q}\right)^{-1}, \quad \mathbb{E}[\mathbf{w}] = \sum_q \widetilde{\mathbf{p}}_q \mathbf{w}_q = 1$$

$$L_{cl}(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\sum_{h,w} v(\mathbf{Z}_{h,w}) \sum_q \mathbf{Z}_{h,w,q} \log(\widehat{\mathbf{Z}}_{h,w,q})$$



❖ Class Probabilities to Point Estimates

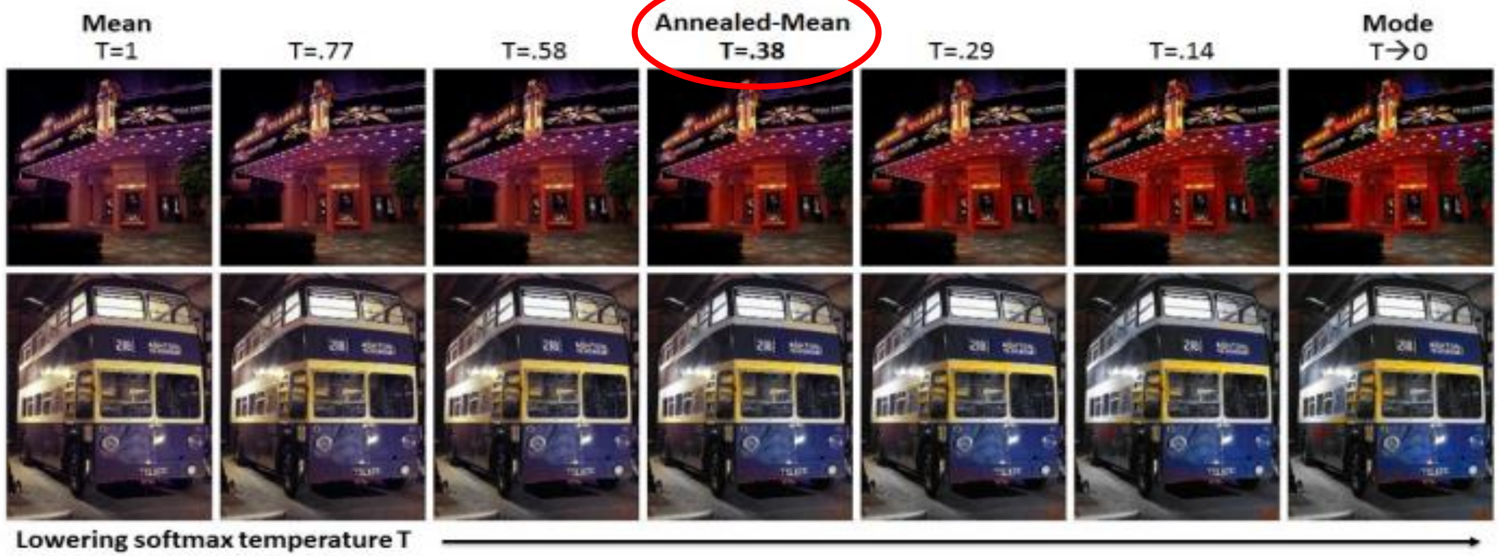
- **Re-adjusting** the temperature T of the soft max distribution

$$\mathcal{H}(\mathbf{Z}_{h,w}) = \mathbb{E}\big[f_T(\mathbf{Z}_{h,w})\big], \quad f_T(\mathbf{z}) = \frac{\exp(\log(\mathbf{z})/T)}{\sum_q \exp(\log(\mathbf{z}_q)/T)}$$

Choice is to take the mode of the predicted distribution for each pixel

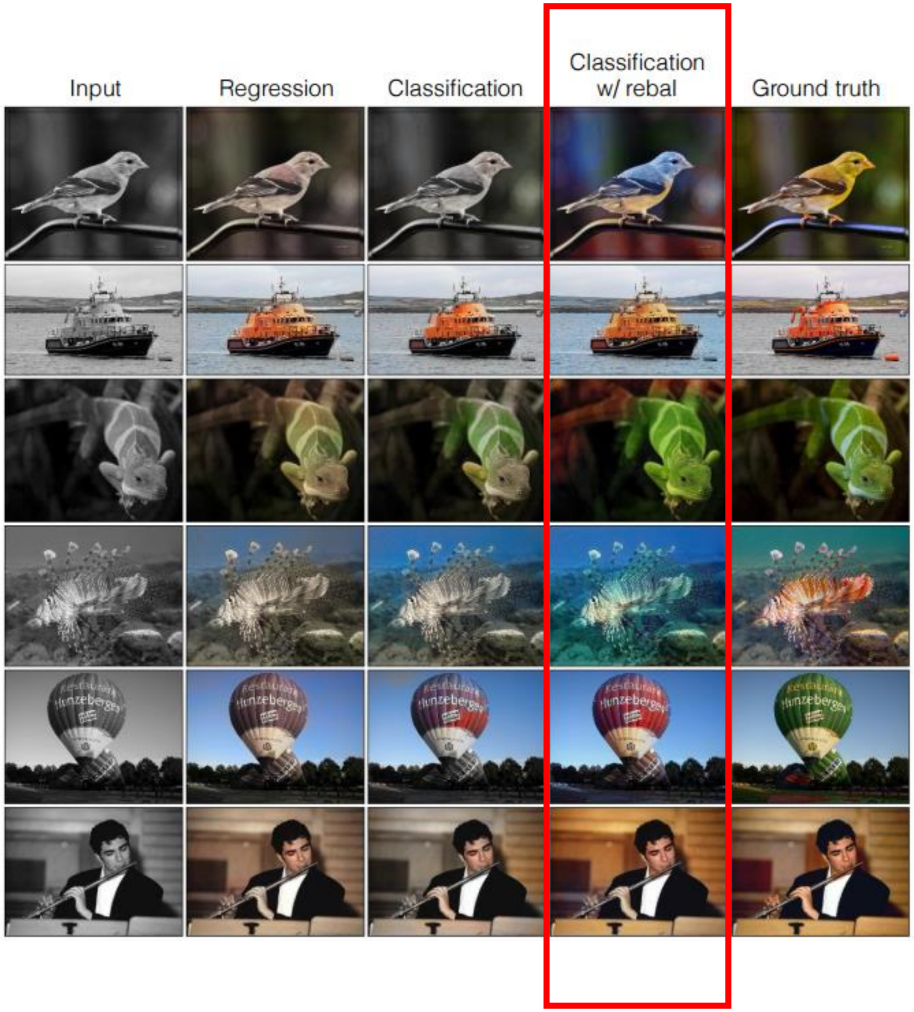
(T → 0 results in 1-hot encoding at the distribution mode)

→ **Vibrant** but **inconsistent** result

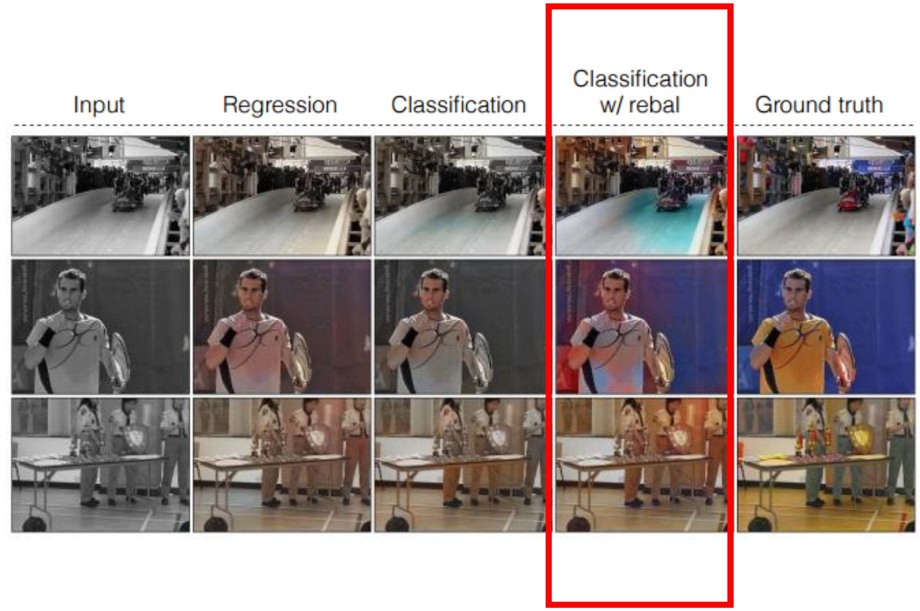


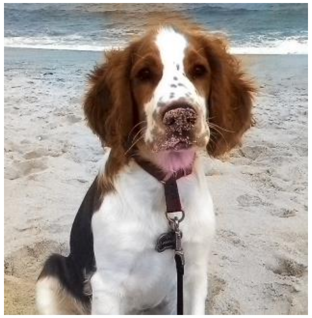
Paper Results

Success

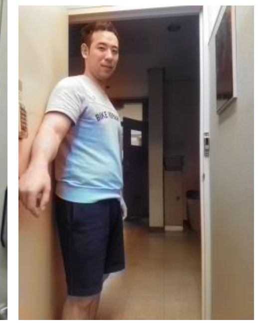


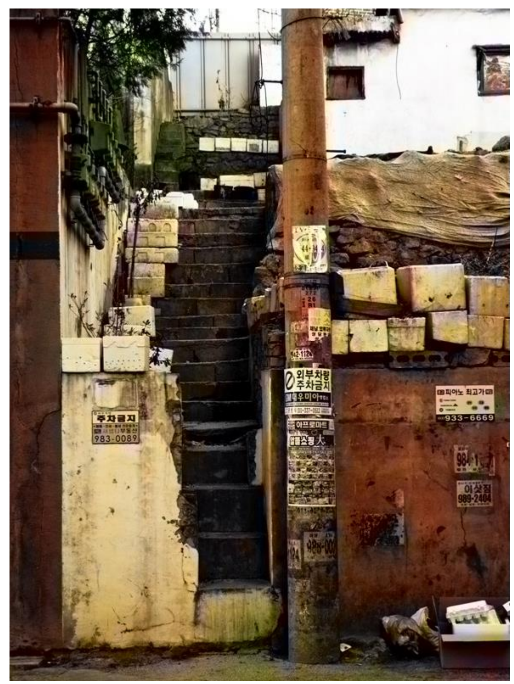
Failure

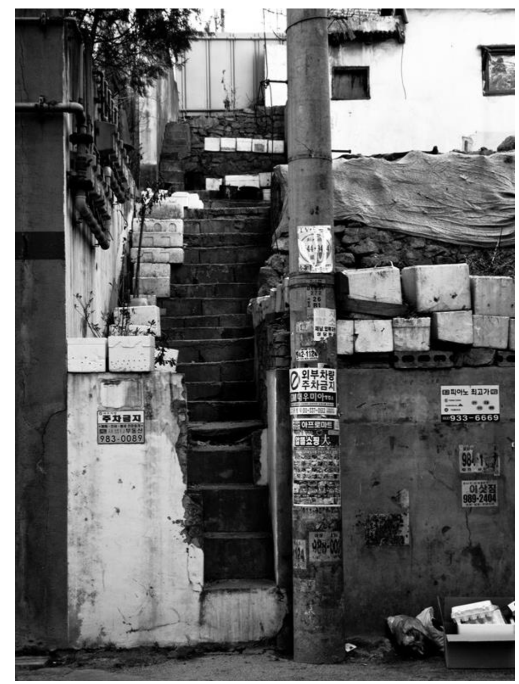




Our Result(Output / Input)







Discussion

- ❖ Future Work.
  - Classification? Segmentation?