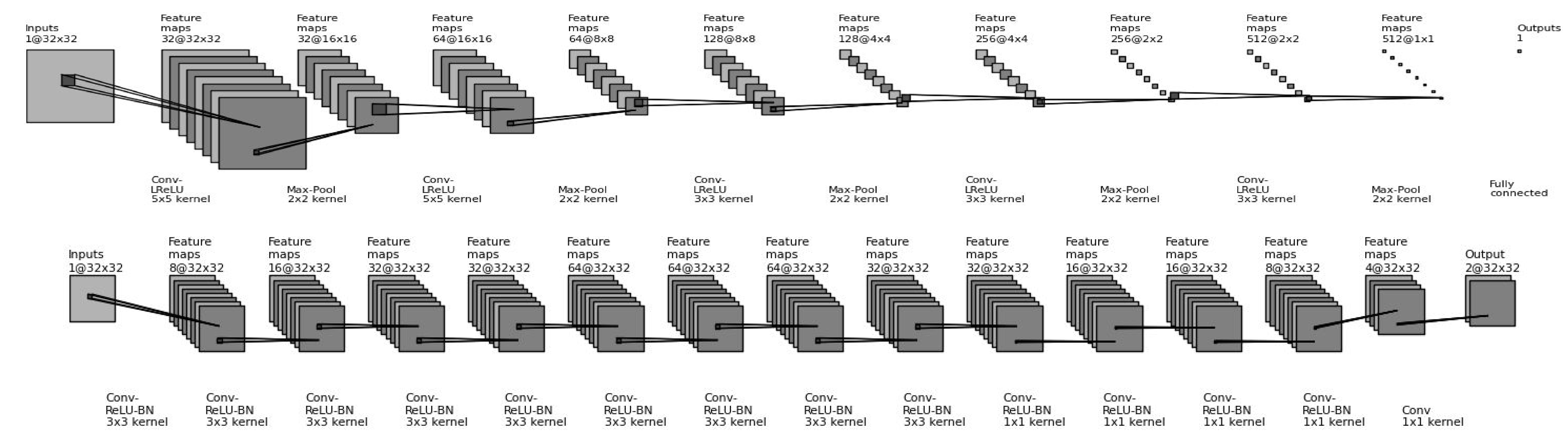
AutoColorization of Monochrome Images

Kushaagra Goyal, Yash Malviya, Bhuvneshwar
Department of Electrical Engineering, Stanford University

Introduction

- AutoColorization of gray images has powerful applications in image/video compression and info-mining from historic video/images.
- Inherently, it's a **multi-modal** problem. e.g. differently colored dresses in right are correct outputs for a gray scale version this image.
- Problem statement is not to produce the original images, but rather reasonably colored images that can fool a human observer.



Methodology

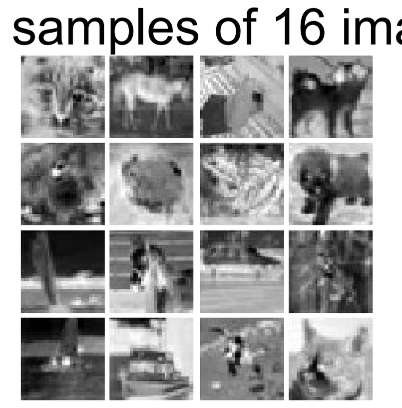
- Dataset -> CIFAR-10 -> 50,000 training and 10,000 validation images of size 32*32*3. Low resolution images are computationally cheaper to train and hence enables faster prototyping.
- Designed a CNN architecture which takes in grayscale image and generates a coloured version of the input image.
  a. **Output Space :** Explored two options for the output space.
     - **RGB images** -> Regenerate all three components (RGB) as output of the model using the input grayscale image.
     - **UV component** -> Generate the UV components from the luminance (Y) and concatenate YUV components to generate the final image.
  b. **Loss Functions** :
     - **Regression Losses:** Tried *L2, L1* and *Huber* loss.
       Huber loss is a combination of L1 and L2 loss, eq [1]
     - **Classification Loss** : Modelled colorization as a classification problem where we predict each pixel's class. Class values are picked from a discretized UV space.
     - **Generative Adversarial Networks**: Finally we also tried employing a GAN network for colorization.
- **Huber loss** is calculated as follows, (delta=0.5, 1.0). Huber loss for delta = 1.0 is **smooth L1 loss**

$$L_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \le \delta, \\ \delta(|a| - \frac{1}{2}\delta), & \text{otherwise.} \end{cases} \quad (1)$$

- To train the GAN network we perform gradient ascent on the generator network and discriminator network, their losses are as-
  - Generator loss $= -\mathbb{E}_{z\sim p(z)}[\log D(G(z))]$
  - Discriminator loss $= -\mathbb{E}_{x\sim p_{data}}[\log D(x)] - \mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))]$

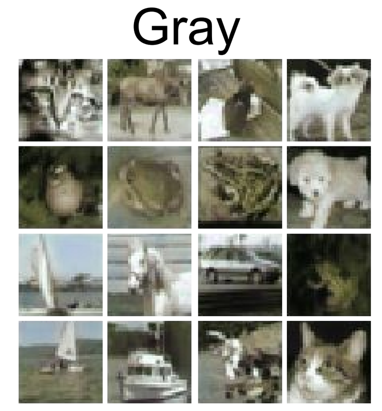
Results

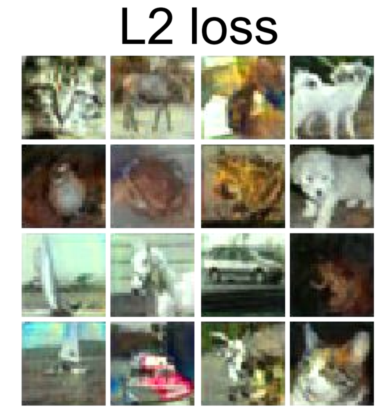
The following figures show the different results generated by our models for two samples of 16 images each in the test set


Gray

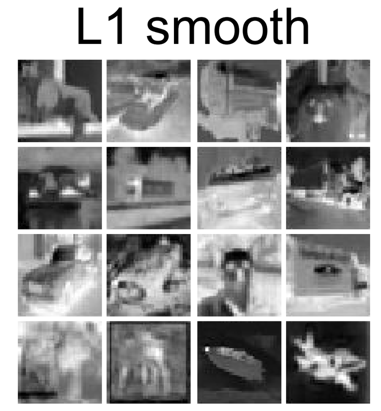

L2 loss


L1 loss


L1 smooth

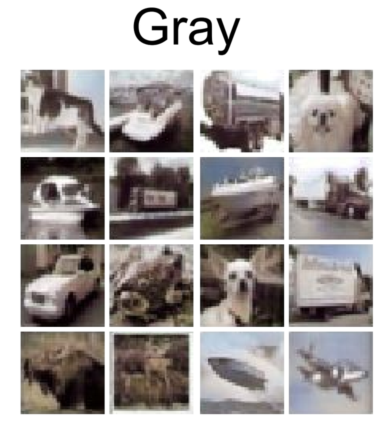

GAN


Ground Truth

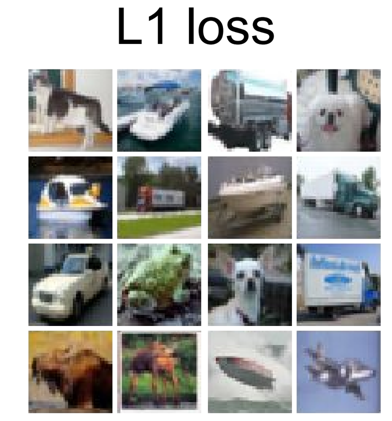

Gray


L2 loss


L1 loss


L1 smooth


GAN


Ground Truth

Model	AUC (%)	Evaluation Test (%)
Grayscale	80.33	22.19
L2 Loss	98.37	67.75
GAN	97.26	61.24
Ground Truth	100	77.76

- Area under the curve (**AUC**) measures AUC of the cumulative error distribution in RGB space as we sweep across different thresholds.
- **Evaluation Test** measures classification accuracy of generated images on a pre-trained model. For Grayscale, UV component were set to 0.

Conclusions

- Models trained with L1, L2 and Huber/L1 smooth loss give similar results for most images
- Images generated using L2 loss are generally sharp and crisp However the colors are desaturated.
- To avoid this conservative estimate we are trying to train a classification model inspired from [1]
- We trained a GAN model; training a GAN was a challenging involving careful adjustment of hyperparameters to avoid unstable learning.
- GANs generate more colorful images as compared to other models, but they are less sharper and have artifacts too.
- In future work, we plan to train L2 loss model along with Gaussian blurring of input image.
- Also, we plan to tune our classification based colorization model and experiment with LAB color space for our model output.

1. **Colorful Image Colorization**, Richard Zhang et al, ECCV 2016