Computer Vision (67542) – Final Project

Image Auto-Colorization

Or Tal

Introduction:

In this project I have implemented an image auto-colorization model based on the work "Let there be Color!", Iizuka et al. [2016]. In this paper I will first introduce the concept of image auto-colorization and the main changes in Iizuka's approach that differ his purposed method from previous work at the time. Second, I will describe the general architecture chosen for this implementation and focus on concepts we have encountered in this course. Last, I will present some of the results received by my implementation.

Image Auto-Colorization: In their work, Iizuka et al. mention several previous works done before their paper, claiming that traditional colorization requires significant user interaction in one form or another. One example would be "Colorization using Optimization", Levin et al. [2004]. In this work, Levin et al. offer an approach based on a simple principle: neighboring pixels that have similar intensities should have similar colors. In their method they chose to work with YUV color space, where Y (luminance) is passed as input, aside to "colored scribbles". By their assumption, neighboring pixels with similar intensities should have similar colors, hence, in a wide sense we would expect different colors around edges, and similar colors in "smoother" areas. Following this assumption, when passing "colored scribbles" on areas of the image (mask) we give the algorithm a concept of the actual colors in that region and it does the completion. They formalize their approach using a quadratic cost function, obtain an optimization problem and use standard methods to solve it, hence the name of the paper. When dealing with auto-colorization, approaches that requires user interaction are then of course not automated. In their approach, Iizuka et al. purpose to leverage semantic information from images to anticipate the main color of a region. Following the principle showed in Levin's work, this would then replace the needed human interaction for the colorization. In their approach they offer an architecture that jointly extracts global and local features from an image and then fuse them together to perform the colorization.

Purposed method:

The purposed method is based on the work done by Krizhevsky et al. [2012], (Alex net), where it was proven that deep convolutional neural networks can learn complex mapping from large amounts of training data. The purposed architecture is formed by several sub-complements that form a directed acyclic graph and could be roughly divided into two main parts: classification and colorization. By training a classification model, they allow weights-share of global feature detection layers, and feed forward of local feature detection layers. Both outputs are fused (tiled and stacked) and fed as an input to the colorization network. As seen in figure 1, in a wide sense, the model is a type of auto-encoder, having feature extraction forced by the classification error in the encoding phase and colorization on the decoding phase (up-scaling).

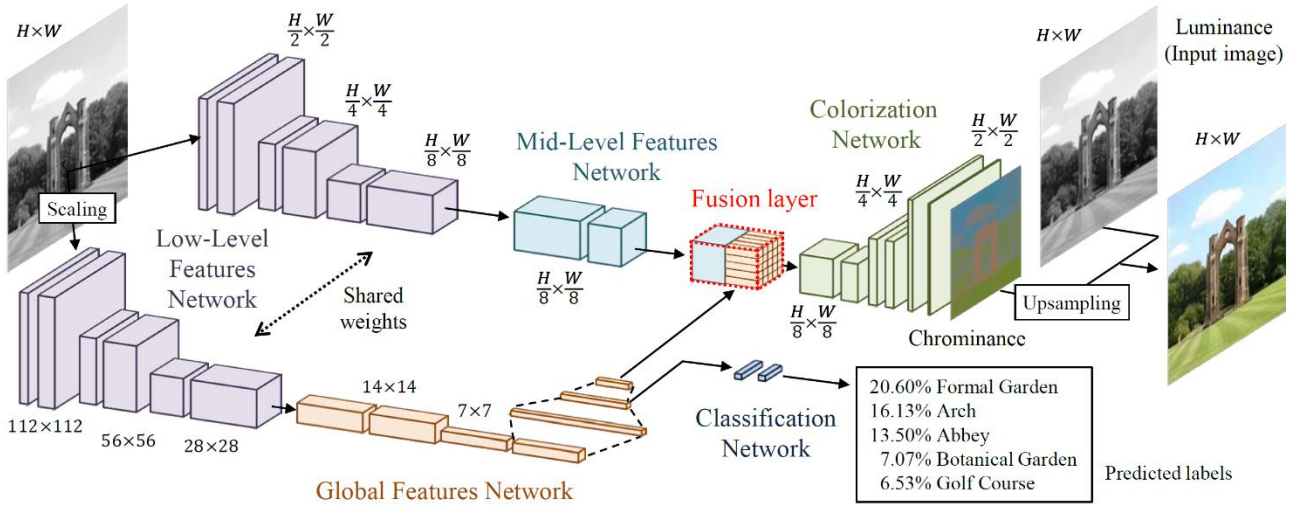

Figure 1: overview of the purposed model

(a) Low-Level Features network

Type	Kernel	Stride	Outputs
conv.	3 × 3	2 × 2	64
conv.	3 × 3	1 × 1	128
conv.	3 × 3	2 × 2	128
conv.	3 × 3	1 × 1	256
conv.	3 × 3	2 × 2	256
conv.	3 × 3	1 × 1	512

(b) Global Features network

Type	Kernel	Stride	Outputs
conv.	3 × 3	2 × 2	512
conv.	3 × 3	1 × 1	512
conv.	3 × 3	2 × 2	512
conv.	3 × 3	1 × 1	512
FC	-	-	1024
FC	-	-	512
FC	-	-	256

(c) Mid-Level features network

Type	Kernel	Stride	Outputs
conv.	3 × 3	1 × 1	512
conv.	3 × 3	1 × 1	256

(d) Colorization network

Type	Kernel	Stride	Outputs
fusion	-	-	256
conv.	3 × 3	1 × 1	128
upsample	-	-	128
conv.	3 × 3	1 × 1	64
conv.	3 × 3	1 × 1	64
upsample	-	-	64
conv.	3 × 3	1 × 1	32
output	3 × 3	1 × 1	2

Table 1: layer specification

Integration of topics learned in course:

- The purposed method works in 'Lab' color space, taking a grayscale image (luminance) as input and regenerate the color channels. As seen in class, this color space is designed to approximate human vision, making it a better contestant than previously used spaces such as YUV.
- Batch normalization is done before each activation layer by keeping a running mean and standard deviation of the input to the transfer function, so it is roughly mean centered with a standard deviation of one.
- Loss evaluation – MSE is being used to evaluate the difference between the ground truth colored image and the colored output of the model. As for the classification, a cross entropy loss is being used. Therefore, given a constant factor α, the overall loss would be calculated as:

$$L(y_{class}, y_{color}) = MSE(y_{colorg.t}, y_{color}) - \alpha \cdot CE(y_{class\ g.t}, y_{class})$$

As seen in class, the cross-entropy loss is commonly used for classification, where relaying on having multiple examples of different scenes.

- Weights share and auto-encoders – auto-encoder is the general architecture being used for colorization in this model. As taught in class, this typical architecture had shown good performance in feature extraction and image regeneration, and as the encoder part is forced for feature extraction (in concept) as it shares its weights with a portion of the classification network, the colorization takes place in some of the mid feature layers and mainly in the decoding part. This principle was the key insight that differs the results significantly from state-of-the-art results achieved at the time by Cheng et al. [2015].

Results:

The model was trained on ~2,000,000 examples, using a batch size of 26 images (for memory related issues) and 10 epochs of 75,000 steps per epoch. The model does plausible results for regenerating green-yellow-brown colors yet seems to fail predicting other colors. having more epochs still shows improvement from one epoch to the other hence it seems that with more training the model could still improve.

Summary:

This work was the first time I have implemented a neural network from scratch, it was a challenging task to do and I have learned a lot in the process. There were some technical difficulties such as computing, and damaged input dataset that influenced on the result quality, and I assume that with more time, and more computing power this model could achieve better results.

References:

CHENG, Zezhou; YANG, Qingxiong; SHENG, Bin. Deep colorization. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015. p. 415-423.

IIZUKA, Satoshi, SIMO-SERRA, Edgar and ISHIKAWA, Hiroshi, 2016. 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), pp.1-11.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. 2012. p. 1097-1105.

LEVIN, Anat, LISCHINSKI, Dani and WEISS, Yair, 2004. Colorization using optimization. *In ACM SIGGRAPH 2004 Papers,* pp. 689-694.