Automatic Colorization of Black and White Photos

Daniel Ciugurean, Technical University of Cluj-Napoca

INTRODUCTION

Motivation

There is a large community dedicated to colorizing old black and white photographs. It provides a whole new perspective on the documented past. However, such an activity is time-consuming, ranging from being fully manual to partially automatic. The given constraint, which is to automatically colorize the input image, explains the overall reliance on convolutional neural networks in recent literature.

Problem Statement

- **Input**: A grayscale image
- **Output**: A plausible colorization of the input luminosity signal
- Apply a pretrained CNN model to extract the semantic information from the input image
- **Regression**: feed the accumulated low, mid and high level features to a ResNet-like structure and minimize the L2 distance between the generated colours and the ground truth.
- **Classification**: feed the accumulated low, mid and high level features to a ResNet-like structure and learn the per-pixel labels using a cross-entropy loss function.

APPROACH

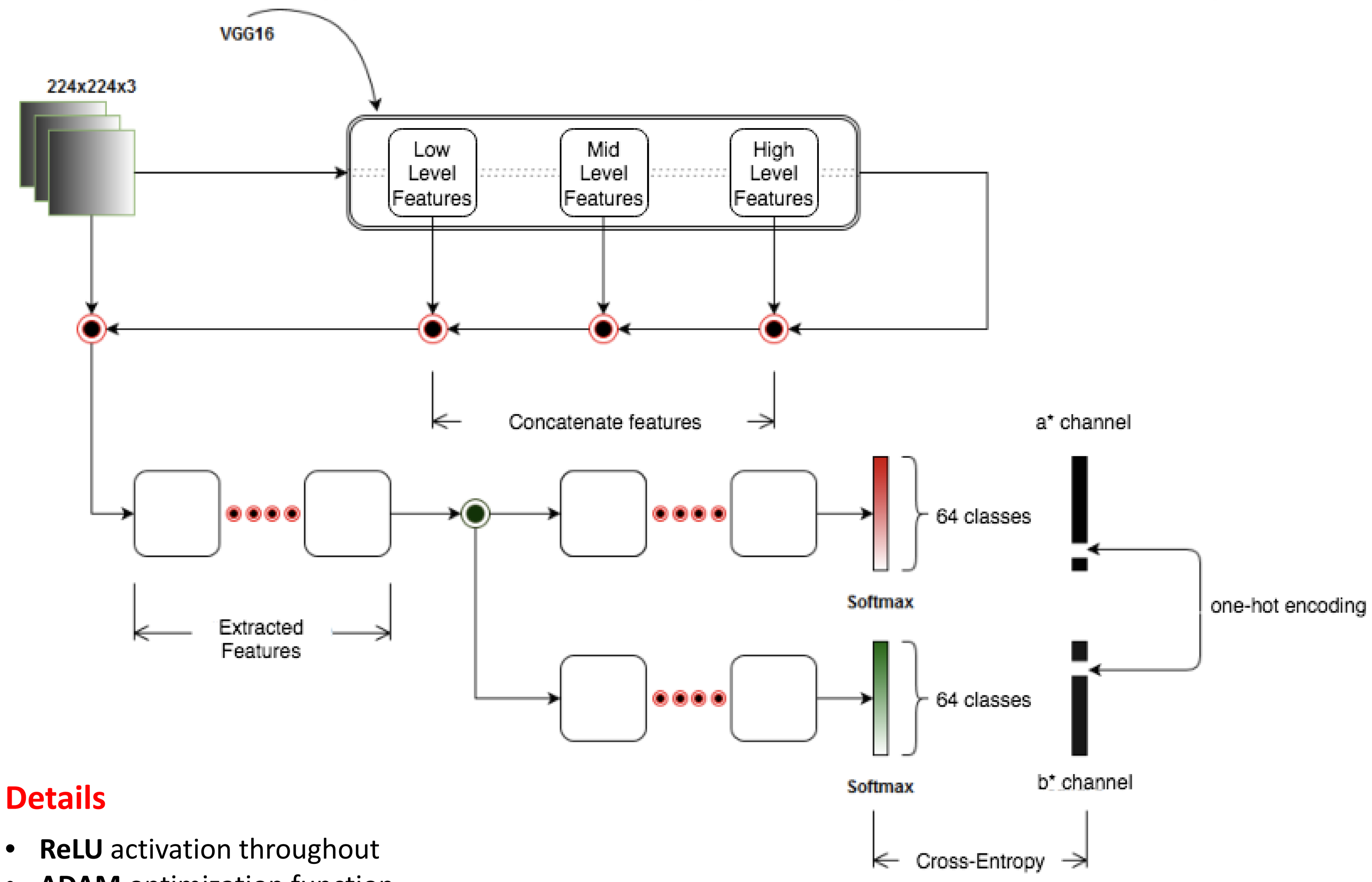
We first apply a colour space transformation: sRGB → LAB. The LAB space is much closer to the way humans perceive colour.

Figure 1 From left to right: the original grayscale image(the luminosity channel), the generated a* channel, the generated b* channel and the final result – the combination of the first three channels.

It also separates the image contrast from the colour information, which allows us to reduce the dimensionality of the problem. Thus, given the luminosity signal of the image(**L**) the model generates the other two channels(**a*** and **b***), and then concatenates the three to produce the final result, which is then upscaled back to the original size. The architecture uses a pre-trained **VGG16**(with locked parameters during training)[5], and uses skip-connections to concatenate low, mid and high level semantic information. These features are then further processed, in order to associate them with their corresponding colours. New, appended layers use residual connections, similar to those found in ResNet[6].

Architecture

Figure 2 The classification-based model tries to segment the image into 64 classes, per pixel, per output channel. The resulting colours are much more vibrant, and the averaging problem found in the first model is gone, but the resulting model proved difficult to converge.



Details

- **ReLU** activation throughout
- **ADAM** optimization function
- Heavy use of **Batch Normalization**
- Heavy use of **ResNet-like skip connections**
- **Classification**: Choose the index of the colour class with the highest probability
- Use **bilinear interpolation** to upscale the output image to the original dimensions

Implementation

- Keras + Tensorflow backend, Python 3.5, GNU/Linux 4.8+
- Numpy
- Skimage for image transformations
- Trainable parameters: 1,158,466
- Non-trainable parameters: 7,637,056

EXPERIMENTS

Train

- Subset of MSCOCO[3], totaling 70k images
- Batch size of 8 images
- 24 hours on a free Floydhub GPU instance(K80 11 GB)
- 6 hours/epoch

Test

- 10k images for validation
- Overfit on a few images to test robustness
- PSNR metric

RESULTS

Visual Results

Figure 3 Left – the original grayscale image, generated colorization by the regression based model and the generated colorization by the classification based model. Right – the original grayscale image, result presented by Iizuka et al. [4], our result.

Figure 4 From left to right: the original grayscale image, result presented by Zhang et al. [1], our result.

Evaluation

- To evaluate our results, we use the **PSNR(Peak Signal to Noise Ratio**) metric, as a quantitative measure of performance.
- We used over 100, randomly chosen, coloured images, which were then re-coloured using our system.
- Excellent quality: PSNR ∈ [30dB, 50dB]
- Acceptable quality: PSNR ≈ 25dB

$$EPM = \frac{1}{N \times M} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (X_{i,j} - Y_{i,j})^2$$

$$PSNR = 10 \log_{10} \frac{L_Y^2}{MSE}$$

$$L_Y \leftarrow 1.0$$

$$Result = 34dB$$

CONCLUSION

We were able to demonstrate two successful models in the task of automatic colorization of grayscale images: a regression based model, and a classification based model, which tries to assign a colour class to each individual pixel. The latter model was tweaked in order to overcome the averaging problem, which was present in the first version. As such, the model is trying to attain a plausible colour distribution for each pixel, in the context of a detected object. However, the higher the freedom space of the colour class, the harder it is to train the model.

FUTURE WORK

- Experiment with a larger dataset, e.g. ImageNet
- Use Generative Adversarial Networks(GANs) to solve the averaging problem as well as the unnaturally sharp colour transitions of the classification based model

REFERENCES

[1] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization" *ECCV*, 2016.
[2] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift" *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167
[3] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dolľ ar, and C. L. Zitnick, "Microsoft COCO: common objects in context" *CoRR*, vol. abs/1405.0312, 2014
[4] S. Iizuka, E. Simo-Serra, and H. 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 (Proc. of SIGGRAPH2016)*, vol. 35, no. 4, pp. 110:1–110:11, 2016.
[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition" *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556
[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition" *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385