INFRARED IMAGE COLORIZATION USING A S-SHAPE NETWORK

Ziyue Dong[+], *Sei-ichiro Kamata*[+], *Toby P.Breckon*[*]

[+]Graduate School of Information, Production and Systems, Waseda University
[*]Engineering and Computing Science, Durham University

ABSTRACT

This paper proposes a novel approach for colorizing near infrared (NIR) images using a S-shape network (SNet). The proposed approach is based on the usage of an encoder-decoder architecture followed with a secondary assistant network. The encoder-decoder consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. The assistant network is a shallow encoder-decoder to enhance the edge and improve the output, which can be trained end-to-end from a few image examples. The trained model does not require any user guidance or a reference image database. Furthermore, our architecture will preserve clear edges within NIR images. Our overall architecture is trained and evaluated on a real-world dataset containing a significant amount of road scene images. This dataset was captured by a NIR camera and a corresponding RGB camera to facilitate side-by-side comparison. In the experiments, we demonstrate that our SNet works well, and outperforms contemporary state-of-the-art approaches.

Index Terms— Infrared, Colorization, S-shape network, Convolutional neural network

1. INTRODUCTION

In recent years, image acquisition devices have largely expanded and sensor technology is increasing. For example, to improve the safety of driving at night, advanced driver assistance systems have become more popular, which use camera sensors for object detection and driver alerting. At night, near-infrared (NIR) cameras can get more information than regular RGB (color) cameras and human vision (e.g. pedestrian, animals, road and roadside information). As a result, NIR images can segment images according to the material of the object, which means that light reflection in the NIR spectral band depends on the material. As such, NIR light can be used to illuminate the scene in low light conditions. However, NIR light is out of the range of human visual perception and lacks color discrimination, making it difficult for the user to understand. As a result, conversion from nocturnal, illuminated NIR images to natural looking RGB images has several applications in the user application and visualization aspects of NIR sensing solutions.

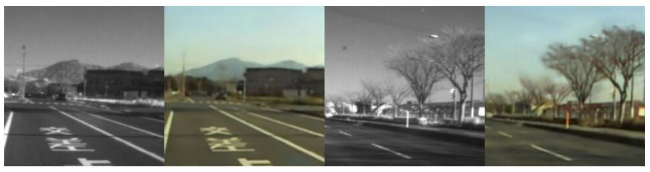


Fig. 1. Colorization results by our SNet.

Converting a grayscale NIR image into a multi-channel RGB is closely related to *Image Colorization*, where regular grayscale images are colorized. Although they have some particularities, their techniques are not suitable to colorize NIR images. For image colorization, the input grayscale images are used as luminance and only chrominance needs to be estimated, so the resulting output is sharp without blurring of scene detail [1]. However, the NIR images cannot be used as color luminance directly because they measure material dependant NIR reflectance. Subsequently, the results of colorizing NIR images are often blurry and lack high frequency scene detail [2].

This paper proposes a novel method based on S-shape network (SNet) to transfer a NIR image to RGB image automatically, which can not only colorize the NIR images but also retain the NIR images texture (Fig. 1). The skip connected encoder-decoder mainly generates the RGB outputs while the shallow encoder-decoder network, which has a 'loss function' between the outputs and ground truth, is used to enhance edges in the RGB outputs and stabilize the textureless region. In summary, this paper makes the following contributions: (1) We construct a dataset of dual infrared and RGB color image pairs (1978 pairs) and perform feature-based registration; (2) We propose a novel end-to-end neural network with encoder-decoder architecture with a skip connection between the encoder and decoder layers. Although previous work has considered similar network structures [3][4], we uniquely add an edge-preserving assistant network to perform NIR images colorization.

2. RELATED WORK

For *Colorization*, traditional approaches [5][6][7] require user interactions, such as user strokes (scribbles). In addition, example-based colorization techniques instead utilize reference images that are similar to the input images by using feature extraction and matching [8]. However suitable reference images are not conveniently available.

Furthermore, fully automatic colorization models are proposed with the recent advancement of Convolutional Neural Networks (CNN) [1][9][10][11]. Some methods directly estimate chrominance values [1][9] and others quantize the chrominance space into discrete colors [10][11] which initialize their networks with publicly available pre-trained models and adapt them to do colorization. The work of [1] proposes a model that combines both global and local image features via a fusion layer. The model is trained by a classification loss for colorization, which exploits the class labels of the dataset to more efficiently learn the global features. Additionally, both [1] and [11] require combining the raw output of the CNN with the input image that used as luminance and transfers the details of the grayscale images to the final RGB images, which is not suitable for NIR colorization.

Recently, Limmer et al. [12] propose an approach that uses CNN to perform an automatic integrated colorization from a NIR image. The transfer is performed by feeding a locally normalized image pyramid to a deep multi-scale CNN, which uses the mean filtered input image as an additional input to the final fully connected layer to deblur the output. Besides, a triplet based colorization model is proposed in the same scheme of architectures of DCGAN, which generates three instances, each corresponding to one of channels of the (RGB) image [2]. However [2] is trained and tested via image patches, which is not suitable for large scale images, and their results are not clear.

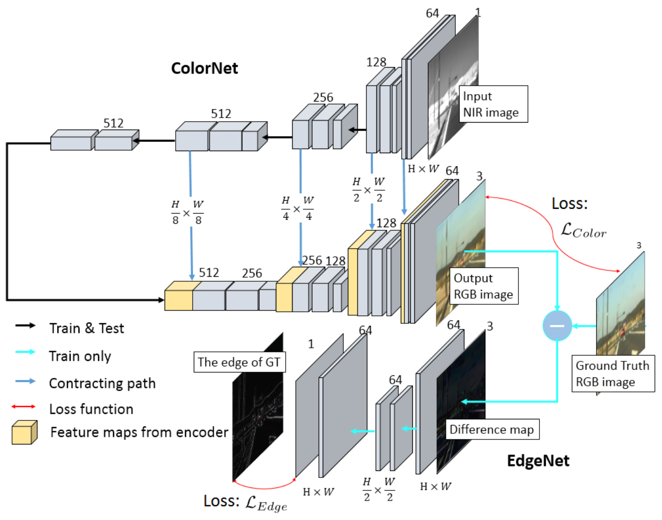
Our colorization model does not rely on any hand-crafted or pre-trained model. Due to the proposed architecture, the network propagates context information to higher resolution layers, which remain the details of the input NIR images. Furthermore, our model can process images of any resolution and we learn everything in an end-to-end fashion.

3. S-SHAPE NEURAL NETWORK

This section describes S-shape neural network, in short SNet, to colorize NIR images. The architecture is inspired by UNet [3] and combines this with a shallow edge loss network which is used as a self-generated loss function. The SNet model is a combination of a skip connected encoder-decoder pipeline named ColorNet with an edge loss network which is also a shallow encoder-decoder pipeline to enhance edges, named EdgeNet, as illustrated in Fig. 2.

3.1. ColorNet

The encoder takes a NIR image as input and produces a latent feature representation of that image. The decoder takes this feature representation and generates the RGB image. It is also



ColorNet
512
512
256
128
64
1
Input NIR image
H × W
$\frac{H}{8} \times \frac{W}{8}$
$\frac{H}{4} \times \frac{W}{4}$
$\frac{H}{2} \times \frac{W}{2}$
512
256
256 128
128
64
3
Output RGB image
Loss: $\mathcal{L}_{Color}$
3
Ground Truth RGB image
Train & Test
Train only
Contracting path
Loss function
Feature maps from encoder
The edge of GT
1
64
64
64
3
Difference map
EdgeNet
Loss: $\mathcal{L}_{Edge}$
H × W
$\frac{H}{2} \times \frac{W}{2}$
H × W

Fig. 2. Overview of our S-shape network.

important to connect the encoder and the decoder through a contracting path.

Our encoder consists of 5 convolution blocks. The input is a single channel of NIR image and output is a $512 \times 7 \times 7$ dimensional feature representation. Each block consists of two 3×3 convolutions, each followed by a Batch Normalization (BN) layer and a rectified linear unit (ReLU). After each block (except the last block), we use a max pooling (factor=2) layer and double the number of feature maps.

Our decoder consists of 4 convolution blocks. Each block firstly up-samples the input feature maps then concatenates with the cropped feature maps from the symmetric encoder. Due to the loss of border pixels in every convolution, the cropping is necessary. In addition, each block is followed by two 3×3 convolutions with BN layer and ReLU layer, which is similar to the encoder but we quarter the number of feature maps. Finally, behind the last block, a 1×1 convolution followed with a tanh() activation layer, which is suitable for generating images [11], is used to map to a three channel RGB output.

We train our color network by regressing to the ground truth of RGB images. We require a loss function for measuring generation errors to minimize the distance between two images pixel-wise. The first consideration is L_2 regression. Our objective is to learn a mapping predicted $\hat{y} = \mathcal{F}(x)$ to the ground truth y. For a pixel, we defined the loss function as:

$$\ell(\hat{y}, y) = \|\hat{y} - y\|^2, \tag{1}$$

and subsequently, for a batch of images, the loss function is:

$$\mathcal{L}_{Color}(\mathcal{F}(\mathbf{X};\theta), \mathbf{Y}) = \sum_{b}^{B} \sum_{h,w}^{H,W} \ell(\mathcal{F}(\mathbf{X};\theta)_{b,h,w}, \mathbf{Y}_{b,h,w}), \tag{2}$$

where, $\mathbf{X} \in \mathbb{R}^{H\times W\times 1\times B}$ is a set of one channel NIR images; $\mathbf{Y} \in \mathbb{R}^{H\times W\times 3\times B}$ presents a set of the RGB color channels of the images; H, W, B are height, weight and batch size; the mapping is learned with ColorNet $\mathcal{F}$, parameterized by θ.

3.2. EdgeNet

Our EdgeNet is illustrated in Fig. 2. It consists of a shallow encoder-decoder symmetrically: 2 layers of encoding and 2 layers of decoding. In addition, the number of feature maps is increasing in the encoder and decreasing in the decoder. Each encoding layer consists convolutions with stride 2 for down-sizing, batch normalization, leaky ReLU activations and each decoding layer consists of transposed convolutions with stride 2 for up-sampling, batch normalization and tanh() activation. Here both are 3×3 kernel size.

We use this network as a smart 'loss function' for not only enhancing edges but also learning the color of other regions in the ground truth again, which can be trained together with the ColorNet to jointly improve performance. The loss network can become the most suitable 'loss function' between the generated result and the ground truth trough training. The input is the difference map between the generated result and the ground truth (original RGB images), and the output is the edges of the original RGB image, which means the ground truth (GT) of EdgeNet is the edge of the original RGB images.

Although the ground truth of our loss network is an edge image of ground truth that is a single channel with value 0 and 1, which seems like a classification task for the EdgeNet with cross-entropy loss, we instead see it as a regression task for which we will get a better result. Subsequently, the same as ColorNet, we also use L_2-loss for a batch of images:

$$\mathcal{L}_{Edge}(\mathcal{F}_e(\mathbf{D};\theta_e),\mathbf{E}) = \sum_{b}^{B}\sum_{h,w}^{H,W} \ell(\mathcal{F}_e(\mathbf{D};\theta_e)_{b,h,w}, \mathbf{E}_{b,h,w}), \tag{3}$$

where, $\mathcal{L}_{Edge}$ is the loss function of the EdgeNet; $\mathbf{D} \in \mathbb{R}^{H\times W\times 3\times B}$ is a set of the difference maps between the output of ColorNet and GT; $\mathbf{E} \in \mathbb{R}^{H\times W\times 3\times B}$ presents a set of the edge of RGB images; the mapping is learned with EdgeNet $\mathcal{F}_e$, parameterized by θ_e.

The task of loss network is to assist ColorNet to get a clearer result but not to really get an edge image from the difference map that presents weak regions of the generated color image compared to the GT. The goal of this network is to reduce the errors of difference map which is similar to the purpose of a loss function. In fact, we expect that the loss network works well, but if we set the GT of EdgeNet to 0, the weights of loss network will tend to be all 0, and cannot perform well. Instead, we change the GT of EdgeNet to be an edge image of the original RGB image, which can not only be successfully trained but also enhance the edge and stabilize other regions to improve the result of ColorNet. As we know, except the edge of GT, most values are equal to 0 in the edge image, which means that the edge loss network tries to let the values of the difference map tend to be all 0 except edges. Subsequently, if color regions in a color image are learned well, the edges of these regions will be clear too. In fact, this is a better way to enhance the edges in color images. From Equation 4 to Equation 6, we see how EdgeNet works ('$\rightarrow$' means 'tend to'):

$$\mathbf{D} = \mathcal{F}(\mathbf{X};\theta) - \mathbf{Y} \rightarrow \mathbf{E}, \tag{4}$$

$$\mathcal{F}(\mathbf{X};\theta) \rightarrow \mathbf{E} + \mathbf{Y}, \tag{5}$$

$$\begin{aligned}&\mathcal{L}_{Edge}(\mathcal{F}_e(\mathbf{D};\theta),\mathbf{E}) \\ &= \sum_{b}^{B}\sum_{h,w}^{H,W} \ell(\mathcal{F}_e(\mathcal{F}(\mathbf{X};\theta) - \mathbf{Y};\theta)_{b,h,w}, \mathbf{E}_{b,h,w}).\end{aligned} \tag{6}$$

We define the overall loss function as:

$$\mathcal{L} = \mathcal{L}_{color} + \mathcal{L}_{edge}. \tag{7}$$

4. HIBIKINO DATASET

The model was trained and evaluated by real-world images of Japanese road scenes. As the mentioned application scope is mainly for assisting drivers and [12] does not open its dataset to the public, a dataset needs to be assembled accordingly. The images in the dataset were taken by two cameras: one is RGB camera (Artary camera Artcam-1300mi-nir); one is NIR camera (Logitech web camera Carl Zeiss Tessar). Although these two cameras are fixed together, they have different extrinsic alignment and intrinsic parameters. Therefore, the RGB images and NIR images are matched using a pixel to pixel registration. We use a feature-based method to find correspondence between image features such as points, lines, and contours. Given manually the correspondence between a number of points in two images, a geometrical transformation is then determined to map the target image to the reference images, thereby establishing point-by-point correspondence. Finally, 1806 image pairs were collected for the training set, 97 pairs for validation set and 75 pairs for the testing set, which is smaller than [12] but more complex with regard to contents (containing buildings road). The image pairs in the testing set are not contained in training set. Fig. 3 shows various example image pairs from the dataset.

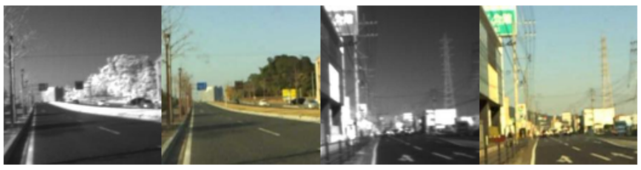


Fig. 3. Exemplary images from the dataset.

5. EXPERIMENTS

We train the SNet model with images of 224×224 pixels. While our model is able to process images of any size, it is optimum when the size of the input image is 224×224 pixels. The SNet was trained using stochastic Adam optimizer which prevents overfitting and leads to convergence faster [13]. During the learning process, we use the following hyper-parameters: learning rate 0.0001 for both ColorNet and EdgeNet; leak ReLU 0.2; batch size 4.

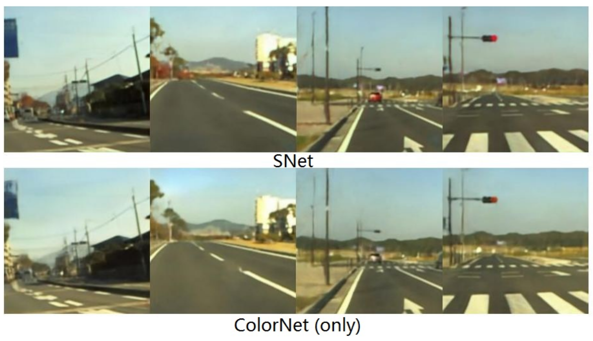


Fig. 4. Experimental results: the first row is the results of SNet and the second row is from ColorNet without EdgeNet.

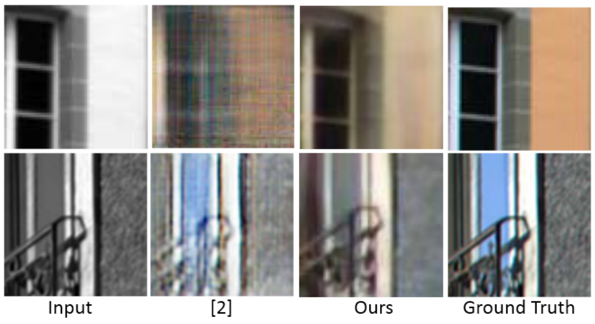


Fig. 5. Comparison with prior work of [2].

We can see our EdgeNet improves the clearness of edges and enhances the color of trees and cars from Fig. 4. Besides, Fig. 5 shows results applied on images from [14] in the same situation with [2]. Our results are much clearer than [2].

Fig. 6 shows results from other contemporary colorization methods. We can see our method is more suitable for infrared images colorization than other colorization methods, producing a qualitative output that is most similar to the GT. We calculate the cosine similarity and PNSR statistical evaluation measures to provide quantitative performance analysis based on the GT in Table.1.

The proposed method can colorize NIR images fully automatically by our SNet. However, some information cannot be recovered from a single channel NIR images. For example, the traffic signal, cars, and buildings sometimes are colorized by false color that depends on the color of the object in the dataset (Fig. 7). In addition, the dataset is small and only contains the road scene, which limits the general robustness of our network trained by this dataset.

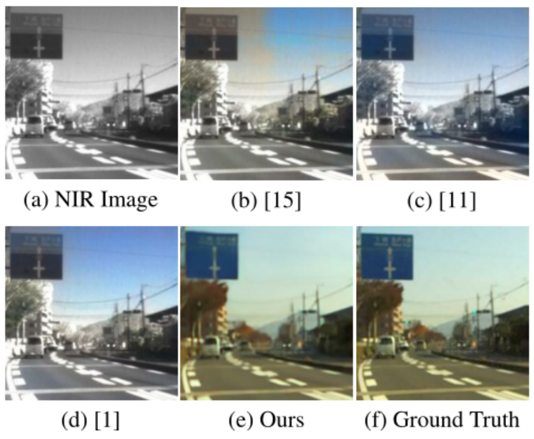


Fig. 6. Comparison with other Colorization methods.

Table 1. Comparison results with other methods.

	NIR Image	[15]	[11]	[1]	SNet
cosine()	0.9211	0.9164	0.9158	0.9144	0.9080
PSNR	13.89	14.09	14.03	14.07	22.53

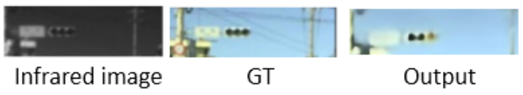


Fig. 7. Exemplar erroneous colorization results.

6. CONCLUSION

This paper presented a novel architecture called SNet for the colorization of NIR image, which consists of a ColorNet and an EdgeNet. It is a novel way to use the EdgeNet to not only enhance the edges but also stabilize color regions. We can see from the results that the SNet is able to obtain colorful and clear RGB images from the given NIR image. Compared to other grayscale image colorization or NIR image colorization. Our SNet has many potential applications such as segmentation. Future work focuses on collecting NIR and RGB images to make a bigger training and testing dataset, try other datasets, and improve the results.

7. ACKNOWLEDGMENTS

This work was partially supported by JSPS KAKENHI Grant Number 18K11380 and Waseda University Grant for Special Research Projects (Project No.2017K-260).

8. REFERENCES

[1] 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)*, vol. 35, no. 4, pp. 110, 2016.

[2] Patricia L Suárez, Angel D Sappa, and Boris X Vintimilla, "Infrared image colorization based on a triplet dcgan architecture," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2017, pp. 212–217.

[3] Thorsten Von Eicken, Anindya Basu, Vineet Buch, and Werner Vogels, "U-net: A user-level network interface for parallel and distributed computing," in *ACM SIGOPS Operating Systems Review*. ACM, 1995, vol. 29, pp. 40–53.

[4] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[5] Anat Levin, Dani Lischinski, and Yair Weiss, "Colorization using optimization," in *ACM transactions on graphics (tog)*. ACM, 2004, vol. 23, pp. 689–694.

[6] Liron Yatziv and Guillermo Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1120–1129, 2006.

[7] Bin Sheng, Hanqiu Sun, Shunbin Chen, Xuehui Liu, and Enhua Wu, "Colorization using the rotation-invariant feature space," *IEEE computer graphics and applications*, vol. 31, no. 2, pp. 24–35, 2011.

[8] Aditya Deshpande, Jason Rock, and David Forsyth, "Learning large-scale automatic image colorization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 567–575.

[9] Tech. Rep. D. Ryan, "Automatic colorization," `http://tinyclouds.org/colorize/`.

[10] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich, "Learning representations for automatic colorization," in *European Conference on Computer Vision*. Springer, 2016, pp. 577–593.

[11] Richard Zhang, Phillip Isola, and Alexei A Efros, "Colorful image colorization," in *European Conference on Computer Vision*. Springer, 2016, pp. 649–666.

[12] Matthias Limmer and Hendrik PA Lensch, "Infrared colorization using deep convolutional neural networks," in *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*. IEEE, 2016, pp. 61–68.

[13] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[14] Matthew Brown and Sabine Süsstrunk, "Multi-spectral sift for scene category recognition," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 177–184.

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.