Generating Visible Spectrum Images from Thermal Infrared

Amanda Berg[1,2], Jörgen Ahlberg[1,2], Michael Felsberg[2]
[1]Termisk Systemteknik AB, Diskettgatan 11 B, 583 35 Linköping, Sweden
[2]Computer Vision Laboratory, Dept. EE, Linköping University, 581 83 Linköping, Sweden
{amanda.,jorgen.ahl}berg@termisk.se, {amanda.,jorgen.ahl,michael.fels}berg@liu.se

Abstract

Transformation of thermal infrared (TIR) images into visual, i.e. *perceptually realistic color (RGB) images, is a challenging problem. TIR cameras have the ability to see in scenarios where vision is severely impaired, for example in total darkness or fog, and they are commonly used,* e.g.*, for surveillance and automotive applications. However, interpretation of TIR images is difficult, especially for untrained operators. Enhancing the TIR image display by transforming it into a plausible, visual, perceptually realistic RGB image presumably facilitates interpretation. Existing grayscale to RGB, so called, colorization methods cannot be applied to TIR images directly since those methods only estimate the chrominance and not the luminance.*

In the absence of conventional colorization methods, we propose two fully automatic TIR to visual color image transformation methods, a two-step and an integrated approach, based on Convolutional Neural Networks. The methods require neither pre- nor postprocessing, do not require any user input, and are robust to image pair misalignments. We show that the methods do indeed produce perceptually realistic results on publicly available data, which is assessed both qualitatively and quantitatively.

1. Introduction

This paper addresses the problem of transforming *thermal infrared* (TIR) images to visual, *i.e.* perceptually realistic RGB images. The process of adding color to black-and-white photography or visual grayscale images is commonly known as *colorization*. Colorization of grayscale visual images is an ambiguous, yet well-researched problem [3, 7, 14, 15, 18, 25, 27, 32, 39]. It is ambiguous in the sense that a grayscale intensity value can correspond to multiple color values. Despite this ambiguity, recent methods show impressive results, see *e.g.* the Colorful Image Colorization[1] [39] and Let there be color![2] [18] demos.

When colorizing visual grayscale images, the luminance is taken from the input image and only the chrominance has to be estimated. In contrast, colorizing TIR images requires estimation of both luminance and chrominance. Further, there is no direct relation between object appearance in TIR (the thermal signature) to its visual appearance (the perceived color). Hence, the problem at hand is more difficult than that of grayscale colorization and requires a process that generates RGB images *from a semantic representation* of the TIR.

Recent grayscale and NIR colorization methods base their success on Convolutional Neural Networks (CNNs) [3, 4, 7, 14, 18, 28, 32, 33, 34, 39] since they are able to model the semantic representation of an image. These techniques are dependent on large sets of training data. Finding training data for visual grayscale colorization is simple. In contrast, publicly available large datasets with corresponding TIR and visual RGB image pairs are rare, only one suitable instance could be found; the KAIST-MS traffic scene dataset [17]. For this dataset, a method for colorization of TIR images must be able to handle *image pair misalignments*. Since glass is opaque in TIR wavelengths, optics for thermal cameras are typically made of materials like germanium. Thus, it is difficult (but not impossible) to acquire thermal and visual images with the same optical axis. As a consequence, there are to our knowledge no available TIR and visual image pair datasets with perfect pixel to pixel correspondence.

Thermal infrared cameras, *long-wave infrared* (LWIR) cameras in particular, have seen an increase in popularity in recent years due to increased resolution and decreased cost. While previously being mostly of interest for military purposes, thermal cameras are entering new application areas [10]. Thermal cameras are now commonly used, *e.g.*, in cars and in surveillance systems. The main advantages of thermal cameras are their ability to see in total darkness, their robustness to illumination changes and shadow effects, and less intrusion on privacy [2]. Due to the variety of inter-

[1] http://richzhang.github.io/colorization/

[2] http://hi.cs.waseda.ac.jp:8082/

esting applications, but at the same time for humans difficult interpretation, transformation of TIR is highly relevant. Accurate transformation of TIR images significantly improves observer performance and reaction times in, *e.g.*, tasks that involve scene segmentation and classification [35].

Contributions This paper proposes two methods, a two-step and an integrated approach, that transform TIR images into visual RGB images, a problem that is previously unaddressed (except for [29]). The proposed methods convert TIR images to plausible visual luminance and chrominance and are robust to image pair misalignments. The methods are evaluated, both qualitatively and quantitatively, on a publicly available dataset with convincing results.

2. Background

2.1. Infrared light and cameras

The infrared band of the electromagnetic spectrum is large and is usually divided into smaller parts depending on their properties. The *near infrared (NIR)* band is dominated by reflected radiation and is dependent on illumination. Essentially, it behaves just as visual light, except that we cannot see it. In contrast, the *thermal infrared (TIR)* band is dominated by *emitted* radiation. That is, the "light" received by the camera is mostly emitted from the observed objects and is related to the temperature of the object, not reflected from a light source (such as the sun). TIR is commonly subdivided into *mid-wave (MWIR, 3-5 μm), long-wave (LWIR, 8-12 μm)*, and sometimes also *far infrared (FIR)*. Objects at normal everyday temperatures emit mostly in the LWIR band (while a hot object like the sun emits most in the visual band), thus making the LWIR band the most suitable for night vision. In addition, cameras for LWIR based on microbolometer sensors have become more common and less expensive in recent years.

2.2. Related work

Early visual grayscale colorization methods have been heavily dependent on user input and interaction in the form of *e.g.* scribbles [27], texture classification [31], extracted features [16], as well as reference images for color transfer [15, 19, 37]. In recent years, deep Artificial Neural Networks, or more specifically, Convolutional Neural Networks (CNNs) [9, 26] have been successfully applied to a wide range of topics, *e.g.* image classification [23], image style transfer [21], and super resolution [6]. The success of deep learning inspired automatic CNN based visual grayscale image colorization methods [3, 4, 7, 14, 18, 32, 39] as well as NIR colorization methods [28, 33, 34].

Automatic colorization methods have a few common problems. First, colorization is, as previously mentioned, an ambiguous problem that is heavily dependent on semantics. Global priors, *e.g.* time of day, weather or location, can affect the color scheme in an image. Current colorization methods mitigate this problem either by using millions of training images [18, 25, 39], by limiting the method to a specific type of images (such as bedroom images [3], or faces & churches [7]), or by including global priors from the dataset itself (day/night, indoor/outdoor, etc.) [18]. The latter approach requires a dataset with such annotations.

Second, colors, when automatically learnt, often tend to be desaturated. Further, certain objects, *e.g.* cars, can have various colors while other, related objects, *e.g.* police cars, should have a specific color. The problem of diverse colorization is addressed in several works [3, 7, 14, 25, 32, 39]. Zhang *et al.* [39] and Larsson *et al.* [25] address the color diversity problem by treating it as a classification rather than a regression task. Larsson *et al.* [25] predict per-pixel histograms and Zhang *et al.* [39] quantize the chrominance space and use class-rebalancing at training time. The loss is, however, per-pixel based and does not enforce spatial coherence which occasionally results in speckle noise. Further, Cao *et al.* [3] employ a conditional GAN architecture and, unlike most other methods, do not use an autoencoder structure. Deshpande *et al.* [7] use a Variational Autoencoder, and Guadarrama *et al.* and Royer *et al.* [14, 32] propose methods based on probabilistic PixelCNNs.

Third, a question arises: What is accurate colorization? How does one provide a measure of the accuracy of a colorization? Some papers use image distance error measures like Root Mean Square Error (RMSE), Peak Signal to Noise Ratio (PSNR), or Structural Similarity (SSIM) [7, 25, 33] while others provide user studies [3, 18] or Visual Turing tests [14, 39].

The above-mentioned colorization methods estimate the chrominance from the luminance, and are thus not directly applicable to colorization of infrared images, where also the luminance has to be estimated. A few recent publications treat colorization of NIR images [28, 33, 34]. NIR images are dominated by reflected radiation (such as sunlight) reflected on the objects in the scene. The difference between NIR and visual red light is just a small shift in wavelength; NIR is thus quite similar to visual light, especially to the red channel of an RGB image. TIR images, on the other hand, are dominated by emitted radiation, which is correlated to the temperature, not the color, of the observed objects. The color of objects can thus be retrieved from a TIR image only by some higher level semantic information. Limmer *et al.* [28] propose a method for transferring the RGB color spectrum to NIR images using deep multi-scale CNNs. In order to preserve details, the high frequency features are transferred in a post-processing step. Suárez *et al.* [34] utilize a DCGAN architecture to colorize 64×64 patches of architectural NIR images. They improve the method in [33] by separating estimation of each channel into a three channel DCGAN architecture. No merging of patches is done and

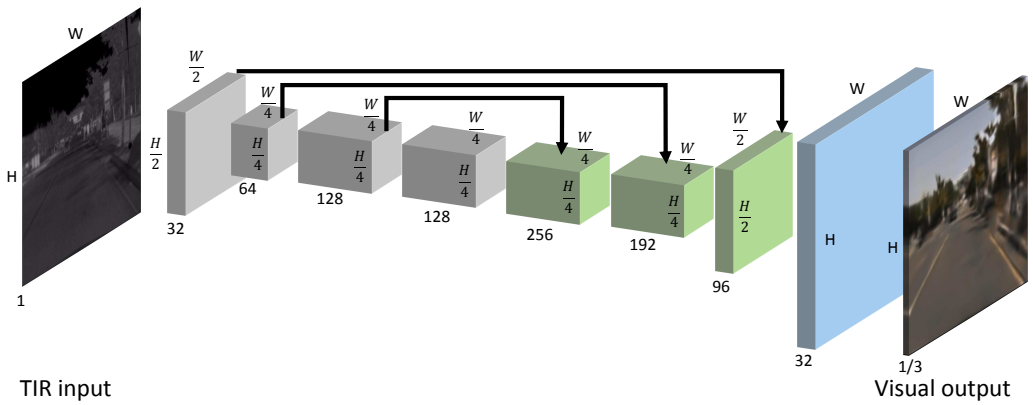


Figure 1: Overview of the architecture used in the two proposed methods. Each gray block represents a `Convolution`, `BatchNormalization`, and `LeakyReLU` layer and each green block consists of one `Convolution`, `BatchNormalization`, `Dropout`, `Concatenation`, and `ReLU` layer. The second and third green block also contains an `Upsampling` layer. The blue block contains a `Convolution`, `BatchNormalization`, and `Dropout` layer. Arrows show skip connections between layers.

the method is evaluated per patch only.

Infrared color fusion or mapping, which are well studied research areas [1, 13, 35, 41, 40], combine multispectral images by mixing pixel intensities or require a manually selected reference image or object segmentations [12], in contrast to the proposed methods.

Cross-domain image to image translation methods have made much progress recently [8, 20, 29, 38, 42]. The unsupervised method proposed by Liu *et al*. [29] is evaluated on thermal images in an early version of the paper. The translated TIR to RGB images have cartooning effects, probably due to regularization or a piecewise constant prior. The main difference to our approach is that we achieve smooth RGB results just by adjusting the objective function in a suitable way. We intended to directly compare to the method by Liu *et al*., but we did not have access to their TIR to RGB network model, neither on request.

In the following, we describe two novel methods that transform full TIR images into visual, *i.e*. perceptually realistic, RGB images, without requiring post-processing or user input.

3. Method

The two proposed methods are based on one proposed architecture. The architecture is inspired by the generator architecture in [20], an overview can be seen in Fig. 1. It has an autoencoder structure with an encoder and a decoder part. Details and key features are further discussed below.

3.1. Network architecture

Since there is no direct relation between the object appearance in TIR to its visual appearance, the proposed architecture is required to generate RGB images from a semantic representation of the TIR images. Motivated by the success of previous CNN based colorization methods, we assume that these underlying representations common to TIR and RGB images can be modelled using deep learning, and more specificly, CNNs with autoencoder structure. This approach has been verified in [29] where a Coupled GAN with a Variational Autoencoder (VAE) is used as generator.

An overview of the proposed architecture is presented in Fig. 1. The network has an encoder-decoder structure based on the generator architecture in [20]. The encoder contains four blocks, each block consists of a `Convolution`, a `BatchNormalization`, and a `LeakyReLU` layer. The `Convolution` layers in the first two blocks have stride equal to two and, thus, downsample the image to one quarter of its height and width. As the size is decreased, the number of channels are increased from 1 to 128.

The decoder has three blocks. Each block consists of a `Convolution`, `BatchNorm`, `Dropout`, `Concatenation`, and `ReLU` layer. In addition, the second and third green block also contain an `Upsampling` layer. The `Concatenation` layers are called skip connections. Skip connections are used in [20] in order to preserve structure. A skip connection simply concatenates all channels at layer i with those at layer $n - i$ (where n is the total number of layers).

Both proposed methods are based on the described architecture. The first method is a two-step approach where the proposed architecture estimates the luminance and an existing grayscale to RGB method is used to estimate the chrominance from the luminance. The second method is an integrated approach that estimates both the luminance and the chrominance using the proposed architecture.

(a) TIR input (b) True RGB (c) L1 (d) DSSIM (e) DSSIM + L1 (f) Proposed

Figure 2: Example of the effect of using f) the proposed separation of the objective function for the luminance and chromatic channels as opposed to using c) L_1, d) DSSIM, or e) DSSIM regularized with L_1, on all three channels.

3.2. Representation

As input, the proposed architecture accepts one channel 8-bit TIR images (due to the 8-bit limitation of the dataset) of any width and height. The output is either a one or a three channel visual image of the same width and height as the input image. Regarding internal color representation, any representation could be used. CIELAB[3] was chosen since it is designed to mimic human perception of colors. Hence, distances between colors in the CIELAB space corresponds to perceptual differences. The CIELAB color space is also used in [4, 18, 39]. Iizuka *et al.* [18] performed an evaluation of three color spaces (RGB, YUV, and CIELAB) and concluded that CIELAB gave the most perceptionally reasonable results.

Regardless of the color space used, the pixel values are normalized to the range $[0, 1]$ since the Sigmoid function is used in the output layer.

3.3. Objective function

The imperfect pixel to pixel correspondence of the TIR and RGB image pairs in the dataset (see Section 4.1) discourages the use of simple pixel to pixel loss functions, like *e.g.* L_1 (and L_2). In addition, it is well known that the L_1 (and L_2) loss produces blurry results on image generation tasks [24]. To mitigate this problem, the loss is separated into a luminance and a chrominance loss.

The human visual system has lower acuity for color differences than for luminance [30]. This is, *e.g.*, exploited by perception-based compression methods like JPEG (which uses fewer bits for chrominance than luminance), and also by the grayscale colorization method designed by Guadarrama *et al.* [14]. Therefore, we propose to separate the loss for the luminance ($\mathcal{L}_L$) and the chromatic ($\mathcal{L}_{ab}$) channels and to use the L_1 loss between the ground truth image y_t^{ab} and the estimated image y_e^{ab} on the chromatic channels only, as

$$\mathcal{L}_{ab}(y_t^{ab}, y_e^{ab}) = |y_t^{ab} - y_e^{ab}| \tag{1}$$

For the luminance channel (L), where humans have higher acuity, we employ an image quality measurement, Structural Similarity (SSIM) [36], between the ground truth image y_t^L and the estimated image y_e^L:

$$\text{SSIM}(y_t^L, y_e^L) = \frac{1}{M}\sum_{j=1}^{M}\frac{(2\mu_{jt}\mu_{je} + c_1)(2\sigma_{jte} + c_2)}{(\mu_{jt}^2 + \mu_{je}^2 + c_1)(\sigma_{jt}^2 + \sigma_{je}^2 + c_2)}$$

where the local statistics, the mean μ_{jt} and μ_{je}, the variance σ_{jt}^2 and σ_{je}^2, and the covariance σ_{jte} are calculated within M pixel neighbourhoods using a sliding window[4]. The size of the sliding window is chosen to incorporate the imperfect pixel to pixel correspondence (see Section 4.3). The constants, $c_1 = (k_1 L_{max})^2$ and $c_2 = (k_2 L_{max})^2$ stabilizes the division with a weak denominator. L_{max} is the dynamic range ($L_{max} = 1$ in this case) and $k_1 = 0.01$ and $k_2 = 0.03$. Color (3-channel) SSIM applied to CIELAB leads to strong clouding effects, as colors shift continuously in uniform regions. From SSIM the Structural Dissimilarity (DSSIM) is derived, which is suitable as a loss function:

$$\mathcal{L}_L(y_t^L, y_e^L) = \text{DSSIM}(y_t^L, y_e^L) = \frac{1 - \text{SSIM}(y_t^L, y_e^L)}{2}$$

The total loss $\mathcal{L}$ is the sum of the two:

$$\mathcal{L} = \mathcal{L}_L + \mathcal{L}_{ab} \tag{2}$$

and the objective is to minimize the loss. In Fig. 2, examples of using the L_1 and DSSIM loss functions on all channels can be seen together with examples of using DSSIM regularized with L_1 as well as the proposed division of the loss function.

[3] CIE L*a*b* D65 to be precise.

[4] We use the implementation by https://github.com/farizrahman4u/keras-contrib/blob/master/keras_contrib/losses/dssim.py

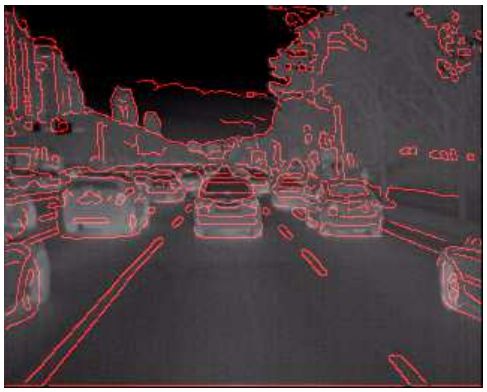


Figure 3: Example of the misalignment of image pairs in the KAIST-MS dataset. The edges from the RGB image has been extracted using a Canny edge detector and overlayed on the LWIR image. Note the large error on the cars at the left image border.

4. Experimental results and evaluation

4.1. Dataset

We use the dataset from the KAIST Multispectral Pedestrian Detection Benchmark [17]. The dataset contains around 95 000 color-thermal (RGB-TIR), 8-bit, day and night image pairs (640×512) captured from a vehicle. The day training set consists of 33 395 RGB-TIR image pairs. Since the images are captured from a vehicle, a number of subsequent images are assumed to be similar depending on the vehicles speed. Therefore, only one quarter, 8 336 pairs (randomly chosen), were included in the training phase. The day validation set consists of 29 168 RGB-TIR image pairs. The complete validation set was used for evaluation.

The dataset has been recorded using a FLIR A35 microbolometer LWIR camera with a resolution of 320×256 pixels. During image alignment, the TIR images were upsampled to 640×512 [17] (using an unknown method). Therefore, the RGB-TIR pairs were downsampled back (using nearest neighbour) to 320×256 prior to training.

While studying the RGB-TIR image pairs, it became clear that they do not have perfect pixel to pixel correspondence. The pixel error in the vertical direction is estimated to be up to 4 pixels, and up to 16 pixels in the horizontal direction, increasing towards the lower left corner, see Fig. 3. Further, inspecting *e.g.* example 5 in Fig. 4, there seems to be a radial distortion not accounted for in the TIR images. Further, the error appears to increase when the car is moving, indicating a slight offset in camera synchronization.

4.2. Methods

Three different TIR to visual image transformation approaches have been evaluated. The latter two, TIR2L and TIR2Lab are the proposed methods:

Naïve Estimating the chrominance using the TIR image as luminance using an existing grayscale colorization method.

TIR2L Estimating the luminance using the proposed network and then the chrominance using an existing grayscale colorization method.

TIR2Lab Estimating both the luminance and the chrominance using the proposed network.

The method proposed by Zhang et al. [39] was chosen as the reference colorization method. The reasons were that it does not require global priors (as *e.g.* the method by Iizuka *et al.* [18]) and neither does it produce multiple versions of the same grayscale image (like *e.g.* Guadarrama *et al.* [14] and Royer *et al.* [32]). The reference colorization method was trained from scratch on the given dataset.

4.3. Training

The proposed architecture is implemented in Keras [5] with Tensorflow back end. For network training, we use the ADAM optimizer [22]. Weigths are initialized with Xavier normal initialization [11]. The proposed architecture was trained for 100 epochs using 8 336 samples with a batch size of 8 samples. The parameters of the ADAM optimizer were set to $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and learning rate 0.001. The `LeakyReLU` layers in the encoder had $\alpha = 0.2$ and the `Dropout` layer had a dropout rate of 0.5.

The size of the sliding window in the DSSIM loss function was set to 4×16 pixels in order to incorporate the misalignment of the dataset described in Section 4.1.

Training of the proposed architecture was done on a NVIDIA GTX1080 GPU. A single training epoch with batch size 8, 1 042 iterations, and 8 336 samples took about 10 minutes. In total, the network was trained for 100 epochs (about 18 hours) at image resolution 320×256 pixels.

As implementation of the method by Zhang *et al.* [39] the Keras implementation by de Boissière[5] has been chosen. The experiments have been performed on a NVIDIA K40 GPU. A single training epoch with batch size 8, 1 042 iterations, and 8 336 samples took around 1 hour and 36 minutes. In total, the network was trained for 38 epochs (about 2 days and 18 hours) at image resolution 320×256 pixels. At 38 epochs, the loss had converged and the training was considered to be finished.

4.4. Quantitative evaluation

Four image distance error measurements for quantitative evaluation have been used: L_1, Root Mean Squared Error (RMSE), Peak Signal To Noise Ratio (PSNR), and Structural Similarity (SSIM). Error measurements are calculated in the RGB color space in range $[0, 1]$, because this is the expected format for SSIM. Results for the Naïve, TIR2L, and

[5] https://github.com/tdeboissiere/DeepLearningImplementations/tree/master/Colorful

Method	L1	RMSE	PSNR	SSIM
Baseline	0.24 ±0.04	0.84 ±0.10	9.34 ±1.23	0.43 ±0.08
Naïve	0.24 ±0.04	0.82 ±0.09	9.55 ±1.21	0.45 ±0.08
TIR2L	0.14 ±0.04	**0.46** **±0.09**	**14.7** **±2.18**	**0.64** **±0.08**
TIR2Lab	**0.13** **±0.04**	**0.46** **±0.09**	**14.7** **±2.20**	**0.64** **±0.08**

Table 1: Image distance validation results, mean and standard deviation for 29 168 image pairs.

TIR2Lab methods can be seen in Table 1. In addition, error measurements are calculated between the input TIR image and the true RGB image in order to provide a baseline. All evaluations were performed using the day validation set.

The Naïve method gives a slight, but not significant, improvement compared to the baseline for all image distance measurements in the evaluation. Comparing TIR2Lab to the baseline, it is clear that it gives a significant improvement in terms of image distance.

TIR2L and TIR2Lab have similar results, indicating that the chrominance can be estimated by the proposed network equally well as with [39] for this dataset. However, estimating both luminance and chrominance simultaneously using TIR2Lab is less computationally demanding than first estimating the luminance and then the chrominance from the luminance in a two-step approach as in TIR2L. A forward-pass through TIR2Lab with batch size 8 takes around 1.33 seconds while a forward-pass through TIR2L takes around 2.83 seconds on a NVIDIA GTX1080 GPU.

4.5. Qualitative evaluation

In Fig. 4, six transformation examples for the Naïve, TIR2L, and TIR2Lab methods are provided.

The Naïve method gave a slight improvement compared to the baseline. Since the luminance is taken directly from the TIR image, there is no degradation in terms of structure compared to the original TIR image. The method proposed by Zhang *et al*. [39] does, however, fail to correctly estimate the chrominance which is clearly visible in Fig. 4b.

The colorized images by TIR2L, Fig. 4c, and TIR2Lab, Fig. 4d, have similar appearance. TIR2L appears to have a stronger tendency to colorize the sky in a more pink color which is more similar to ground truth in some cases (ex. 1) and less in others (ex. 5, 6). The dataset was recorded during sunset and the images in the training set have skies with a varying degree of pink. TIR2L also colorizes some road markings in a more vivid orange than TIR2Lab (ex. 3) but sometimes it is the other way around (ex. 1).

Based on our observations, we conclude that the results are similar in terms of subjective assessment. TIR2L transforms the TIR image to a more plausible RGB image in some cases, and vice versa.

Fig. 5 provides a few transformation examples of particular interest. Note that objects that have adopted the background temperature and objects for which different colors is not equivalent to different thermal properties will not be plausible colorized unless they are hallucinated or there is some semantic information related to the different colors. In ex. 5a there is a faint line visible to the left of the blue line in the true RGB image. This line is barely visible in the TIR image and the proposed methods colorizes it only partially. TIR2L colorize the blue line white while TIR2Lab chooses an orange color. The same goes for the case in ex. 5d.

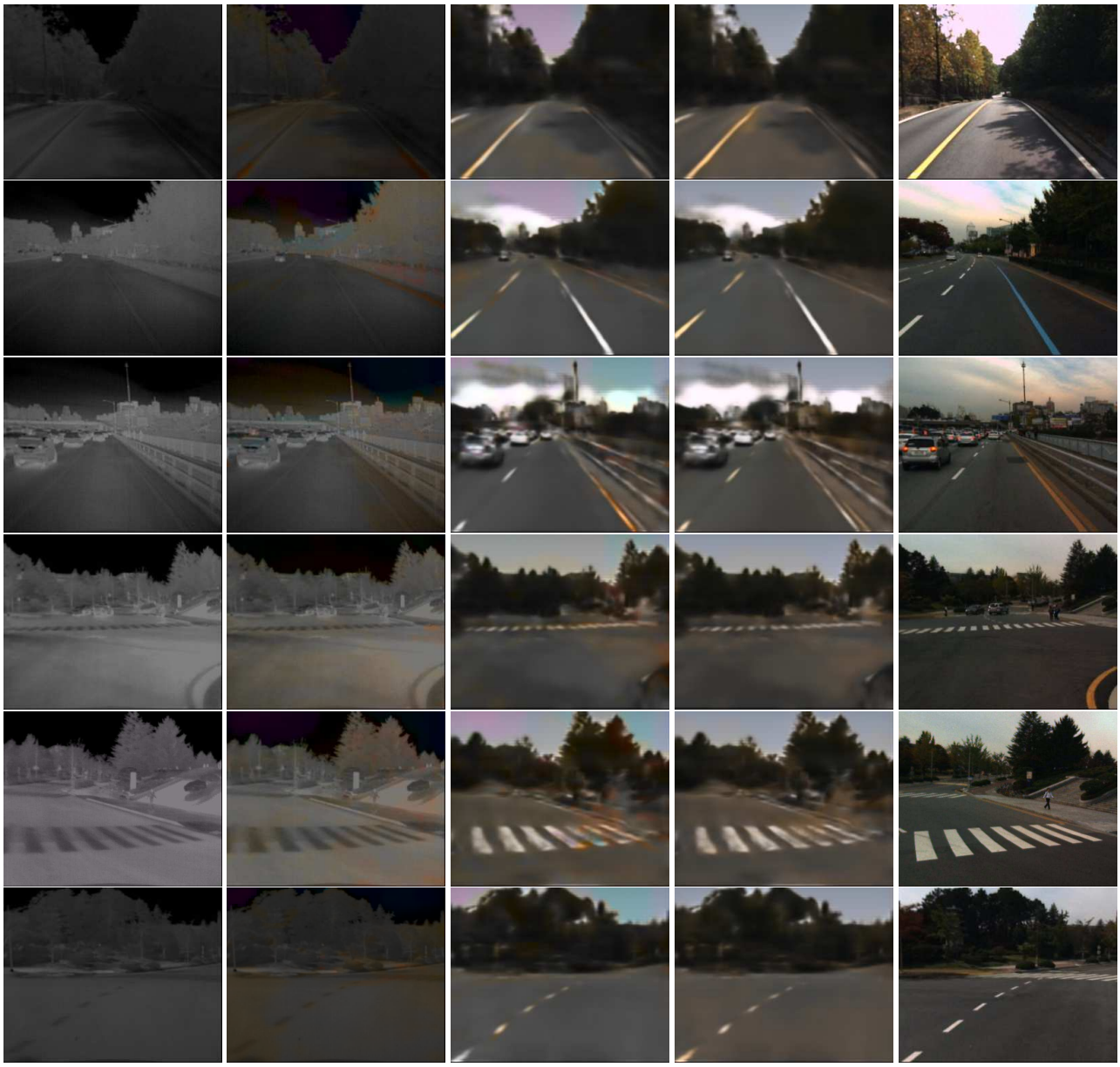
In ex. 5b, both TIR2L and TIR2Lab fail to colorize the road markings since they have the same apparent temperature as the road paving. A similar scenario is the one in ex. 5c where TIR2Lab fails to colorize the crossing as both white and orange (there are both kinds in the dataset). TIR2L on the other hand adds orange markings between the white markings.

Further, in ex. 5d, the brake lights of the vehicles are not colorized correctly. Turning on the brake light does not change the apparent temperature of the lamp cover. However, for the specific application of night vision in traffic scenes, it is possible to fuse colorized images with the true RGB image, where the brake lights (when it is dark) will be clearly visible.

There are two scenarios for which both TIR2L and TIR2Lab fail in most cases. Both are shown in ex. 5e. We believe that this is because urban environments and cars close to the camera are not as frequently occurring in the dataset as more rural environments and cars at longer distances.

4.6. Night to day

In addition to the above mentioned experiments, LWIR night images from the KAIST-MS dataset have been colorized using TIR2Lab (trained on day images). Two examples of colorized day RGB images together with true RGB night images can be seen in Fig. 6. At night, the surroundings will adopt a more homogeneous temperature, thus the contrast in the LWIR images will be lower than during the day. This can potentially be (partly) compensated for by adjusting the dynamic range of the 16-to-8 bit conversion, but was unfortunately not done in the dataset. The low contrast makes it difficult for the network to correctly recognize the different objects in the scene and the output RGB image, colorized with a day color scheme, looks blurred.



(a) TIR input (b) Naïve (c) TIR2L (d) TIR2Lab (e) True RGB

Figure 4: Transformation examples for three different methods.

5. Conclusion

We have addressed the problem of transforming TIR images into visual, perceptually realistic color images, an unaddressed problem despite its relevance for practical applications. In our work, we have proposed a general, deep learning based approach to generate luminance and chrominance information from TIR images. We further suggest two methods based on different variants of the proposed approach to determine chromatic information. The methods are robust to image pair misalignments and have been evaluated both qualitatively and quantitatively on a publicly available dataset. The evaluation was, however, limited to traffic scene images due to the lack of large, publicly available TIR-RGB image pair datasets.

In comparison to grayscale to RGB colorization, which only estimates the chrominance, a TIR to RGB transformation method has to estimate both luminance and chrominance. There is no direct relation between object appearance in TIR to its visual appearance. The proposed approach was, therefore, based on Convolutional Neural Net-

Figure 5: Failure cases for the proposed methods, first row shows input TIR image, second row the output from TIR2L, third row the output from TIR2Lab, and fourth row the true RGB images.

(a) TIR input (b) TIR2Lab (c) True RGB

Figure 6: Examples of a) two night TIR images, colorized using b) TIR2Lab trained on day images and their corresponding c) true RGB images.

works, due to their ability to model semantic representations. The first proposed method estimates both plausible luminance and chrominance using the proposed approach. The second proposed method estimates luminance using the proposed approach and chrominance using an existing grayscale to color transformation method, the method proposed by Zhang *et al*. [39] in this case[6].

Further work is twofolded. First, in order to provide a more extensive evaluation of future TIR to RGB transformation methods, new large TIR-RGB image pair datasets that target other application areas are needed. Second, future work for the method includes tuning of weights for the losses $\mathcal{L}_L$ and $\mathcal{L}_{ab}$. Failure cases for the proposed methods include a slight loss of structure compared to the input TIR image and cloudy colorization of uniformly colored areas, *e.g*. pavements. We do, however, conclude that separating the objective function into luminance and chrominance loss is favourable.

Acknowledgements

The research was funded by the Swedish Research Council through the project Learning Systems for Remote Thermography (2013-5703) and the project Energy Minimization for Computational Cameras (2014-6227).

[6]Code is available at https://gitlab.ida.liu.se/amabe60/PBVS2018

References

[1] E. A. Ali, H. Qadir, and S. P. Kozaitis. Color Night Vision System for Ground Vehicle Navigation. In *Proceedings of SPIE*, volume 9070, page 90700I. International Society for Optics and Photonics, jun 2014. 3

[2] A. Berg, J. Ahlberg, and M. Felsberg. A Thermal Object Tracking Benchmark. In *12th International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, aug 2015. 1

[3] Y. Cao, Z. Zhou, W. Zhang, and Y. Yu. Unsupervised Diverse Colorization via Generative Adversarial Networks. *CoRR abs/1702.06674*, feb 2017. 1, 2

[4] Z. Cheng, Q. Yang, and B. Sheng. Deep Colorization. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, apr 2015. 1, 2, 4

[5] F. Chollet. Keras. https://github.com/fchollet/keras, 2015. 5

[6] R. Dahl, M. Norouzi, and J. Shlens. Pixel Recursive Super Resolution. *CoRR abs/1702.00783*, 2017. 2

[7] A. Deshpande, J. Lu, M.-C. Yeh, M. J. Chong, and D. Forsyth. Learning Diverse Image Colorization. *CoRR abs/1612.01958*, 2016. 1, 2

[8] H. Dong, P. Neekhara, C. Wu, and Y. Guo. Unsupervised Image-to-Image Translation with Generative Adversarial Networks. *CoRR abs/1701.02676*, jan 2017. 3

[9] K. Fukushima. Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition. *Neural Networks*, 1(2):119–130, 1988. 2

[10] R. Gade and T. B. Moeslund. Thermal Cameras and Applications: A Survey. *Machine Vision and Applications*, 25(1):245–262, jan 2014. 1

[11] X. Glorot and Y. Bengio. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*, 2010. 5

[12] X. Gu, M. He, and X. Gu. Thermal Image Colorization using Markov Decision Processes. *Memetic Computing*, 9(1):15–22, mar 2017. 3

[13] X. Gu, S. Sun, and J. Fang. Real-Time Color Night-Vision for Visible and Thermal Images. In *2008 International Symposium on Intelligent Information Technology Application Workshops*, pages 612–615. IEEE, dec 2008. 3

[14] S. Guadarrama, R. Dahl, D. Bieber, M. Norouzi, J. Shlens, and K. Murphy. PixColor: Pixel Recursive Colorization. *CoRR abs/1705.07208*, may 2017. 1, 2, 4, 5

[15] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong. Image Colorization Using Similar Images. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 369–378, New York, NY, USA, 2012. ACM. 1, 2

[16] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu. An Adaptive Edge Detection Based Colorization Algorithm and Its Applications. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, pages 351–354, New York, NY, USA, 2005. ACM. 2

[17] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon. Multispectral Pedestrian Detection: Benchmark Dataset and Baseline. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1037–1045. IEEE, jun 2015. 1, 5

[18] 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 SIGGRAPH 2016)*, 35(4), 2016. 1, 2, 4, 5

[19] R. Irony, D. Cohen-Or, and D. Lischinski. Colorization by Example. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, EGSR '05, pages 201–210, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association. 2

[20] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *CoRR abs/1611.07004*, nov 2016. 3

[21] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *CoRR abs/1603.08155*, mar 2016. 2

[22] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *CoRR abs/1412.6980*, dec 2014. 5

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, pages 1097—-1105. Curran Associates, Inc., 2012. 2

[24] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding Beyond Pixels using a Learned Similarity Metric. *CoRR abs/1512.09300*, dec 2015. 4

[25] G. Larsson, M. Maire, and G. Shakhnarovich. Learning Representations for Automatic Colorization. In *14th European Conference on Computer Vision (ECCV)*, 2016. 1, 2

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[27] A. Levin, D. Lischinski, and Y. Weiss. Colorization Using Optimization. *ACM Transactions on Graphics*, 23(3):689–694, aug 2004. 1, 2

[28] M. Limmer and H. P. A. Lensch. Infrared Colorization Using Deep Convolutional Neural Networks. In *15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 61–68, apr 2016. 1, 2

[29] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised Image-to-Image Translation Networks. *CoRR abs/1703.00848v1*, mar 2017. 2, 3

[30] M. Livingstone. *Vision and art: the biology of seeing*. Harry N. Abrams, New York, 2002. 4

[31] Q. Luan, F. Wen, D. Cohen-Or, L. Liang, Y.-Q. Xu, and H.-Y. Shum. Natural Image Colorization. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pages 309–320, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. 2

[32] A. Royer, A. Kolesnikov, and C. H. Lampert. Probabilistic Image Colorization. *CoRR abs/1705.04258*, may 2017. 1, 2, 5

[33] P. L. Suárez, A. D. Sappa, and B. X. Vintimilla. Infrared Image Colorization Based on a Triplet DCGAN Architecture. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 212–217. IEEE, jul 2017. 1, 2

[34] P. L. Suárez, A. D. Sappa, and B. X. Vintimilla. Learning to Colorize Infrared Images. In *Advances in Intelligent Systems and Computing*, pages 4353–4358. Springer, Cham, jun 2017. 1, 2

[35] A. Toet and M. A. Hogervorst. Progress in Color Night Vision. *Optical Engineering*, 51(1):010901, feb 2012. 2, 3

[36] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, apr 2004. 4

[37] T. Welsh, M. Ashikhmin, and K. Mueller. Transferring Color to Greyscale Images. *ACM Transactions on Graphics*, 21(3):277–280, 2002. 2

[38] Z. Yi, H. Zhang, P. Tan, and M. Gong. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. *CoRR abs/1704.02510*, apr 2017. 3

[39] R. Zhang, P. Isola, and A. A. Efros. Colorful Image Colorization. In *14th European Conference on Computer Vision (ECCV)*, mar 2016. 1, 2, 4, 5, 6, 8

[40] Y. Zheng. An Overview of Night Vision Colorization Techniques using Multispectral Images: From Color Fusion to Color Mapping. In *2012 International Conference on Audio, Language and Image Processing*, pages 134–143. IEEE, jul 2012. 3

[41] Y. Zheng and E. A. Essock. A Local-Coloring Method for Night-Vision Colorization Utilizing Image Analysis and Fusion. *Information Fusion*, 9(2):186–199, apr 2008. 3

[42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *CoRR abs/1703.10593*, mar 2017. 3