Automatic Colorization of Comics through Deep Learning

Simon Schellaert
Supervisors: prof. dr. ir. Wilfried Philips, Joris Roels, dr. ir. Jan Aelterman
Ghent University

ABSTRACT

Line art colorization is a challenging problem. Techniques used for photo colorization are typically not applicable since the input is devoid of any texture. Furthermore, color assignment is usually highly ambiguous due to the sparseness of the input. Recently, Generative Adversarial Nets (GANs) based methods have demonstrated great success at image synthesis tasks. We build upon this work and propose a novel network architecture and loss function that both significantly increase the realism of our results.

Specifically, we modify the standard U-Net architecture to increase its performance for the task at hand. This involves introducing an extra downsampling level, dilated convolutions and replacing batch normalization with instance normalization. That way, our method is able to efficiently process high-resolution imagery with a sufficiently large receptive field. Next, we introduce a multi-scale discriminator to efficiently assess the realism of these high-resolution images. Finally, we propose a novel loss based on total variation regularization that substantially increases the quality of the generated colorizations.

We evaluate our method and variants thereof on two datasets. The first dataset consists of line art generated from comic albums that were published in color. This allows us to compare the results of our method to a ground truth. The second dataset consists of old comic albums that were originally published in black-and-white. Images from this dataset are more challenging due to the use of an older drawing style and the presence of halftoning patterns. We show that our method can produce and realistic and aesthetically pleasing colorizations for both datasets.

CCS CONCEPTS

• **Computing methodologies** → **Image processing**; *Neural networks*;

KEYWORDS

Colorization; Comics; GANs; Convolutional Neural Network

1 INTRODUCTION

Nowadays, most comics are known for their extensive use of color. A few decades ago, however, color printing was still considered too expensive. So, there is an enormous collection of older comics that only appeared in black-and-white. Colorizing these older comics breathes new life into them and makes them more attractive for a wider audience. Unfortunately, doing this manually is a time-consuming process.

Another scenario where colorization takes a lot of time and effort is the creation of a new comic. Typically, a cartoonist starts by drawing the line art of the comic. These black-and-white images are then handed over to a colorist, who is responsible for adding color to them. This involves the colorist examining each image and manually applying an appropriate color to each object.

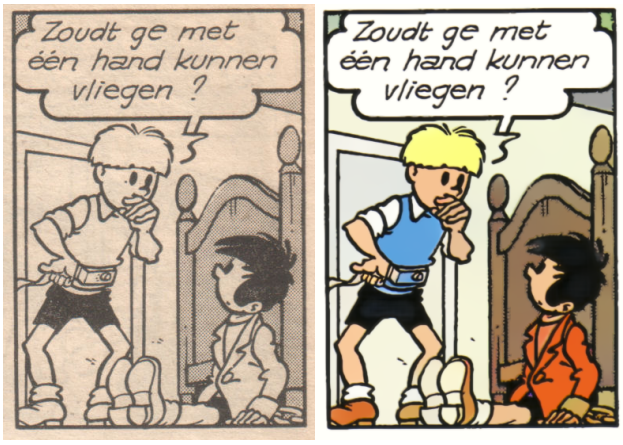


Figure 1: Our proposed method is able to colorize line art from B&W albums (left) without any assistance from the user.

In both cases, automating the colorization process would save a significant amount of time. Colorization, however, is a heavily ill-posed problem. There are frequently multiple conceivable colors that could be assigned to an object. This leads us to focus on generating plausible colorizations that are convincing and aesthetically pleasing for a human observer. Specifically, we tackle the problem of automatically generating a plausible colorization of a given binary comic book image.

2 RELATED WORK

The problem of fully automatically colorizing comic book images has not yet been widely studied. Prior work on comic colorization typically requires the user to provides color hints in the form of so-called *scribbles* [9, 17]. Since our goal is to develop a method that does not require any user intervention, this work is of limited relevance for us. Instead, we identify two different problems that share some important characteristics with our problem. This allows us to determine some approaches that are also relevant to the task at hand.

2.1 Photo Colorization

Photo colorization deals with the problem of hallucinating a plausible color version of a grayscale photograph. Just like in our problem, the input doesn't necessarily contain enough information to

recover the ground truth color. Therefore, rather than reproducing the ground truth, the goal is once again to generate plausible colorizations that can fool human observers.

Modern approaches for this problem successfully leverage large-scale data and convolutional neural networks (CNN) [10, 16, 28]. Recent research has indicated that these networks rely primarily on texture instead of shape [4]. This makes sense since the texture information in the grayscale channel is a big aid for object recognition. In our case, however, the input is a line drawing and is thus devoid of any texture. As a result, architectures that are effective for photo colorization don't necessarily perform well on our task.

Due to the multi-modal nature of the problem, optimizing the aforementioned networks using the traditional Euclidean distance yields unsatisfactory results. If an object can take on multiple distinct colors, the optimal solution according to the L_2-loss will be the mean of these colors. This results in grayish, desaturated colorizations. Consequently, Zhang et al. [28] propose to treat the problem as multinomial classification. While this indeed results in more vivid colorizations, the output also exhibits frequent splotches as the mode of the predicted distribution changes.

2.2 Image-to-image Translation

The goal of image-to-image translation is to translate an input image from one domain to another domain given input-output pairs as training data. This definition encloses a wide variety of tasks that have traditionally been tackled with separate, special-purpose machinery [2, 8, 26, 28]. Example applications include noise removal, super-resolution and image synthesis. Since L_1 loss typically leads to blurry images for many of these tasks [12, 14], the adversarial loss has become a popular choice [6] .

Recently, Isola et al. proposed the pix2pix framework [12], which employs image-conditional GANs for a variety of applications. These GANs learn a loss that penalizes unnatural results, while simultaneously training a generative model to minimize this loss. Because the learned loss automatically adapts to the data, it can be applied to a multitude of tasks that traditionally required very different kinds of loss functions.

Specifically, the architecture consists of two networks: a generator G and a discriminator D. For our task, the objective of the generator G is to translate a line drawing to realistic color image, while the discriminator D aims to distinguish between real and fake images. By pitting these two networks against each other, the generator gradually learns to produce colorizations that are indistinguishable from real ones.

3 APPROACH

Since deep learning is the approach that leads to the best results on related problems, we also opt to use a convolutional neural network. In what follows, we discuss the key features of our network architecture and the loss formulation.

3.1 Network Architecture

A straightforward idea might be to train a less powerful network to generate color hints. Afterward, these hints could be used to color connected components, which are groups of white pixels enclosed within a connected region of black pixels. More closely examining human colorizations, however, reveals that the use of one color per connected component is a crude approximation. In reality, comic images frequently contain multiple colors per connected component.

As such, we opt to utilize a fully convolutional network (FCN) [21] as an end-to-end solution. That way, the network can be applied to a line drawing of any size and maps it to a colorized version with the same resolution. The network thus needs to combine high-level semantics (to recognize objects) with low-level details (to color inside the lines). A popular architecture that provides this ability is the U-Net [19]. The use of skip connections in this architecture allows the upsampling path to exploit contextual information and location information from the downsampling path.

The standard U-Net architecture has a receptive field of 140 x 140 pixels [23]. Since objects in our input images are frequently larger than that size, an increase in contextual information is required. An efficient way to exponentially increase the receptive field is the use of dilated convolutions [27]. Specifically, we introduce an additional downsampling step and introduce dilated convolutions in the two lowest levels of the network. These changes allow us to increase the receptive field to 795 x 795 with only a modest increase in computational cost [3].

Another improvement we make is introducing normalization layers throughout the network. A recent recommendation is to use batch normalization to improve the speed, performance and stability of neural networks [11]. The effectiveness of batch normalization, however, is strongly dependent on the batch size. Since our input and output have a high resolution, the maximal batch size is limited by the available GPU memory. For this reason, we opt to use instance normalization [22]. This allows us to retain some of the benefits of batch normalization while being independent of the batch size [25].

3.2 Loss Formulation

As previously remarked, constructing an effective loss function is challenging due to the multi-modal nature of the problem. Traditional losses, such as the L_2-loss, lead to desaturated colorizations with annoying artifacts. This is caused by these losses only caring about the distance to the ground truth, and not about the realism or plausibility of the results. Since the goal here is to generate colorizations that are convincing for a human, we base our loss function on the adversarial one used in the pix2pix framework [12]. The discriminator can handle the multi-modality of the problem and penalizes deviations that expose our colorization as fakes.

Early experiments revealed that the use of a discriminator significantly boosts the realism of the colorizations. Nevertheless, the results still contain some mistakes that are insufficiently penalized by the discriminator. As such, we introduce another term in the loss function that penalizes these mistakes. This term is based on total variation regularization and is subsequently named *total variation loss*. The final loss function then consists of three terms: L_1-loss, discriminator loss and total variation loss. In what follows, we examine each of these terms in detail.

3.2.1 L_1-loss. While only using L_1-loss does not necessarily result in realistic colorizations, it is nevertheless a useful loss function. For a lot of objects, such as skin, sky or trees, the colorization is

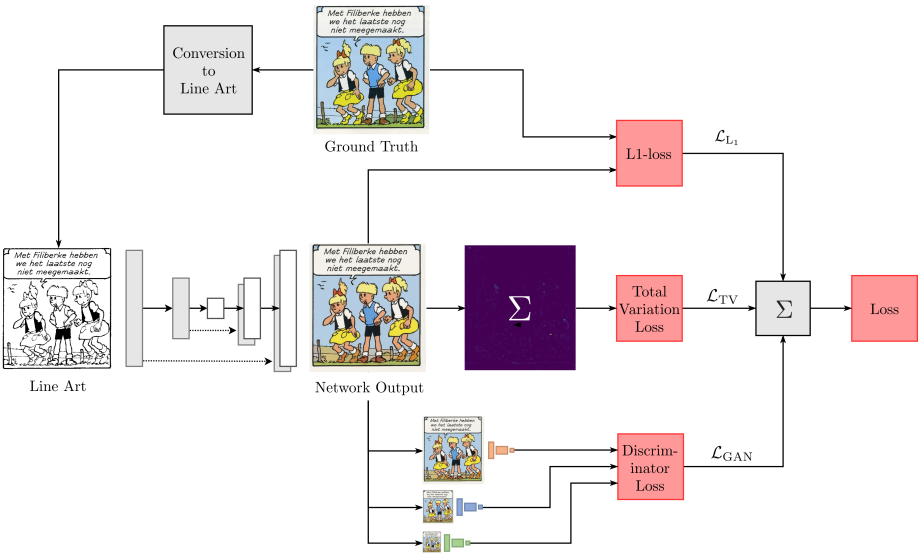


Figure 2: Schematic overview of training our method. The ground truth images are converted to line art and fed to the network as input. The network weights are then optimized based on a loss function consisting of three terms: L_1-loss, discriminator loss and total variation loss.

unambiguous and using this loss is a good approximation. Furthermore, adding an L_1 term makes sure that the output respects the input. This can be seen by noting that this loss penalizes the distance between ground truth outputs, which correctly match the input, and synthesized outputs, which may not. We calculate all distances in the CIELAB color space since distances in this space model perceptual distance.

3.2.2 Discriminator loss. Exclusively using L_1-loss results in reasonable colorizations for a lot of characters and objects. Nevertheless, the colorizations frequently contain imperfections that expose them as fake. The three most prominent mistakes are splotches, unwanted color transitions and desaturated colors.

While these mistakes are conspicuous to human observers, it is hard to construct a loss function that penalizes them. Based on this observation, we introduce a discriminator in our architecture. The task of this discriminator is then to make our colorizations indistinguishable from human colorizations.

Since our output has a high resolution, the receptive field of the discriminator has to be sufficiently large to evaluate it. A discriminator with such a large receptive field, however, is prone to overfitting and has a large memory footprint. To address the issue, we use a multi-scale discriminator, as proposed by Wang et al. [24]. We use 3 discriminator networks that have an identical architecture but operate at different image scales. Specifically, the output of the network is downsampled to scales $\frac{1}{2}$, $\frac{1}{4}$ and $\frac{1}{8}$ before being fed to the respective discriminator. This allows the discriminator at the finest scale to penalize splotches and other artifacts, while the coarsest discriminator enforces global consistency.

The architecture of each individual discriminator network is an unconditional *PatchGAN*, a FCN that classifies 70 x 70 patches as real or fake [12, 21]. The task of the colorization network is then no longer just to be near the ground truth, but also to fool the discriminators by producing natural colorizations.

3.2.3 Total variation loss. While introducing a discriminator reduces the number of artifacts, we still frequently observe unwanted color transitions. Since the discriminator doesn't seem to rectify these as much as other mistakes, we introduce a loss term that penalizes such transitions and thus improves the realism of our results. This term is based on total variation regularization, a process that is traditionally employed for noise removal. The key observation is that noise leads to a high total variation, i.e., the integral of the absolute gradient of the signal is high. Adapted to our problem, color transitions similarly lead to a high total variation of the output image.

Directly applying this idea to our case, however, leads to the gradient being dominated by the transitions between colored areas and black lines. To avoid this, we mask the matrix of gradient magnitudes with a dilated version of the input line art. Afterward, we apply the hyperbolic tangent function to the resulting matrix before summing it. This procedure leads to an estimate of the severity of color transitions in the output. By adding this term to the loss function, we discourage the network from introducing color changes within a surface without robbing it of the ability to do so if needed.

The final loss function is a weighted sum of the three terms defined above. Since each of these terms is made up of differentiable functions, we are able to apply backpropagation to optimize the weights of the network. The final architecture is depicted in figure 2.

4 DATASET

A crucial aspect of every machine learning pipeline is the dataset. By carefully choosing a dataset, we can get more insight into the performance of our method. In what follows, we elaborate on the choice and processing of our dataset.

4.1 Comic Series

Our method can be applied to any comic, as long as a sufficiently large set of colorized training samples is available. We opt to train and test our method on *Jommeke*, a popular Belgian comic series. A primary reason for choosing this series is that the first 91 of the over 290 comics were published in black-and-white. With the introduction of album #92, the series switched to publications in colors. This allows us to test our method on original black-and-white comics. In total, our dataset consists of 2818 scans of black-and-white pages and 9329 scans of color pages.

4.2 Conversion of Color Comics to Line Art

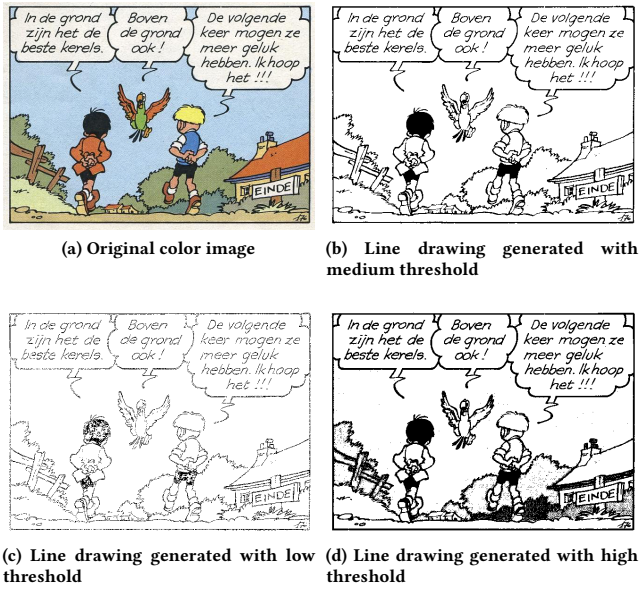
An important remark is that the dataset contains each album either in black-and-white or in color. We thus don't possess the pairs of black-and-white images with corresponding colorized versions that are required for training the network. We can, however, take advantage of the fact that converting color images to black-and-white images is typically much easier than the other direction.

We thus automatically generate a binary line drawing for each color image, to use as the input of our network. In this line drawing, every black pixel should correspond to a black line or surface in the original image. Since our datasets consist of scans, however, the pixels that we want to keep in the line art are not necessarily perfectly black in the scanned color image. Because the actual colors are perturbed by various independent sources of noise, we define a multivariate normal distribution over the CIELAB color space. This distribution expresses the probability that a pixel with a specific color belongs to a black line or surface. We then use this distribution and a chosen threshold to determine which pixels to color black in the line art.

The ideal threshold, however, varies from image to image and is hard to determine without human intervention. Rather than manually determining an appropriate threshold for each image, we randomly pick one from an interval containing the most common thresholds. Thus, each time we feed an image to the network during training, we uniformly sample this interval and generate a line drawing based on the chosen threshold. This results in the network seeing each ground truth color image with a variety of line drawings, varying in the used threshold, as input. This procedure can also be seen as a kind of data augmentation. Sometimes the threshold will be too high, resulting in extraneous black pixels in the input (see figure 3d). Other times, the threshold will be too low, resulting in the disappearance of some line segments (see figure 3c). This diversity makes the network robust to noisy input images.

4.3 Conversion of Old B&W Comics

In the previous section, we proposed a process for converting color images to line art. This allows us to generate the pairs of line art and color image that are needed to train the network. After training, we can use the network to colorize the line art drawn during the production process of a comic. Another major use case is the colorization of old black-and-white comics. As discussed in section 4.1, we also have 2818 scans of black-and-white pages available. These scans, however, are not binary images, as our network expects as input. Rather, they are color scans of yellowed black-and-white pages with specks of dust. This is problematic since an important assumption in supervised learning is that the train and test set are both sampled from the same underlying distribution [5]. We now discuss exactly how these scans differ from the images our network is trained on and how we tackle these differences.



(a) Original color image

(b) Line drawing generated with medium threshold

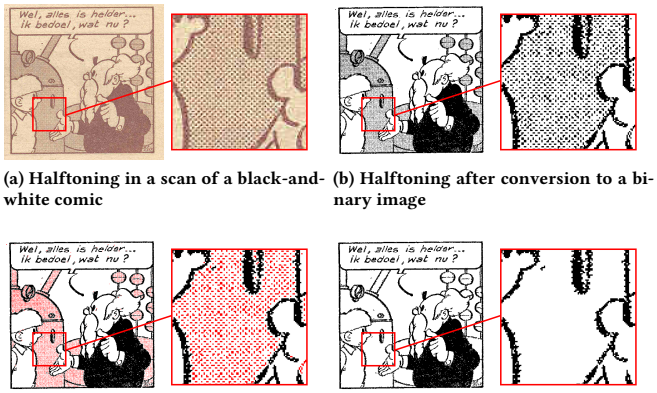
(c) Line drawing generated with low threshold

(d) Line drawing generated with high threshold

Figure 3: Influence of the used threshold on the generated line drawings

4.3.1 Halftoning removal. To convert these scans to binary images, that can be fed to our network, we start by applying the same procedure as above with a different threshold range. The result of applying this procedure to the image in figure 4a, is shown in figure 4b. We observe that the black lines and surfaces are correctly transferred to the binary image. A complication, however, is the presence of halftoning dots in the source image, and thus also in the resulting binary image. In particular, these black-and-white albums use one level of halftoning to add some depth to the scene. The usage of this pattern is purely an artistic choice and doesn't necessarily convey any information about the actual color of an object. Its presence, however, does interfere with the operation of our network as the network is not used to this kind of pattern.

The process of reconstructing a continuous-tone images from a halftone version, is called *inverse halftoning* or *descreening* [13, 15]. A straightforward way to do so is the application of a low-pass filter, such as a Gaussian filter. A problem with this approach is that it blurs the image and thus also throws away relevant details around the edges. For our purposes, however, the halftoning patterns don't



(a) Halftoning in a scan of a black-and-white comic

(b) Halftoning after conversion to a binary image

(c) Binary image with small connected components indicated in red. Notice how this are primarily halftoning dots, but also a piece of the hand.

(d) Binary image after removal of small connected components

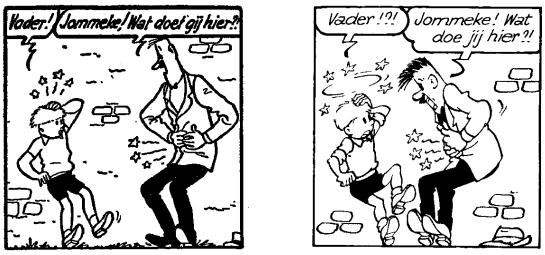
Figure 4: Example of the removal of halftoning patterns

contain useful information, so we can focus solely on removing them instead of converting them to a continuous tone.

To remove these patterns, we make use of *connected-component labeling*, which groups black pixels in the binary image in groups of adjacent pixels [7]. After grouping the black pixels by connected component, we mark the components that consist of less than 7 pixels, as shown in red on figure 4c. These small connected components are mainly halftoning dots. By removing them, we obtain an image without halftoning dots, as shown in figure 4d. This is, however, only an approximation so some artifacts remain around the edges and useful lines are sometimes discarded as well. Nevertheless, the process works well in general and can remove most halftoning dots from the binary image.

4.3.2 Drawing Style. A second important difference between the black-and-white albums and the color albums is the used drawing style. As mentioned in section 4.1, the black-and-white albums were all published before the color albums that our network was trained on. Throughout the years, however, the drawing style has evolved and is now very distinct from the style used in earlier albums. This difference is illustrated in figure 5. The line art on the left is generated from the first edition of the first album in the series, which appeared in black-and-white. The line art on the right is generated from a recently revised version of that first album. This difference is problematic since a lot of the features that distinguish characters in later albums are missing in earlier albums.

Resolving these discrepancies between older and more recent comics turns out to be challenging. The aforementioned album is the only one of which we have a version in both the old and new drawing style. These two versions, however, not only differ in drawing style but also frequently in content. This makes it hard to train a mapping from the old to the new drawing style. Luckily, the drawing style of these old albums converges rather quickly to the recent one. At the 15^{th} album (out of 91 black-and-white albums), the drawing style is already quite close to the contemporary style



(a) Line art from black-and-white album #1

(b) Line art from a recently redrawn version of album #1

Figure 5: Example demonstrating the difference in drawing style between older and more recent albums.

for most characters. From that point on, we can rely on the network robustness to variations in the drawn lines.

To improve the performance on earlier albums, we could use *domain adaptation* [1] or manually colorize some older comics as extra training data. The first option, however, introduces a significant amount of extra complexity, while the second option is not in keeping with our method being fully automatic. We thus don't elaborate on these techniques and accept that the performance is degraded on the earliest albums.

5 RESULTS

Evaluating the quality of the generated colorizations is an open and difficult problem [20]. Traditional metrics, such as the mean absolute error (MAE) or peak signal-to-noise ratio (PSNR), measure the distance to the ground truth rather than the realism of the results. For this reason, we set up a human perceptual study that allows us to evaluate how plausible our colorizations look to human observers. In the following section, we discuss the design and results of this study.

5.1 Variants

As previously discussed, our final loss function consists of three terms. To evaluate the impact of each of these terms on the realism of the results, we train three variants of the networks. For each of these variants, the network architecture and training procedure are identical, except for omitting certain terms in the loss function. Specifically, we train the following variants:

- L_1-loss
- L_1-loss + discriminator loss
- L_1-loss + discriminator loss + total variation loss

The last of these variants is our full method. Besides training three variants, we also evaluate the performance on two datasets: *color comics* and *black-and-white comics*. The first dataset consists of line art that is generated from a set of color albums. This dataset thus has the same characteristics as the data our network trained on and allows comparison of the results to a ground truth. The second dataset consists of line art that is generated from a set of older black-and-white albums. This dataset is harder due to the aforementioned issues with the drawing style. Furthermore, there is no ground truth available for these albums.

5.2 Human Perceptual Study

The primary goal of our method is to produce colorizations that look natural to human observers. We thus follow the procedure outlined by Iizuka et al. and base our study around the question *"Does this image look natural to you?"* [10]. Specifically, we set up an experiment where 84 human observers are each shown 90 random color panels of which they have to determine if they look real or fake. These color images are randomly chosen from sets of colorized images for each of the three variants or from a set of ground truth images. As explained in the previous section, we also split based on whether the input was line art generated from a color album or from a black-and-white album. After viewing each panel, the user has to indicate whether the panel looks real or fake.

During these tests, we want to prevent participants from focusing too much on details they normally wouldn't pay attention to. To do this, we give participants instructions to treat the images like they would do when reading an ordinary comic and use to their gut feeling in case of doubt. Moreover, we also artificially limit the time that each image is shown. Previous experiments indicated that readers take roughly two and a half seconds per panel to read and inspect it. We thus show each panel for three seconds, after which the participant has unlimited time to indicate whether it looks real or not.

5.3 Analysis

The percentage of panels deemed real per network variant and per dataset is shown in table 1. We start by noting that participants labeled 92.6 % of the ground truth images as real. This demonstrates that participants are competent at the task but critical in case of doubt.

Next, we look at the results of colorizations based on line art from color albums (the column *color album*). Our full method can fool participants on 65.2 % of the panels. This is a substantial improvement over the 53.0 % when training the network using only L_1-loss. A remarkable result is that the performance of the variant L_1 + discriminator is lower than the variant trained without a discriminator. We're able to explain this phenomenon by more closely examining the resulting colorizations. While the discriminator leads to more vivid colors, it also leads to the introduction of checkerboard artifacts in the output [18]. The most effective way to fix these artifacts is to change the kernel size of the convolutions in the discriminator to be a multiple of the stride. Since these artifacts, however, are not present in our full method and we only modify the loss function for each variant, we don't do that here.

Finally, we inspect the results when applying our method to albums that originally appeared in black-and-white. The performance of our full method on this set is, with 45.9 %, significantly lower than when applied to line art generated from color albums. We didn't test the other variants of the network on this dataset since we expect the reductions in performance to be comparable to those determined above. Analyzing the least convincing images shows us that the major hurdles are indeed the aforementioned halftoning and drawing style. First, the halftoning removal is frequently imperfect, with the remaining dots leading to grayish and noisy colorizations. Manually removing these remaining dots typically results in a dramatic increase in performance. Secondly, the network is not always able to recognize characters when they are drawn in a more primitive style in the earlier albums.

Variant	Color album	B&W album
L_1	53.0 %	-
L_1 + discriminator	45.6 %	-
L_1 + discriminator + TV	**65.2 %**	45.9 %
Ground truth	92.6 %	N/A

Table 1: Percentage van de prenten dat als echt werd beoordeeld per variant en verzameling.

5.4 Examples

We show colorization results on the set color comics (figure 6) and B&W comics (figure 7). These images were chosen to demonstrate the ability of our method to colorize a wide variety of comic book panels. Note that all these results were generated automatically without any human intervention.

6 CONCLUSION

In this paper, we have presented an end-to-end trainable approach for automatic colorization of line art. We extended the pix2pix framework with a more capable U-Net and loss function that is custom-tailored for comic colorization. We then ran a perceptual study to evaluate the performance of our model on two datasets. This study indicates that our novel loss significantly improves the realism of our results. We've also shown that our method can be used to revive degraded line art of older black-and-white comics.

REFERENCES

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A Theory of Learning from Different Domains. *Mach. Learn.* 79, 1-2 (May 2010), 151–175. https://doi.org/10.1007/s10994-009-5152-4

[2] Antoni Buades, Bartomeu Coll, and J.-M Morel. 2005. A non-local algorithm for image denoising. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2, 60– 65 vol. 2. https://doi.org/10.1109/CVPR.2005.38

[3] Vincent Dumoulin and Francesco Visin. 2016. A guide to convolution arithmetic for deep learning. *CoRR* abs/1603.07285 (2016). http://dblp.uni-trier.de/db/journals/corr/corr1603.html#DumoulinV16

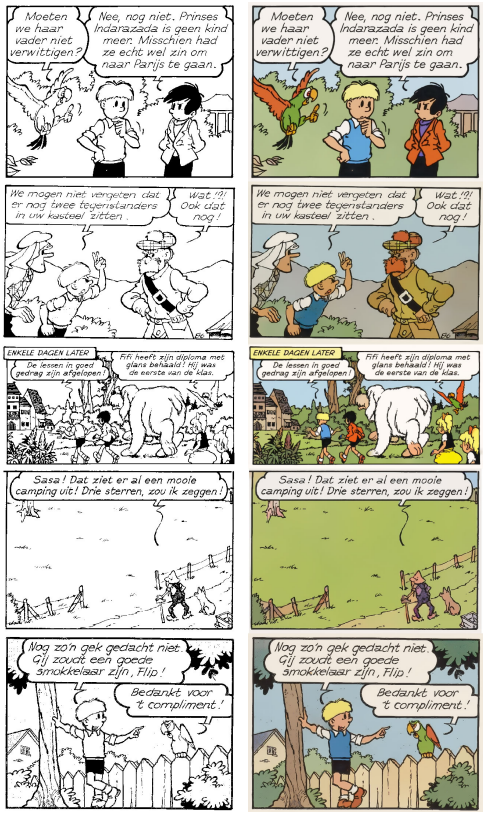
[4] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. 2018. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. *CoRR* abs/1811.12231 (2018). arXiv:1811.12231 http://arxiv.org/abs/1811.12231

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2672–2680. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

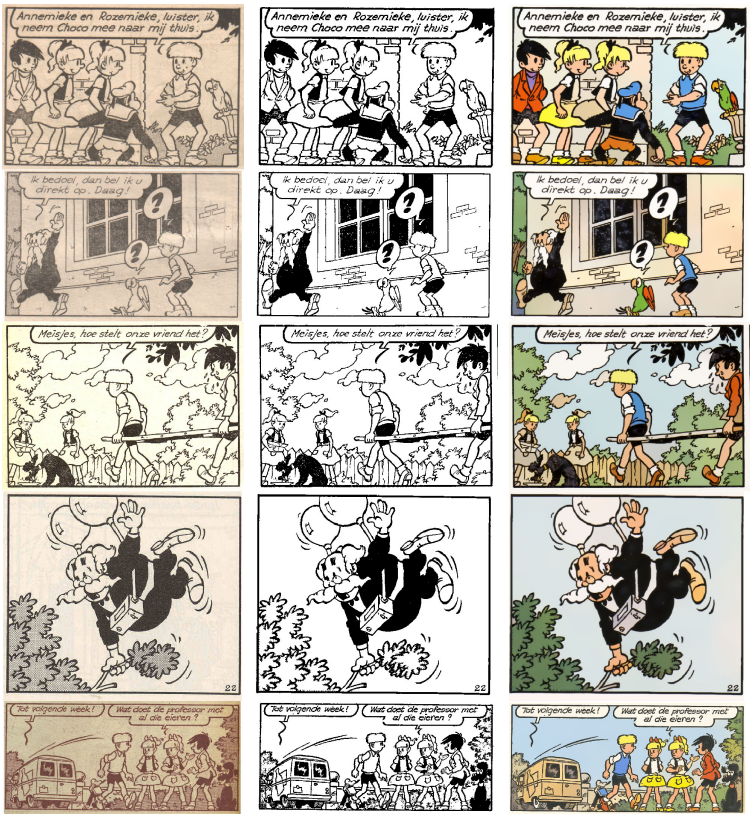
[7] Lifeng He, Xiwei Ren, Qihang Gao, Xiao Zhao, Bin Yao, and Yuyan Chao. 2017. The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognition* 70 (2017), 25 – 43. https://doi.org/10.1016/j.patcog.2017.04.018

[8] Aaron Hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, and David H. Salesin. 2001. Image Analogies. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*. ACM, New York, NY, USA, 327–340. https://doi.org/10.1145/383259.383295



(a) Binary Input (b) Output

Figure 6: Results on 'Color Comics'



(a) Original Scan (b) Binary Input (c) Output

Figure 7: Result on 'B&W Comics'

[9] Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu. 2005. An adaptive edge detection based colorization algorithm and its applications. 351–354. https://doi.org/10.1145/1101149.1101223

[10] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2016. Let There Be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization with Simultaneous Classification. *ACM Trans. Graph.* 35, 4, Article 110 (July 2016), 11 pages. https://doi.org/10.1145/2897824.2925974

[11] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR* abs/1502.03167 (2015). arXiv:1502.03167 http://arxiv.org/abs/1502.03167

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2016. Image-to-Image Translation with Conditional Adversarial Networks. *CoRR* abs/1611.07004 (2016). arXiv:1611.07004 http://arxiv.org/abs/1611.07004

[13] Zhangying Jin and Enyin Fang. 2018. *Print Inverse Halftoning and Its Quality Assessment Techniques.* 211–220. https://doi.org/10.1007/978-981-10-7629-9_26

[14] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *CoRR* abs/1603.08155 (2016). arXiv:1603.08155 http://arxiv.org/abs/1603.08155

[15] Ilya Kurilin, Ilia Safonov, Hokeun Lee, and Sang Kim. 2010. Descreening of scanned images. *Proceedings of SPIE - The International Society for Optical Engineering* 7528 (01 2010). https://doi.org/10.1117/12.838444

[16] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2016. Learning Representations for Automatic Colorization. *CoRR* abs/1603.06668 (2016). arXiv:1603.06668 http://arxiv.org/abs/1603.06668

[17] Simeng Li, Qiaofeng Liu, and Haoyu Yuan. 2018. Overview of Scribbled-Based Colorization. *Art and Design Review* 06 (01 2018), 169–184. https://doi.org/10.4236/adr.2018.64017

[18] Augustus Odena, Vincent Dumoulin, and Chris Olah. 2016. Deconvolution and Checkerboard Artifacts. *Distill* (2016). https://doi.org/10.23915/distill.00003

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR* abs/1505.04597 (2015). arXiv:1505.04597 http://arxiv.org/abs/1505.04597

[20] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved Techniques for Training GANs. *CoRR* abs/1606.03498 (2016). arXiv:1606.03498 http://arxiv.org/abs/1606.03498

[21] Evan Shelhamer, Jonathan Long, and Trevor Darrell. 2017. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 4 (2017), 640–651. https://doi.org/10.1109/TPAMI.2016.2572683

[22] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. 2016. Instance Normalization: The Missing Ingredient for Fast Stylization. *CoRR* abs/1607.08022 (2016). arXiv:1607.08022 http://arxiv.org/abs/1607.08022

[23] Freerk G. Venhuizen, Bram van Ginneken, Bart Liefers, Freekje van Asten, Vivian Schreur, Sascha Fauser, Carel Hoyng, Thomas Theelen, and Clara I. Sánchez. 2018. Deep learning approach for the detection and quantification of intraretinal cystoid fluid in multivendor optical coherence tomography. *Biomed. Opt. Express* 9, 4 (Apr 2018), 1545–1569. https://doi.org/10.1364/BOE.9.001545

[24] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2017. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. *CoRR* abs/1711.11585 (2017). arXiv:1711.11585 http://arxiv.org/abs/1711.11585

[25] Yuxin Wu and Kaiming He. 2018. Group Normalization. *CoRR* abs/1803.08494 (2018). arXiv:1803.08494 http://arxiv.org/abs/1803.08494

[26] Saining Xie and Zhuowen Tu. 2015. Holistically-Nested Edge Detection. *CoRR* abs/1504.06375 (2015). arXiv:1504.06375 http://arxiv.org/abs/1504.06375

[27] Fisher Yu and Vladlen Koltun. 2015. Multi-Scale Context Aggregation by Dilated Convolutions. *CoRR* abs/1511.07122 (2015). arXiv:1511.07122 http://arxiv.org/abs/1511.07122

[28] Richard Zhang, Phillip Isola, and Alexei A. Efros. 2016. Colorful Image Colorization. *CoRR* abs/1603.08511 (2016). arXiv:1603.08511 http://arxiv.org/abs/1603.08511