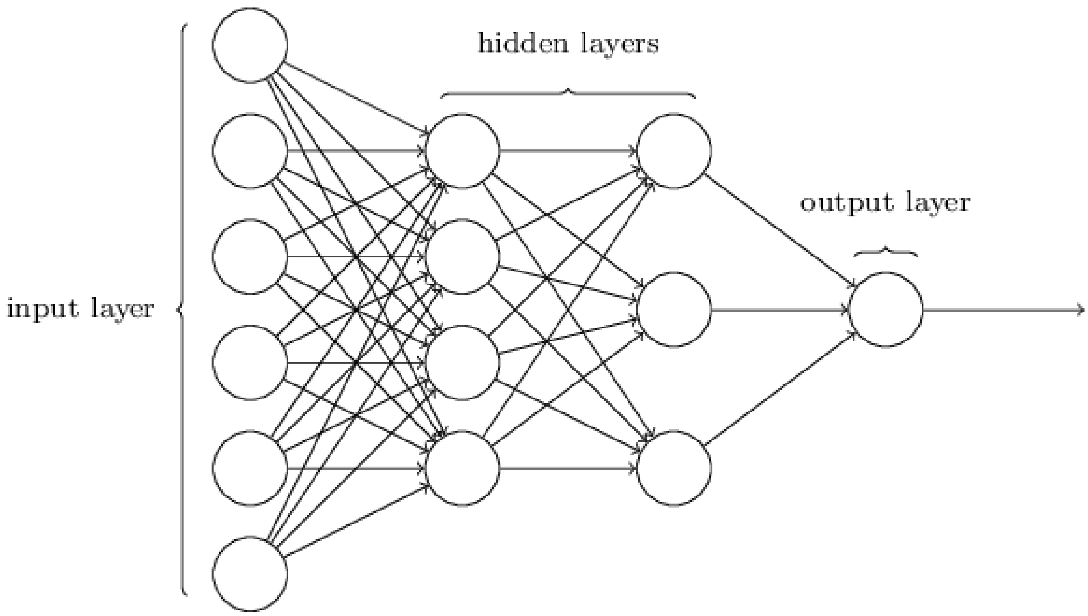
Delft University of Technology

IN4015 Neural Networks
Project description



Author:
J.L. Dorscheidt (4091981)
D.D. Hage (4190696)
D.E. van der Hoff (4139925)
J.F. Meulenbeld (4103548)

March 7, 2016

1 Introduction

The main problem that is going to be addressed is that of automatic image colorization. There is a substantial amount of black and white photographic material. Currently, these images are colorized, if at all, by hand using photo editing software, which is very labor intensive. However, color images may have a greater impact on people, which gives the need for automatic colorization algorithms.

In addition, automation of this process can be applied to real time colorization of video. Specifically (infra-red) surveillance cameras often save video in grayscale format. With the automated colorization techniques it may be possible to generate real-time color video, such that a human may more quickly understand what is seen in a video. In order to do this, it is required that the algorithm can colorize an image without intervention of a human.

As will be explained in Section 2, non-machine learning techniques are available to automatic image colorization. However, the drawback of these techniques is that they require either an image comparable to the grayscale image in terms of content, or user input on what color to use for different parts of the image. While this makes things much easier than the hand-made solution with photo editing software, it still requires a human to specify these inputs, which in turn disallows a real-time solution. This makes the case for a trained machine (i.e. neural network), which uses pre-supplied training data to learn, and while actually colorizing an image does not require additional input from a human. In addition, a large training dataset can easily be generated from color images.

2 Literature Review

Automatic image colorization is a problem that has been attempted to solve using different approaches. Separated by input type, there are three approaches to make the computer convert a grayscale image to a color image. Every approach uses the texture and intensity gradients of the image to link a part of an image to either a learned or specified color. The three approaches are:

Colorize by example: Image colorization can be performed by using a target image that is related to the grayscale image in that it contains similar objects with similar colors. The objects in the grayscale image are compared to the target image via the patterns of the similar objects to find what color the similar patterns should have. For example: to colorize a grayscale image of a zebra in the Savannah, another image of a zebra in a Savannah is required. The more similar the image, the better the result is. This method is used in [1], [2] and [3].

Colorize by user input: In this approach, the user specifies the color of different parts of the image by hand. While this is quite labor intensive, it guarantees that correct colors are used for the different objects in an image. For example, in the previous method the color of grass may be specified as green in the target image, whereas in the grayscale image the grass is actually colored brown. Due to the similarities in contrast the grass will thus be colored green, but in the colorize by user input this will not be a problem. However, if the user does not know the original colors of the grayscale image, colorization is not possible. This method is used in [4] and [5].

Colorize using a trained machine: Techniques like convolution neural networks allow training of a machine that recognizes specific patterns in an image and couples the recognized pattern to a color. This requires the use of training images of which both grayscale and color versions are available. Since no human intervention is required after the training is completed, this may lead to a vast decrease in time consumption during colorization. However, not all objects have the same color always, i.e. the same car model can have different colors, and a trained machine cannot identify what color a specific object is based on its class. It was found that a generic sepia tone results if enough examples of an object are in the training set. This method is used in [6], [7], [8] and [9].

This specific image task requires a deep neural network to be able to couple color to features in black and white images, most likely in the form of a convolutional neural network, as used in the literature. More literature about convolutional neural networks can be found in [10]. The biggest problem with the machine learning approach to automatic colorization until now, is the case where an object doesn't have a characteristic color, such as a car or clothing. To solve this problem it is proposed to use generative adversarial networks, to influence the discriminator to be able to generate the right colors where there could be multiple solutions. Generative adversarial networks is described by [11] and [12]. Another method to tackle this problem are variational Auto-encoders, altering a direct copy of the output, (variational) auto encoders are described by [13], [14] and [10].

3 Research Questions

The goal of this project is to design a neural network that can automatically color black and white images. The algorithm should match human colorization skills. Training data for the neural network is easy generating by applying a black and white filter on color images.

In order to find a solution to the automatic colorization of black and white images, a set of research questions have been identified:

1. What is a suitable input format which results in most efficient training of the network?
2. What is a suitable output format that results in the most realistic colorization of the images?
3. What cost function quantifies the resulting colorisation of the images as realistic as a human would quantify it?
4. What network architecture should be used in order to achieve the lowest cost function?
5. What training algorithm and parameters should be applied in order to achieve the fastest convergence rate?

These research questions will be more elaborated on in section 4.

4 Implementation steps

Different steps have to be taken in order to find answers to the research questions and to find a solution to the problem. These required steps are listed below.

1. **An extensive literature study.** It is of great importance to study the literature mentioned in 2. There are already a few well documented solutions for this problem, these need to be studied in-depth. It will become clear what the best network architecture to be used is and

whether it is within the scope of the project to build this network ourselves or to modify existing solutions.

2. **Determination of in and output data structure.** The size of the input and output images is of great importance to the complexity of the problem. That is, the more pixels the input and output images have, the more neurons are required for the network. It is therefore essential to determine the in and output data structure in the early stages of the project since this affects the total architecture of the neural network. Since limited computation power is available, a preliminary study should be done on estimating the effect of image size on computation time.

3. **Determination of the cost function.** The cost function is a crucial parameter in evaluating the performance of the neural network. It is therefore essential to determine a cost function in the early stages of the project.

4. **Determination of neural network architecture.** There are a lot of different possibilities as a choice for the Neural Network architecture. It has been shown that convolutional neural networks perform very well on the subject [9]. Determining the network architecture is one of the most essential steps, e.g. whether it used convolution, max-pooling, down sampling etc. and in what order. The amount of neurons in the input and output of the network should match the dimensions of the input and output data.

5. **Determination of neural network training algorithm.** The training algorithm has a large effect on the convergence of the network output. An efficient training method should find the weights and biases of the network that correspond to the highest performance function. Stochastic gradient descent updates the network after each sample, while for example the Levenberg-Marquardt method is a batch update method[15]. The training method should also be able to escape local minima, techniques such as added momentum can help solve this problem.

6. **Designing the neural network.** After the most important parameters have been determined, construction of the neural network can begin. This is the biggest phase of the project since it includes the programming and debugging of the chosen algorithms. A suitable programming language has to be selected for end-use in combination with a programming language for debugging and prototyping, Matlab would be a good candidate for the latter. Verification of code modules can already be done once modules are finished. Once all the modules are verified the total verification of the algorithm can be done.

7. **Training of the neural network.** Training of the neural network is one of most time-intensive phases. The determined training algorithm will be applied to the designed neural network. Different weight initializations and training parameters have to be chosen in order to result in the best convergence of the cost function of the network.

8. **Analysis of the results.** Once the network is trained, the performance of the network should be assessed. A measure of the quality of the network may be to what extent the colorized image is perceived by a human as realistic. This measure is less exact than for example the euclidean distance in color between the original color image and the colorized image. However, this kind of assessment leaves more room for the network to come up with realistic colors instead of the generic sepia tone that are seen on objects without a characteristic color, as explained in section 2.

5 Time Schedule

In table 1 a preliminary estimate of the time spent on each subject is made. There is no division of labor present which will be determined in a later stage of the project.

The amount of time spent on the literature study is a direct copy of the amount of ECTS available for the literature study. The remaining amount of time is also based on the assigned number of ECTS. This resulted in a total time to divide of 470.4 hours.

The determination of preliminary design choices will take up a considerable amount of time. Included in this preliminary design is an extensive literature study which is required to get sufficient knowledge about the subject to make these preliminary design choices. It is assumed that once this information is gained, deciding on the design choices only takes a small amount of time. The preliminary design choices expected to be answered in this phase are:

1. **Design of in and output data structure.**
   The design of the Neural Network architecture has a great dependency on the input and output data structure. Only when in and output parameters such as number of pixels and color channel type are being set, the design of the Network can begin.

2. **Determination of cost function.**
   The cost function is the crucial parameter in determining the performance of the network. It is therefore essential that a cost function is determined in the early stages of the project.

After these parameters have been established the actual design of the neural network has to begin. This is where most of the time assigned to the project is going to be put in. The design of the Neural Network architecture is determined to be the most time intensive task. This task can be done simultaneous with designing the best training algorithm, e.g. once a preliminary network is constructed, training methods can be designed and evaluated using this preliminary network.

Table 1: Time allocation

Task / time [hours]	Per person	total time
Literature Study.	19,6	78,4
Determination of preliminary design choices.	-	-
Determination of the in and output data structure.	2	8
Determination of the Neural Network Architecture.	3	12
Determination of the cost function.	1	4
Determination of Neural Network Training Algorithm.	1	4
Design of the Neural Network	-	-
Design of the in and output data structure.	15	60
Design of the Neural Network architecture.	35	140
Design of the Neural Network Training Algorithm.	10	40
Verification Validation of the neural network	5	20
Others(System integration, optimization, debugging, etc)	26	104
Total Time (h)	117.6	470,4

References

[1] G. Charpiat, M. Hofmann, and B. Schlkopf, *Automatic image colorization via multimodal predictions*, pp. 126–139. Springer, 2008.

[2] 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*, pp. 369–378, ACM.

[3] Y. Zheng and E. A. Essock, "A local-coloring method for night-vision colorization utilizing image analysis and fusion," *Information Fusion*, vol. 9, no. 2, pp. 186–199, 2008.

[4] T. Horiuchi, "Colorization algorithm using probabilistic relaxation," *Image and Vision Computing*, vol. 22, no. 3, pp. 197–202, 2004.

[5] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 689–694, ACM.

[6] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 415–423.

[7] H. Ho and V. Ramesh, "Automatic image colorization,"

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105.

[9] R. Dahl, "Automatic colorization." `http://tinyclouds.org/colorize/`, 2016. [Visited on: 04-03-2016].

[10] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning." Book in preparation for MIT Press, 2016.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672–2680.

[12] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[13] K. Gregor, I. Danihelka, A. Graves, and D. Wierstra, "Draw: A recurrent neural network for image generation," *arXiv preprint arXiv:1502.04623*, 2015.

[14] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[15] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *Neural Networks, IEEE Transactions on*, vol. 5, no. 6, pp. 989–993, 1994.