


Carpets Color and Pattern Detection Based on Their Images [†]

Sayedeh Marjaneh Hosseini ^{1,‡}, Ali Mohhamad-Djafari ^{2,‡}, Adel Mohammadpour ^{3,*}, , Sobhan Mohammadpour ^{4,5,‡} and Mohammad Nadi ^{6,‡}

¹ Department of Computer Science, Amirkabir University of Technology, Tehran, Iran; marjanehosseini@aut.ac.ir

² Laboratoire des Signaux & Systèmes (Supélec, CNRS, Université Paris Sud), 91190 Gif-sur-Yvette, France; ali.mohammad-djafari@l2s.centralesupelec.fr

³ Department of Statistics, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

⁴ Department of Computer Engineering, Sharif University of Technology, Tehran, Iran; sobhan.mohammadpour@umontreal.ca

⁵ Department of Computer Science and Operation Research, Université de Montréal, Montreal, QC H3T 1J4, Canada

⁶ E-CarpetGallery, Montreal, QC H4E 4N7, Canada; mohammad.nadi@gmail.com

* Correspondence: adel@aut.ac.ir

† Presented at the 39th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Garching, Germany, 30 June–5 July 2019.

‡ Authors' names are alphabetically arranged, the 4th author contributed 40% to this work.

§ Present address: Department of Mathematics and Statistics, McGill University, Montreal, QC H3A 0B9, Canada.

Published: 24 December 2019



Abstract: In these days of fast-paced business, accurate automatic color or pattern detection is a necessity for carpet retailers. Many well-known color detection algorithms have many shortcomings. Apart from the color itself, neighboring colors, style, and pattern also affects how humans perceive color. Most if not all, color detection algorithms do not take this into account. Furthermore, the algorithm needed should be invariant to changes in brightness, size, and contrast of the image. In a previous experiment, the accuracy of the algorithm was half of the human counterpart. Therefore, we propose a supervised approach to reduce detection errors. We used more than 37,000 images from a retailer's database as the learning set to train a Convolutional Neural Network (CNN, or ConvNet) architecture.

Keywords: color detection; pattern detection; carpet images; convolutional neural network

1. Introduction

The wave of digitalization is taking over many business areas; carpet selling is no exception. With its recent exponential growth, electronic carpet retailers are selling more carpets than ever. Like most businesses, these retailers need to automate their process, which typically involves getting a container of carpets, tagging them and adding them to an online catalog. While the first and the last phase have already been automated, due to the lack of a common and extensive standard between major carpet exporters from major hand-turfed carpet hubs like Iran, India, and Afghanistan, many retailers have to employ carpet experts to generate the metadata visually. Current literature does not seem to provide any acceptable computational model that requires a minimal change in the production pipeline. We took e-Carpet Gallery (www.ecarpetgallery.com) (e-CG) work-flow as an example and tried to automate its second step.

E-CG's work-flow consists of three steps. First, a container full of carpet arrives at the warehouse, the employees unload the container, take pictures of the rugs and fill out the empty fields of features of carpets for each carpet one by one then move them to storage. Due to the volume of operation, employees have a limited time to tag the carpets. The repetitive nature of the task mixed with its complexity makes either the process slow(multiple checks) or very error-prone. The markets have started automating the more deterministic and less opinionated parts of the process like the measurement of weight, width, length, and shape of carpets. However, there is no method to classify nominal features like material and quality of the carpets. Accurate determination of qualitative or hybrid features such as color, pattern, and style, are essential for shoppers but is very difficult. While this process has matured over the years, its accuracy much to desire. In this work, we apply a few selected image processing algorithms to e-CG's database. This e-CG is a production-level database with a reasonable error rate and contains fields like top-down image, pattern, and color (main color). We focus on the automatic labeling of two qualitative features: color and pattern. Well-known and intuitive color detection method such as the pixel count of clustered image's pixels fail due to the difference between a visual human color assessment of a carpet(or any other abstract image) and, e.g., counting the number of high-frequency pixels. This is due to the association between color and pattern in a carpet. A reoccurring example is that the color of a carpet with a big red flower in its center is usually red based on a human assessment, even when most of the carpet is a lighter color like beige. To the authors' knowledge, CNNs are not yet applied as a classification tool for patterns and colors of carpets. However, CNN applies in color detection [1], for example, in the vehicle color recognition [2]. In the next section, we introduce two features of carpets and their levels (or labels). In Section 3, we point out a few technical issues. We apply a few classification methods as well as CNN for pattern and color.

2. Database and Features

E-CG has a database with more than one million carpet images. We only consider a sample of 37,000 carpets from the last two years to speed up the classification process. We introduce three features: color, pattern, and style via frequency bar charts. The level names of each feature are listed alphabetically in the corresponding graph. Figure 1 shows color levels and a sample of images to demonstrate color levels in a real database. One can find that the red color has a high-frequency level and is one of the unbalancing element of the database. Levels' frequency chart of patterns is plotted in Figure 2. The number of pattern levels is fixed based on the dataset. Figure 3 illustrates sample image patterns and their corresponding edges to clarify the different patterns.

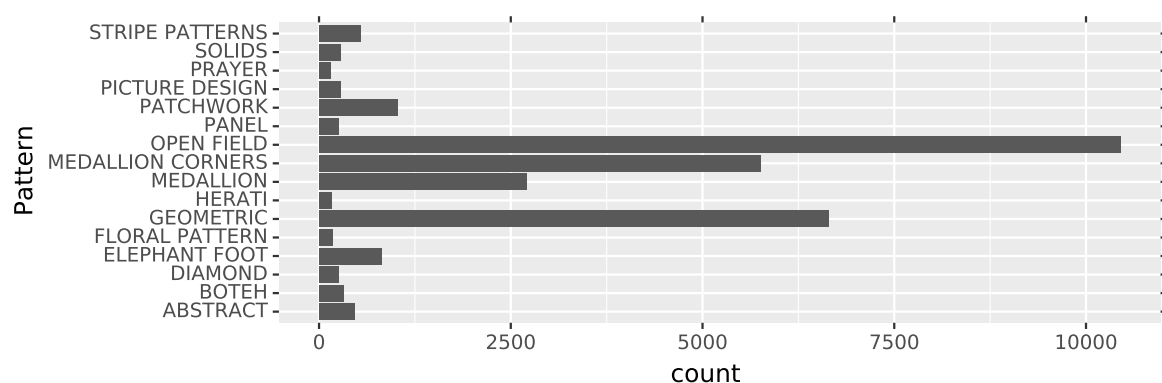


Figure 1. Frequency bar chart of pattern levels. Level names are sorted alphabetically.

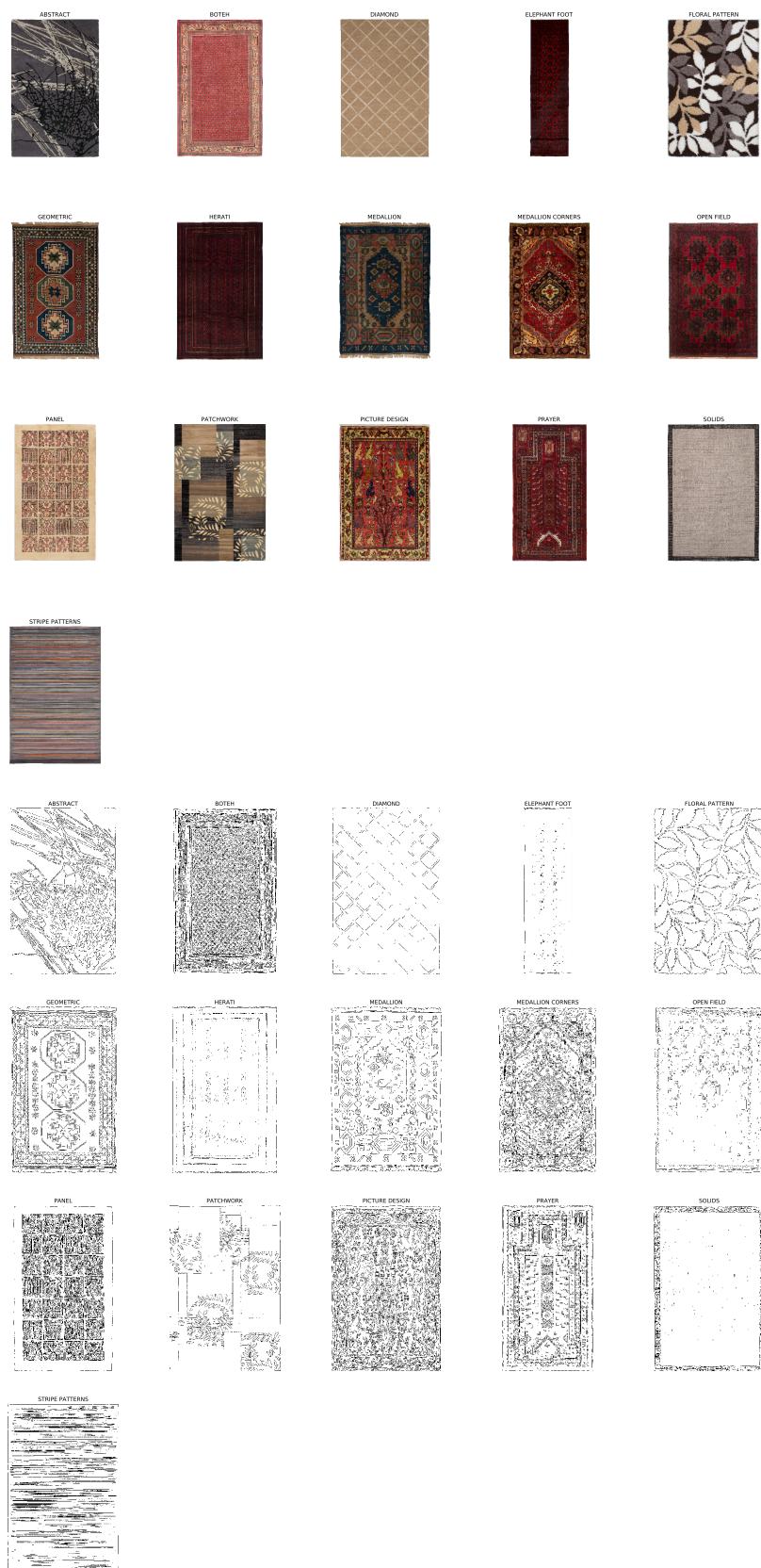


Figure 2. (Top): Sample carpet with their pattern levels. (Bottom): Corresponding edges of top images.

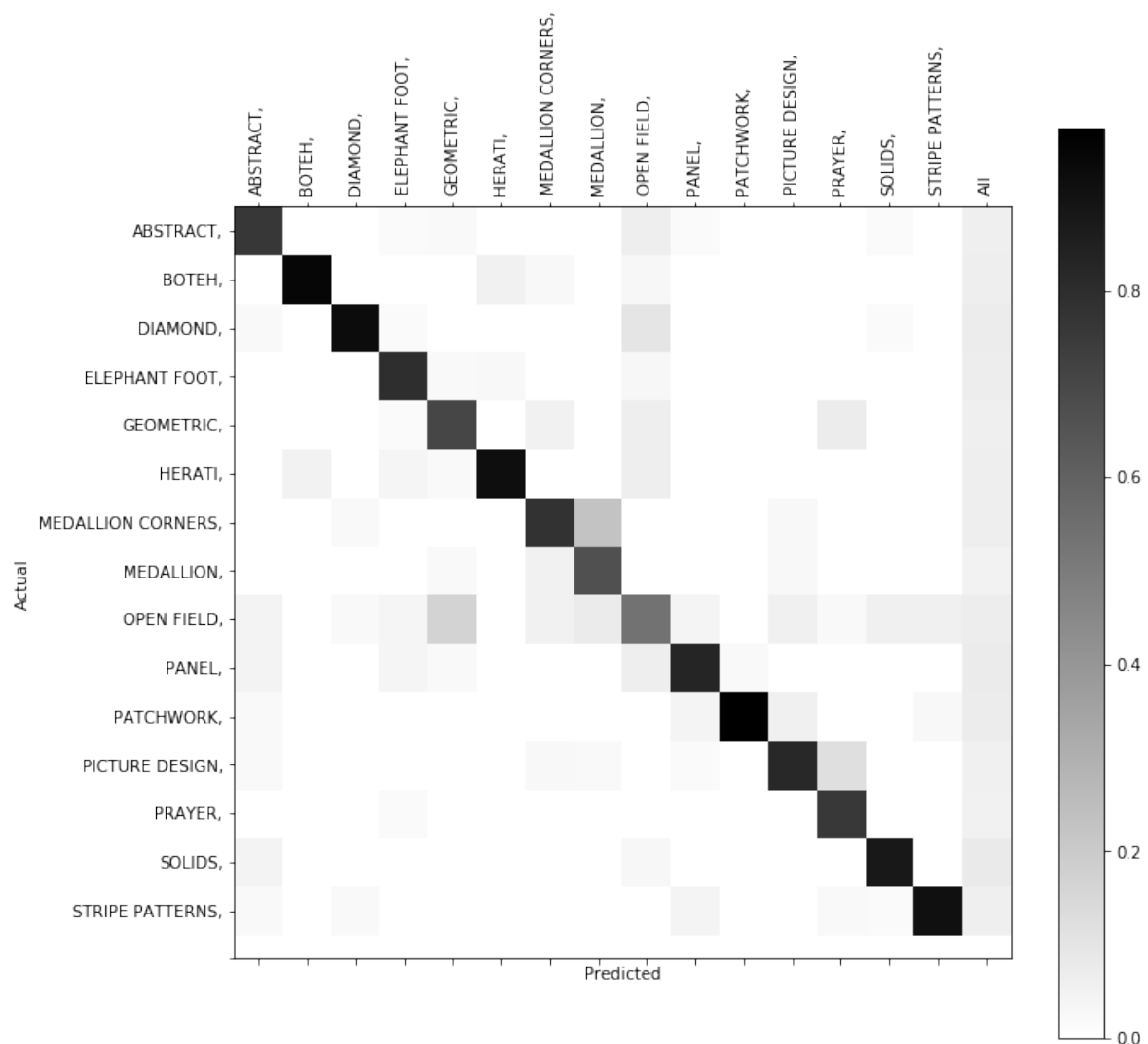


Figure 3. Pattern classification confusion matrix.

We consider a sample of recorded data that each feature level had at least 100 occurrences in our selected time frame. While 100 may have arbitrarily selected, this ensures that we do not have to deal with features and categories that do not often appear in production, thus reducing the model complexity and the error rate introduced by using unbalanced data.

3. Classification

Before turning to the methodology of classification algorithms, we point out to essential technical issue:

- Color accuracy has always been an issue for professional photographers, designers, and printers to deal with it on a daily basis. The images we got were not very uniformly photographed. They were not very uniformly lighted, with the top of the image being the brightest and bottom being darkest.
- The size of carpets are very irregular; this makes the use of CNN rather hard because runner carpets will end up with a lot of white space.

3.1. Pattern

For pattern, we decided that it would be reasonable to start with CNN. Since we did not have the computational power to train a CNN from scratch, we used the Xception architecture [3], with pre-trained weights on ImageNet, a model with a high top 1 accuracy on ImageNet, which seems good at understanding patterns. Xception architecture has fewer parameters and more accuracy with respect to a few tested architectures. However, one may find a better CNN architecture with high-level hardware. We removed the top layers and added two dense layers, one with twice as many nodes as the number of patterns with an ELU (Exponential Linear Unit) activation and a softmax layer to classify. We trained the network with an Adam optimizer [4], for 60 epoch with randomly rotated, sheared, and flipped images. Due to the unbalancedness of the database, we decided to under-sample everything. It's worth noting that patterns that are similar or not well defined like "open field" and "diamond" tend to get misclassified often by humans and machines.

In Figure 4, we report the percentage of classified test dataset that was 20% of database images; this figure present pattern classification confusion matrix. The actual and predicted values are row and column of the confusion matrix, respectively. We make a column-wise normalization for a better interpretation. The last column shows the sum of misclassified that can be considered as a bias indicator. One can observe that the diagonal of the matrix is bolded that shows the performance of the proposed algorithm is more than 80%.

3.2. Color

We Used *K*-means clustering algorithm to find a value representing each image, then run a classification algorithm, AdaBoost [5], on the result. This two-step classification method is used to speed up AdaBoost algorithm. The accuracy was around 45%. Three contributing factors where observed:

1. Colors are tightly stacked, and very similar colors like ivory and beige or dark copper and red get misclassified often,
2. The image was not calibrated.
3. The majority does not always mean most dominant as having a small area of a dominant color like red or black will result in a red or black carpet. Therefore, we decided to reuse the network from Pattern Recognition.

With all of this in mind were-trained the CNN we used for pattern recognition and got a similar result to the two steps classification. As we had expected, the network often confused similar colors like the previous experiment. It's possible that unless the pictures are very well calibrated nothing can give an accurate result. The results were acceptable despite the low accuracy because color ranges are not really well defined and the network mostly confused resemblant colors like beige and ivory.

The confusion matrix of CNN for color, similar to the pattern feature, is plotted in Figure 5. We can observe that the performance is similar to the above two steps *k*-means-AdaBoost algorithm is less than 50%.

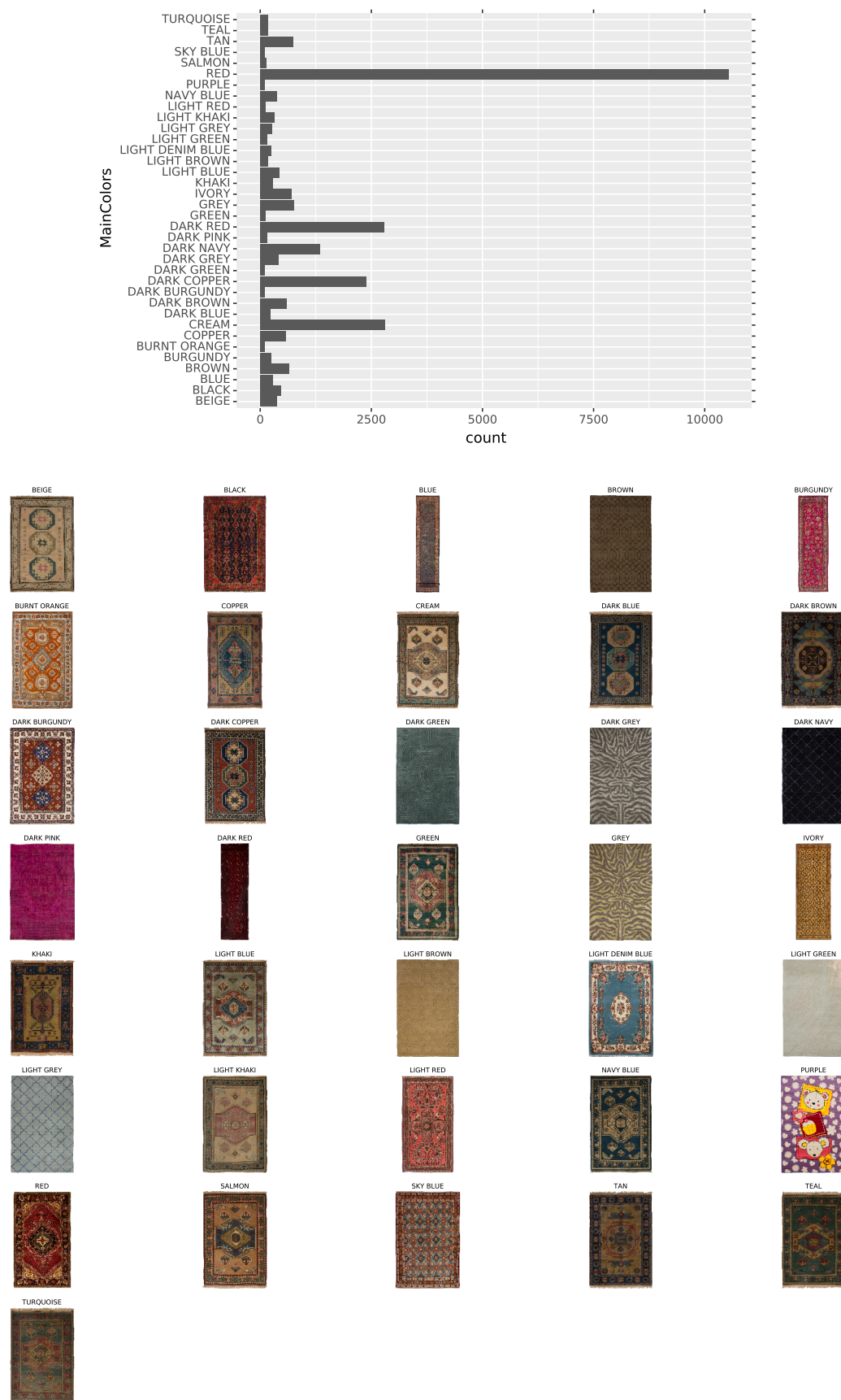


Figure 4. Frequency bar chart and a sample of color level labels. Label names are sorted alphabetically.

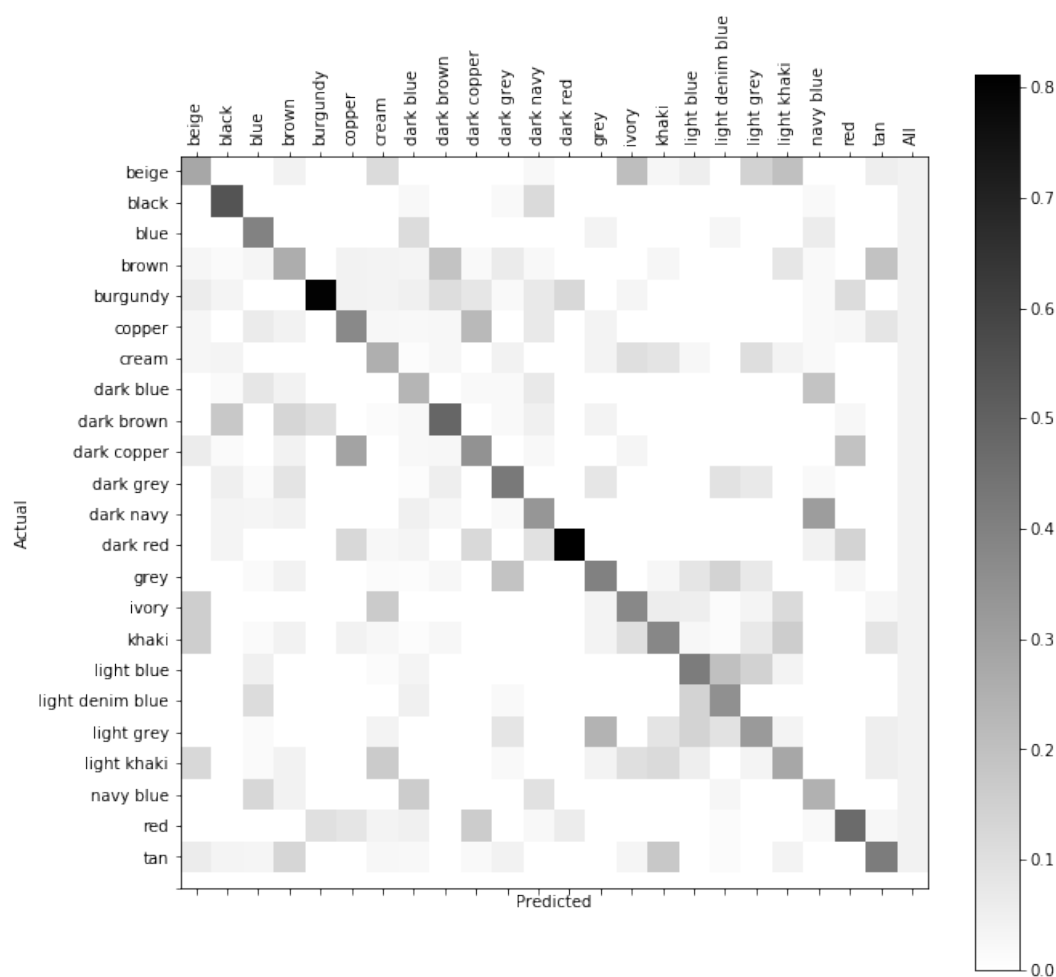


Figure 5. Color classification confusion matrix.

Acknowledgments: The authors would like to thank two anonymous referees for their helpful comments and for careful reading that greatly improved the article. We thank E-CarpetGallery for access to their database and support of three last authors during their research.

References

1. Luong, T.X.; Kim, B.; Lee, S. Color image processing based on nonnegative matrix factorization with convolutional neural network. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014; pp. 2130–2135.
2. Zhang, Q.; Zhuo, L.; Li, J.; Zhang, J.; Zhang, H.; Li, X. Vehicle color recognition using multiple-layer feature representations of lightweight convolutional neural network. *Signal Process.* **2018**, *147*, 146–153.
3. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.
4. Kingma, D.P.; Ba, J.L. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representation (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
5. Schapire, R.; Freund, Y. *Boosting: Foundations and Algorithms*; MIT: Cambridge, MA, USA.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).