Scalable Visual Attribute Extraction through Hidden Layers of a Residual ConvNet

Andres Baloian
DCC-University of Chile
Av. Beauchef 851, Santiago, Chile
abaloian@dcc.uchile.cl

Nils Murrugarra-Llerena
Snap Research
2772 D. Douglas L. N, S. Monica, CA, US
nmurrugarraller@snap.com

Jose M. Saavedra
Impresee Inc.
600 California, San Francisco, US
jose.saavedra@impresee.com

Abstract

Visual attributes play an essential role in real applications based on image retrieval. The traditional manner to build an attribute extractor is by training a convnet-based classifier with a fixed number of classes. However, this approach does not scale for real applications where the number of attributes changes frequently. Therefore in this work, we propose an approach for extracting visual attributes from images, leveraging the learned capability of the hidden layers of a general convolutional network to discriminate among different visual features. We run experiments with a resnet-50 trained on Imagenet, on which we evaluate the output of its different blocks to discriminate between colors and textures. Our results show that the second block of the resnet is appropriate for discriminating colors, while the fourth block can be used for textures. In both cases, the achieved accuracy of attribute classification is superior to 93%*. We also show that the proposed embeddings form local structures in the underlying feature space, which makes it possible to apply reduction techniques like UMAP, maintaining high accuracy and widely reducing the size of the feature space.*

1. Introduction

Advances in deep learning have made it possible to bring innovative solutions to the industry, particularly with solutions that involve computer vision. Searching by text is the primary querying modality used by eCommerce search engines. However, its effectiveness depends on a detailed description of the product in a store. Also, to make the querying process easier, some engines include the alternative of searching by images [4]. A special kind of this modality is when users express their intention through a drawing that leads the sketch-based image retrieval task [9, 1]. In fact, sketches represent a natural way of communication between human beings, and mobile devices' massification powers its use.

Visual attributes play an essential role in eCommerce, as they represent specific information of what users are needing. Unfortunately, many of the products we find in eCommerce are not described enough, lacking information as color and texture, among others. Consequently, the accuracy of the retrieval results decreases. Therefore, including a visual attribute extractor within the search engine increases the chance the products fit the user's query. Visual attribute extraction benefits not only the searching by text modality but also the searching by images. Extracting attributes from image queries can produce more accurate results as attributes provide a high level of semantic of a user's desire.

We can build a visual attribute extractor by collecting images representing the attribute set we are interested in extracting and then training a model with the collected data. The problem with this simple approach is that it requires a large set of images, and it does not scale to a dynamic set of attributes. If we need to incorporate new attributes, we will need to retrain the model with the new data, which is not feasible in real-world applications.

Therefore, the contribution of this work is to propose a scalable method for extracting visual attributes from images based on convolutional networks. Instead of training a specific network to fit a fixed number of attributes, we leverage the capability of a convolutional network to learn visual information through its hidden neurons when it is trained in

a large dataset like Imagenet [3]. To this end, we use a ResNet-50 [5] as the general convnet.

As we a priori do not know how different layers behave in front of different attributes, we conduct different experiments to determine the hidden layer's capability to classify visual information. We present a study for two types of visual attributes, color, and textures.

To produce a trainless classifier, we take advantage of the kNN classifier. In this way, we will need to collect few image examples associated with each attribute we are interested in classifying. Then, the attribute class of a new image is inferred by the representative class of its nearest neighbors, so the space generated must be characterized by forming local structures grouping elements of the same class.

This document is organized as follows. Section 2 is devoted to describing our approach in detail. Section 3 describes the evaluation experiments, and finally Section 4 presents the conclusions.

2. Approach

2.1. Dataset

We collected two datasets using Kaggle[1] and online resources, one grouped by color and the other by texture.

- **Color dataset**: We gather 100 clothing images from each of the following colors: red, black, blue, green, yellow, gray, brown, pink, purple, and orange.
- **Texture dataset**: We gather 100 clothing images from each of the following texture patterns: squared, striped, flowers, leopard, polka, basic, paisley, argyle, crows feet, and sequin; which are depicted in Figure 1. Our dataset is compiled from diverse clothes ranging from socks to shirts.

2.2. Feature extraction

We selected the outputs of the five blocks of a resnet-50 [5] trained on ImageNet, using the Keras framework [2]. For the sake of clarity, we name *Block 1* the first convolution layer, and *Block 2* to *Block 5* are the residual blocks.

To create a compact representation of the features, we apply a global average pooling over spatial dimension at the output of each block. Hence, we generate vectors of sizes 64, 256, 512, 1024, and 2048. These compacted vectors are feed to a machine-learning algorithm in the following section.

2.3. Classification

For the classification task, we choose the kNN method over the generated features space by the residual blocks' outputs. In this way, we do not need to train the model each time a new attribute is added. Instead, we only need to collect some examples of the target attribute, and the classifier will infer the class of an image as the representative class among its nearest neighbors.

[1]https://www.kaggle.com/paramaggarwal/fashion-product-images-dataset

(a) Agryle (b) Basic (c) Crowsfeet (d) Flowers (e) Leopard

(f) Paisley (g) Polka (h) Sequin (i) Squared (j) Striped

Figure 1: Ten different textures used in our dataset.

3. Experimental Evaluation

In this section, we provide insights on how to identify network blocks with texture and color. First, we provide details of our experimental setup. Second, we use output blocks to train a classifier for color and texture and find the most accurate one. Finally, we visualize images among the blocks to observe how patterns are grouped.

3.1. Setup and metrics

We evaluate our classification experiments using a stratified 5-fold cross validation with accuracy metric and a kNN classifier[2]. While for clustering, we use HDBSCAN [7] on the whole dataset with Euclidean distance, $min\_cluster\_size$ of 50, $min\_samples$ of 10 and $leaf\_size$ of 40. We evaluate performance using adjusted rand index [6] and adjusted mutual information score [10].

3.2. Quantitative experiments

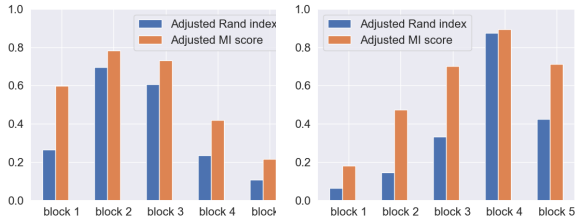
We show the behavior from outputs of five different blocks of our baseline network for grouping similar colors and textures. First, We show accuracy values for a kNN classifier in Table 1. For color, block two is the most discriminative (higher accuracy), while, for texture, block fourth is the most effective.

Also, we perform a clustering study in Figure 2. For color, we observe that that clustering metrics start increasing until block two, and then decrease consecutively in

[2]k=5 is our best parameter in our preliminary experiments

#Block	Color	Texture
Block 1	0.900	0.587
Block 2	**0.937**	0.826
Block 3	0.929	0.915
Block 4	0.886	**0.939**
Block 5	0.784	0.925

Table 1: Accuracy achieved by the hidden layers of a resnet-50 [5] in the color and texture classification task using kNN with k=5



(a) Color (b) Texture

Figure 2: Cluster metrics for color and texture features among the first five blocks on a resnet-50 [5].

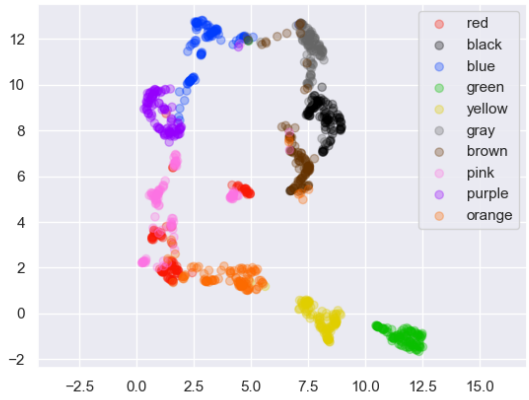


Figure 3: UMAP visualization for color features on block two on a resnet-50 [5]

blocks three to five. On the other hand, for texture, we observe that clustering metrics increase from block one to block four.

These insights confirm that neural networks learn low-level concepts in initial layers (e.g. color), and intermediate concepts in upper-level ones (e.g. texture).

3.3. Qualitative experiments

In order to analyze the internal behaviour of the blocks, we visualize the feature vectors using UMAP [8] on Figures 3 and 4 for color and textures, respectively.

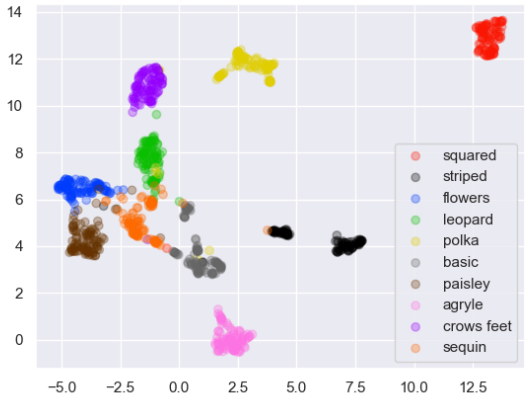


Figure 4: UMAP visualization for color features on block four on a resnet-50 [5]

These visualizations only depict block two and block four for color and texture features, respectively. These blocks are the ones with higher performance, and we can identify clear group frontiers. For color, we observe yellow and green colors from their cohesive groups and easily differentiate them from other colors. Other colors are still cohesive, however, boundaries with their neighbor groups' colors are less clear. Similarly, block four for textures identify squares, stripes, argyle, and polka as the best cohesive and clear groups.

We can also leverage the local topology produced by both feature spaces to apply a locality-preserving dimension reduction technique. Our experiments show that the feature spaces for color and texture reduced to 8 dimensions by UMAP [8] achieve an accuracy of 0.901 and 0.910, respectively.

4. Conclusion

We described an approach to identify semantic attributes by using different layers of a pre-trained residual neural network. We find that block two of a resnet-50 is effective for color classification, while block four is effective for textures.

Additionally, we show that a locality preserving dimension reduction technique like UMAP allows us to work with a low-dimension feature space obtaining still high performance.

References

[1] Tu Bui, Leonardo Ribeiro, Moacir Ponti, and John Collomosse. Sketching out the details: Sketch-based image retrieval using convolutional neural networks with multi-stage regression. *Computers & Graphics*, 71, 2018. 1

[2] François Chollet. Keras, 2015. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009. 2

[4] Shiv Ram Dubey. A decade survey of content based image retrieval using deep learning, 2020. 1

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2, 3

[6] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985. 2

[7] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205, 2017. 2

[8] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018. 3

[9] Jose M. Saavedra and Juan Manuel Barrios. Sketch based image retrieval using learned keyshapes (LKS). In *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015*, pages 164.1–164.11, 2015. 1

[10] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.*, 11:2837–2854, Dec. 2010. 2