Data Science Case Study

# Improving eCommerce Internet Search with Computer Vision

*By working with Mosaic, this leading retailer leveraged cutting-edge AI vision models to allow users to customize their web searches, finding similar products more efficiently & effectively.*

---

**Industry**

Retail & Ecommerce

**Use Case**

Improving Customer Experience

**Techniques**

Deep Learning & Computer Vision

**Outcome**

Better Site Searchability

## INTRODUCTION

Filtering search results is an essential part of any eCommerce website. Can you remember the last time you shopped online without filtering on product attributes such as color, size, brand, etc. Additionally, rich product attributes are critical to Google SEO which drives traffic and website sales.

Mosaic's client, a leading retailer, has many of these product attribute filters on their website, but they felt they were missing at least one crucial filter - garment pattern. Garment pattern refers to solid, striped, graphic-T, etc. The retailer's problem was, how to filter on garment patterns if vendors do not usually supply with this information as part of the product description? Mosaic's answer was with deep learning-based image classification.

Typically to build a successful deep learning model, one need thousands of examples per class one wishes to detect. In some cases, it would be preferable to have more to get the desired level of accuracy, reaching tens of thousands of labels per class. Most of the time, though, there are not enough resources available to devote to such a time-consuming labeling effort.

The retail firm found themselves in this predicament. They had only a few thousand (~3K) total labels for products they previously offered, internally generated, across all nine distinct classes that they wished to include in the filter. For this regime, transfer learning is an appealing solution.

## TRAINING A DEEP LEARNING MODEL WITH FEW EXAMPLES

The idea behind transfer learning is to take lessons learned from a general problem, for which one has plenty of training examples, and then apply these lessons to the issue at hand where one may have few training examples. Usually, the initial learning step takes advantage of extensive and high-quality publicly available datasets. Transfer learning reduces, or in some cases eliminates, the burden and need for many training examples.



We like this image produced by Towards Data Science.[1]
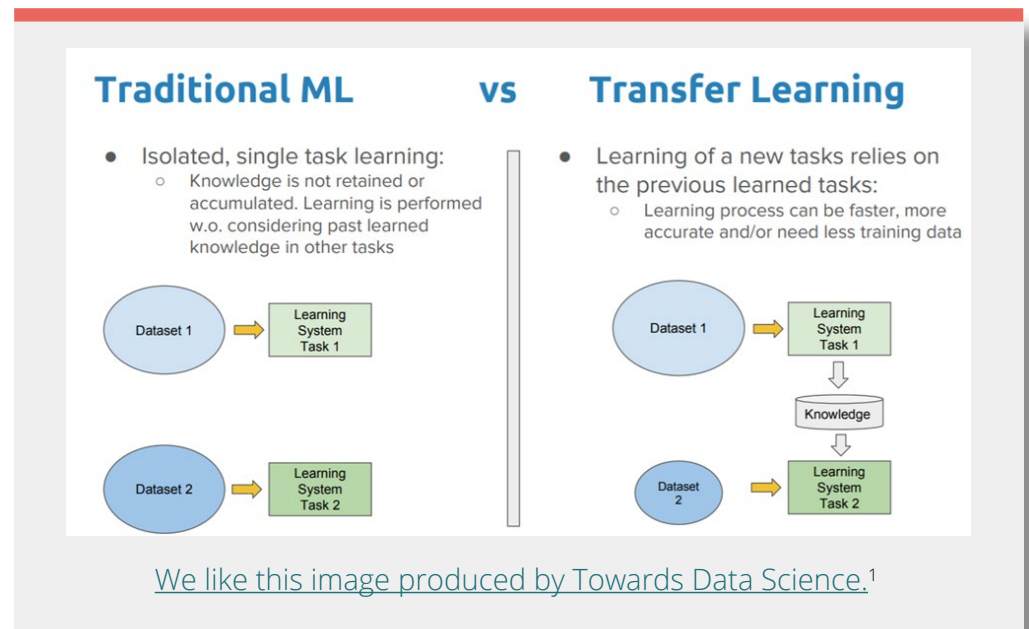
It may seem complicated to implement a transfer learning approach for deep learning classification; however, Keras dramatically simplifies this task. Due to the popularity of image classification Keras conveniently provides several pre-trained

models and architectures as a part of their package. This blog post does an excellent job for demonstrating how to set up transfer learning on image classification in Keras.

In conjunction with the retailer, Mosaic prototyped a few different algorithms to learn on the sparse training examples. After testing several approaches, Softmax logistics function and an Xception model were deployed because the good tradeoff between size (and speed to train/predict) and accuracy on benchmarks.

# BOOSTING PERFORMANCE

The first cut of implementing this approach provided surprisingly good results, leading to an average accuracy of about 70% for some classes. However, a few of the classes performed relatively poorly, with precision <50%. Although this was encouraging, it was not near enough established accuracy goals.

To improve accuracy, Mosaic's computer vision experts focused on two main areas: 1) Data labeling improvement and accuracy, and 2) model architecture.

1. Data labeling improvement and accuracy

    a. Errors in the initial model helped prioritize which classes to focus on in terms of gathering more labels. This ensured that any additional labeling effort was maximally helpful. This was important because most of the labeling was donated time from client stakeholders with no time to give. Additionally, some

classes were often confused with other specific classes in the initial model. This led to an investigation into the criteria labelers used for some of the classes. For example, many solid-colored objects were labeled as colorblock when they contained a small patch of color. Improvements to the labeled sets led to the most significant gain in performance.

2. Modeling architecture

    a. Mosaic prototyped and presented several architecture variations. Mosaic varied the number of hidden layers between the pre-trained and output layers, the number of nodes in these hidden layers, the activation functions in these layers, the loss function, the solver, and augmentation methods for data images. None of these gave significant improvements over the baseline. In the end, the only architectural improvement that resulted in substantial performance improvements was adding a secondary input to the images to make use of product color descriptions. Rarely was actual pattern information provided for products, but sometimes products would contain some other hints for the type of pattern it was. For instance, colorblock shirts would contain references to multiple colors, whereas solid colors would only have one color in the description. Mosaic used one-hot-encoding of these descriptions as input into a secondary feed-forward neural network to incorporate this information. The final layers merged the output of this feed-forward network with the pretrained Xception model's production, improving overall confidence and taking advantage of both descriptions and visual cues.

# HUMAN-IN-THE-LOOP IMPROVEMENTS

After improving the computer vision model, some classes still performed slightly less than desired acceptance criteria. To address this Mosaic's data scientists made use of a voting strategy. The retailer often has many sub-types (and associated images) associated with a specific product; by voting on the dominant class for a particular product (across color codes), Mosaic could further improve accuracy.

# RESULTS

All put together, each class gave near or above 90% accuracy in the hold-out test set, which was well within acceptance criteria. Additionally, because each product has a likelihood score, products with higher confidence for a particular pattern can be displayed first, reducing users' likelihood of seeing misclassifications, improving the customer experience.

The results can be seen in the screen-shot below.
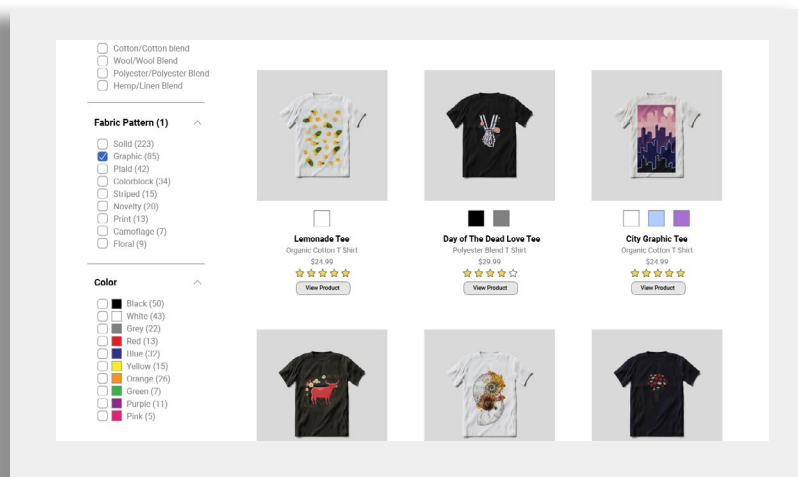


One area for improvement is how the retailer can display the results to a website visitor. Currently, a garment may get through a filter where only one of the product colors has the specified pattern. For example, one color for a product may be a striped pattern, but the rest are solid.

Including this product in the filter is not wrong; however, the default display image is usually the first color listed in the database, not the striped shirt. Improvements in how the results are displayed will go a long way in improving the overall customer experience. Mosaic developed the model with an eye towards future improvements, in the form feedback from users and internal employees, as misclassification are fed back into the original model for fine-tuning.

## Endnotes

1. https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a?gi=33287fca270b